WHEN DOES CONTINUATION-CONDITIONED SUPERVISION IMPROVE TOOL-USING AGENTS?

Anonymous authors
Paper under double-blind review

ABSTRACT

Counterfactual supervision values an action by executing the remainder of a task with a chosen controller. Even an exact return can favor an action that is poorly suited to a different continuation. We study this effect in a controlled tool-use simulator, holding checkpoints, candidate actions, and the current-action risk rule fixed. We derive sufficient conditions under which continuation changes preserve the learner-visible optimal action, accounting for fitting error. Fixed-checkpoint diagnostics show that reference-conditioned selection is strictly suboptimal under a frozen deployment continuation at 16.8% of 800 checkpoints. We then evaluate newly fitted policies on held-out complete tasks, keeping training states and fitting procedures fixed across targets. Relative to reference supervision, fully deployment-conditioned targets yield mean paired utility gains of 0.076 for a cost-sensitive classifier and 0.099 for a multilayer perceptron. Both learner classes gain 3.0 success percentage points on average over three seeds, with improvements in utility and success in all six matched comparisons. These findings show that continuation choice can affect both the actions favored by supervision and the complete-task performance of policies learned from it.

1 INTRODUCTION

A tool-using agent must decide when to act, seek information, verify, delegate, or escalate. Each decision changes the state from which the task must be completed. Counterfactual supervision can value a candidate action by restoring a checkpoint, forcing that action, and executing the remaining trajectory. Even an exact return answers a controller-specific question: how well does this action work when a particular policy handles the rest of the task? A label computed with a reference controller can therefore favor a different action from one computed with the controller intended for deployment.

A recorded clarification checkpoint illustrates this distinction. With four decisions remaining, reference-conditioned selection chooses ACT, while deployment-conditioned selection chooses VERIFY. When both choices are followed by the same frozen deployment controller, their task utilities are 2.918 and 2.939, respectively; both tasks succeed (Appendix C.2). The difference concerns how to complete the task, not simply whether completion is possible. We call the difference between action values under the two suffix policies *continuation mismatch*. Language agents motivate such decisions by combining reasoning with tools and interacting with stateful environments (Yao et al., 2023; Schick et al., 2023; Karpas et al., 2022; Nakano et al., 2021; Shen et al., 2023; Patil et al., 2023; Qin et al., 2023; Yao et al., 2022; Zhou et al., 2024; Lu et al., 2024; Yao et al., 2024; Barres et al., 2025).

Visited states and rollout policies shape sequential supervision. SEARN derives supervision through search, DAgger addresses learner-induced state shift, AggreVaTe uses expert cost-to-go on learner-visited states, and LOLS studies alternative roll-in and roll-out policies (Daumé et al., 2009; Ross & Bagnell, 2010; Ross et al., 2011; Ross & Bagnell, 2014; Chang et al., 2015). Scheduled sampling and Professor Forcing address related train–test mismatch in sequence models (Bengio et al., 2015; Goyal et al., 2016). We study the decision consequences of continuation choice when counterfactual returns are exact. Holding the checkpoint, candidate action, feasible-action set, and risk rule fixed, we compare targets defined by a reference controller and a frozen deployment controller. We thereby

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

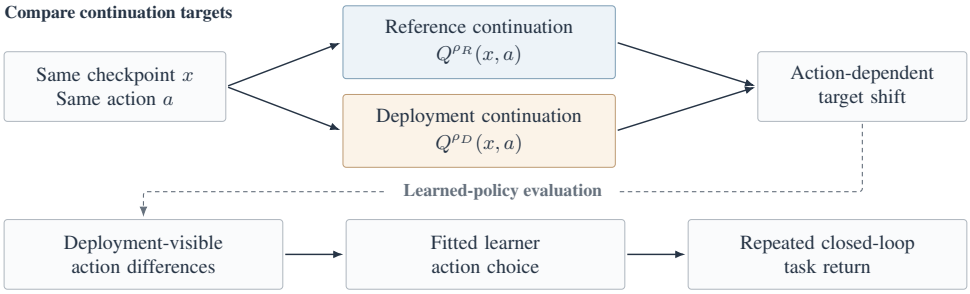


Figure 1: Continuation-conditioned supervision at a fixed decision. The upper comparison holds the complete simulator state and current action fixed, varying only the suffix controller; no new decision rule is fitted. The lower evaluation tests whether the resulting target differences remain observable, survive fitting, and affect closed-loop task return.

isolate target choice from state coverage and ask when it changes the preferred action, survives fitting, and improves complete-task return under the learned policy.

A change in target values need not change a decision: shifts shared by all actions preserve their ranking, and even action-dependent shifts may be too small to reverse the existing margin. A ranking change can also depend on hidden state unavailable to the learner or disappear during fitting. Finally, changing one action need not improve the full trajectory. We therefore connect a margin-based analysis of decision relevance with matched evaluations of fitted policies on complete tasks.

We study these stages in a controlled tool-use simulator, using CSOAA and MLP policies with structured observations. An eight-seed comparison identifies state coverage and learner capacity as the main drivers of aggregate performance. Fixed-state diagnostics isolate continuation choice: reference selection is strictly suboptimal under the deployment continuation at 16.8% of 800 checkpoints. Averaged over all 800 checkpoints, deployment-conditioned selection yields ideal one-step gains of $+0.068$ utility and $+1.88$ success percentage points. A separate historical label comparison shows concentrated disagreement, with its selected-value and risk-label differences reported separately from this fixed-gate diagnostic. These diagnostics locate potential gains before fitting; the full-policy evaluation tests whether learned agents realize them.

We fix targets and seed-specific holdouts before inspecting full-policy outcomes, comparing each target with reference supervision within learner and seed. Targets from the frozen deployment controller yield mean paired utility gains of $+0.076$ for CSOAA and $+0.099$ for MLP, with $+3.0$ success percentage points for both across three seeds. All six comparisons improve on both metrics, although most task-level confidence intervals include zero. Continuation-dependent target differences can thus survive fitting and improve complete-task performance in this controlled setting; intermediate target mixtures yield less consistent gains.

Contributions. Our contributions are threefold. **First**, we derive sufficient conditions under which changing the continuation preserves action choice, relating action-dependent target shifts to the learner-visible margin and fitting error. **Second**, we separate continuation choice from state coverage and learner capacity through controlled comparisons, and show that archived label comparisons concentrate disagreement among states with large selected-value gaps. **Third**, we demonstrate that changing continuation targets alone can improve held-out complete-task performance within matched learner–seed comparisons, while documenting mixed-target failures and statistical uncertainty. Together, these results connect continuation-dependent supervision to decision relevance, learnability, and full-policy outcomes.

2 Continuation-conditioned action values

Action values under a continuation. At a decision checkpoint, let x be the complete simulator state used to restore the environment for counterfactual evaluation. The state includes the decision time and accumulated task quantities. Write $o = h(x)$ for all learner-visible information, including legal-action information; hidden simulator variables remain available only to the evaluator. Let

$F_\eta(o)$ denote the legal actions that pass the common risk rule, with η indexing its fixed configuration. The analysis uses the same finite set $F_\eta(o)$ and tie rule for both targets. We consider at least two eligible actions; a singleton fixes the decision. Empty-set fallback is specified in Appendix B.

