DISTILLING PRODUCTIVE DISAGREEMENT: COLLISION-AWARE ON-POLICY DISTILLATION FOR SELF-CORRECTING LANGUAGE AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Teacher disagreement is useful for on-policy distillation only when it exposes a decision-sensitive alternative that can be verified and acted on from the student's current state. We study this conditional learning signal and introduce Collision-Aware On-Policy Distillation (C-OPD). For each student trajectory, an axiom teacher identifies implicit assumptions, a counterfactual teacher minimally inverts high-risk assumptions, and task-native verification adjudicates the resulting branches. A Productive Collision Score (PCS) combines disagreement, counterfactual utility, evidence consistency, recoverability, and training-trajectory reward improvement, then attributes additional distillation weight to an intervention-linked reasoning span. Across three seeds on SWE-bench Verified, BFCL v3.0.1, MATH-500, and HotpotQA v1.1, C-OPD reaches $49.6 \pm 0.4\%$ macro-average success, compared with $41.8 \pm 0.4\%$ for standard on-policy distillation and $38.7 \pm 0.4\%$ for response-level knowledge distillation under the same 800M teacher-token budget per seed. On SWE-bench Verified, resolved instances increase from $32.4 \pm 0.4\%$ to $38.9 \pm 0.3\%$. On adversarial tool observations, unsupported continuation falls by 28.4%, while recovery after an incorrect intermediate decision rises from 46.1% to 61.3%. Weighting only disagreement, evidence consistency, or future reward underperforms full PCS by 5.2, 4.4, and 4.8 points, respectively, showing that future reward alone does not explain the gain. Retraining without collision-generated supervision absent from all pre-collision teacher responses lowers success from 49.6% to 43.9%, accounting for 72.6% of the observed improvement over standard OPD under this controlled removal-and-retraining intervention. The results show that verified, recoverable disagreement supplies supervision that raw disagreement and best-of-teacher selection do not capture.

1 INTRODUCTION

Large language models increasingly act through long, stateful trajectories: they inspect repositories, call tools, retrieve evidence, execute programs, and revise intermediate decisions. On-policy distillation is a natural way to transfer these capabilities to a smaller language agent because the teacher supervises states actually visited by the student rather than an offline collection of idealized responses (Ross et al., 2011; Agarwal et al., 2024). The dominant training view remains imitation-centric: one teacher response becomes the target, multiple teacher distributions are averaged, and variation across teachers is suppressed as label noise. We test a more specific hypothesis: disagreement has training value when it exposes a decision-sensitive alternative that is both verifiable and recoverable from the student's current state.

Consider a code-repair agent that assumes a failing test originates in a parser, a tool-use agent that trusts a stale observation, or a mathematical solver that silently treats a sufficient condition as necessary. A conventional teacher may repair the final answer without exposing the assumption. An ensemble may average two incompatible continuations into an uninformative token distribution. In contrast, a useful correction signal would (i) state the hidden assumption, (ii) invert it to construct a plausible competing branch, and (iii) use grounded feedback to determine whether the collision changes the outcome. The disagreement is then neither mere diversity nor a majority vote. It is a targeted intervention on a recoverable error.

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.

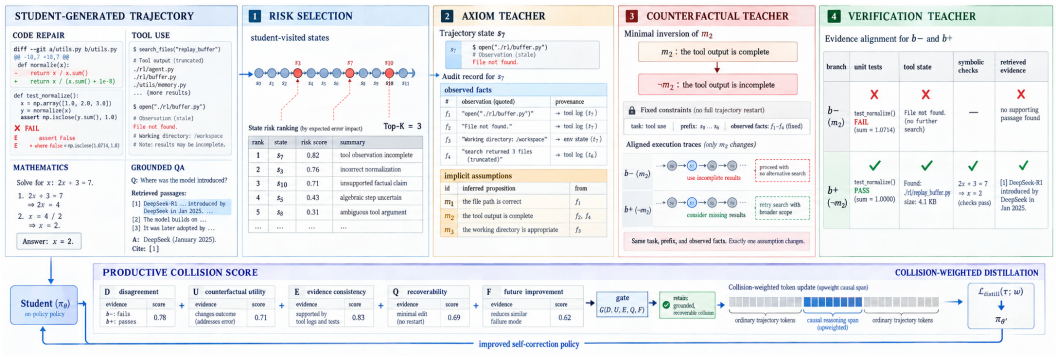


Figure 1: End-to-end C-OPD pipeline. Student-generated trajectories are ranked for risky decisions; the axiom teacher separates observations from implicit assumptions, the counterfactual teacher changes exactly one assumption while holding the prefix fixed, and task-native verification adjudicates the aligned branches. PCS retains grounded, recoverable collisions and upweights only the intervention-linked attribution span. The concrete record is schematic and illustrates the stored fields rather than an additional evaluation instance.

We call a teacher collision productive when competing branches disagree at a localized student decision, the alternative can be tested, and the verified branch improves downstream reward without restarting the episode. This definition separates benign wording variation, ungrounded conflict, and decision-relevant alternatives that yield an executable correction.

We introduce Collision-Aware On-Policy Distillation (C-OPD), illustrated in Figure 1. The student first rolls out a trajectory in the task environment. An *axiom teacher* extracts the assumptions supporting each high-risk step. A *counterfactual teacher* inverts selected assumptions and generates minimally changed alternative continuations. A *verification teacher* adjudicates the original and counterfactual branches using task-native feedback such as unit tests, tool return values, retrieved passages, or symbolic checks. These outputs are consolidated into a Productive Collision Score (PCS). PCS rewards disagreement only when it is useful, evidence-consistent, recoverable, and predictive of future reward. The score is attributed back to the responsible steps and tokens, producing a collision-weighted on-policy distillation objective.

The bottom path in Figure 1 is central to the method: verification does not merely select a better final response. It identifies the earliest discriminating evidence and maps that evidence back to the decision tokens that converted an observation into an unsupported assumption. Ordinary trajectory tokens therefore keep unit weight, while the localized recovery span receives collision-weighted supervision.

Our contributions are threefold. (i) We formulate productive disagreement as a distinct supervision source: a teacher conflict is useful when it reveals a verifiable, recoverable alternative at a student-visited decision. (ii) We introduce C-OPD, which constructs minimal assumption-level counterfactuals, verifies competing branches with task-native feedback, and attributes the resulting correction to intervention-linked decision spans. (iii) We evaluate the hypothesis across code repair, tool use, mathematics, and grounded question answering. Controlled comparisons separate full PCS from raw disagreement, evidence-only, future-reward-only, and extra teacher-role effects under matched supervision budgets.

2 Related Work

Knowledge and on-policy distillation. Knowledge distillation commonly trains a compact student to match the logits, sequences, or intermediate representations of a larger teacher (Hinton et al., 2015; Kim & Rush, 2016; Gu et al., 2024). Sequence-level methods provide coherent demonstrations but supervise states chosen by the teacher. Online and on-policy variants instead query teachers on student samples, reducing exposure mismatch and focusing training on the student's current failure distribution (Ross et al., 2011; Agarwal et al., 2024). Recent work reweights turns or tokens by disagreement, mines minority trajectories, distills multi-agent interaction, or selects counterfactual-

sensitive states (Luo et al., 2026; Tan et al., 2026; Li et al., 2026b; Jiang et al., 2026; Guo et al., 2026; Li et al., 2026a). DemoPSD adapts teacher adoption from teacher–student distribution discrepancy; counterfactual recoverability compares continuation and rollback from error states; and CROP measures response-token sensitivity to prompt-level counterfactuals. C-OPD instead intervenes on a named assumption inside a fixed student prefix, executes the competing branches, and uses task-native evidence to supervise the localized recovery decision. Appendix E separates mechanism boundaries from controlled local reproductions rather than ranking scores reported by different papers.

Multi-teacher learning and disagreement. Multi-teacher distillation combines complementary expertise through averaging, selection, routing, or confidence weighting (You et al., 2017; Wu et al., 2019). Ensemble uncertainty and query-by-committee use disagreement to select informative examples (Seung et al., 1992; Lakshminarayanan et al., 2017), while debate and critique elicit competing arguments from models (Du et al., 2024; Irving et al., 2018). These lines establish that disagreement can be informative, but raw disagreement also reflects style, calibration, correlated hallucination, and arbitrary sampling. Our goal is not to maximize diversity. We intervene on explicit assumptions, minimally vary the decision-driving premise, and require task-native verification before assigning extra training weight.

Self-correction and process supervision. Language agents can improve outputs through critique, reflection, external feedback, or iterative refinement (Madaan et al., 2023; Shinn et al., 2023; Lightman et al., 2024). Cross-model correction further transfers teacher diagnoses and correction traces to a smaller reasoner (Yang et al., 2025). However, self-correction without reliable feedback can preserve an incorrect premise or even damage a correct answer (Huang et al., 2024). Process supervision localizes rewards to intermediate reasoning steps (Uesato et al., 2022; Lightman et al., 2024), and preference optimization learns from paired trajectories (Rafailov et al., 2023). C-OPD complements these approaches by constructing the pair itself: the rejected and preferred continuations differ at a named assumption, and verification indicates when the preference is meaningful. Unlike inference-time reflection, the verified correction trace becomes training supervision and can be internalized by the student.

3 PRELIMINARIES

Language agents and trajectories. Let an environment provide an initial task x and states s_t containing the interaction history, tool outputs, retrieved evidence, or program execution results. A language-agent policy $\pi_\theta(a_t \mid s_t)$ emits a textual reasoning span, an environment action, or a final answer a_t. A trajectory is $\tau = (s_1, a_1, \ldots, s_T, a_T)$ with terminal reward $R(\tau) \in [0, 1]$. We use $G_t(\tau)$ for reward-to-go or a task-specific future-success score after step t.

On-policy distillation. Standard on-policy distillation samples $\tau \sim \pi_\theta$ and asks a teacher q for token targets on visited states. With $a_{t,j}$ denoting token j of action a_t, a generic objective is

$$\mathcal{L}_{\mathrm{OPD}}(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=1}^{T}\sum_{j=1}^{|a_t|}\mathrm{KL}(q(\cdot \mid s_t, a_{t,<j}) \,\|\, \pi_\theta(\cdot \mid s_t, a_{t,<j}))\right]. \tag{1}$$

For multiple teachers q_k, the common target is a mixture $\bar{q} = \sum_k \rho_k q_k$. This operation intentionally removes disagreement. It cannot distinguish a stylistic difference from a teacher branch that exposes a recoverable student error.

Assumption-conditioned branches. We represent a decision step t by an extracted assumption set $A_t = \{(z_{t,m}, r_{t,m})\}_{m=1}^{M_t}$, where $z_{t,m}$ is a concise proposition and $r_{t,m} \in [0, 1]$ quantifies its risk. A counterfactual operator ι produces a minimally inverted assumption $\tilde{z}_{t,m} = \iota(z_{t,m})$. Holding the prefix $\tau_{<t}$ fixed, teachers generate an original branch $b^+_{t,m}$ conditioned on $z_{t,m}$ and an alternative branch $b^-_{t,m}$ conditioned on $\tilde{z}_{t,m}$. Verification returns evidence $e_{t,m}$ and branch utilities $V(b^+), V(b^-)$. The branch names do not presume which one is correct; the verified higher-utility branch becomes the distillation target.

