Learning Adaptive Tutor Policies for Conversational Language Learning via Offline Reinforcement Learning

Aditya Bora
Department of Computer Science, Stanford University
adibora@stanford.edu CS224R Final Project

Extended Abstract

Problem. Building genuinely engaging conversational tutors requires adapting to the learner as a dialogue unfolds, yet most current systems do not. They rely on a fixed script or a fixed set of heuristics, producing largely the same responses regardless of how the student replies. This is a missed opportunity, since effective teaching is inherently sequential: the best move at any turn depends on the student's most recent response and on the direction of the conversation. We therefore aim to learn a tutoring strategy that selects the most engaging move at each turn. Because experimentation on real learners is impractical, we adopt a fully offline approach, learning from previously logged conversations and evaluating the resulting policies without deployment.

Approach. We formulate tutoring as a Markov Decision Process. At each turn, a high-level *director* policy observes the recent conversation, represented as a 384-dimensional MiniLM embedding of the last six turns, and selects one of eight teaching moves, such as asking a follow-up, clarifying, or providing an example. A language model then renders the selected move as fluent text, following the SayCan-style separation of a high-level policy and a low-level executor. We train the director on **503 real tutoring episodes (10,222 transitions)** from a deployed product, using a dense engagement reward derived from the student's subsequent reply, and we compare four policies: Behavior Cloning (BC), Implicit Q-Learning (IQL), Conservative Q-Learning (CQL), and two scripted baselines. As deployment is infeasible, we evaluate every policy with Fitted-Q Evaluation (FQE), an off-policy evaluation (OPE) method, against the logged human tutors, whose value is 26.55.

Findings. Behavior Cloning (24.30) and IQL (24.72) both recover roughly human-level performance, indicating that the setup is well calibrated. CQL appears substantially stronger, exceeding the human tutors by 58% (41.97); this result proved the most informative of the study, as the gain is not genuine. We show that it reflects OPE over-estimation, supported by three lines of evidence. First, CQL alters its behavior only marginally, shifting roughly 11% of its action mass, yet its estimated value rises by 60%. Second, a temperature ablation shows the inflated value increasing as the policy becomes more deterministic. Third, a deterministic baseline that remains in-distribution exhibits no inflation. A separate reward ablation further demonstrates that the choice of reward materially shapes the learned policy.

Takeaway. The central lesson is that, in offline tutoring RL, the principal difficulty is not learning a policy but evaluating it reliably. CQL's conservatism reshapes the policy without correcting the evaluation, so the trustworthy comparison is between BC and IQL, both of which align with human-level performance. More generally, off-policy evaluation can be confidently wrong precisely when a policy drifts toward actions that the data poorly supports, which is exactly when a reliable estimate is most needed.

Abstract

Engaging conversational tutors must adapt to the learner as a dialogue progresses, a capability that remains limited in current AI tutors, which rely largely on fixed prompting. In this work we aim to learn an adaptive tutoring strategy directly from logged conversations using offline reinforcement learning. We formulate tutoring as a Markov Decision Process over eight high-level teaching moves, learn a director policy from 503 real tutoring episodes (10,222 transitions), and evaluate it with Fitted-Q Evaluation. Behavior Cloning and Implicit Q-Learning recover human-level engagement value, whereas Conservative Q-Learning appears to exceed human tutors by 58%. Using a temperature ablation and an in-distribution control, we show that this apparent gain is off-policy-evaluation over-estimation driven by determinism toward out-of-distribution actions, rather than a genuine improvement. Our findings indicate that reliable off-policy evaluation, rather than policy learning, is the principal challenge in offline tutoring RL, and we outline a path toward closed-loop validation.

1 Introduction

A growing share of language learning takes place through conversational AI tutors, and recent large language models (LLMs) can sustain dialogue with considerable fluency. What these systems still lack is the ability to adapt to the individual learner. Most tutors, including LLM-based ones, operate from a *fixed script* or *static prompting*, producing largely the same responses regardless of how the student behaves. This is a meaningful limitation, as it overlooks what is arguably the defining feature of good teaching: that it is *adaptive* and *sequential*. The most effective move at any moment depends on the student's most recent response, and it shapes the remainder of the conversation.

This is precisely the setting that reinforcement learning (RL) is designed to address. The obstacle is a practical one: no faithful simulator of a human learner exists, and live experimentation on real students is slow, costly, and ethically fraught. We therefore adopt an offline approach, learning a policy from previously logged conversations and, critically, evaluating that policy without deployment through off-policy evaluation (OPE).

