POLICY IMPROVEMENT AS DECISION REVISION: PRESERVING THE INCUMBENT UNDER EXECUTABLE FEEDBACK

Anonymous authors
Paper under double-blind review

ABSTRACT

Policy improvement requires both what is better and what is being revised. Executable counterfactual evaluation provides utilities for alternative actions, but utilities alone do not identify the incumbent decision. We identify a supervision-interface problem: the evaluation object is an *evaluated decision*, (p_{old}, Q), pairing the incumbent action distribution with executable utilities over a common semantic-action support. The projection $(p_{\mathrm{old}}, Q) \mapsto Q$ is many-to-one: any Q-only supervision is invariant to the incumbent and loses its identity before optimization begins. This motivates a representation principle: **learn a revision of the evaluated decision relative to its incumbent.** We distinguish *revision specification*, which defines the intended target, from *revision realization*, which fits that target with a finite model. Distributional Decision Revision (DiRe) provides a deliberately standard instantiation, using an established KL-regularized target and a reference-relative policy family. The contribution is the interface formulation and its controlled identification.

We test the principle through controlled interventions at both stages. With executable utilities fixed, changing the incumbent leaves Q-only targets unchanged but changes the DiRe revision. On matched evaluated decisions, Hard-Q, utility-only supervision, and DiRe encode similar expected executable improvements (+0.0645, +0.0625, and +0.0638) while inducing substantially different incumbent-relative displacements (JS 0.1101, 0.1554, and 0.0688): executable gain alone does not characterize the revision. Holding the DiRe target, model family, cross-entropy objective, optimization budget, and initialization fixed, using the evaluated incumbent rather than a uniform realization reference reduces target–policy JS from 0.1313 to 0.0572 and changes realized executable improvement from -0.0162 to +0.0053 on 59 state-held-out decisions. Exact reconstruction establishes representability in principle; a semantic residual scorer realizes 50.9% of intended improvement on a later state-disjoint batch from a development-exposed conversation. These results support incumbent identity as part of the revision problem at both stages, while finite-model realization remains a distinct and heterogeneous bottleneck. Independent conversation-level generalization remains untested.

1 INTRODUCTION

Language-model agents learn from executable feedback on search, tool use, and web interaction (Jin et al., 2025; Qi et al., 2025; Luo et al., 2025; Zhang et al., 2024). Policy improvement involves two questions: *what is better*, and *what is being revised*? Evaluating alternative actions supplies executable utilities Q for the first question. It does not identify the incumbent action distribution p_{old} needed for the second.

Consider two agents facing the same actions and utilities. One already concentrates probability on high-utility actions; the other favors low-utility actions. The evidence may confirm much of the first decision while calling for a larger revision of the second. We therefore take the supervision object to be the *evaluated decision*, $\mathcal{D}(s) = (p_{\mathrm{old}}(\cdot \mid s), Q(s, \cdot))$, on a common semantic-action support. The projection $(p_{\mathrm{old}}, Q) \mapsto Q$ is many-to-one: distinct evaluated decisions become indistinguishable to

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.

any Q-only supervision mapping. This loss of incumbent identity occurs before optimization begins. It does not preclude learning useful actions from Q; it prevents that supervision from identifying which decision is being revised.

The incumbent's role here is distinct from its familiar use as a regularization reference. A reference can constrain optimization; the evaluated incumbent additionally identifies the revision problem itself. Even exact optimization cannot recover that identity from Q alone. This motivates a representation principle: preserve (p_{old}, Q) when specifying the revision, then retain the incumbent when realizing it. We separate the pipeline into *revision specification*, $(p_{\mathrm{old}}, Q) \mapsto q^*$, and *revision realization*, $q^* \mapsto \pi_\theta$. Whether the specified revision survives finite fitting is a separate empirical question.

DiRe (**Di**stributional Decision **Re**vision) provides a deliberately standard instantiation: an established KL-regularized target (Peters et al., 2010; Abdolmaleki et al., 2018; Geist et al., 2019; Tomar et al., 2022) and an incumbent-relative residual parameterization. Neither the operator nor the function family is new. Our contribution is the supervision-interface problem, its specification/realization decomposition, and controlled identification of incumbent effects within that decomposition.

Fixed-Q interventions test what information reaches specification; fixed-target interventions test the realization reference independently of target construction. Exact reconstruction and richer-scorer analyses then characterize the finite realization gap. These support controlled and state-level holdout evidence; prior exposure of all available LoCoMo groups leaves independent conversation-level generalization untested.

Our contributions are:

- **Supervision-interface problem.** We identify how projecting evaluated decisions onto Q discards incumbent identity before optimization; an elementary formal result makes this identification limit explicit.
- **Two-stage formulation.** We separate revision specification from finite-model realization and use DiRe as a standard instantiation preserving the evaluated incumbent in both stages.
- **Controlled identification.** Fixed-Q and fixed-target interventions isolate incumbent effects; reconstruction, richer realization, and stress tests characterize the remaining gap and its inference scope.

2 From Executable Evidence to Decision Revision

Tool, retrieval, web, and search agents turn executable outcomes into post-training supervision (Schick et al., 2023; Asai et al., 2024; Jin et al., 2025; Qi et al., 2025; Zhang et al., 2024; Luo et al., 2025). We study the interface between these operations: after alternatives at the same state have been evaluated, what information must be retained to specify a revision of the evaluated decision?

2.1 Evaluated Decisions

At state s, let $\mathcal{A}(s) = \{a_1, \ldots, a_K\}$ denote a runtime-defined set of semantic actions. An action may correspond to issuing a retrieval query, invoking a tool, interacting with an interface element, or committing to a response. Before incorporating new evidence, the frozen policy induces an incumbent distribution $p_{\mathrm{old}}(a \mid s)$ for $a \in \mathcal{A}(s)$. This is the frozen policy's actual distribution over the same runtime-defined actions later subjected to executable evaluation, not an arbitrary prior chosen for regularization. It records the decision to be revised, including probability mass on alternatives to the selected action.

We evaluate each candidate by intervening on the current decision while holding downstream behavior fixed. Under a common continuation policy π_{cont}, define

$$Q(s, a) = \mathbb{E}\left[R(\tau) \mid s, a, \pi_{\mathrm{cont}}\right]. \tag{1}$$

Here a is fixed at the current decision and subsequent actions follow the same continuation protocol. This resembles value- and process-based supervision in assigning consequences to intermediate decisions (Zhang et al., 2024; Luo et al., 2025), but our use of Q is deliberately local: it measures the

consequence of changing the current decision under a fixed continuation and need not be a globally optimal action-value function.

We call the aligned incumbent and executable evidence an *evaluated decision*,

$$\mathcal{D}(s) = \{p_{\text{old}}(a \mid s), Q(s,a)\}_{a \in \mathcal{A}(s)}. \tag{2}$$

2.2 From Evaluation to Post-Training Supervision

A broad family of post-training interfaces converts evaluated alternatives into selected or reweighted actions (Peters & Schaal, 2007; Peng et al., 2019), pairwise preferences (Rafailov et al., 2023; Calandriello et al., 2024), or selected trajectories and policy updates (Qi et al., 2025; Zhang et al., 2024; Luo et al., 2025).

These methods need not use Q-only supervision. Our identification result concerns a mapping restricted to Q as its input, which is invariant to the incumbent distribution.

To make this distinction explicit, consider two evaluated decisions with the same executable evidence, $\mathcal{D}_1 = (p_1, Q)$ and $\mathcal{D}_2 = (p_2, Q)$ with $p_1 \neq p_2$. The same evidence may largely confirm one incumbent while implying a larger revision of the other. The following elementary observation formalizes the information lost before any choice of policy-improvement operator.

Proposition 1 (Incumbent non-identifiability under Q-only supervision)**.** *Let $\rho(p, Q) = Q$ project an evaluated decision onto its executable evidence, and let T be any supervision mapping whose input is only Q. For two decisions (p_1, Q) and (p_2, Q) on the same action support with $p_1 \neq p_2$,*

$$(T \circ \rho)(p_1, Q) = (T \circ \rho)(p_2, Q) = T(Q). \tag{3}$$

Thus the resulting supervision object cannot identify which incumbent was evaluated.

Proof. Both decisions project to the same Q, so T receives the same input. □

For intended distributions $R_i = R(p_i, Q)$, $i \in \{1, 2\}$, a Q-only rule obeys $\max_i d(T(Q), R_i) \geq \frac{1}{2} d(R_1, R_2)$ for a metric d: separated revisions cannot both be reproduced without incumbent information (Corollary 1, Appendix A). This bounds revision error, not achievable utility.

2.3 Relation to Policy-Relative Improvement

REPS, MPO, mirror descent, and regularized MDPs already use policy-relative or KL-regularized updates (Peters et al., 2010; Abdolmaleki et al., 2018; Geist et al., 2019; Tomar et al., 2022). AWR uses advantage-weighted policy fitting, and DPO uses a reference policy (Peng et al., 2019; Rafailov et al., 2023; Calandriello et al., 2024). Separating improvement from parametric fitting is also familiar in MPO and weighted regression (Abdolmaleki et al., 2018; Peters & Schaal, 2007; Peng et al., 2019). We use this separation to study an interface question before operator choice: which information from executable evaluation identifies the revision problem? Using the current policy as reference is compatible with our principle. Our distinction is the evaluated incumbent's identity and its controlled intervention at specification and realization, not the first use of a reference policy or a two-stage update.

3 Distributional Decision Revision

DiRe instantiates this interface in two stages (Figure 1): *revision specification* maps the evaluated decision (p_{old}, Q) to a target q^*; *revision realization* fits that target with a finite policy π_θ relative to the same incumbent. Their separation lets us test where reference identity matters and where the finite model fails to realize the specified revision.

