Consistency-Aware Imitation Learning from Noisy Demonstrations

Seongil Heo
University of Utah
Salt Lake City, UT, USA
seongil.heo@utah.edu

Blake Lawlor
University of Utah
Salt Lake City, UT, USA
u1392237@utah.edu

Dongjoo Lee
University of Utah
Salt Lake City, UT, USA
u1391800@utah.edu

Robbie DeMars
University of Utah
Salt Lake City, UT, USA
u1411323@utah.edu

Abstract

Imitation learning from human demonstrations can fail when feedback is noisy or inconsistent, since standard behavioral cloning has no mechanism to detect whether similar states received contradictory actions. We propose Consistency-IL, a consistency-aware extension of ThriftyDAgger that encodes expert-control trajectory segments, retrieves similar past segments, and filters, corrects, or re-queries segments whose actions conflict with prior behavior in similar contexts. On PandaPickAndPlace, feedback noise is more damaging than bootstrap demonstration noise, and Consistency-IL improves over ThriftyDAgger particularly under medium and high online feedback noise, while remaining competitive when feedback is mostly clean, and outperforms ThriftyDAgger across all bootstrap demonstration counts.

1 Introduction

Behavioral cloning (BC) treats imitation learning as supervised regression over expert state-action pairs, avoiding manual reward design and unsafe exploration. Its weakness is that the learned policy inherits the quality of its labels. When human demonstrations contain inconsistent or noisy actions, BC has no mechanism to identify which labels are trustworthy, and covariate shift amplifies any errors once the policy leaves the training distribution.

Interactive imitation learning methods such as DAgger and ThriftyDAgger reduce covariate shift by collecting corrective feedback on states visited by the learner [7, 4]. However, these methods assume that queried feedback is reliable once obtained. In practice a demonstrator may hesitate, over-correct, or give a locally plausible action that contradicts how they behaved in a similar context.

This project studies whether checking the consistency of feedback before committing it to the training set makes interactive imitation learning more robust. Our hypothesis is that large action discrepancies between segments in similar contexts are useful evidence of noisy supervision, and that filtering or correcting such segments improves policy quality without requiring additional expert queries.

Our contributions are:

- We implement Consistency-IL, a segment-level consistency filter on top of ThriftyDAgger using an LSTM trajectory encoder, cosine-similarity retrieval, conflict detection, and automatic re-query.

Class Project for Advanced AI, Spring 2026.

- We evaluate on LunarLanderContinuous-v3 and PandaPickAndPlace-v3 under separated bootstrap and online noise, finding that feedback noise is more damaging than demonstration noise and the full consistency pipeline improves success under medium and high feedback noise on PandaPickAndPlace.

2 Related Work

2.1 Imitation Learning and Dataset Aggregation

Behavioral cloning reduces imitation learning to supervised learning on expert state-action pairs. Its main failure mode is covariate shift: small errors move the learner into states not covered by the expert dataset, where further errors compound. DAgger addresses this by rolling out the learner, querying an expert on visited states, and aggregating the resulting labels into the training set [7]. ThriftyDAgger makes this process more practical by querying only when the learner appears novel or risky, using ensemble disagreement and a learned risk signal to gate expert intervention [4]. Our method keeps this budget-aware interaction structure, but adds a second question before storing new feedback: is the new expert-control segment consistent with previously observed behavior in similar contexts?

2.2 Learning from Human Feedback

Human feedback is not simply a clean oracle label. COACH shows that feedback is policy-dependent: the same human signal can mean different things depending on what the learner was about to do [6]. Deep TAMER learns from scalar human feedback in high-dimensional state spaces, focusing on reward prediction rather than direct action imitation [9]. These systems highlight the usefulness and difficulty of human-in-the-loop learning. Consistency-IL differs because it treats the feedback as action labels for imitation learning and focuses on conflicts among labels after they are collected, rather than learning a reward model from evaluative feedback.

2.3 Noise-Robust Imitation and Preference Learning

Several methods address imperfect demonstrations. D-REX learns a reward function from ranked demonstrations, allowing the learner to exceed suboptimal demonstrators [2]. Counterfactual Behavior Cloning augments imperfect demonstrations with counterfactual actions to infer intended behavior [8]. CANDERE-COACH studies reinforcement learning from noisy human feedback and explicitly denoises feedback before policy updates [5]. These methods are related in spirit, but they usually operate at the reward, trajectory-ranking, or sample-augmentation level. Our approach instead operates on contiguous takeover segments and asks whether actions disagree across similar local contexts.