We cast tutoring as a high-level decision problem. At each turn, a *director* policy observes the conversation and selects one of eight teaching moves, which a language model then renders as natural language. We learn the director from real logged sessions and pose two questions. First, can offline RL match or surpass human tutoring strategy under honest evaluation? Second, how far can that evaluation be trusted once the learned policy diverges from the data on which it was trained? The second question ultimately shaped the direction of the project.

We make four contributions. First, we cast a deployed tutor's interaction logs as an offline-RL problem with an explicit state, action, and reward formulation. Second, we benchmark imitation (BC), an offline policy method (IQL), and an offline value method (CQL) against scripted baselines, evaluating all of them with FQE relative to a behavior reference. Third, we identify and characterize an OPE *over-estimation* effect: CQL's apparent +58% improvement is an artifact of evaluating a deterministic policy on out-of-distribution (OOD) actions, which we confirm through a temperature ablation and an in-distribution control. Finally, we articulate the resulting lesson, that reliable OPE rather than the training method is the limiting factor, and we outline a concrete plan for closed-loop validation.

2 Related Work

Reinforcement learning has a long history in dialogue, particularly in task-oriented settings where agents optimize task success or user satisfaction. Young et al. [4] review POMDP-based spoken-dialogue systems, which treat dialogue as sequential decision-making but depend on hand-crafted state representations suited to narrow domains. We retain this sequential-decision framing while targeting open-ended tutoring:

rather than engineering states by hand, we represent them with general-purpose sentence embeddings and define a pedagogical action space over them.

The work most closely related to our method is offline reinforcement learning, in which the objective is to learn from a fixed dataset without further interaction. Its central difficulty is a tendency to over-value actions that are absent from the data. Kumar et al. [2] address this with Conservative Q-Learning (CQL), which augments the Bellman objective with a penalty that lowers the value of out-of-distribution actions and raises the value of those observed in the data. Kostrikov et al. [1] propose Implicit Q-Learning (IQL), which avoids out-of-distribution queries by never maximizing over actions, instead employing expectile regression and advantage-weighted policy extraction. We adopt both methods, and we find that despite CQL's conservatism its *evaluated* value can still be inflated in our setting. A conservative training objective, it turns out, does not guarantee a conservative evaluation.

Finally, our work relates to the literature on intelligent tutoring systems. VanLehn [3] finds that such systems can approach the effectiveness of human tutors, though the systems examined are predominantly rule-based or supervised rather than sequential decision-makers. We replace these hand-built rules with a learned RL policy over high-level moves, and our director/executor decomposition follows the SayCan pattern, in which a high-level policy proposes actions that a language model executes. For evaluation we rely on Fitted-Q Evaluation, a standard OPE estimator; accordingly, our analysis is fundamentally a study of when that estimator can be trusted.

3 Approach

3.1 MDP formulation

We model a tutoring conversation as a Markov Decision Process $(\mathcal{S}, \mathcal{A}, r, P, \gamma)$, defined as follows:

- **State** s: a 384-dimensional MiniLM embedding of the last six dialogue turns, summarizing what the conversation is about.
- **Action** a: one of eight discrete pedagogical moves (`ASK_FOLLOW_UP`, `CLARIFY_CONFUSION`, `PROVIDE_EXAMPLE`, `CORRECT_ERROR`, `VALIDATE_RESPONSE`, `SWITCH_TOPIC`, `END_CONVERSATION`, `OTHER`).
- **Reward** r: a dense engagement signal computed from the *next* student turn (detailed below); a sparse variant is used for ablation.
- **Transition** P: the next tutor turn within the same episode, with *done* when there is no student response or it is the last tutor turn.
- **Discount** $\gamma = 0.99$.

The problem is formally partially observable. We approximate it as fully observed through the recent-history embedding, which is a standard and pragmatic choice for dialogue RL.

3.2 State representation

We encode states with `sentence-transformers/all-MiniLM-L6-v2` (384-d, L_2-normalized), computing the embeddings once on GPU for all 10,222 `state_history` strings and caching them. Freezing the encoder keeps the downstream policies small MLPs that train in seconds, which lets us afford many seeds and ablations cheaply. The trade-off, which we note as a limitation, is that the encoder is general-purpose rather than fine-tuned for tutoring dialogue.