For a forced current action a and continuation policy ρ, define the expected complete-task utility

$$Q^\rho(x,a) = \mathbb{E}[G(\tau) \mid x_t = x,\ a_t = a,\ a_{t+1:T-1} \sim \rho], \tag{1}$$

Here the current action is forced, and the expectation is over any remaining environment or policy randomness; $G(\tau)$ is total task utility, including the fixed prefix. In the restored deterministic simulator, a suffix execution gives its exact return. The current-action risk gate is held fixed across target comparisons. A reference continuation ρ_R and a deployment continuation ρ_D assign two values to the same state-action pair. Their difference is

$$\Delta_Q(x,a) = Q^{\rho_D}(x,a) - Q^{\rho_R}(x,a). \tag{2}$$

From value shift to decision shift. To compare targets at the learner's information level, fix a common checkpoint distribution μ for both continuations. The observation-level values average over hidden states under this same distribution:

$$q^\rho(o,a) = \mathbb{E}_\mu[Q^\rho(X,a) \mid O = o], \qquad \overline{\Delta}_Q(o,a) = q^{\rho_D}(o,a) - q^{\rho_R}(o,a). \tag{3}$$

Write $q_R = q^{\rho_R}$. A continuation can change every action value without changing the decision because any shift shared by all eligible actions cancels from pairwise comparisons. Decision relevance therefore depends on how differently the continuation moves competing actions.

Let $a_R^*(o)$ maximize q_R over $F_\eta(o)$ under the common tie rule. Its margin over the best alternative is

$$\gamma_R(o) = q_R(o, a_R^*) - \max_{a \in F_\eta(o),\, a \neq a_R^*} q_R(o,a). \tag{4}$$

We summarize the action-dependent continuation shift by

$$\epsilon_{\mathrm{vis}}(o) = \frac{1}{2}\left(\max_{a \in F_\eta(o)} \overline{\Delta}_Q(o,a) - \min_{a \in F_\eta(o)} \overline{\Delta}_Q(o,a)\right). \tag{5}$$

The factor of one half makes $2\epsilon_{\mathrm{vis}}(o)$ the largest pairwise difference between continuation-induced action shifts. For any competing action, its advantage over the reference choice can increase by at most $2\epsilon_{\mathrm{vis}}(o)$. The reference-optimal action therefore remains uniquely optimal whenever

$$\gamma_R(o) > 2\epsilon_{\mathrm{vis}}(o). \tag{6}$$

For illustration, suppose two eligible actions have reference values $(1.0, 0.9)$. Adding 0.2 to both values changes the absolute returns but leaves the margin unchanged, with $\epsilon_{\mathrm{vis}} = 0$. Shifts of $(-0.1, +0.1)$ instead yield $(0.9, 1.0)$ and reverse the preferred action. Here $\gamma_R = 0.1$ and $2\epsilon_{\mathrm{vis}} = 0.2$: the differential shift exceeds the original margin. This example shows why absolute target differences alone cannot characterize decision relevance.

This sufficient condition certifies ranking stability, including when absolute values change substantially. Its failure leaves the ranking unresolved. The comparison holds μ fixed; it does not equate the state distributions induced by deploying different policies. Appendix A gives the derivation and the extension to fitted scores.

3 Matched continuation targets

For each checkpoint x and eligible action a, we restore the same complete state, force a, and execute the remaining trajectory with either the reference controller or a frozen deployment controller π_0, obtaining $Q_{\mathrm{ref}}(x,a)$ and $Q_{\pi_0}(x,a)$. Here π_0 generates labels; the policy fitted to those labels is subsequently evaluated on complete tasks. We keep π_0 fixed across target weights, so deployment-conditioned supervision refers to this declared continuation, not the final fitted policy's own continuation. The experiment tests whether changing this labeling assumption improves the resulting policy.

The held-out study interpolates between these two targets:

$$T_\alpha(x,a) = (1-\alpha)Q_{\mathrm{ref}}(x,a) + \alpha Q_{\pi_0}(x,a), \qquad \alpha \in \{0, 0.1, 0.5, 1\}. \tag{7}$$

Here $\alpha = 0$ uses reference labels and $\alpha = 1$ uses labels generated by π_0. Intermediate targets are arithmetic mixtures of the two recorded values for each state–action pair; no controller is sampled during target construction. Since

$$T_\alpha - Q_{\mathrm{ref}} = \alpha \left(Q_{\pi_0} - Q_{\mathrm{ref}} \right),$$

the action-dependent shift measured by ϵ_{vis} scales linearly with α before fitting. This does not imply linear or monotone changes in the fitted policy's return: fitting, action selection, and subsequent state visitation intervene between the target and the task outcome.

The paired label record separates counterfactual execution from target interpolation. Once both continuation values have been computed for a state–action pair, every T_α is obtained from those same values. Changing α therefore changes neither the number of recorded checkpoints nor the information available to the fitted learner. At deployment, the fitted policy chooses actions from its observation and the common risk scores; it does not perform the counterfactual suffix executions used to construct its training labels.

Within each learner–seed comparison, we hold the checkpoints, task weights, learner-visible inputs, eligible actions, and initialization and stopping rules fixed; only T_α changes. Each fitted policy is compared with the reference-supervised baseline ($\alpha = 0$) on the same held-out tasks under the same deployment gate. This isolates target choice from training coverage, learner capacity, and evaluation-task differences. Appendix B provides implementation details.

4 From target shift to deployed improvement

We examine three questions separately: does continuation choice change the preferred action given the learner's observation, does fitting produce different action choices, and do the fitted policies achieve higher complete-task utility and success? These distinguish ideal decision sensitivity, the learner's response to the targets, and task outcomes. A difference at one level does not establish improvement at the next.

Decision relevance. Equation 6 guarantees action invariance, but violating it need not reverse the ranking. Complete-state action disagreement and ideal one-step headroom identify sensitivity to continuation choice. These diagnostics motivate testing whether the learner can observe and exploit the differences; they do not establish that transfer.

Learnability. Fitting can preserve or alter the action ranking implied by the targets. For weight α, the ideal scores under the common checkpoint distribution are

$$q_\alpha(o, a) = q_R(o, a) + \alpha \overline{\Delta}_Q(o, a).$$

For fitted scores on the target utility scale, define maximum absolute errors

$$\delta_j(o) = \max_{a \in F_\eta(o)} \left| \widehat{q}_j(o, a) - q_j(o, a) \right|, \qquad j \in \{R, \alpha\}.$$

Under the common feasible set and tie rule, both fitted policies choose the reference-optimal action whenever

$$\gamma_R(o) > 2 \max\{\delta_R(o), \alpha \epsilon_{\mathrm{vis}}(o) + \delta_\alpha(o)\}. \tag{8}$$

This is an analytical sufficient condition, not an estimated certificate for the fitted models. Its proof (Appendix A) combines pairwise target shifts with score approximation errors.