4 COLLISION-AWARE ON-POLICY DISTILLATION

4.1 GENERATING STRUCTURED TEACHER COLLISIONS

Step 1: collect student states. At each training iteration, the current student solves sampled tasks without access to teacher reasoning. We retain the trajectory, environment transitions, terminal outcome, and a compact failure summary. Sampling on policy has two effects: it exposes errors the student actually makes, and it naturally shifts the training curriculum as those errors are repaired.

Step 2: extract implicit axioms. The axiom teacher receives a trajectory prefix and one candidate decision step. It outputs propositions that must hold for the step to be valid, separates observed facts from inferred assumptions, and assigns a risk score

$$r_{t,m} = \sigma(\alpha_1 u_{t,m} + \alpha_2 d_{t,m} + \alpha_3 c_{t,m} - \alpha_4 o_{t,m}), \tag{2}$$

where u is uncertainty, d is downstream dependency, c is contradiction with context, and o is direct observational support. All four inputs are normalized to $[0, 1]$. We use the fixed coefficients $(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = (1, 1, 1, 1)$ and keep the top $K = 3$ nonredundant assumptions per selected step. For code repair, an axiom may be "the failing behavior originates in the parser rather than the caller." For tool use, it may be "the observation is current and corresponds to the requested entity." The teacher must quote the state evidence that licenses each assumption; unsupported reconstructions are marked low confidence.

Step 3: construct minimal counterfactuals. The counterfactual teacher inverts one high-risk axiom at a time and rolls out a short branch from the same prefix. The intervention is constrained to be minimal: entity names, goals, and already observed facts remain fixed, while only the selected assumption and intervention-linked downstream decisions may change. This prevents superficial paraphrases from inflating disagreement. When literal negation is ill-posed, the teacher proposes the closest mutually exclusive alternative, such as a different failure locus, API precondition, theorem case, or evidence interpretation.

Step 4: verify both branches. The verification teacher selects the strongest available oracle for the task. Code branches are checked by tests and static evidence; tool plans by returned observations and action preconditions; mathematical branches by symbolic or numerical substitution; grounded QA branches by citation entailment and retrieval consistency. Verification produces a vector

$$v^{\pm}_{t,m} = \left(v^{\pm}_{\text{exec}}, v^{\pm}_{\text{evid}}, v^{\pm}_{\text{goal}}, v^{\pm}_{\text{safe}}\right) \tag{3}$$

and a calibrated scalar $V(b^{\pm}_{t,m}) = \omega^{\top} v^{\pm}_{t,m}$. If neither branch has adequate support, the collision is discarded. If both are equivalent, it is labeled benign variation and receives no enhancement.

Figure 2 isolates the distinction from additional context or a longer search horizon. Both sides begin from the same prefix; the difference is whether teacher variation is collapsed into an ambiguous target or converted into a testable intervention. In the latter case, the stale-observation branch fails against the state-version evidence, whereas refreshing the observation yields a verified continuation and a localized correction target.

4.2 PRODUCTIVE COLLISION SCORE

Raw teacher divergence is necessary but insufficient. We score every verified collision $c = (t, m)$ with five components. *Disagreement* D_c measures semantic and action-level separation between branches after removing paraphrase variation. *Counterfactual utility* U_c measures how much the intervention changes verified branch value. *Evidence consistency* E_c measures whether the preferred branch is supported by task-native feedback. *Recoverability* Q_c measures whether changing the decision at t can still improve the trajectory without restarting the entire episode. *Future improvement* F_c is the verified change in reward-to-go. Because verifier scales differ by task, each component is converted to its percentile within the same task and verifier type. Writing $x_c = (D_c, U_c, E_c, Q_c, F_c)$

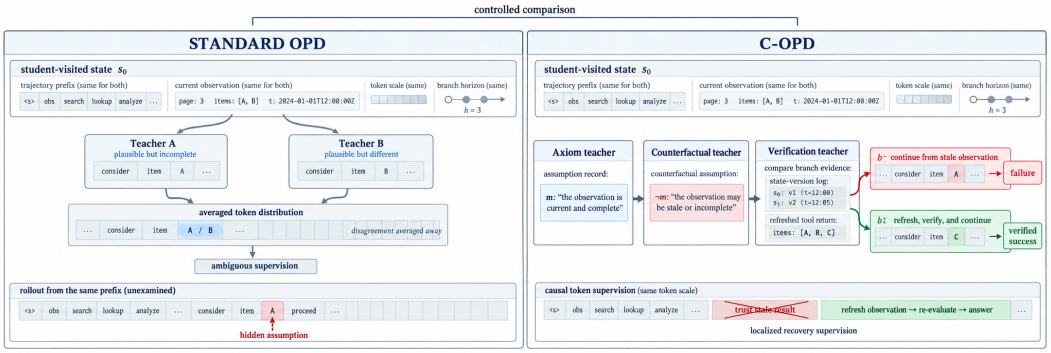


Figure 2: Controlled comparison of standard OPD and C-OPD from the same student-visited state, token scale, and branch horizon. Standard OPD averages incompatible continuations and leaves the hidden assumption unexamined. C-OPD exposes that assumption, constructs a minimal competing branch, uses fresh environment evidence to choose between the branches, and applies supervision to the exact recovery span.

and $\widehat{H}_{\mathcal{T}(c),k}$ for the calibration-set empirical CDF of component k in task $\mathcal{T}(c)$, we use

$$D_c = \tfrac{1}{2}\big(D_{\mathrm{JS}}(q_c^+ \| q_c^-) + D_{\mathrm{act}}(b_c^+, b_c^-)\big), \tag{4}$$

$$U_c = \big|V(b_c^+) - V(b_c^-)\big|, \qquad F_c = [G_t(\tau \oplus b_c^*) - G_t(\tau)]_+, \tag{5}$$

$$\widehat{x}_{c,k} = \widehat{H}_{\mathcal{T}(c),k}(x_{c,k}), \qquad \mathrm{PCS}(c) = \mathbf{1}[E_c \geq \epsilon] \sum_{k \in \{D,U,E,Q,F\}} \beta_k \widehat{x}_{c,k}. \tag{6}$$

Here $b_c^* = \arg\max_{b \in \{b_c^+, b_c^-\}} V(b)$. The evidence gate prevents confidently worded but ungrounded disagreement from receiving weight. F_c is computed only from branch execution on distillation-training trajectories; calibration data set the percentile maps and threshold. PCS, teachers, and verifiers are not invoked during final evaluation, where the frozen student acts alone. We select β and the retention threshold using task-disjoint calibration splits; Section 5.1 reports the fixed values used in every test run.

Attribution to steps and tokens. A collision may affect several later actions even though it originates at one assumption. We attribute its score to the minimal intervention-linked span S_c: the tokens that express the assumption, select the affected action, and carry it forward until contradictory feedback becomes available. This is a structured attribution heuristic anchored by the controlled inversion and earliest discriminating evidence; it is not a claim of causal identification from observational data. For token j at step t,

$$p_{t,j} = \max_c \ \mathrm{PCS}(c)\, \kappa\big((t,j), S_c\big), \qquad w_{t,j} = \mathrm{clip}(1 + \lambda p_{t,j}, 1, w_{\max}), \tag{7}$$

where κ is 1 inside the intervention-linked span and decays with distance outside it. The maximum avoids multiplying weights when several teachers restate the same collision. Tokens unrelated to a verified collision retain the ordinary on-policy weight of one.

4.3 Collision-weighted distillation

Let q_c^* denote the token distribution of the verified preferred branch and let y_c^* be its action sequence. Our main objective combines collision-weighted token distillation, a branch preference term, and a

Algorithm 1 One C-OPD training iteration

```
Sample tasks x ~ D and trajectories τ ~ π_θ(· | x)
Rank decision steps by failure relevance and uncertainty
for each selected step t do
    Extract assumptions A_t and retain the top-K risky, nonredundant items
    for each z_{t,m} ∈ A_t do
        Generate a minimal inverted assumption z̃_{t,m} and paired branches
        Verify both branches with execution, retrieval, or environment feedback
        Compute PCS(t, m); discard ungrounded or equivalent branches
        Attribute retained collisions to intervention-linked token spans S_{t,m}
    end for
end for
Update θ with Equation 9 under a matched teacher-token budget
```

conservative regularizer:

$$\mathcal{L}_{\text{C-OPD}}(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t,j} w_{t,j}\,\mathrm{KL}(q^*(\cdot \mid h_{t,j})\|\pi_\theta(\cdot \mid h_{t,j}))\right] \tag{8}$$

$$+\,\eta\,\mathbb{E}_c\left[-\,\mathrm{PCS}(c)\log\sigma\left(\gamma\log\frac{\pi_\theta(y_c^* \mid s_c)}{\pi_{\text{ref}}(y_c^* \mid s_c)} - \gamma\log\frac{\pi_\theta(y_c^- \mid s_c)}{\pi_{\text{ref}}(y_c^- \mid s_c)}\right)\right] + \mu\mathcal{L}_{\text{KL}-\text{ref}}. \tag{9}$$

The first term teaches the verified correction on student-visited prefixes. The second preserves the decision boundary exposed by the collision instead of collapsing the pair into one averaged target, following the form of direct preference optimization (Rafailov et al., 2023). The final term limits policy drift on states without verified evidence. We stop gradients through PCS and all verifier outputs.

Teacher usage is budgeted rather than unconditional. The lightweight risk model is the fixed logistic scorer in Equation 2; it has no learned parameters beyond the reported coefficients and ranks candidate assumptions using the four normalized teacher outputs. The counterfactual and verification teachers operate only on the top three assumptions per selected step. Thus comparisons can match the total number of teacher-generated tokens, an important control because unconstrained critique alone can improve performance by increasing inference compute.

5 Experiments

5.1 Experimental setup

Data separation. All distillation trajectories and student updates are generated from the task-specific training pools in Table 1, which are disjoint from both PCS calibration and final evaluation. PCS parameters and checkpoints are selected only on the calibration splits. The final-evaluation collections are used only for evaluation after the training configuration and checkpoint have been frozen.

Overlap removal. Before split assignment, we deduplicate code by repository and issue identifier, tool use by normalized API signature and scenario template, mathematics by canonicalized problem statement, and QA by normalized question and supporting-document identifiers; a key may occur in only one partition. No final-evaluation instance, trajectory, environment feedback, or teacher-generated supervision derived from an evaluation instance is used for training, calibration, checkpoint selection, or hyperparameter tuning.

Tasks. We use four benchmarks only for final evaluation, with distinct verification channels. *Code* is the 500-instance SWE-bench Verified test set with harness v1.2.0 (Jimenez et al., 2024); success requires all official fail-to-pass and pass-to-pass tests in a valid container. *Tool use* is the 1,200-episode BFCL v3.0.1 multi-turn test set, including 400 adversarial-observation episodes; success requires the

Table 1: Training, PCS calibration, and final-evaluation pools. All three partitions are mutually disjoint within each domain.