3.3 Reward design

Our dense reward reflects the student's engagement on the following turn. If the student does not respond, we assign $r = -1.0$. Otherwise, the base reward is $+1.0$, plus $+0.5$ at each of ≥ 5, ≥ 15, and ≥ 30 words, and minus 0.25 if the reply is ≤ 2 words; a further -1.0 applies if the episode ended early and this is the last tutor turn. For the ablation we use a sparse variant: $+1$ for any response, -1 for silence, and -1 for an early end.

3.4 Learned policies

All of our learners are small multilayer perceptrons (two hidden layers of width 256, ReLU) over the frozen embeddings, implemented in pure PyTorch.

Behavior Cloning (BC). BC is a supervised classifier trained with cross-entropy to predict the human tutor's action from the state. It captures the average human policy, and because it never observes the reward, it also serves as a useful control in our reward ablation.

Implicit Q-Learning (IQL). This is an offline policy method that avoids querying out-of-distribution actions by never taking a max over actions. It learns a value function via expectile regression,

$$\mathcal{L}_V = \mathbb{E}\big[\, |\tau - \mathbf{1}\{u < 0\}|\, u^2 \big], \quad u = \bar{Q}(s,a) - V(s),$$

with expectile $\tau = 0.7$; a Q-function via a Bellman backup onto V, with target $r + \gamma(1 - \mathit{done})\, V(s')$; and a policy via advantage-weighted regression with weight $\min(\exp(\beta A), 100)$, advantage $A = \bar{Q}(s,a) - V(s)$, and $\beta = 3.0$.

Conservative Q-Learning (CQL). This is an offline value method that uses a double-DQN Bellman backup augmented with a conservatism penalty,

$$\mathcal{L} = \mathcal{L}_{\mathrm{TD}} + \alpha\, \mathbb{E}\big[\, \log \textstyle\sum_a e^{Q(s,a)} - Q(s, a_{\mathrm{data}}) \big],$$

with $\alpha = 1.0$; the greedy $\arg\max_a Q(s,a)$ is the policy.

Scripted baselines. `always_follow_up` always selects the dominant in-distribution action, while `random` selects uniformly over the eight actions. These instantiate the fixed heuristic policies against which we contextualize the learned policies.

3.5 Off-policy evaluation

Because we cannot deploy to real learners, we score each policy with Fitted-Q Evaluation (FQE), which fits an evaluation critic Q^π for the target policy by repeatedly regressing onto its own bootstrap,

$$\text{target} = r + \gamma(1 - \mathit{done}) \textstyle\sum_a \pi(a \mid s')\, Q^\pi(s', a),$$

and reports the expected value over initial states, $J(\pi) = \mathbb{E}_{s_0}[\sum_a \pi(a \mid s_0)\, Q^\pi(s_0, a)]$. As a calibration anchor, we compute the model-free Monte-Carlo return of the logged human tutors, which is 26.55 under the dense reward and 16.21 under the sparse reward. FQE values near this line indicate a well-calibrated estimate, whereas values far above it warrant scrutiny, as we discuss in Section 6.

4 Experiments

4.1 Data

Our logs originate from a deployed conversational language-tutoring product. A processing pipeline parses each lesson transcript defensively and segments it at "ended-early/restart" markers, so that 500

Table 1: Dataset statistics by action (dense reward).

Action	N	Resp. rate	Avg words	Diseng.
`ASK_FOLLOW_UP`	7469	97.6%	17.6	17.9%
`PROVIDE_EXAMPLE`	1445	98.3%	8.9	13.1%
`OTHER`	630	87.3%	11.1	12.7%
`VALIDATE_RESPONSE`	477	**70.0%**	14.0	10.3%
`SWITCH_TOPIC`	88	80.7%	11.6	15.9%
`CLARIFY_CONFUSION`	58	98.3%	18.1	20.7%
`END_CONVERSATION`	33	27.3%	1.6	9.1%
`CORRECT_ERROR`	22	100.0%	7.9	9.1%

Table 2: Off-policy value (FQE), dense reward, 5 seeds. Behavior (human) Monte-Carlo value = 26.55.