Closed-loop transfer. One-step headroom measures an action change at a fixed checkpoint under a specified continuation. Full-policy evaluation uses the fitted controller throughout the task, allowing later decisions and visited states to change. We measure complete-task utility and success directly because local headroom does not guarantee improvement under repeated execution.

For each seed, all targets and learners share a 100-task holdout. Paired-bootstrap intervals resample tasks within each learner–seed comparison, holding fitted policies fixed. They measure task-level uncertainty, not variability from retraining. Appendices B and D report the protocol and individual intervals.

The experiments separate target choice from coverage and capacity, locate continuation-sensitive decisions, and test whether target changes improve fitted policies on held-out tasks.

Table 1: **Controlled eight-seed attribution study.** Cells are across-seed means; bootstrap 95% confidence intervals are reported in Appendix D, Table 5. The fixed global blend is retrospective.

Coverage	Target	Learner	Success (%) ↑	Utility ↑	Regret ↓
Logged	Reference	Linear	8.9	−0.114	0.338
Replay	Reference	Linear	39.0	1.056	0.143
Replay	Global blend	Linear	39.1	1.050	0.144
Replay	Reference	MLP	52.5	1.497	0.070
Replay	Global blend	MLP	50.8	1.433	0.080
—	Exact oracle	Planner	63.4	1.916	0.000

5 EXPERIMENTS

The simulator covers clarification, tracking, delegation, verification, and escalation tasks within six decisions. Legal actions (ASK, DELEGATE, VERIFY, ACT, ESCALATE) consume budget and change task state. Two runtime configurations differ in action skills. Learners observe public features such as uncertainty, progress, and remaining budget; counterfactual evaluation restores the complete state, including hidden variables. Utility combines successful completion with faithfulness, quality, and safety, while penalizing action cost, interventions, and unsafe events. Two successful trajectories can therefore receive different returns, as in the introductory checkpoint; success alone would not distinguish them.

5.1 WHAT DRIVES AGGREGATE PERFORMANCE?

Table 1 compares coverage, targets, and learner classes across eight seeds. With the linear learner and reference target fixed, replay raises success from 8.9% to 39.0% and utility from -0.114 to 1.056. Keeping replay and reference supervision fixed, replacing the linear learner with an MLP adds 13.5 success percentage points and 0.441 utility. The retrospective global blend provides no comparable gain: under replay, linear success changes by $+0.1$ point and utility by -0.006, while MLP success falls by 1.7 points and utility by 0.064. Thus coverage and learner choice drive the large gains in this study; this particular target blend does not improve either learner's mean utility. Reference-supervised MLP success remains below the planner's 63.4%.

5.2 WHERE DOES CONTINUATION CHOICE CHANGE THE DECISION?

Reference-conditioned selection is strictly suboptimal under the deployment continuation at 134 of 800 checkpoints (16.75%), excluding ties. Literal selected-action disagreement is 136/800 (17.0%). Averaged over all 800 checkpoints, deployment-conditioned selection improves utility by 0.068 (95% CI $[0.026, 0.113]$) and success by 1.88 percentage points (95% CI $[0.63, 3.25]$) relative to reference-conditioned selection. Utility headroom is nonnegative by construction because both choices are evaluated under the deployment continuation. Its magnitude measures the cost of reference-based selection; the success contrast need not have the same sign.

Figure 2 distinguishes coverage differences from continuation sensitivity. Panel (a) shows that late states constitute 1.1% of logged data, versus 19.3% and 17.0% of reference and deployment rollouts, documenting the coverage change accompanying replay. Panel (b) holds checkpoints fixed: delegation has the largest mean one-step headroom among the five families, at $+0.162$ utility and $+5.00$ success percentage points. Panel (c) uses a separate pool of 10,933 shared states, where selected-label action disagreement rises from 15.8% overall to 59.0% in the highest-$|\Delta v_{\text{sel}}|$ decile. This historical comparison can change both value and risk labels. Here Δv_{sel} is the gap between values of the separately selected actions, not the same-action shift Δ_Q in Equation 2; Appendix C.1 specifies the comparison.

Table 2 reports a retrospective development-set comparison from archived candidate grids. Replay corresponds to success increases of 33.8 and 33.3 percentage points under reference and deployment-conditioned targets. Within logged data, the target contrast is $+1.0$ success point and $+0.164$ utility; after replay it is $+0.5$ point and $+0.001$ utility. Candidate configurations were se-

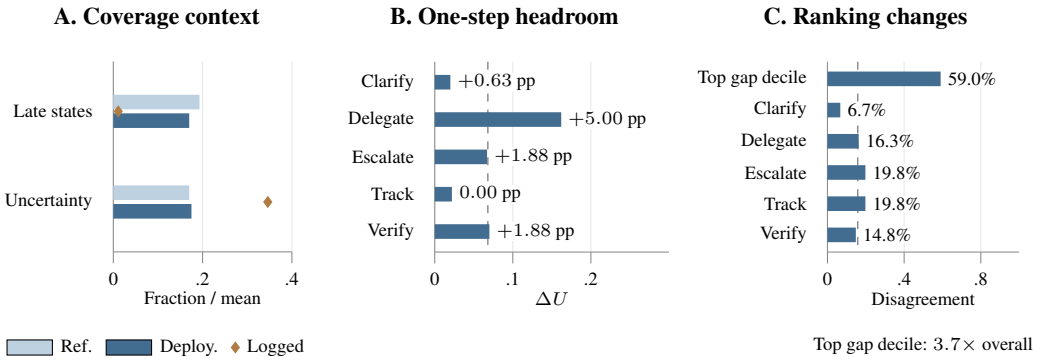


Figure 2: **Localization of continuation effects.** (a) Logged supervision occupies a different state regime from reference and deployment rollouts. (b) Ideal deployment-conditioned selection has heterogeneous one-step headroom across task families; labels report success-point changes. (c) Historical selected-label action disagreement is 15.8% overall and 59.0% in the highest-$|\Delta v_{\text{sel}}|$ decile. Dashed guides mark overall values. Panels (b,c) use separate pools; (c) can change value and risk labels, with Δv_{sel} defined in the text.

Table 2: **Retrospective coverage–target development diagnostic.** Panel A reports selected configurations; Panel B gives descriptive differences within one learner class. Held-out full-policy target comparisons appear in Table 3.

A. Factorial cells				
Coverage	Target continuation	Success (%) ↑	Utility ↑	Regret ↓
Logged	Reference	9.5	−0.071	0.321
Logged	Deployment-conditioned	10.5	0.092	0.293
Replay	Reference	43.3	1.235	0.103
Replay	Deployment-conditioned	43.8	1.237	0.103
B. Descriptive contrasts				
Effect	Contrast	Δ success (pp)	Δ utility	Δ regret
Coverage @ reference	Replay − Logged	**+33.8**	**+1.307**	**−0.218**
Coverage @ deployment	Replay − Logged	**+33.3**	**+1.144**	**−0.191**
Target @ logged	Deployment − Reference	+1.0	+0.164	−0.027
Target @ replay	Deployment − Reference	+0.5	+0.001	0.000

lected within cells, so these contrasts describe the archived configurations rather than isolate target choice alone. The study below fixes the fitting protocol across targets and evaluates separate held-out tasks.

