AGENTS, THINK ONLY WHEN IT HELPS

Anonymous authors
Paper under double-blind review

ABSTRACT

LLM agent distillation transfers agentic capabilities to student models by training on complete teacher trajectories that typically interleave thoughts and actions. We question how critical thought supervision is for agents' ultimate action performance. To answer this, we disentangle their impacts by independently perturbing teacher thoughts and actions during distillation. On widely adopted agent benchmarks across model scales, altering or discarding thoughts has little effect on student performance. We find, however, that thought supervision becomes important when information about the thought is not disclosed in the corresponding action during training. By identifying such discrepant cases, our findings challenge the common practice of distilling entire teacher trajectories for LLM agents, and offer a new insight that thought supervision can be a discernible choice for efficiency depending on whether the information of thought is disclosed in the action. An anonymized version of the code is provided in the supplementary material for reproducibility.

1 INTRODUCTION

LLM agent distillation has emerged as a promising approach for transferring the capabilities of powerful teacher agents to student models through teacher-generated trajectories (Hashemzadeh et al., 2025; Qiu et al., 2025; Liu et al., 2026b; Kang et al., 2026; Lyu et al., 2025). Existing agent distillation methods typically build on the ReAct framework (Yao et al., 2023), in which trajectories interleave a natural-language *thought*, an environment-facing *action*, and the resulting *observation*. Standard practice trains student models on complete teacher trajectories, jointly supervising both thoughts and actions.

On widely adopted agent benchmarks, thoughts dominantly occupy the trajectory budget. Nevertheless, we often observe that such a lengthy thought process does not always lead to correct reasoning and a correct answer, as shown in the example in Figure 1: a student trained on the complete trajectory follows an incorrect reasoning path and produces an incorrect answer, whereas a student trained without thoughts ends up with the correct one.

These observations make us question how critical thought supervision is for LLM agents' ultimate action performance. We examine this question by intervening in the thought and action channels, observing that corrupting teacher thoughts preserves near-clean student performance, whereas corrupting teacher actions causes substantial degradation.

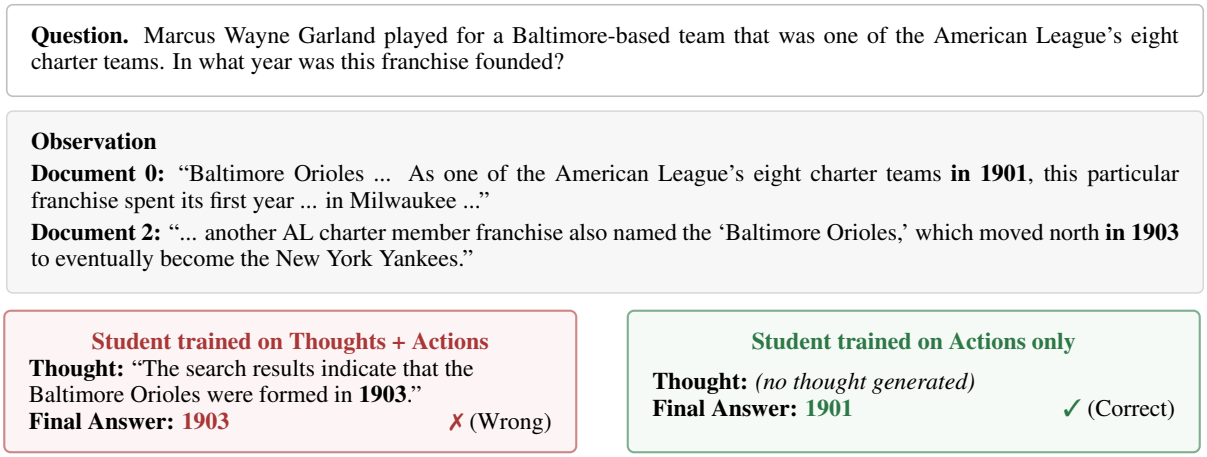


Figure 1: Two students distilled from the same teacher trajectories on HotpotQA (Yang et al., 2018), differing only in whether teacher thoughts are included in the supervision during training.

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.

This asymmetry persists across student scales, teacher models, student model families, corruption granularities, and distillation objectives. On these benchmarks, we attribute this to the *information of the thought being already disclosed in the action during training*, which can make thought supervision not essential or detrimental.

In contrast, we find that on quantitative reasoning benchmarks, thought information is not (or only a little) disclosed to the action during training, and thought supervision matters. In these settings, discarding teacher thoughts consistently degrades student performance, indicating that thought supervision leads to better reasoning and a greater likelihood of correct answers. To test whether this dependence arises from the disclosure of thought information conveyed by the actions, we increase the thought information disclosed to the action step without altering the underlying computation, and find that student models become less reliant on thought supervision.

Our central finding is that the importance of thought supervision depends on how actions are represented: specifically, on whether the information contained in the thought is already exposed in the action. When the thought information is already exposed in the action, the thought can be omitted with little or no loss in downstream performance—at times even improving it. When such information is not exposed, thought supervision yields more correct reasoning and answers. This challenges the default practice of indiscriminately distilling complete teacher trajectories, and instead motivates a more efficient, discerning use of thought supervision that preserves decision-relevant information while avoiding supervision on redundant thought tokens.

Our contributions are as follows:

1. By disentangling thought and action supervision in agent distillation, we show that on widely adopted agent benchmarks, even corrupting or removing thoughts still largely preserves student performance.
2. We find that the importance of thought supervision depends on how much the thought information is disclosed in the action.
3. We show that, when the thought's information is already disclosed in the action, both training and inference cost can be saved by omitting thought supervision with little performance loss.

2 Related Work

Distillation for LLM Agents. Transferring agentic competence to a student model from a teacher model has converged on a common recipe: roll out a tool-using teacher in a ReAct-style loop (Yao et al., 2023) and fine-tune the student on the resulting thought–action–observation trajectories (Chen et al., 2023; Kang et al., 2026). AgentTuning (Zeng et al., 2024) fine-tunes open LLMs on multi-task teacher trajectories in which every action is paired with a thought. Agent-FLAN (Chen et al., 2024) attributes the inadequate agent learning of such corpora to the entanglement of format following and reasoning, and recasts them as multi-turn natural conversation. Tang & Zhao (2026) select teacher rollouts that match student capacity. A few methods depart from trajectory imitation, transferring reusable task modules without training on traces (Qiu et al., 2025) or replacing full teacher rollouts with student-generated trajectories that receive sparse corrections at their earliest errors (Lyu et al., 2025). Several studies rethink how teacher reasoning is used in agent distillation. Hashemzadeh et al. (2025) replace free-form teacher reasoning with sub-goals over teacher trajectories and train a hierarchical agent whose planning and execution modules are supervised separately. Penaloza et al. (2026) study distillation when teacher reasoning is unavailable, using expert action trajectories as reference information for self-distillation. Liu et al. (2026b) ablate reasoning and action supervision, and Liu et al. (2026a) study whether chain-of-thought training improves direct action prediction or actions produced after reasoning. Unlike those studies, our work finds that the value of thought supervision varies with the extent to which thought information is exposed in the demonstrated action.

Reasoning Trace Supervision. Prior work has explored teacher-generated reasoning traces as an additional source of supervision for smaller models. Chain-of-thought distillation has shown that training students on intermediate rationales, rather than final answers alone, can improve reasoning performance and data efficiency (Magister et al., 2023; Hsieh et al., 2023; Li et al., 2023). However, subsequent work has questioned whether these gains require the student to faithfully reproduce or rely on the demonstrated reasoning process. For example, Wadhwa et al. (2024) find that rationale supervision can remain beneficial even when rationales are placed after the target labels, suggesting that its value does not necessarily arise from enabling step-by-step reasoning at inference time. Chen et al. (2025) further show that the effectiveness of reasoning supervision varies with rationale granularity and student capacity. These findings suggest that reasoning traces can provide useful supervision, but leave open when and why they are critical. This question takes on a distinct form in agent distillation, where each teacher thought is paired with an environment-facing action that may already expose part of the information contained in the reasoning trace. We investigate how the information exposed in demonstrated actions relates to the value of thought supervision.

3 INTERVENING ON THOUGHT AND ACTION SUPERVISION

3.1 AGENT TRAJECTORIES AND DISTILLATION

We build on top of the ReAct paradigm (Yao et al., 2023) that existing agent distillation studies have widely adopted (Hashemzadeh et al., 2025; Qiu et al., 2025; Liu et al., 2026b; Kang et al., 2026; Tang & Zhao, 2026). In this framework, a teacher trajectory τ is an interleaved sequence of observations, thoughts, and actions,

$$\tau = (o_0, t_1, a_1, o_1, t_2, a_2, o_2, \ldots, t_T, a_T) \tag{1}$$

where o_i denotes the environment observation at step i, t_i the natural language thought preceding the i-th action, a_i the action executed in the environment, and T the total number of decision steps. A trajectory is generated by a teacher policy π_{teacher} and retained for distillation only if it reaches a successful terminal state, that is, $\text{Success}(\tau) = 1$ under the benchmark-specific success criterion. Then, we train the student parameters θ via supervised fine-tuning (SFT) using a standard autoregressive next-token prediction objective, with observation tokens o_i masked out from the loss.

$$\mathcal{L}_{\text{SFT}}(\theta) = -\mathbb{E}_{\tau \sim \mathcal{D}} \left[\sum_{j \in \mathcal{S}(\tau)} \log \pi_\theta(y_j \mid y_{<j}) \right] \tag{2}$$

where y_j denotes the j-th token in τ, $\mathcal{S}(\tau)$ is the set of token positions corresponding to thoughts and actions, and $\mathcal{D}$ is the set of successful teacher trajectories. We additionally evaluate a KL-based distillation objective as an alternative training objective to test whether our findings are robust to the choice of distillation objective in Section 4.6.

