TEACHING AGENTS IN THEIR OWN STYLE: BLACK-BOX TEACHER DISTILLATION FOR WEB AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Frontier web agents using closed models have high inference and computational costs. Smaller open models can be an alternative, but to be performant, they require training supervision from a human or teacher model. Token-level teacher distillation is not an option for closed teachers as it needs access to the model's logits. The standard approach is to imitate teacher demonstrations offline, but it trains the student on states it has never visited, and in a style it does not write in. We introduce WEBKD, where the teacher is the editor of the student's failures. Given a failed episode, the teacher finds the first point of mistake, and rewrites the student's reasoning traces, harness states, and actions from that point onward. The editing reveals that the teacher's corrections can be more surprising to the student than the student's own text. WEBKD instead aims to edit minimally. In the data space, the teacher writes each correction in the student's style. In the objective space, the reasoning is weighted by its contribution to the action. A role-asymmetric KL anchor allows the corrected action to move freely while keeping the reasoning near the student's prior. On the TimeWarp benchmark, WEBKD students gain about 5.5-10.7pp over behavior cloning, outcome-reward RL, and teacher editing baselines. The performance transfers to unseen versions and to the live web in the OnlineMind2Web benchmark. WEBKD opens a new way of distilling black-box models to improve the capabilities of open-weight web agents.

1 INTRODUCTION

Agents built on the largest proprietary large language models (LLMs) now perform remarkably well on web tasks (Song et al., 2025; Xue et al., 2025). Such a task hands the agent a goal and a browser, and asks it to navigate complex, changing interfaces, plan across tens of steps, and carry intermediate findings through to a final answer. Deploying these agents for everyday use is another matter. Frontier models have high latency because they are large reasoning models. The computational cost makes them less viable as agents that run in the background of daily work. Robotics faced a similar problem, which led to compact Vision-Language-Action (VLA) models distilled for real-time control (Zitkovich et al., 2023; Kim et al., 2025). Web agents also need a small and cheap open-weight model that runs at interactive latency but carries the competence closer to its frontier teacher.

This raises the main challenge we address in this paper: *how do we effectively transfer the competence of the black-box, teacher model to the small, more efficient model?* Web tasks can be long-horizon over partially observed pages, and the agent must plan explicitly while reasoning in natural language (Xue et al., 2025). The harness truncates the history to recent observations due to the limited context. So the agent also keeps a scratchpad memory, and what it writes there, together with its earlier responses, is the only state that persists across steps (de Chezelles et al., 2025) (§3.1).

Secondly, the outcome reward is a single value per episode, far too sparse to shape multiple structured decisions with reinforcement learning (RL) (Shao et al., 2024; Wei et al., 2025). Teacher supervision is denser, but the strongest teachers are black boxes that expose no internals or logits. Every distillation method that consumes the teacher's next-token distribution, *e.g.*, token-level knowledge distillation (KD), reverse Kullback–Leibler (KL) objectives, and on-policy distillation (Gu et al., 2024; Agarwal et al., 2024; Lu & Thinking Machines Lab, 2025), is therefore unavailable. What a frontier teacher *can* emit is text. Our research asks *how much of a black-box frontier teacher's competence can be distilled into a small web agent through text alone*.

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.

The standard approach is behavior cloning, which involves sampling teacher demonstrations and fine-tuning the student on them offline (Kim & Rush, 2016). For web agents, this is known to help substantially (He et al., 2025; Ishmam & Marino, 2026), and it is where our pipeline begins. But behavior cloning creates two new problems. It trains the student only at states the *teacher* visits, so at deployment, small errors carry the student into states its supervision never covered, and the errors compound over the horizon (Ross et al., 2011). Secondly, it trains the student on the teacher's surface distribution, unintentionally importing a writing style that the small model cannot comfortably produce. *Is offline imitation really the most a teacher has to offer?*

The classical answer to this is DAgger (Ross et al., 2011), where the expert relabels the states the *learner* visits. However, DAgger corrects only the actions while a web agent's generation is a structured response (Fig. 1). With truncated history, the non-action parts of that response form the belief state that guides later decisions. Fixing the click while leaving the reasoning that led to it unchanged, therefore fixes the immediate mistake but not its source. Conversely, on-policy expert correction (Lauffer et al., 2026) gives control to the teacher from the point of failure and lets it finish the episode. So the completed suffix is the teacher's trajectory, which is off-policy in state and style.

An alternative is to use the teacher as an *editor* of the student's own failures. The teacher reads a failed episode, localizes the earliest erroneous step, and rewrites the student's full structured response, reasoning, and harness traces included, from that step to the end. Each rewrite is a single-step teacher rollout on a state the student actually visited, so the supervision is on-policy. However, it is not on-policy in style. A frontier teacher's corrections are *surprising* to the student, which are at several times the perplexity of the student's own steps. This gap is especially large in the reasoning components, where perplexity can increase by up to an order of magnitude, while the action itself remains much closer to the student's distribution.

Under uniform fine-tuning, the loss is largest where the supervision is most foreign, which is the teacher's *wording* rather than its *decisions*. WEBKD attacks this *perplexity premium* by editing the rationale *minimally*. In data space, the teacher writes each correction in the student's own style from in-context exemplars of the student's writing. In objective space, each reasoning span is weighted by how much it raises the student's probability of the corrected action. A role-asymmetric anchor allows the corrected action to move freely while keeping the reasoning close to the student's prior. The student adopts the teacher's decisions in full, while being closer to its own writing style.

Our contributions are as follows:

- **Problem.** We formulate distillation from a *black-box* frontier teacher into a small web agent by using teacher-edits and identify a key obstacle, the *on-policy perplexity premium* (Defn. 1). The premium is the excess surprisal of the student on the teacher supervision, and it concentrates on the reasoning components rather than the action (Fig 2).
- **Method.** We introduce WEBKD, a new distillation method that uses the teacher to edit the student's failures without the teacher's logits (Fig.1). The teacher localizes the first erroneous step and rewrites the subsequent reasoning traces, scratchpad memory, and actions in the student's style instead of imitating teacher-style traces (§3.3). The student then trains under a novel objective function that weights each reasoning span by how much it informs the action (§3.3.1) and anchors the reasoning to the student's prior (§3.3.2).
- **Evidence.** On the TimeWarp benchmark, WEBKD lifts the performance of students by $5.5 - 10.7$ pp across behavior cloning, outcome-reward RL, and teacher editing baselines using Qwen and Gemma models (Tab. 1). The gains translate to unseen TimeWarp versions, and the live web benchmark, OnlineMind2Web (Fig. 4). Our ablations vary the source of the edits and styles, training objectives, and hyperparameters (Tables 2, 3, Fig. 5).

2 RELATED WORK

2.1 WEB AGENTS

Web agents, LLM-driven agents that operate a browser on a user's behalf (Deng et al., 2023), are a subcategory of computer-use agents, which perceive rendered user interfaces and act through UI-level commands (Xie et al., 2024). The benchmark ecosystem spans curated demonstrations (Deng et al., 2023), sandboxed sites (Zhou et al., 2024), live websites (He et al., 2024), and unified har-

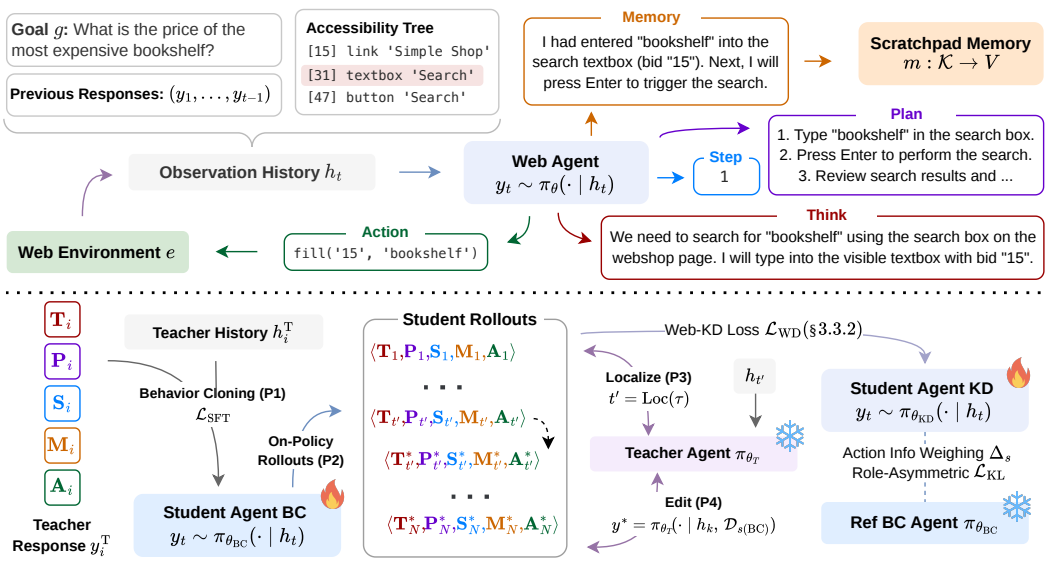


Figure 1: **Overview of WEBKD.** *(Top)* At step t, the harness generates the observation history from the goal, accessibility tree, and previous responses. The web agent emits a five-component response conditioned on the observation history. The Think, Plan, and Step components carry their reasoning and running plan, the Memory component reads from and writes to the scratchpad memory, and only the Action component is executed in the web environment, which returns the next observation. *(Bottom)* The student is behavior-cloned on the teacher's rollouts **(P1)** and then generates rollouts from the training tasks **(P2)**. The teacher localizes each failure at t' **(P3)** and rewrites the faulty trajectory suffix in the student's style **(P4)**. The resulting corpus $\mathcal{D}$ trains the student under action informativeness weighing, and KL anchored objective of §3.3.1–§3.3.2.

nesses (de Chezelles et al., 2025). TimeWarp (Ishmam & Marino, 2026) additionally varies the *era* of the interface and tests robustness to temporal UI change. On these suites, proprietary frontier LLMs reach strong success rates, but their cost and latency make them impractical as always-on daily drivers. That gap motivates the small open-weight students this paper targets.

Training pipelines for such students span imitation and RL. WebRL (Qi et al., 2025), WebAgent-R1 (Wei et al., 2025), and DigiRL (Bai et al., 2024) train with outcome-reward RL and inherit its sparse signal, while OpenWebVoyager (He et al., 2025) and TimeWarp (Ishmam & Marino, 2026) fine-tune on teacher demonstrations, the recipe we start from. We differ in what happens after imitation. The teacher corrects the student's own failures instead of supplying more demonstrations or rewards. Closest to us is WebCoT (Hu et al., 2025), which also distills reconstructed chain-of-thought rationales. However, the teacher rewrites stored records with template heuristics against observations simulated in the style of WebDreamer (Gu et al., 2025), and it does not localize the first fatal step before rewriting. WEBKD edits *actual* failures against the observations the student saw from the localized step in the student's own style, using both the data and training objective.

2.2 KNOWLEDGE DISTILLATION

Knowledge distillation transfers a teacher's behavior to a smaller student (Hinton et al., 2015), classically by matching token-level distributions or by fine-tuning on teacher generations at the sequence level (Kim & Rush, 2016). For LLMs, the white-box distillation methods sharpen this line of work with reverse-KL objectives (Gu et al., 2024), on-policy distillation that scores *student-sampled* prefixes under the teacher (Agarwal et al., 2024; Lu & Thinking Machines Lab, 2025), and skew-divergence stabilizations (Ko et al., 2024). All of them consume the teacher's next-token distribution over a shared vocabulary. Black-box agents do not expose these, which forces distillation through text only. This introduces the notion of a perplexity premium (§3.2), which has previously been observed with LLM distillation (Ren et al., 2024) and is one of our paper's main focuses.

A second lineage imitates the expert at the learner's own states. DAgger (Ross et al., 2011) showed that relabeling learner-visited states turns quadratic compounding error into linear error, and LLM-

era variants revive the recipe through turn-level policy interpolation (Li et al., 2026a), mid-trajectory expert takeover (Lauffer et al., 2026), and iterated fine-tuning on privileged feedback (Choudhury & Sodhi, 2025). We keep the state-distribution insight but change the label. Our teacher rewrites the full structured response offline instead of relabeling actions online. A third line removes the external teacher. Rejection fine-tuning keeps self-generated successes (Zelikman et al., 2022; Yuan et al., 2023), and on-policy self-distillation conditions the same model on privileged information such as a gold solution (Zhao et al., 2026), consensus (Li et al., 2026c), or rich feedback (Hübotter et al., 2026), with agentic variants that target failure-relevant spans (Yeo et al., 2026; Li et al., 2026b). Self-distillation presupposes the competence the model already has, which a 4B student on realistic web tasks typically lacks (Tab. 1). So we keep an external frontier teacher and hold its supervision close to the student's own states and style.

3 METHODOLOGY

3.1 BACKGROUND

Web tasks as POMDP. A web task is specified by a natural-language goal g on an interactive browser environment e (Zhou et al., 2024; de Chezelles et al., 2025). Performing a web task is naturally formalized as a finite-horizon partially observable Markov decision process (POMDP) (Kaelbling et al., 1998) $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, \mathbb{T}, \mathbb{T}_0, \mathbb{O}, R \rangle$, with a step budget of T actions per episode. The underlying browser state $s_t \in \mathcal{S}$ contains the full document object model (DOM), session, and site state, and is never observed directly. The initial state is drawn as $s_0 \sim \mathbb{T}_0$. The agent receives the rendering $o_t = \mathbb{O}(s_t) \in \mathcal{O}$, in the form of HTML or an AXTree. An LLM policy π_θ generates a structured response $y_t \sim \pi_\theta(\cdot \mid h_t)$ autoregressively, conditioned on the observation history h_t. The browser-level action $a_t \in \mathcal{A}$, *e.g.* `click('63')`, is deterministically parsed from y_t and executed, producing o_{t+1} through the effectively deterministic transition $\mathbb{T}$. The reward is sparse and terminal: $R \in \{0, 1\}$, assigned once at the end of the episode. The task goal g is rendered into every observation, so the full interaction history is $(o_{1:t}, y_{1:t-1})$. Carrying several high-dimensional observations (a single AXTree runs to thousands of tokens) overflows the context window, so the history is truncated to the *most recent* observation together with the full response history, $h_t = (o_t, y_{1:t-1})$.

Web harness. Between the policy and the browser sits the *harness*, the interface that turns the raw POMDP into an agentic loop. A harness also maintains its internal state across steps, namely the response history $y_{1:t-1}$, which carries the scratchpad memory s_m as the sequence of memory components emitted so far (§A.4). At step t, the harness (i) generates the observation history h_t, (ii) obtains the structured response $y_t \sim \pi_\theta(\cdot \mid h_t)$, (iii) parses the action $f_{\text{parse}}(y_t)$, and (iv) executes the action to advance the environment $s_{t+1} = \mathbb{T}(s_t, a_t)$ and $o_{t+1} = \mathbb{O}(s_{t+1})$.