5.3 Does the signal survive fitted deployment?

We evaluate full-policy transfer with four target weights ($\alpha \in \{0, 0.1, 0.5, 1\}$), two learner classes (CSOAA and MLP), and three training seeds, yielding 24 fitted policies. Each seed has a separate 100-task holdout fixed before outcome inspection and shared across its eight target–learner combinations. We compute task-paired effects against $\alpha = 0$ within each learner and seed, then summarize these effects across the three seeds. This tests whether changing supervision improves complete-task performance after fitting.

Table 3 shows positive utility and success effects in all six learner–seed comparisons at $\alpha = 1$. Mean paired gains ($\pm$ sample SD across three seeds) are $+0.076 \pm 0.056$ utility and $+3.0 \pm 1.7$ success percentage points for CSOAA, and $+0.099 \pm 0.038$ utility and $+3.0 \pm 2.0$ points for MLP. Across all non-reference targets, 16 of 18 utility effects and 15 of 18 success effects are positive; these counts describe direction, not statistical significance. Although the tabulated means increase with α, individual runs need not: MLP utility improves over reference in all nine comparisons, whereas CSOAA has two negative utility effects at intermediate weights.

Table 3: **Held-out full-policy target comparison.** Values are mean ± sample SD over three independently trained policies, each evaluated on a distinct fixed 100-task panel. Paired effects are computed within learner/seed.

		Absolute outcome		Paired effect vs. reference	
Learner	Deployment wt. α	Utility ↑	Success (%) ↑	ΔU ↑	Δ success (pp)↑
CSOAA	0 (Ref.)	1.404 ± 0.123	48.0 ± 3.6	—	—
	0.1	1.414 ± 0.119	48.3 ± 3.5	$+0.009 \pm 0.033$	$+0.3 \pm 1.2$
	0.5	1.423 ± 0.098	49.0 ± 3.0	$+0.019 \pm 0.034$	$+1.0 \pm 1.0$
	1.0	1.480 ± 0.112	51.0 ± 3.6	$+0.076 \pm 0.056$	$+3.0 \pm 1.7$
MLP	0 (Ref.)	1.229 ± 0.183	45.3 ± 3.8	—	—
	0.1	1.318 ± 0.183	47.3 ± 4.7	$+0.089 \pm 0.013$	$+2.0 \pm 1.0$
	0.5	1.319 ± 0.258	48.0 ± 6.6	$+0.091 \pm 0.084$	$+2.7 \pm 3.1$
	1.0	1.327 ± 0.219	48.3 ± 5.7	$+0.099 \pm 0.038$	$+3.0 \pm 2.0$

$\alpha = 0$ is reference supervision and $\alpha = 1$ is fully deployment-conditioned supervision. Figure 3 shows the run-level paired effects; Appendix D reports exact task-level paired 95% intervals.

The paired effects also distinguish target benefit from absolute learner performance. CSOAA has higher mean utility than MLP under both reference and fully deployment-conditioned supervision, while MLP has the larger mean utility gain from changing targets. The target comparison asks how each learner responds to the same change in supervision. It does not require the learner with the larger gain to be the stronger controller in absolute terms.

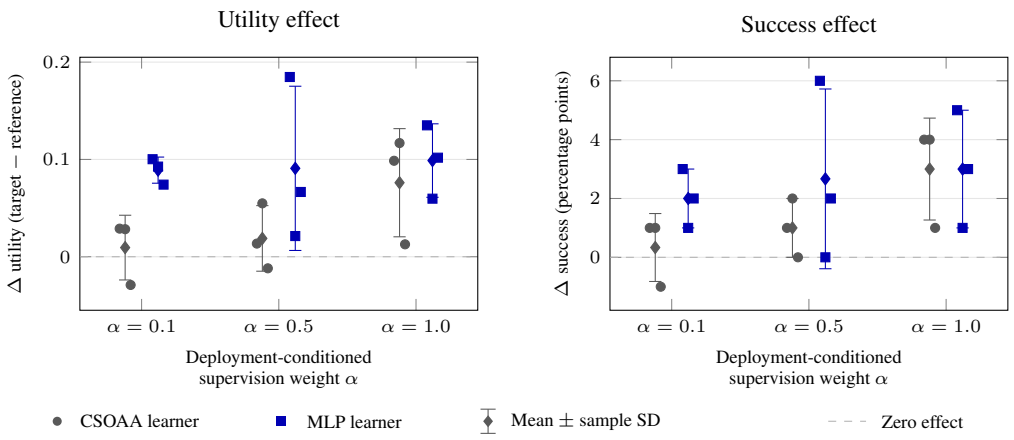


Figure 3: **Paired policy gains from deployment-conditioned supervision.** The weight α varies the training target from reference supervision ($\alpha = 0$, the baseline) to fully deployment-conditioned supervision ($\alpha = 1$). Both panels show gains relative to that baseline within the same learner and seed; positive values indicate improvement. Gray circles and blue squares identify CSOAA and MLP learners, respectively, with three runs per target. Diamonds and whiskers show means ± sample SD across runs, not confidence intervals. At $\alpha = 1$, all six matched runs improve on both metrics.

From target differences to policy gains. Figure 3 shows the run-level effects underlying Table 3. At $\alpha = 1$, every point lies above zero in both panels, showing consistent improvement across the evaluated learner–seed pairs. At intermediate weights, some CSOAA effects fall below zero. The whiskers summarize dispersion across the three fitted-policy/holdout pairs; they are sample standard deviations, not confidence intervals. Task-level paired-bootstrap intervals address a different question: uncertainty in each fitted pair's effect across test tasks. Most utility intervals include zero (Appendix D), so directional consistency alone does not establish statistical significance.

5.4 BLANKET TARGET MIXING IS NOT SUFFICIENT

A retrospective experiment illustrates a failure of global mixing. For each suffix, it samples either the reference or current-learner continuation with equal probability and uses the resulting return as a label, unlike the arithmetic mixtures in Equation 7. On a reused 100-task panel, mean utility falls from 1.416 to 1.312 and success from 50.3% to 48.0% across three training seeds. Across the 300 seed–task evaluations, the mixed-target policies repair four reference-policy failures but disrupt eleven successes. The excess of disruptions over repairs explains the negative success contrast for this sampled-mixture protocol. Appendix E reports the task-level distribution of gains and losses.

6 DISCUSSION

Coverage determines which states supply training examples; continuation determines how candidate actions are valued there. Replay produces the larger aggregate gains in the retrospective attribution studies. The matched held-out study nevertheless shows that target choice can improve fitted policies with training states and fitting procedures held fixed. These findings identify complementary supervision choices, with effects that depend on the learner and evaluation setting. These results suggest a concrete reporting practice: a counterfactual dataset should identify its suffix controller alongside its checkpoint and action definitions. Otherwise, differences in labels can be mistaken for value-estimation errors even when each return is computed exactly under its own continuation.