3.1 Incumbent-Relative Revision Specification

For valid actions $a_i \in \mathcal{A}_{\text{v}}(s)$, let $p_i = p_{\text{old}}(a_i \mid s)$ and $Q_i = Q(s, a_i)$. DiRe defines the revised decision by

$$q^*_\beta = \arg \max_{q \in \Delta(\mathcal{A}_{\text{v}}(s))} \left\{ \sum_i q_i Q_i - \beta D_{\text{KL}}(q \| p) \right\}, \qquad \beta > 0, \tag{4}$$

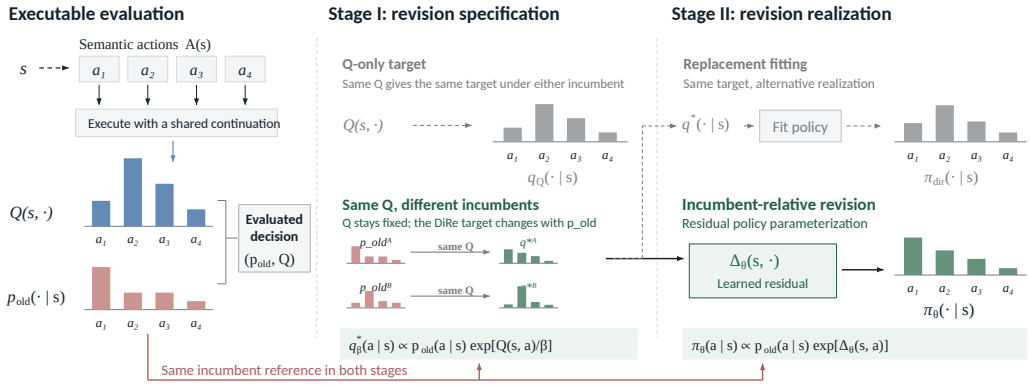


Figure 1: **DiRe overview.** With fixed Q, the Q-only target remains unchanged across incumbents, whereas DiRe targets change with the incumbent. Stage II compares incumbent-relative revision with replacement fitting under the same DiRe target. Gray paths mark comparisons; bars are schematic.

where the utility term favors actions with higher executable returns and the KL term controls displacement from the identified incumbent. Retaining incumbent identity identifies which decision is being revised; this particular regularizer determines a tradeoff for that revision. Replacing p by a reference distribution p_{ref} on valid support gives the general solution

$$q^*_{\beta,p_{\mathrm{ref}}}(a \mid s) = \frac{p_{\mathrm{ref}}(a \mid s)\exp(Q(s,a)/\beta)}{\sum_{b\in\mathcal{A}_{\mathrm{v}}(s)} p_{\mathrm{ref}}(b \mid s)\exp(Q(s,b)/\beta)}. \tag{5}$$

DiRe takes $p_{\mathrm{ref}} = p_{\mathrm{old}}$ and writes $q^*_\beta = q^*_{\beta,p_{\mathrm{old}}}$. Equivalently, $\log q^*_\beta(a \mid s) = \log p_{\mathrm{old}}(a \mid s) + Q(s,a)/\beta - \log Z_\beta(s)$, so executable evidence acts as an additive correction to incumbent log probability. The target is therefore a revision of p_{old}, rather than a distribution constructed from Q alone.

The exponential tilt is established; its base measure here is the observed incumbent over the evaluated actions. Proposition 1 motivates retaining that identity but does not uniquely determine this operator. Appendix A provides its derivation and boundary properties.

3.2 Revision Geometry

The dependence on the incumbent is explicit in the revised odds:

$$\log\frac{q^*_\beta(a_i \mid s)}{q^*_\beta(a_j \mid s)} = \log\frac{p_{\mathrm{old}}(a_i \mid s)}{p_{\mathrm{old}}(a_j \mid s)} + \frac{Q_i - Q_j}{\beta}. \tag{6}$$

Thus identical executable evidence can induce different revised decisions when applied to different incumbent distributions. This property directly motivates the same-Q/different-incumbent intervention.

The target also recovers the incumbent when the evidence is uninformative: if $Q_1 = \cdots = Q_K$, then $q^*_\beta(\cdot \mid s) = p_{\mathrm{old}}(\cdot \mid s)$. More generally, $q^*_\beta \to p_{\mathrm{old}}$ as $\beta \to \infty$ and $q^*_\beta \to \delta_{a^*}$ as $\beta \to 0^+$ for a unique executable maximizer a^* with positive incumbent probability. Thus β traces an incumbent-relative path from preserving the current decision toward following executable evidence. This is target geometry, not a new tuning algorithm. Finite parametric fitting is a separate problem.

3.3 Incumbent-Relative Realization: Learning a Revision

A direct replacement models $\pi_\theta = \mathrm{softmax}\, f_\theta$ and fits the fixed target by cross-entropy,

$$\mathcal{L}_{\mathrm{direct}}(\theta) = -\frac{1}{|\mathcal{S}_{\mathrm{tr}}|}\sum_{s\in\mathcal{S}_{\mathrm{tr}}}\sum_{a\in\mathcal{A}_{\mathrm{v}}(s)} q^*_\beta(a \mid s)\log\pi_\theta(a \mid s). \tag{7}$$

Direct fitting already receives incumbent information through q^*_β; it is not a Q-only target. Stage II asks whether representing the incumbent explicitly as the policy's base measure helps realize that fixed target. DiRe represents the *revision* itself.

For DiRe, define the target log-residual

$$r^*(a \mid s) = \log\big(q^*_\beta(a \mid s) + \epsilon\big) - \log(p_{\mathrm{old}}(a \mid s) + \epsilon)\,. \tag{8}$$

A reference-relative scorer $f_\theta(s, a)$ reconstructs a policy as

$$\pi_\theta^{(p_{\mathrm{ref}})}(a \mid s) = \frac{p_{\mathrm{ref}}(a \mid s)\exp(f_\theta(s, a))}{\sum_{b \in \mathcal{A}_{\mathrm{v}}(s)} p_{\mathrm{ref}}(b \mid s)\exp(f_\theta(s, b))}. \tag{9}$$

DiRe sets $p_{\mathrm{ref}} = p_{\mathrm{old}}$ and $f_\theta = \Delta_\theta$, writing $\pi_\theta = \pi_\theta^{(p_{\mathrm{old}})}$ and learning a change to the incumbent decision; a uniform p_{ref} gives direct softmax fitting. RQ2 holds the target and fitting protocol fixed while changing only this realization reference. The family itself is an established parameterization; our contribution is its use to preserve and test incumbent identity in decision revision. It can be fitted with different objectives. The historical **Residual-DiRe** variant centers the predicted and target log-residuals over the valid support and minimizes their incumbent-weighted squared error,

$$\mathcal{L}_{\mathrm{res}}(\theta) = \frac{1}{|\mathcal{S}_{\mathrm{tr}}|} \sum_{s \in \mathcal{S}_{\mathrm{tr}}} \sum_{a \in \mathcal{A}_{\mathrm{v}}(s)} p_{\mathrm{old}}(a \mid s)\,\big(\bar{\Delta}_\theta(s, a) - \bar{r}^*(s, a)\big)^2\,, \tag{10}$$

where $\bar{\Delta}_\theta$ and $\bar{r}^*$ denote the corresponding centered residuals. **Residual-CE** instead minimizes Eq. 7 using Eq. 9 with $p_{\mathrm{ref}} = p_{\mathrm{old}}$. The fixed-target reference intervention and the expanded semantic scorer use this CE formulation; the squared-error objective is not required by DiRe.

With $p_{\mathrm{ref}} = p_{\mathrm{old}}$, zero residual recovers p_{old} and the target residual r^* recovers q^*_β, up to the numerical stabilization in Eq. 8. The two-stage pipeline is

$$\underbrace{(p_{\mathrm{old}}, Q)}_{\text{evaluated decision}} \xrightarrow{\text{revision specification}} q^*_\beta \xrightarrow[\text{retain } p_{\mathrm{old}}]{\text{revision realization}} \pi_\theta. \tag{11}$$

A DiRe target specified relative to the incumbent is therefore also *parameterized* relative to that incumbent during policy realization. The boundary conditions above characterize the representation; whether finite parametric fitting preserves the intended revision is an empirical question studied in Section 4.

Implementation. Executable evaluation precedes parameter optimization. For each state, the candidate inventory, valid-action mask, p_{old}, Q, q^*_β, and r^* are fixed before Stage II. All quantities share the same valid semantic support, and invalid actions receive zero probability. Target and optimization hyperparameters are selected using development data and frozen before test evaluation. Full support handling, optimization, and selection details are provided in Appendix B.

4 Experiments

We ask three questions: **RQ1**, what information and revision geometry are lost by Q-only target construction? **RQ2**, does reference identity matter under a fixed target and common CE protocol? **RQ3**, how much of the specified revision survives finite fitting? Primary comparisons share the corrected V2 protocol and immutable valid semantic support. RQ1 tests target specification, RQ2 isolates the realization reference, and RQ3 studies the remaining realization bottleneck.

4.1 Experimental Setup

The primary study uses the corrected V2 protocol with 59 held-out decision states, nine canonical candidate slots per state, and five matched seeds. The 13 disjoint DEV states are also the fitting set; there is no independent training/validation split. We freeze each candidate inventory, valid semantic support, incumbent distribution, and executable utilities obtained by forcing valid actions under a common continuation. Every method thus operates on matched evaluated decisions (p_{old}, Q). This design supports the controlled interface question, not broad external validity. Separately, on 72

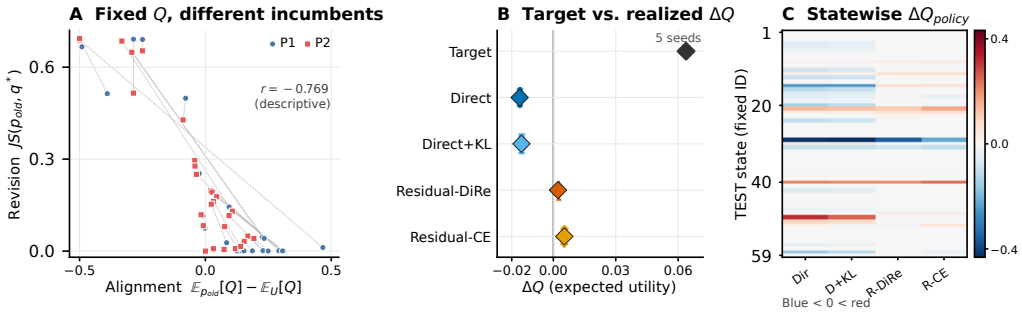


Figure 2: **Incumbent identity and finite realization on 59 TEST states. (A)** Paired fixed-Q incumbent intervention; the pooled correlation is descriptive. **(B)** The black diamond is the nonparametric target gain; small colored marks are five seed means and large diamonds are their average. **(C)** State-level ΔQ_{policy} averaged over seeds, sorted by fixed ID, with a symmetric zero-centered color scale. The states come from previously exposed conversations.

matched questions near a 1,250-token budget, category routing beats lexical retrieval by $+0.0833$ (conversation-cluster 95% CI $[+0.0417, +0.1250]$), validating task relevance of the decision interface rather than DiRe (Appendix C.1).