3.2 CORRUPTION OPERATORS

We perturb teacher thoughts while keeping the corresponding observations and actions fixed. Given a clean trajectory τ, we define a corruption operator $C_{f,p} : \tau \mapsto \tau'_{f,p}$, where f specifies the intervention and $p \in [0, 1]$ controls the corruption ratio. For each thought t_i, we independently sample $z_i \sim \text{Bernoulli}(p)$ and define

$$t'_i = \begin{cases} f(t_i), & \text{when } z_i = 1 \\ t_i, & \text{when } z_i = 0 \end{cases} \tag{3}$$

We consider three interventions:

$$\begin{aligned} \texttt{Drop:} \quad & f_{\text{drop}}(t_i) = \varnothing \\ \texttt{Replace:} \quad & f_{\text{repl}}(t_i) = t^*, \quad t^* \sim \mathcal{D}^t_{\text{unrel}} \\ \texttt{Paraphrase:} \quad & f_{\text{para}}(t_i) = \pi_{\text{paraphrase}}(t_i) \end{aligned} \tag{4}$$

`Drop` removes thought supervision from the training trajectory, `Replace` substitutes a thought with one from an unrelated task instance, and `Paraphrase` preserves its semantics while altering surface form. These interventions allow us to assess how omitting thought spans, replacing their content, and changing their wording affect the distilled student's performance. Here, $\mathcal{D}^t_{\text{unrel}}$ denotes the distribution of thoughts drawn from other task instances within the same benchmark, and $\pi_{\text{paraphrase}}$ denotes the paraphrasing model. We instantiate $\pi_{\text{paraphrase}}$ with Qwen2.5-32B-Instruct, the same model used as our default teacher. The resulting corrupted trajectory is

$$\tau'_{f,p} = (o_0, t'_1, a_1, o_1, \ldots, t'_T, a_T) \tag{5}$$

Replaced and paraphrased thought tokens remain supervised targets rather than being masked out, so the student is trained to generate them. At evaluation time, all students receive the ReAct-style prompts (Figures 12–14). Students trained with all thoughts dropped (p=1) nonetheless generate actions directly, as they have never observed thoughts during training.

Likewise, we intervene on the action channel while keeping all thoughts t_i and observations o_i fixed. We apply the same corruption procedure and corruption ratio p defined above to actions:

$$\begin{aligned} \texttt{Drop:} \quad & f_{\text{drop}}(a_i) = \varnothing \\ \texttt{Replace:} \quad & f_{\text{repl}}(a_i) = a^*, \quad a^* \sim \mathcal{D}^a_{\text{unrel}} \end{aligned} \tag{6}$$

Here, $\mathcal{D}^a_{\text{unrel}}$ denotes the distribution of actions drawn from other task instances within the same benchmark. We do not apply paraphrasing to actions, as changes in surface form can alter their executable semantics.

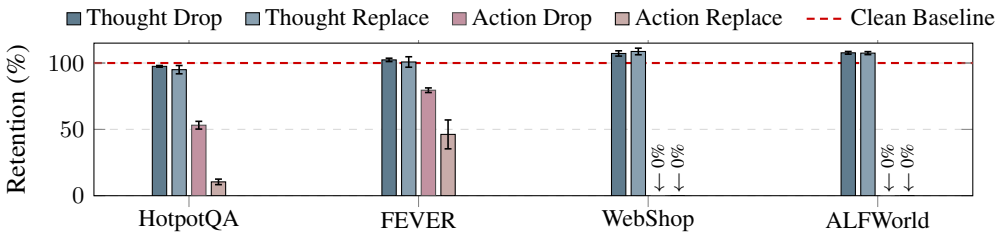


Figure 2: Retention under thought and action corruption at p=1.0 (Qwen2.5-3B-Instruct student). Error bars denote standard deviation across three runs. Action drop and action replace yield 0% retention on WebShop and ALFWorld.

3.3 Experimental Setup

Unless otherwise specified, teacher trajectories are generated by Qwen2.5-32B-Instruct (Yang et al., 2024), and students are initialized from Qwen2.5-3B-Instruct and trained with the SFT objective as in Eq. 2. To study robustness to student capacity, we evaluate Qwen2.5-Instruct students at four scales, 0.5B, 3B, 7B, and 14B. For each benchmark, all intervention conditions are constructed from the same underlying teacher trajectories and use identical training settings, differing only in the applied intervention. All students are evaluated under identical decoding settings, and results are averaged over three independent training and evaluation runs.

Evaluation Metrics. We evaluate each student model using task accuracy and its retention relative to the corresponding clean-trajectory baseline. Task accuracy is defined as

$$\text{Accuracy}(\pi_\theta, \mathcal{B}) = 100 \times \frac{1}{|\mathcal{B}|} \sum_{x \in \mathcal{B}} \mathbb{1}\left[\text{Success}\left(\tau_\theta(x)\right) = 1\right] \tag{7}$$

where $\mathcal{B}$ denotes the benchmark evaluation set, $\tau_\theta(x)$ is the trajectory produced by rolling out the student policy π_θ on task instance x, and Success$(\cdot)$ denotes the benchmark-specific success criterion. For HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), Bamboogle (Press et al., 2023), and MuSiQue (Trivedi et al., 2022), we use GPT-4o-mini (OpenAI, 2024) as an LLM judge to assess semantic equivalence between the predicted and reference answers. Full evaluation details and the judge prompt are provided in Appendix B. Besides, to quantify performance degradation under corruption, we report *retention*, defined as the accuracy of a student trained under intervention (f, p), $\pi_\theta^{(f,p)}$, relative to that of the corresponding student trained on clean trajectories, π_θ:

$$\text{Retention}(f, p, \mathcal{B}) = 100 \times \frac{\text{Accuracy}(\pi_\theta^{(f,p)}, \mathcal{B})}{\text{Accuracy}(\pi_\theta, \mathcal{B})} \tag{8}$$

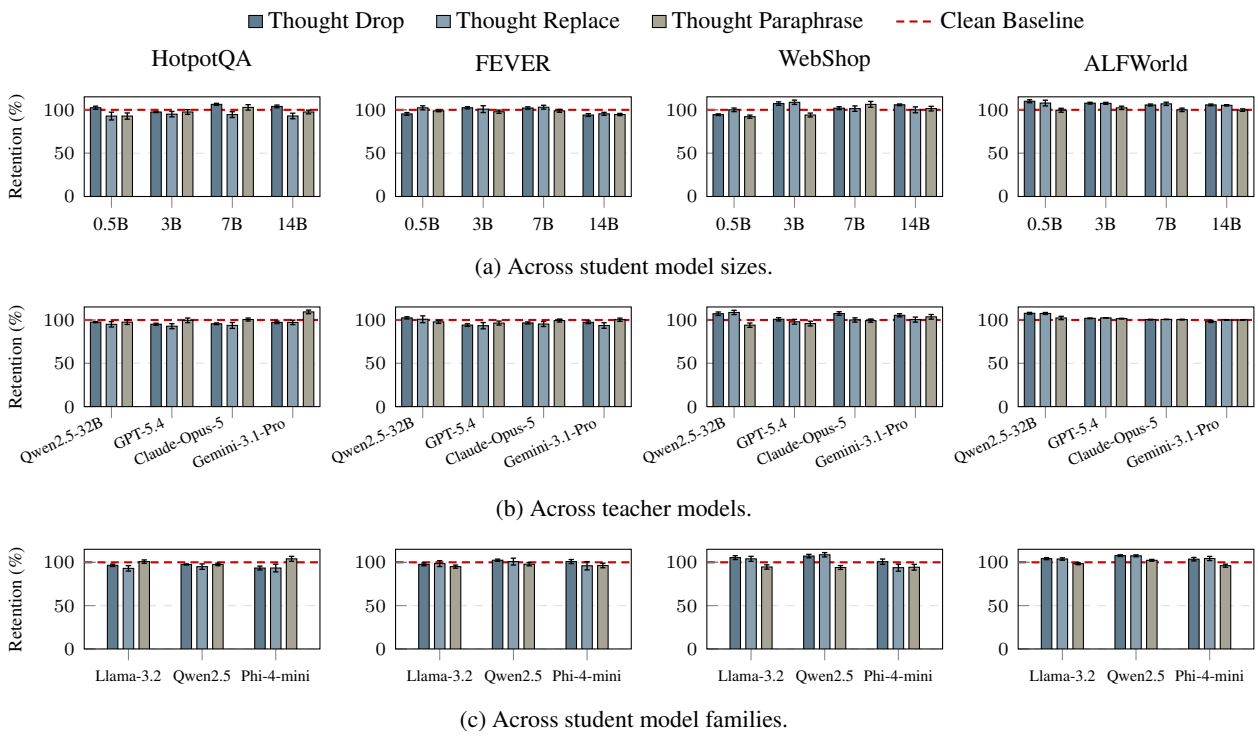
Implementation Details. We fine-tune all student models for two epochs using LoRA with rank 64, scaling factor 128, dropout 0.05, and all linear layers as target modules. We use AdamW with a learning rate of 1×10^{-4}, weight decay of 0.01, and an effective batch size of 8 trajectories. All experiments are conducted on a single NVIDIA H100 NVL GPU (96GB). Further details on training, inference, and evaluation are provided in Appendix B.

4 Thought Supervision Is Often Not Critical on Common Agent Distillations

In this section, we examine how critical thought supervision is in agent distillation for downstream action performance across four benchmarks adopted from the original ReAct evaluation (Yao et al., 2023): HotpotQA (Yang et al., 2018) for multi-hop question answering, FEVER (Thorne et al., 2018) for fact verification, WebShop (Yao et al., 2022) for web navigation, and ALFWorld (Shridhar et al., 2021) for embodied decision-making. We directly compare the effect of corrupting the thought and action channels, and then test whether the resulting trends are consistent across model scales, model families, teacher models, corruption granularities, and distillation objectives. Detailed dataset statistics are provided in Table 2 in Appendix A. Full numerical results for all experiments in Sections 4.1–4.7 are tabulated in Appendix D.

4.1 Actions Matter More Than Thoughts

Figure 2 reports retention when the thought and action channels are independently corrupted with drop and replace at corruption ratio p=1.0. Action corruption causes substantial degradation: action drop retains 53.1% of clean performance on HotpotQA and 79.5% on FEVER, while action replacement retains only 10.4% and 46.2%, respectively.



(a) Across student model sizes.

(b) Across teacher models.

(c) Across student model families.

Figure 3: Retention under thought corruption across four benchmarks at $p = 1.0$. (a) Results across student model sizes. (b) Results across teacher models. (c) Results across student model families.

On FEVER, action replacement reduces task accuracy to 29.3% (Table 3, Appendix D), which is slightly below the chance level of 33.3% for a three-choice question. On WebShop and ALFWorld, both action interventions yield 0.0% retention. In contrast, thought drop and replacement preserve near-clean performance across all benchmarks, reaching 108.7% on WebShop. We observe a clear distinction between the two channels: action corruption disrupts direct supervision of the observation-to-action mapping, whereas thought corruption leaves this mapping intact. In particular, the robustness to thought replacement suggests that preserving the original thought content is not always necessary for transferring the demonstrated agent behavior. We also report various corruption ratios p for both thought and action interventions in Figure 17 (Appendix D). This discrepancy suggests that action supervision in agent distillation is critical, while the thought channel is largely redundant.

4.2 Robustness Across Student Scales

Figure 3a validates the insight across Qwen2.5-Instruct (Yang et al., 2024) student scales of 0.5B, 3B, 7B, and 14B on all four benchmarks, reporting retention under thought replacement, removal, and paraphrase at p=1.0. Across all benchmarks, corruption operators, and student sizes, retention remains close to or above clean performance, ranging from 92.1% to 109.9%. This suggests that the limited utility of thought supervision is not a consequence of insufficient student model capacity.