2.4 Experimental Domains

We use Gymnasium's LunarLanderContinuous-v3 as a continuous-control benchmark with dense but failure-prone landing dynamics [1]. We also use PandaPickAndPlace-v3 from panda-gym, a goal-conditioned robotic manipulation environment where success requires coordinated reaching, grasping, lifting, and placing [3].

3 Problem Statement

We consider imitation learning in a Markov decision process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \rho_0, \gamma)$ with continuous state space $\mathcal{S}$ and action space $\mathcal{A}$. The learner seeks a policy $\pi_\theta(a \mid s)$ that maximizes expected return

$$J(\pi_\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T-1} \gamma^t R(s_t, a_t) \right], \tag{1}$$

but it does not directly optimize this reward during data collection. Instead, it receives a set of demonstrations or online corrective labels from an expert.

Let the collected human-labeled dataset be $\mathcal{D}_H = \{(s_i, \tilde{a}_i, r_i, s'_i, d_i)\}_{i=1}^N$, where $d_i \in \{0, 1\}$ is the terminal flag, and $\tilde{a}_i$ may differ from the intended expert action $a_i^\star$ because of noise, hesitation, or inconsistent human control:

$$\tilde{a}_i = a_i^\star + \epsilon_i. \tag{2}$$

The noise ϵ_i is not assumed to be independent or uniformly random. It can be structured over time, meaning an entire takeover segment may be suboptimal, or isolated to a few individual actions.

The key challenge is to learn a high-performing policy from $\mathcal{D}_H$ while using a limited expert-intervention budget. A standard BC objective

$$\min_\theta \mathbb{E}_{(s,\tilde{a})\sim\mathcal{D}_H} \left[\|\pi_\theta(s) - \tilde{a}\|_2^2\right] \tag{3}$$

treats all labels as equally reliable. Interactive methods improve state coverage by adding labels from learner-visited states, but they still suffer if those new labels are noisy.

We define a trajectory segment $c = (s_{t:t+L-1}, a_{t:t+L-1}, r_{t:t+L-1})$ as a contiguous block of expert control. Given a candidate segment c and a segment buffer $\mathcal{B}$ of previous segments, the goal is to decide whether c should be accepted into $\mathcal{D}_H$, corrected, re-queried, or discarded. This decision should be based on local consistency: if two segments represent similar contexts, their actions should not differ substantially unless there is a meaningful reason. The overall problem is therefore:

$$\max_{\pi_\theta,\, g} J(\pi_\theta;\, g) \quad \text{s.t.} \quad \sum_t \mathbf{1}\{\text{expert queried at } t\} \le B, \tag{4}$$

where g is a data-admission rule that controls which candidate segments enter $\mathcal{D}_H$, and B is the available intervention budget.

4 Method

The key idea is that noisy feedback is often inconsistent before it is obviously bad. If a new expert-control segment is embedded near past segments but asks for very different actions, the learner should not immediately trust it. We build Consistency-IL as a post-episode filter inside a ThriftyDAgger-style training loop.

4.1 Base Interactive Learner

The base learner is ThriftyDAgger. The policy is an ensemble of behavior-cloned networks; novelty is measured by the ensemble standard deviation:

$$n(s_t) = \sqrt{\mathrm{Var}_m[\pi_m(s_t)]}. \tag{5}$$

A separate Q-network estimates a risk-like value $Q(s_t, \bar{a}_t)$. The system asks for expert control when

$$n(s_t) > \tau_n \quad \text{or} \quad Q(s_t, \bar{a}_t) < \tau_q, \tag{6}$$

and τ_n, τ_q are adapted from recent rollout statistics to stay near the target intervention budget.

4.2 Segment Encoding

After an episode, contiguous expert-control blocks of length at least L are extracted. Each block c is encoded by concatenating state and action at each step and passing the sequence through an LSTM trajectory encoder:

$$z_c = f_\phi([s_t, a_t]_{t=1}^L). \tag{7}$$

The encoder is pre-trained with supervised contrastive learning over bootstrap expert segments, so nearby embeddings represent similar local contexts.