Policy	FQE value	vs. behavior
`random`	9.94 ± 1.12	-16.6
BC	24.30 ± 0.98	-2.25
`always_follow_up`	26.17 ± 0.69	-0.39
IQL	24.72 ± 1.36	-1.83
CQL	$\mathbf{41.97 \pm 3.62}$	$\mathbf{+15.42}$

logged rows yield 503 clean episodes. It then labels each tutor turn with one of eight actions via a keyword heuristic, computes the dense and sparse rewards from the next student turn, and assembles $(s, a, r, s', done)$ tuples. The result is **10,222 transitions across 503 episodes**; the per-action counts sum to 10,222. Table 1 reports descriptive statistics, and the corpus is heavily dominated by `ASK_FOLLOW_UP` (73%).

4.2 Experimental details

We train each configuration over **5 random seeds** and report means ± standard deviations. Embeddings are computed and cached once, and the policies are small MLPs that train in seconds on a single GPU. As a result, the full BC/IQL/CQL sweep over five seeds plus FQE completes in only a couple of minutes of GPU time, fanned out in parallel on serverless (Modal) T4 GPUs. The full set of hyperparameters appears in Table 4.

5 Results

5.1 Quantitative Evaluation

We begin with the main sweep (Table 2, Figure 1), which gives the FQE value of each policy under the dense reward, measured against the human behavior value of 26.55. Behavior Cloning (24.30) and IQL (24.72) fall essentially at human level, the `always_follow_up` baseline (26.17) lies close to it as well, and `random` drops to 9.94. CQL, by contrast, stands well apart at 41.97, roughly +58% above the human tutors, a result we examine in Section 6.

We next turn to a temperature ablation, shown in Table 3 and Figure 2. Here we evaluate CQL as softmax(Q/T) across a range of temperatures. As $T \to 0$ the policy becomes greedy and deterministic; as $T \to \infty$ it approaches a uniform policy. The FQE value rises monotonically with determinism, climbing from 22.34 near uniform (roughly equal to the behavior reference) up to 42.34 when fully greedy.

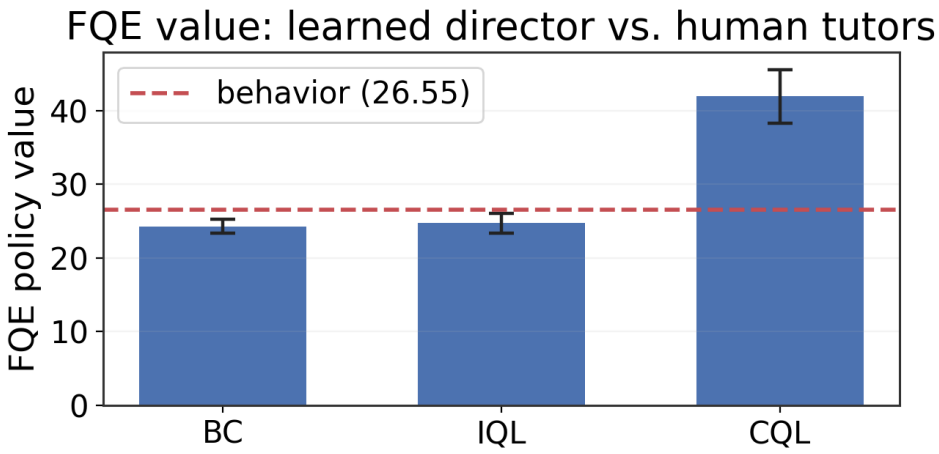

Figure 1: FQE value of each learned director vs. the human-behavior reference line. BC and IQL land at human level; CQL appears far higher, an artifact diagnosed in Section 6.

Table 3: CQL temperature ablation (FQE, 5 seeds).

T ($\to 0$ = greedy)	FQE value
0.1	42.34 ± 4.16
0.5	41.31 ± 3.99
1.0	39.98 ± 3.36
2.0	33.75 ± 1.83
5.0	22.34 ± 2.42

Finally, we run a reward-design ablation. Under the sparse reward, whose behavior value is 16.21, FQE gives BC 14.67 ± 0.63, IQL 14.54 ± 0.93, and CQL 19.90 ± 0.99. CQL is again inflated above the behavior reference, showing the same over-estimation signature we saw under the dense reward, whereas BC and IQL remain near the calibrated line.