4.3 Robustness Across Teacher Models

Figure 3b compares the default teacher, Qwen2.5-32B-Instruct (Yang et al., 2024), against three frontier teachers—GPT-5.4 (OpenAI, 2026), Claude-Opus-5 (Anthropic, 2026), and Gemini-3.1-Pro (Google, 2026)—reporting retention under thought interventions at p=1.0. Retention remains between 92.8% and 109.3% of clean performance regardless of the teacher used to generate the trajectories. These results show that robustness to thought interventions persists across diverse teacher models.

4.4 Robustness Across Student Model Families

Figure 3c distills the same Qwen2.5-32B-Instruct trajectories into three 3B-scale students—Llama-3.2-3B-Instruct (Meta AI, 2024), Qwen2.5-3B-Instruct (Yang et al., 2024), and Phi-4-mini-instruct (Abouelenin et al., 2025).

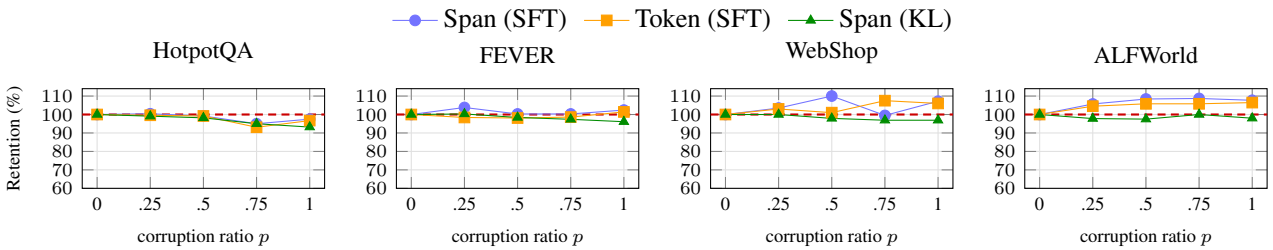


Figure 4: Retention under thought drop across corruption ratios p, comparing span-level and token-level corruption under SFT and span-level corruption under the KL objective (Qwen2.5-3B-Instruct student).

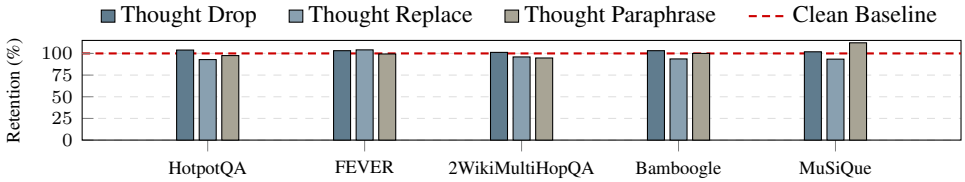


Figure 5: Out-of-distribution robustness under thought interventions (p=1.0), with HotpotQA as in-distribution.

Retention ranges from 93.0% to 108.7% of clean performance, and thought drop slightly exceeds clean performance in the WebShop and ALFWorld settings. This suggests that the redundancy of thought supervision is not tied to a particular student model family.

4.5 Robustness to Token-level Corruption

The preceding experiments corrupt each thought as a whole. We further test whether the observed robustness depends on corruption granularity by independently dropping each token within every thought with probability $p \in \{0.25, 0.5, 0.75, 1.0\}$. Figure 4 shows that token- and span-level thought-drop produce similar retention patterns across all four benchmarks. Under token-level corruption, retention ranges from 93.1% to 107.5% of clean performance and averages 101.6%, compared with 103.1% under span-level corruption. These results show that robustness to thought corruption persists across corruption granularities. Additional results for token-level thought replacement and paraphrase are reported in Table 6 (Appendix D).

4.6 Robustness Across Distillation Objectives

To test whether our findings are specific to the SFT objective, we repeat the span-level thought-drop sweep under the KL-based distillation objective $\mathcal{L}_{\text{KL}}$ (Eq. 9, Appendix B). Figure 4 shows that, at p=1.0, retention under KL-based distillation ranges from 93.2% to 98.0% across the four benchmarks, averaging 96.0%, compared with an average of 103.7% under SFT. These results suggest that the limited benefit of thought supervision is not specific to a particular distillation objective.

4.7 Out-of-Distribution Robustness

We evaluate the out-of-distribution robustness of thought interventions by distilling a Qwen2.5-14B-Instruct student on HotpotQA (Yang et al., 2018) and evaluating it on FEVER (Thorne et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), Bamboogle (Press et al., 2023), and MuSiQue (Trivedi et al., 2022) without target-dataset fine-tuning. As shown in Figure 5, thought interventions preserve performance across all out-of-distribution evaluation sets, retaining 93.4–112.3% of clean performance. These results show that robustness to thought interventions largely extends to cross-dataset transfer.

5 Thought Supervision Matters When Thought Is Not Disclosed in Action

5.1 High-disclosure vs. Low-disclosure Settings

In Section 4, we found that corrupting or even removing thoughts leaves performance largely unchanged, suggesting that thought supervision is often redundant for agents' action performance. Across all benchmarks above, we observed

(a) THOUGHT is disclosed in ACTION (FEVER)

QUESTION Claim: "Winter's Tale is a 1983 novel." Verify using evidence from Wikipedia.

THOUGHT I will use the tool `web_search` to get the information about Winter's Tale and its publication year.

ACTION
```
evidence = web_search(query="Winter's Tale publication year")
print(evidence)
```

(b) THOUGHT is *not* disclosed in ACTION (MathQA)

QUESTION A shop owner sells 15 mtr of cloth and gains sp of 10 mtrs. Find the gain %?

THOUGHT To find the gain percentage, I need to calculate the profit made on the cost price, which is the difference between the selling price of 15 meters and the gain of 10 meters.

ACTION
```
step0 = 15 - 10
step1 = 10 / step0
print(step1 * 100.0)
```

Figure 6: Examples illustrating whether task-relevant information of the thought is disclosed in the action during training. (a) Parts of thought information are disclosed in the action; (b) Thought is not disclosed in the action.

that information about the preceding thought is exposed in the action channel. In HotpotQA and FEVER, the action is a call to a web search tool, and the search query directly contains the entity or relation identified in the thought. In ALFWorld and WebShop, the action is a natural language command whose object names and attributes are constrained by the environment's valid action space, so the action accordingly contains this information. That is, the information introduced by the thought is disclosed verbatim or near-verbatim in the resulting action. We refer to such settings as *high-disclosure*. Such disclosure makes thought supervision not very useful because much of the task-relevant information it provides is already conveyed by the action.

Disclosure is much lower when actions are the symbolic programs provided by FinQA (Chen et al., 2021b), MultiHiertt (Zhao et al., 2022), MathQA (Amini et al., 2019), and GeoQA (Chen et al., 2021a). These programs consist of short operations over anonymous intermediate variables; we convert them into a unified step-wise executable format while preserving their original computation and anonymous references. The resulting action specifies how to compute the answer, but not what each operand represents or why that computation is appropriate. That information appears only in the accompanying thought, generated by the Qwen2.5-32B-Instruct teacher, leaving room for thought supervision to provide additional benefits beyond the actions. We refer to these as *low-disclosure* settings. Figure 6 shows contrasting examples from the two different settings. In Figure 6 (a), taken from FEVER, parts of information introduced in the thought are repeated almost verbatim in the action's search query, creating the disclosure described above. In Figure 6 (b), taken from MathQA, the action performs the arithmetic operations based on the thought but does not reveal what each quantity means or why those quantities should be combined in that way.

5.2 Removing Thought in Low-disclosure Settings

To examine whether thought supervision becomes important when thoughts are not disclosed in the action, we ablate thoughts from the teacher trajectories during distillation. Figure 7 shows a clear contrast between the two regimes: thought drop substantially impairs agent performance in low-disclosure settings, whereas performance remains largely preserved on high-disclosure settings across all model scales. For the 14B student in low-disclosure settings, retention drops to 77.3% on FinQA, 83.9% on MultiHiertt, 32.8% on MathQA, and 54.0% on GeoQA. In contrast to the high-disclosure settings, where thought removal largely preserves or even improves performance, low-disclosure settings consistently benefit from thought supervision in which task-relevant information of the thought is not exposed in the action. This pattern provides correlational evidence that thought supervision becomes more important as less thought information is disclosed in the action. We further control for inference-time thought generation and find that full-trajectory-trained students continue to outperform action-only students even when thoughts are suppressed at inference, showing that this benefit cannot be explained merely by differences in inference-time reasoning (Appendix F). Full results are provided in Table 9 (Appendix D).

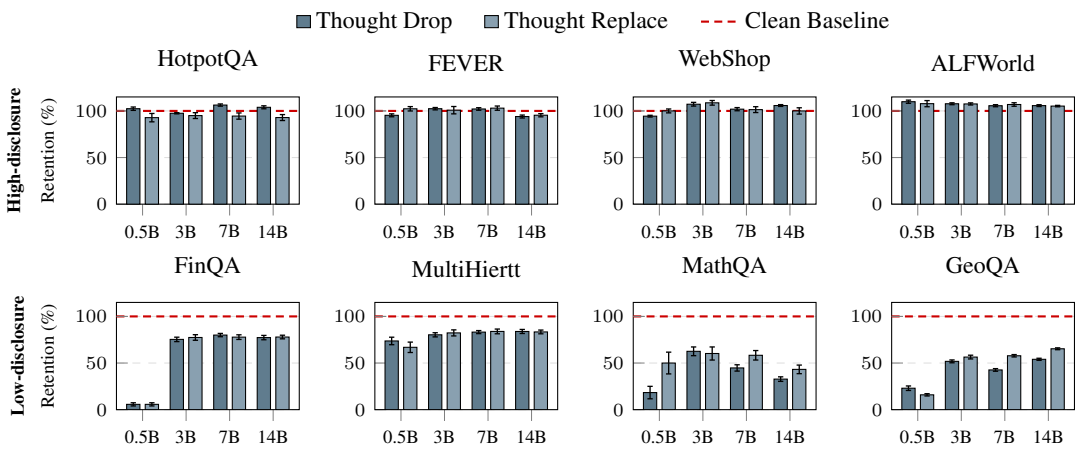


Figure 7: Retention under thought drop and replace across student sizes for high- and low-disclosure settings (p=1.0).

After artificially disclosing—THOUGHT is disclosed in ACTION (MathQA)

QUESTION A shop owner sells 15 mtr of cloth and gains sp of 10 mtrs. Find the gain %?

THOUGHT To find the gain percentage, I need to calculate the profit made on the cost price, which is the difference between the selling price of 15 meters and the gain of 10 meters.