4.3 Segment Memory and Conflict Detection

Accepted segments are stored in memory $\mathcal{B}$. For a candidate c, we retrieve the top-k stored segments by cosine similarity and compute action and reward discrepancies to each match:

$$\Delta_a(c, c') = \|\bar{a}_c - \bar{a}_{c'}\|_2, \qquad \Delta_r(c, c') = |\bar{r}_c - \bar{r}_{c'}|. \tag{8}$$

Let $\hat{s} = \max_{c'\in\mathcal{B}} \mathrm{sim}(z_c, z_{c'})$ be the maximum cosine similarity to any stored segment, and let f_{conflict} be the fraction of retrieved neighbors whose mean action discrepancy exceeds τ_a. The rule-based decision policy maps these values to four outcomes:

- **Keep**: $\hat{s} < \tau_{\text{sim}}$; no sufficiently similar reference exists, so the block is accepted without modification.
- **Discard**: $\hat{s} \geq \tau_{\text{sim}}$ and $f_{\text{conflict}} \geq \tau_{\text{frac}}$; a majority of similar neighbors strongly disagree, so the entire block is dropped.
- **Correct**: $\hat{s} \geq \tau_{\text{sim}}$, $f_{\text{conflict}} < \tau_{\text{frac}}$, and only a small number of individual time steps have $\|a_t - \bar{a}_{c'}\|_2 > \tau_a$; those steps are replaced with the corresponding nearest-neighbor actions.
- **Requery**: $\hat{s} \geq \tau_{\text{sim}}$ and the conflict is moderate (between Correct and Discard thresholds); the block is re-labeled by substituting an action chunk from the most similar stored segment.

Algorithm 1 summarizes the full post-episode filtering procedure.

Algorithm 1 Consistency-IL post-episode feedback filtering

```
Collect an episode with ThriftyDAgger switching.
Extract contiguous expert-control blocks of length at least L.
for each candidate block c do
    Encode c into z_c with the LSTM trajectory encoder.
    Retrieve similar memory segments using cosine similarity.
    Compute action and reward gaps to similar segments.
    Decide Keep, Discard, Correct, or Requery.
    if Keep then
        Add c to the human buffer and segment memory.
    else if Correct or Requery then
        Replace the selected actions, then add the clarified block.
    else
        Drop the block from the human buffer.
    end if
end for
Retrain the ensemble policy and Q-network on the updated buffers.
```

5 Experimental Design

5.1 Hypotheses

We test four hypotheses:

- **H1**: Consistency-aware filtering improves robustness relative to BC and ThriftyDAgger when feedback is noisy.
- **H2**: Feedback noise during online takeover is more damaging than noise in the initial bootstrap demonstrations.
- **H3**: Similarity-only filtering is insufficient; conflict detection plus correction or re-query is responsible for the main gain.
- **H4**: More bootstrap demonstrations improve performance, but Consistency-IL should remain better than ThriftyDAgger across demo counts.

5.2 Domains and Data

We evaluate on two environments. LunarLanderContinuous-v3 is a continuous-control landing task with a two-dimensional action space for main and side thrusters. PandaPickAndPlace-v3 is a robotic manipulation task in which a Franka Panda arm must pick up an object and move it to a target location. The Panda domain is the harder and more informative setting because a short sequence of bad gripper or end-effector commands can prevent task completion.

For automated experiments, each environment provides a heuristic expert that stands in for a human. Bootstrap demonstration noise is injected by adding Gaussian noise to generated expert demonstrations. Online feedback noise is injected during expert takeover. We vary both the magnitude of the noise and the probability that a given expert action is corrupted.

5.3 Methods Compared

We compare:

- **BC**: behavior cloning from bootstrap demonstrations only.
- **ThriftyDAgger**: online query-efficient imitation learning with novelty and risk gating, but no consistency filter.
- **Consistency-IL similarity only**: filters using encoder similarity but does not perform conflict-based re-query.
- **Consistency-IL no requery**: detects conflicts and corrections, but disables the segment-level re-query stage.
- **Consistency-IL full**: uses similarity, conflict detection, correction, and automatic re-query.

5.4 Experiments

Table 1 summarizes the four experiments conducted in this work, each targeting one or more of the hypotheses above.

Table 1: Experiments run in this work.

