A^{2}IQL: Adaptive Asymmetric Implicit Q Learning for Automated Warehouse Consolidation

Guangyi Liu
Amazon
gyliu@amazon.com

Andrea Angiuli
Amazon
aangiuli@amazon.com

Mirko Ristivojevic
Amazon
ristivo@amazon.com

Joseph W Durham
Amazon
josepdur@amazon.com

Michael Caldara
Amazon
caldaram@amazon.com

Michael M. Zavlanos
Duke University and Amazon
michael.zavlanos@duke.edu

Abstract

We apply offline reinforcement learning to large-scale warehouse consolidation, an application that requires ranking tens of thousands of candidates at each decision epoch with an architecture that scales gracefully with variable-cardinality action spaces. Since the available reward signal is only an approximation of the true target-domain objective, the policy must also transfer robustly. We propose a two-stage framework, Asymmetric Implicit Q-Learning (AIQL) and its adaptive cross-domain extension A^{2}IQL. First, AIQL learns a scoring policy from fixed offline consolidation logs by decoupling a per-candidate scoring actor from a critic over a fixed-dimensional, order-aware aggregate of the induced ranking, keeping value estimation tractable and within the support of logged data. IQL is one instantiation of this asymmetric decomposition. Second, A^{2}IQL adapts policy extraction from the source-domain critic to a new target objective using scarce target-domain data: a Bayesian reward predictor supplies an uncertainty-penalized conservative bonus that modifies only the advantage-weighted actor extraction, leaving the source critic unchanged. On a single-domain benchmark, AIQL improves throughput by 19% over a heuristic baseline using offline data alone. When adapted to a new target objective via A^{2}IQL, the method outperforms source-only training, data pooling, and off-dynamics baselines on average by 9% to 17% under a $30{:}1$ source-to-target data ratio, and it matches or exceeds transfer baselines under reward shift on a synthetic ranking benchmark and MuJoCo HalfCheetah.

1 Introduction

Automated storage and retrieval systems (ASRS) such as Amazon Robotics' Sequoia [1] are used in Amazon's warehouse fulfillment; these facilities may house millions of items spread across hundreds of thousands of storage containers. As items are picked to satisfy demand, container fullness depletes: a once full container may store only a few items, leaving most of its capacity unused. Left unchecked, this *inventory fragmentation* steadily erodes storage utilization. The facility runs out of space for incoming goods not because every container is full, but because a growing fraction of containers are nearly empty. The consolidation workflow is the principal remedy: items from under-utilized *donor* containers are transferred into non-full *destination* containers, and the emptied donor containers are reclaimed for reuse. Effective choice of containers for consolidation improves throughput by selecting donors whose items pack efficiently into destinations, emptying donors faster and increasing the rate at which storage capacity is reclaimed, while keeping long-term inventory fragmentation in check.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

In container-based ASRS, consolidation operates by ranking candidate containers according to their suitability for processing and routing the highest-ranked candidates to available stations. Scoring policies that produce these rankings are simple, scalable, and compatible with diverse execution systems. However, hand-designed scoring rules are typically myopic: they optimize immediate container-level properties without accounting for how today's consolidation decisions reshape the inventory distribution that future operations must handle.

Reinforcement learning offers a principled framework for this kind of long-horizon sequential decision-making, but three characteristics of the consolidation setting make standard approaches difficult to apply. First, each decision epoch requires ranking a large, time-varying candidate set—potentially thousands of containers—making direct action enumeration infeasible (**variable-cardinality action space**). Second, realistic inventory models must track large-scale item flows across coupled workflows, making online or environment-in-the-loop training prohibitively costly (**costly environment interaction**). Third, even with abundant training data under one reward function, the target environment may weight throughput and other operational metrics differently, creating a mismatch between training and target-domain objectives (**objective mismatch**).