ACTION
```
cost_price = 15 - 10
gain_over_cost_price = 10 / cost_price
print(gain_over_cost_price * 100.0)
```

Figure 8: Example of the disclosure intervention applied to the MathQA instance in Figure 6(b). Anonymous variables in the action are renamed with semantic names, exposing information that was previously available only in the thought.

5.3 Ablation: Disclosing Thought in Action

To test whether increasing the amount of thought information disclosed in the action affects the effect of thought supervision, we directly intervene in the action format of the low-disclosure settings by increasing the amount of thought information disclosed in the action while preserving the underlying computation. In low-disclosure settings, the actions contain anonymous intermediate references that specify how to compute the answer but reveal little about what each quantity represents. We therefore create a more disclosed variant by replacing these anonymous variables with semantic names that describe the quantities they hold. The numeric values, operators, execution order, and final outputs are kept unchanged; only the degree to which the action reveals the reasoning-relevant content is increased. Figure 8 shows an example from MathQA, before and after the disclosure intervention. In the original trace, the thought identifies `cost_price` as the quantity that `15 - 10` denotes, but the action exposes none of these: `step0` is semantically empty, so the mapping from the question's entities to the computed quantities is recoverable only from the thought. After renaming, that mapping is carried by the identifiers themselves, and the action explicitly discloses content that previously existed only in the thought.

Figure 9 shows that exposing more thought information in the action reduces the student's dependence on the thought channel during distillation. Before the intervention, retention ranges from 51.8–80.3%, well below 97.5–107.7% observed on high-disclosure settings. After the intervention, retention consistently improves on all four benchmarks, rising to 84.4–94.6%. Importantly, this improvement is not an artifact of a reduced clean baseline: as shown in Table 1, clean accuracy actually increases slightly across all four benchmarks, while thought-drop accuracy increases substantially more. Once task-relevant semantics are partially revealed in the action itself, removing the thought becomes substantially less harmful. Because the underlying computation remains unchanged, this improvement reflects a change in the information available through action supervision. Taken together with the earlier comparisons across settings, this supports our central claim: thought supervision becomes less useful as more thought information is disclosed in the action. To verify that this gain stems from semantic correspondence of the identifiers rather than from the mere presence of thought-derived words, we also evaluate names that are derived from the thoughts but do not describe the quantities they represent (Appendix E.3).

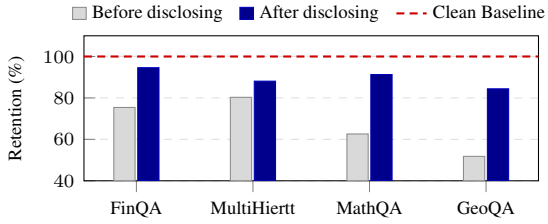


Figure 9: Thought-drop retention before and after artificially disclosing thought information in low-disclosure settings (Qwen2.5-3B-Instruct student).

Table 1: Accuracy before and after artificially disclosing thought information in low-disclosure settings (p=1.0). (The same data points in Figure 9)

	Before disclosing		After disclosing	
Benchmark	Clean	Drop	Clean	Drop
FinQA	$63.3_{(0.7)}$	$47.7_{(1.5)}$	$65.3_{(0.7)}$	$61.8_{(1.2)}$
MultiHiertt	$49.2_{(0.6)}$	$39.5_{(1.1)}$	$50.6_{(0.6)}$	$44.6_{(1.3)}$
MathQA	$17.1_{(0.9)}$	$10.7_{(0.8)}$	$18.3_{(0.8)}$	$16.7_{(0.9)}$
GeoQA	$38.8_{(0.6)}$	$20.1_{(0.5)}$	$41.0_{(0.8)}$	$34.6_{(0.8)}$

6 Bypassing Thought Supervision for Efficient Agent Distillation

By recognizing and distinguishing between high-disclosure and low-disclosure settings, we find that thought supervision provides no clear benefit in high-disclosure settings and can often be omitted from them with little or no loss in performance. The distinction motivates an alternative approach to supervision for such settings: distilling action only in order to gain efficiency in both training and inference.

In the evaluated high-disclosure settings, omitting thoughts improves training and inference efficiency while largely preserving, and at times improving, task performance. Thought tokens account for 59.1–71.5% of assistant-generated tokens across four benchmarks and four student scales (Figure 10). Removing thought tokens shortens the training sequences, which reduces the computation required per example during training. Students trained without thoughts generate actions directly in response to the same evaluation prompts, so the choice of training supervision also affects inference efficiency. Relative to full-trajectory distillation, action-only distillation reduces training wall-clock time by 5.3–10.2% and inference wall-clock time by 11.1–41.7% across the four benchmarks (Figure 11). These savings show a practical benefit of omitting redundant thought supervision during both distillation and deployment, while largely maintaining model utility.

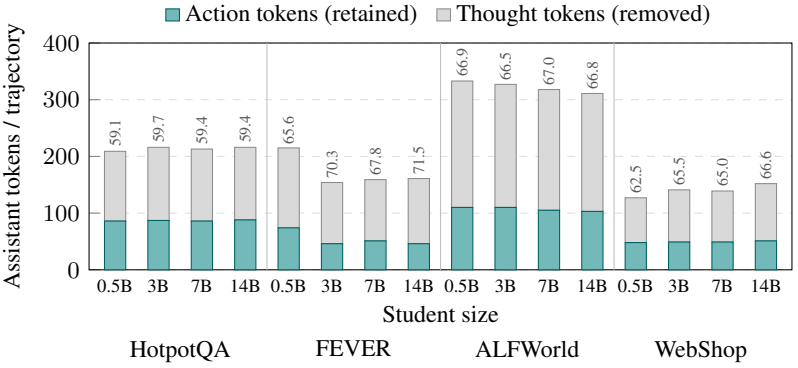


Figure 10: Average assistant-generated tokens per trajectory across high-disclosure settings and student scales, split into action and thought tokens. Numbers above bars show the thought-token share (%).

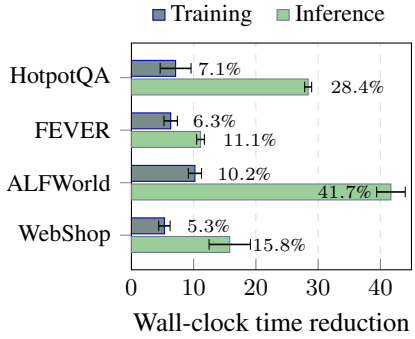


Figure 11: Wall-clock time reduction (%) from action-only distillation relative to full-trajectory distillation.

7 Conclusion

We revisit the role of thought supervision in LLM agent distillation by disentangling thought from action supervision within complete trajectories. Across benchmarks, model scales, teacher models, student families, and distillation objectives, we find that thoughts provide little additional benefit when their task-relevant information is already exposed in the teacher's actions. In contrast, removing thoughts becomes harmful when information conveyed by the reasoning is not disclosed in the action. These results collectively suggest that the value of thought supervision depends on what information is absent from the action channel. More broadly, our findings separate the role of reasoning in generating successful actions from its role as supervision for capability transfer. A teacher may need to reason extensively to produce an action, but the student may not need to faithfully imitate that reasoning if the information is already disclosed in the action. This motivates a selective principle for agent distillation: the benefit of thought supervision depends on how much of the task-relevant information in the thought is already exposed in the demonstrated action.

AI USE STATEMENT

In this work, we used generative AI for code generation, but the code was manually inspected and tested before use. The research questions, experimental decisions, and final scientific claims were determined by the authors. AI tools have ***not*** been used for any research ideation. The authors take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We provide the code used to reproduce our experiments in the supplementary material. Additional implementation and evaluation details, including compute resource specifications, are provided in the main paper and appendices.

REFERENCES

Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, et al. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *arXiv preprint arXiv:2503.01743*, 2025.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2357–2367, 2019.

Anthropic. Introducing Claude Opus 5. `https://www.anthropic.com/news/claude-opus-5`, July 2026.

Baian Chen, Chang Shu, Ehsan Shareghi, Nigel Collier, Karthik Narasimhan, and Shunyu Yao. Fireact: Toward language agent fine-tuning. *arXiv preprint arXiv:2310.05915*, 2023.

Jiaqi Chen, Jianheng Tang, Jinghui Qin, Xiaodan Liang, Lingbo Liu, Eric Xing, and Liang Lin. Geoqa: A geometric question answering benchmark towards multimodal numerical reasoning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 513–523, 2021a.

Jiaqi Chen, Tong Li, Jinghui Qin, Pan Lu, Liang Lin, Chongyu Chen, and Xiaodan Liang. Unigeo: Unifying geometry logical reasoning via reformulating mathematical expression. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pp. 3313–3323, 2022.

Xinghao Chen, Zhijing Sun, Guo Wenjin, Miaoran Zhang, Yanjun Chen, Yirong Sun, Hui Su, Yijie Pan, Dietrich Klakow, Wenjie Li, et al. Unveiling the key factors for distilling chain-of-thought reasoning. In *Findings of the association for computational linguistics: ACL 2025*, pp. 15094–15119, 2025.

Zehui Chen, Kuikun Liu, Qiuchen Wang, Wenwei Zhang, Jiangning Liu, Dahua Lin, Kai Chen, and Feng Zhao. Agent-flan: Designing data and methods of effective agent tuning for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 9354–9366, 2024.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan R Routledge, et al. Finqa: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3697–3711, 2021b.

Google. Gemini 3.1 Pro: A smarter model for your most complex tasks. `https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-1-pro/`, February 2026.

Maryam Hashemzadeh, Elias Stengel-Eskin, Sarath Chandar, and Marc-Alexandre Côté. Sub-goal distillation: A method to improve small language agents. In *Proceedings of The 3rd Conference on Lifelong Learning Agents*, volume 274 of *Proceedings of Machine Learning Research*, pp. 1053–1075. PMLR, 2025.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, 2020.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the association for computational linguistics: ACL 2023*, pp. 8003–8017, 2023.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Minki Kang, Jongwon Jeong, Seanie Lee, Jaewoong Cho, and Sung Ju Hwang. Distilling llm agent into small models with retrieval and code tools. *Advances in Neural Information Processing Systems*, 38:106501–106538, 2026.

Liunian Harold Li, Jack Hessel, Youngjae Yu, Xiang Ren, Kai-Wei Chang, and Yejin Choi. Symbolic chain-of-thought distillation: Small models can also "think" step-by-step. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2665–2679, 2023.

Jingyu Liu, Zhiwen Wang, Yuxin Jing, Huanyu Zhou, and Yong Liu. Where do cot training gains land in llm based agents? *arXiv preprint arXiv:2606.26935*, 2026a.

Jun Liu, Zhenglun Kong, Peiyan Dong, Changdi Yang, Tianqin Li, Yanyue Xie, Yifan Gong, Xuan Shen, Pu Zhao, Hao Tang, et al. Structured agent distillation for large language model agents. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems*, pp. 3676–3685, 2026b.