Experiment	Focus	Runs
Ablation	Component contribution under fixed medium noise	30
Demo and feedback noise grid	Demo noise vs. feedback noise, 4×4 grid	216
Feedback frequency sweep	Noise probability at magnitude 0.3	107
Bootstrap demo count sweep	Number of initial demos (1, 3, 5, 10)	71

5.5 Metrics

The primary metric is autonomous success rate after training, evaluated without expert switching. We also report mean return, total expert steps, and filter statistics (conflicts detected, corrected, re-labeled, discarded segments). A hypothesis is supported when success rate and return improve at comparable expert step counts, especially on PandaPickAndPlace.

6 Results and Discussion

6.1 Ablation

Table 2 summarizes the component ablation. On LunarLander, Consistency-IL with conflict handling reaches 100% success, while BC and ThriftyDAgger are much weaker in this fixed medium-noise setting. On PandaPickAndPlace, the full method improves success from 16.7% for ThriftyDAgger to 28.3%. The similarity-only variant is weak relative to the full method, especially on Panda, which supports H3: merely embedding and comparing segments is not enough; the benefit comes from using conflicts to correct, re-label, or discard data.

Table 2: Ablation results averaged over three seeds. Success is autonomous success rate after training.

Method	Lunar success	Lunar return	Panda success	Panda return
BC	6.7%	-106.6	3.3%	-48.3
ThriftyDAgger	20.0%	-21.0	16.7%	-44.1
Consistency-IL similarity only	26.7%	4.0	6.7%	-48.2
Consistency-IL no requery	100.0%	267.9	20.0%	-41.9
Consistency-IL full	100.0%	210.8	28.3%	-43.3

Figure 1 visualizes these per-variant differences. The segment statistics explain the underlying trend. In the full Consistency-IL condition, the system corrected an average of 14.7 segments on LunarLander and 2.0 segments on PandaPickAndPlace. Requery was rare in automated mode, which

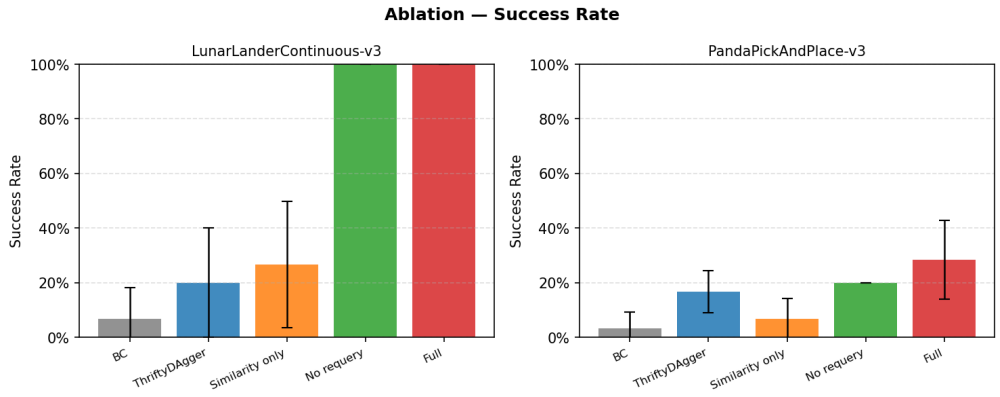

Figure 1: Ablation success rates for BC, ThriftyDAgger, and Consistency-IL variants.

means most of the measured improvement came from conflict-aware correction and discarding rather than frequent requery. This is useful but also reveals a limitation: the automatic re-query mechanism is conservative and only replaces a chunk when nearby memory segments provide sufficient support.

6.2 Demo Noise vs. Feedback Noise

The demo-feedback grid is the clearest robustness test. LunarLander is mostly saturated: across the online grid, both ThriftyDAgger and Consistency-IL average 99.6% success. PandaPickAndPlace is more informative. With high bootstrap demonstration noise, increasing feedback noise sharply reduces success for both online methods, but Consistency-IL is consistently better at the noisier end: under high demo and feedback noise, success is 5.0% for Consistency-IL and 1.7% for ThriftyDAgger. Averaged across all Panda demo-noise levels, Consistency-IL improves medium-feedback success from 19.6% to 29.6%, and high-feedback success from 3.3% to 13.8%.

Table 3: PandaPickAndPlace robustness when averaging across all bootstrap demo noise levels. Each cell uses twelve runs.