Agent response. The harness enforces a five-component response, $y_t = \langle\, y_t^{\texttt{think}},\ y_t^{\texttt{plan}},\ y_t^{\texttt{step}},\ y_t^{\texttt{memory}},\ y_t^{\texttt{action}} \,\rangle = \langle \mathrm{T}, \mathrm{P}, \mathrm{S}, \mathrm{M}, \mathrm{A} \rangle$, generated sequentially (Fig 1) (de Chezelles et al., 2025). `think` tokens constitute the free-form reasoning grounded in the current observation, `plan` tokens form the running, numbered multi-step plan, re-emitted and revised at every step, the `step` token(s) is a single integer naming the current plan step, `memory` tokens are the agent's interface to the scratchpad memory s_m (§A.4), and `action` tokens form one executable browser command. Only the `action` tokens ($y_t^{\texttt{action}} = a_t$) interact with the environment. Every other component acts on the harness state alone and enters future contexts as part of the observation history h_t.

We provide additional preliminary discussion on behavior cloning (§A.1), knowledge distillation (§A.2), and the black-box constraint (§A.3) in the appendix.

3.2 PERPLEXITY PREMIUM

Teacher edits are high-perplexity text for the student. The perplexity of a response y under a policy π with context x,

$$\mathrm{PPL}_\pi(y \mid x) \;=\; \exp\Big(-\tfrac{1}{|y|} \textstyle\sum_i \log \pi(y_i \mid x,\, y_{<i}) \Big), \tag{1}$$

measures how natural the text is to that policy. 1 means the policy would have emitted it with certainty, and each unit of log-perplexity is a nat of average per-token surprise. The relative quantity that matters for training is the premium the student pays on its supervision.

Definition 1 (On-policy perplexity premium)**.** *For a step with student response y and supervision target y^* at the same observation history h, the premium is the excess per-token surprisal, under the behavior-cloned student $\pi_{\theta_{BC}}$ (§A.1), of the supervision relative to the student's own text:*

$$\Pi(y^*, y) \;=\; \frac{1}{|y^*|}\sum_i -\log \pi_{\theta_{BC}}(y_i^* \mid h, y_{<i}^*) \;-\; \frac{1}{|y|}\sum_i -\log \pi_{\theta_{BC}}(y_i \mid h, y_{<i})\,. \tag{2}$$

This premium is one of the obstacles to distilling a frontier teacher through text. As shown in Fig. 2, the teacher edits are high-perplexity for the student, while WEBKD's corpus stays close to the student's own continuations. Per-token cross-entropy is the surprisal inside Eq. 1, and under uniform SFT, the loss is largest on the tokens most foreign to the student. On raw teacher edits, those are mostly the rationale tokens. The action tokens that fix the failure carry only a few percent of the loss. The surprising spans mix the teacher's style with its content and reasoning. What it shows is where uniform SFT spends its loss, which motivates the objective of Eq.5. Any method that trains on teacher edits must address the premium. Our method attacks it from both sides. *In data space*, before training, the correction is written in the student's own style (P4 of §3). *In objective space*, during training, the learning signal is concentrated on the tokens that inform the action (§3.3.1), and the rationale is anchored to the student's own prior so that the premium the student is forced to absorb stays bounded (§3.3.2).

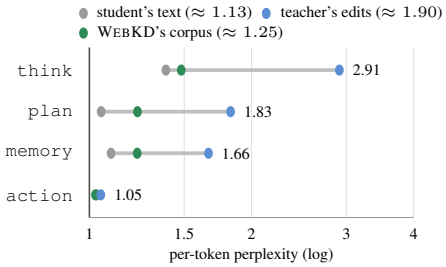


Figure 2: The perplexity premium by response components against the student's continuations, the teacher's raw corrections, and the WEBKD's corpus with in-context edits.

3.3 WEB KNOWLEDGE DISTILLATION (WEBKD)

The core of WEBKD is to correct the student's decisions at the states it visited, rewriting only the steps from the failure onward. *How can the student policy be improved while minimizing the student's surprisal to teacher edits?* WEBKD starts from a student agent fine-tuned to replicate the teacher (**P1**). We then collect rollouts from the fine-tuned student agent and keep its failures (**P2**). The edit itself is a two-stage teacher intervention: *where did the student fail?* (**P3**), and *how should the teacher correct it?* (**P4**). The corpus of corrected trajectories trains the student under a premium-aware objective: action-informative token weighting (§3.3.1) and a role-asymmetric anchor to the student's prior (§3.3.2). Figure 1 gives an overview, and Algorithm 1 states the full method. We characterize in closed form what the WEBKD objective optimizes (Proposition 2) and bound the student's failure rate by the teacher's localization and correction errors (Theorem 1).

P1: Behavior cloning on teacher rollouts. WEBKD first replicates the teacher rollouts on the training tasks via behavior cloning at the sequence level. For the teacher π_{θ_T} acting as the agent under the same harness, the base model π_{θ_0} is fine-tuned on the teacher demonstration trajectories $\mathcal{D}_T = \{\tau_n\}_{n=1}^N$, where each $\tau_n = (h_1, y_1, \ldots, h_T, y_T)$ is a closed-loop teacher rollout, $y_i \sim \pi_{\theta_T}(\cdot \mid h_i)$, kept only if its terminal reward is $R(\tau_n) = 1$. Training uses the uniform BC objective $\mathcal{L}_{\text{BC}}$ of Eq. 6 to produce $\pi_{\theta_{\text{BC}}}$, which is then frozen and reused in the following phases.

P2: Student on-policy rollout. We collect on-policy rollouts from the behavior-cloned student on the training tasks. Each rollout is generated by the agentic loop of §3.1, every response drawn from the student's own policy at a context the student itself reached, $y_t \sim \pi_{\theta_{\text{BC}}}(\cdot \mid h_t)$ and we write $\tau \sim \pi_{\theta_{\text{BC}}}$ for the induced trajectory distribution. $\mathcal{D}_{\text{BC}}$ collects these rollouts. The failed trajectories $\mathcal{D}_{f(\text{BC})} = \{\tau \in \mathcal{D}_{\text{BC}} : R(\tau) < 1\}$ become the substrate for editing, and the solved episodes are kept as a *style pool* $\mathcal{D}_{s(\text{BC})} = \{\tau \in \mathcal{D}_{\text{BC}} : R(\tau) = 1\}$, mined for in-context exemplars in P4. Every observation in $\mathcal{D}_{f(\text{BC})}$ has been visited by the student, which places the teacher's supervision of the following phases within the student's own state distribution.

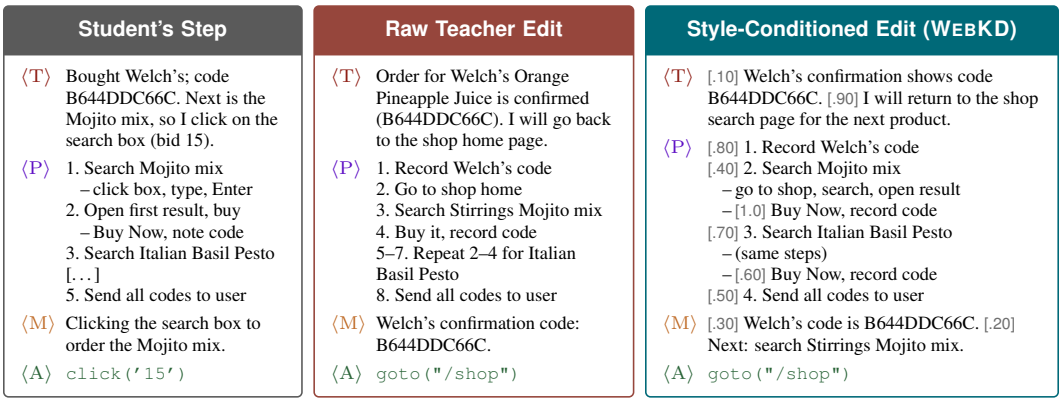


Figure 3: **Edits for the same student step.** The raw teacher edit replaces the student's action but rewrites the plan in its own style. The style-conditioned edit keeps the student's nested plan format. Values in brackets give the span weight v_s. ⟨**T**⟩: think, ⟨**P**⟩: plan, ⟨**M**⟩: memory, ⟨**A**⟩: action.

P3: Failure localization. For each failed trajectory, the teacher first localizes the error to prevent editing the correct steps. We establish that web trajectories carry enough structure for the "point of failure" to be a well-defined property of the trajectory (§B.2). The teacher call is an *estimator* of $t^\star$, reads the full trajectory $\tau \in \mathcal{D}_{f(\mathrm{BC})}$ and returns an index, $t' = \mathrm{Loc}(\tau) \in \{1, \ldots, T\}$, the *earliest* step whose action sets the trajectory on the path to failure. Localization splits the trajectory into a replayed prefix $\tau_p = (h_1, y_1, \ldots, h_{t'-1}, y_{t'-1})$, which is kept verbatim, and a faulty suffix $\tau_s = (h_{t'}, y_{t'}, \ldots, h_T, y_T)$, which will be rewritten.

P4: Teacher editing in the student's style. After localization, the teacher corrects the structured response of every step in the faulty suffix. Each correction is conditioned on the observation the student saw at that step, re-rendered by replaying the student's original actions in the deterministic container, so the teacher sees exactly what the student saw. The correction query also carries the student's response style *in context*. The corpus targets are

$$y_k^* = y_k \quad \forall k < t' \text{ (replay)}, \qquad y_k^* \sim \pi_{\theta_T}\big(\cdot \mid h_k, \mathcal{D}_{s(\mathrm{BC})}\big) \quad \forall k \geq t' \text{ (edit)}, \tag{3}$$

where each edit is a 1-step rollout of the teacher policy at the *student's* context. π_T is conditioned on the observation history h_k the student saw (the same rendered goal, AXTree, and response history) and on in-context exemplars from the student's style pool $\mathcal{D}_{s(\mathrm{BC})}$, one independent call per step. The edited structured response y_k^* comprises the edited tokens $\langle \mathrm{T}^*, \mathrm{P}^*, \mathrm{S}^*, \mathrm{M}^*, \mathrm{A}^* \rangle$ (Details in §B.3).

3.3.1 Action-informative token weighting

P4 first lowers the premium in data space by having the teacher write in the student's style. It cannot lower it to zero without giving up the correction. Some premium must be paid, and Fig. 2 shows that on raw teacher edits, it falls almost entirely on the rationale components. The question for the objective is therefore *which* rationale tokens are worth their premium.

We ask, *which tokens carry the learning signal?* A rationale span is worth its premium to the extent that it changes the action, so Eq.4 ranks the spans of each step by their measured effect on the student's probability of the corrected action. Second, *how far the policy may move?* Eq.5 adds a role-asymmetric anchor that damps how far the rationale moves toward the teacher's phrasing while the action follows the teacher in full. Together, the objective pays the premium mainly where it buys the action. We weigh each rationale span by how much it contributes to the decision that follows.

Let s be a non-action segment, a sentence or clause of the edited rationale $\langle \mathrm{T}^*, \mathrm{P}^*, \mathrm{M}^* \rangle$. Its leave-one-out contribution to the teacher's corrected action, measured under the *student* is

$$\Delta_s = \log P_{\pi_{\theta_{\mathrm{BC}}}}(a^* \mid h, y^*) - \log P_{\pi_{\theta_{\mathrm{BC}}}}(a^* \mid h, y^* \setminus s), \quad s \subset \langle \mathrm{T}^*, \mathrm{P}^*, \mathrm{M}^* \rangle, \tag{4}$$

where $a* = \langle \mathrm{A}^* \rangle$ denotes the teacher-edited action tokens, h is the observation history, and $y^* \setminus s$ is the edited response with segment s deleted. As Δ_s is the *student's own* credit assignment over the teacher's rationale, it needs no teacher logits and is computable in the black-box setting. Perplexity says how foreign a span is to the student. Δ_s says how much the span raises the student's probability

Table 1: Success rate (%) on the TimeWarp test split, in domain on v6 and on the five held-out versions v1–v5. Results are averaged over three runs with the standard error as the subscript.

Model	Baseline	v1	v2	v3	v4	v5	v6 (seen)
–	Teacher (GPT-5.5)	$71.8_{\pm 4.43}$	$71.8_{\pm 4.43}$	$68.0_{\pm 4.60}$	$68.0_{\pm 4.60}$	$69.9_{\pm 4.52}$	$70.9_{\pm 4.58}$
Qwen 3.5 4B	Zero-Shot	$1.0_{\pm 0.57}$	$1.0_{\pm 0.57}$	$0.3_{\pm 0.31}$	$1.6_{\pm 0.71}$	$0.0_{\pm 0.00}$	$1.0_{\pm 0.57}$
	Behavior Cloning	$10.7_{\pm 1.76}$	$9.4_{\pm 1.66}$	$9.7_{\pm 1.68}$	$10.7_{\pm 1.76}$	$5.2_{\pm 1.26}$	$9.7_{\pm 1.68}$
	Behavior Cloning (TW)	$27.5_{\pm 2.54}$	$28.5_{\pm 2.57}$	$27.8_{\pm 2.55}$	$26.2_{\pm 2.50}$	$28.2_{\pm 2.56}$	$29.8_{\pm 2.60}$
	m-GRPO	$30.1_{\pm 2.62}$	$27.2_{\pm 2.54}$	$26.2_{\pm 2.50}$	$28.2_{\pm 2.56}$	$30.1_{\pm 2.61}$	$30.4_{\pm 2.62}$
	Self/Student Edit	$25.2_{\pm 2.47}$	$19.4_{\pm 2.25}$	$27.2_{\pm 2.53}$	$24.3_{\pm 2.56}$	$23.3_{\pm 2.61}$	$23.6_{\pm 2.62}$
	Off-policy Takeover	$24.4_{\pm 2.45}$	$24.3_{\pm 2.44}$	$25.2_{\pm 2.48}$	$23.3_{\pm 2.41}$	$27.2_{\pm 2.54}$	$25.6_{\pm 2.49}$
	Student Re-style Edit	$28.8_{\pm 2.58}$	$23.6_{\pm 2.42}$	$23.7_{\pm 2.43}$	$26.5_{\pm 2.52}$	$28.3_{\pm 2.57}$	$26.9_{\pm 2.53}$
	WEBKD	$\mathbf{37.9}_{\pm 2.77}$	$\mathbf{36.3}_{\pm 2.73}$	$\mathbf{35.3}_{\pm 2.72}$	$\mathbf{38.8}_{\pm 2.77}$	$\mathbf{36.9}_{\pm 2.74}$	$\mathbf{41.1}_{\pm 2.80}$
Qwen 3.5 9B	Zero-Shot	$0.0_{\pm 0.00}$	$0.3_{\pm 0.31}$	$1.6_{\pm 0.71}$	$1.3_{\pm 0.65}$	$0.0_{\pm 0.00}$	$0.6_{\pm 0.44}$
	Behavior Cloning	$12.3_{\pm 1.87}$	$10.0_{\pm 1.71}$	$6.8_{\pm 1.43}$	$11.7_{\pm 1.82}$	$4.9_{\pm 1.23}$	$12.3_{\pm 1.87}$
	Behavior Cloning (TW)	$25.9_{\pm 2.49}$	$24.3_{\pm 2.44}$	$22.0_{\pm 2.36}$	$25.2_{\pm 2.47}$	$26.2_{\pm 2.50}$	$25.2_{\pm 2.47}$
	m-GRPO	$26.9_{\pm 2.52}$	$27.8_{\pm 2.54}$	$28.8_{\pm 2.58}$	$31.1_{\pm 2.62}$	$31.4_{\pm 2.65}$	$26.5_{\pm 2.51}$
	Self/Student Edit	$22.3_{\pm 2.37}$	$23.2_{\pm 2.40}$	$22.7_{\pm 2.38}$	$25.7_{\pm 2.49}$	$22.7_{\pm 2.38}$	$23.0_{\pm 2.39}$
	Off-policy Takeover	$28.4_{\pm 2.57}$	$25.4_{\pm 2.48}$	$29.4_{\pm 2.59}$	$30.1_{\pm 2.61}$	$27.3_{\pm 2.53}$	$25.0_{\pm 2.46}$
	Student Re-style Edit	$17.3_{\pm 2.15}$	$19.5_{\pm 2.25}$	$18.7_{\pm 2.22}$	$14.4_{\pm 2.00}$	$16.4_{\pm 2.10}$	$20.2_{\pm 2.28}$
	WEBKD	$\mathbf{34.6}_{\pm 2.71}$	$\mathbf{32.3}_{\pm 2.66}$	$\mathbf{32.4}_{\pm 2.66}$	$\mathbf{34.5}_{\pm 2.70}$	$\mathbf{34.9}_{\pm 2.71}$	$\mathbf{34.6}_{\pm 2.71}$
Gemma 4 E4B	Zero-Shot	$1.9_{\pm 0.78}$	$7.8_{\pm 1.53}$	$7.8_{\pm 1.53}$	$1.9_{\pm 0.77}$	$3.9_{\pm 1.10}$	$2.6_{\pm 0.90}$
	Behavior Cloning	$17.5_{\pm 2.16}$	$16.5_{\pm 2.11}$	$18.4_{\pm 2.20}$	$15.5_{\pm 2.06}$	$16.5_{\pm 2.11}$	$19.4_{\pm 2.25}$
	Behavior Cloning (TW)	$28.2_{\pm 2.56}$	$28.2_{\pm 2.56}$	$32.0_{\pm 2.65}$	$30.1_{\pm 2.61}$	$28.8_{\pm 2.58}$	$29.4_{\pm 2.58}$
	m-GRPO	$16.5_{\pm 2.11}$	$14.6_{\pm 2.01}$	$17.5_{\pm 2.16}$	$18.4_{\pm 2.20}$	$14.6_{\pm 2.01}$	$20.7_{\pm 2.30}$
	Self/Student Edit	$22.3_{\pm 2.37}$	$24.3_{\pm 2.44}$	$21.4_{\pm 2.33}$	$22.3_{\pm 2.37}$	$21.7_{\pm 2.34}$	$23.0_{\pm 2.39}$
	Off-policy Takeover	$29.8_{\pm 2.60}$	$28.2_{\pm 2.56}$	$29.1_{\pm 2.58}$	$30.1_{\pm 2.61}$	$33.0_{\pm 2.66}$	$34.3_{\pm 2.70}$
	Student Re-style Edit	$20.4_{\pm 2.29}$	$25.2_{\pm 2.47}$	$23.3_{\pm 2.40}$	$24.3_{\pm 2.45}$	$23.3_{\pm 2.40}$	$28.2_{\pm 2.56}$
	WEBKD	$\mathbf{37.2}_{\pm 2.75}$	$\mathbf{34.0}_{\pm 2.69}$	$\mathbf{37.9}_{\pm 2.76}$	$\mathbf{36.6}_{\pm 2.74}$	$\mathbf{38.8}_{\pm 2.77}$	$\mathbf{39.8}_{\pm 2.78}$