Yuanjie Lyu, Chengyu Wang, Jun Huang, and Tong Xu. From correction to mastery: Reinforced distillation of large language model agents. *arXiv preprint arXiv:2509.14257*, 2025.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. Teaching small language models to reason. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1773–1781, 2023.

Meta AI. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models. `https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/`, September 2024.

OpenAI. GPT-4o mini: Advancing cost-efficient intelligence, 2024. `https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence`.

OpenAI. Introducing GPT-5.4. `https://openai.com/index/introducing-gpt-5-4/`, March 2026.

Emiliano Penaloza, Dheeraj Vattikonda, Nicolas Gontier, Alexandre Lacoste, Laurent Charlin, and Massimo Caccia. Privileged information distillation for language models. In *Forty-third International Conference on Machine Learning*, 2026.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5687–5711, 2023.

Jiahao Qiu, Xinzhe Juan, Yimin Wang, Ling Yang, Xuan Qi, Tongcheng Zhang, Jiacheng Guo, Yifu Lu, Zixin Yao, Hongru Wang, et al. Agentdistill: Training-free agent distillation with generalizable mcp boxes. *arXiv preprint arXiv:2506.14728*, 2025.

Aymeric Roucher, Albert Villanova del Moral, Thomas Wolf, Leandro von Werra, and Erik Kaunismäki. smolagents: A smol library to build great agentic systems, 2025. URL `https://github.com/huggingface/smolagents`.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Cote, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021.

Guokai Tang and Feng Zhao. Smartad: Capacity-aligned agent distillation for small language models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 27045–27057, 2026.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Fever: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 809–819, 2018.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

Somin Wadhwa, Silvio Amir, and Byron C Wallace. Investigating mysteries of cot-augmented distillation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 6071–6086, 2024.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. Executable code actions elicit better llm agents. *arXiv preprint arXiv:2402.01030*, 2024a.

Xingyao Wang, Zihan Wang, Jiateng Liu, Yangyi Chen, Lifan Yuan, Hao Peng, and Heng Ji. Mint: Evaluating llms in multi-turn interaction with tools and language feedback. In *International Conference on Learning Representations*, volume 2024, pp. 32593–32627, 2024b.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William 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*, pp. 2369–2380, 2018.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757, 2022.

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

Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. Agenttuning: Enabling generalized agent abilities for llms. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3053–3077, 2024.

Yilun Zhao, Yunxiang Li, Chenying Li, and Rui Zhang. Multihiertt: Numerical reasoning over multi hierarchical tabular and textual data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6588–6600, 2022.

A Datasets and Trajectory Construction

Table 2 summarizes the datasets used in our agent distillation experiments. We consider eight benchmarks covering multi-hop question answering (HotpotQA (Yang et al., 2018)), fact verification (FEVER (Thorne et al., 2018)), interactive tasks (ALFWorld (Shridhar et al., 2021) and WebShop (Yao et al., 2022)), and quantitative reasoning (FinQA (Chen et al., 2021b), MultiHiertt (Zhao et al., 2022), MathQA (Amini et al., 2019), and GeoQA (Chen et al., 2021a)). HotpotQA (Yang et al., 2018) is a knowledge-intensive multi-hop question answering benchmark requiring iterative retrieval and reasoning over Wikipedia. FEVER (Thorne et al., 2018) is fact verification, where the agent retrieves evidence and classifies a claim as supported, refuted, or unverifiable. WebShop (Yao et al., 2022) is for web navigation, where the agent searches, compares, and purchases a product matching a natural language instruction. For embodied decision-making, ALFWorld (Shridhar et al., 2021) is a text-based household environment requiring long-horizon planning.

For quantitative reasoning, FinQA (Chen et al., 2021b) is a financial question answering benchmark built on earnings reports, where the agent must extract relevant values from both text and tables and perform multi-step numerical computation. MultiHiertt (Zhao et al., 2022) extends this setting to documents with multiple hierarchical tables and long unstructured text, requiring the agent to locate evidence across several tables before carrying out numerical reasoning. MathQA (Amini et al., 2019) is a large-scale collection of multiple-choice math word problems annotated with operation programs, testing the agent's ability to translate natural language problems into correct sequences of arithmetic operations. GeoQA (Chen et al., 2021a) is a geometry problem-solving benchmark collected from middle school exams, where the agent must reason about geometric relations such as angles, lengths, and areas to compute the answer; we use its English version released with UniGeo (Chen et al., 2022) and train and evaluate on a text-only subset of problems solvable without the diagram.

Unless otherwise specified, teacher trajectories are generated using Qwen2.5-32B-Instruct (Yang et al., 2024) in the ReAct (Yao et al., 2023) format. For the quantitative reasoning benchmarks, actions are instead constructed from dataset-provided symbolic programs, as described in Section 5.1. Only trajectories that satisfy the benchmark-specific success criterion are eligible for distillation; the table reports the final training set sizes after selection. During training, thoughts and actions serve as supervised targets, while observations provide context and are masked out from the loss. For each benchmark, the clean baseline and all corruption conditions use the same underlying training trajectories and evaluation set.

For the out-of-distribution evaluation in Section 4.7, we additionally consider 2WikiMultiHopQA (Ho et al., 2020), Bamboogle (Press et al., 2023), and MuSiQue (Trivedi et al., 2022). These benchmarks are used only for evaluation, without target-dataset training or fine-tuning.

For each benchmark, *Teacher Trajectories* reports the size of the teacher trajectory pool, *Training Data* reports the number of trajectories used for student training using Qwen2.5-32B-Instruct, and *Eval Set Size* reports the number of evaluation instances. For HotpotQA, FEVER, 2WikiMultiHopQA, and MuSiQue, we use at most 500 evaluation instances, following the evaluation protocol of prior agent distillation work (Kang et al., 2026; Wang et al., 2024b). For ALFWorld, we evaluate on the full validation set of 274 games, comprising 140 `valid_seen` and 134 `valid_unseen` games.

Table 2: Dataset statistics for agent distillation experiments.

Benchmark	Teacher Trajectories	Training Data	Eval Set Size
HotpotQA	1,000	597	500
FEVER	1,000	650	500
ALFWorld	3,553	2,872	274
WebShop	10,587	3,555	500
FinQA	5,695	1,500	300
MultiHiertt	6,217	1,500	832
MathQA	17,167	1,500	1,000
GeoQA	2,362	1,500	500
2WikiMultiHopQA	–	–	500
Bamboogle	–	–	125
MuSiQue	–	–	500

B Training, Inference, and Evaluation Details

B.1 Training

KL Loss Function. We evaluate KL-based distillation as an alternative to the SFT objective. For each training trajectory, we minimize the token-level forward KL divergence between the teacher and student output distributions:

$$\mathcal{L}_{\mathrm{KL}}(\theta) = \mathbb{E}_{\tau\sim\mathcal{D}}\left[\sum_{j\in\mathcal{S}(\tau)} \mathrm{KL}(\pi_{\text{teacher}}(\cdot \mid y_{<j}) \,\|\, \pi_\theta(\cdot \mid y_{<j}))\right] \tag{9}$$

Both distributions are conditioned on the same trajectory prefix $y_{<j}$. The set $S(\tau)$ contains the supervised token positions under the corresponding intervention, following the same masking convention as the SFT objective.

Retrieval Environment. For reproducibility of experiments, we use a fixed 2018 Wikipedia corpus as the retrieval knowledge base. Following Search-R1 (Jin et al., 2025), we encode both documents and search queries using `e5-base-v2` (Wang et al., 2022) and retrieve the top-3 passages using an exact flat index. The resulting retriever is exposed to the agent as a `web_search` tool. The agent is implemented following the CodeAct (Wang et al., 2024a) framework in `smolagents` (Roucher et al., 2025). We use the same Wikipedia corpus and retriever across all retrieval-based conditions to ensure a consistent external knowledge source.

B.2 Inference

Inference Prompt. The inference prompt presented to the student is assembled from a system prompt specifying the agent interaction format, a user prompt provided by the agent framework or benchmark, and the benchmark-specific task input (Figures 12, 13, and 14). For HotpotQA, FEVER, FinQA, MultiHiertt, MathQA, and GeoQA, we use a shared ReAct-style system prompt that defines the Thought–Code–Observation interaction cycle and available tools. ALFWorld and WebShop instead use benchmark-specific interaction prompts that define their respective action spaces and include fixed demonstration examples. For MathQA, we omit the multiple-choice options and evaluate it in an open-ended setting. Within each benchmark, the prompt template is held fixed across all intervention conditions, and only the task instance varies.

Decoding. For ALFWorld and WebShop, each decoding call generates a single-line command with a 100-token generation limit. Episodes are capped at 49 and 15 environment-action steps, respectively. The `think:` and `think[...]` commands do not count toward these interaction budgets and do not consume an environment-action step. Thus, students trained with and without thoughts are allowed the same maximum number of environment-interacting actions. For all other benchmarks, each agent step jointly generates the thought and code with a limit of 1,024 new tokens, and episodes are capped at five steps. The fallback that forces a terminal answer when the interaction budget is exhausted is disabled, so any episode that exceeds the budget is scored as a failure.

B.3 Evaluation

Success Criteria. An LLM judge is used for HotpotQA, 2WikiMultiHopQA, Bamboogle, and MuSiQue. We use `GPT-4o-mini` with structured decoding that constrains the output to a JSON object containing a free-text explanation and a binary score; Figure 15 gives the prompt verbatim. FEVER predictions are one of three labels and are scored by matching the normalized prediction against the gold label. FinQA, MultiHiertt, MathQA, and GeoQA are scored by symbolic and numeric equivalence on the extracted final answer, so that equivalent surface forms are treated identically. WebShop and ALFWorld use the benchmark-provided success signal.