Local headroom and full-policy gains answer different questions. Value shifts can preserve rankings, complete-state differences may be invisible to the learner, fitting can change rankings, and repeated execution changes visited states. Our studies examine these stages separately rather than estimate a common signal's attenuation through them. The sampled-mixture failure further shows that changing continuation does not uniformly improve performance.

The margin analysis suggests testing selective correction where learner-visible shifts are large relative to reference margins. Such a rule requires reliable estimates, thresholds chosen without test feedback, and a matched comparison with reference and fully deployment-conditioned supervision. Whether selection improves the benefit–cost tradeoff remains open.

7 LIMITATIONS

The controlled simulator enables exact state restoration, structured observations, and a fixed action vocabulary, allowing continuation choice to be isolated. The experiments therefore establish effects within this setting. They do not test language-based action generation, stochastic external tools, or real execution latency. Extending the findings to native language-model agents requires evaluating how these factors affect target estimation and policy learning; the observed gains alone do not establish performance improvements in such systems.

The held-out study uses three training seeds per target and learner, each with a separate 100-task panel. Across-run dispersion therefore reflects both training and test-panel variation, while task-level intervals condition on fitted policies; most utility intervals include zero. The historical 50–50 comparison reuses a 100-task panel and provides retrospective evidence. For the held-out study, task-identity checks exclude matches against each seed's inherited training, development, and historical-test pools under the declared normalization, but do not exclude shared generative templates. These controls prevent identified task reuse without establishing generalization to new task structures.

Finally, we do not implement a rule that selects which states receive continuation-aware correction. The margin conditions certify action invariance, not improvement when they fail. An operational selection rule would require estimating learner-visible margins and shifts, choosing thresholds without test-outcome feedback, and validating the resulting fitted policies. Its benefits and computational cost remain unmeasured.

8 RELATED WORK

Learning to search and sequence mismatch. Learning-to-search methods construct supervision from sequential decisions and their downstream costs (Daumé et al., 2009; Ross & Bagnell, 2010).

DAgger aggregates expert labels on learner-visited states, while AggreVaTe uses expert cost-to-go and AggreVaTeD develops differentiable imitation learning (Ross et al., 2011; Ross & Bagnell, 2014; Sun et al., 2017). LOLS directly studies roll-in and roll-out choices and provides guarantees relative to both the reference policy and local deviations from the learned policy (Chang et al., 2015). Scheduled sampling and Professor Forcing address related training–generation mismatch in sequence models (Bengio et al., 2015; Goyal et al., 2016). Our study builds on this dependence of supervision on continuation. We fix checkpoints, candidate actions, and the risk rule to examine when exact continuation-conditioned values change action rankings and whether those target differences affect fitted-policy return, separating this comparison from changes in state coverage.

Offline reinforcement learning, off-policy evaluation, and safety. Offline RL addresses policy learning from fixed data through behavior constraints and conservative or implicit improvement objectives (Fujimoto et al., 2019; Kumar et al., 2020; Kostrikov et al., 2022; Levine et al., 2020). Off-policy evaluation estimates another policy's value from logged data (Jiang & Li, 2016; Nachum et al., 2019), while safe policy improvement and constrained optimization address performance guarantees and constraints (Laroche et al., 2019; Achiam et al., 2017). Our simulator instead yields continuation-specific checkpoint returns by restoring complete state and executing deterministic suffixes. This removes return-estimation uncertainty at those checkpoints, but leaves fitting and test-task uncertainty. A common risk gate isolates continuation choice without providing a new safety guarantee.

Selective prediction, deferral, and mixed initiative. Selective prediction studies when to abstain, learning-to-defer considers when to assign a decision to an expert, and second-opinion work predicts disagreement among experts (Chow, 1970; El-Yaniv & Wiener, 2010; Geifman & El-Yaniv, 2017; 2019; Madras et al., 2018; Raghu et al., 2019; Mozannar & Sontag, 2020). Mixed-initiative interfaces and dialogue policies also address control allocation and information gathering over an interaction (Horvitz, 1999; Williams & Young, 2007). Our focus is the supervision assigned to such actions within a continuing task. In our simulator, ASK, VERIFY, DELEGATE, and ESCALATE change the state from which subsequent decisions are made. Holding the current checkpoint and candidate action fixed, we examine how the controller used for those subsequent decisions changes the action's target value and the policy learned from it.

Tool-using and interactive language agents. Language agents combine reasoning with external tools and API calls (Yao et al., 2023; Schick et al., 2023; Qin et al., 2023). Benchmarks evaluate sequential interaction in web, software, and computer environments (Zhou et al., 2024; Jimenez et al., 2024; Xie et al., 2024). ToolSandbox and the τ-bench family explicitly couple tool calls with evolving state and user interaction (Lu et al., 2024; Yao et al., 2024; Barres et al., 2025). These settings motivate continuation-conditioned supervision; our experiments isolate that question with structured observations and fixed actions.

Step supervision and self-correction. Process supervision supplies feedback on intermediate reasoning steps (Lightman et al., 2023). Reflexion uses feedback and verbal reflection to guide subsequent attempts, while Self-Refine iteratively revises outputs using self-generated feedback (Shinn et al., 2023; Madaan et al., 2023). These methods connect intermediate feedback to later behavior. We examine how the continuation used to complete a fixed state–action pair defines its return-based supervision target. This question concerns the meaning of a target under a specified future controller, rather than the generation of critiques or revised reasoning traces.

9 Conclusion

Exact counterfactual returns depend on the controller that completes the trajectory. We characterize sufficient conditions under which changing that controller preserves the learner-visible optimal action, accounting for fitting error. In the controlled simulator, reference selection is strictly suboptimal under the frozen continuation at a minority of checkpoints. The held-out study shows that changing targets alone can improve full-policy utility and success, while a separate sampled-mixture experiment yields losses. Together, these results make continuation choice an explicit supervision-design variable whose decision effects and deployed benefits should be evaluated separately.

AI USE STATEMENT

The research direction and central ideas were developed by the authors. Generative AI tools assisted with refining hypotheses and mathematical arguments, code and analysis support, interpretation of results, manuscript editing, and figure preparation. The authors reviewed and verified AI-assisted outputs against the cited literature, retained source and result artifacts, and executable analyses used in the paper. The authors take full responsibility for the final content, results, and conclusions.

REPRODUCIBILITY STATEMENT

The main text defines the continuation targets and matched evaluation design. Appendix A proves the margin conditions. Appendix B specifies the simulator metrics, data pools, learner settings, task weighting, held-out task construction, and common deployment rule. Appendix C.1 documents the diagnostic populations and historical configuration selection. Appendix D provides the eight-seed uncertainty estimates and all 18 held-out paired utility intervals; Appendix E reports the historical sampled-continuation outcomes and paired success repairs and disruptions.

REFERENCES

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 22–31. PMLR, 2017. URL https://proceedings.mlr.press/v70/achiam17a.html.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-bench: Evaluating conversational agents in a dual-control environment. *arXiv preprint arXiv:2506.07982*, 2025. URL https://arxiv.org/abs/2506.07982.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL https://papers.nips.cc/paper_files/paper/2015/hash/e995f98d56967d946471af29d7bf99f1-Abstract.html.