of the corrected action, and two spans can be equally foreign while differing in this. The weighting in §B.4 ranks the spans of a step by Δ_s, so the span that moves the action the most keeps full weight, and the span that moves it the least drops to the floor. Hence, the premium is paid mainly where it changes the decision rather than where it only changes the wording or style. Details on the action informativeness mapped to the WEBKD loss have been provided in Appendix §B.4.

3.3.2 ROLE-ASYMMETRIC ANCHORING AND THE WEBKD OBJECTIVE

The weighting decides which spans contribute to the action, but does not bound the magnitude of their contribution. A decisive span in $\langle \mathrm{T}^* \rangle$ keeps full weight, and fitting it under the weighed SFT objective still asks the student to absorb that span's influence in full. This can potentially cause SFT to move the policy farther than intended. We therefore add a KL penalty (Hinton et al., 2015) toward $\pi_{\text{ref}} \equiv \pi_{\theta_{\text{BC}}}$, the frozen checkpoint each run starts from. The KL acts on the rationale tokens $\langle \mathrm{T}, \mathrm{P}, \mathrm{M} \rangle$ only, *i.e.*, we allow these tokens to drift more than the non-rationale $\langle \mathrm{A}, \mathrm{S} \rangle$ tokens. With $\beta \geq 0$ the anchor strength and $\gamma_i \in [0, 1]$ a per-token role mask, the WEBKD objective is

$$\mathcal{L}_{\text{WD}} = \underbrace{\frac{\sum_{i \in T} w_i \, \mathrm{CE}_i}{\sum_{i \in T} w_i}}_{\mathcal{L}_{\text{SW}}} + \beta \cdot \underbrace{\frac{\sum_{i \in T} \gamma_i \, \mathrm{kl}_i}{\sum_{i \in T} \gamma_i}}_{\mathcal{L}_{\text{KL}}} \tag{5}$$

where $d_i := \log \pi_{\text{ref}} - \log \pi_\theta(y_i^*)$, $\mathrm{kl}_i := e^{d_i} - d_i - 1$, and $\pi_{\theta_{\text{WD}}}$ is the student produced. We note that γ is *role-asymmetric*. $\gamma_i = 1$ on the $\langle \mathrm{T}, \mathrm{P}, \mathrm{M} \rangle$ tokens and $\gamma_i \approx 0$ on $\langle \mathrm{A}, \mathrm{S} \rangle$, so the student is free to adopt the teacher's corrected decisions while being held near its prior on deliberation. Normalizing by $\sum_i \gamma_i$ makes β a penalty *per anchored token*, comparable across masks that cover different numbers of tokens. Because kl_i is evaluated on corpus tokens rather than on samples from π_θ, it is a pointwise anchor penalty on the target sequence. Details on the intuition behind the update and theoretical analysis are provided in §E.

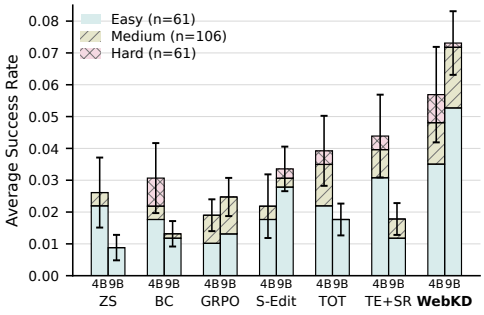


Figure 4: Success rate on Online-Mind2Web for the Qwen-3.5 4B and 9B students. The contributions are stacked across easy, medium, and hard categories.

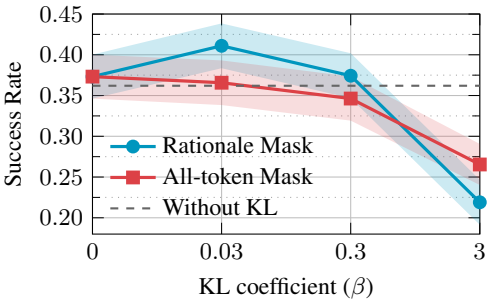


Figure 5: Success rate of Qwen-3.5 4B on TimeWarp v6 as a function of the anchor strength β with $\mathcal{L}_{\text{WD}}$, anchoring the rationale tokens or all tokens. The dashed line is without KL.

4 EXPERIMENTS

We organize our experiments around four research questions: **RQ1 (performance).** Does WEBKD's teacher edit distillation outperform behavior cloning, outcome-reward RL, or other methods of teacher edit? **RQ2 (transfer).** Do the gains survive interface versions never seen in training, and the live web? **RQ3 (edit-style).** Who should edit the student's decisions and in whose style? **RQ4 (objective).** Which tokens should the loss emphasize, and how should the policy be anchored?

4.1 EXPERIMENTAL SETUP

We describe our setup concisely here and include additional details in the Appendix (§C).

Models, Harness, Baselines. We compare latest open-weight models: Qwen-3.5 4B and 9B (Team, 2026), and Gemma 4 E4B (Team et al., 2026), in BrowserGym with the AgentLab harness (de Chezelles et al., 2025). We evaluate (a) zero-shot, (b) vanilla behavior cloning on actions, and the TimeWarp approach on the full response (Ishmam & Marino, 2026), (c) outcome-based RL baseline using multi-turn Group Relative Policy Optimization (m-GRPO) (Wei et al., 2025), and (d) black-box distillation baselines on student self-editing, off-policy takeover with the teacher completing the trajectory, the teacher prompting the student to restyle the edits, and (e) our method, WEBKD. The average results of three seeds are reported with the standard error.

Benchmarks. We evaluate on: (a) TimeWarp (TW) (Ishmam & Marino, 2026), a containerized web benchmark with websites from different eras of the internet, and (b) OnlineMind2Web (OM2W) (Xue et al., 2025), a live web benchmark with realistic tasks using real user workflows.

4.2 RESULTS

WEBKD outperforms every baseline on the seen distribution (RQ1). WEBKD improves over the strongest baseline by 10.7pp for Qwen 3.5 4B, 8.1pp for Qwen 3.5 9B, and 5.5pp for Gemma 4 E4B, on the seen TimeWarp version v6 (Tab. 1). Zero-shot students are near 0% on every version, so the task is out of reach without training. Behavior cloning is competitive only when non-action tokens are included in the loss (BC-TW). No single alternative is reliably second best. m-GRPO is the strongest baseline on both Qwen models, but falls on Gemma. The other teacher-text baselines show the opposite pattern. Off-policy takeover is the strongest baseline on Gemma (34.3%), but every teacher baseline trail m-GRPO on either the Qwen models.

The gains hold on unseen versions and transfer to the live web (RQ2). On the unseen versions v1–v5, WEBKD beats the strongest baseline by 3.0–10.6pp. Version shift itself costs little once a student is trained, and the seen–unseen gap stays below 5pp. WebKD's gains hold on the live web, where Online-Mind2Web's numbers are low for all methods (below 8%, Fig. 4). WEBKD is the only method that clearly separates from the rest, roughly doubling the strongest baseline for the 9B student. The gains are mostly concentrated on the `easy` tasks, which have a shorter horizon.

Table 2: **Edit–style ablation.** Teacher edits paired with in-context student style perform best. Qwen-3.5 4B baselines are trained using $\mathcal{L}_{\text{WD}}$, KL rationale anchor at $\beta = 0.03$.

ES	SS	Variant	S.R. ↑
T	T	Teacher takeover *(off-policy)*	$17.48_{\pm 2.16}$
T	T	Teacher edit + teacher style	$33.01_{\pm 2.68}$
T	S_{ic}	Teacher edit + student style *(in-context)*	$\mathbf{41.10_{\pm 2.80}}$
T	S_{inf}	Teacher edit + student re-style *(2-pass)*	$25.89_{\pm 2.49}$
S/T	S/T	Student proposes, teacher judges/fixes	$27.83_{\pm 2.55}$
S	S	Student-only edit	$24.27_{\pm 2.44}$

ES: Edit Source; SS: Style Source. T = Teacher, S = Student.

Table 3: **Loss ablation.** Token weightings on the filtered teacher rollouts, each with and without the KL rationale anchor at $\beta = 0.03$ using the Qwen-3.5 4B model.

Loss	w/ KL	w/o KL
$\mathcal{L}_{\text{CE}}$	$37.43_{\pm 2.75}$	$36.25_{\pm 2.73}$
$\mathcal{L}_{\text{WD}}$	$\mathbf{41.10_{\pm 2.80}}$	$\mathbf{38.51_{\pm 2.77}}$
$\mathcal{L}_{\text{GND}}$	$35.92_{\pm 2.73}$	$36.03_{\pm 2.73}$
$\mathcal{L}_{\text{CONS}}$	$37.54_{\pm 2.75}$	$35.81_{\pm 2.73}$
$\mathcal{L}_{\text{PPL}}$	$37.54_{\pm 2.75}$	$33.01_{\pm 2.68}$

The teacher should edit in the student's style, but minimally (RQ3). Following Tab. 2, off-policy takeover underperforms as it drifts too far from the states the student has visited. Edits or suggestions written by the student are worse still, which we can attribute to the student lacking a reliable sense of its own correctness. Asking the student to restyle the teacher's edit on a second pass can also undermine the rationale. This is most evident on Qwen 3.5 9B, where the student re-style edit is a weaker baseline (Tab.1). Although vanilla teacher editing works, its outputs deviate more from the student's style and carry a larger perplexity premium (Fig.2). In-context student re-styling, as used by WEBKD, stays closer to the student distribution and outperforms it by 8.1pp.

Loss should emphasize action informativeness, and anchor the rationale(RQ4). Tab. 3 compares token weightings of the loss, with and without the KL anchor on the rationale. $\mathcal{L}_{\text{WD}}$ with the kl anchor (Eq.5) outperforms the other token-weighting schemes (§C.2) by 3.6pp. Sweeping the anchor strength shows that $\beta = 0.03$ gives the highest success with rationale masking (Fig. 5).

Discussion. *Teacher capability matters.* With the same prompts, states, and style exemplars, using Claude Opus 5 or Qwen-3.8 27B as the editor trains students to 33.3% and 26.9%, versus 41.1% with GPT-5.5. As Qwen-3.8 is a smaller, less-performant model, the distilled student's performance is worse. *Why can't students self-improve?* On average, self-/student edit falls below BC (TW) on every backbone (Tab. 1). This is consistent with the findings on RQ3, *i.e.*, without an external signal, the student cannot reliably tell which of its own decisions were wrong. *How does* WEBKD *affect the behavior of the student?* It mostly prunes redundant actions, *e.g*, scrolling nearly disappears (-99.7%) and key presses drop by 24.5% (Fig. 6). `send_msg` rises by 12.3%, indicating higher task completion rate. *Where do the gains come from?* Our analysis reveals WEBKD's strongest gains come from verifying the steps and answer +14.6pp and exiting loops +9.4pp (detailed analysis in §D.1).

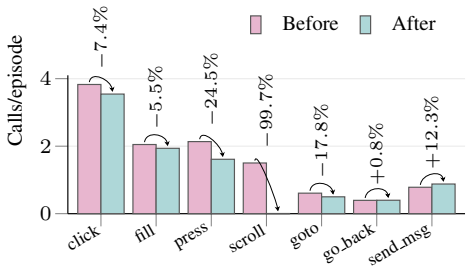


Figure 6: Change in action distribution before and after teacher distillation using WEBKDfor the Qwen-3.5 4B baseline.

5 CONCLUSION

Small, open-weight web agents are key to making web use automation widespread and accessible. Frontier agents have become highly capable, and distilling them into smaller models is a natural next step. Yet closed frontier models expose only their generated text, which rules out logit-based distillation. WEBKD addresses this by turning the teacher into an editor of the student's failures. It rewrites failed episodes in the student's own style and trains on them, using an objective that weights tokens based on action informativeness and a role-asymmetric KL anchor. Although WEBKD produced strong gains, the student still reaches only about half of the teacher's success rate, leaving substantial room for improvement. We hope this work spurs further research on black-box distillation and helps narrow the gap between closed frontier models and small open-weight agents.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*, 2024.

Hao Bai, Yifei Zhou, Mert Cemri, Jiayi Pan, Alane Suhr, Sergey Levine, and Aviral Kumar. DigiRL: Training in-the-wild device-control agents with autonomous reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Sanjiban Choudhury and Paloma Sodhi. Better than your teacher: LLM agents that learn from privileged AI feedback. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=st7XqFgbAH.