Domain	Distillation training pool	PCS calibration	Final evaluation
Code	SWE-bench train subset, 1,994 issues	300 held-out SWE-bench training issues	SWE-bench Verified, 500
Tool	BFCL v3.0.1 multi-turn and adversarial training pool, 6,000 episodes	600 held-out BFCL episodes	BFCL v3.0.1 fixed test, 1,200
Math	MATH train subset, 7,000 problems	500 held-out MATH training problems	MATH-500, 500
QA	HotpotQA v1.1 distractor fixed train subset, 20,000 examples	fixed validation subset, 1,000 examples	disjoint fixed evaluation subset from HotpotQA v1.1 validation, 1,000 examples

correct function, arguments, call order, and final state. *Mathematics* is MATH-500 (Hendrycks et al., 2021), scored by normalized exact match or SymPy 1.13 equivalence. *Grounded QA* uses 1,000 fixed HotpotQA v1.1 distractor examples (Yang et al., 2018); a success requires exact answer match, supporting-fact F1 of at least 0.80, and support for every atomic claim.

Baselines. Response KD distills complete offline responses, while standard OPD queries the same teacher on student-visited states using Equation 1. Controlled variants replace PCS with uniform disagreement weights, individual components, or $D + E$; Appendix Table 15 reports DemoPSD and CROP under the same reproduction protocol. Each method and seed trains one unified student across all four domains. We sample domains with equal probability (1:1:1:1) and sample tasks uniformly within a domain. The combined training pools contain 34,994 task instances. Each sampled task produces four rollouts, a $1{:}4$ task-to-rollout ratio; all variants run 12,000 updates and record 48,000 rollout environment calls per seed under an 800M teacher-token budget. We report seeds 42, 123, and 2027 and use their sample standard deviation.

Teachers and training. The student is Qwen2.5-7B-Instruct, fully fine-tuned in bfloat16 with DeepSpeed ZeRO-3 on eight A100-80GB GPUs. Response KD and all teacher roles use Qwen2.5-72B-Instruct on four H100-80GB GPUs. Axiom extraction, counterfactual construction, and structured verification share this model but use separate prompts; task-native unit tests, state-version checks, SymPy, and retrieved-evidence checks supply the adjudicating evidence. We train with AdamW, learning rate 10^{-5}, global batch size 64, cosine decay, 500 warmup steps, weight decay 0.1, gradient clipping 1.0, and a 16,384-token context. Generation is capped at 2,048 tokens and counterfactual branches at 1,024 tokens. Student rollouts and counterfactual generation use temperature 0.7 and top-$p = 0.95$; response and axiom teachers use temperature 0.2 and top-$p = 0.95$; structured verification is greedy with temperature 0 and top-$p = 1$. We retain the top $K = 3$ assumptions per selected step, set $(\eta, \gamma, \mu) = (0.5, 0.1, 0.02)$ in Equation 9, and mix collision records with ordinary successful on-policy traces at a 1:1 ratio. We set $(\beta_D, \beta_U, \beta_E, \beta_Q, \beta_F) = (.15, .15, .25, .20, .25)$, evidence gate $\epsilon = .70$, retention threshold .62, $\lambda = 2$, and $w_{\max} = 4$ from the calibration splits, then freeze them for evaluation.

5.2 Main results

Figures 3 and 4 report all four domains rather than only their macro-average and expose the behavioral structure of the result; Appendix Table 5 gives the exact values and uncertainty. Standard OPD improves on response KD in every domain. C-OPD adds 6.5 points on code, 9.8 on tool use, 8.1 on mathematics, and 6.8 on grounded QA over standard OPD, producing a 7.8-point macro-average gain.

Figure 3(a) shows that the gain is consistent across qualitatively different feedback channels, with the largest absolute increase on tool use ($+9.8$ points) and a $+7.8$-point macro-average increase. Figure 4 makes the baseline-relative contribution explicit: the collision-aware increment remains positive for code execution, environment interaction, symbolic verification, and retrieved-evidence grounding. Figure 3(b) further shows that higher terminal success is accompanied by fewer repeated or unsupported actions and by substantially greater recovery after an identified intermediate error, rather than only by better first-pass answers.

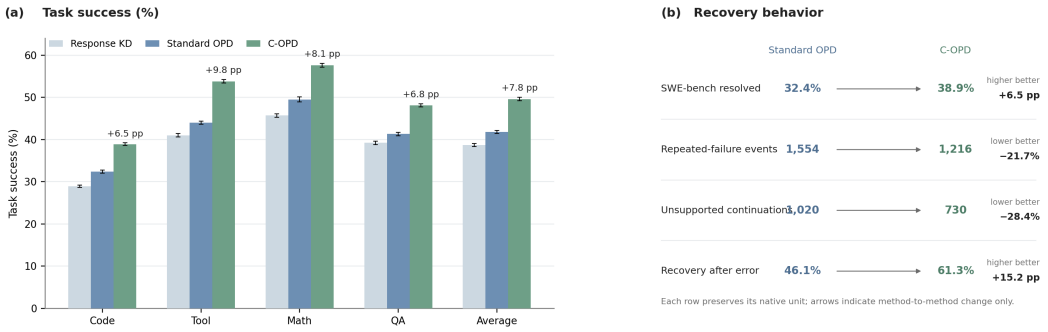


Figure 3: Main results under the matched 800M teacher-token budget per seed. (a) Task success for response-level KD, standard OPD, and C-OPD; error bars show sample standard deviation over three seeds, and annotations give the absolute gain of C-OPD over standard OPD. (b) Recovery behavior in each metric's native unit. Arrows denote the method-to-method change only; lower values are favorable for repeated failures and unsupported continuations.

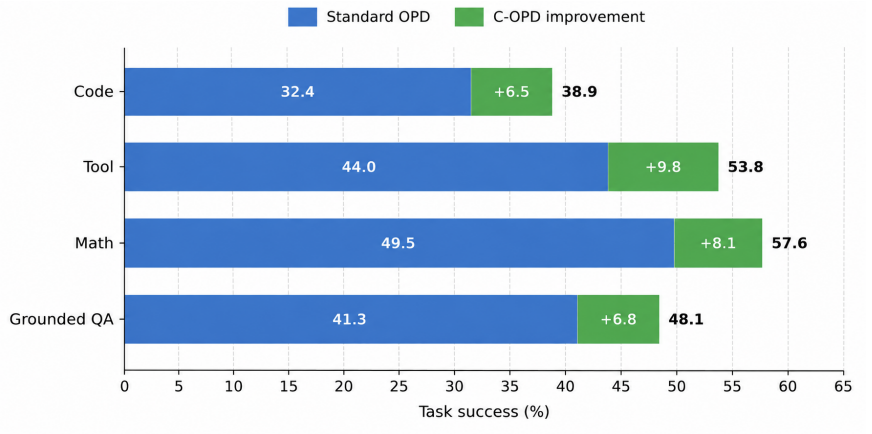


Figure 4: Domain-wise improvement over standard OPD. Each blue segment is the standard-OPD success rate, each green segment is the absolute gain from collision-aware supervision, and the endpoint is the final C-OPD score. Improvements occur in all four domains rather than being driven by a single benchmark.

The gain is positive in all 12 domain–seed comparisons (6.4–9.8 points; Appendix Table 4). Paired resampling gives a $+7.8$-point macro gain with 95% CI $[6.2, 9.5]$ and permutation $p = .004$ versus standard OPD; Appendix Table 6 reports domain intervals and the response-KD comparison.

On SWE-bench Verified, C-OPD resolves 583 of 1,500 seed-instance evaluations, compared with 486 for standard OPD. Repeated-failure events fall from 1,554 to 1,216. Axiom extraction frequently identifies commitment to the first suspected file or failure mechanism; counterfactual inversion preserves an alternative until a test discriminates between them. The student is trained to abandon a branch after contradictory execution evidence rather than merely copy a successful patch.

The BFCL stress split tests stale, incomplete, or adversarially distracting observations. Across 3,600 seed-episode evaluations, unsupported continuations fall from 1,020 to 730. Among 1,500 incorrect intermediate decisions identified automatically by task-native verification per method, successful recovery increases from 692 to 920. The most heavily weighted spans are those that convert an observation into an assumption and choose whether to continue, refresh, or seek corroboration.

Configuration	Average success (%)	Drop
No counterfactual inversion	43.8 ± 0.4	-5.8
Monolithic critique-and-repair	43.8 ± 0.4	-5.8
Uniform disagreement weight	45.7 ± 0.4	-3.9
D only	44.4 ± 0.4	-5.2
E only	45.2 ± 0.4	-4.4
F only	44.8 ± 0.4	-4.8
$D + E$	47.4 ± 0.4	-2.2
Full PCS	$\mathbf{49.6 \pm 0.4}$	—

Table 2: PCS, construction, and critique-and-repair comparisons under 800M teacher tokens per seed. Drops are percentage points relative to full PCS.

5.3 What makes a collision productive?

Table 2 separates PCS from its strongest individual signals. Uniformly weighting every disagreement reaches 45.7%, and disagreement alone reaches 44.4%. Evidence consistency and future improvement alone reach 45.2% and 44.8%, respectively. Combining disagreement and evidence closes part of the gap, but remains 2.2 points below full PCS. Thus the full result cannot be explained by access to future reward F alone.

Removing counterfactual inversion lowers success to $43.8 \pm 0.4\%$. A single prompt that asks one teacher to critique and repair obtains the same 43.8%, whereas the shared Qwen2.5-72B model with three role-specific prompts reaches $49.6 \pm 0.4\%$ under the same token budget. The gain therefore comes from structuring the interaction, not from hosting three independent teacher models.

Of 12,480 retained preferred branches, 1,960 exactly match a decisive action in an initial teacher response and 2,290 pass bidirectional entailment under DeBERTa-v3-large at a 0.80 threshold; the remaining 8,230 are collision-generated. Retraining after removing these novel records lowers average success from 49.6% to 43.9%. The 5.7-point decrease accounts for 72.6% of the observed 7.8-point gain over standard OPD under this controlled removal-and-retraining intervention.

5.4 Collision calibration and efficiency

Across 12,480 collisions, PCS correlates with correction success at pooled Spearman $\rho = .58$ (95% CI $[.56, .60]$) and with later reward gain at $\rho = .46$ (95% CI $[.44, .48]$); both have $p < 10^{-6}$. Appendix Figure 5(a) shows positive associations in every domain. From the bottom to the top PCS quintile, correction success rises by 41.3 points on code, 42.7 on tool use, 34.2 on mathematics, and 34.7 on grounded QA (Appendix Figure 5(b)). The pooled correction-success AUC is .76 and expected calibration error is .041.

Teacher-token matching is not total-compute matching: full C-OPD uses 18.4K additional verifier calls and 35.5% more GPU-hours than standard OPD. Under an equal 1,028.2 GPU-hour budget, compute-matched C-OPD still reaches $47.3 \pm 0.5\%$ versus $41.8 \pm 0.4\%$ (Appendix Table 10). A 500-collision audit by three annotators finds criterion precision of .89–.94 and mean pairwise Cohen's $\kappa = .79$ (Appendix Table 18).

6 Conclusion

We introduced C-OPD, which turns structured teacher disagreement into localized recovery supervision. Across four domains, it reaches $49.6 \pm 0.4\%$ macro-average success versus $41.8 \pm 0.4\%$ for standard OPD under the same 800M teacher-token budget. Counterfactual and PCS ablations show that verified alternatives at student-visited decisions provide signal beyond initial teacher responses.