At the target level, we compare Selected-SFT, Hard-Q, Utility-only, $q(a \mid s) \propto Q(s,a)$ on valid actions (uniform if their utility sum is zero), and DiRe, $q(a \mid s) \propto p_{\text{old}}(a \mid s) \exp(Q(s,a)/\beta)$. Target construction and policy realization are evaluated separately. For the same fixed DiRe target, we compare direct fitting, direct fitting with a development-selected incumbent KL penalty (Direct+KL), and incumbent-relative residual realization. Residual-CE uses the same residual parameterization under cross-entropy to separate the realization reference from the original residual fitting objective.

We measure target and policy executable improvement, incumbent-relative JS, target–policy JS and forward KL, and selected-action outcomes. The inherited frozen target temperature is $\beta = 0.03$. Optimization is selected on DEV and frozen before state-disjoint evaluation; principal comparisons use paired state-bootstrap intervals. Appendix B details support, splits, selection, metrics, and reproducibility.

4.2 RQ1: Does Discarding the Incumbent Lose Revision Information?

RQ1 tests revision specification in two steps: whether the supervision identifies the incumbent, and whether executable gain alone characterizes the revision. We first hold Q, valid semantic support, terminal outcomes, and continuation fixed, intervening only on the incumbent. Hard-Q and Utility-only must remain unchanged because they use only Q; DiRe can respond because it retains the evaluated decision. No policy fitting enters this comparison, so the contrast concerns information available before optimization.

For each state, let $P1$ denote the original incumbent. We construct a pre-specified counter-aligned incumbent

$$P2(a \mid s) \propto \mathbf{1}_{\text{valid}}(a) P1(a \mid s) \exp(-Q(s,a)/0.10)\,, \tag{12}$$

while holding Q, the valid semantic support, and terminal outcomes fixed. This intervention changes incumbent–utility alignment from $+0.0467$ under $P1$ to -0.0296 under $P2$. Alignment is expected Q under the incumbent minus expected Q under the uniform distribution on valid actions.

Table 1 gives the invariance contrast: Q-only targets remain identical, whereas the counter-aligned incumbent induces a larger DiRe revision and target-level executable improvement.

Table 1: **Fixed Q, different incumbents.** Δ_{ℓ_1} is the maximum change in any Q-only target; $JS(P, q^*)$ and ΔQ_{target} describe the DiRe revision.

Incumbent	Δ_{ℓ_1}	$JS(P, q^*)$	ΔQ_{target}
Original $P1$	0	0.0688	+0.0638
Counter-aligned $P2$	0	0.1365	+0.1380

Figure 2A provides a complementary state-level view of this intervention. The pooled Pearson correlation is -0.769, while only 50.8% of states have larger DiRe revision under $P2$ than $P1$. We treat these correlations as descriptive rather than a statewise monotonicity guarantee; the identifying result is the controlled invariance contrast above: Q-only targets cannot respond when only the incumbent changes, whereas DiRe does.

Table 2: **Matched target revisions.** Same 59 states and executable Q; Top Chg. marks changed target actions.

Target	ΔQ_{target}	$JS(p_{\text{old}}, q)$	Top Chg.
Hard-Q	+0.0645	0.1101	20.3%
Utility-only	+0.0625	0.1554	20.3%
DiRe	+0.0638	0.0688	15.3%

Matched target revision geometry. The targets yield similar executable gains but different incumbent-relative revisions. DiRe's paired JS reductions versus Hard-Q and Utility-only are -0.0414 (95% CI $[-0.0779, -0.0113]$) and -0.0866 ($[-0.1122, -0.0597]$). Its target utility is 0.0007 lower than Hard-Q; this is not an equivalence claim.

The shared states and executable Q exclude different evidence as the source of this contrast: similar gain can encode substantially different revisions (Appendix C.17). Smaller displacement here describes the revision geometry, not a general preference for conservative updates.

Executable-action diagnostic. On 200 MiniWoB states, Hard-Q, Utility-only, and DiRe have identical success (0.540) but incumbent-relative JS of 0.4346, 0.2860, and 0.2596. Thus, equal discrete outcomes can mask different decision distributions. Separate per-state fitting makes this an in-panel diagnostic, not a held-out benchmark (Appendix C.8, Table 16).

4.3 RQ2: Revision versus Replacement under a Fixed Target

Reference-identity intervention. To test realization independently of target construction and changes in the fitting protocol, we hold q^*, the linear scorer, CE loss, initialization, Adam optimizer, valid support, DEV/TEST split, and training budget fixed, intervening only on p_{ref} in $\pi_\theta \propto p_{\text{ref}} \exp f_\theta$. The references are the aligned incumbent, uniform on valid actions, and a counter-aligned distribution proportional to $p_{\text{old}} \exp(-10Q)$; q^* is never recomputed. Learning rate 0.03 and 440 epochs were selected on aligned DEV and shared across all three references and five matched seeds, not separately tuned by reference (Appendix C.9).

Table 3: Fixed-target reference identity under the same CE objective and budget. Top Agree is target–policy argmax agreement.

Reference	$JS(p_{\text{old}}, \pi)$	$JS(q^*, \pi)$	$KL(q^* \Vert \pi)$	ΔQ_{policy}	Top Agree
Aligned incumbent	0.0336	0.0572	0.5401	0.0053	91.5%
Uniform	0.1006	0.1313	0.5208	-0.0162	74.9%
Counter-aligned	0.0697	0.1100	1.3870	-0.0540	80.7%

Aligned (Residual-CE) has the lowest target JS and the only positive realized utility; uniform reproduces DEV-selected Direct (Table 3). The fixed target, CE, and protocol exclude changes in specification, loss, or budget as explanations. Counter-aligned is closer to the incumbent than uniform but has worse utility, so displacement alone does not order these arms. Aligned forward KL is slightly higher than uniform; the gains do not extend to every metric.

Exact-displacement control. A post-hoc control calibrates a DEV-selected Direct-CE scorer statewise to Residual-CE's incumbent displacement on the same 59 evaluation states.

Table 4: **Paired exact-displacement outcomes.** Both mean $JS(p_{\text{old}}, \pi) = 0.03358$; median per-state matching error is 3.16×10^{-17}.

Metric	Residual-CE	Direct-CE
$JS(q^*, \pi)$	0.05721	0.06489
ΔQ_{policy}	+0.00528	+0.00174

Residual-CE retains directionally lower target JS (difference -0.00768, 95% CI $[-0.02218, +0.00120]$) and higher realized utility ($+0.00354$, $[-0.00045, +0.01012]$). Both intervals include zero; generic proximity is not ruled out (Table 4; Appendix C.10).

4.4 RQ3: What Limits Finite Realization?

We next test whether a specified revision survives finite fitting. For the fixed target ($\Delta Q_{\text{target}} = +0.0638$), DEV-selected Direct realizes -0.0162. At the historical fixed Direct endpoint, 20/59 states (33.9%) reverse the intended sign; mean realization loss is -0.0811 (median -0.0141; 95% CI $[-0.1344, -0.0377]$). The gap is not due only to a few extreme states (Figure 2B–C; Appendix C.3).

Table 5 compares direct fitting, a DEV-selected incumbent-KL penalty, and two residual objectives. Direct+KL remains negative: adding a proximity penalty does not reproduce the residual outcome within this search. Residual-CE shares Direct's CE; RQ2 also fixes the budget. The controls do not exhaust all regularizers or optimization schedules.

Table 5: **Realizing the same DiRe target.** Means over 59 TEST states and five seeds; target $\Delta Q = +0.0638$ and $JS(p_{\text{old}}, q^*) = 0.0688$. Repair/Reg./Net are counts (Net = Repair − Reg.); Top Agree is target–policy argmax agreement. Fixed direct is the historical 220-epoch endpoint (Figure 3); Figure 2B–C uses DEV-selected learned policies.

	Selected-Action Outcome				Distributional Realization				
Method	**Sel. Q**	**Repair**	**Reg.**	**Net**	$\mathbf{JS}(\mathbf{p}_{\text{old}}, \pi)$	$\mathbf{JS}(\mathbf{q}^*, \pi)$	$\mathbf{KL}(\mathbf{q}^* \Vert \pi)$	$\mathbf{\Delta Q}_{\text{policy}}$	**Top Agree**
Fixed direct	0.3535	3.6	1.0	+2.6	0.0890	0.1237	0.4679	-0.0173	75.6%
Direct	0.3684	3.2	1.0	+2.2	0.1006	0.1313	0.5208	-0.0162	74.9%
Direct+KL	0.3583	2.0	1.0	+1.0	0.0847	0.1211	0.4868	-0.0153	72.9%
Residual-DiRe	0.3712	4.0	0.0	+4.0	0.0185	0.0587	0.5933	+0.0022	91.5%
Residual-CE	0.3731	5.0	0.0	+5.0	0.0336	0.0572	0.5401	+0.0053	91.5%

Realization controls and consequences. Relative to fixed Direct, Residual-DiRe improves expected utility by $+0.0195$ (95% CI $[+0.0038, +0.0355]$) and lowers target-policy JS by -0.0650 ($[-0.0973, -0.0304]$). Selected-action gains remain inconclusive, so evidence is stronger for distributional realization. Matched DEV optimization and exact-residual checks weaken the original-schedule explanation and rule out algebraic error, but do not establish global optimization superiority (Appendices C.7, C.5, and C.6).

Post-hoc loss correlates with Q margin and revision magnitude, but revision-stratified comparisons do not show a monotone residual advantage (Appendices C.4 and C.14). These diagnostics do not establish lower function-class complexity. Lower mean JS also coexists with heavy forward-KL tails (Table 8), leaving target coverage unresolved.

Representability and richer realization. On historical DEV, analytical residual reconstruction error is 1.19×10^{-7} and free-logit $JS(q^*, \pi)$ is approximately 3.7×10^{-8}. These checks exclude an algebraic or unrestricted statewise representability barrier; they do not establish generalization. The richer CE-trained semantic scorer realizes 50.9% of intended gain on previously unused states from development-exposed `conv-44` (Table 6). This shows substantial state-level realization, but changed features and splits preclude a matched capacity ablation (Appendix C.12).

Table 6: **Richer-scorer realization.** 24 state-disjoint states from development-exposed `conv-44`; five seeds. Policy entries are mean ± seed SD.