Thibault Le Sellier de Chezelles, Maxime Gasse, Alexandre Drouin, Massimo Caccia, Léo Boisvert, Megh Thakkar, Tom Marty, Rim Assouel, Sahar Omidi Shayegan, Lawrence Keunho Jang, et al. The BrowserGym ecosystem for web agent research. *Transactions on Machine Learning Research*, 2025.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=kiYqbO3wqw.

Yu Gu, Kai Zhang, Yuting Ning, Boyuan Zheng, Boyu Gou, Tianci Xue, Cheng Chang, Sanjari Srivastava, Yanan Xie, Peng Qi, Huan Sun, and Yu Su. Is your LLM secretly a world model of the internet? model-based planning for web agents. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=c6l7yA0HSq.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=5h0qf7IBZZ.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. WebVoyager: Building an end-to-end web agent with large multimodal models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Hongming Zhang, Tianqing Fang, Zhenzhong Lan, and Dong Yu. OpenWebVoyager: Building multimodal web agents via iterative real-world exploration, feedback and optimization. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27545–27564, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1336. URL https://aclanthology.org/2025.acl-long.1336/.

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

Minda Hu, Tianqing Fang, Jianshu Zhang, Jun-Yu Ma, Zhisong Zhang, Jingyan Zhou, Hongming Zhang, Haitao Mi, Dong Yu, and Irwin King. WebCoT: Enhancing web agent reasoning by reconstructing chain-of-thought in reflection, branching, and rollback. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, 2025.

Jonas Hübotter, Frederike Lübeck, Lejs Deen Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=QkfkxyRizZ.

Md Farhan Ishmam and Kenneth Marino. TimeWarp: Evaluating web agents by revisiting the past. In *Fortieth Conference on Neural Information Processing Systems Evaluations and Datasets Track*, 2026. URL https://arxiv.org/abs/2603.04949. To appear.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, 1998.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

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

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards streamlined distillation for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Niklas Lauffer, Xiang Deng, Srivatsa R Kundurthy, Brad Kenstler, and Jeff Da. Imitation learning for multi-turn LM agents via on-policy expert corrections. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=l0LA4SUA50.

Changhao Li, Rushi Qiang, Jiawei Huang, Chenxiao Gao, Chao Zhang, Niao He, and Bo Dai. Revisiting DAgger in the era of LLM-agents. *arXiv preprint arXiv:2605.12913*, 2026a.

Xiaozhe Li, Tianyi Lyu, Yang Li, Yichuan Ma, Peiji Li, Linyang Li, Qipeng Guo, Dahua Lin, and Kai Chen. What and when to distill: Selective hindsight distillation for multi-turn agents. *arXiv preprint arXiv:2605.19447*, 2026b.

Yijiang Li, Bingyang Wang, Yijun Liang, Yunjie Tian, Di Fu, and Nuno Vasconcelos. On-policy self-distillation without any supervision. *arXiv preprint arXiv:2608.06296*, 2026c.

Kevin Lu and Thinking Machines Lab. On-policy distillation. Thinking Machines Lab: Connectionism, 2025. https://thinkingmachines.ai/blog/on-policy-distillation.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022.

Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, et al. WebRL: Training LLM web agents via self-evolving online curriculum reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

Xuan Ren, Biao Wu, and Lingqiao Liu. I learn better if you speak my language: Understanding the superior performance of fine-tuning large language models with LLM-generated responses. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 10225–10245, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.571. URL https://aclanthology.org/2024.emnlp-main.571/.

Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 661–668, 2010.

Stéphane Ross, Geoffrey J Gordon, and J Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 627–635, 2011.

John Schulman. Approximating KL divergence, 2020. http://joschu.net/blog/kl-approx.html.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. DeepseekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Yixiao Song, Katherine Thai, Chau Minh Pham, Yapei Chang, Mazin Nadaf, and Mohit Iyyer. BEARCUBS: A benchmark for computer-using web agents. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=0JzWiigkUy.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Zhepei Wei, Wenlin Yao, Yao Liu, Weizhi Zhang, Qin Lu, Liang Qiu, Changlong Yu, Puyang Xu, Chao Zhang, Bing Yin, Hyokun Yun, and Lihong Li. WebAgent-r1: Training web agents via end-to-end multi-turn reinforcement learning. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 7909–7928, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.401. URL https://aclanthology.org/2025.emnlp-main.401/.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=tN61DTr4Ed.

Tianci Xue, Weijian Qi, Tianneng Shi, Chan Hee Song, Boyu Gou, Dawn Song, Huan Sun, and Yu Su. An illusion of progress? Assessing the current state of web agents. In *Conference on Language Modeling (COLM)*, 2025.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in GPT-4V. *arXiv preprint arXiv:2310.11441*, 2023.

Woongyeng Yeo, Yumin Choi, Taekyung Ki, and Sung Ju Hwang. HinT-SD: Targeted hindsight self-distillation for long-horizon agents. *arXiv preprint arXiv:2605.17873*, 2026.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*, 2023.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. In *ICLR 2026 Workshop on Lifelong Agents: Learning, Aligning, Evolving*, 2026. URL https://openreview.net/forum?id=31wkVZXEaJ.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. In *The Twelfth International Conference on Learning Representations*, 2024.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao

Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

A Additional Background Details

A.1 Behavior Cloning

Given a corpus $\mathcal{D}$ of (context, target) pairs, behavior cloning is performed via supervised fine-tuning, which minimizes the teacher-forced cross-entropy (Ouyang et al., 2022),

$$\mathcal{L}_{\mathrm{BC}}(\theta) \;=\; -\,\mathbb{E}_{(x,y)\sim\mathcal{D}}\,\frac{1}{|y|}\sum_{i=1}^{|y|}\log \pi_\theta(y_i \mid x,\, y_{<i})\,. \tag{6}$$

Behavior cloning is typically performed on the action tokens. However, the TimeWarp version (Ishmam & Marino, 2026) uses the full response, including the reasoning traces, and is the default type of behavior cloning in our experiments. We note that in Eq. 6, every target token carries the same weight $1/|y|$, whether it belongs to the executed action or to filler thinking tokens. The state distribution is the teacher's, not the student's. At deployment, the policy conditions on its own generations and observations, and errors compound, which is the classic distribution-shift argument of imitation learning, under which a per-step error rate ε can inflate to $O(\varepsilon T^2)$ cost over a T-step horizon unless supervision is collected on the student's own state distribution (Ross et al., 2011).

A.2 Knowledge Distillation

Distillation replaces gold targets with supervision from a stronger teacher (Hinton et al., 2015). For autoregressive models, the token-level form matches full next-token distributions along a sequence,

$$\mathcal{L}_{\mathrm{KD}}(\theta) \;=\; \mathbb{E}_x\,\mathbb{E}_y\,\sum_i D_{\mathrm{KL}}\big(p_T(\cdot \mid x, y_{<i})\,\big\|\,\pi_\theta(\cdot \mid x, y_{<i})\big), \tag{7}$$

where the choice of the sequence y (teacher-generated, gold, or student-generated) and of the divergence defines the family. Sequence-level KD (Kim & Rush, 2016) sidesteps the inner distributions entirely and applies Eq. 6 to teacher-generated text. On-policy distillation evaluates the divergence at prefixes the *student* samples, correcting the train–inference mismatch (Agarwal et al., 2024).

A.3 Black-box Constraint

Every token-level instantiation of Eq. 7 needs two things from the teacher: its full next-token distribution $p_T(\cdot \mid \cdot)$, and a shared tokenizer. The strongest available teachers for web agents are black-box models that usually provide neither. The distillation channel is therefore *only textual*: the teacher writes, and the student trains on the writing via Eq. 6. But text entangles two signals, (a) the teacher's *decisions* (which element to click, what to answer) and (b) its *surface distribution* (style, phrasing, tone). Training a small student on off-policy frontier text imports both at once, and the student pays a significant *perplexity premium* on the teacher's style (Fig. 2).

A.4 Scratchpad Memory

The scratchpad belongs to the *harness* state, not to the environment. It is the sequence of memory components $(y_1^{\texttt{memory}}, \ldots, y_{t-1}^{\texttt{memory}})$ the agent has emitted so far, carried inside the response history $y_{1:t-1}$ that the harness places in every context h_t (§3.1). The agent interacts with it only through

the memory tokens. It *writes* by emitting $y_t^{\texttt{memory}}$ and *reads* only what the harness surfaces back at later steps. Under truncated history, a fact from a page the agent has since left survives only if it was written into a rationale component, most durably into the scratchpad memory.

A.5 Extended Problem Formulation

Details deferred from the Background's POMDP formulation $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, \mathbb{T}, \mathbb{T}_0, \mathbb{O}, R \rangle$ (Kaelbling et al., 1998).

Observation modalities. The observation function $\mathbb{O} : \mathcal{S} \to \mathcal{O}$ may render the browser state as raw HTML, a set-of-marks screenshot (Yang et al., 2023), or an accessibility tree. We use the AXTree throughout, a textual serialization of the rendered page, annotated with the element identifiers the action space refers to.

Action space. $\mathcal{A}$ is a subset of BrowserGym's browser-level action space (de Chezelles et al., 2025): page interactions (e.g. `click`, `fill`), navigation (e.g. `goto`), tab operations (e.g. `tab_focus`), and user interaction (`send_msg_to_user`, which is also how the agent submits its answer). The generated action is deterministically parsed from the `action` component of the response and passed to the browser for execution. The transition $\mathbb{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is treated as deterministic.

Initial states and episodes. $\mathbb{T}_0$ draws the initial state uniformly over the benchmark's tasks. An episode ends when the agent messages the user or exhausts its step budget, at which point the LLM-judge evaluator (Zheng et al., 2023) assigns the terminal $R \in \{0, 1\}$ against the task specification.

Environment versions. The environment comes in multiple versions $\{\mathcal{E}_1, \ldots, \mathcal{E}_n\}$, the same sites captured at different points in time, each defining its own instance of the POMDP tuple. Training and evaluation fix one version, except when cross-version transfer is itself under test.

B Methodology Details

B.1 Teacher Details

Teacher-as-agent. The teacher is prompted with the *same* rendered system and user prompt the student saw: identical goal, AXTree serialization, history format, and action-space description. This keeps the teacher's output in-distribution as an SFT target rather than free-form advice, and makes the edit a drop-in replacement for the student's own response. The teacher receives no privileged input in our method: no gold walkthrough, no hints.

Localization. The teacher makes one call per failed episode over the full logged trajectory. The instruction reads:

```
The agent's trajectory above FAILED. Identify the FIRST
erroneous step -- the earliest step whose action set the
trajectory on the path to failure.  Reply with ONLY the
integer step index (0-based).  Nothing else.
```

Section contract. Every edit query ends with a format contract requiring exactly the five tags `<think>`, `<plan>`, `<step>`, `<memory>`, `<action>`: *think-aloud* reasoning grounded only in what is visible in the observation, the running multi-step plan, a single integer in `<step>` naming the plan step the action belongs to, facts worth carrying forward in `<memory>`, and exactly one executable action. The contract exists because the student is trained and evaluated with the thinking, plan, and memory components enabled. A teacher that omits one of these components would silently teach the student to drop the field.

Style block. The $K{=}3$ exemplars are appended *after* the contract, at the very end of the query. The contract is the instruction whose loss is most expensive (a missing tag costs the whole step), so it must not be pushed out of the recency window, whereas a style hint that is only partly followed

Require: base model π_{θ_0}, black-box teacher π_T (text only), training tasks, exemplar count K, grounding threshold $\rho_{\min}$, weight floor ϕ, dead zone τ_{dz}, anchor strength β
Ensure: $\pi_{\theta_{\text{WD}}}$
1: **P1 (seed):** $\pi_{\theta_{\text{BC}}} \leftarrow$ fine-tune π_{θ_0} on successful teacher rollouts $\mathcal{D}_{\text{T}}$ with $\mathcal{L}_{\text{BC}}$ (Eq. 6); freeze
2: **P2 (rollout):** run $\pi_{\theta_{\text{BC}}}$ on every training task; split episodes into failures $\mathcal{D}_{f(\text{BC})}$ and style pool $\mathcal{D}_{s(\text{BC})}$
3: $\mathcal{D} \leftarrow \emptyset$
4: **for all** $\tau = (h_1, y_1, \ldots, h_T, y_T) \in \mathcal{D}_{f(\text{BC})}$ **do**
5: **P3 (localize):** $t' \leftarrow \text{Loc}(\tau)$, the earliest erroneous step (estimate of $t^\star$, Eq. 8)
6: **Replay:** $\mathcal{D} \leftarrow \mathcal{D} \cup \{(h_k, y_k)\}$ for $k = 1, \ldots, t' - 1$
7: **P4 (edit):** $y_k^* \sim \pi_T(\cdot \mid h_k, \mathcal{D}_{s(\text{BC})})$ with K style exemplars, unexecuted (Eq. 3), for $k = t', \ldots, T$
8: **Filter:** $\mathcal{D} \leftarrow \mathcal{D} \cup \{(h_k, y_k^*) : \rho_k \geq \rho_{\min}\}$
9: **end for**
10: **Weights:** for each edited $(h_k, y_k^*) \in \mathcal{D}$, score segments by Δ_s under $\pi_{\theta_{\text{BC}}}$ (Eq. 4), rank-map to $v_s \in [\phi, 1]$ (Eq. 10), spread to token weights w_i (Eq. 12); pin $\langle\text{A}\rangle, \langle\text{S}\rangle$ tokens at $w_i = 1$; replayed records keep uniform weights
11: **Mask:** $\gamma_i \leftarrow 1$ on $\langle\text{T}\rangle, \langle\text{P}\rangle, \langle\text{M}\rangle$ tokens; $\gamma_i \leftarrow 10^{-6}$ on $\langle\text{A}\rangle, \langle\text{S}\rangle$ tokens (Appendix B.6)
12: **Train:** $\pi_{\theta_{\text{WD}}} \leftarrow \arg\min_\theta \mathcal{L}_{\text{WD}}$ (Eq. 5) over $\mathcal{D}$, initialized at $\pi_{\theta_{\text{BC}}}$ with anchor $\pi_{\text{ref}} = \pi_{\theta_{\text{BC}}}$
13: **return** $\pi_{\theta_{\text{WD}}}$

Algorithm 1 WEBKD: black-box distillation through grounded teacher edits

still leaves a usable record. Exemplar selection is deterministic given the task. Solved episodes are preferred, the task being edited is excluded, and at most one exemplar is drawn per source task. Exemplars carry the rationale components and the step index but never `<action>`.

Correction prompt. The correction query is assembled as the student's rendered context, then the section contract, then the style block. The style block opens with the following instruction:

```
[STYLE -- write this step in the student agent's own style]
The five sections you emit become training data for the
agent whose steps appear above.  Match ITS writing style:
same register, same typical length, same amount of detail,
the same way of phrasing a plan and of noting a memory.
Below are verbatim examples of how THIS agent writes.

Copy the STYLE only.  Do NOT copy their content, their
decisions, or their conclusions -- they are unrelated steps
from other tasks.  Your decision for this step must remain
exactly the one you would have made without these examples;
only the wording changes.
```

It is followed by the K exemplars, each delimited by `how the student writes, example` n and `end example` n. The correction call is made once per step $k \geq t'$, teacher-forced on the student's original prefix.