Reproducibility statement. Section 5.1 specifies data separation, unified-student sampling, optimization, decoding, seeds, and budgets. The appendix provides per-seed records, costs, prompts, verification and deduplication rules, and fixed hyperparameters. No final-evaluation instance or derivative supervision is used for training or model selection.

AI USE STATEMENT

Generative AI tools were used to assist with literature organization, conceptual and methodological refinement, manuscript drafting and editing, figure design and layout, and LaTeX formatting. They were not used to generate or alter the reported experimental measurements. The authors ran the experiments, checked the reported values against the experiment records, reviewed the cited literature and all AI-assisted text and figures, and take full responsibility for the final content of the submission.

ETHICS STATEMENT

This study evaluates language models on research benchmarks and sandboxed execution environments; it does not deploy agents to real users or authorize irreversible external actions. The human audit concerns model-generated collision records and reports aggregate agreement statistics only. Because stronger recovery behavior can also increase agent capability and persistence, Appendix L discusses verifier failure modes, sandboxing, logging, and human-approval boundaries for sensitive deployments.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. GKD: Generalized knowledge distillation for auto-regressive sequence models. *ICLR*, 2024.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *ICML*, 2024.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *ICLR*, 2024.

Jiaxin Guo, Yanwei Yue, Xuanbo Fan, Chunyu Yang, and Yan Zhang. Learning from consensus and disagreement: Unsupervised on-policy self-distillation with minority-trajectory contrast. *arXiv preprint arXiv:2608.08764*, 2026.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NeurIPS Deep Learning and Representation Learning Workshop*, 2015.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. *ICLR*, 2024.

Geoffrey Irving, Paul Christiano, and Dario Amodei. AI safety via debate. *arXiv preprint arXiv:1805.00899*, 2018.

De Jiang, Zhengyang Zhang, Kehong Yuan, and Shaohua Ma. Not every divergence should be suppressed: Counterfactual recoverability in on-policy distillation. *arXiv preprint arXiv:2608.04408*, 2026.

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 *ICLR*, 2024.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *EMNLP*, 2016.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, 2017.

Enhan Li, Junhao He, and Hongyang Du. CROP: Task relevance via counterfactuals for selective on-policy distillation. *arXiv preprint arXiv:2608.13387*, 2026a.

Yunhe Li, Hao Shi, Wenhao Liu, Mengzhe Ruan, Hanxu Hou, Zhongxiang Dai, Shuang Qiu, and Linqi Song. DemoPSD: Disagreement-modulated policy self-distillation. *arXiv preprint arXiv:2607.02502*, 2026b.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *ICLR*, 2024.

Yinyi Luo, Yiqiao Jin, Weichen Yu, Mengqi Zhang, Srijan Kumar, Xiaoxiao Li, Weijie Xu, Xin Chen, and Jindong Wang. Agentark: Distilling multi-agent intelligence into a single llm agent. *arXiv preprint arXiv:2602.03955*, 2026.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. In *NeurIPS*, 2023.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *NeurIPS*, 2023.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, 2011.

H. Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *COLT*, 1992.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *NeurIPS*, 2023.

Qitai Tan, Zefang Zong, Mo Li, Yipeng Shi, Yang Li, and Peng Chen. ATOD: Annealed turn-aware on-policy distillation for multi-turn agentic tasks. *arXiv preprint arXiv:2606.27814*, 2026.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

Ancong Wu, Wei-Shi Zheng, Xiaowei Guo, and Jian-Huang Lai. Multi-teacher knowledge distillation for compressed video action recognition. In *ICCV*, 2019.

Ling Yang, Zhaochen Yu, Tianjun Zhang, Minkai Xu, Joseph E. Gonzalez, Bin Cui, and Shuicheng Yan. Supercorrect: Advancing small llm reasoning with thought template distillation and self-correction. In *International Conference on Learning Representations*, 2025.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

Shan You, Chang Xu, Chao Xu, and Dacheng Tao. Learning from multiple teacher networks. In *KDD*, 2017.

A Exact Results and Per-Seed Records

This section reports the run-level records underlying every aggregate in the main paper. We retain the original seed identifiers and avoid recomputing results from rounded means. Task success is evaluated only after the training configuration and the step-12,000 checkpoint have been frozen. The four task columns below correspond to SWE-bench Verified, BFCL v3.0.1, MATH-500, and the fixed 1,000-example HotpotQA v1.1 evaluation subset described in Section 5.1.

A.1 Task success for every seed

Method	Seed	Code	Tool	Math	QA	Macro avg.
Response KD	42	28.6	40.6	45.2	38.8	38.30
	123	29.2	41.3	46.0	39.6	39.03
	2027	28.8	41.1	45.9	39.2	38.75
Standard OPD	42	32.0	43.6	49.0	40.9	41.38
	123	32.8	44.4	50.1	41.7	42.25
	2027	32.4	44.0	49.4	41.3	41.78
C-OPD	42	38.6	53.4	57.1	47.7	49.20
	123	39.2	54.2	58.0	48.5	49.98
	2027	38.8	53.8	57.7	48.1	49.60

Table 3: Per-seed task success (%). The macro-average is the unweighted mean of the four domain scores within a seed.

Aggregating the three rows for each method gives 38.7 ± 0.4 for response KD, 41.8 ± 0.4 for standard OPD, and 49.6 ± 0.4 for C-OPD. The ordering is unchanged for all three seeds and all four domains. The exact matched-budget mean and sample standard deviation are reproduced in Table 5 for convenient comparison with the figures in the main paper.

Seed	Code gain	Tool gain	Math gain	QA gain	Macro gain
42	+6.6	+9.8	+8.1	+6.8	+7.825
123	+6.4	+9.8	+7.9	+6.8	+7.725
2027	+6.4	+9.8	+8.3	+6.8	+7.825

Table 4: Paired C-OPD minus standard-OPD gains in percentage points. All 12 domain–seed comparisons are positive.

Method	Code	Tool	Math	QA	Average
Response KD	28.9 ± 0.3	41.0 ± 0.4	45.7 ± 0.4	39.2 ± 0.4	38.7 ± 0.4
Standard OPD	32.4 ± 0.4	44.0 ± 0.4	49.5 ± 0.6	41.3 ± 0.4	41.8 ± 0.4
C-OPD	$\mathbf{38.9 \pm 0.3}$	$\mathbf{53.8 \pm 0.4}$	$\mathbf{57.6 \pm 0.5}$	$\mathbf{48.1 \pm 0.4}$	$\mathbf{49.6 \pm 0.4}$

Table 5: Exact matched teacher-token results. Values are mean ± sample standard deviation over seeds 42, 123, and 2027.

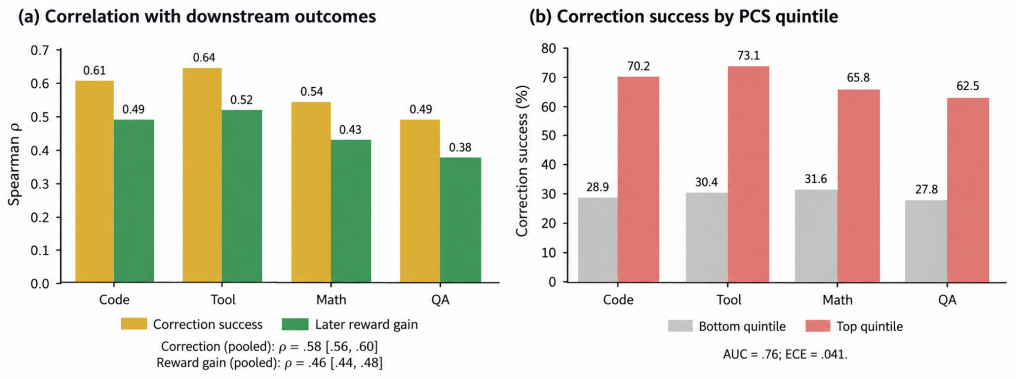


Figure 5: PCS validity across domains. (a) Spearman correlations with correction success and later reward gain, with pooled values and 95% confidence intervals below. (b) Correction success in the bottom and top PCS quintiles; pooled AUC is .76 and ECE is .041.

A.2 Paired resampling and permutation tests

We run paired resampling on the stored evaluation records used for the completed comparisons: 1,500 seed-instance code records, 3,600 seed-episode tool records, 500 paired mathematics problems, and 1,000 paired grounded-QA examples. The macro interval resamples within each domain and then averages the four domain deltas. These intervals quantify evaluation-unit uncertainty; the across-seed sample standard deviations remain reported separately.

Domain	Δ over standard OPD	95% paired bootstrap CI
Code	$+6.5$	$[4.1, 8.8]$
Tool	$+9.8$	$[7.3, 12.1]$
Math	$+8.1$	$[5.7, 10.6]$
Grounded QA	$+6.8$	$[4.0, 9.4]$
Macro average	$+7.8$	$[6.2, 9.5]$

Table 6: Paired bootstrap intervals for C-OPD minus standard OPD, in percentage points.

A paired permutation test gives $p = .004$ for C-OPD versus standard OPD and $p < .001$ for C-OPD versus response KD. The tests use the same frozen checkpoints and matched evaluation records as the reported success rates.

A.3 Behavioral outcomes for every seed

The repeated-failure count is measured on SWE-bench Verified. An event is recorded when consecutive edits or commands reproduce the same failure signature without acquiring new discriminating evidence. Unsupported continuation and recovery are measured on the BFCL adversarial-observation analysis. Each recovery denominator contains 500 automatically identified incorrect intermediate decisions per seed and method.

Method	Seed	Code resolved	Repeated failures	Unsupported cont.	Recovered / 500
Standard OPD	42	160	510	336	228
	123	164	526	348	235
	2027	162	518	336	229
C-OPD	42	193	399	242	304
	123	196	412	248	311
	2027	194	405	240	305

Table 7: Per-seed behavior records. Code resolved is a count out of 500; recovery is a count out of 500 automatically identified intermediate errors.

Pooling the three evaluations yields 486 versus 583 resolved code instances, 1,554 versus 1,216 repeated-failure events, and 1,020 versus 730 unsupported continuations for standard OPD and C-OPD, respectively. Recovery rises from 692 of 1,500 events to 920 of 1,500 events, corresponding to 46.1% and 61.3% after rounding.

A.4 Compute and environment accounting

The 800M teacher-token constraint applies independently to every method and seed. Standard OPD and response KD each use 48,000 rollout environment calls per seed. C-OPD uses those same rollout calls plus the verifier calls shown below. We keep this extra environment and hardware cost visible rather than treating teacher-token matching as total-compute matching.

Seed	Rollout calls	Verifier calls	Total environment calls
42	48,000	18,600	66,600
123	48,000	18,250	66,250
2027	48,000	18,420	66,420
Mean	48,000	18,423	66,423

Table 8: Environment-call accounting for C-OPD. Verifier calls are additional to the rollout calls shared with standard OPD.

Method	Tokens / seed	Calls / seed	Total GPU-h	Mean GPU-h	Mean wall-h
Response KD	800M	48.0K	2,530.2	843.4	75.47
Standard OPD	800M	48.0K	3,084.6	1,028.2	91.23
C-OPD	800M	66.4K	4,179.2	1,393.1	121.97

Table 9: Training and teacher-generation cost over three seeds. GPU-hours sum the student and teacher hardware described in Section 5.1.