Metric	Result
Target ΔQ	+0.08259
Policy ΔQ	$+0.04202 \pm 0.00981$
Target gain realized	50.9%
$JS(q^*, \pi)$	0.06655 ± 0.01852
Top-action agreement	87.5%

Table 7: **Selected DEV representation controls.** Five-seed means for model selection, not independent evaluation.

Representation / objective	$JS(q^*, \pi)$	ΔQ	$KL(q^* \| \pi)$	Agree	Severe
Base MLP-64, CE	.06784	+.01371	.600	.883	0
Candidate-ID MLP-128, CE	.05715	+.01818	.629	.917	0
Utility-pairwise MLP-128	.04526	+.02137	1.769	.938	0
Qwen-semantic MLP-128, CE	.04953	+.03658	.346	.900	0

Utility-pairwise minimizes JS, while Qwen-semantic has the highest executable utility and lowest forward KL among these DEV controls; model selection is metric-dependent (Appendix C.12).

The DEV miss counts, `conv-44` batch, and historical-panel tails below are separate diagnostics and establish no conversation-level transfer.

Table 8: **Forward-KL tails.** Historical 59-state TEST panel at $\beta = 0.03$. Values are state means after averaging five seeds; P95 and maximum are computed over all states without tail filtering.

Method	Mean	Median	P95	Max.
Direct	0.5208	0.2479	2.0521	4.9203
Direct+KL	0.4868	0.2173	2.1669	4.7990
Residual-DiRe	0.5933	0.0004	4.1366	12.1175
Residual-CE	0.5401	0.0080	5.6641	9.9491

Historical-panel target coverage. Residual fits have lower median forward KL than Direct but much heavier tails. The largest residual miss assigns about 0.999598 target mass to one candidate while one seed's policy assigns only 4.75×10^{-6}. Bounded-distance gains therefore do not guarantee coverage of strongly intended revisions (Appendix C.11).

Retrospective cross-conversation stress test. Across six development-exposed conversations, five have positive and one negative mean realized utility. Equal-conversation realization is $+0.00835$ (13.0%; descriptive conversation-bootstrap 95% CI $[+0.00119, +0.01592]$); mean target-policy JS is 0.06627 versus 0.05929 for the incumbent. The negative group has target $\Delta Q = +0.12356$ but realized $\Delta Q = -0.00462$. Positive realization thus occurs beyond the historical panel, but utility and fit remain heterogeneous. Prior exposure makes this a retrospective robustness diagnostic, not independent generalization (Appendix C.13, Figure 4).

5 Conclusion

Executable utilities describe what is better; the incumbent identifies what is being revised. Q-only supervision collapses distinct evaluated decisions before optimization. DiRe preserves (p_{old}, Q) through revision specification and realization. Fixed-Q and fixed-target interventions identify incumbent effects at both stages under the tested protocol. Exact reconstruction and semantic realization distinguish representability from finite fitting. The favorable but inconclusive exact-displacement control leaves generic proximity unresolved; heterogeneous utility and forward-KL tails limit claims of general algorithmic superiority.

Limitations. Results use frozen incumbents, valid candidates/support, and shared continuations; candidate or utility errors, changing actions, adaptation, and noisy evidence remain untested. Zero-support actions cannot gain target mass; smoothing is untested. All ten LoCoMo groups were exposed, so neither the 59-state panel, later `conv-44` batch, nor six-group analysis establishes conversation-level generalization. Five seeds measure optimization, not conversation replication; MiniWoB is in-panel. The fixed-target control does not separate approximation floor from optimization, and gauge-dependent scales cannot resolve this (Appendix C.14).

Future work. Future studies should test the frozen pipeline on untouched conversations with equal-conversation reporting and independent fitting/selection groups. Matched function-class controls should separate approximation from optimization; support smoothing and tail-aware realization should address excluded actions and undercoverage. Natural decisions with uncertain executable evidence also remain to be tested (Appendix C.15).

STATEMENTS

AI use statement. Generative AI was used to polish the language of the manuscript. The authors reviewed the resulting text and take responsibility for the final submission.

Ethics statement. The study uses existing benchmark and simulation data and collects no new human-subject data. It does not target high-stakes decisions. Incorrect candidate actions or utility estimates could lead to undesirable revisions; claims are limited to the evaluated settings.

Reproducibility statement. An anonymous release of experiment scripts and compact reference outputs is available at the anonymous repository; it does not include conversation data, model weights, or cached traces. The main text specifies the 59-state TEST protocol, matched comparisons, and five-seed averaging. Appendix B details target construction, valid-support handling, configurations, selection, and reproducibility checks; Appendix C.15 records conversation exposure. Model and optimizer choices are selected on DEV and frozen before TEST evaluation.

REFERENCES

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Rémi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. In *International Conference on Learning Representations*, 2018.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations*, 2024.

Daniele Calandriello, Zhaohan Daniel Guo, Remi Munos, Mark Rowland, Yunhao Tang, Bernardo Avila Pires, Pierre Harvey Richemond, Charline Le Lan, Michal Valko, Tianqi Liu, Rishabh Joshi, Zeyu Zheng, and Bilal Piot. Human alignment of large language models through online preference optimisation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 5409–5435, 2024.

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized Markov decision processes. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2160–2169, 2019.

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

Xufang Luo, Yuge Zhang, Zhiyuan He, Zilong Wang, Siyun Zhao, Dongsheng Li, Luna K. Qiu, and Yuqing Yang. Agent lightning: Train any ai agents with reinforcement learning. *arXiv preprint arXiv:2508.03680*, 2025.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Jan Peters and Stefan Schaal. Reinforcement learning by reward-weighted regression for operational space control. In *Proceedings of the 24th International Conference on Machine Learning*, pp. 745–750, 2007. doi: 10.1145/1273496.1273590.

Jan Peters, Katharina Mülling, and Yasemin Altun. Relative entropy policy search. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, pp. 1607–1612, 2010. doi: 10.1609/aaai.v24i1.7727.

Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Jiadai Sun, Xinyue Yang, Yu Yang, Shuntian Yao, Wei Xu, Jie Tang, and Yuxiao Dong. WebRL: Training LLM web agents via self-evolving online curriculum reinforcement learning. In *International Conference on Learning Representations*, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741, 2023.

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.

Manan Tomar, Lior Shani, Yonathan Efroni, and Mohammad Ghavamzadeh. Mirror descent policy optimization. In *International Conference on Learning Representations*, 2022.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. ReST-MCTS*: LLM self-training via process reward guided tree search. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

A Properties of Distributional Decision Revision

This section collects the basic properties of the distributional revision defined in Section 3. For a fixed state, let $p = (p_1, \dots, p_K)$ denote the incumbent distribution, $Q = (Q_1, \dots, Q_K)$ the executable utilities, and

$$\mathcal{I}_p = \{i : p_i > 0\}$$

the support of the incumbent.

Corollary 1 (Unavoidable error for distinct intended revisions)**.** *For intended distributions $R_i = R(p_i, Q)$ on the same support with $R_1 \neq R_2$, any distribution-valued Q-only rule T and metric d obey* $\max_{i\in\{1,2\}} d(T(Q), R_i) \geq \frac{1}{2} d(R_1, R_2)$.

Proof. By the triangle inequality, $d(R_1, R_2) \leq \sum_i d(T(Q), R_i)$. □

The bound assumes no other incumbent input to T and concerns revision error in a metric (e.g., total variation), not utility, KL, or unrooted JS.

A.1 Optimality and Uniqueness

Proposition 2 (Unique distributional revision)**.** *For any $\beta > 0$, the objective in Eq. 4 has a unique optimum on the simplex over $\mathcal{I}_p$, given by*

$$q_i^* = \frac{p_i \exp(Q_i/\beta)}{\sum_{j\in\mathcal{I}_p} p_j \exp(Q_j/\beta)}. \tag{13}$$

Proof. The utility term $\sum_i q_i Q_i$ is linear in q, while $D_{\mathrm{KL}}(q\|p)$ is strictly convex over the relative interior of the simplex on $\mathcal{I}_p$. Hence

$$\sum_i q_i Q_i - \beta D_{\mathrm{KL}}(q\|p)$$

is strictly concave for $\beta > 0$, so its stationary point is the unique global maximizer.

Introducing a multiplier λ for $\sum_i q_i = 1$ gives

$$Q_i - \beta\left(\log(q_i/p_i) + 1\right) + \lambda = 0.$$

Thus

$$q_i \propto p_i \exp(Q_i/\beta),$$

and normalization over $\mathcal{I}_p$ yields Eq. 13. □

A.2 Revision in Log-Odds Space

Proposition 3 (Executable evidence revises incumbent odds)**.** *For any $i, j \in \mathcal{I}_p$,*

$$\log \frac{q_i^*}{q_j^*} = \log \frac{p_i}{p_j} + \frac{Q_i - Q_j}{\beta}. \tag{14}$$

Proof. Taking the ratio of two probabilities in Eq. 13 cancels the common normalizer:

$$\frac{q_i^*}{q_j^*} = \frac{p_i}{p_j} \exp\left(\frac{Q_i - Q_j}{\beta}\right).$$

Taking logarithms gives Eq. 14. □

Equation 14 makes the role of the incumbent explicit: executable evidence changes existing odds rather than replacing them. In particular, if $Q_i = Q_j$, their relative probability is unchanged.

A.3 Limiting Cases and Support

Proposition 4 (Boundary behavior)**.** *Let q_β^* be defined by Eq. 13. Then:*

1. *as $\beta \to \infty$, $q_\beta^* \to p$;*
2. *if $Q_i = c$ for all $i \in \mathcal{I}_p$, then $q_\beta^* = p$ for every $\beta > 0$;*
3. *if i^* is the unique utility maximizer and $p_{i^*} > 0$, then as $\beta \to 0^+$, $q_\beta^* \to \delta_{i^*}$.*

Proof. For the first claim, $\exp(Q_i/\beta) \to 1$ for finite Q_i, so Eq. 13 reduces to p_i. For the second, the common factor $\exp(c/\beta)$ cancels exactly. For the third, divide numerator and denominator by $\exp(Q_{i^*}/\beta)$. Every non-maximizing term then contains

$$\exp\left(\frac{Q_i - Q_{i^*}}{\beta}\right) \to 0,$$

while $p_{i^*} > 0$, so the normalized mass converges to the unique maximizer. □

The KL direction also fixes the support of the revision. If $p_i = 0$ and $q_i > 0$, then $D_{\mathrm{KL}}(q\|p) = \infty$. Consequently,

$$p_i = 0 \quad \Longrightarrow \quad q_i^* = 0. \tag{15}$$

Thus DiRe redistributes probability within incumbent support; it does not create support on an action assigned zero incumbent probability.