Feedback noise	Thrifty success	Consistency success	Thrifty return	Consistency return
Medium	19.6%	29.6%	-45.1	-42.9
High	3.3%	13.8%	-48.6	-46.1

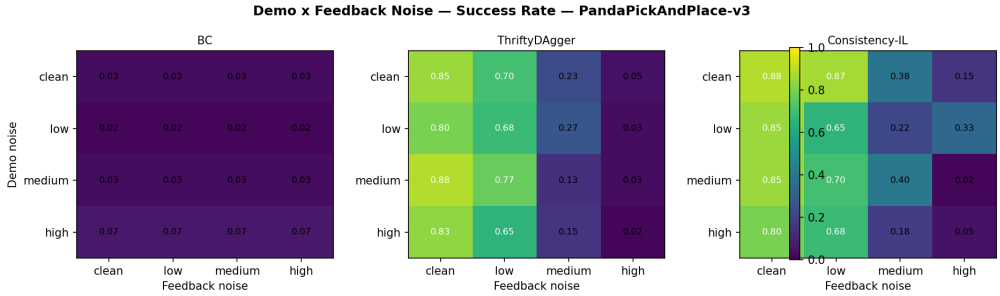

Figure 2: PandaPickAndPlace success rates in the demo-feedback noise grid.

Table 3 and Figure 2 summarize these results, which support H1 and H2, but with nuance. Feedback noise is more harmful than demo noise because it enters during states visited by the partially trained learner; bad labels at those states directly affect future corrections. However, Consistency-IL is not uniformly better at low noise. When the expert feedback is mostly clean, ThriftyDAgger can perform as well as or better than the consistency filter because extra filtering adds conservatism without much noise to remove.

6.3 Feedback Noise Frequency

The frequency sweep fixes feedback noise magnitude at 0.3 and changes only the probability of corrupting an expert action. On PandaPickAndPlace with medium demo noise, ThriftyDAgger performs better at lower noise probabilities, but Consistency-IL becomes better when noisy feedback is frequent. At probability 0.75, Consistency-IL reaches 46.7% success compared with 36.7% for ThriftyDAgger. At probability 1.00, the gap is 21.7% versus 18.3%, and Consistency-IL also has a better mean return.

Table 4: PandaPickAndPlace feedback-frequency sweep with medium bootstrap demo noise and feedback-noise magnitude fixed at 0.3.

Noise probability	Thrifty success	Consistency success	Thrifty return	Consistency return
0.25	63.3%	58.3%	-34.8	-36.4
0.50	61.7%	51.7%	-35.4	-37.6
0.75	36.7%	46.7%	-41.0	-40.1
1.00	18.3%	21.7%	-47.5	-44.0

Figure 3: Success rate as the probability of noisy online feedback increases.

Table 4 and Figure 3 show this trend, which clarifies what the method is buying. Consistency-IL is most useful when feedback corruption is common enough that blindly aggregating feedback would systematically poison the human buffer. When corruption is rare, the filter can reject or alter useful data and therefore may not improve performance.

6.4 Bootstrap Demonstration Count

The bootstrap-count sweep supports H4; Table 5 and Figure 4 summarize the results. On PandaPickAndPlace, Consistency-IL dominates ThriftyDAgger across every initial demo count. With only one bootstrap episode, Consistency-IL reaches 31.7% success while ThriftyDAgger reaches 11.7%. With ten bootstrap episodes, the gap is 51.7% versus 38.3%. BC remains poor in this domain because the initial demonstrations alone do not cover enough learner-induced states.

Table 5: PandaPickAndPlace bootstrap-count sweep. One BC run at demo count 3 is excluded because of a method-label mismatch, so that BC value uses two runs.

Method	1 demo	3 demos	5 demos	10 demos
BC	0.0%	12.5%	1.7%	0.0%
ThriftyDAgger	11.7%	23.3%	30.0%	38.3%
Consistency-IL	31.7%	40.0%	36.7%	51.7%