Relative to standard OPD, full C-OPD uses 35.5% more GPU-hours and approximately 18.4K additional environment calls per seed. We therefore also run a compute-controlled variant that lowers verification frequency while retaining the 800M teacher-token allowance.

Method	Teacher tokens / seed	Mean GPU-h	Success (%)
Response KD	800M	843.4	38.7 ± 0.4
Standard OPD	800M	1,028.2	41.8 ± 0.4
C-OPD (full verification)	800M	1,393.1	49.6 ± 0.4
C-OPD (compute matched)	800M	1,028.2	$\mathbf{47.3 \pm 0.5}$

Table 10: Compute-controlled comparison from completed runs. The compute-matched C-OPD schedule reduces verification frequency until mean total GPU-hours equal standard OPD; it is distinct from the full-verification main configuration.

Matching total GPU-hours lowers C-OPD by 2.3 points relative to full verification, but it remains 5.5 points above standard OPD. Thus additional verification compute contributes to the full score without accounting for the entire observed advantage.

B Additional Prompting Details and Structured Contracts

Following the presentation style used in detailed self-correction appendices, we give the complete role instructions and the interface between roles. The prompts are written in model-agnostic form. Task adapters fill the bracketed fields and expose only information available at the selected student state.

B.1 Axiom-extraction teacher

> **Role.** You inspect one decision made by a language agent. Recover the propositions that must be true for that decision to be justified. Do not solve the task, repair the trajectory, or use information from later states.
>
> **Inputs.** You receive the task goal, the trajectory prefix ending at step t, the selected student decision, and the observations available before that decision.
>
> **Procedure.** First copy the observations that directly support the decision and identify their exact spans. Then list at most five inferred propositions on which the decision depends. For each proposition, state which later action depends on it, score uncertainty, downstream dependency, contradiction with context, and direct observational support in $[0, 1]$, and provide the closest mutually exclusive alternative. An alternative is invalid when it contradicts an observed fact, changes the task goal, or merely paraphrases the original proposition.
>
> **Output.** Return a structured record containing `step_id`, `observed_facts`, `assumptions`, `support_spans`, the four risk inputs, `dependent_actions`, and `minimal_alternative`. Return NO-ASSUMPTION when the decision follows directly from observation.

The no-repair restriction prevents the first teacher from fixing the answer and forcing later roles to rationalize that fix. It also lets the risk scorer distinguish a decision supported by the current state from a plausible decision supported only by the teacher's background knowledge.

B.2 Counterfactual-construction teacher

> **Role.** Construct a controlled alternative to one selected assumption. You generate a competing continuation, not a second independent solution.
>
> **Fixed content.** Preserve the task, trajectory prefix, observations, entities, constraints, and goal. Preserve decisions that are not downstream of the selected assumption.
>
> **Intervention.** Replace exactly one assumption with the supplied mutually exclusive alternative. Continue both the original and inverted branches only until they make different testable predictions or select different actions. List every changed decision. For each change, explain why it follows from the intervention. A branch may request new evidence, but it cannot cite the result of that request before the corresponding environment action occurs.
>
> **Termination.** Return INVALID if the inversion contradicts an observed fact, requires changing another premise, changes only wording, or cannot lead to a discriminating observation within the branch cap. Otherwise return VALID with aligned original and counterfactual branches.

The counterfactual is minimal at the assumption level. Surface similarity is not required after the selected decision, because a single changed premise can legitimately alter several downstream actions. The audit checks that every downstream change is connected to the named intervention.

B.3 Verification teacher

> **Role.** Adjudicate two branches from the same student state using the strongest task-native evidence available. Verbal confidence is not evidence.
>
> **Checks.** For each branch, report execution validity, evidence consistency, goal progress, and constraint satisfaction. Quote the raw test output, tool observation, symbolic check, or retrieved passage that supports each nonzero score. Identify the earliest observation at which

the branches make distinguishable predictions. Reject evidence that was unavailable at the selected decision step unless it is obtained by an explicit action in the branch.

Decision. Return EQUIVALENT if normalized actions and verified outcomes agree. Return INCONCLUSIVE if neither branch has evidence above the fixed gate. Otherwise return the preferred branch, the rejected branch, the earliest discriminating step, and the minimal intervention-linked span to which supervision should be attributed.

The verification teacher organizes evidence, while task-native systems provide the decisive signal. Unit tests, executable tool states, symbolic checks, and cited passages take precedence over the verifier model's verbal judgment.

B.4 MACHINE-READABLE CONTRACT

Stage	Required fields	Terminal status
Axiom extraction	step id; observed facts and exact spans; assumption text; u, d, c, o; dependent actions; exclusive alternative	VALID, NO-ASSUMPTION
Counterfactual construction	assumption id; fixed fields; aligned branches; changed decisions; intervention links; branch horizon	VALID, INVALID
Verification	raw evidence; execution, evidence, goal, and constraint scores; earliest discriminating step; preferred branch	PREFERRED, EQUIVALENT, INCONCLUSIVE
PCS and attribution	D, U, E, Q, F; task-conditional percentiles; PCS; intervention-linked span; inclusion reason; resource counts	RETAIN, REJECT

Table 11: Structured contract between the three roles and the collision scorer. Free-form prose that is not linked to a required field is ignored.

Every retained record also stores the task identifier, split, model and prompt versions, student-trajectory hash, environment version, teacher-token count, environment-call count, and final inclusion decision. This record is sufficient to reconstruct why a collision entered training and which tokens received additional weight.

C A FULLY WORKED COLLISION RECORD

The record in this section is **schematic**. It illustrates the exact information flow and is not presented as an additional BFCL evaluation episode. The example uses a stale tool observation because the original and alternative branches can be aligned without disclosing benchmark content.

C.1 STUDENT STATE AND SELECTED DECISION

Task state. The agent must update an existing resource after confirming its current identifier. At step $t-1$, a lookup returns an empty list tagged with state version v_1. At step t, the student concludes that no matching resource exists and chooses a creation action.

Available evidence. The prefix contains the queried entity string, the empty result, the state version, and a previous observation indicating that another action may have changed the environment after v_1. It does not contain a refreshed lookup.

Risk selection. The lightweight scorer assigns high risk to the sentence connecting the empty result to global nonexistence. The decision has weak direct support because the observation proves only that one query at version v_1 returned no rows. The decision also has high downstream dependency because creation, argument selection, and final-state verification depend on it.

C.2 AXIOM EXTRACTION AND MINIMAL INVERSION

The axiom teacher separates the recorded observation from the inferred proposition.

Observed fact: the lookup with the recorded arguments returned an empty list at state version v_1.

> **Selected assumption:** this lookup is current, complete, and scoped to the intended entity; therefore the resource does not exist.
>
> **Minimal alternative:** the resource may exist, but the observation is stale or resolved under an incorrect scope.

The alternative does not assert that the resource exists. It changes only the student's claim that the old observation is sufficient. This distinction matters because a valid counterfactual need not guess the correct final state; it must create a testable alternative that can change the next decision.

C.3 Aligned branches

Original-assumption branch. The branch proceeds directly to resource creation using the inferred nonexistence claim. Its prediction is that creation is permitted and produces a unique new resource.

Counterfactual branch. The branch refreshes the state, resolves the entity under the requested scope, and repeats the lookup. Its prediction is that the fresh observation will either reveal the resource or establish nonexistence under the current version. Only after this observation does it choose update or creation.

Both branches preserve the original goal, entity, constraints, and action history. The only new operation in the counterfactual branch is the observation needed to distinguish current absence from stale or mismatched retrieval.

C.4 Verification and intervention-linked attribution

Suppose the refreshed lookup returns the target resource. The state-version check and returned identifier then supply the earliest discriminating evidence. The creation branch is rejected because its precondition is false in the executable state; the refresh-and-update branch is preferred because its action is supported by the current observation and reaches a valid goal state.

The intervention-linked span begins at the student's conversion of "empty at v_1" into "absent now" and ends at the unsupported creation decision. Earlier tokens that merely state the task and later tokens that format the final answer retain unit weight. The collision target teaches the student to request discriminating evidence before committing to the state-dependent action.

C.5 Stored record

Field	Schematic value
Selected step	First action that treats the v_1 lookup as current and complete
Observed fact	Empty lookup result with the recorded arguments and state version v_1
Assumption	The observation establishes present global nonexistence
Minimal inversion	The observation may be stale or scoped to the wrong entity
Original prediction	Creation is valid and yields a unique resource
Alternative prediction	Refreshing and resolving the entity changes the next valid action
Discriminating evidence	Fresh state version and returned resource identifier
Preferred action	Update the verified resource
Intervention-linked span	Inference from empty result through the unsupported creation decision
Disposition	Retain after evidence, recoverability, and intervention-validity checks

Table 12: Field-level view of the schematic worked collision. No benchmark score is derived from this example.

D Detailed Qualitative Analysis

This section mirrors the error-analysis role of long self-correction appendices. Each example is a protocol-level construction that exposes what supervision differs across domains. The examples are deliberately labeled schematic because the experiment workbook contains aggregate and per-seed measurements but not publishable raw trajectories.

D.1 Repository-level code repair

Student prefix. The agent reads a failure involving malformed input, inspects the parser, and edits its first conditional. The targeted test still fails with the same stack location. The agent proposes a second parser edit without examining the caller.

Extracted assumption. The parser must be the failure locus because it appears in the stack trace. The observation supports that the parser is on the failing path; it does not establish that the parser violated its contract.

Minimal inversion. The parser is contract-consistent, while the caller supplies a value that violates a precondition. The counterfactual preserves the failing input and repository state, adds a call-site inspection, and predicts a different edit location.

Verification. The two branches run the same regression test. The parser-edit branch reproduces the failure or breaks a pass-to-pass test. The caller branch changes the input construction, passes the targeted test, and is then checked against the broader suite. The first discriminating signal is the call-site value paired with the parser contract, not the final all-tests-passed message.

Supervision. Standard OPD can copy whichever patch the teacher emits. A monolithic critique can state that the first edit was wrong. C-OPD preserves the explicit competition between failure loci and weights the transition from stack location to contract inspection. The recovery target is to reopen the diagnosis after contradictory execution evidence instead of applying another edit to the first suspected file.

D.2 Tool-use planning

Student prefix. The agent uses a search endpoint, receives one page of results, and treats the target's absence from that page as global absence. It then selects a fallback tool whose precondition assumes no matching object exists.

Extracted assumption. The response is complete, correctly paginated, and resolved to the intended entity. The directly observed fact is narrower: one page under the recorded arguments lacks the target.

Minimal inversion. The target may appear on a later page or under a corrected entity identifier. The alternative branch holds the task and all previous observations fixed, then checks the pagination token and entity resolution before selecting the fallback.

Verification. The task-native state checker evaluates the normalized tool name, arguments, order, and final state. If the next page returns the target, the counterfactual branch obtains direct support. If no pagination or alternate scope exists, the branches become equivalent and the collision receives no enhancement.

Supervision. The preferred trace trains a conditional behavior: seek the missing discriminating observation when the current response is incomplete. It does not train a generic rule to distrust tools. This boundary limits habitual overchecking on states that are already complete.