B DiRe Implementation and Reproducibility

This section specifies the implementation contract used for the corrected DiRe experiments. The central invariant is that executable evidence is collected and frozen before post-training. Methods may differ in how an evaluated decision is converted into supervision or realized by a learned policy, but not in the states, candidate inventories, incumbent distributions, executable utilities, or continuation protocol from which that evidence is constructed.

B.1 Frozen Evaluated Decisions

For every state s, we construct an ordered runtime candidate inventory

$$\mathcal{A}(s) = (a_1, \ldots, a_K)$$

and record the corresponding valid semantic support $\mathcal{A}_{\mathrm{v}}(s)$. Candidate identifiers are unique within a state, and their identities and ordering are fixed before executable evaluation.

The frozen pre-update policy is scored over the valid inventory to obtain $p_{\text{old}}(\cdot \mid s)$. Each valid candidate is then forced at the intervention state and followed by the same frozen continuation policy π_{cont}, yielding

$$Q(s, a_i) = \mathbb{E}\left[R(\tau) \mid s, a_i, \pi_{\text{cont}}\right]. \tag{16}$$

The expectation reduces to the observed return when the environment and continuation are deterministic. When genuine evaluation stochasticity is present, repeated executions are aggregated under the fixed evaluation protocol.

The resulting record

$$\mathcal{D}(s) = (\mathcal{A}(s), \mathcal{A}_{\text{v}}(s), p_{\text{old}}(\cdot \mid s), Q(s, \cdot))$$

is cached before target construction or policy training. Executable utilities are not recomputed as the trainable policy changes, and the trainable policy is never substituted for π_{cont} when constructing Q.

B.2 Candidate and Support Integrity

Candidate identity is treated as an explicit invariant. The same ordered inventory must be used for incumbent scoring, executable evaluation, target construction, training, and post-training comparison. We reject records with duplicate or missing candidate identifiers, inconsistent candidate ordering without explicit identity-preserving realignment, or non-finite incumbent scores or executable utilities.

DiRe targets are constructed only over the valid incumbent support. In log space, we compute

$$\ell_i = \log p_{\text{old}}(a_i \mid s) + \frac{Q(s, a_i)}{\beta}, \qquad q^*_\beta(\cdot \mid s) = \text{softmax}(\boldsymbol{\ell}), \tag{17}$$

with log-sum-exp normalization recommended for numerical stability. The frozen experiments computed masked $p_{\text{old}} \exp(Q/\beta)$ in finite precision on the unchanged finite utilities; the read-only audit verified finite normalized targets. This historical implementation is not claimed invariant to arbitrarily large additive utility shifts. Zero-support candidates are masked rather than assigned an artificial probability floor, consistent with Eq. 15.

B.3 Matched Supervision and Policy Realization

All conditions share cached evaluated decisions, state split, candidate inventories, and five paired initializations. The original target-interface comparison uses Adam at learning rate 0.03 for 220 epochs. The newer realization controls share a DEV search grid, but their selected budgets may differ. Only the reference-identity control fixes the loss and selected budget across all three arms (Appendix C.9).

The target-level controls are

$$q_{\text{hard}}(a \mid s) = \mathbf{1}\left[a = \arg\max_b Q(s, b)\right], \tag{18}$$

$$q_{\text{utility}}(a \mid s) \propto Q(s, a) \quad \text{on valid actions}, \tag{19}$$

$$q_{\text{DiRe}}(a \mid s) \propto p_{\text{old}}(a \mid s) \exp(Q(s, a)/\beta). \tag{20}$$

Utility-only uses a uniform distribution over valid actions when their utility sum is zero. Selected-action SFT uses the original selected action when valid; otherwise it uses the valid incumbent maximizer, without consulting Q.

For direct realization, the fixed DiRe target is fitted by

$$\mathcal{L}_{\text{direct}}(\theta) = -\frac{1}{|\mathcal{S}_{\text{tr}}|} \sum_{s \in \mathcal{S}_{\text{tr}}} \sum_{a \in \mathcal{A}_{\text{v}}(s)} q^*_\beta(a \mid s) \log \pi_\theta(a \mid s). \tag{21}$$

Residual-DiRe instead represents the target relative to the incumbent:

$$r^*(a \mid s) = \log(q^*_\beta(a \mid s) + \epsilon) - \log(p_{\text{old}}(a \mid s) + \epsilon), \tag{22}$$

and reconstructs the learned policy as

$$\pi_\theta(a \mid s) \propto p_{\text{old}}(a \mid s) \exp(\Delta_\theta(s, a)). \tag{23}$$

The same q^*_β, candidate support, fitting split, and DEV search grid are used when comparing the selected direct and residual realizations; their selected optimization budgets differ.

Training is offline over the frozen evaluated decisions. The controlled experiments do not introduce PPO, GRPO, online environment exploration, or recomputation of executable evidence during optimization.

B.4 Hyperparameter Selection and Test Isolation

Development and test states are separated before hyperparameter selection. Target temperature and optimization settings are selected using development data only and frozen before test evaluation. Test outcomes are not used to choose candidate inventories, sampling strata, target temperature, or optimization hyperparameters.

The archived development temperature grid is

$$\beta \in \{0.03, 0.05, 0.10, 0.20, 0.35, 0.50, 0.75, 1.00\}.$$

The archived DEV search tied on its score across this grid and retained $\beta = 0.03$. Corrected V2 inherited this frozen temperature; this is not a new temperature-selection experiment under corrected support. The target-level sensitivity analysis in Appendix C.2 recomputes only the non-parametric target from the same frozen evaluated decisions and is not additional test-set model selection.

For learned-policy comparisons, the optimizer configuration, learning rate, number of updates, random seed, data split, candidate inventory, and evaluator revision are recorded with each run. The matched optimization sweep in Appendix C.5 evaluates direct and residual parameterizations under the same predefined development grid.

The Direct+KL control is

$$\mathcal{L}_{\text{direct}} + \lambda D_{\text{KL}}\left(\pi_\theta(\cdot \mid s) \| p_{\text{old}}(\cdot \mid s)\right). \tag{24}$$

The corrected reference-control run selects λ exclusively from 13 DEV fitting states, minimizing mean $JS(q^*, \pi)$ over seeds 0–4. The grid is $\lambda \in \{0, 10^{-4}, 3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}, 0.01, 0.03, 0.1, 0.3, 1\}$, crossed with learning rates $\{0.0075, 0.015, 0.03\}$ and $\{110, 220, 440\}$ epochs. Exact ties prefer fewer epochs, smaller learning rate, then smaller λ. Direct+KL selects $\lambda = 0.1$, learning rate 0.03, and 440 epochs. Direct ($\lambda = 0$) and Residual-CE also select 0.03/440; Residual-DiRe selects 0.0075/110. All use full-batch Adam and the same seeded linear candidate scorer. The selections are frozen before evaluating these new TEST endpoints. A fixed reproduction check reads historical TEST predictions before that freeze; it is not used for selection. There is no independent TRAIN/DEV split. The original frozen 220-epoch endpoint remains explicitly labeled as fixed direct and is not interchangeable with the newer selected Direct.

B.5 Evaluation Metrics

We evaluate target specification and learned-policy realization separately. For a target distribution q, incumbent-relative displacement and expected executable improvement are

$$d_{\text{target}} = \frac{1}{|\mathcal{S}|} \sum_s D_{\text{JS}}\left(p_{\text{old}}(\cdot \mid s) \| q(\cdot \mid s)\right), \tag{25}$$

$$\Delta Q_{\text{target}} = \frac{1}{|\mathcal{S}|} \sum_s \left[\mathbb{E}_{a \sim q} Q(s, a) - \mathbb{E}_{a \sim p_{\text{old}}} Q(s, a)\right]. \tag{26}$$

For a learned policy π_θ, we report

$$d_{\text{policy}} = \frac{1}{|\mathcal{S}|} \sum_s D_{\text{JS}} \left(p_{\text{old}}(\cdot \mid s) \| \pi_\theta(\cdot \mid s) \right), \tag{27}$$

$$d_{\text{fit}} = \frac{1}{|\mathcal{S}|} \sum_s D_{\text{JS}} \left(q^*_\beta(\cdot \mid s) \| \pi_\theta(\cdot \mid s) \right), \tag{28}$$

$$\Delta Q_{\text{policy}} = \frac{1}{|\mathcal{S}|} \sum_s \left[\mathbb{E}_{a \sim \pi_\theta} Q(s,a) - \mathbb{E}_{a \sim p_{\text{old}}} Q(s,a) \right]. \tag{29}$$

Selected-action executable Q is evaluated separately as

$$Q \left(s, \arg\max_a \pi_\theta(a \mid s) \right). \tag{30}$$

We additionally report repairs, regressions, and target–policy top-action agreement where relevant. In the primary protocol, a repair is an increase in selected-action Q relative to the incumbent by more than 10^{-8}; a regression is a decrease by more than 10^{-8}. These are graded-utility transitions, not necessarily binary wrong-to-correct events. Net is repair minus regression. Counts are averaged over five seeds; rate differences divide these counts by 59. MiniWoB uses binary success and stratum-specific rates relative to historical base outcomes, as specified in Appendix C.8.

Distributional metrics and selected-action outcomes are intentionally kept separate: two learned policies may select the same action while representing different revisions of the underlying decision distribution. Principal matched comparisons use paired bootstrap confidence intervals over the 59 matched state units with 2,000 percentile bootstrap draws, retaining all five seeds for each sampled state. These intervals do not account for dependence among questions from the same conversation.

B.6 Independent Numerical Verification

We independently verify the analytical target against direct numerical optimization of Eq. 4. For each verification instance, we solve

$$\max_{q \in \Delta^{K-1}} \left[q^\top Q - \beta D_{\text{KL}}(q \| p) \right] \tag{31}$$

in float64 using an independent simplex-constrained numerical implementation and compare the resulting optimum q_{num} with the analytical q^*_β.

A log-probability floor of 10^{-300} is used only for finite numerical evaluation of logarithms; it does not alter the analytical target or introduce mass outside incumbent support. The predefined acceptance criterion is

$$\left\| q_{\text{num}} - q^*_\beta \right\|_\infty < 10^{-8}.$$

The maximum observed elementwise discrepancy is

$$2.99 \times 10^{-9},$$

which satisfies this criterion. This is an independent implementation check at $\beta = 0.1$, predating the final V4 diagnostic; it does not verify the V4 learned-policy outcomes.