B.2 P3 FAILURE LOCALIZATION DETAILS

Definition 2 (Recoverability). *For a trajectory τ, let $\nu_t \in \{0, 1\}$ indicate that, from the underlying browser state reached after the first t actions of τ,* some *action sequence within the remaining step budget $T - t$ ends with $R = 1$.*

Proposition 1 (Every failed web trajectory has a unique first fatal step). *Assume the harness of §3.1: a finite step budget T, deterministic transitions and parsing, a terminal binary reward, and a task solvable from its initial state ($\nu_0 = 1$). Then along any failed trajectory ν_t is non-increasing and equals 0 at the episode's end, so*

$$t^\star \;=\; \min\{t \geq 1 : \nu_t = 0\} \tag{8}$$

exists and is unique: every step before $t^\star$ leaves the task solvable, every step from $t^\star$ on leaves it unsolvable, and at step $t^\star$ a recoverability-preserving action existed but was not taken. (Proof and discussion in § F.1.)

The two ways t' can miss $t^\star$ have very different costs. If the teacher localizes too early ($t' < t^\star$), it rewrites steps from which the task was still solvable. The state at t' is still recoverable, so the only cost is extra edits to the corpus. If it localizes too late ($t' > t^\star$), the replayed prefix includes, and so endorses, the fatal steps, and by Proposition 1 no continuation from $h_{t'}$ can succeed. This asymmetry is why the localization prompt asks for the *earliest* erroneous step (§B.1), and it shows up again as the one-sided error $\varepsilon_{\mathrm{loc}}$ in the guarantee of §E.2.

Localization Accuracy. We sample 200 failed episodes from the rollout pools of the three students. Two human annotators, blind to the teacher's prediction and to each other, independently mark the earliest step after which success within the remaining budget became impossible, following the same recoverability instruction given to the teacher. Under the strict budget reading, a step counts as unrecoverable only once the remaining budget is too small to finish the task. Here, the teacher is never late. It is early on 93% of episodes and exact on the remaining 7%. Under the loop-onset reading, the failure point is the step where the agent enters a loop that it never escapes. Here the teacher is early on about two-thirds of episodes, exact on about a quarter, and late on roughly 10%. Under both readings, its errors fall mostly on the early side, which is the benign direction.

B.3 P4 Teacher Edit Details

(i) Style conditioning. The teacher's query ends with a style block containing $K{=}3$ verbatim exemplars of the student's own writing. The exemplars are mined from the style pool $\mathcal{D}_{s(\mathrm{BC})}$, excluding the task being edited, and are selected deterministically per task, with at most one exemplar per source task. The block opens with an explicit instruction to match this agent's register, typical length, and level of detail, and to copy only the style, keeping the decision "exactly the one you would have made without these examples, but only the wording changes" (Appendix B.1). Exemplars include the student's rationale components but *never* its $\langle \mathrm{A} \rangle$ component: an action is a decision, and the style block should transfer the student's style, not its decisions. This directly targets the perplexity premium of §3.2. Apart from the correction itself, the corrected trajectory should read as low-surprise text to the student.

(ii) Edits are never executed. The environment always steps on the student's *original* action. An edit at step k therefore leaves the observation at step $k{+}1$ unchanged, corrections never compound, and every h_k in the corpus is a state the student actually reached.

Each teacher call conditions on the student's original prefix, never on an earlier edit. This mirrors the central idea of on-policy distillation, where the supervising distribution is evaluated on prefixes the student itself produced (Agarwal et al., 2024; Zhao et al., 2026).

B.4 WebKD Loss Details

Remark 1 (Δ_s is a conditional pointwise mutual information)**.** *Writing $c = (h,\, y^* \setminus s)$ for the context with s deleted,*

$$\Delta_s \;=\; \log \frac{P_{\pi_{\theta_{BC}}}(\langle \mathrm{A}^* \rangle \mid s,\, c)}{P_{\pi_{\theta_{BC}}}(\langle \mathrm{A}^* \rangle \mid c)} \;=\; \mathrm{PMI}_{\pi_{\theta_{BC}}}(\langle \mathrm{A}^* \rangle;\, s \mid c)\,, \tag{9}$$

the pointwise mutual information, under the student's own distribution, between the segment and the decision it precedes. Negative Δ_s is meaningful as it marks segments whose presence makes the student less *certain of the correct decision, which is why such segments stay in the ranking and are never masked.*

We use ranks rather than raw magnitudes by default. Because $\log P(\langle \mathrm{A}^* \rangle)$ varies by two orders of magnitude across steps, a magnitude-based weighting would implicitly weight segments by action length. Ranking within each step removes this dependence on scale. Span weights v_s become token weights $w_i \in [\phi, 1]$ by character-coverage average, so a token straddling a segment boundary receives the area-weighted blend (Eq. 12, Appendix B.5).

Scoring policy and the complete map. Segments are sentences or lines of the ⟨T⟩, ⟨P⟩, and ⟨M⟩ components. ⟨S⟩ and ⟨A⟩ are never segmented, and the merge rules and caps are given in Appendix B.5. Both terms of Eq. 4 are scored once, offline, under the *frozen seed checkpoint* $\pi_{\theta_{\text{BC}}}$, the same policy that provides the initialization and, in §3.3.2, the anchor, so the weights describe the student the training starts from, not a moving target. We write $r_s \in [0,1]$ for the normalized average rank of Δ_s within its step (ties share their mean rank), $\sigma = \max_s \Delta_s - \min_s \Delta_s$ for the within-step spread in nats, and $\tau_{\text{dz}} > 0$ for the dead-zone width. Then

$$v_s \;=\; \text{clip}_{[\phi,\,1]}\Big(\phi + (1-\phi)\Big(\tfrac{1}{2} + \big(r_s - \tfrac{1}{2}\big)\,\min\big(1,\;\sigma/\tau_{\text{dz}}\big)\Big)\Big), \tag{10}$$

where $\tau_{\text{dz}} = 0.05$ nats. So a step whose segments are statistically indistinguishable ($\sigma \ll \tau_{\text{dz}}$) degrades to the midpoint weight $(1+\phi)/2$ on every segment, which is the mean over all ranked steps, since average ranks normalized to $[0,1]$ have mean $1/2$. A single-segment step has $\sigma = 0$ and lands at the same midpoint.

The imitation term of the objective is a *weighted mean* of per-token cross-entropies,

$$\mathcal{L}_{\text{SW}} \;=\; \frac{\sum_{i\in T} w_i\,\text{CE}_i}{\sum_{i\in T} w_i}, \tag{11}$$

where T indexes the supervised response tokens and $\text{CE}_i = -\log \pi_\theta(y_i^* \mid h, y_{<i}^*)$ is the cross-entropy of corpus token i. The denominator is the *weight sum*, never the token count. This matters because it makes $w_i \equiv c$ reproduce standard SFT exactly for any $c > 0$, so a scheme whose weights average $\bar{w}$ does not silently train at an effective learning rate of $\bar{w}\cdot\eta$, for learning rate η.

Remark 2 (The update resamples tokens rather than rescaling the loss)**.** *Since $q_w(i) = w_i / \sum_j w_j$ is a probability distribution over supervised positions, $\mathcal{L}_{\text{SW}} = \mathbb{E}_{i\sim q_w}[\text{CE}_i]$, so gradient descent on Eq. 11 is gradient descent on ordinary SFT over a corpus in which tokens are resampled in proportion to their weights.*

B.5 Action-Informative Weighting: Constants and Scoring Interface

Segmentation. Boundaries are newlines, or whitespace following sentence-final punctuation (`.` `!` `?` `;` `:`). Each separator stays with the preceding segment, so segments exactly partition each component's inner span. Segments shorter than 24 non-whitespace characters are merged into the previous segment (into the next, if first). A step is capped at 24 segments by repeatedly merging the shortest adjacent pair. Merges never cross a component boundary.

Scoring interface. Both terms of Eq. 4 are computed by teacher-forced echo-scoring against a served copy of the seed checkpoint: the full (context, target) string is submitted with the prompt echoed and per-token log-probabilities returned, at temperature zero and with no new tokens generated. The log-probability of ⟨A*⟩ is the sum over its token positions. The ablated variant deletes the segment's characters (including its trailing separator) and re-tokenizes. Numbered plan lines are deliberately *not* renumbered after a deletion. Renumbering would perturb segments other than the ablated one and break the leave-one-out contract. The chat template's pre-opened `<think>` tag is stripped before the target is appended, so the scored string matches the training-time rendering.

Characters to tokens. With $\text{chars}(i)$ the character span of token i under the tokenizer's offset mapping, the token weight is the coverage-weighted average

$$w_i \;=\; \frac{\sum_s \big|\text{chars}(i) \cap s\big| \cdot v_s \;+\; \big|\text{chars}(i) \setminus \bigcup_s s\big| \cdot 1}{|\text{chars}(i)|}, \tag{12}$$

clamped to $[\phi, 1]$. Characters outside every rationale segment (tags, separators, and the components pinned at weight 1) contribute weight 1, zero-width special tokens take weight 1, and the response envelope (the template's closing tokens) is never down-weighted.

B.6 Anchor Implementation

Estimator. kl_i in Eq. 5 is the "k_3" estimator (Schulman, 2020): it is non-negative and low-variance, and unbiased for $D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})$ under sampling from π_θ. The log-ratio d_i is clamped

to $[-20, 8]$. The asymmetry is deliberate. $d_i \approx 0$ at initialization and drifts *negative* as the student fits the corpus, where k_3 grows only linearly and is harmless. The positive tail is the direction in which e^d can overflow 16-bit precision, and it is not the direction the penalty exists to control.

Reference pass. π_{ref} is the frozen seed checkpoint, held in 16-bit precision and evaluated in a second, gradient-free forward pass per micro-batch. Per-token log-probabilities are gathered with a fused cross-entropy rather than a full log-softmax, avoiding a $|y| \times |\mathcal{V}|$ 32-bit intermediate. The anchor's marginal memory cost is nonetheless substantial, an *unsharded* 16-bit copy of the reference policy on every rank (measured ≈27 GB at 9B scale) on top of the loss computation's 32-bit logit upcast, which is why training runs single-rank with ZeRO-3 parameter partitioning and CPU-offloaded optimizer state (Appendix C.3).

Why normalize by $\sum_i \gamma_i$. In our corpus, the rationale mask anchors roughly 88% of supervised characters, while an action mask would anchor roughly 9%. Normalized by token count instead, the same β would be an order of magnitude weaker under the action mask because actions are short.

C Additional Experimental Details

C.1 Benchmark Baselines

We describe each baseline of Tab. 1 and Fig. 4 below.

1. **Zero-Shot:** The base model without fine-tuning.
2. **Behavior Cloning (BC):** The base model is trained using supervised fine-tuning (Eq. 6) on the teacher rollouts containing the observation history, prompts, and actions only.
3. **Behavior Cloning (TW):** The TimeWarp version of behavior cloning (Ishmam & Marino, 2026). It is identical to vanilla BC, except that it also includes the thinking, planning, step, and memory tokens during behavior cloning.
4. **m-GRPO**: Multi-turn Group Relative Policy Optimization (m-GRPO) is an outcome-based RL baseline, which uses the trajectory's reward for credit assignment (Wei et al., 2025).
5. **Self/Student Edit**: An edit-style baseline in which the student, instead of an external teacher model, edits its own failed trajectories.
6. **Off-policy Takeover:** An edit-style baseline where, instead of the teacher editing the student's failed trajectories, the teacher takes over from the point of failure and completes the trajectory. The student is then trained *off-policy* on the edited trajectories. This is similar to on-policy expert correction (Lauffer et al., 2026).
7. **Student Re-style Edit:** A two-pass edit, in which the teacher first edits the student's failed trajectories, and the student is then prompted to rewrite the edits in its own style. The student is then trained on-policy on the re-styled edits.

C.2 Loss Ablation Baselines

Each variant keeps the weighted objective of Eq. 11 and changes only the token weights $w_i \in [\phi, 1]$. The $\langle \text{A} \rangle$ and $\langle \text{S} \rangle$ tokens stay at $w_i = 1$ in every variant. Span weights v_s become token weights w_i by the same character-coverage average as in WebKD. The "w/ KL" column of Table 3 adds the anchor of Eq. 5 unchanged.

1. $\mathcal{L}_{\text{CE}}$ (uniform). Standard SFT cross-entropy with equal weight on every response token, the control for all weighted variants.
2. $\mathcal{L}_{\text{GND}}$ (grounding). A soft, within-step version of the grounding filter of Algorithm 1, which floors any rationale span that makes a claim not supported by the goal or by the AXTrees observed up to step k. The claims $\mathcal{C}(s)$ of a span are its quoted strings, numbers, element identifiers, and capitalized entities, and a claim is grounded if its case- and whitespace-folded form occurs in the goal or in one of those AXTrees,

$$v_s = \begin{cases} \phi & \text{if some } c \in \mathcal{C}(s) \text{ is not grounded,} \\ 1 & \text{otherwise.} \end{cases} \tag{13}$$

3. $\mathcal{L}_{\text{CONS}}$ (teacher self-consistency). A span is weighted by how much of it the teacher reproduces when it re-edits the same step $k = 3$ times, net of the overlap expected by chance,

$$v_s \;=\; \phi + (1-\phi)\,\mathrm{clip}_{[0,1]}\!\left(\frac{c_{\text{same}}(s) - c_{\text{null}}(s)}{1 - c_{\text{null}}(s)}\right), \tag{14}$$

where $c_{\text{same}}(s)$ is the IDF-weighted containment of s in the teacher's resampled edits of the same step and $c_{\text{null}}(s)$ is the same statistic against resampled edits of steps from other tasks.

4. $\mathcal{L}_{\text{PPL}}$ (student perplexity). Each token is weighted by how likely the seed student already finds it, so the high-perplexity tokens of the teacher's edit are floored,

$$w_i \;=\; \phi + (1-\phi)\,r_i, \tag{15}$$

where $r_i \in [0,1]$ is the normalized average rank of $\log \pi_{\theta_{\text{BC}}}(y_i^* \mid \cdot)$ among the rationale tokens of its step.

C.3 Training Configuration

The corpus records are step-level: one (context, response) pair per step, the context being the full rendered agent state (goal, AXTree, history) and the response the five-component output. On the corpus used throughout, 15 failed training episodes yield 432 step records. The teacher edits 403 of them (the remaining 29 precede t' and are replayed), the grounding filter of Algorithm 1 removes 40 edited steps (9.9%), and 392 records survive (363 edited, 29 replayed). The edited targets total ≈64k response tokens (16.6k `think`, 19.3k `plan`, 8.7k `memory`, 5.2k `action`, 13.8k structural). Rollouts (P2) use temperature 0.01 and a 30-step budget, with the thinking, plan, and memory components enabled and no hints, the same interface as evaluation. Training is full-parameter fine-tuning from the seed checkpoint with the hyperparameters of Table 4.

Compute. Each fine-tune runs on a single H200 GPU with ZeRO-3 parameter partitioning and CPU-offloaded optimizer state. Corpus generation and evaluation parallelize over a farm of single-GPU workers, one containerized browser environment per GPU, and all teacher and judge calls go through hosted APIs.