This paper addresses the first two challenges through an offline reinforcement learning framework called Asymmetric Implicit Q-Learning (AIQL). AIQL learns a scoring policy entirely from fixed historical logs, eliminating the need for online exploration or environment interaction during training. The policy assigns each candidate container a scalar score based on local container features and a compact global state summary, and the resulting scores induce a ranking that determines which containers are routed to consolidation stations. Since directly evaluating the value of an entire ranked list is intractable in a variable-cardinality action space, AIQL decouples policy execution from value estimation: the actor scores individual containers to produce fine-grained rankings, while the critic evaluates only a fixed-dimensional aggregate statistic of the induced ranking. This asymmetric design keeps the critic within the support of logged data without requiring enumeration over combinatorial actions. Building on Implicit Q-Learning (IQL) [2], AIQL avoids explicit maximization over unseen actions and extracts high-value behavior through advantage-weighted policy extraction, with IQL as one instantiation of the asymmetric decomposition that this paper contributes (Section 6.4).

To address the third challenge, objective mismatch between training and target domains, we propose Adaptive Asymmetric IQL (A^2IQL). Learning from scarce target logs alone gives high-variance policies, and naively pooling training and target data dilutes the scarce target rewards (Tables 2 and 10). A^2IQL instead assigns distinct roles to each data source: training-domain logs drive long-horizon critic learning, while target-domain logs train a Bayesian reward predictor whose uncertainty-penalized lower confidence bound provides a conservative bonus during actor extraction. This bonus adds weight only to source transitions whose conservative target-reward estimate exceeds a threshold. Because the target data train only the reward predictor and the critic is learned from source data alone, A^2IQL addresses target domains that differ from the training domain in their reward or objective.

Our main contributions are:

(1) We propose an asymmetric actor-critic decomposition for offline RL over rankings, with Asymmetric IQL (AIQL) as its IQL instantiation, in which the actor applies a shared scoring function independently to each candidate, reducing policy learning to per-container regression.

(2) The critic evaluates a fixed-dimensional aggregate statistic of the induced ranking, making value learning independent of the candidate set size and keeping it within the support of logged data. Proposition 4.3 bounds the suboptimality when rewards and transitions vary within an aggregate.

(3) We develop A^2IQL, a cross-domain extension that adapts to a new target reward or objective by augmenting advantage-weighted actor extraction with a conservative, uncertainty-penalized bonus from a Bayesian reward predictor, leaving the training critic unchanged.

2 Related Work

Offline reinforcement learning. Offline RL learns policies from fixed datasets without environment interaction [3, 4]. The central challenge is distributional shift: standard off-policy methods may overestimate out-of-distribution (OOD) actions. Prior work addresses this through policy constraints that keep the learned policy close to the behavior distribution [5, 6], conservative value estimation

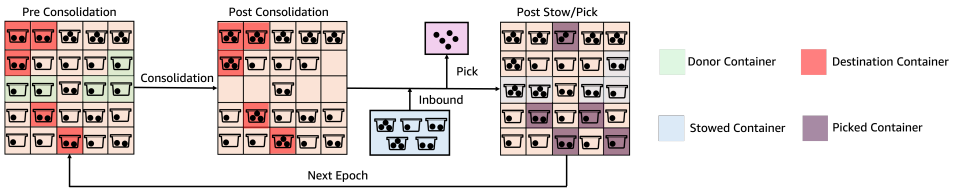

Figure 1: One decision epoch of the consolidation workflow with donor containers (green) selected for emptying and destination containers (red) selected to receive items.

that penalizes Q-values for OOD actions [7], behavior-cloning regularization added to actor-critic learning [8], or sequence modeling that casts offline RL as conditional trajectory generation [9]. Implicit Q-Learning (IQL) [2] approximates the value of the best in-support actions at each state via a state-conditional upper expectile of the Q-function, without explicitly querying OOD actions, and extracts the policy via advantage-weighted regression (AWR) [10]. We build on IQL but introduce an asymmetric actor-critic design in which the actor and critic operate on different action representations to handle variable-cardinality ranking actions. IQL is well suited here because its expectile-based critic never requires maximizing over unseen actions, which is essential when the action space is a combinatorial ranking that cannot be enumerated.