We separately verify the residual realization operator. A zero residual reconstructs p_{old} exactly, while the analytical residual in Eq. 22 reconstructs q^*_β with maximum probability error 1.19×10^{-7} and zero mass outside the valid support. These checks validate the target and reconstruction implementations independently of learned-policy performance.

B.7 Reproducibility Invariants

A reproduction of the matched experiments should preserve the following invariants:

1. construct and freeze candidate identities, ordering, and valid support before target construction;
2. compute p_{old} from the frozen pre-update policy over exactly that candidate inventory;

3. evaluate candidate actions under a common frozen continuation protocol;
4. cache $(\mathcal{A}, \mathcal{A}_{\mathrm{v}}, p_{\mathrm{old}}, Q)$ before post-training;
5. construct all executable-feedback targets from the same cached records;
6. preserve the state split, initialization, matched seeds, and the stated fixed budget or DEV search grid for each comparison;
7. select target and optimization hyperparameters using development data only; and
8. evaluate learned policies without retrospectively modifying the frozen target construction.

These constraints separate two operations that are otherwise easy to conflate: collecting executable evidence and choosing how that evidence revises an existing decision.

C Additional Experimental Analyses

The analyses support the main evidence chain without changing its inference scope. For revision specification, Appendices C.2, C.8, and C.17 expand the utility–displacement distinction. For revision realization, Appendices C.9 and C.10 detail the identity and proximity controls; Appendices C.3–C.7 audit fitting error, optimization, reconstruction, and paired effects. Appendices C.11–C.14 characterize coverage, richer conditioning, and retrospective heterogeneity. Appendix C.15 records exposure, and Appendix C.16 records excluded results. The interface validation below supports task relevance rather than the DiRe update.

C.1 Matched Validation of the Decision Interface

The main experiments assume that the semantic decision interface contains task-relevant information before executable post-training. We validate this assumption separately using 72 matched questions under an approximately 1,250-token context budget. Compressed conditions use the same question set and candidate construction, and uncertainty is estimated by paired bootstrap over conversation clusters. Full History is included as an uncompressed reference.

Table 9: **Decision-interface validation on 72 questions.** Compressed methods use a 1,250-token budget; CIs are paired conversation-cluster bootstrap intervals versus Lexical. Full History is uncompressed.

Method	**Score ↑**	**Tokens ↓**	**Δ vs. Lexical [95% CI]**
Full History	0.2317	25,624.7	–
Recent	0.2569	1,221.7	-0.0479 $[-0.0592, -0.0326]$
Lexical	0.3048	1,224.0	–
Category only	**0.3881**	1,222.9	$+0.0833$ $[+0.0417, +0.1250]$
Category + residual	0.3858	**1,140.5**	$+0.0810$ $[+0.0417, +0.1250]$

Category-only routing improves over matched lexical retrieval by $+0.0833$ (95% CI $[+0.0417, +0.1250]$) at essentially the same context budget. Recent-context selection instead underperforms lexical retrieval by 0.0479. Adding residual reranking does not improve aggregate score (0.3858 versus 0.3881), although it uses fewer tokens. This experiment therefore supports the task relevance of the category-level decision representation; it does not evaluate the DiRe update rule or use the executable post-training intervention studied in the main experiments.

C.2 Target Temperature Sensitivity

The DiRe temperature β controls the strength of executable evidence relative to the incumbent distribution at target construction. We vary β while holding the evaluated decisions (p_{old}, Q) fixed. This intervention acts directly on the non-parametric target and does not require policy retraining.

Table 10: **Target temperature sensitivity.** Same 59 TEST states and valid support.

β	$\mathbf{JS}(\mathbf{p}_{\text{old}}, \mathbf{q}^*_\beta)$	$\Delta\mathbf{Q}_{\text{target}}$
0.03	0.0688	+0.0638
0.10	0.0272	+0.0421
0.35	0.0059	+0.0185
1.00	0.0007	+0.0059

Increasing β continuously weakens the revision. From $\beta = 0.03$ to 1.00, incumbent-relative displacement decreases from 0.0688 to 0.0007 JS, while expected executable improvement decreases from $+0.0638$ to $+0.0059$. This behavior agrees with the limiting case derived in Section 3: as executable evidence is downweighted, q^*_β approaches p_{old}.

This table is a target-only intervention. Legacy policy temperature runs used superseded invalid support and remain excluded. The later corrected reference-control run also contains $\beta = 0.10$ and 0.35 endpoints; they are not used to select the primary $\beta = 0.03$ or to claim uniformly positive realized change across temperatures.

C.3 Anatomy of the Target–Policy Realization Gap

The main results show that direct fitting can reverse the utility direction encoded by the target. We examine whether this aggregate result reflects a small number of atypical states or a broader realization failure. Define the state-level realization loss as

$$\Delta Q_{\text{loss}} = \Delta Q_{\text{policy}} - \Delta Q_{\text{target}}, \tag{32}$$

so that negative values indicate loss of intended executable improvement during parametric realization.

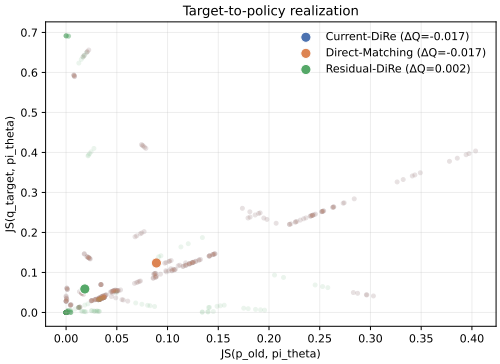


Figure 3: **State-level realization geometry.** 59 TEST states. Both direct controls coincide with the fixed 220-epoch endpoint, not Direct+KL (Table 5).

The realization loss has mean -0.0811 and median -0.0141, with a state-bootstrap 95% interval of $[-0.1344, -0.0377]$. More directly, 20 of 59 states (33.9%) exhibit a sign reversal: their DiRe target has positive expected executable improvement while the directly learned policy has negative improvement. These states have mean realization loss -0.1417, compared with -0.0500 over the remaining 39 states. The aggregate reversal is therefore not attributable solely to a small number of extreme states.

Table 11: **Direct-realization gap.** 59 TEST states, averaged over seeds; state-bootstrap 95% CIs.

Metric	**Mean**	**Median**	**IQR**	**95% CI**
ΔQ_{target}	+0.0638	+0.0004	0.0071	$[+0.0200, +0.1164]$
ΔQ_{policy}	-0.0173	0.0000	0.0314	$[-0.0431, +0.0049]$
ΔQ_{loss}	-0.0811	-0.0141	0.0559	$[-0.1344, -0.0377]$
$JS(q^*, \pi)$	0.1237	0.0696	0.1091	$[0.0921, 0.1591]$
$KL(q^* \Vert \pi)$	0.4679	0.2223	0.3834	$[0.3124, 0.6575]$

C.4 State-Level Correlates of Realization Loss

We next examine which properties of an evaluated decision are associated with loss of intended executable improvement under direct realization. This analysis is descriptive and is not used for model or hyperparameter selection.

Table 12: **Correlates of realization loss.** 59 TEST states; more negative ΔQ_{loss} means greater loss.

Predictor	Pearson	Spearman
Q margin	-0.758	-0.805
$JS(p_{\text{old}}, q^*)$	-0.689	-0.777
$JS(q^*, \pi)$	-0.728	-0.012
$KL(q^* \| \pi)$	-0.797	+0.001
Target entropy	+0.140	+0.381
Valid-action count	-0.128	+0.043

The most consistent associations are with executable Q margin and target revision magnitude. States with larger utility separation and larger incumbent-relative target revisions tend to lose more of their intended improvement during direct fitting. By contrast, target–policy JS and forward KL have strong Pearson but near-zero Spearman correlations with realization loss, indicating that their relationships are not well described by a monotone statewise ordering. We therefore treat these quantities as diagnostics rather than evidence of a causal mechanism.

C.5 Optimization-Budget Control for Policy Realization

A possible explanation for the realization gap is that direct and residual parameterizations require different optimization schedules. We therefore perform a matched development sweep over learning rate and training length for both realizations. The grid contains learning rates $\{0.0075, 0.015, 0.03\}$ and training lengths $\{110, 220, 440\}$ epochs, with five matched seeds for every configuration. Target construction and evaluated decisions remain fixed.

Table 13: **DEV-selected optimization.** Lowest mean target–policy JS in the matched 3×3 grid.

Realization	LR	Epochs	$\mathbf{JS}(\mathbf{q}^*, \pi)$
Direct matching	0.0300	440	0.0932
Residual-DiRe	0.0075	110	0.0645

Direct matching benefits from a larger optimization budget, but its best development target–policy JS within the tested grid remains above the best residual value. Residual realization reaches its best development fit at the smallest tested learning rate and training length. Thus, the realization difference is not explained solely by evaluating both parameterizations under the original optimization schedule. This finite grid does not establish that residual realization is uniformly easier to optimize outside the tested range.

C.6 Residual-Realization Sanity Checks

We verify the residual boundary properties independently of learned-policy performance. Zero residual should recover the incumbent; the analytical residual in Eq. 8 should reconstruct the target.

Table 14: **Residual reconstruction.** Maximum probability errors on corrected valid support.

Check	Maximum error
Zero residual $\Rightarrow p_{\text{old}}$	0
Perfect residual $\Rightarrow q^*_\beta$	1.19×10^{-7}
Invalid-support probability mass	0
Status	PASS

The zero-residual reconstruction is exact, and the analytical residual reconstructs q^*_β to within 1.19×10^{-7} maximum probability error while assigning no mass outside the valid support. The observed target–policy realization gap therefore cannot be attributed to an algebraic inconsistency in the residual reconstruction itself.

C.7 PAIRED REALIZATION EFFECTS

For completeness, Table 15 reports the paired effects underlying the realization comparison. The historical 220-epoch direct fit of the DiRe target is the fixed direct endpoint, distinct from the 440-epoch DEV-selected Direct. This appendix comparison uses the historical fixed endpoint.

Table 15: **Residual-DiRe minus fixed direct.** 59 TEST states, five seeds; paired state-bootstrap 95% CIs.