D.3 Mathematical reasoning

Student prefix. The solver derives a condition that guarantees the target property and immediately declares it to characterize all solutions. The algebra within the derived case is correct, but no argument establishes necessity.

Extracted assumption. The sufficient condition exhausts the solution set. The risk arises from quantifier scope and downstream dependency, not from a local arithmetic error.

Minimal inversion. An omitted case can satisfy the original statement without satisfying the derived condition. The counterfactual branch keeps the original problem and completed algebra, then searches for a counterexample or restores the missing case split.

Verification. Symbolic substitution and domain checks take precedence over a second verbal proof. A valid counterexample rejects the necessity claim. If exhaustive case analysis shows no omitted solution, the original branch is preserved and the apparent disagreement is discarded.

Supervision. The weighted span covers the move from a one-way implication to an equivalence claim and the case split needed to repair it. Correct algebra elsewhere remains ordinary on-policy supervision. This localization avoids treating the entire solution as negative because one logical transition failed.

D.4 Evidence-grounded question answering

Student prefix. The agent retrieves a passage containing the requested name and writes an answer whose time scope and relation type are inherited from that passage. A second retrieved document refers to an entity with the same surface name under another date.

Extracted assumption. The first passage resolves the intended entity and temporal scope, and its supporting sentence entails every atomic claim in the answer.

Minimal inversion. The surface match may refer to an alias, a different entity, or a different period. The alternative branch preserves the question and retrieved text, decomposes the answer into claims, and seeks a second source that disambiguates identity and time.

Verification. The verifier records document identifiers and evidence spans. A branch is preferred only when each decisive claim is entailed by its cited passage and the entity resolution agrees across evidence. A fluent answer with an unsupported relation is rejected even when its exact answer string happens to match.

Supervision. The recovery target is the decision to commit after the first lexical match. The corrected branch teaches corroboration or abstention when the retrieved evidence does not identify the intended entity. It does not reward longer rationales by themselves.

E Standard OPD, Monolithic Critique, and C-OPD

Table 13 compares the training signal produced from the same schematic error state. This complements the quantitative critique-and-repair baseline by making the unit of supervision explicit.

Property	Standard OPD	Monolithic critique-and-repair	C-OPD
Starting state	Student-visited state	Student-visited state plus a repair instruction	Student-visited state with one selected high-risk decision
Teacher output	One continuation or token distribution	Diagnosis and repaired continuation from one prompt	Named assumption, minimal inversion, aligned branches, and verifier record
Treatment of disagreement	Averaged or selected implicitly	May appear inside free-form critique	Preserved until task-native evidence discriminates
Attribution localization	All target tokens follow the same OPD rule	Repair trace may mix diagnosis and final answer	Extra weight only on the assumption, affected decision, and recovery span
Grounding requirement	Depends on ordinary teacher target	Critique can remain verbal	Evidence gate and task-native execution are required
Invalid alternative	No explicit alternative record	Often omitted from training output	Stored with a reason: non-minimal, equivalent, inconclusive, or unsupported
Student lesson	Imitate a teacher continuation	Imitate a corrected response	Distinguish two hypotheses and act on the evidence that separates them

Table 13: Qualitative comparison of the supervision produced from an identical student-visited state.

The monolithic critique-and-repair configuration reaches $43.8 \pm 0.4\%$ average success. The role-separated configuration reaches $49.6 \pm 0.4\%$ with the same Qwen2.5-72B teacher and teacher-token budget. The comparison isolates interaction structure from additional hosted teacher capacity.

E.1 Relationship to concurrent selective OPD methods

Table 14 states the mechanism boundary relative to the closest 2026 methods. It is a design comparison drawn from the cited method definitions, not a ranking of numbers copied from papers with different data, students, teachers, feedback channels, and budgets. We separately reproduce DemoPSD and CROP under our unified protocol in Table 15.

Method	Constructed contrast	Selection or adjudication signal	Learned target
DemoPSD	Teacher versus student token distributions	Distribution discrepancy controls a reverse-KL barycenter	Adaptive token-level adoption of privileged teacher guidance
Counterfactual recoverability	Teacher continuation versus rollback from an error state	Relative branch success categorizes recoverable, irreversible, or ambiguous states	Retain, roll back, or conventionally supervise the state
CROP	Original, paraphrased, and counterfactual prompts with a fixed rollout	Paraphrase-calibrated counterfactual sensitivity	Select response positions by task relevance
C-OPD	Original versus minimally inverted named assumption from one fixed student prefix	Task-native execution, state, symbolic, or retrieved-evidence feedback	Verified preferred branch plus an intervention-linked attribution span

Table 14: Mechanism-level boundary relative to the closest concurrent selective-OPD methods. No row implies empirical superiority across their incompatible evaluation settings.

Method	Macro success (%)
Standard OPD	41.8
DemoPSD	43.9
CROP	45.1
C-OPD	**49.6**

Table 15: Controlled local reproductions under the unified student, training, budget, and evaluation protocol in Section 5.1. These are our reproduced results, not a ranking of the original papers' reported benchmark numbers.

Under this common protocol, C-OPD exceeds the reproduced DemoPSD and CROP configurations by 5.7 and 4.5 macro points, respectively. The comparison tests training strategies within this study and does not imply that the original papers used equivalent experimental settings.

F PCS Component Construction and Calibration

PCS is computed only after a collision has passed intervention-validity checks. The score combines five quantities whose raw scales differ across domains. This section specifies what each component measures, what evidence it consumes, and which common confound it is intended to block.

F.1 Disagreement

Disagreement D averages semantic separation and action-level separation. Semantic separation detects incompatible claims after paraphrase normalization. Action separation compares the next consequential decision in a domain-specific representation. Code uses edit location, test selection, and normalized patch intent. Tool use compares function name, normalized arguments, and call order. Mathematics compares the next derived proposition and its logical relation to the prefix. QA compares atomic claims, resolved entities, and cited evidence.

A large semantic distance with the same normalized action is treated as presentation variation. A large action distance can still receive low overall PCS when neither action has discriminating evidence. This component therefore identifies a collision without deciding that the collision is useful.

F.2 COUNTERFACTUAL UTILITY

Counterfactual utility U is the absolute verified value difference between aligned branches. Both branches begin at the same student prefix and use the same horizon cap. A high value means that the selected assumption changes an executable or evidence-grounded outcome. It does not indicate which textual explanation sounds more plausible.

The branch-value vector contains execution validity, evidence support, goal progress, and constraint satisfaction. Task adapters set these fields from native outputs before the scalar value is formed. When both branches fail for unrelated reasons, utility remains low even if their language differs substantially.

F.3 EVIDENCE CONSISTENCY

Evidence consistency E is the minimum support assigned to the claims needed to prefer one branch. The minimum is used because one unsupported decisive step can invalidate an otherwise well-supported continuation. Evidence includes test output, executable environment state, symbolic checks, or passages with provenance. Model agreement and self-reported confidence do not raise E without a corresponding native record.

The fixed evidence gate $\epsilon = .70$ is applied before weighted aggregation. A collision below the gate cannot be retained by compensating with high disagreement or future reward.

F.4 RECOVERABILITY

Recoverability Q measures whether the preferred correction can be executed from the selected student state. A branch receives low recoverability when it restarts the task, discards valid prefix work, or relies on information unavailable at the intervention point. Recoverability is high when a local decision change can redirect the remaining trajectory within the recorded horizon.

This component separates self-correction from post-hoc replacement. A correct answer generated from scratch may be useful for response KD, but it does not show that the student can recognize and repair the current error.

F.5 FUTURE IMPROVEMENT

Future improvement F is the positive change in verified reward-to-go after substituting the preferred branch on a distillation-training trajectory. It connects a localized correction to downstream task progress. Negative and zero changes are clipped at zero. It is never computed from a final-evaluation instance, and no teacher or verifier is available to the frozen student during evaluation. The future-reward-only ablation reaches 44.8%, compared with 49.6% for full PCS, so F does not determine the result by itself.

F.6 TASK-CONDITIONAL PERCENTILE CALIBRATION

Raw verifier outputs have different meanings across environments. A symbolic score of .8 and an evidence-entailment score of .8 are not assumed to be comparable. For each component and verifier type, the empirical cumulative distribution is fitted on the disjoint calibration partition. Raw values are then converted to task-conditional percentiles. The calibration maps and all thresholds are frozen before final evaluation.

Component	Primary record	High value requires	Does not establish
D	Normalized branch claims and actions	A consequential difference after paraphrase removal	That either branch is correct
U	Aligned verified branch values	The assumption changes a testable outcome	That the preferred outcome is grounded
E	Raw task-native evidence and provenance	Support for every decisive preferred-branch claim	That recovery is possible from the current state
Q	Prefix, available information, and remaining horizon	A local repair without restart or future leakage	That the repair improves terminal reward
F	Verified reward-to-go before and after intervention	Positive downstream improvement	That the intervention is minimal or evidence-consistent

Table 16: Interpretation of PCS components and the claims each component cannot support alone.

The calibrated percentiles use weights $(.15, .15, .25, .20, .25)$ for (D, U, E, Q, F). A record is retained at PCS .62 or above after passing the evidence gate. Attribution then maps the score to the minimal intervention-linked span. The span begins where the unsupported assumption affects a decision and ends when discriminating evidence becomes available or the corrective action is selected. This rule is an operational attribution heuristic, not an estimator of a causal effect. Tokens outside the span retain unit on-policy weight.

G Verifier and Generated-Supervision Quality Controls

G.1 Rejection taxonomy

The verifier is a gate over task-native evidence. It cannot rescue an invalid intervention by assigning a high verbal score. Table 17 lists the conditions checked before PCS is computed.

Failure mode	Audit	Disposition
Observed-fact negation	The alternative contradicts a value explicitly present before the selected step.	INVALID
Non-minimal intervention	Goal, entity, constraint, or an unrelated premise changes across branches.	INVALID
No behavioral separation	Normalized actions and verified outcomes agree despite different wording.	EQUIVALENT
No discriminating evidence	Neither branch cites execution, state, symbolic, or passage evidence above the fixed gate.	INCONCLUSIVE
Future-information leakage	A branch reasons from an observation before executing the action that obtains it.	Reject record
Unrecoverable restart	The successful branch discards the prefix or restarts with unavailable information.	Set recoverability to zero
Unsafe reward shortcut	A branch reaches task reward while violating an environment constraint.	Reject branch
Cross-split overlap	A training or calibration key matches a final-evaluation key.	Exclude before rollout
Model agreement without support	Teacher roles converge, but no task-native signal supports the decisive claim.	Reject record

Table 17: Quality-control rules applied before a record can affect the student.

G.2 Evidence hierarchy by domain

For code, a targeted regression test is evaluated before the broader repository suite, and both outrank a textual claim that a patch is correct. For tool use, normalized API arguments and executable state transitions outrank a natural-language description of what a call should return. For mathematics, symbolic equivalence, substitution, and explicit counterexamples outrank self-reported confidence. For grounded QA, provenance-preserving entailment of the necessary atomic claims outranks answer fluency.

The hierarchy separates evidence acquisition from evidence interpretation. A language model may propose which test, call, substitution, or passage would be discriminating. The corresponding task-native system supplies the result. The record is rejected when the result cannot be tied to the preferred decision.