Cost and latency. Most of the teacher cost is in P1, which makes $4{,}448$ calls and uses 21.8 M prompt tokens to produce the demonstrations. The correction stage is much cheaper. P3 makes one call per failed episode (15 calls, 0.47 M tokens), and P4 makes one call per edited step (403 calls, 4.0 M tokens, 6% of which is the style block). Correction is therefore 17% of the pipeline's 26.3 M prompt tokens, paid once, or about the cost of 135 teacher episodes at the teacher's in-harness rate of 33k prompt tokens per episode. These counts are lower bounds, since the prompts of 13% of the P1 conversations were not kept and hidden reasoning tokens were not logged (0.78 M visible completion tokens). A fine-tune takes 1.0 GPU-hour on one H200, and evaluating the 103 v6 tasks takes 20 minutes on one 48 GB L40S. Measured inside the harness, the trained 4B student takes a median of 6.10 s per step (38.8 s per episode), against 9.05 s per step for the seed and 12.80 s per step (68.1 s per episode) for the teacher acting as the agent.

Table 4: Training hyperparameters for the combined objective.

Hyperparameter	Value
fine-tuning	full-parameter, from the seed checkpoint
learning rate	1×10^{-5}, cosine decay, warmup ratio 0.1
epochs	2
context cutoff	20,480 tokens
batch	1 per device $\times$ 16-step gradient accumulation (effective 16)
precision	bf16, gradient checkpointing
partitioning	ZeRO-3 with CPU offload
CE weight floor ϕ	0.1
dead zone τ_{dz}	0.05 nats
anchor strength β	0.03 (per anchored token)
anchor mask γ	1 on `think`/`plan`/`memory`, 10^{-6} elsewhere

D Additional Experimental Results

D.1 Error Analysis

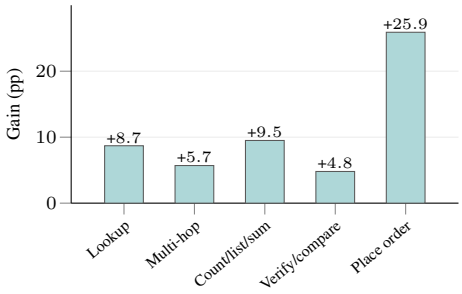


Figure 7: Absolute gain by task type, in percentage points.

Figure 7 breaks the overall improvement down by task type. Gains are positive in all five categories, but their size varies considerably. Placing an order improves the most, rising from 0% to 25.9% (+25.9 pp), and it is the only category where performance started at zero. Counting/listing (+9.5 pp, from 11.9% to 21.4%) and lookup (+8.7 pp) show moderate gains, while multi-hop (+5.7 pp) and verification (+4.8 pp) improve the least. Because each category has only 9–29 examples, these differences should be read with caution: only the gains on placing an order and counting/listing have 95% confidence intervals that exclude zero.

D.2 A Worked Example

This subsection gives the unabridged text of the step shown in Fig. 3.

Student, step 5.

```
<think> We successfully purchased the Welch's item and are
on the thank-you page showing code B644DDC66C. Next, we need
to search for the Mojito mix.  I will click the search box
(bid "15") to bring up the search input.  </think>
<plan> 1.  Search for Mojito mix   - Click search box   -
Type query   - Press Enter   2.  Open first result and
buy   - Click first result link   - Click Buy Now and note
confirmation code   3.  Search for Italian Basil Pesto
[...]    5.  Send all confirmation codes to the user </plan>
<step> 2 </step>
<memory> I am clicking the search box to prepare for
ordering the Mojito mix.  </memory>
<action> click('15') </action>
```

Raw teacher edit.

```
<think> The page shows the order confirmation for Welch's
Orange Pineapple Juice with code B644DDC66C. I need now
search for the second product, so I will navigate back to
the shop home page within the allowed SHOP environment.
</think>
<plan> 1. Record Welch's confirmation code. 2. Navigate
to shop home/search page. 3. Search for Stirrings Simple
Classic Cocktail Mix Mojito. 4. Open matching product,
buy it, and record confirmation code. 5. Navigate to shop
home/search page. 6. Search for Italian Basil Pesto. 7.
Open matching product, buy it, and record confirmation code.
8. Send all confirmation codes to the user. </plan>
<step> 2 </step>
<memory> Welch's Orange Pineapple Juice confirmation code:
B644DDC66C. </memory>
<action> goto('http://localhost:5002/abc') </action>
```

Style-conditioned teacher edit (the WEBKD target), with span weights v_s in brackets.

```
<think> [0.10] The purchase confirmation page for Welch's
Orange Pineapple Juice is displayed, and the code shown is
B644DDC66C. [0.90] I need to continue shopping for the next
product, so I will return to the shop search page. </think>
<plan> [0.80] 1. Record Welch's confirmation code [0.40]
2. Search for Stirrings Simple Classic Cocktail Mix Mojito
- Go to shop page - Type query and press Enter - Open
matching result [1.00] - Click Buy Now and record code
[0.70] 3. Search for Italian Basil Pesto - Go to shop
page - Type query and press Enter - Open matching result
[0.60] - Click Buy Now and record code [0.50] 4. Send all
confirmation codes to the user </plan>
<step> 2 </step>
<memory> [0.30] Welch's Orange Pineapple Juice confirmation
code is B644DDC66C. [0.20] Next I need to search for
Stirrings Simple Classic Cocktail Mix Mojito. </memory>
<action> goto('http://localhost:5002/abc') </action>
```

E THEORETICAL INTUITION AND ANALYSIS

E.1 WHAT THE ANCHORED UPDATE DOES

Write $p_i = \pi_\theta(y_i^* \mid \cdot)$ and $q_i = \pi_{\text{ref}}(y_i^* \mid \cdot)$ for the two policies' probabilities of corpus token y_i^*, $\mathbf{p}^{(i)}$ for the student's full next-token distribution at that position, $\mathbf{e}_{y_i^*}$ for the one-hot target, and collect the two normalized coefficients acting on token i: $a_i = w_i / \sum_j w_j$ (imitation) and $b_i = \beta\gamma_i / \sum_j \gamma_j$ (anchor). Because the pointwise anchor depends on the position-i logits $\mathbf{z}^{(i)}$ only through $\log p_i$, the gradient of Eq. 5 at that position factorizes exactly:

$$\nabla_{\mathbf{z}^{(i)}} \mathcal{L} \;=\; \underbrace{\Big(a_i \;-\; b_i\big(1 - q_i/p_i\big)\Big)}_{\text{throttle } c_i} \cdot \underbrace{\big(\mathbf{p}^{(i)} - \mathbf{e}_{y_i^*}\big)}_{\text{SFT direction}}. \tag{16}$$

Eq. 16 says the anchor never *rotates* the update. Every anchored token still moves along the SFT direction, toward the corpus, and the anchor only rescales it through the data-dependent scalar coefficient c_i, which we call the throttle:

- **At initialization the anchor is inert.** Training starts from π_{ref}, so $p_i = q_i$ and $c_i = a_i$, and the first update is exactly weighted SFT. The penalty acts on drift accumulated during training, never on the starting point.
- **Below the reference, the anchor helps.** If interference drives p_i under q_i (the student losing probability on text its prior found natural), then $q_i/p_i > 1$ and $c_i > a_i$, so the anchor *adds* force toward the corpus token, protecting prior competence rather than fighting the corpus.

- **Above the reference, it throttles and can stop.** As p_i climbs past q_i the throttle falls monotonically toward $a_i - b_i$. When $b_i > a_i$ it reaches zero at a finite probability, which is the content of Proposition 2.
- **On $\langle \mathrm{A} \rangle$ and $\langle \mathrm{S} \rangle$, it is absent.** $\langle \mathrm{A} \rangle$ and $\langle \mathrm{S} \rangle$ carry $\gamma_i \approx 0$, so $b_i \approx 0$ and $c_i = a_i$ throughout, and the teacher's decision is learned at full strength, all the way to the one-hot target.

Proposition 2 (Per-token target of the anchored objective)**.** *Fix a supervised position i, treat its logits as free parameters, and hold the context fixed. The contribution of position i to Eq. 5 is minimized at*

$$p_i^\star \;=\; \begin{cases} \min\left\{1,\;\; q_i \cdot \dfrac{b_i}{b_i - a_i}\right\}, & b_i > a_i, \\ 1, & b_i \le a_i. \end{cases} \tag{17}$$

(Proof in Appendix F.)

That is, the anchor does not shrink the *step* toward the corpus. It moves the *target*, from the one-hot corpus token to a bounded multiple of the reference policy's own probability.

Corollary 1 (Critical anchor strength)**.** *The cap in Eq. 17 exists iff $\beta > \bar{\beta}_i \equiv (w_i/\gamma_i) \cdot \big(\sum_j \gamma_j / \sum_j w_j\big)$. Because $\bar{\beta}_i \propto w_i$, the cap binds* first *on exactly the spans the action-informative weighting marked as least informative about the action.*

The condition $\beta > \bar{\beta}_i$ makes the interior optimum of Eq. 17 finite. That optimum is below 1, and so actually limits the token, only under the stronger condition $q_i\, b_i/(b_i - a_i) < 1$, that is, $b_i(1 - q_i) > a_i$.

E.2 A Single-Round Relabeling Bound

The pipeline admits an imitation-learning guarantee in the style of DAgger (Ross et al., 2011), with the classical action oracle replaced by the two calls a black-box teacher actually answers: localize (P3) and correct (P4). Write $\rho(\pi)$ for the success rate of a policy on the task distribution and $\rho_{\mathrm{BC}} = \rho(\pi_{\theta_{\mathrm{BC}}})$ for the seed. Three error rates describe one correction round, each a population quantity over the seed's failed episodes. $\varepsilon_{\mathrm{loc}} = \Pr[t' > t^\star]$ is the *late*-localization rate of P3 (early misses are benign for the guarantee; they only spend edits). $\varepsilon_{\mathrm{corr}}$ is the teacher's own failure rate when it takes over as the agent from a still-recoverable localized state $h_{t'}$ and plays the episode out. $\bar{\varepsilon}$ is the trained student's per-step disagreement, over the full structured response, with the relabeled process that replays the seed through step $t' - 1$ and then follows the teacher, measured on that process's own history distribution (formal definitions in Appendix F.3).

Theorem 1 (Single-round relabeling bound, informal)**.** *Under the assumptions of Proposition 1, the student $\pi_{\theta_{\mathrm{WD}}}$ trained on the P1–P4 corpus satisfies*

$$1 - \rho(\pi_{\theta_{\mathrm{WD}}}) \;\le\; \underbrace{(1 - \rho_{\mathrm{BC}})\big(\varepsilon_{\mathrm{loc}} + \varepsilon_{\mathrm{corr}}\big)}_{\textit{teacher floor}} + T\,\bar{\varepsilon}, \qquad \bar{\varepsilon} \;\le\; \rho_{\mathrm{BC}}\,\varepsilon_{\mathrm{ret}} + (1 - \rho_{\mathrm{BC}})\big(\varepsilon_{\mathrm{fit}} + \varepsilon_{\mathrm{rec}}\big), \tag{18}$$

where $\varepsilon_{\mathrm{ret}}$ is drift from the seed on episodes the seed already solves (retention), $\varepsilon_{\mathrm{fit}}$ is disagreement with the corpus labels at supervised states, and $\varepsilon_{\mathrm{rec}}$ is disagreement at the post-correction states no corpus record covers. (Formal statement and proof in Appendix F.3.)

Eq. 18 separates what the teacher limits from what the student limits. With a perfect student ($\bar{\varepsilon} = 0$), the remaining failure is the fraction of seed failures that the teacher either localizes too late or cannot repair itself. We estimate both error rates on the 15 seed failures the corpus is built from. Taking over at t', the teacher solves 14 of them. The one it fails was localized at the first step, a state that is recoverable by assumption, so the failure counts against correction rather than localization, giving $\varepsilon_{\mathrm{loc}} = 0/15$ and $\varepsilon_{\mathrm{corr}} = 1/15$ on this pool. The teacher term is then $0.70 \times 0.07 \approx 0.05$, well below the observed student failure rate of 0.61, so the teacher's errors are not the main limit. This is an estimate rather than a direct measurement of the term, because the takeover failure rate comes from training-split states under one judge and the seed failure rate from the test split under another. For reference, the teacher succeeds on 0.71 of test episodes in the same harness (Tab. 1).

The student term $T\bar{\varepsilon}$ is linear in T because $\bar{\varepsilon}$ is evaluated on the history distribution of the relabeled process, which the corpus supervises except at the post-correction states counted in $\varepsilon_{\mathrm{rec}}$. This is

where collecting the corpus on the student's own states enters the bound. The bound concerns terminal failure, whereas the $O(T^2\varepsilon)$ bound for cloning off-policy demonstrations (§A.1) concerns cumulative cost, so the two are not directly comparable.

E.3 Premium Split, Rank Weighting, and the Anchor Budget

This subsection relates the per-token result of Proposition 2 to the premium of Definition 1. Under uniform SFT, the premium determines how the update splits between rationale and action (Proposition 3). The rank map of Eq. 10 orders the update optimally to first order (Theorem 2). The anchor limits how much premium each token can absorb, and the limit is largest on the spans that the weighting marks as decisive (Corollary 2). Proofs are in Appendix F.4.

Where the uniform update goes. Fix a set of corpus steps and score them under the seed $\pi_{\theta_{\mathrm{BC}}}$, the policy every run starts from. Write $\mathcal{I}_{\mathrm{rat}}$ for the positions of the rationale tokens ($\langle \mathrm{T}^* \rangle$, $\langle \mathrm{P}^* \rangle$, $\langle \mathrm{M}^* \rangle$) and $\mathcal{I}_{\mathrm{act}}$ for the positions of the action tokens across those steps, with $n_{\mathrm{rat}} = |\mathcal{I}_{\mathrm{rat}}|$ and $n_{\mathrm{act}} = |\mathcal{I}_{\mathrm{act}}|$. Write $\mathrm{PPL}_{\mathrm{rat}}$ and $\mathrm{PPL}_{\mathrm{act}}$ for the per-token perplexity of Eq. 1 pooled over each set, so that $\log \mathrm{PPL}_C = \frac{1}{n_C} \sum_{i \in \mathcal{I}_C} \mathrm{CE}_i$, and $G_C = \sum_{i \in \mathcal{I}_C} \|\nabla_{\mathbf{z}^{(i)}} \mathrm{CE}_i\|_2$ for the logit-gradient mass that uniform SFT places on set C.

Lemma 1 (Gradient norm is monotone in surprisal)**.** *At every supervised position i, $1 - e^{-\mathrm{CE}_i} \le \|\nabla_{\mathbf{z}^{(i)}} \mathrm{CE}_i\|_2 \le \sqrt{2}\,(1 - e^{-\mathrm{CE}_i})$.*

Assumption 1 (Bounded surprisal)**.** *Every per-token surprisal is bounded, $\mathrm{CE}_i \le \Lambda$ for some $\Lambda < \infty$.*

Proposition 3 (The premium fixes the split of the uniform update)**.** *(i) Restricted to $\mathcal{I}_{\mathrm{rat}} \cup \mathcal{I}_{\mathrm{act}}$, the share of $\mathcal{L}_{\mathrm{BC}}$ that the action tokens contribute at initialization is exactly*

$$\frac{n_{\mathrm{act}} \log \mathrm{PPL}_{\mathrm{act}}}{n_{\mathrm{act}} \log \mathrm{PPL}_{\mathrm{act}} + n_{\mathrm{rat}} \log \mathrm{PPL}_{\mathrm{rat}}}. \tag{19}$$

(ii) Under Assumption 1, the action's share of the logit-gradient mass satisfies

$$\frac{G_{\mathrm{act}}}{G_{\mathrm{act}} + G_{\mathrm{rat}}} \le \frac{X}{X + Y}, \qquad X = \sqrt{2}\, n_{\mathrm{act}} \log \mathrm{PPL}_{\mathrm{act}}, \qquad Y = \frac{1 - e^{-\Lambda}}{\Lambda}\, n_{\mathrm{rat}} \log \mathrm{PPL}_{\mathrm{rat}}. \tag{20}$$

Why the rank map is the right order. Proposition 3 says how much of the update lands on the rationale. The weighting of §3.3.1 decides how that part is spent across segments. Fix a corpus step with corrected action $\langle \mathrm{A}^* \rangle$ and rationale segments $s_1, \ldots, s_n$ (Appendix B.5), and write Δ_j for the score of Eq. 4 on segment s_j.