Large and combinatorial action spaces. RL with large or combinatorial action spaces is challenging because direct action enumeration becomes infeasible. Strategies include embedding actions into continuous spaces with nearest-neighbor lookup [11], learning action representations for generalization over large discrete sets [12], decomposing factored actions into lower-dimensional components for offline value learning [13], neural combinatorial optimization that guides search or constructs solutions autoregressively [14, 15], permutation-invariant set architectures for variable-cardinality inputs [16], and branch-based value estimation that captures sub-action dependencies in combinatorial offline RL [17]. Our problem has a distinct structure: the policy must rank a large, time-varying candidate set, and the offline critic must evaluate rankings without leaving the support of logged data. Existing approaches either assume a fixed action vocabulary, require sequential decoding, or rely on sub-action independence—none of which hold for variable-cardinality rankings. We sidestep these limitations by summarizing ranked lists through tier-level aggregate statistics, reducing critic learning to a fixed-dimensional regression independent of candidate set size.

Cross-domain and off-dynamics RL. Transferring policies across domains with differing dynamics or reward functions is a common challenge in applied RL [18]. Off-dynamics RL compensates for mismatch using reward corrections from domain classifiers [19], dynamics-aware reward augmentation that characterizes shift via state-action-next-state triples [20], optimal-transport-based data filtering under dataset-support constraints [21], or contrastive representations that measure the dynamics gap for selective data sharing [22]. Benchmarks such as ODRL [23] provide controlled evaluation settings across diverse dynamics shifts. Hybrid methods combine offline data with online simulator interaction via dynamics-aware policy evaluation [24]. Under severely limited target data, prior work uses support constraints for OOD actions and dynamics [25] or dynamics- and value-aligned filtering [26]. These methods primarily address dynamics mismatch and assume sufficient target data for either critic learning or reliable filtering; under extreme source-to-target imbalance with reward mismatch, filtering discards too much source data while pooling dilutes the critic. Our extension follows a different principle: source logs drive long-horizon critic learning, while scarce target logs adapt only the actor extraction step through a conservative, uncertainty-aware reward bonus, avoiding cross-domain critic corruption entirely.

3 Problem Formulation

3.1 Consolidation MDP

We model the system as a collection of containers whose item contents evolve over discrete decision epochs through three recurring workflows, as illustrated in Figure 1. **Consolidation** transfers items from selected donor containers into non-full destination containers. Donor containers that become empty are removed, freeing capacity for future inbound. For instance, if a donor container holds 2

items and a destination container has sufficient remaining volume, consolidation moves the items into the destination, empties the donor container, and reclaims its storage position on the floor. **Pick** operations remove items from containers to satisfy demand, changing the fragmentation pattern. **Inbound** operations introduce new containers into available capacity. These workflows are coupled: effective consolidation creates empty slots for inbound, while pick alters the inventory distribution that future consolidation must handle. The RL agent controls only the consolidation container selection; pick and inbound are exogenous processes governed by customer demand and upstream supply. Let X_k denote the full partition of items into containers at decision epoch k. The evolution of X_k is governed jointly by consolidation decisions, picked items, and newly inducted containers, but the exact transition is complex and unknown.

We formulate the consolidation control problem as an infinite-horizon discounted Markov Decision Process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$ with discount factor $\gamma \in (0, 1)$. We operate in the offline setting: the agent has no access to the environment during training and instead learns from a fixed dataset $\mathcal{D}$ of transitions collected by a behavior policy π_β.

State space $\mathcal{S}$. The raw inventory partition X_k is intractable as a state representation due to the scale of the problem. We define a compact state that consists of two components $s_k = (F_k, \mathbf{x}_k)$: a global floor-state vector $F_k \in \mathbb{R}^{d_s}$ summarizing distributional statistics of item counts and volumes across all containers, and a variable-size set of candidate container feature vectors $\mathbf{x}_k$ available at decision time. The construction of these representations is detailed in Section 3.2.