Metric	Δ	95% CI
Selected-action Q	+0.0177	$[+0.0000, +0.0524]$
Repair rate	+0.0068	$[+0.0000, +0.0203]$
Regression rate	-0.0169	$[-0.0508, +0.0000]$
Net repair rate	+0.0237	$[+0.0000, +0.0644]$
$JS(p_{\text{old}}, \pi)$	-0.0706	$[-0.0961, -0.0451]$
$JS(q^*, \pi)$	-0.0650	$[-0.0973, -0.0304]$
ΔQ_{policy}	+0.0195	$[+0.0038, +0.0355]$

Under this state-bootstrap analysis, residual realization reduces both incumbent-relative and target-relative JS and increases realized expected executable improvement. Selected-action improvements are directionally consistent but have confidence intervals that reach zero. These paired results therefore support the distributional realization claim more directly than a claim of uniformly improved discrete task outcomes.

C.8 MINIWOB EXECUTABLE-ACTION DIAGNOSTIC

The MiniWoB experiment tests whether the distinction between selected-action behavior and decision-distribution revision persists when the candidates are executable environment actions. The frozen diagnostic contains 200 states, deliberately stratified into 100 preserve and 100 repair states using base-policy outcomes and metadata before applying any target method. States are matched by task, interaction depth, and action type. Candidate actions are replayed from the same environment snapshot under a common continuation protocol. The panel has 43 DEV and 157 nominal TEST states. DEV selects $\beta = 0.01$, SGD learning rate 0.1, and 800 epochs. The diagnostic then fits a separate logit vector to each of all 200 states for each of five seeds; it therefore does not measure held-out policy generalization. Utility-only here uses $\mathrm{softmax}(Q/\beta)$, unlike normalized-$Q$ in the primary protocol. We use it only for the stated diagnostic contrast.

Table 16 reports the panel outcomes; it is not a held-out generalization evaluation.

Table 16: **MiniWoB in-panel fitting diagnostic.** All 200 states have separately fitted logits, including nominal TEST states. Preserve/Reg. use 100 preserve states; Repair uses 100 repair states; other metrics use all 200. Utility denotes expected Q.

	Outcome and Expected Utility					Decision Revision	
Method	**Preserve**	**Reg.**	**Repair**	**Success**	**Utility**	**JS**	ℓ_1
Selected-SFT	0.58	0.42	0.01	0.295	0.2926	0.4104	1.5615
Hard-Q	0.79	0.21	0.29	0.540	0.5345	0.4346	1.6160
Utility-only	0.79	0.21	0.29	0.540	0.5343	0.2860	1.1074
DiRe	0.79	0.21	0.29	0.540	0.5343	0.2596	0.9217

Hard-Q, Utility-only, and DiRe have identical preserve, regression, repair, and success outcomes in this panel, and nearly identical expected executable utility. Their distributional revisions nevertheless differ substantially: JS decreases from 0.4346 for Hard-Q to 0.2860 for Utility-only and 0.2596 for DiRe, while ℓ_1 displacement decreases from 1.6160 to 1.1074 and 0.9217. Paired repair and regression differences between Utility-only or DiRe and Hard-Q are exactly zero.

This panel has two important limitations. First, historical base outcomes disagree with rerun incumbent-action utilities for 43 of the 200 retained states. These states are retained rather than filtered after observing the rerun, so individual preserve/regression transitions are not interpreted causally. Second, the saved panel has 97 zero-margin and 103 unit-margin states; these are state margin counts, not counts of candidate utility values. This binary-margin diagnostic does not support a calibrated analysis of revision strength across evidence margins. We use MiniWoB only as an

executable-action diagnostic showing that matching top-action outcomes need not imply matching decision distributions.

C.9 Fixed-Target Reference-Identity Control

We hold q^* at $\beta = 0.03$, the valid support, 13 DEV fitting states, 59 TEST states, linear scorer, and CE objective fixed. All arms use Adam, learning rate 0.03, 440 epochs, and identical random initialization for seeds 0–4. This budget is selected on DEV for the aligned-reference CE arm and shared by all three arms. Only the reference in $\pi \propto p_{\mathrm{ref}} \exp f_\theta$ changes: the aligned incumbent, a uniform distribution on valid actions, or $p_{\mathrm{ref}} \propto p_{\mathrm{old}} \exp(-10Q)$. The target is not recomputed. Uniform-reference predictions exactly reproduce the DEV-selected Direct endpoint at this operating point.

The results are reported once, in Table 3 (Section 4.3).

C.10 Exact-Displacement Calibration and Paired Analysis

The exact-displacement control reuses the frozen 13-state fitting set, 59-state evaluation panel, valid support, q^*, and five-seed linear scorers. The Direct-CE candidates are an unregularized scorer and a scorer trained with a JS proximity penalty of 10, each at learning rate 0.03 for 880 steps. After statewise calibration on DEV, unregularized Direct-CE has lower target–policy JS (0.01371 versus 0.02232) and is selected before this calibration is applied to the 59 evaluation states. The latter states had already appeared in earlier experiments, so the analysis is retrospective rather than a new independent holdout.

For each state and seed, we set the comparison radius to Residual-CE's $JS(p_{\mathrm{old}}, \pi)$ and adjust Direct-CE along its logit ray from p_{old} to its uncalibrated output. If the direct ray cannot reach the radius, both policies are projected to the largest common reachable radius. No state requires this cap for the DEV-selected unregularized Direct-CE scorer: Residual-CE's output is unchanged on all 59 evaluation states. The direct control receives the residual policy's statewise radius, but calibration uses neither evaluation Q nor q^*.

Before projection, mean Direct-CE displacement is 0.10159; afterward both policies have mean displacement 0.03358. Mean absolute matching error over state–seed pairs is 2.24×10^{-16}, the median seed-averaged per-state error is 3.16×10^{-17}, and the largest state–seed error is 1.98×10^{-14}. Paired effects appear in Table 4. We average seeds within each state, then bootstrap the 59 paired state differences with 10,000 resamples. The 59 states come from three previously used conversations; these intervals do not measure conversation-level transfer.

Both means favor Residual-CE, but both intervals include zero. Exact displacement matching therefore strengthens the proximity control without establishing statistically reliable superiority over this direct direction. The fixed-target, equal-objective reference intervention in Section 4.3 remains the primary Stage-II identification result.

C.11 Forward-KL and Severe Missed Revisions

Table 8 computes quantiles after averaging five seeds within each of the 59 TEST states; no tail states are filtered. The illustrative seed-level miss in the main text is state `conv-50:152`, candidate index 5, seed 3. It is distinct from the state-mean quantiles in the table.

C.12 Expanded Semantic-Realization Study

Split construction. The expanded realization study is separate from the legacy 59-state evaluation. Legacy test conversations are excluded before state selection. The expanded split contains 96 training states from `conv-26`, `conv-41`, `conv-42`, and `conv-47`; 48 development states from `conv-43` and `conv-48`; and a previously unused 24-state evaluation batch from `conv-44`. Training/development conversations are disjoint from the evaluation conversation, but other scorers had already been tested on earlier state-disjoint batches from `conv-44`. Within eligible groups, questions are selected using a deterministic SHA256 ranking fixed before executable Q values are evaluated. Candidate generation, candidate ordering, valid-action support, incumbent distribu-

Table 17: **Development diagnostics for shared residual realization.** Five-seed development means. Lower JS and KL are preferred; higher executable ΔQ and top-action agreement are preferred.

Representation / objective	JS ↓	ΔQ ↑	KL ↓	Agree ↑	Severe
Base MLP-64, CE	.06784	+.01371	.600	.883	0
Candidate-ID MLP-128, CE	.05715	+.01818	.629	.917	0
Candidate-ID MLP-128, $\alpha = 1.25$	.05599	+.02228	.666	.925	0
Candidate-ID MLP-64, $\alpha = 1.5$	.05561	+.02843	.844	.913	0
Utility-pairwise MLP-128	**.04526**	+.02137	1.769	.938	0
TF–IDF/SVD MLP-128	.04670	+.02104	1.097	.938	0
Qwen-semantic MLP-128, CE	.04953	**+.03658**	**.346**	.900	0

tions, executable evaluation, and target temperature $\beta = 0.03$ are frozen before realization training. Architecture, representation, objective, optimizer, and checkpoint selection for this run use training/development data. The later 24-state batch is evaluated once for the final semantic scorer; the archive does not establish that earlier `conv-44` test outcomes were independent of subsequent method development.

Oracle decomposition. The exact residual required to reconstruct the DiRe target is

$$r^*(s, a) = \log(q^*(a \mid s) + \epsilon) - \log(p_{\text{old}}(a \mid s) + \epsilon).$$

Substitution into the realization operator reconstructs q^* with maximum probability error approximately 1.19×10^{-7} and assigns zero probability outside valid support. We additionally optimize independent residual logits for each development state under the same target-fitting objective. This free-logit oracle reaches $\text{JS}(q^*, \pi) \approx 3.7 \times 10^{-8}$ on the historical development objective. These checks exclude an algebraic reconstruction barrier and demonstrate fit with unrestricted per-state logits. They do not isolate the cause of shared-scorer error or establish held-out generalization.

Semantic features and scorer. The final model augments the original candidate/state features and candidate identity with frozen semantic question–answer representations extracted from Qwen3-4B. The semantic encoder is not trained by the DiRe objective. During development, PCA and feature standardization are fitted on training data only; the semantic representation is reduced to 32 dimensions. For the final fit, these transforms are refitted using the 144 training and development states, without using test features or labels. The shared scorer is a two-layer MLP with hidden dimension 128 and zero-initialized output layer. Thus, before training, $\Delta_\theta(s, a) = 0$ and $\pi_\theta(a \mid s) = p_{\text{old}}(a \mid s)$. The final scorer is trained by cross-entropy to the fixed DiRe target, with learning rate 0.01 and gradient clipping at 5. The final frozen policy is evaluated on this later 24-state batch across five optimization seeds.

Development representation study. Table 17 summarizes principal development candidates. Values are five-seed means. They are used for model selection and diagnosis, not as independent test evidence. Selection considers executable improvement, forward KL, and severe misses alongside target–policy JS. The severe-miss indicator is one when $\max_a q^*(a \mid s) / \max_{a:q^*(a|s)>10^{-12}} \pi(a \mid s) > 20$.

The Qwen-semantic configuration is selected because it provides the strongest balanced development executable improvement with comparatively low forward KL and no severe misses. The lowest development JS belongs to utility-pairwise scoring, but its forward KL is substantially higher. This illustrates why target–policy JS alone is not an adequate selection metric.