Kai-Wei Chang, Akshay Krishnamurthy, Alekh Agarwal, Hal Daumé, III, and John Langford. Learning to search better than your teacher. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2058–2066. PMLR, 2015. URL https://proceedings.mlr.press/v37/changb15.html.

C. K. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46, 1970. doi: 10.1109/TIT.1970.1054406.

Hal Daumé, III, John Langford, and Daniel Marcu. Search-based structured prediction. *Machine Learning*, 75(3):297–325, 2009. doi: 10.1007/s10994-009-5106-x.

Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11(53):1605–1641, 2010. URL https://jmlr.org/papers/v11/el-yaniv10a.html.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2052–2062. PMLR, 2019. URL https://proceedings.mlr.press/v97/fujimoto19a.html.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://papers.nips.cc/paper_files/paper/2017/hash/4a8423d5e91fda00bb7e46540e2b0cf1-Abstract.html.

Yonatan Geifman and Ran El-Yaniv. SelectiveNet: A deep neural network with an integrated reject option. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2151–2159. PMLR, 2019. URL https://proceedings.mlr.press/v97/geifman19a.html.

Anirudh Goyal, Alex Lamb, Ying Zhang, Saizheng Zhang, Aaron Courville, and Yoshua Bengio. Professor Forcing: A new algorithm for training recurrent networks. In *Advances in Neural Information Processing Systems*, volume 29, 2016. URL https://proceedings.neurips.cc/paper/2016/file/16026d60ff9b54410b3435b403afd226-Paper.pdf.

Eric Horvitz. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 159–166, 1999. doi: 10.1145/302979.303030.

Nan Jiang and Lihong Li. Doubly robust off-policy value evaluation for reinforcement learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 652–661. PMLR, 2016. URL https://proceedings.mlr.press/v48/jiang16.html.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

Ehud Karpas, Omri Abend, Yonatan Belinkov, Barak Lenz, Opher Lieber, Nir Ratner, Yoav Shoham, Hofit Bata, Yoav Levine, Kevin Leyton-Brown, Dor Muhlgay, Noam Rozen, Erez Schwartz, Gal Shachaf, Shai Shalev-Shwartz, Amnon Shashua, and Moshe Tenenholtz. MRKL systems: A modular, neuro-symbolic architecture that combines large language models, external knowledge sources and discrete reasoning. *arXiv preprint arXiv:2205.00445*, 2022. URL https://arxiv.org/abs/2205.00445.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=68n2s9ZJWF8.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1179–1191, 2020. URL https://papers.nips.cc/paper_files/paper/2020/hash/0d2b2061826a5df3221116a5085a6052-Abstract.html.

Romain Laroche, Paul Trichelair, and Remi Tachet des Combes. Safe policy improvement with baseline bootstrapping. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3652–3661. PMLR, 2019. URL https://proceedings.mlr.press/v97/laroche19a.html.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020. URL https://arxiv.org/abs/2005.01643.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step, 2023. URL https://arxiv.org/abs/2305.20050.

Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Felix Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, Zirui Wang, and Ruoming Pang. ToolSandbox: A stateful, conversational, interactive evaluation benchmark for LLM tool use capabilities. *arXiv preprint arXiv:2408.04682*, 2024. URL https://arxiv.org/abs/2408.04682.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-Refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*, 2023. URL https://arxiv.org/abs/2303.17651.

David Madras, Toniann Pitassi, and Richard Zemel. Predict responsibly: Improving fairness and accuracy by learning to defer. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://arxiv.org/abs/1711.06664.

Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7076–7087. PMLR, 2020. URL https://proceedings.mlr.press/v119/mozannar20b.html.

Ofir Nachum, Yinlam Chow, Bo Dai, and Lihong Li. DualDICE: Behavior-agnostic estimation of discounted stationary distribution corrections. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://papers.nips.cc/paper/2019/hash/cf9a242b70f45317ffd281241fa66502-Abstract.html.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. WebGPT: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021. URL https://arxiv.org/abs/2112.09332.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: Large language model connected with massive APIs. *arXiv preprint arXiv:2305.15334*, 2023. URL https://arxiv.org/abs/2305.15334.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. *arXiv preprint arXiv:2307.16789*, 2023. URL https://arxiv.org/abs/2307.16789.

Maithra Raghu, Katy Blumer, Rory Sayres, Ziad Obermeyer, Robert Kleinberg, Sendhil Mullainathan, and Jon Kleinberg. Direct uncertainty prediction for medical second opinions. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5281–5290. PMLR, 2019. URL https://proceedings.mlr.press/v97/raghu19a.html.

Stéphane Ross and J. Andrew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 661–668. PMLR, 2010. URL https://proceedings.mlr.press/v9/ross10a.html.

Stéphane Ross and J. Andrew Bagnell. Reinforcement and imitation learning via interactive no-regret learning. *arXiv preprint arXiv:1406.5979*, 2014. URL https://arxiv.org/abs/1406.5979.

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*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635. PMLR, 2011. URL https://proceedings.mlr.press/v15/ross11a.html.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. HuggingGPT: Solving AI tasks with ChatGPT and its friends in Hugging Face. *arXiv preprint arXiv:2303.17580*, 2023. URL https://arxiv.org/abs/2303.17580.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *arXiv preprint arXiv:2303.11366*, 2023. URL https://arxiv.org/abs/2303.11366.

Wen Sun, Arun Venkatraman, Geoffrey J. Gordon, Byron Boots, and J. Andrew Bagnell. Deeply AggreVaTeD: Differentiable imitation learning for sequential prediction. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3309–3318. PMLR, 2017. URL https://proceedings.mlr.press/v70/sun17d.html.

Jason D. Williams and Steve Young. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422, 2007. doi: 10.1016/j.csl.2006.06.008.

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. *arXiv preprint arXiv:2404.07972*, 2024. URL https://arxiv.org/abs/2404.07972.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. WebShop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://arxiv.org/abs/2207.01206.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024. URL https://arxiv.org/abs/2406.12045.

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 *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=oKn9c6ytLx.

A Proofs of the Margin Conditions

Fix an observation o, a finite eligible set with at least two actions, and the common conditional state distribution $\mu(X \mid O = o)$. All scores below use this same distribution and action set. For any competitor $a \neq a_R^*$,

$$q^{\rho_D}(o, a_R^*) - q^{\rho_D}(o, a) = q_R(o, a_R^*) - q_R(o, a) + \overline{\Delta}_Q(o, a_R^*) - \overline{\Delta}_Q(o, a) \tag{9}$$

$$\geq \gamma_R(o) - 2\epsilon_{\text{vis}}(o). \tag{10}$$

Thus Equation 6 makes every pairwise gap strictly positive. A tie in the reference maximizers gives $\gamma_R = 0$ and is not certified by this strict condition. A singleton eligible set fixes the action without a margin calculation. When the risk threshold admits no action, the implementation uses minimum risk followed by the utility/cost tie rule; the stated certificate concerns the nonempty threshold-feasible set.