Action space $\mathcal{A}_k$. At each epoch k, the consolidation station processes candidates from a ranked list of donor containers and a ranked list of destination containers, consuming them sequentially in rank order. The policy must rank all available candidates to ensure the station does not exhaust its recommendations before the epoch ends. The action a_k therefore corresponds to producing complete ranked lists for both donor and destination containers over the current candidate set.

Transition dynamics P. The transition $s_{k+1} \sim P(\cdot \mid s_k, a_k)$ is governed by the coupled consolidation, pick, and inbound processes. A decision epoch ends once consolidation has emptied enough donor containers to free sufficient capacity for the next round of inbound, at which point the warehouse transitions to the next state.

Reward r. The reward $r_k = r(s_k, a_k)$ measures consolidation throughput at epoch k, defined as the number of emptied donor containers per unit time. In practice, the simulator reward is an approximation: the target domain may weight throughput and other operational metrics differently, creating a reward mismatch that motivates the adaptive extension A^{2}IQL (Section 5).

3.2 State and Action Representation

A fundamental challenge in applying RL to warehouse consolidation is the scale mismatch between evaluation and control. System performance is measured by aggregate metrics such as throughput, yet the policy must make fine-grained, per-container decisions over thousands of candidates. Offline RL methods based on actor-critic architectures [7, 8, 2] are well suited to this setting because the critic can learn long-horizon value estimates from logged data while the actor extracts an improved policy without online interaction. However, the standard assumption that actor and critic share the same action representation breaks down here: a critic that evaluates individual container scores would face an intractably large, variable-cardinality action space, while an actor that operates on aggregate statistics would lose the per-container resolution needed for execution. Our asymmetric design resolves this by defining three complementary representations, each serving a distinct role in the architecture of Section 4. We use deliberately engineered, interpretable features throughout this paper for clarity of presentation; the framework is agnostic to the feature construction and can accommodate learned encoders.

Container features (Actor input). For each candidate container i at epoch k, we construct a feature vector $x_{k,i} = [c_{k,i}, n_{k,i}, V_{k,i}]$ of container type, item count and aggregate volume.

Global state (Actor and Critic input). The floor state is summarized by the epoch index and low-order statistics of the container feature distributions: $F_k = [k, \rho_k, \mu_{n,k}, \sigma_{n,k}, \mu_{V,k}, \sigma_{V,k}] \in \mathbb{R}^6$, where ρ_k is the share of containers of one type. This compact vector is shared by both the Actor and the Critic, capturing the macroscopic inventory condition.

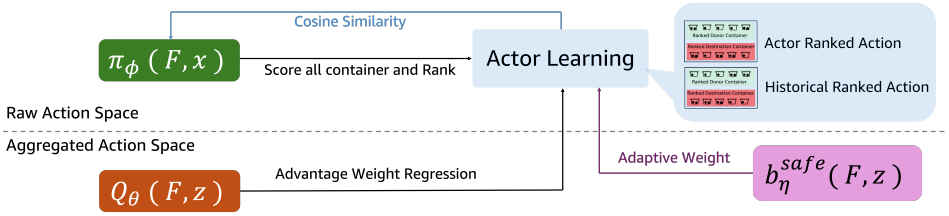

Figure 2: A^{2}IQL framework. The actor $\pi_\phi(F, \mathbf{x})$ scores containers and is trained via cosine similarity to historical rankings, weighted by advantages from critic $Q_\theta(F, z)$. For cross-domain adaptation, $b_\eta^{\texttt{safe}}(F, z)$ provides an adaptive weight modifying actor extraction without changing the critic.