Final state-batch result and variability. On the later state-disjoint `conv-44` batch, the target improvement is $+0.08259$ and the mean realized improvement is $+0.04202$ (50.9%). Mean $\text{JS}(q^*, \pi) = 0.06655$, mean forward KL is 0.48407, and top-action agreement is 87.5%. The mean severe-miss rate is zero under the criterion above. Across five optimization seeds, policy ΔQ has standard deviation 0.00981, target–policy JS has standard deviation 0.01852, and selected-action Q has standard deviation 0.00676. These seeds measure optimization variability on the same state batch, not independent conversation-level replications.

Scope. This study evaluates Stage-II parametric realization with richer conditioning. It does not replace the matched reference-identity control in Section 4.3, where target, model, objective, and optimization configuration are held fixed to isolate the realization reference. The expanded study instead asks whether a more informative shared scorer can realize an already specified DiRe target on a later state batch from a previously tested conversation. Its cross-study contrast is not a matched feature ablation.

C.13 Retrospective Cross-Conversation Stress Test

We evaluated the frozen Qwen-semantic residual architecture in six grouped folds over `conv-26`, `conv-41`, `conv-42`, `conv-43`, `conv-47`, and `conv-48`. Each fold tests one 24-state conversation, selects on two other 24-state conversations, and fits on the remaining three; the final fit uses all five non-test groups. `conv-44` is excluded from every fold. The six evaluated conversations had been used in earlier development, and the residual scorer improved target–policy JS over the incumbent on only two of six DEV folds. This is a retrospective stress test rather than independent replication. The five optimization seeds are nested within conversations.

Table 18: **Retrospective cross-conversation realization.** Each conversation has 24 states; values are five-seed means. Ratio is realized divided by target ΔQ; the last row uses the ratio of equal-conversation means. The incumbent column is $JS(q^*, p_{\text{old}})$.

Conversation	Target ΔQ	Realized ΔQ	Ratio	$JS(q^*, \pi)$	Incumbent JS	$KL(q^* \| \pi)$
`conv-26`	+0.04801	+0.01774	36.9%	0.07350	0.08294	0.47687
`conv-41`	+0.07047	+0.00163	2.3%	0.07938	0.05908	0.87033
`conv-42`	+0.03175	+0.01244	39.2%	0.02975	0.01624	0.47840
`conv-43`	+0.04465	+0.00165	3.7%	0.03346	0.03650	0.85002
`conv-47`	+0.06624	+0.02124	32.1%	0.06236	0.03846	0.61123
`conv-48`	+0.12356	-0.00462	-3.7%	0.11918	0.12252	1.28393
Equal-group mean	+0.06411	+0.00835	13.0%	0.06627	0.05929	0.76180

The equal-group mean target improvement is $+0.06411$, versus realized $+0.00835$ (13.0%). Five conversations have positive mean realized change; `conv-48` is negative at -0.00462 despite target $+0.12356$. Mean $JS(q^*, \pi)$ is 0.06627, worse than 0.05929 for the unchanged incumbent. A 10,000-draw conversation bootstrap gives a 95% interval $[+0.00119, +0.01592]$ for mean realized change, but the earlier development exposure prevents confirmatory generalization inference.

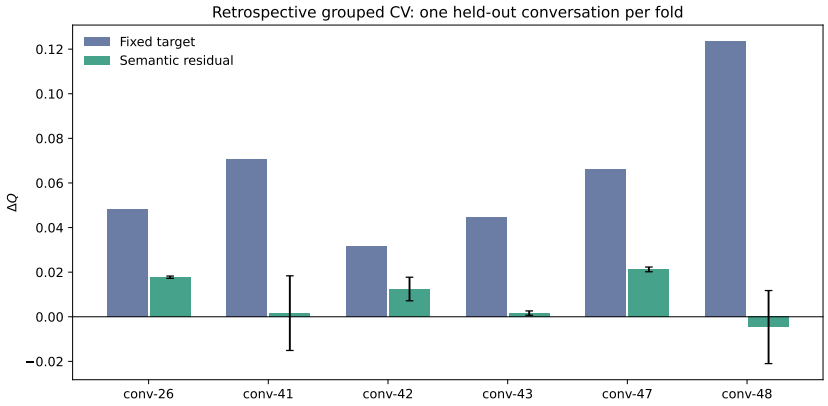


Figure 4: **Retrospective utility realization by conversation.** Fixed target and semantic residual across six groups. Error bars show standard deviation over five optimization seeds, not uncertainty across conversations. All groups were development-exposed; this diagnoses heterogeneity, not independent replication.

C.14 Post-Hoc Revision-Magnitude Diagnostic

On positive valid support, Eq. 8 without the numerical ϵ has the gauge-centered form $\overline{r^*} = \overline{Q/\beta}$, whereas the direct target logits are $\overline{\log p_{\text{old}} + Q/\beta}$. Thus the two fits represent different objects, although the identity alone does not establish which is easier to learn. In the historical 59-state panel, the actionwise RMS of the centered residual is smaller than that of the centered direct logits in 45 states; their respective medians are 0.855 and 4.284. These gauge-dependent scales are descriptive, not a function-class complexity measure.

Table 19: **Post-hoc realization by target revision magnitude.** States are grouped by $JS(p_{\text{old}}, q^*)$; the 29 zero-revision states are kept together and the 30 positive-revision states are split by rank. Values are state means after averaging five matched seeds. This is a descriptive analysis of the exposed historical panel, not an independent test.

Revision	States	$JS(p_{\text{old}}, q^*)$	$JS(q^*, \pi)$		ΔQ_{policy}	
			Direct	Residual	Direct	Residual
Zero	29	0.00000	0.13377	0.01187	0.00000	0.00000
Lower nonzero	15	0.00135	0.06851	0.00373	-0.06889	-0.00338
Higher nonzero	15	0.26913	0.18930	0.19836	+0.00528	+0.02414

Five zero-revision states have one valid action; on the other 24, Residual-CE has lower target JS in every paired state. For higher revisions, its target JS and forward KL are worse than Direct-CE despite higher realized utility. The strata therefore do not support a monotone advantage as revision size grows. The fixed-target reference intervention in Section 4.3 remains the identification result.

C.15 Conversation Provenance

We audited all ten available LoCoMo conversations before considering an untouched-group test. Every group had contributed to earlier DiRe fitting, development, evaluation, or diagnostics. Sampling new states from any of these groups would not create an untouched conversation-level test.

Table 20: **Provenance and inference scope of evaluation groups.** All LoCoMo groups listed here were previously exposed during DiRe development or analysis. MiniWoB is a separate executable environment.

Group	Earlier role	Exposure	Supported use
Historical three	DEV/TEST	Yes	Controlled state-level intervention
Expanded six	TRAIN/DEV	Yes	Retrospective heterogeneity
`conv-44`	Earlier TEST batches	Yes	Later state-level holdout
MiniWoB panel	Separate environment	In-panel	Executable-action diagnostic

C.16 Excluded or Non-Identified Controls

The legacy policy temperature analyses are excluded from empirical comparisons: learned-policy temperature runs for $\beta \in \{0.10, 0.35, 1.00\}$ were produced under an earlier invalid-support target protocol. Their target-level quantities remain well defined under the corrected protocol and are reported in Table 10, but the corresponding learned-policy statistics are not used.

The historical zero-selected KL variant is not evidence for the newer Direct+KL result. The latter is supported by its separate corrected DEV search and frozen selection in Appendix B.4.

C.17 TARGET-LEVEL REVISION GEOMETRY

Table 21: **Targets and learned policies on corrected V2.** Means over 59 TEST states and five seeds. Upper block: fixed 220-epoch direct fits; lower block: DEV-selected realizations of the same DiRe target. Repair/Reg. are counts; Sel. Q is selected-action utility.

	Target Revision			Learned Policy					
Method	$\Delta\mathbf{Q}_{\text{target}}$	$\mathbf{JS}(\mathbf{p}_{\text{old}}, \mathbf{q})$	**Top Chg.**	**Sel. Q**	$\Delta\mathbf{Q}_{\text{policy}}$	$\mathbf{JS}(\mathbf{p}_{\text{old}}, \pi)$	$\mathbf{JS}(\mathbf{q}, \pi)$	**Repair**	**Reg.**
Selected-SFT	+0.0376	0.1291	–	0.3474	-0.0143	0.2094	0.2026	4.0	4.0
Hard-Q	+0.0645	0.1101	20.3%	0.3543	-0.0069	0.0987	0.0966	4.0	1.0
Utility-only	+0.0625	0.1554	20.3%	0.3492	-0.0046	0.1280	0.0564	4.0	2.0
DiRe	+0.0638	0.0688	15.3%	0.3535	-0.0173	0.0890	0.1237	3.6	1.0
DEV-selected realization controls for the fixed DiRe target									
Direct	+0.0638	0.0688	15.3%	0.3684	-0.0162	0.1006	0.1313	3.2	1.0
Direct+KL	+0.0638	0.0688	15.3%	0.3583	-0.0153	0.0847	0.1211	2.0	1.0
Residual-DiRe	+0.0638	0.0688	15.3%	0.3712	+0.0022	0.0185	0.0587	4.0	0.0

Top Chg.: fraction of states whose target maximizer differs from the incumbent action.

This characterization measures the utility–displacement tradeoff of different supervision interfaces before parametric fitting. As shown in Table 21, Hard-Q, Utility-only, and DiRe encode nearly identical expected executable improvements: $+0.0645$, $+0.0625$, and $+0.0638$, respectively. Their revisions of the incumbent, however, differ substantially. DiRe yields a mean target JS of 0.0688, compared with 0.1101 for Hard-Q and 0.1554 for Utility-only, while changing the target maximizer on 15.3% rather than 20.3% of states.

Because these targets are constructed from the same states and executable utilities, the displacement differences admit paired comparisons. Relative to Hard-Q, DiRe reduces incumbent-relative target JS by 0.0414 (95% CI $[-0.0779, -0.0113]$); relative to Utility-only, the reduction is 0.0866 (95% CI $[-0.1122, -0.0597]$). The gains are numerically close, but this is not an equivalence claim: relative to Hard-Q, DiRe has a small target-utility deficit of 0.0007 (95% CI $[-0.0013, -0.0002]$) alongside the larger JS reduction. Thus the result is a measured utility–displacement tradeoff. The interfaces therefore encode similar executable improvement through substantially different revisions of the incumbent decision.

The temperature trajectory is reported in Appendix C.2.