For $\alpha \geq 0$, the ideal mixed-target gap is at least $\gamma_R - 2\alpha\epsilon_{\text{vis}}$. An absolute approximation error of at most δ_α at each action reduces a pairwise fitted gap by at most $2\delta_\alpha$. Hence

$$\widehat{q}_\alpha(o, a_R^*) - \widehat{q}_\alpha(o, a) \geq \gamma_R(o) - 2\alpha\epsilon_{\text{vis}}(o) - 2\delta_\alpha(o).$$

For the reference fit the analogous bound is $\gamma_R - 2\delta_R$. Equation 8 makes both bounds positive. The errors are analytical quantities on a common utility scale; no calibration or measured bound on these errors is asserted for the trained CSOAA or MLP scores. The result is local to the fixed distribution; failure of the sufficient condition leaves the ranking unresolved.

B Experimental Protocol

The following settings specify the held-out target comparison and its common deployment interface. Complete uncertainty estimates are reported in Appendix D.

B.1 Environment and metrics

The five action types consume normalized budget: ASK costs 0.12, DELEGATE 0.16, VERIFY 0.14, ACT 0.10, and ESCALATE 0.13. A task terminates when progress reaches 0.95, the six-decision horizon is reached, or budget is exhausted. Success additionally requires adequate faithfulness, quality, and safety, an acceptable risk level or granted permission, and no unsafe events. Default faithfulness, quality, and safety thresholds are 0.60, 0.55, and 0.75. The scalar utility is

$$G = 3S + 0.8F + 0.4Q + 0.3A - 1.2C - 0.35I - 1.8N,$$

where S is binary success; F, Q, A are normalized faithfulness, quality, and safety; C is total action cost; I counts interventions; and N counts unsafe events. The simulator computes these attributes from its terminal state. Reported regret averages the reference-planner value gap between the best legal action and the executed action over decisions; it is distinct from a paired full-task utility difference. Exact transitions and task-specific thresholds are specified by the accompanying simulator source.

B.2 Checkpoints, targets, and learners

At a checkpoint, the labeler restores the identical complete state for every feasible candidate, forces that candidate once, and executes the remaining task with a declared continuation. Hidden simulator state is used only by the counterfactual evaluator; the learner receives the deployment observation and feasible-action information.

The held-out study was specified after the retrospective comparison and before fitted-policy outcomes were inspected. It fixes the train/development tasks and checkpoints, the restore-and-force interface, reference- and deployment-continuation values, the common risk gate, and the target grid. The train/development pool contains 3,739 decision states from 630 tasks. For seeds 19, 29, and 39, the train/development row counts are respectively 830/414, 827/418, and 833/417.

For every stored legal action, deterministic label construction records the reference value Q_{ref}, the frozen deployment-continuation value Q_{π_0}, and the four target columns in Equation 7. Each action–continuation pair uses one deterministic suffix execution.

The frozen deployment controller for each seed loads the reference-trained MLP fast/slow utility and risk models. Newly fitted CSOAA and MLP policies are separate from this label-generating controller. The study contains 24 target–learner–seed cells: four targets, two learner paths (CSOAA and MLP), and three seeds. CSOAA uses five action classes, five passes, and a 24-bit feature space in Vowpal Wabbit; its cost for action a is $\max_b T_\alpha(x, b) - T_\alpha(x, a)$, a state-dependent shift of negative utility. The MLP uses hidden layers of 64 and 32 units, fast features, ReLU activations, regularization 10^{-4}, early stopping enabled, and at most 80 iterations. Both use the declared training seed. No target-specific hyperparameter search or development-set selection is performed. All 12 MLP fits reached this iteration limit with convergence warnings. The same budget was retained for every target; no cell was rerun in response to test outcomes.

B.3 Task-balanced training

The locked pool assigns each task total mass one, so a task with n_i recorded decisions gives each of its states weight $1/n_i$. The two learner paths do not expose a common per-example weight interface. To preserve equal task mass without changing either learner, the fit uses a deterministic expansion. With $n_i \in \{5, 6\}$ and $L = \mathrm{lcm}(5, 6) = 30$, every state of task i is replicated exactly $r_i = L/n_i$ times. Thus the unweighted average data loss on the expanded sample satisfies

$$\frac{1}{ML} \sum_{i=1}^{M} \sum_{j=1}^{n_i} \sum_{r=1}^{L/n_i} \ell_\theta(x_{ij}; \tau) = \frac{1}{M} \sum_{i=1}^{M} \frac{1}{n_i} \sum_{j=1}^{n_i} \ell_\theta(x_{ij}; \tau). \tag{11}$$

Every target–seed bundle therefore uses 4,200 task-balanced training rows obtained from the same 140 original training tasks. Replication equalizes additive task-loss mass, not optimization trajectories. The independent training units remain the 140 original tasks; all targets use the identical expansion.

B.4 Held-out task construction

For each seed s, the original task generator and frozen source configuration produce exactly 100 test-only tasks before any fitted-policy outcome is inspected. Let ϕ be the declared semantic normalization used for task identity and let $\mathcal{P}_s$ contain all inherited train, development, and historical-test tasks. The acceptance condition is

$$\phi(\mathcal{H}_s) \cap \phi(\mathcal{P}_s) = \varnothing. \tag{12}$$

The resulting holdouts for seeds 19, 29, and 39 have zero task-identity collisions against their inherited pools under that normalization. Test tasks are not replaced in response to model performance or evaluation failure. Semantic disjointness prevents direct task reuse under the declared identity map; it does not prove that the generator has no higher-level template overlap.

B.5 Common deployment rule

The deployment gate requires a score for every legal action. The MLP path directly provides fitted utility scores. The ordinary CSOAA prediction interface returns only its winning class, so evaluation recovers its test-only raw class scores without changing the fitted model. If $c_g(o, a)$ is the returned CSOAA cost score, evaluation uses $s_g(o, a) = -c_g(o, a)$ on the original legal-action set. Both learner paths then apply the identical frozen risk gate and fallback ordering:

$$a_g(o) = \begin{cases} \arg\max_{a \in \mathcal{F}(o)} s_g(o, a), & \mathcal{F}(o) \neq \varnothing, \\ \arg\max_{a \in \arg\min_{b \in \mathcal{A}(o)} r(o,b)} s_g(o, a), & \mathcal{F}(o) = \varnothing, \end{cases} \tag{13}$$

Here $\mathcal{F}(o) = \{a \in \mathcal{A}(o) : r(o, a) \leq \kappa\}$. Residual ties follow the original legal-action order. For CSOAA, test mode requests raw scores for the full legal class list with dummy zero costs; it does not update the model. A frozen-model check verified that minimum raw cost agrees with the ordinary winning-class interface.

C Checkpoint Diagnostics and Historical Comparisons

C.1 Populations and comparison definitions

The reported checkpoint result uses the 800 anchors of the frozen reference-trained MLP group, with 160 per family and one anchor per task. Both ideal selectors use the same frozen current-action risk scores. The reference-selected action is strictly worse under the deployment continuation at 134 anchors (16.75%); literal selected-action disagreement occurs at 136 (17.0%), including two ties in deployment value. Both forced actions are evaluated with the same deployment suffix. Utility headroom is nonnegative by construction; the success difference is not constrained to be positive.