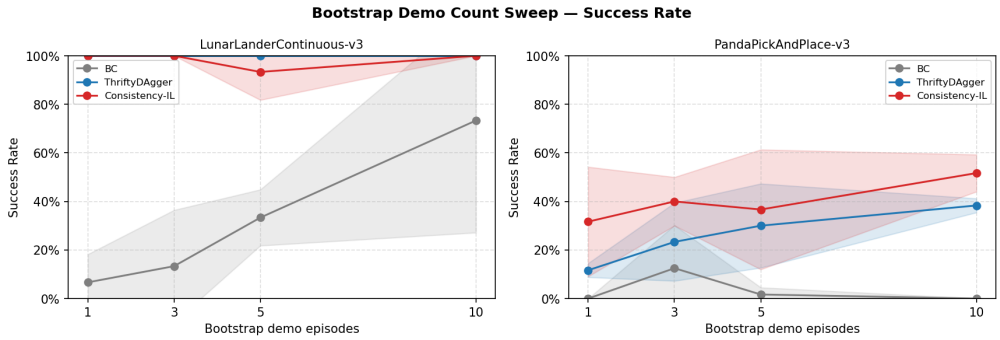

Figure 4: Effect of the number of bootstrap demonstrations on autonomous success rate.

6.5 Overall Discussion

The main conclusion is that Consistency-IL helps most when the environment is hard and the online feedback is noisy. LunarLander is useful as a sanity check, but it saturates: most online methods eventually solve it. PandaPickAndPlace separates the methods because the task requires a longer sequence of correct decisions and is more sensitive to inconsistent action labels.

Consistency checking is not free. When feedback is mostly clean, ThriftyDAgger can match Consistency-IL because extra filtering adds conservatism without noise to remove.

7 Conclusion and Future Work

We studied imitation learning under noisy demonstrations and corrective feedback, where interactive methods risk collecting more data while still trusting bad labels. Consistency-IL addresses this by checking expert-control trajectory segments against similar past segments before adding them to the human buffer.

The experiments show three main findings. LunarLanderContinuous-v3 is largely saturated by online imitation methods, limiting its diagnostic value. PandaPickAndPlace-v3 is more informative: feedback noise is more damaging than demonstration noise, and Consistency-IL improves under medium and high noise. The ablation confirms that conflict-aware correction and re-query drive the gain; similarity alone is insufficient.

There are several limitations. The experiments use heuristic experts with controlled noise rather than real humans; future work should evaluate with live demonstrators. The re-query stage is conservative and rarely relabels segments; stronger retrieval or uncertainty-aware selection could improve it. The conflict detector also relies on hand-tuned thresholds. With more time, we would learn the data-admission rule directly, test richer embeddings, and evaluate on additional manipulation tasks.

Code Availability. Code is available at `https://github.com/ReQuery-IL/Consistency-IL`.

References

[1] John U. Balis, Mark Towers, Ariel Kwiatkowski, J. K. Terry, Ryan Perez, Omar Younis, and Richard M. Liaw. Gymnasium: A standard interface for reinforcement learning environments. In *Thirty-eighth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

[2] Daniel S Brown, Wonjoon Goo, and Scott Niekum. Better-than-demonstrator imitation learning via automatically-ranked demonstrations. In *Conference on robot learning*, pages 330–359. PMLR, 2020.

[3] Quentin Gallouédec, Nicolas Cazin, Emmanuel Dellandréa, and Liming Chen. panda-gym: Open-source goal-conditioned environments for robotic learning. In *4th Robot Learning Workshop: Self-Supervised and Lifelong Learning at NeurIPS*, 2021.

[4] Ryan Hoque, Ashwin Balakrishna, Ellen Novoseller, Albert Wilcox, Daniel S Brown, and Ken Goldberg. Thriftydagger: Budget-aware novelty and risk gating for interactive imitation learning. *arXiv preprint arXiv:2109.08273*, 2021.

[5] Yuxuan Li, Srijita Das, and Matthew E Taylor. Candere-coach: Reinforcement learning from noisy feedback. *arXiv preprint arXiv:2409.15521*, 2024.

[6] James MacGlashan, Mark K Ho, Robert Loftin, Bei Peng, Guan Wang, David L Roberts, Matthew E Taylor, and Michael L Littman. Interactive learning from policy-dependent human feedback. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

[7] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[8] Shahabedin Sagheb and Dylan P Losey. Counterfactual behavior cloning: Offline imitation learning from imperfect human demonstrations. *arXiv preprint arXiv:2505.10760*, 2025.

[9] Garrett Warnell, Nicholas Waytowich, Vernon Lawhern, and Peter Stone. Deep tamer: Interactive agent shaping in high-dimensional state spaces. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.