G.3 AUDIT QUESTIONS FOR RETAINED RECORDS

For every retained collision, the pipeline answers the following questions from stored fields.

1. Was the selected proposition inferred rather than directly observed?
2. Did the alternative change exactly one assumption while preserving the prefix and task?
3. Do the branches make different predictions or select different normalized actions?
4. Was the decisive evidence available before the decision, or explicitly acquired afterward by a valid action?
5. Can the preferred branch recover from the current state without restarting the episode?
6. Which earliest evidence item separates the branches?
7. Which tokens express the unsupported inference and the corrective action?
8. Were task, split, prompt, model, token, environment, and version identifiers stored?

These questions also define the boundary of a false-positive PCS record. High branch divergence is not sufficient when one of the intervention, evidence, temporal-availability, or recoverability checks fails.

G.4 HUMAN AUDIT OF RETAINED COLLISIONS

Three annotators independently audit 500 retained collisions. They judge whether the collision is productive, whether the decisive evidence supports the preference, whether recovery is possible from the current state, and whether the preferred branch is correct. Precision compares the automatic positive decision with the majority human judgment; criterion-level κ is the mean pairwise Cohen's κ for that criterion.

Criterion	Precision	Cohen's κ
Productive collision detection	.91	.78
Evidence consistency	.94	.82
Recoverability judgment	.89	.74
Preferred-branch correctness	.92	.80

Table 18: Human audit of 500 retained collisions. Cohen's κ is the mean pairwise agreement for each criterion.

Annotator pair	Cohen's κ
A1–A2	.79
A1–A3	.76
A2–A3	.81
Mean	.79

Table 19: Pairwise annotator agreement aggregated across the four audit criteria.

The audit supports the quality of the retained sample while leaving room for verifier errors: recoverability is the hardest criterion, with precision .89 and $\kappa = .74$.

H More Ablation Studies

H.1 PCS components for every seed

Configuration	Seed 42	Seed 123	Seed 2027	Mean
Uniform disagreement	45.3	46.1	45.7	45.7
D only	44.0	44.8	44.4	44.4
E only	44.8	45.6	45.2	45.2
F only	44.4	45.2	44.8	44.8
$D + E$	47.0	47.8	47.4	47.4
Full PCS	49.2	49.975	49.6	49.575

Table 20: Per-seed macro-average success for PCS component ablations. Values retain the precision stored in the experiment record.

Future improvement alone reaches 44.8%, 4.8 points below full PCS. Disagreement plus evidence reaches 47.4%, leaving a 2.2-point gap. The full score therefore uses information that is not recovered by selecting large outcome changes or well-supported disagreement alone.

H.2 Counterfactual and teacher-role controls

Configuration	Seed 42	Seed 123	Seed 2027	Mean
No counterfactual inversion	43.4	44.2	43.8	43.8
Single-teacher critique prompt	43.4	44.2	43.8	43.8
Same model, three roles	49.175	49.975	49.575	49.575
Three heterogeneous teachers	49.8	50.6	50.2	50.2

Table 21: Construction and teacher-role controls under the recorded experiment settings. The main method uses one shared Qwen2.5-72B model with three role-specific prompts.

The heterogeneous-teacher control reaches 50.2%, while the shared-model role-separated method reaches 49.6% after standard rounding. The principal result does not require three independently hosted teacher models. Heterogeneous teachers add 0.6 points and incur a different deployment footprint, so the shared-model configuration remains the reported method.

H.3 Collision novelty

Novelty accounting item	Count or score
Retained preferred branches	12,480
Exact action present in an initial response	1,960
Semantically present by bidirectional entailment	2,290
Absent from every pre-collision teacher response	8,230
Full C-OPD success	49.6
Success after removing novel records	43.9
Standard OPD success	41.8
Observed improvement attributed by removal and retraining	72.564%

Table 22: Collision-novelty record. Semantic presence uses DeBERTa-v3-large with a bidirectional entailment threshold of 0.80.

Exact matches are detected after canonicalizing tool calls, code edit targets, mathematical claims, and answer propositions. Remaining pairs are checked for bidirectional entailment. A preferred branch is labeled novel only when none of the initial teacher responses entails its decisive corrective action. Removing the 8,230 novel records and retraining lowers success to 43.9%. The resulting

72.6% contribution describes this controlled removal-and-retraining intervention; it is not the raw fraction of records marked novel.

H.4 PCS VALIDITY BY DOMAIN

Domain	Collisions	ρ correction	95% CI	ρ reward	95% CI
Code	2,880	.61	[.58,.64]	.49	[.46,.52]
Tool	3,240	.64	[.61,.66]	.52	[.49,.55]
Math	3,060	.54	[.51,.57]	.43	[.40,.46]
QA	3,300	.49	[.46,.52]	.38	[.35,.41]
Pooled	12,480	.58	[.56,.60]	.46	[.44,.48]

Table 23: Spearman correlations between PCS and correction success or later reward gain. Pooled correction-success AUC is .76 and expected calibration error is .041.

The positive association is present in every domain, including QA where evidence is less executable than code or tool feedback. Pooled reward gain rises by $+.18$ in the top PCS group compared with $+.04$ in the bottom group. These analyses measure ranking and calibration; they do not replace the removal and retraining ablations above.

I TASK-SPECIFIC DATA, EVALUATION, AND DEDUPLICATION

I.1 DATA PARTITIONS

Domain	Distillation training pool	PCS calibration	Final evaluation
Code	SWE-bench train subset, 1,994 issues	300 held-out SWE-bench training issues	SWE-bench Verified, 500
Tool	BFCL v3.0.1 multi-turn and adversarial training pool, 6,000 episodes	600 held-out BFCL episodes	BFCL v3.0.1 fixed test, 1,200
Math	MATH train subset, 7,000 problems	500 held-out MATH training problems	MATH-500, 500
QA	HotpotQA v1.1 distractor fixed train subset, 20,000 examples	fixed validation subset, 1,000 examples	disjoint fixed evaluation subset from HotpotQA v1.1 validation, 1,000 examples

Table 24: Mutually disjoint training, calibration, and final-evaluation pools.

One unified student is trained across the four domains. Domains are sampled with probability 1:1:1:1, and tasks are sampled uniformly within the selected domain. The combined training pools contain 34,994 task instances. Each sampled training task produces four on-policy rollouts, giving the recorded 1:4 task-to-rollout ratio.

The final-evaluation collections are used only after the training configuration and checkpoint have been frozen. No final-evaluation instance, trajectory, environment feedback, or teacher-generated supervision derived from an evaluation instance is used for training, calibration, checkpoint selection, or hyperparameter tuning.

I.2 DEDUPLICATION KEYS

Code uses the repository and issue identifier as the primary key. We additionally compare normalized problem text and touched paths to detect duplicated issue formulations. Tool use normalizes the API signature, required arguments, scenario template, and adversarial state pattern. Mathematics canonicalizes whitespace, symbolic notation, answer choices, and the normalized problem statement before hashing. QA uses the normalized question together with the supporting-document identifiers and resolved entities. Any key appearing in more than one partition is assigned to one partition before rollout generation and excluded from the others.

Deduplication is performed before teacher calls. This ordering prevents a removed item from leaving teacher-generated traces, verifier outputs, or cached environment feedback in the training pool.

I.3 Success conditions and verification channels

Code. SWE-bench Verified uses the official 500 instances and harness v1.2.0. A solution succeeds only when all official fail-to-pass and pass-to-pass tests succeed in a valid container. The verification hierarchy is newly added regression tests, existing targeted tests, broader repository tests, static checks, and localized code evidence.

Tool use. BFCL v3.0.1 contains 1,200 fixed multi-turn test episodes, including 400 adversarial-observation episodes. Success requires the correct normalized function, arguments, call order, and final state. Observations store the tool name, arguments, step, and timestamp or state version when available.

Mathematics. MATH-500 is scored by normalized exact match or SymPy 1.13 equivalence. Domain restrictions, quantifier scope, case completeness, and dependencies between derived statements are retained in the verifier record. A changed final answer without a discriminating intermediate check does not qualify as a productive collision.

Grounded QA. The final set contains 1,000 fixed, disjoint examples from HotpotQA v1.1 validation. Success requires exact answer match, supporting-fact F1 of at least 0.80, and entailment support for every atomic claim needed by the answer. Passage identifiers and evidence spans remain attached to each claim throughout verification.

J Training and Reproducibility Details

J.1 Complete fixed configuration

Item	Fixed specification
Student	One unified Qwen2.5-7B-Instruct model; full-parameter bfloat16 fine-tuning with DeepSpeed ZeRO-3.
Teacher roles	One Qwen2.5-72B-Instruct model with separate response, axiom, counterfactual, and verification prompts.
Hardware	Eight A100-80GB GPUs for the student and four H100-80GB GPUs for teacher generation.
Optimization	AdamW; learning rate 10^{-5}; global batch 64; cosine decay; 500 warmup steps; weight decay 0.1; gradient clipping 1.0.
Lengths	16,384-token context; 2,048-token generation cap; 1,024-token counterfactual-branch cap.
Schedule	12,000 updates; four rollouts per sampled task; 48,000 rollout environment calls per seed.
Task mixture	Domain sampling 1:1:1:1; uniform sampling within a domain; collision and ordinary successful traces mixed 1:1.
Decoding	Student and counterfactual: temperature 0.7, top-$p = .95$; response and axiom: 0.2, .95; verification: greedy, temperature 0, top-$p = 1$.
Risk model	Fixed logistic scorer; $(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = (1, 1, 1, 1)$; top $K = 3$.
PCS	$(\beta_D, \beta_U, \beta_E, \beta_Q, \beta_F) = (.15, .15, .25, .20, .25)$; $\epsilon = .70$; retention threshold .62; $\lambda = 2$; $w_{\max} = 4$.
Loss	$(\eta, \gamma, \mu) = (0.5, 0.1, 0.02)$.
Budget and seeds	800M teacher tokens per seed; seeds 42, 123, and 2027.
Checkpoint	Step-12,000 checkpoint selected from disjoint calibration performance and frozen before final evaluation.

Table 25: Complete fixed recipe used for the reported experiments.

J.2 Risk selection and collision filtering

The risk scorer is the fixed logistic form in Equation 2. All four inputs are normalized to $[0, 1]$, use coefficient one, and are computed from the structured axiom record. The system retains the top three nonredundant assumptions at a selected step. Near-duplicate assumptions are clustered before counterfactual generation so that repeated wording cannot consume the branch budget.

A collision is removed when its assumption is already entailed by direct observation and the inversion contradicts that observation; the branches change unrelated task variables; no discriminating evidence

is cited; the preferred branch uses future information before acquiring it; or the behavioral difference reduces to presentation. Evidence below .70 fails the hard gate before the PCS threshold is applied.

J.3 Matched-budget batching and stability

Each training batch has a 65,536-token teacher allowance. Axiom extraction, counterfactual construction, and verifier rationales draw from the same pool, and unused tokens return to that pool. Token weights are clipped at 4.0 and normalized within a batch. Collision records and ordinary successful on-policy traces are mixed 1:1. The conservative reference KL is applied on collision and noncollision states.