The 10,933 shared-state rows compare archived reference value/risk labels with the selected label configuration: policy-conditioned for seeds 19 and 29, reference-conditioned for seed 39. Each side selects its own action using its own value and risk labels. Define $\Delta v_{\mathrm{sel}} = v_{\mathrm{new}}(a_{\mathrm{new}}) - v_{\mathrm{old}}(a_{\mathrm{old}})$. This descriptive gap is not $\Delta_Q(x, a)$ for a fixed action and does not isolate value changes from risk-label changes. There are 1,724 action disagreements overall. The explicit rule $|\Delta v_{\mathrm{sel}}| \geq q_{0.9}$, using the linearly interpolated empirical quantile, selects 1,094 rows at threshold 0.0753506; 646 disagree (59.049%, displayed as 59.0%).

Table 2 is a three-seed development-set diagnostic. Its logged/reference row comes from the seed-model selection history; other cells select the highest-utility candidates within the appropriate target group from warm-start or replay-iteration grids, with success, safety, cost, and regret as tie breakers. Configuration selection and development evaluation are therefore coupled. Table 3 instead uses fixed fitting rules and seed-specific test-only tasks.

C.2 Recorded checkpoint example

The introductory example is the first strict-disagreement record ordered by seed and task identifier in the frozen MLP checkpoint population: seed 19, `clarify_test_0005`, decision index 2, runtime A. Four decisions remain. Reference-conditioned selection chooses ACT; deployment-conditioned selection chooses VERIFY. Their recorded utilities, both followed by the frozen deployment suffix, are respectively 2.918043 and 2.939489, and both succeed. This illustrates different action preferences among successful continuations.

C.3 Family-level breakdown

Table 4: Family-level continuation sensitivity. Headroom columns use 160 fixed complete-state checkpoints per family; disagreement columns compare historical selected value/risk labels on a separate pool of 10,933 shared states. The two diagnostics are reported side by side for localization only and are not pooled.

Task family	Headroom ΔU	Headroom Δ success (pp)	Shared states	Top-action disagreement
Clarification	$+0.020$	$+0.63$	1,929	6.7%
Delegation	$+0.162$	$+5.00$	2,178	16.3%
Escalation	$+0.067$	$+1.88$	2,486	19.8%
Tracking	$+0.022$	0.00	2,091	19.8%
Verification	$+0.070$	$+1.88$	2,249	14.8%

D Additional Statistical Results

D.1 Eight-seed attribution study

Table 5 reports uncertainty across the eight experiment seeds for the main-text attribution study.

D.2 Held-out paired utility contrasts

Table 6 reports every non-reference target against $\alpha = 0$ within the same learner and seed. Checkpoints, task weights, inputs, fitting rules, and deployment gate are fixed across targets. Each interval

Table 5: Bootstrap 95% confidence intervals for the eight-seed controlled study in Table 1. Values are mean [95% CI] over experiment seeds.

Coverage	Target	Learner	Success	Utility	Regret
Logged	Reference	Linear	$0.089\,[0.043, 0.139]$	$-0.114\,[-0.274, 0.063]$	$0.338\,[0.301, 0.372]$
Replay	Reference	Linear	$0.390\,[0.352, 0.428]$	$1.056\,[0.919, 1.191]$	$0.143\,[0.125, 0.162]$
Replay	Global blend	Linear	$0.391\,[0.362, 0.419]$	$1.050\,[0.948, 1.147]$	$0.144\,[0.126, 0.163]$
Replay	Reference	MLP	$0.525\,[0.495, 0.552]$	$1.497\,[1.401, 1.588]$	$0.070\,[0.063, 0.076]$
Replay	Global blend	MLP	$0.508\,[0.453, 0.556]$	$1.433\,[1.252, 1.582]$	$0.080\,[0.064, 0.101]$
—	Exact oracle	Planner	$0.634\,[0.604, 0.666]$	$1.916\,[1.823, 2.010]$	$0.000\,[0.000, 0.000]$

resamples paired outcomes from that seed's 100 holdout tasks, conditional on its fitted policies; different seed panels are not pooled.

Table 6: Prospective full-policy utility contrasts on separate seed-specific holdouts. Each cell is target minus reference supervision ($\alpha = 0$) for the same learner path and training seed; brackets are deterministic paired-bootstrap 95% intervals over that seed's 100 holdout tasks.

Learner	α	Seed 19	Seed 29	Seed 39
CSOAA	0.1	$+0.029\,[-0.003,\ 0.091]$	$+0.028\,[-0.005,\ 0.088]$	$-0.029\,[-0.113,\ -0.0002]$
CSOAA	0.5	$+0.014\,[-0.027,\ 0.082]$	$+0.055\,[-0.003,\ 0.142]$	$-0.012\,[-0.103,\ 0.068]$
CSOAA	1.0	$+0.099\,[0.007,\ 0.217]$	$+0.117\,[0.027,\ 0.256]$	$+0.013\,[-0.138,\ 0.187]$
MLP	0.1	$+0.100\,[-0.028,\ 0.246]$	$+0.093\,[-0.071,\ 0.261]$	$+0.074\,[0.004,\ 0.167]$
MLP	0.5	$+0.185\,[0.019,\ 0.359]$	$+0.021\,[-0.133,\ 0.192]$	$+0.067\,[-0.088,\ 0.225]$
MLP	1.0	$+0.135\,[-0.066,\ 0.349]$	$+0.060\,[-0.123,\ 0.239]$	$+0.102\,[-0.119,\ 0.317]$

At $\alpha = 1$, all six utility point estimates are positive, consistent with the full-policy gains in the main text. Across all target weights, 16 of 18 estimates are positive. Four positive intervals and one negative interval exclude zero; the remaining 13 include zero. These task-level intervals measure uncertainty conditional on fitted policies, whereas the main-text standard deviations describe variation across training-seed/holdout pairs.

E Historical Sampled-Continuation Comparison

A fair coin chooses the reference or current-learner controller once per suffix, unlike the held-out study's arithmetic targets. Each training seed is evaluated on the same reused 100-task panel. The task-level decomposition is descriptive and computed after observing outcomes; aggregate estimates retain all tasks. Tables 7 and 8 report aggregate outcomes and paired task changes.

Table 7: Historical matched learned-target comparison on one reused 100-task panel. Values are mean $\pm$ sample SD across three training seeds.

Training target	Utility	Success
Reference continuation L_R	1.416 ± 0.144	0.503 ± 0.055
50–50 mix-per-roll L_{M50}	1.312 ± 0.186	0.480 ± 0.070
Mean difference	-0.104	-0.023

Table 8: Descriptive task-level decomposition of the historical L_{M50} versus L_R comparison. "Improved/unchanged/worse" refer to utility; repairs and disruptions refer to binary success.

Seed	Improved	Unchanged	Worse	Repairs	Disruptions
19	12	53	35	3	7
29	22	56	22	1	2
39	7	58	35	0	2