C Corruption Operator Implementation

Paraphrase. The `Paraphrase` operator is instantiated with Qwen2.5-32B-Instruct, the same model used as our default teacher. Each selected thought is rewritten independently: the paraphraser receives the body of a single thought t_i, and its output is reinserted in place of t_i, so that only the surface form of the thought changes. The prompt (Figure 16) instructs the model to diverge from the original in wording and sentence structure as much as possible while holding every fact, entity, number, plan step, and conclusion fixed, and to preserve approximate length. We decode with temperature 0.8, top-p 0.95, a 512-token cap, and a fixed seed (0). Paraphrased thoughts remain supervised targets and are not masked from the loss.

````
=== System prompt ===
You are an expert assistant who can solve any task using code blobs. You will be given a task to solve as best
you can.
To do so, you have been given access to a list of tools: these tools are basically Python functions which you
can call with code.
To solve the task, you must plan forward to proceed in a series of steps, in a cycle of 'Thought:', 'Code:',
and 'Observation:' sequences.

At each step, in the 'Thought:' sequence, you should first explain your reasoning towards solving the task and
the tools that you want to use.
Then in the 'Code:' sequence, you should write the code in simple Python. The code sequence must start with
'```py' and end with '```<end_code>', else you will fail. For example:
Code:
```py
result = 2 + 2
final_answer(result)
```<end_code>
During each intermediate step, you can use 'print()' to save whatever important information you will then need.
These print outputs will then appear in the 'Observation:' field, which will be available as input for the next
step.
In the end you have to return a final answer using the `final_answer` tool.

On top of performing computations in the Python code snippets that you create, you only have access to these
tools:
- web_search: Uses semantic search to retrieve the parts of 2018 wikipedia that could be most relevant to
answer your query.
    Takes inputs: {'query': {'type': 'string', 'description': '...'}}
    Returns an output of type: string

Now Begin! If you solve the task correctly, you will receive a reward of $1,000,000.

=== User prompt ===
New task:
{task}

=== Benchmark-specific task input ===
HotpotQA / FEVER   : {question or claim}
FinQA / MultiHiertt : {question} + "\n\nContext:\n" + {pre_text, table, post_text}
MathQA / GeoQA : {question}
````

Figure 12: Assembled inference prompt for HotpotQA, FEVER, FinQA, MultiHiertt, MathQA, and GeoQA.

Replace. For `Replace`, each selected thought is independently substituted with a thought sampled uniformly at random from the pool of thoughts of other training instances of the same benchmark.

Drop. For `Drop`, the selected thought is removed from the trajectory entirely, together with its span-level formatting markers, rather than being replaced by an empty placeholder or masked from the loss.

D Full Intervention Results

This appendix provides the full numerical results underlying the analyses in the main text, together with detailed corruption-ratio sweeps. All accuracy and success-rate values are reported as percentages.

Thought and action corruption across student scales (Sections 4.1–4.2). Table 3 reports the results of thought and action interventions across four Qwen2.5 student scales: 0.5B, 3B, 7B, and 14B. The 3B results support the comparison between thought and action corruption in Section 4.1, while the thought-intervention results across all four scales support Section 4.2. Each intervention is compared with the clean baseline for the same student model.

Thought and action corruption across corruption ratios (Section 4.1). Figure 17 reports retention under span-level corruption of thoughts (top row) and actions (bottom row) across corruption ratios $p \in \{0.25, 0.5, 0.75, 1.0\}$. Thought retention stays near the clean baseline at every ratio and for all three operators, whereas action retention decays with p and collapses to 0% on the interactive benchmarks at $p = 1.0$. These results complement the full-corruption comparison by including settings in which only a fraction of spans is corrupted.

Robustness across teacher models (Section 4.3). Table 4 reports the full results under thought drop, replacement, and paraphrase for trajectories generated by Qwen2.5-32B-Instruct, GPT-5.4, Claude-Opus-5, and Gemini-3.1-Pro.

```
=== System prompt ===
You are an agent interacting with the ALFWorld household environment.
At each turn, output exactly one command and nothing else. Do not add a prefix such as "Action:" or
"Assistant:", and do not explain an environment command outside a `think:` command.

Valid command forms are:
- think: <brief reasoning about the next action>
- go to <receptacle>
- open <receptacle>
- close <receptacle>
- take <object> from <receptacle>
- put <object> in/on <receptacle>
- use <object>
- heat <object> with <receptacle>
- clean <object> with <receptacle>
- cool <object> with <receptacle>
- inventory
- look
- examine <object or receptacle>

Use the exact object and receptacle names and numeric IDs shown by the environment. The `think:` command does
not change the environment. Never invent observations; wait for the environment response after each command.

=== User prompt ===
Interact with a household to solve a task. Here are two examples.
{Example 1} + {Example 2}

Here is the task.
{room description} + "\nYour task is to: " + {task instruction}
```

Figure 13: Assembled inference prompt for ALFWorld.

```
=== System prompt ===
You are an agent shopping on the WebShop website to fulfill a customer instruction. At each turn, output
exactly one command and nothing else. Do not add a prefix such as "Action:" or "Assistant:", and do not explain
a website command outside a `think[...]` command.

Valid command forms are:
- think[<brief reasoning about the next action>]
- search[<query>]
- click[<exact text of a button, product code, or option shown on the page>]

`search[...]` is only available on the search page. `click[...]` must use the exact text shown in brackets on
the current page. The `think[...]` command does not change the page. Buying with click[Buy Now] ends the
episode, so select the required options first. Never invent observations; wait for the page after each command.

=== User prompt ===
{Example 1}

Here is the task.
{webshop_instruction}
{initial_search_page}
```

Figure 14: Assembled inference prompt for WebShop.

The student is fixed to Qwen2.5-3B-Instruct, and each intervention is compared with the clean baseline obtained using the same teacher.

Robustness across student model families (Section 4.4). Table 5 reports the full results under thought interventions for Llama-3.2-3B-Instruct, Qwen2.5-3B-Instruct, and Phi-4-mini-instruct. All students are distilled from the same Qwen2.5-32B-Instruct teacher trajectories, with a separate clean baseline for each student model.

Robustness to corruption granularity (Section 4.5). Table 6 reports the full corruption-ratio sweeps for thought drop, replacement, and paraphrase at both span and token levels. At the token level, each thought token is independently selected with probability p. For replacement, selected tokens are independently sampled from a deduplicated token pool constructed from thoughts in other trajectories. For paraphrase, we instead prompt the Qwen2.5-32B-Instruct teacher with the surrounding thought context to generate a context-appropriate synonym for each selected token, preserving the local meaning while altering its lexical realization. For drop, we remove the selected tokens.

```
Compare the following predicted answer with the gold (correct) answer and determine if they match in meaning
given the question.
Be reasonable - phrasing differences are okay if the core meaning is the same. You should accept the aliases
and the answer conveying the same conclusion.
If the predicted answer is overly verbose and fails to capture the key information found in the gold answer
(i.e., low recall), consider it a false answer.

Question: {question}
Predicted answer: {predicted}
Gold answer: {gold}

Output your evaluation as a JSON object with two fields:
- explanation: Brief explanation of your scoring decision
- score: 0 or 1 indicating if answers match
```

Figure 15: Judge prompt used for HotpotQA, 2WikiMultiHopQA, Bamboogle, and MuSiQue.

```
=== System prompt ===
You rewrite one step of an AI agent's reasoning. Rewrite the user's text so that the wording and sentence
structure are as different as possible while the meaning stays exactly the same: keep every fact, entity,
number, plan step, and conclusion; do not add, remove, or correct anything. Keep roughly the same length.
Output only the rewritten text, with no prefix, no quotes, and no commentary.

=== User prompt ===
{thought}
```

Figure 16: Prompt used for the `Paraphrase` intervention (Eq. 4). The placeholder {`thought`} is replaced by the natural-language body of a single teacher thought t_i.

Robustness across distillation objectives (Section 4.6). Table 7 compares SFT and KL-based distillation under span-level thought drop at matched corruption ratios. For each objective, the result at $p = 0$ serves as the clean baseline for assessing the effect of thought removal.

Robustness to out-of-distribution datasets (Section 4.7). Table 8 reports the full results for a Qwen2.5-14B-Instruct student distilled on HotpotQA under thought drop, replacement, and paraphrase at $p = 1.0$. Evaluation includes HotpotQA as the in-distribution benchmark and FEVER, 2WikiMultiHopQA, Bamboogle, and MuSiQue as out-of-distribution benchmarks, without target-dataset fine-tuning. Each intervention is compared with the clean model on the same evaluation dataset.

Thought interventions in low-disclosure settings (Section 5.2). Table 9 reports accuracy under clean training, thought drop, and thought replacement on FinQA, MultiHiertt, MathQA, and GeoQA across four student scales. Both interventions reduce accuracy in every reported benchmark–scale combination, with larger students also remaining sensitive to the removal or replacement of thoughts.

Training and inference efficiency (Section 6). Table 10 reports the absolute wall-clock training and inference times for full-trajectory and action-only distillation across the high-disclosure settings. Time reductions are computed relative to full-trajectory distillation for each benchmark.

E Details of the Disclosure Intervention

E.1 Thought Generation for Low-disclosure Trajectories

For the low-disclosure settings, the accompanying thoughts are generated with Qwen2.5-32B-Instruct after constructing and verifying the corresponding executable action. The model is provided with the task input and the verified step-wise computation underlying the action, including its intermediate values and final result. As shown in Figure 18, it is instructed to narrate this computation concisely in the first person, as if working through the problem for the first time, without solving it anew.

Table 3: Task accuracy under thought and action corruption across Qwen2.5-{0.5B, 3B, 7B, 14B}-Instruct students distilled from Qwen2.5-32B-Instruct (p=1.0). Results are averaged over three independent training and evaluation runs, with standard deviations reported in parentheses.

			Thought			Action	
Benchmark	Size	Clean	Drop	Replace	Paraphrase	Drop	Replace
HotpotQA	0.5B	33.4$_{(0.9)}$	34.2$_{(0.6)}$	31.0$_{(1.5)}$	31.0$_{(1.2)}$	7.2$_{(1.6)}$	10.0$_{(0.5)}$
	3B	48.0$_{(0.7)}$	46.8$_{(0.3)}$	45.6$_{(1.5)}$	46.8$_{(1.3)}$	25.5$_{(1.4)}$	5.0$_{(1.0)}$
	7B	50.8$_{(1.1)}$	54.0$_{(0.6)}$	48.0$_{(1.7)}$	52.2$_{(1.6)}$	29.0$_{(1.8)}$	4.2$_{(2.1)}$
	14B	56.4$_{(1.2)}$	58.6$_{(0.9)}$	52.4$_{(1.8)}$	55.0$_{(1.2)}$	2.2$_{(1.3)}$	5.2$_{(1.7)}$
FEVER	0.5B	51.2$_{(0.8)}$	48.8$_{(0.8)}$	52.4$_{(1.2)}$	50.6$_{(0.5)}$	40.0$_{(1.2)}$	44.6$_{(1.2)}$
	3B	63.4$_{(1.2)}$	64.9$_{(0.8)}$	63.9$_{(2.5)}$	62.0$_{(1.2)}$	50.4$_{(1.1)}$	29.3$_{(6.9)}$
	7B	66.4$_{(1.1)}$	67.8$_{(0.9)}$	68.4$_{(1.5)}$	65.6$_{(1.1)}$	50.0$_{(1.0)}$	25.6$_{(4.5)}$
	14B	69.8$_{(1.1)}$	65.6$_{(1.1)}$	66.6$_{(1.2)}$	66.0$_{(0.9)}$	51.0$_{(1.2)}$	18.6$_{(3.5)}$
WebShop	0.5B	43.0$_{(0.5)}$	40.6$_{(0.4)}$	43.0$_{(0.9)}$	39.6$_{(0.8)}$	0.0$_{(0.0)}$	0.0$_{(0.0)}$
	3B	40.2$_{(0.9)}$	43.1$_{(0.8)}$	43.7$_{(1.0)}$	37.8$_{(0.9)}$	0.0$_{(0.0)}$	0.0$_{(0.0)}$
	7B	41.4$_{(0.9)}$	42.2$_{(0.7)}$	42.0$_{(1.3)}$	44.0$_{(1.4)}$	0.0$_{(0.0)}$	0.0$_{(0.0)}$
	14B	41.4$_{(0.5)}$	43.8$_{(0.4)}$	41.4$_{(1.4)}$	42.0$_{(1.1)}$	0.0$_{(0.0)}$	0.0$_{(0.0)}$
ALFWorld	0.5B	88.0$_{(2.3)}$	96.7$_{(1.5)}$	94.9$_{(2.9)}$	87.6$_{(2.0)}$	0.0$_{(0.0)}$	0.0$_{(0.0)}$
	3B	88.3$_{(1.0)}$	95.1$_{(0.9)}$	94.9$_{(1.0)}$	90.3$_{(1.7)}$	0.0$_{(0.0)}$	0.0$_{(0.0)}$
	7B	90.5$_{(1.1)}$	95.6$_{(1.0)}$	96.7$_{(1.7)}$	90.5$_{(1.8)}$	0.0$_{(0.0)}$	0.0$_{(0.0)}$
	14B	92.3$_{(1.0)}$	97.6$_{(0.9)}$	97.1$_{(0.7)}$	92.0$_{(1.3)}$	0.0$_{(0.0)}$	0.0$_{(0.0)}$