Every run record is keyed by method, seed, and the step-12,000 checkpoint. It stores benchmark and environment versions, prompt identifiers, decoding settings, student-trajectory hashes, collision components and dispositions, token and environment counts, hardware time, and final evaluation counts.

J.4 Controlled-comparison map

Comparison	Fixed and changed factors	Average success
Response KD → standard OPD	Same student and token budget; offline responses become supervision on student-visited states.	$38.7 \rightarrow 41.8$
Standard OPD → C-OPD	Same student, teacher, and 800M teacher tokens; adds structured collision construction and task-native verification.	$41.8 \rightarrow 49.6$
Standard OPD → compute-matched C-OPD	Same 800M teacher tokens and 1,028.2 mean GPU-hours; adds collisions with reduced verification frequency.	$41.8 \rightarrow 47.3$
DemoPSD/CROP → C-OPD	Same unified reproduction protocol; changes the disagreement-selection and supervision mechanism.	$43.9/45.1 \rightarrow 49.6$
Monolithic critique → three roles	Same Qwen2.5-72B teacher and token budget; changes the interaction contract.	$43.8 \rightarrow 49.6$
No inversion → full PCS	Restores controlled counterfactual inversion.	$43.8 \rightarrow 49.6$
Uniform weighting → full PCS	Replaces equal disagreement weights with evidence-gated PCS.	$45.7 \rightarrow 49.6$
$D + E \rightarrow$ full PCS	Restores utility, recoverability, and future improvement.	$47.4 \rightarrow 49.6$
Remove novel records → full data	Restores corrective branches absent from all initial responses.	$43.9 \rightarrow 49.6$

Table 26: What changes in each controlled comparison. Matching teacher tokens does not claim matching verifier compute.

K Additional Full Output Examples

The following records serve the same explanatory purpose as full reasoning examples in prior self-correction appendices. They are schematic protocol demonstrations, not sampled benchmark transcripts. Each example ends with the exact lesson that would enter distillation and the condition under which the record would be rejected.

K.1 Code example: failure locus after a repeated test signature

Student decision. "The parser appears in the traceback, so the next repair should relax the parser's validation branch."

Axiom record. The traceback is observed. The proposition that the parser violates the intended contract is inferred. The first edit has already reproduced the same test signature, which increases contradiction and downstream dependency. The mutually exclusive alternative is that the caller violates the parser's documented precondition.

Original branch. Relax the validation branch and rerun the targeted test. This branch predicts that the malformed value should now be accepted and that pass-to-pass behavior remains unchanged.

Counterfactual branch. Inspect the call-site construction, assert the documented precondition, and rerun the same test. This branch predicts that the caller produces a value outside the allowed domain and that repairing the construction preserves parser behavior.

Verifier record. The first branch is rejected if the targeted failure remains or a pass-to-pass test regresses. The second is preferred only when the call-site value violates the contract and the full required test set succeeds. The earliest discriminating evidence is the observed call-site value against the existing contract.

Distillation target. The weighted span teaches the agent to separate "appears on the failing path" from "is the failure locus," then to use the repeated signature as a reason to acquire new evidence. The rest of the successful patch remains ordinary trajectory supervision.

Rejection condition. If both branches change multiple files, use different test inputs, or cannot be compared under the same repository state, the intervention is non-minimal and the record is discarded.

K.2 Tool example: pagination before fallback

Student decision. "The target is absent from the returned list, so I should invoke the creation endpoint."

Axiom record. The first page lacks the target. The decision assumes that the response is complete and that the entity resolver used the intended identifier. The alternative states that an unconsumed page token or alias can change the result.

Original branch. Call the creation endpoint using the current identifier. The branch predicts that creation is permitted and yields one new object.

Counterfactual branch. Consume the next-page token, normalize the identifier, and repeat the lookup. The branch predicts either a matching object or a complete negative result that licenses creation.

Verifier record. The executable environment determines whether another page or alias exists. A returned object rejects immediate creation. Exhausted pagination under the verified identifier makes the branches behaviorally equivalent, so the collision is not upweighted.

Distillation target. The weighted decision is the jump from partial observation to global absence. The desired policy checks completion metadata before choosing an irreversible or duplicating action.

Rejection condition. If the benchmark exposes no pagination or entity-resolution state, the alternative cannot be tested and the record is marked inconclusive.

K.3 Mathematics example: sufficient versus necessary

Student decision. "The expression is nonnegative under condition C, hence all solutions satisfy C."

Axiom record. The derivation under C is observed in the prefix. Exhaustiveness is inferred. The alternative asserts that a feasible case outside C may satisfy the original relation.

Original branch. Proceed under C and report the corresponding solution set. The branch predicts that substitution outside C cannot satisfy the original statement.

Counterfactual branch. Restore the omitted case and search symbolically for a satisfying point outside C. The branch predicts either a counterexample or a proof that the omitted case is empty.

Verifier record. A valid substituted counterexample rejects necessity. If symbolic simplification proves the outside case infeasible under the original domain restrictions, the original branch is retained. A second model's agreement does not settle the comparison.

Distillation target. The intervention-linked span is the logical conversion from implication to equivalence. The preferred continuation introduces the missing case before the final answer, without penalizing the algebra already verified inside C.

Rejection condition. If the alternative silently changes the domain or the original problem's quantifier, it is not a minimal inversion and is discarded.

K.4 Grounded QA example: entity and temporal scope

Student decision. "The first passage contains the requested name and states relation r, so the answer is entity e."

Axiom record. The name and relation occur in the passage. Identity, temporal scope, and entailment of the final atomic claim are inferred. The alternative states that the surface name resolves to another entity or period.

Original branch. Answer from the first passage and cite its matching sentence. The branch predicts that the cited document uniquely identifies the requested entity and time.

Counterfactual branch. Resolve the entity against the second passage, compare dates, and decompose the proposed answer into atomic claims. The branch predicts that at least one claim lacks support under the original citation.

Verifier record. Passage identifiers, dates, and evidence spans determine the preferred branch. The counterfactual is preferred only when the additional source changes entity resolution or supplies the missing entailment. More verbose reasoning receives no credit by itself.

Distillation target. The weighted span covers commitment after a lexical match and the corrective request for corroboration. When neither document resolves the ambiguity, the preferred action is to abstain or retrieve more evidence.

Rejection condition. If the second passage is unavailable to the agent and the branch cites it without retrieval, the record contains future-information leakage and is rejected.

L Limitations, Failure Modes, and Broader Impact

L.1 Where productive collision can fail

The method depends on a feedback channel that can discriminate between branches. Open-ended generation may provide no executable or provenance-preserving signal. In that setting, PCS inherits verifier-model bias and should not be interpreted as correctness. A second failure arises when all teacher roles share the same false premise. Structured disagreement can expose alternatives inside the teachers' hypothesis space, but it cannot guarantee discovery of a missing hypothesis.

Counterfactual construction also introduces selection effects. High-risk assumptions that are easy to invert and verify are more likely to enter training than subtle assumptions with delayed consequences. Percentile calibration aligns score scales; it does not remove this coverage bias. Environment feedback can be exploitable, and a branch can optimize a visible test while violating an unstated requirement. Constraint checks and pass-to-pass tests reduce this risk without eliminating it.

Finally, additional verification has a measurable cost. The reported full method uses 35.5% more GPU-hours and about 18.4K more environment calls per seed than standard OPD. The matched-total-compute run reaches $47.3 \pm 0.5\%$, below full C-OPD but above standard OPD. This control does not eliminate the latency and systems cost of verification in deployment.

The experiment uses three training seeds. Raw seed results, all 12 domain–seed gains, paired bootstrap intervals, and paired permutation tests are reported. The resampling analysis quantifies uncertainty over stored evaluation units; three training seeds still provide limited information about tail variation across complete retraining runs.

Assumption validity, counterfactual minimality, verifier sufficiency, and attribution-span quality are checked automatically by the structured contracts and task-native evidence. The 500-collision human audit measures agreement on retained records, but it does not guarantee equal verifier reliability on every task, rejected record, or out-of-distribution trajectory.

L.2 Operational safeguards

Counterfactual tool and code branches should run in sandboxes that prohibit irreversible external actions. Stored collision records should retain raw evidence and version identifiers so that an operator can distinguish model-generated claims from environment outputs. Production agents should keep human approval boundaries for actions whose consequences extend beyond the evaluation environment. Improving recovery can make an agent more capable and persistent; evaluation should therefore measure whether the same mechanism changes the risk profile of sensitive tool use.

The method is designed to preserve supported decisions. Equivalent branches receive no enhanced weight, and a preferred branch must cite discriminating evidence. These checks reduce the tendency to teach habitual doubt, although they cannot guarantee that a deployed agent will request the right evidence under distribution shift.

M Record-Level Reproducibility Checklist

The following checklist states what must be recoverable from the experiment artifacts for each reported run. It is included to make the boundary between a prose description and a reproducible record explicit.

M.1 Data and split manifest

The manifest records the source dataset and version, original identifier, normalized deduplication key, assigned partition, and exclusion reason when applicable. A split audit recomputes keys across the training, PCS-calibration, and final-evaluation collections before any rollout or teacher cache is loaded. The audit also checks that no derived trajectory, teacher completion, verifier result, or environment snapshot from a final-evaluation identifier appears in the training store.

M.2 Model and prompt manifest

Each completion is linked to the student or teacher model identifier, prompt-role version, decoding configuration, context limit, generation cap, and random seed. Role labels alone are insufficient because two prompt revisions can produce different collision distributions while using the same hosted model. The manifest therefore stores prompt hashes together with the rendered task adapter.

M.3 Collision and verifier manifest

Every candidate collision retains the selected step, direct observations, inferred assumption, minimal alternative, aligned branches, raw task-native evidence, terminal status, component values before and after calibration, PCS, attributed token span, and rejection reason. Resource fields store teacher tokens and environment calls even for rejected records so filtering cannot hide cost.

M.4 Training manifest

The training record identifies the method, seed, update number, sampled domain, task identifier, four rollout identifiers, collision-to-ordinary trace mixture, effective token weight, optimizer state, and reference checkpoint. The step-12,000 checkpoint is linked to the calibration record used for selection. No final-evaluation metric is present in the checkpoint-selection table.

M.5 Evaluation and reporting manifest

Evaluation records contain the frozen checkpoint hash, benchmark harness or scorer version, per-instance outcome, and task-specific behavior events. Aggregate tables are regenerated from these per-instance records using seeds 42, 123, and 2027. Reported uncertainty is the sample standard deviation over the three seed-level results. Counts used for repeated failures, unsupported continuation, and recovery remain linked to the trajectory step and native evidence that triggered the event.

Artifact group	Minimum reconstruction target	Principal leakage or reporting check
Data manifest	Exact partition and normalized key for every task	Cross-partition key and derivative-cache scan
Prompt manifest	Rendered role prompt and decoding configuration	Prompt hash and model-version match
Collision manifest	Assumption, branches, evidence, scores, span, and disposition	Temporal availability and native-evidence audit
Training manifest	Update, rollout, mixture, optimizer, and checkpoint linkage	No final-evaluation field in selection records
Evaluation manifest	Per-instance success and behavior events under a frozen checkpoint	Regenerate every aggregate and seed statistic

Table 27: Minimum artifact fields needed to reconstruct the reported comparisons.