Assumption 2 (Additive twin)**.** *For every subset $\mathcal{U} \subseteq \{1, \ldots, n\}$ of segments present in the rationale, the student's log-probability of the corrected action is $\log P_\pi(\langle \mathrm{A}^* \rangle \mid h, \mathcal{U}) = \alpha_0 + \sum_{j \in \mathcal{U}} \Delta_j$ for some constant α_0, so the leave-one-out score of Eq. 4 recovers Δ_j exactly.*

Assumption 3 (Independent emission)**.** *At deployment the student writes segment s_j into its own rationale independently with probability $\kappa_j \in (0, 1)$, and raising $\log \kappa_j$ by $\mathrm{d}u$ nats costs $\mathrm{d}u$ nats of premium on that segment (Definition 1).*

Theorem 2 (Rank weighting is first-order optimal credit assignment)**.** *Under Assumptions 2 and 3, the deployment probability of the corrected action is $P_\pi(\langle \mathrm{A}^* \rangle \mid h) = e^{\alpha_0} \prod_{j=1}^{n} \left(1 - \kappa_j + \kappa_j e^{\Delta_j}\right)$, and the gain in $\log P_\pi(\langle \mathrm{A}^* \rangle \mid h)$ per nat of premium absorbed on segment s_j is*

$$\mu_j = \frac{\partial \log P_\pi(\langle \mathrm{A}^* \rangle \mid h)}{\partial \log \kappa_j} = \frac{\kappa_j \left(e^{\Delta_j} - 1\right)}{1 - \kappa_j + \kappa_j e^{\Delta_j}}, \tag{21}$$

which is strictly increasing in Δ_j, zero at $\Delta_j = 0$, and negative for $\Delta_j < 0$. If moreover $\kappa_j = \kappa$ for all segments of the step and a step on $\mathcal{L}_{\mathrm{SW}}$ sheds premium on segment s_j at a rate proportional to its weight v_j (Remark 2), then for any fixed multiset of segment weights $\{v_1, \ldots, v_n\}$ the first-order gain $\sum_j \mu_j v_j$ is maximized by assigning the weights in the order of Δ_j, and the maximizing assignment depends on the Δ_j only through that order.

Remark 3 (What the twin predicts)**.** *The twin makes two predictions. First, segments with $\Delta_j < 0$ have $\mu_j < 0$, so learning them lowers the deployment probability of the corrected action. The weighting of §B.4 floors them at ϕ rather than masking them, and 42% of segments fall in this class. The twin predicts that masking them costs nothing and may help. Second, the gain of $\mathcal{L}_{WD}$ over $\mathcal{L}_{CE}$ in Table 3 should grow with the within-step spread σ of Δ_j, since a step with $\sigma \approx 0$ has nothing to reorder, which is what the dead zone of Eq. 10 already assumes.*

The anchor as a premium budget. Proposition 2 gives the per-token target of the anchored objective. Read in the units of Definition 1, it is a budget.

Corollary 2 (Per-token premium budget)**.** *Under the anchored objective of Eq. 5, the premium the student can shed on a rationale token i, its CE_i at initialization minus its CE_i at the per-token optimum $p_i^\star$ of Eq. 17, satisfies*

$$\log \frac{p_i^\star}{q_i} \;\le\; -\log\Big(1 - \frac{\bar\beta_i}{\beta}\Big) \quad \textit{whenever } \beta > \bar\beta_i, \qquad \bar\beta_i = \frac{w_i}{\gamma_i} \cdot \frac{\sum_j \gamma_j}{\sum_j w_j}, \tag{22}$$

with equality when the interior cap $q_i b_i/(b_i - a_i)$ is at most 1 (when it exceeds 1 the token is fit to $p_i^\star = 1$ and sheds its full CE_i, which is below the right-hand side), and is the full CE_i at initialization (the token may be fit to $p_i^\star = 1$) when $\beta \le \bar\beta_i$. The budget decreases in β and increases in w_i, so at a fixed β the spans the weighting marks as decisive receive the largest budget.

F Proofs and Theoretical Remarks

F.1 The first fatal step: proof of Proposition 1

Fix a task and a failed trajectory τ executed under the harness of §3.1. Write $s_{(t)}$ for the underlying browser state after the first t actions of τ, with $s_{(0)}$ the initial state, and recall the recoverability indicator ν_t of Definition 2. The assumptions of Proposition 1, stated in full:

Assumption 4 (Finite budget)**.** *The harness enforces a step budget T, and an episode that issues a terminal action earlier enters an absorbing state with its reward fixed.*

Assumption 5 (Determinism)**.** *$\mathbb{T}$, $\mathbb{O}$, and f_{parse} are deterministic, so the state reached by a given action sequence from $s_{(0)}$ is unique and replayable.*

Assumption 6 (Terminal binary reward)**.** *$R \in \{0, 1\}$ is assigned once, when the episode ends.*

Assumption 7 (Feasibility)**.** *$\nu_0 = 1$: the task is solvable from its initial state within the budget.*

Lemma 2 (Recoverability is absorbing)**.** *Under Assumptions 4–6, $\nu_t = 0$ implies $\nu_{t+1} = 0$ along τ.*

Proof. Suppose $\nu_{t+1} = 1$: some action sequence $(a'_1, \dots, a'_m)$ with $m \le T - (t+1)$ reaches $R = 1$ from $s_{(t+1)}$. Since $s_{(t+1)} = \mathbb{T}(s_{(t)}, a_{t+1})$ deterministically, the sequence $(a_{t+1}, a'_1, \dots, a'_m)$ has length $m + 1 \le T - t$ and reaches $R = 1$ from $s_{(t)}$, so $\nu_t = 1$. The contrapositive gives the claim. □

Proof of Proposition 1. $\nu_0 = 1$ by Assumption 7. At the episode's end $\nu = 0$. If the budget is exhausted, the only continuation within budget 0 is the empty one, and it ends with the episode's own reward $R(\tau) = 0$. If a terminal action ended the episode early, the state is absorbing with $R = 0$ (Assumption 4). By Lemma 2 the sequence $\nu_0, \nu_1, \dots$ is non-increasing over $\{0, 1\}$, so it crosses exactly once, and $t^\star = \min\{t \ge 1 : \nu_t = 0\}$ exists and is unique, with $\nu_t = 1$ for all $t < t^\star$ and $\nu_t = 0$ for all $t \ge t^\star$. For the final clause, $\nu_{t^\star - 1} = 1$ supplies a within-budget success sequence from $s_{(t^\star - 1)}$. Its first action a leaves the task solvable in the remaining budget, so a recoverability-preserving action existed at step $t^\star$, and the executed $a_{t^\star}$ was not one, since it produced $\nu_{t^\star} = 0$. □

Remarks. *(i) The role of the budget.* Recoverability is defined *within the remaining budget*, so even a diffuse failure with no single wrong click, an agent that dithers, loops, or re-reads pages, has a unique $t^\star$: the first step after which the shortest completion no longer fits in the budget. Without

Assumption 4 the proposition is false, since an agent on an unbounded horizon can wander indefinitely without failure ever becoming irreversible. *(ii) What $t^\star$ is and is not.* $t^\star$ is an interventional quantity, the earliest step after which *no* policy could still succeed, which is what the localization instruction of Appendix B.1 asks for. By Assumption 5 it is a property of the executed actions and the environment alone: attainability depends only on the browser state and the remaining budget, so rationale tokens influence which action a policy *will* take but never whether success is still attainable. It need not coincide with the causal "root cause" a human would narrate (a misread page three steps earlier), which is why the localization study of §B.2 anchors annotators to the recoverability definition rather than to blame. *(iii) Live sites.* Assumption 5 holds exactly in a containerized benchmark and only approximately on the live web, where transitions are stochastic. There ν_t generalizes to the optimal success probability from $s_{(t)}$, monotonicity can fail (a lucky refresh can restore a lost option), and $t^\star$ must be defined relative to a threshold on ν_t. Nothing in the paper relies on this extension. *(iv) Infeasible tasks.* If $\nu_0 = 0$, the task is unsolvable from the start, $t^\star$ is undefined, and no correction can lead to success. Assumption 7 is thus a benchmark-design requirement, witnessed per task by any successful demonstration.

F.2 Proof of Proposition 2

Restrict Eq. 5 to a single supervised position i and write p for $p_i = \pi_\theta(y_i^* \mid \cdot)$, suppressing the index. Substituting $e^{d_i} = q_i/p$ into $\mathrm{kl}_i = e^{d_i} - d_i - 1$, position i contributes

$$L_i(p) \;=\; -\,a_i \log p \;+\; b_i \left(\frac{q_i}{p} + \log \frac{p}{q_i} - 1 \right), \tag{23}$$

with $a_i = w_i / \sum_j w_j$ and $b_i = \beta\gamma_i / \sum_j \gamma_j$ as in Appendix E. Differentiating,

$$\frac{dL_i}{dp} \;=\; -\frac{a_i}{p} + b_i \left(-\frac{q_i}{p^2} + \frac{1}{p} \right) \;=\; -\,\frac{c_i(p)}{p}, \qquad c_i(p) \;=\; a_i - b_i \left(1 - \frac{q_i}{p} \right), \tag{24}$$

recovering the throttle of Eq. 16. For $b_i > 0$ the function $c_i(p)$ is strictly decreasing in p, with $\lim_{p \to 0^+} c_i = +\infty$ and $c_i \to a_i - b_i$ as $p \to \infty$, and for $b_i = 0$ it is the constant a_i. If $b_i \le a_i$, then $c_i(p) = (a_i - b_i) + b_i q_i / p > 0$ for every $p \in (0, 1]$ (here $a_i > 0$, since $w_i \ge \phi > 0$), so L_i is strictly decreasing on $(0, 1]$ and the minimum sits at the boundary $p = 1$. If $b_i > a_i$, solving $c_i(p) = 0$ gives the unique interior stationary point

$$p_i^\star \;=\; \frac{q_i\, b_i}{b_i - a_i}, \tag{25}$$

with $c_i > 0$ below it and $c_i < 0$ above it, so it is the minimum. Clipping at the boundary yields Eq. 17. For the full-softmax statement, note $\nabla_{\mathbf{z}}(-\log p) = \mathbf{p} - \mathbf{e}_y$ and $\nabla_{\mathbf{z}} \log p = \mathbf{e}_y - \mathbf{p}$, while $\partial \mathrm{kl}_i / \partial \log p = 1 - e^{d_i} = 1 - q_i/p$. Combining gives Eq. 16, which vanishes iff $c_i = 0$ or $\mathbf{p} = \mathbf{e}_y$. Under gradient flow from the initialization $\mathbf{p} = \mathbf{q}$ (the reference's full distribution, since $\theta_0 = \theta_{\text{ref}}$), every logit moves along the SFT direction scaled by c_i. The target logit rises at rate $c_i(1 - p)$ and each off-target logit z_v falls at rate $c_i\, p_v$, in proportion to its current probability, so the gap between two off-target logits $z_u - z_v$ moves at rate $c_i(p_v - p_u)$ and the more probable alternatives lose ground faster. The off-target distribution therefore flattens rather than keeping the reference's shape, while p rises toward $p_i^\star$, where c_i vanishes when $p_i^\star$ is interior. □

Two scope notes. First, the closed form exists *because* the anchor is pointwise in p: a full-distribution KL to π_{ref} would couple every vocabulary coordinate and add a second, non-collinear term to Eq. 16. Second, the tabular idealization treats each position's logits as free and ignores parameter sharing across positions. It characterizes the force each token contributes, not the coupled optimum of the network.

F.3 A single-round imitation bound: proof of Theorem 1

P1 is behavior cloning on teacher rollouts, whose error can compound quadratically in the horizon (Ross & Bagnell, 2010). P2–P4 implement the oracle that DAgger uses to make it linear (Ross et al., 2011), expert-quality labels at states drawn from $d_{\pi_{\theta_{\text{BC}}}}$, but in a setting the classical theory does not cover: the expert is black-box, answers only in *text*, runs for a single round rather than DAgger's iterated no-regret scheme, and its labels can fail in ways an action oracle's cannot (a mislocalized

failure, a fluent reference to state that was never observed). This subsection accounts for these differences. Throughout, Assumptions 4–7 hold, R is treated as ground truth (the LLM judge that implements it adds measurement noise orthogonal to this analysis), and success rates $\rho(\cdot)$ average over the training-task distribution and any policy stochasticity.

The relabeled process. The comparator is the process $\tilde{\pi}$ that the corpus of P2–P4 supervises. On a task whose seed rollout succeeded, $\tilde{\pi}$ is that rollout: $\tilde{\pi} = \pi_{\theta_{\mathrm{BC}}}$ and nothing is edited. On a task whose seed rollout failed with localization t', $\tilde{\pi}$ replays the logged responses through step $t'-1$ and then hands control to the teacher-as-agent of Appendix B.1, style-conditioned exactly as in P4, which plays the episode out closed-loop from $h_{t'}$. The replay is well defined because P2 rolls out at temperature 0.01 (Appendix C.3), so the seed's trajectory on a task is effectively unique and the relabeled process is a policy of the task.

Error terms. All are population quantities, conditioned on failed seed rollouts where indicated:

$$\begin{aligned}
\varepsilon_{\mathrm{loc}} &= \Pr\big[\, t' > t^\star \,\big|\, \text{seed rollout failed} \,\big], && \text{late localization (P3),}\\
\varepsilon_{\mathrm{corr}} &= \Pr\big[\, \text{teacher-as-agent fails from } h_{t'} \,\big|\, \text{failed},\ t' \le t^\star \,\big], && \text{corrector capacity (P4),}\\
\delta_t &= \mathbb{E}_{h\sim d^t_{\tilde{\pi}}}\Big[\mathrm{TV}\big(\pi_{\theta_{\mathrm{WD}}}(\cdot \mid h),\, \tilde{\pi}(\cdot \mid h)\big)\Big], && \text{student disagreement,}
\end{aligned}$$

where $d^t_{\tilde{\pi}}$ is the step-t history distribution of the relabeled process and the total variation is over the full structured response y. Remark iii explains why the action alone does not suffice.

Lemma 3 (Imitation on a deterministic harness)**.** *For any two policies $\pi, \tilde{\pi}$ on the harness of §3.1,* $\rho(\tilde{\pi}) - \rho(\pi) \;\le\; \sum_{t=1}^{T} \mathbb{E}_{h\sim d^t_{\tilde{\pi}}}\big[\mathrm{TV}\big(\pi(\cdot \mid h),\, \tilde{\pi}(\cdot \mid h)\big)\big]$.