5.2 Qualitative Analysis

We first examine what the policies actually do (Figure 3). BC and IQL closely mirror the human tutors, with roughly 73% follow-up questions, whereas CQL shifts only about 11% of its mass, lowering `ASK_FOLLOW_UP` from 73% to 62% and raising `PROVIDE_EXAMPLE` from 14% to 22%. We regard this near-identical behavior as the first qualitative indication that CQL's large value gain cannot be genuine: a policy that behaves almost exactly like the human tutors cannot plausibly be 58% better than them.

The per-action statistics in Table 1 are themselves informative about which moves engage students. Bare `VALIDATE_RESPONSE` ("good job") has the *worst* response rate of the substantive moves, at 70.0%. Asking follow-ups and clarifying confusion, by contrast, sustain very high response rates ($\geq 97.6\%$) and elicit the longest replies. This suggests that bare praise tends to end exchanges, whereas questions and clarifications sustain engagement, consistent with the strategies that our well-calibrated policies (BC and IQL) concentrate on.

Lastly, Figure 4 shows how the reward shapes behavior by contrasting the greedy action mix under the dense and sparse rewards. BC is invariant, as expected, since it never observes the reward and serves as a clean control. CQL is the most reactive: the dense reward draws it toward `PROVIDE_EXAMPLE` (21.6%), whereas the sparse, response-only reward returns it to the high-response-rate `ASK_FOLLOW_UP`

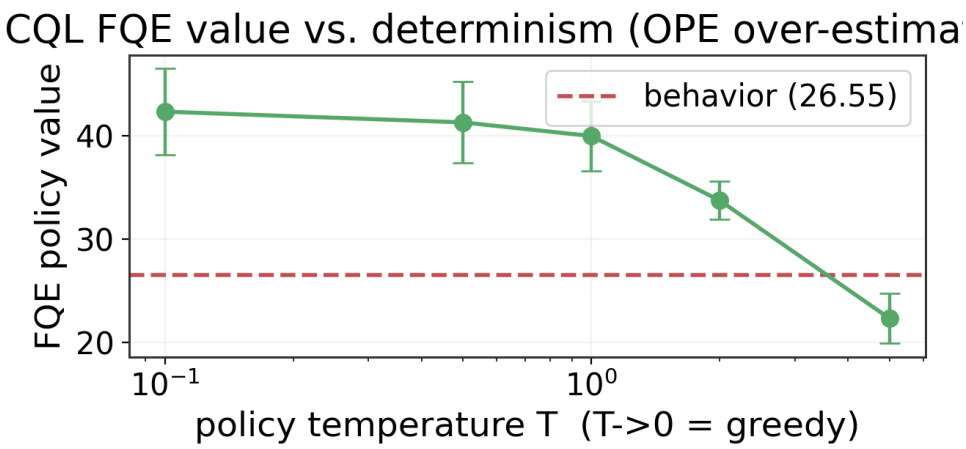

Figure 2: CQL FQE value rises monotonically as the policy becomes more deterministic, collapsing to ≈ behavior when near-uniform. The inflated value is an evaluation artifact of OOD deterministic actions, not a real engagement gain.

(62.1% → 70.0%). Reward design is therefore a first-order determinant of the learned strategy rather than an afterthought.

6 The Central Finding: OPE Over-Estimation

We contend that CQL's apparent +58% improvement is *not* a genuine engagement gain. Three independent lines of evidence point instead to off-policy-evaluation over-estimation.

(1) Value–behavior mismatch. CQL's action distribution barely moves, only about 11% of its mass (Figure 3). Its FQE value, however, rises 60%. Since a small change in behavior cannot produce a large change in *true* value, the gap must originate in the estimator itself. This is the signature of FQE extrapolating on actions for which it has very little data.

(2) Temperature monotonicity. The FQE value we report is, in essence, the expected critic value under the policy's action distribution. As we make CQL more deterministic, it concentrates probability on its single highest-Q move. Two effects then compound. The first is a maximization bias: the arg max over noisy Q estimates systematically selects over-estimated values. The second is OOD extrapolation, since the favored move is often under-supported, making its critic value both the highest and the least reliable. We therefore interpret the monotonic rise from 22.34 to 42.34 as $T \to 0$ (Figure 2) as precisely this artifact; a genuine engagement gain would not vary with a temperature parameter.

(3) An in-distribution control. We then rule out determinism alone as the cause. The fully deterministic `always_follow_up` baseline is *not* inflated (26.2 ≈ behavior), because its single action is heavily in-distribution and is therefore valued accurately. CQL inflates only because it commits to *less-supported* actions, so the over-estimation is driven by **determinism toward OOD actions, not determinism alone**.