Aggregate action statistics (Critic input). Directly modeling the full ranked list of container indices as an action is intractable for offline value learning: the number of possible rankings grows combinatorially with the candidate set size, and most rankings are never observed in the logged data. Instead, we summarize the induced ranking into a fixed-dimensional vector. Each ranked list (donor and destination) is sorted by score and partitioned into C contiguous groups by rank position, reflecting operational priority levels ($C = 4$ unless stated otherwise). For each tier we compute the mean and variance of the container features within that tier:

$$z_k = \text{Concat}_{L \in \{\text{src, dest}\}} \left(\bigcup_{c=1}^{C} \left[\mu(\mathbf{x}^{L,c}),\ \text{Var}(\mathbf{x}^{L,c}) \right] \right), \tag{1}$$

where $\mathbf{x}^{L,c}$ denotes the set of container feature vectors assigned to tier c of list L, so $z_k \in \mathbb{R}^{48}$ for $C = 4$. The Critic evaluates $Q(F_k, z_k)$ on this aggregate statistic, which depends only on the ranking order induced by the scores. We test in Appendix A whether z_k is sufficient for predicting rewards and next states in this simulator, and Proposition 4.3 bounds the loss when it is not.

4 Asymmetric Implicit Q-Learning (AIQL)

AIQL assumes access to a static offline dataset $\mathcal{D} = \{(F_j, z_j, \mathbf{x}_j, \mathbf{y}_j, r_j, F'_j)\}_{j=1}^{M}$, where $\mathbf{y}_j$ denotes historical container scores from the behavior policy and $z_j = \zeta(\mathbf{y}_j)$ is the corresponding aggregate action statistic, computed as order statistics of the container features $\mathbf{x}_j$ under the ranking induced by $\mathbf{y}_j$. The Actor and Critic operate on different components of each transition, as illustrated in Figure 2 and described below.

4.1 Critic Learning

The key assumption enabling critic learning over aggregate statistics is that the mapping ζ preserves all outcome-relevant information:

Assumption 4.1 (Sufficient action statistics)**.** Let $\zeta : \mathcal{Y} \to \mathcal{Z}$ map container-level score vectors $\mathbf{y}$ to aggregate statistics $z = \zeta(\mathbf{y})$. We assume $P(r, F' \mid F, \mathbf{y}) = P(r, F' \mid F, z)$ for all $(F, \mathbf{y})$.

Proposition 4.2. *Under Assumption 4.1, $Q^*(F, \mathbf{y}) = Q^*(F, z)$ for all $\mathbf{y}$ with $\zeta(\mathbf{y}) = z$. (Proof in Appendix B.)*

Proposition 4.3 (Approximate sufficiency)**.** *Let rewards lie in $[0, R_{\max}]$, and for every F let any $\mathbf{y}, \mathbf{y}'$ with $\zeta(\mathbf{y}) = \zeta(\mathbf{y}')$ differ by at most ε_r in expected reward and by at most ε_P in the total variation of the next-state distribution. Let $\hat{\pi}$ be optimal for the aggregated model in which each z takes the reward and transition of a fixed mixture over $\{\mathbf{y} : \zeta(\mathbf{y}) = z\}$. Every ranking policy π whose aggregate $\zeta(\mathbf{y})$ follows $\hat{\pi}(\cdot \mid F)$ then satisfies $V^*(F) - V^\pi(F) \le \frac{2\varepsilon_r}{1-\gamma} + \frac{2\gamma\varepsilon_P R_{\max}}{(1-\gamma)^2}$ for every F, which is zero under Assumption 4.1 (proof in Appendix B).*

The ε_r term is the loss from reward differences within an aggregate, and the ε_P term is the loss from transition differences, as in standard aggregation bounds [27, 28, 29]. Assumption 4.1 is an idealization that we do not expect to hold universally. However, the structural properties of batch consolidation, e.g., long candidate lists and aggregate-driven floor evolution, make it a close

approximation in our setting, as tested empirically in Appendix A. Since z is fixed-dimensional regardless of the candidate set size, value learning is entirely independent of the number of candidates, making offline critic training feasible even when the candidate set contains tens of thousands of containers. Following IQL [2], we learn a state-value function $V_\psi(F)$ via expectile regression and an action-value function $Q_\theta(F, z)$ via Bellman regression:

$$\mathcal{L}_V(\psi) = \mathbb{E}_{\mathcal{D}}\left[L_2^\tau\big(Q_{\hat{\theta}}(F, z) - V_\psi(F)\big)\right], \qquad \mathcal{L}_Q(\theta) = \mathbb{E}_{\mathcal{D}}\left[\Big(r + \gamma V_\psi(F') - Q_\theta(F, z)\Big)^2\right], \quad (2)$$