Proof. Couple the two processes stepwise, drawing the pair of responses at every shared history h from a maximal coupling of $\pi(\cdot \mid h)$ and $\tilde{\pi}(\cdot \mid h)$, under which the responses differ with probability $\mathrm{TV}(\pi(\cdot \mid h), \tilde{\pi}(\cdot \mid h))$. By Assumption 5 the two trajectories are identical until the first step at which the coupled responses differ, and if no such step occurs, the trajectories and hence the terminal rewards coincide. A union bound over the step of first difference, evaluated along $\tilde{\pi}$'s distribution, which the coupled process follows until that step, gives $\Pr[\text{trajectories differ}] \le \sum_t \mathbb{E}_{h\sim d^t_{\tilde{\pi}}}[\mathrm{TV}]$, and $\rho(\tilde{\pi}) - \rho(\pi) \le \Pr[\text{trajectories differ}]$. □

Theorem 3 (Single-round relabeling bound, formal)**.** *Under Assumptions 4–7,*

$$1 - \rho(\pi_{\theta_{\mathrm{WD}}}) \;\le\; (1-\rho_{\mathrm{BC}})\big(\varepsilon_{\mathrm{loc}} + \varepsilon_{\mathrm{corr}}\big) \;+\; \sum_{t=1}^{T} \delta_t. \tag{26}$$

Proof. First bound the relabeled process. On tasks whose seed rollout succeeds, probability ρ_{BC}, $\tilde{\pi}$ reproduces that rollout and succeeds, by Assumption 5. On a failed rollout, distinguish the localization outcomes. If $t' > t^\star$, then $t'-1 \ge t^\star$, so $\nu_{t'-1} = 0$ by Proposition 1 and no continuation from $h_{t'}$ within the remaining budget can succeed: the episode is lost regardless of the teacher. If $t' \le t^\star$, then $\nu_{t'-1} = 1$, the state is recoverable, and the teacher-as-agent fails with probability $\varepsilon_{\mathrm{corr}}$ by definition. Hence

$$1 - \rho(\tilde{\pi}) \;\le\; (1-\rho_{\mathrm{BC}})\big(\varepsilon_{\mathrm{loc}} + (1-\varepsilon_{\mathrm{loc}})\,\varepsilon_{\mathrm{corr}}\big) \;\le\; (1-\rho_{\mathrm{BC}})\big(\varepsilon_{\mathrm{loc}} + \varepsilon_{\mathrm{corr}}\big).$$

Apply Lemma 3 with $\pi = \pi_{\theta_{\mathrm{WD}}}$ and combine. □

Corollary 3 (Regime decomposition)**.** *Partition the states of $d^t_{\tilde{\pi}}$ into three cells. The first holds states of solved seed episodes, where $\tilde{\pi} = \pi_{\theta_{BC}}$ and no record supervises. The second holds states of failed episodes up to and including t', where the corpus supervises exactly the histories $\tilde{\pi}$ visits (replay records before t', the edited record at t'). The third holds post-correction states $t > t'$ of failed episodes, which no record covers. Writing $\varepsilon_{\mathrm{ret}}$, $\varepsilon_{\mathrm{fit}}$, $\varepsilon_{\mathrm{rec}}$ for the per-step average of $\mathrm{TV}(\pi_{\theta_{\mathrm{WD}}}, \tilde{\pi})$ on each cell respectively,*

$$\sum_{t=1}^{T} \delta_t \;\le\; T\Big(\rho_{\mathrm{BC}}\,\varepsilon_{\mathrm{ret}} \;+\; (1-\rho_{\mathrm{BC}})\big(\varepsilon_{\mathrm{fit}} + \varepsilon_{\mathrm{rec}}\big)\Big), \tag{27}$$

since each episode contributes at most T steps to its cells. With $\bar{\varepsilon}$ the bracketed mixture, this is Eq. 18.

Remarks. *(i) The DAgger correspondence.* $\varepsilon_{\mathrm{fit}}$ plays the role of DAgger's training loss on the aggregated on-policy dataset, and $T\bar{\varepsilon}$ its linear-in-horizon error term (Ross et al., 2011). The teacher floor can be measured directly. A takeover run, in which the teacher plays the episode out from $h_{t'}$, reports the bracket $\varepsilon_{\mathrm{loc}} + (1-\varepsilon_{\mathrm{loc}})\,\varepsilon_{\mathrm{corr}}$ of the proof directly, since a late localization fails with probability one, so $(1-\rho_{\mathrm{BC}})$ times the measured takeover failure rate is a tighter floor than the $\varepsilon_{\mathrm{loc}} + \varepsilon_{\mathrm{corr}}$ that Eq. 26 prints and needs no separate estimate of $\varepsilon_{\mathrm{loc}}$. *(ii) Teacher capacity is the floor, and the degradation is graceful.* A weaker teacher enters the bound three times, through ρ_{BC} (worse P1 demonstrations, themselves subject to the off-policy $O(T^2\varepsilon)$ regime above), through $\varepsilon_{\mathrm{loc}}$, and through $\varepsilon_{\mathrm{corr}}$, each linearly. The bound is one-directional: it caps failure and does not preclude the student empirically exceeding the teacher-in-harness, but any such gain is outside the guarantee. *(iii) Why the disagreement is over the full response.* h_{t+1} contains y_t, so two policies that always agree on the executed action but phrase their rationale differently still diverge in history, and the coupling of Lemma 3 breaks. This is the truncated-history argument of Appendix A.4 as it appears in the bound. Rationale tokens are part of the harness state, which is why the method supervises and anchors them. Action-level agreement suffices exactly when rationale differences never change a later decision, a policy-dependent property the anchor pushes toward but does not certify. *(iv) The single-round residual.* $\varepsilon_{\mathrm{rec}}$ lives on the states the teacher's correction steers into, which no record covers by design, since edits never execute (P4ii). Two mechanisms keep it small: recovered states resemble the teacher-demonstration states P1 already cloned, and the anchor holds $\pi_{\theta_{\mathrm{WD}}}$ near $\pi_{\theta_{\mathrm{BC}}}$ wherever the corpus is silent. Iterating P2–P4 from $\pi_{\theta_{\mathrm{WD}}}$ would convert $\varepsilon_{\mathrm{rec}}$ into supervised coverage, which is DAgger's aggregation. Likewise $\varepsilon_{\mathrm{ret}}$ is the retention term the anchor of §3.3.2 targets: on solved episodes the reference behavior is the seed's own, so any drift there is pure loss. Because the anchor is a pointwise, role-masked penalty on corpus tokens (§3.3.2), it lowers $\varepsilon_{\mathrm{ret}}$ and $\varepsilon_{\mathrm{rec}}$ by limiting drift, but it does not bound either through a distributional inequality. *(v) What the deep-suffix records do.* Edited records at $k > t'$ sit at seed-visited states that are unrecoverable whenever $t' = t^\star$ (Proposition 1), so they cannot rescue those episodes and do not appear in Eq. 26. They teach the policy how to behave at bad states and influence the bound only through generalization, that is, through lowering $\varepsilon_{\mathrm{fit}}$ and $\varepsilon_{\mathrm{rec}}$ at nearby states. The grounding filter of Algorithm 1 removes a kind of label error that an action oracle cannot make, fluent references to state that was never observed, which would otherwise corrupt these records.

F.4 Proofs for §E.3

Proof of Lemma 1. Write $\mathbf{p} = \mathbf{p}^{(i)}$ for the student's next-token distribution at position i, $y = y_i^*$ for the target, and $p_y = e^{-\mathrm{CE}_i}$ for its probability. Since $\mathrm{CE}_i = -\log \mathrm{softmax}(\mathbf{z}^{(i)})_y$, the gradient is $\nabla_{\mathbf{z}^{(i)}} \mathrm{CE}_i = \mathbf{p} - \mathbf{e}_y$, so $\|\nabla_{\mathbf{z}^{(i)}} \mathrm{CE}_i\|_2^2 = (1-p_y)^2 + \sum_{v\neq y} p_v^2$. The sum is non-negative, which gives the lower bound, and $\sum_{v\neq y} p_v^2 \le \big(\sum_{v\neq y} p_v\big)^2 = (1-p_y)^2$ because every $p_v \ge 0$, which gives the upper bound. □

Proof of Proposition 3. (i) By Eq. 1 pooled over a set C, $\log \mathrm{PPL}_C = \frac{1}{n_C}\sum_{i\in\mathcal{I}_C} \mathrm{CE}_i$, so $\sum_{i\in\mathcal{I}_C} \mathrm{CE}_i = n_C \log \mathrm{PPL}_C$. The uniform loss restricted to $\mathcal{I}_{\mathrm{rat}} \cup \mathcal{I}_{\mathrm{act}}$ is $\frac{1}{n_{\mathrm{rat}}+n_{\mathrm{act}}}$ times the sum of the two, and the action's share is the ratio in Eq. 19. (ii) By Lemma 1 and $1-e^{-x} \le x$ for $x \ge 0$, $G_{\mathrm{act}} \le \sqrt{2}\sum_{i\in\mathcal{I}_{\mathrm{act}}}(1-e^{-\mathrm{CE}_i}) \le \sqrt{2}\sum_{i\in\mathcal{I}_{\mathrm{act}}} \mathrm{CE}_i = X$. The function $x \mapsto 1-e^{-x}$ is concave, so on $[0,\Lambda]$ it lies above its chord, $1-e^{-x} \ge (1-e^{-\Lambda})\,x/\Lambda$. With Assumption 1 and Lemma 1, $G_{\mathrm{rat}} \ge \sum_{i\in\mathcal{I}_{\mathrm{rat}}}(1-e^{-\mathrm{CE}_i}) \ge \frac{1-e^{-\Lambda}}{\Lambda}\sum_{i\in\mathcal{I}_{\mathrm{rat}}} \mathrm{CE}_i = Y$. Finally $t \mapsto t/(t+s)$ is increasing in t and decreasing in s, so $G_{\mathrm{act}}/(G_{\mathrm{act}}+G_{\mathrm{rat}}) \le G_{\mathrm{act}}/(G_{\mathrm{act}}+Y) \le X/(X+Y)$. □

Scope. The proposition is about gradients with respect to the logits, the same idealization as Proposition 2.

Proof of Theorem 2. Under Assumption 3 the set $\mathcal{U}$ of segments the student writes has $\Pr[\mathcal{U}] = \prod_{j\in\mathcal{U}} \kappa_j \prod_{j\notin\mathcal{U}}(1-\kappa_j)$, and under Assumption 2

$$P_\pi(\langle \mathrm{A}^* \rangle \mid h) = \sum_{\mathcal{U}} \Pr[\mathcal{U}]\, e^{\alpha_0} \prod_{j\in\mathcal{U}} e^{\Delta_j} = e^{\alpha_0} \prod_{j=1}^{n} \big(1-\kappa_j+\kappa_j e^{\Delta_j}\big),$$

since the sum over subsets factorizes coordinate by coordinate. Taking logarithms and differentiating the j-th factor with respect to $\log \kappa_j$ gives Eq. 21. Writing $a = e^{\Delta_j}$, $\mu_j = \kappa_j(a-1)/(1-\kappa_j+\kappa_j a)$ has derivative in a equal to $\left[\kappa_j(1-\kappa_j+\kappa_j a) - \kappa_j^2(a-1)\right]$ over the square of the denominator, which simplifies to $\kappa_j/(1-\kappa_j+\kappa_j a)^2 > 0$, so μ_j is strictly increasing in Δ_j. It vanishes at $a = 1$ and is negative for $a < 1$.

For the ordering claim, the rate hypothesis says a step on $\mathcal{L}_{\mathrm{SW}}$ raises $\log \kappa_j$ by $\eta\, v_j$ for a common $\eta > 0$. To first order the change in $\log P_\pi(\langle \mathrm{A}^* \rangle \mid h)$ is then $\eta \sum_j \mu_j v_j$ by Eq. 21. With $\kappa_j = \kappa$ for all j, μ_j is an increasing function of Δ_j alone, so the μ_j and the Δ_j have the same order. The rearrangement inequality states that for two finite real sequences the sum of products is maximized when both are sorted the same way, so among all assignments of a fixed multiset $\{v_j\}$ to the segments, $\sum_j \mu_j v_j$ is maximized by assigning the weights in the order of Δ_j. The maximizing assignment is a function of the order of the Δ_j alone, which is what Eq. 10 computes, up to ties, which the average rank resolves without changing the maximum. □

Scope. Assumption 2 makes the leave-one-out score exact and removes interactions among segments, so a segment that matters only jointly with another is scored as inert. Assumption 3 treats the segments as independently emitted and prices a nat of log-probability on a segment as a nat of premium, which is Definition 1 applied to that segment. The rate hypothesis holds in logit space when the rationale tokens of a step have equal gradient norm at initialization. A logit-space step of size η on $\mathcal{L}_{\mathrm{SW}}$ at a token i with weight w_i moves the logits by $-\eta\, w_i(\mathbf{p} - \mathbf{e}_y)$ up to the common normalizer $\sum_k w_k$, and since $\nabla_{\mathbf{z}^{(i)}} \log p_i = \mathbf{e}_y - \mathbf{p}$, it raises $\log p_i$ by $\eta\, w_i \|\mathbf{p} - \mathbf{e}_y\|^2$ to first order. Equal surprisal across the step's rationale tokens makes these squared norms equal within a factor of two by Lemma 1, and equal when the off-target mass is spread alike. Without equal naturalness the first-order optimal order is by μ_j, which is increasing in both Δ_j and κ_j.

Proof of Corollary 2. By Proposition 2, if $b_i > a_i$ the per-token optimum is $p_i^\star = \min\{1, q_i\, b_i/(b_i - a_i)\}$, so $\log(p_i^\star/q_i) \le \log\big(b_i/(b_i-a_i)\big) = -\log(1 - a_i/b_i)$. With $a_i = w_i/\sum_j w_j$ and $b_i = \beta\gamma_i/\sum_j \gamma_j$, $a_i/b_i = \frac{w_i}{\gamma_i} \cdot \frac{\sum_j \gamma_j}{\sum_j w_j} \cdot \frac{1}{\beta} = \bar{\beta}_i/\beta$, which is Eq. 22, and $b_i > a_i$ is $\beta > \bar{\beta}_i$ (Corollary 1). Training starts at $p_i = q_i$ (§3.3.2), so the surprisal shed on token i between initialization and the per-token optimum is $\log p_i^\star - \log q_i$. If $b_i \le a_i$ the optimum is $p_i^\star = 1$ and the shed surprisal is the full CE_i at initialization. The budget $-\log(1 - \bar{\beta}_i/\beta)$ is decreasing in β and increasing in $\bar{\beta}_i \propto w_i$. On tokens with $\gamma_i \approx 0$, $b_i \approx 0 < a_i$ and there is no cap. □

Declaration of LLM Usage

The authors acknowledge the use of LLMs to polish the draft, to brainstorm ideas and research directions, to orchestrate experiments, to verify every proof in Lean, and to review code. We take full responsibility for the paper's content and correctness.

Reproducibility Statement

We have taken several steps to ensure our results are reproducible. The complete source code is included in the supplementary material and will be released publicly upon acceptance. It comes with a README describing environment setup, dependencies, and each component. All proofs are formalized in Lean 4, and the formalization is included in the supplementary material, so every theorem can be machine-checked. All datasets are publicly available and have been cited.