Why this matters. Together, these results show that CQL's conservatism (its $\log\sum\exp$ penalty) changes the *policy* without fixing the *evaluation*. The comparison we actually trust is BC versus IQL, both of which sit near the behavior reference, and CQL's value is meaningful only once this caveat is attached. We regard this as the central message of the work: a conservative *learner* is not the same as a trustworthy *evaluator*, and in this setting reliable OPE is what determines how far the approach can go.

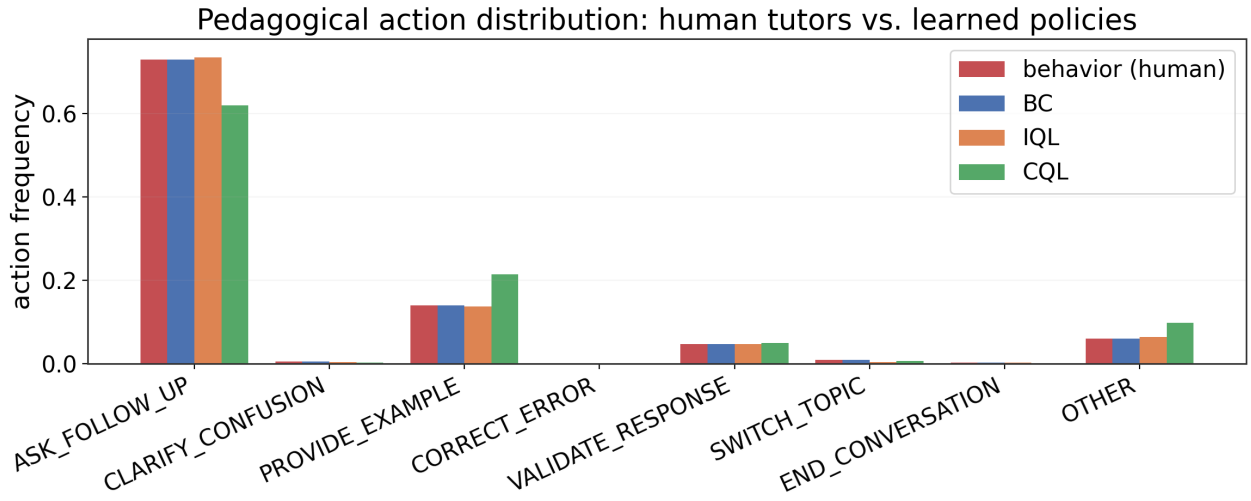

Figure 3: Pedagogical action distribution. BC and IQL track human tutors closely; CQL shifts a small fraction of mass toward `PROVIDE_EXAMPLE` and `OTHER`.

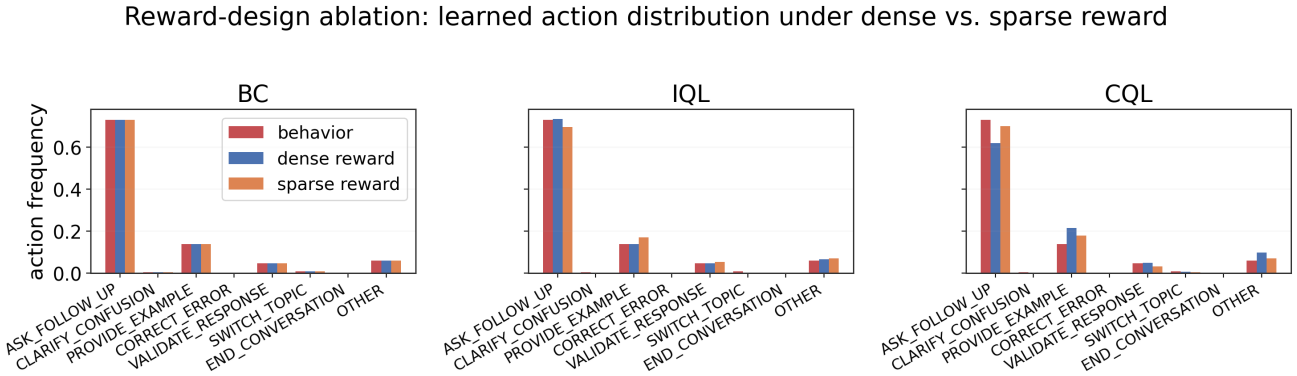

Figure 4: Action mix under the dense vs. sparse reward. BC is invariant (a control); CQL is most reactive, with the dense reward favoring examples and the sparse reward favoring follow-up questions.