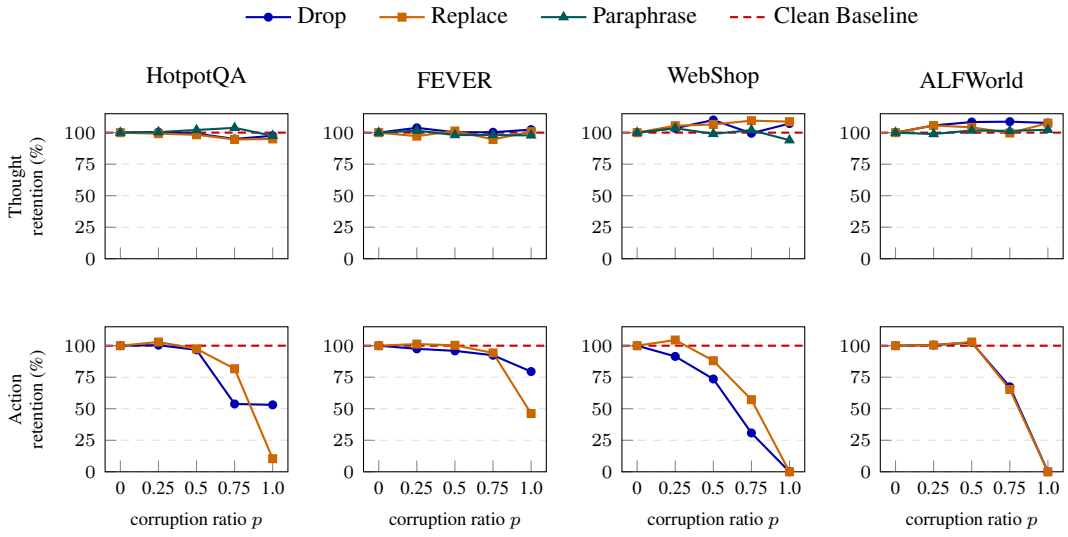


Figure 17: Retention under thought corruption (top) and action corruption (bottom) at varying corruption ratios p (Qwen2.5-3B-Instruct student). Corruption is applied at the span level: each thought t_i or action a_i is dropped, replaced, or paraphrased as a whole with probability p. Paraphrase is defined for thoughts only.

E.2 Semantic Naming Procedure

For the disclosure intervention in Section 5.3, we transform the executable actions of the low-disclosure settings while keeping the task input, thought, numerical computation, and evaluation instances fixed. Starting from the original action containing anonymous intermediate variables (e.g., `step0`, `step1`), we first execute the original code to obtain its output. We then prompt Qwen2.5-32B-Instruct to replace each anonymous variable with a descriptive, question-grounded identifier. The model is explicitly instructed not to change numerical constants, operators, or the order of operations. We decode with temperature 0.2 and a maximum of 320 tokens.

Table 4: Task accuracy under thought interventions across teacher models, using a Qwen2.5-3B-Instruct student (p=1.0).

Benchmark	Intervention	Qwen2.5-32B-Instruct	GPT-5.4	Claude-Opus-5	Gemini-3.1-Pro
HotpotQA	Clean	$48.0_{(0.7)}$	$52.6_{(0.9)}$	$41.0_{(0.6)}$	$42.8_{(0.9)}$
	Drop	$46.8_{(0.3)}$	$50.0_{(0.6)}$	$39.2_{(0.4)}$	$41.6_{(0.6)}$
	Replace	$45.6_{(1.5)}$	$48.8_{(1.5)}$	$38.4_{(1.4)}$	$41.6_{(1.1)}$
	Paraphrase	$46.8_{(1.3)}$	$52.4_{(1.4)}$	$41.2_{(0.7)}$	$46.8_{(0.9)}$
FEVER	Clean	$63.4_{(1.2)}$	$61.0_{(1.0)}$	$59.2_{(0.8)}$	$59.0_{(1.2)}$
	Drop	$64.9_{(0.8)}$	$57.4_{(0.9)}$	$57.2_{(0.7)}$	$57.4_{(0.9)}$
	Replace	$63.9_{(2.5)}$	$57.0_{(2.2)}$	$56.4_{(1.7)}$	$55.2_{(1.9)}$
	Paraphrase	$62.0_{(1.2)}$	$58.8_{(1.4)}$	$58.8_{(1.0)}$	$59.2_{(1.1)}$
WebShop	Clean	$40.2_{(0.9)}$	$47.8_{(1.0)}$	$48.6_{(1.0)}$	$44.6_{(1.0)}$
	Drop	$43.1_{(0.8)}$	$48.2_{(0.9)}$	$52.2_{(1.0)}$	$47.0_{(0.8)}$
	Replace	$43.7_{(1.0)}$	$46.8_{(1.3)}$	$48.6_{(1.2)}$	$44.8_{(1.3)}$
	Paraphrase	$37.8_{(0.9)}$	$45.8_{(1.3)}$	$48.2_{(0.9)}$	$46.2_{(1.2)}$
ALFWorld	Clean	$88.3_{(1.0)}$	$96.4_{(0.4)}$	$98.9_{(0.4)}$	$99.3_{(0.4)}$
	Drop	$95.1_{(0.9)}$	$98.2_{(0.4)}$	$99.3_{(0.4)}$	$97.6_{(1.2)}$
	Replace	$94.9_{(1.0)}$	$98.7_{(0.2)}$	$99.5_{(0.2)}$	$99.3_{(0.4)}$
	Paraphrase	$90.3_{(1.7)}$	$97.8_{(0.4)}$	$99.3_{(0.4)}$	$99.3_{(0.4)}$

Table 5: Task accuracy under thought interventions across student model families, using a fixed Qwen2.5-32B-Instruct teacher (p=1.0).

Benchmark	Intervention	Llama-3.2-3B-Instruct	Qwen2.5-3B-Instruct	Phi-4-mini-instruct
HotpotQA	Clean	$51.2_{(0.7)}$	$48.0_{(0.7)}$	$45.4_{(1.3)}$
	Drop	$49.4_{(0.7)}$	$46.8_{(0.3)}$	$42.4_{(1.0)}$
	Replace	$47.6_{(1.6)}$	$45.6_{(1.5)}$	$42.4_{(2.0)}$
	Paraphrase	$51.6_{(1.0)}$	$46.8_{(1.3)}$	$47.2_{(1.3)}$
FEVER	Clean	$67.6_{(0.8)}$	$63.4_{(1.2)}$	$64.8_{(1.0)}$
	Drop	$66.0_{(1.1)}$	$64.9_{(0.8)}$	$65.4_{(1.5)}$
	Replace	$66.6_{(2.3)}$	$63.9_{(2.5)}$	$62.2_{(3.1)}$
	Paraphrase	$64.2_{(1.2)}$	$62.0_{(1.2)}$	$62.4_{(1.7)}$
WebShop	Clean	$44.6_{(0.8)}$	$40.2_{(0.9)}$	$42.2_{(1.0)}$
	Drop	$47.0_{(1.0)}$	$43.1_{(0.8)}$	$42.5_{(1.3)}$
	Replace	$46.4_{(1.3)}$	$43.7_{(1.0)}$	$39.6_{(1.7)}$
	Paraphrase	$42.2_{(1.1)}$	$37.8_{(0.9)}$	$39.8_{(1.4)}$
ALFWorld	Clean	$94.2_{(0.7)}$	$88.3_{(1.0)}$	$91.6_{(1.0)}$
	Drop	$98.2_{(1.1)}$	$95.1_{(0.9)}$	$94.9_{(1.8)}$
	Replace	$97.8_{(1.3)}$	$94.9_{(1.0)}$	$95.6_{(2.2)}$
	Paraphrase	$92.7_{(1.1)}$	$90.3_{(1.7)}$	$88.0_{(1.6)}$

To verify the transformation, we re-execute each transformed action and confirm that its output matches that of the original program within a relative tolerance of 10^{-6}. We additionally require the rewritten code to include at least one assigned variable whose identifier no longer matches the anonymous `step\d+` pattern. For accepted examples, only the code block in the original trajectory is replaced; the question, context, and thought remain unchanged. The same evaluation instances are used for the original and disclosed conditions.

E.3 Mismatched-Name Control

The intervention in Section 5.3 introduces semantically informative identifiers into the action representation. To further examine the role of semantic correspondence, we construct a *mismatched-name* control in which each anonymous

Table 6: Task accuracy under span-level and token-level thought corruption at varying corruption ratios p (Qwen2.5-3B-Instruct student, Qwen2.5-32B-Instruct teacher). $p = 0$ corresponds to the unperturbed trajectories.