7 Discussion

Our results overturn the naive reading of the leaderboard. At face value CQL is the strongest policy, but on closer inspection the seemingly unremarkable methods (BC and IQL) are the trustworthy ones, while the apparent winner is an evaluation artifact. For practitioners applying offline RL to logged human–AI interaction, we offer three recommendations. First, anchor OPE against a behavior reference and treat large positive gaps as hypotheses to be falsified rather than as confirmed gains. Second, probe any suspected over-estimation with temperature or determinism sweeps together with in-distribution controls. Third, favor methods such as IQL that avoid OOD queries when no independent, trustworthy evaluation channel is available. More broadly, our findings underscore that in offline RL the evaluation and learning problems are entangled: a conservative *learner* does not imply a conservative *evaluator*.

8 Conclusion

In this work we framed conversational language tutoring as an offline reinforcement-learning problem over eight high-level pedagogical moves, learning a director policy from 503 logged tutoring episodes. Under well-calibrated off-policy evaluation, Behavior Cloning and Implicit Q-Learning both recover human-level engagement value. Conservative Q-Learning's apparent 58% improvement is different: through a temperature ablation and an in-distribution control, we show it to be off-policy-evaluation over-estimation driven by determinism toward out-of-distribution actions. The principal lesson is that reliable evaluation, rather than policy learning, is the binding constraint in offline tutoring RL. Our hypothesis that a conservative method would clearly outperform the alternatives proved more nuanced than anticipated, and future work will therefore address the evaluation bottleneck directly. We plan closed-loop testing against LLM-simulated students, complemented by LLM-based action relabeling, a learned correctness reward, and a second OPE estimator for triangulation.

9 Limitations and Future Work

We note several limitations. Our action labels come from a keyword heuristic rather than ground truth, and some moves (`CORRECT_ERROR`, `CLARIFY_CONFUSION`) are quite rare (< 60 samples). Our MiniLM encoder is frozen and generic rather than tutoring-specific. The engagement reward is also only a proxy for *learning*, since longer replies are not always better learning. FQE, too, is a single estimator; we mitigate its uncertainty with a behavior reference, a temperature ablation, and controls, but we do not claim a ground-truth policy value. Finally, our results are on a single lesson type, so we have not yet tested how well they generalize across lessons.

For future work, we plan to (i) replace the keyword labeler with an LLM-based action classifier; (ii) ablate the granularity of the action space; (iii) learn a reward model for whether the student successfully answered, possibly via preference learning from real user-rating feedback; (iv) add a second OPE estimator (weighted importance sampling using BC as the behavior-policy estimate) to triangulate FQE; and (v) pursue closed-loop evaluation by deploying the learned director inside the live tutoring engine and measuring engagement against LLM-simulated student personas, which would supply the trustworthy evaluation channel that offline RL otherwise lacks.

A Hyperparameters

References

[1] I. Kostrikov, A. Nair, and S. Levine. Offline reinforcement learning with implicit Q-learning. *arXiv:2110.06169*, 2021.

[2] A. Kumar, A. Zhou, G. Tucker, and S. Levine. Conservative Q-learning for offline reinforcement learning. In *NeurIPS*, 2020.

[3] K. VanLehn. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221, 2011.

[4] S. Young, M. Gašić, B. Thomson, and J. D. Williams. POMDP-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179, 2013.

Table 4: Hyperparameters. All learners are 2-hidden-layer MLPs (width 256, ReLU) over frozen 384-d MiniLM embeddings; 5 seeds per configuration; Modal T4 GPU.

Component	Settings
Embedding	all-MiniLM-L6-v2, 384-d, L_2-normalized
MLP	2 hidden layers $\times$ 256, ReLU
Discount γ	0.99
BC	lr 10^{-3}, 4000 steps, batch 512
IQL	lr 3×10^{-4}, 6000 steps, batch 512, expectile τ=0.7, AWR β=3.0, adv. clip 100, target EMA 0.005
CQL	lr 3×10^{-4}, 6000 steps, batch 512, α=1.0, target EMA 0.005, double-DQN
FQE	lr 10^{-3}, 6000 steps, batch 512, target EMA 0.005
Seeds	5 per configuration
GPU	Modal T4