		Drop		Replace		Paraphrase	
Benchmark	p	Span	Token	Span	Token	Span	Token
HotpotQA	0 (clean)	48.0					
	0.25	48.2	47.8	47.6	48.4	48.2	47.0
	0.50	47.6	47.6	47.2	46.0	49.0	47.4
	0.75	45.6	44.7	45.4	45.0	49.8	50.6
	1.00	46.8	46.4	45.6	44.8	46.8	48.0
FEVER	0 (clean)	63.4					
	0.25	65.8	62.4	61.6	64.0	64.4	61.8
	0.50	63.6	62.2	64.2	63.2	62.2	64.0
	0.75	63.6	62.5	60.0	62.2	62.2	64.8
	1.00	64.9	64.3	63.9	61.8	62.0	65.2
WebShop	0 (clean)	40.2					
	0.25	41.6	41.4	42.4	40.4	41.6	39.0
	0.50	44.2	40.6	42.8	40.2	39.8	39.0
	0.75	40.0	43.2	44.0	45.0	41.0	39.8
	1.00	43.1	42.6	43.7	45.2	37.8	40.0
ALFWorld	0 (clean)	88.3					
	0.25	93.3	92.3	93.3	91.6	87.3	90.1
	0.50	95.7	93.4	92.0	92.7	89.8	93.8
	0.75	96.0	93.4	88.0	92.3	89.5	93.4
	1.00	95.1	94.0	94.9	95.3	90.3	89.8

Table 7: Accuracy under SFT and KL distillation objectives as a function of the span-level thought-drop ratio p (Qwen2.5-3B-Instruct student, Qwen2.5-32B-Instruct teacher). p=0 corresponds to the unperturbed trajectories.

Objective	p	HotpotQA	FEVER	WebShop	ALFWorld
SFT	0 (clean)	48.0	63.4	40.2	88.3
	0.25	48.2	65.8	41.6	93.3
	0.50	47.6	63.6	44.2	95.7
	0.75	45.6	63.6	40.0	96.0
	1.00	46.8	64.9	43.1	95.1
KL	0 (clean)	43.8	61.6	39.0	89.1
	0.25	43.4	61.8	39.0	87.1
	0.50	43.0	60.6	38.2	86.9
	0.75	41.6	60.0	37.8	89.1
	1.00	40.8	59.2	37.8	87.3

variable is renamed using content words from the corresponding thought that do not describe the quantity stored in that variable (e.g., naming $15 - 10$ as `percentage_of_meters` in Figure 8). This preserves thought-derived variable names while breaking their correspondence with the represented quantities. Numeric values, operators, execution order, and final outputs remain unchanged, so the underlying computation is identical across representations.

Table 11 shows that mismatched variable names do not reproduce the benefit of semantically informative identifiers. Across all four benchmarks, thought-drop retention under the mismatched condition remains below that of the semantic condition. Clean accuracy is consistently lower than in the anonymous condition, suggesting that misleading identifiers may interfere with learning from the demonstrated actions. Together, these results are consistent with semantic alignment between identifiers and their represented quantities contributing to the improved retention under the semantic condition.

Table 8: In-distribution and out-of-distribution performance of a Qwen2.5-14B-Instruct student distilled on HotpotQA under thought drop, replacement, and paraphrase (p=1.0).

Evaluation Dataset	Clean	Thought Drop	Thought Replace	Thought Paraphrase
HotpotQA (In-distribution)	56.4	58.6	52.4	55.0
FEVER (Out-of-distribution)	63.2	65.2	65.8	62.8
2WikiMultiHopQA (Out-of-distribution)	49.4	50.0	47.4	46.8
Bamboogle (Out-of-distribution)	50.4	52.0	47.2	50.4
MuSiQue (Out-of-distribution)	21.2	21.6	19.8	23.8

Table 9: Accuracy under clean training, thought drop, and thought replacement across Qwen2.5-{0.5B, 3B, 7B, 14B}-Instruct students on low-disclosure settings, using a Qwen2.5-32B-Instruct teacher (p=1.0).

			Thought	
Benchmark	Size	Clean	Drop	Replace
FinQA	0.5B	$52.7_{(0.6)}$	$3.0_{(0.3)}$	$3.0_{(0.3)}$
	3B	$63.3_{(0.7)}$	$47.7_{(1.5)}$	$49.0_{(1.9)}$
	7B	$68.0_{(0.6)}$	$54.4_{(1.2)}$	$52.9_{(1.7)}$
	14B	$72.7_{(0.7)}$	$56.2_{(1.6)}$	$56.6_{(1.3)}$
MultiHiertt	0.5B	$32.3_{(1.2)}$	$23.8_{(1.3)}$	$21.6_{(1.8)}$
	3B	$49.2_{(0.6)}$	$39.5_{(1.1)}$	$40.5_{(1.6)}$
	7B	$55.3_{(0.8)}$	$46.0_{(0.9)}$	$46.4_{(1.4)}$
	14B	$60.1_{(0.6)}$	$50.4_{(1.2)}$	$50.1_{(1.1)}$
MathQA	0.5B	$6.0_{(0.3)}$	$1.1_{(0.2)}$	$3.0_{(0.7)}$
	3B	$17.1_{(0.9)}$	$10.7_{(0.8)}$	$10.3_{(1.2)}$
	7B	$29.5_{(0.6)}$	$13.2_{(1.0)}$	$17.2_{(1.5)}$
	14B	$41.2_{(1.2)}$	$13.5_{(1.0)}$	$17.8_{(1.9)}$
GeoQA	0.5B	$25.2_{(0.4)}$	$5.8_{(0.6)}$	$4.0_{(0.3)}$
	3B	$38.8_{(0.6)}$	$20.1_{(0.5)}$	$21.9_{(0.7)}$
	7B	$52.3_{(0.5)}$	$22.3_{(0.7)}$	$30.2_{(0.6)}$
	14B	$56.7_{(0.4)}$	$30.6_{(0.6)}$	$37.0_{(0.5)}$

F Controlling for Inference-Time Thought Generation

Our thought-drop intervention changes not only the supervision available during training but also the student's generation behavior at inference time: full-trajectory students generate a thought before each action, whereas students trained without thoughts produce actions directly. The performance gap on low-disclosure settings (Section 5.2) therefore admits two explanations: teacher thoughts may provide training signal that cannot be learned from actions alone, or full-trajectory students may simply benefit from additional natural-language reasoning at inference time. To separate these two factors, we examine whether full-trajectory students retain their advantage when thought generation is suppressed at inference time.

Specifically, we take the full-trajectory students trained with the original anonymous action representation and evaluate them under a direct-action inference setting (Full + Direct). The model parameters are kept unchanged; only the inference prompt is modified to request the action directly without an intermediate thought. Since neither Full + Direct nor Action-only generates thoughts at inference time, their performance difference cannot be explained by additional inference-time thought generation.

As shown in Table 12, Full + Direct lies between Action-only and Full + Thought on all four benchmarks, recovering 41.2–67.2% of the gap between them. This estimate may be conservative, as Full + Direct is evaluated in an inference format unseen during training. These results show that thought supervision provides benefits beyond inference-time thought generation: full-trajectory students consistently outperform action-only students even when both predict actions directly.

Table 10: Wall-clock training and inference times under full-trajectory and action-only distillation. Absolute times are reported in seconds, and reductions are relative to full-trajectory distillation.

	Training			Inference		
Benchmark	Full (s)	Action-only (s)	Reduction (%)	Full (s)	Action-only (s)	Reduction (%)
HotpotQA	$441.3_{(11.8)}$	$410.0_{(1.0)}$	7.1	$687.8_{(5.2)}$	$492.6_{(1.2)}$	28.4
FEVER	$377.3_{(4.2)}$	$353.7_{(2.3)}$	6.3	$361.7_{(1.5)}$	$321.7_{(2.1)}$	11.1
ALFWorld	$2744.0_{(21.3)}$	$2463.7_{(21.4)}$	10.2	$891.0_{(34.7)}$	$519.3_{(5.7)}$	41.7
WebShop	$1890.3_{(15.9)}$	$1791.0_{(8.5)}$	5.3	$608.3_{(13.0)}$	$512.0_{(19.0)}$	15.8

```
=== System prompt ===
You write short first-person reasoning narration for a math problem-solving agent. You are given the problem
and the EXACT correct arithmetic already worked out. Do not solve anything -- just narrate it naturally and
concisely, as if you were figuring it out for the first time.

=== User prompt ===
Problem: {question}

The correct calculation (already verified correct) is:
{trace}

Final result: {final_result}

Write short reasoning snippets as this agent would say them.
```

Figure 18: Prompt used to generate thoughts for low-disclosure trajectories.

```
=== System prompt ===
You rewrite Python snippets to be readable. You are given a quantitative reasoning problem and a correct
snippet whose variables are called step0, step1, ... Rename every variable to a descriptive name drawn from the
question. NEVER change any number, any operator, or the order of operations. Output only the rewritten code, no
fences and no explanation.

=== User prompt ===
Question: {question}

Snippet:
{code}

Rewritten:
```

Figure 19: Prompt used for semantic naming in the disclosure intervention.

Table 11: Thought-drop retention under the anonymous, semantic, and mismatched-name action representations in the low-disclosure settings, using a Qwen2.5-3B-Instruct student. Δ denotes the absolute accuracy change from Clean to Drop (percentage points).

Benchmark	Action representation	Clean	Drop	Δ	Retention (%)
FinQA	Anonymous (= before disclosing)	63.3	47.7	-15.6	75.4
	Semantic (= after disclosing)	65.3	61.8	-3.5	94.6
	Mismatched	45.3	36.5	-8.8	80.6
MultiHiertt	Anonymous (= before disclosing)	49.2	39.5	-9.7	80.3
	Semantic (= after disclosing)	50.6	44.6	-6.0	88.1
	Mismatched	33.8	26.3	-7.5	77.8
MathQA	Anonymous (= before disclosing)	17.1	10.7	-6.4	62.6
	Semantic (= after disclosing)	18.3	16.7	-1.6	91.3
	Mismatched	16.7	12.7	-4.0	76.0
GeoQA	Anonymous (= before disclosing)	38.8	20.1	-18.7	51.8
	Semantic (= after disclosing)	41.0	34.6	-6.4	84.4
	Mismatched	26.8	16.3	-10.5	60.8

Table 12: Accuracy of full-trajectory-trained students, with and without inference-time thought generation, and action-only students in the low-disclosure settings. FULL + THOUGHT uses the standard thought–action inference format, whereas FULL + DIRECT suppresses thought generation through the prompt and directly predicts the action. ACTION-ONLY denotes students trained without thoughts. All results use Qwen2.5-3B-Instruct students with the anonymous action representation.

Benchmark	FULL + THOUGHT	FULL + DIRECT	ACTION-ONLY
FinQA	$63.3_{(0.7)}$	$57.1_{(1.0)}$	$47.7_{(1.5)}$
MultiHiertt	$49.2_{(0.6)}$	$43.5_{(0.9)}$	$39.5_{(1.1)}$
MathQA	$17.1_{(0.9)}$	$15.0_{(1.2)}$	$10.7_{(0.8)}$
GeoQA	$38.8_{(0.6)}$	$32.0_{(0.7)}$	$20.1_{(0.5)}$