where $Q_{\hat{\theta}}$ is a slowly updated target network, $\mathbb{I}(\cdot)$ is the indicator function, and $L_2^\tau(u) = |\tau - \mathbb{I}(u < 0)|\, u^2$ is the asymmetric expectile loss, which upweights positive residuals when $\tau > 0.5$, biasing V_ψ toward the upper tail of the Q-value distribution. Neither loss requires maximization over actions, which is what makes IQL compatible with our aggregate action representation: the critic never needs to evaluate unseen rankings.

4.2 Actor Learning

The Actor $\pi_\phi(F_k, x_{k,i})$ maps the global floor state F_k and an individual container feature vector $x_{k,i}$ to a scalar score, applied independently to each candidate with a single shared parameter set. This reduces policy learning from the combinatorial space of all possible rankings to a per-container regression problem; the ranking used for execution is obtained by sorting the predicted scores. The actor is trained via Advantage-Weighted Regression (AWR) [10, 2], which fits the actor to logged behavior actions with each transition weighted by its estimated advantage:

$$w_i = \exp\big(A(F_k, z_k)/T\big), \qquad A(F_k, z_k) = Q_\theta(F_k, z_k) - V_\psi(F_k), \quad (3)$$

where $T > 0$ controls the sharpness of the weighting. Since consolidation depends only on relative container ranking, we train the actor to match the ranking order of the behavior policy's scores rather than their absolute values. Concretely, for each logged transition, let $\mathbf{y}$ denote the behavior policy's score vector over all candidate containers and $\hat{\mathbf{y}} = \pi_\phi(F_k, \mathbf{x}_k)$ the actor's predicted scores. We optimize a weighted centered cosine distance [30], equivalent to maximizing the advantage-weighted mean of the Pearson correlation [31] between the two score vectors. Let $\tilde{\mathbf{y}}$ and $\tilde{\hat{\mathbf{y}}}$ denote their mean-centered versions. The Actor loss is

$$\mathcal{L}_{\text{Actor}}(\phi) = \mathbb{E}_{\mathcal{D}}\left[w_i\left(1 - \frac{\langle\tilde{\hat{\mathbf{y}}}, \tilde{\mathbf{y}}\rangle}{\|\tilde{\hat{\mathbf{y}}}\|_2 \|\tilde{\mathbf{y}}\|_2}\right)\right]. \quad (4)$$

On the warehouse the actor outputs one score per container for each of the donor and destination lists, the loss sums Equation (4) over the two lists, and each term averages the centered cosine distance over all candidates with the one over the candidates of highest logged score. This objective is invariant to positive affine transformations of scores, recovering the relative preference structure of high-advantage historical decisions. The full training procedure is shown in Appendix C.

5 Cross-Domain Adaptation (A^{2}IQL)

AIQL learns entirely from source-domain logs, but the target domain may weight throughput and other operational metrics differently or add operational constraints as Lagrangian penalty terms, a reward or objective shift that A^{2}IQL addresses. Since target-domain data is scarce (orders of magnitude smaller than the source dataset), the challenge is to leverage limited target-domain reward information to improve the source-trained policy without corrupting the long-horizon value estimates that the abundant source data provides.

Let $\mathcal{D}^{\mathtt{sim}} = \{(F_i, z_i, \mathbf{x}_i, \mathbf{y}_i, r_i, F_i')\}_{i=1}^{N_{\mathtt{sim}}}$ denote the source-domain dataset and $\mathcal{D}^{\mathtt{tgt}} = \{(F_j, z_j, \mathbf{x}_j, \mathbf{y}_j, r_j^{\mathtt{tgt}}, F_j')\}_{j=1}^{N_{\mathtt{tgt}}}$ a target-domain dataset with $N_{\mathtt{tgt}} \ll N_{\mathtt{sim}}$. Both datasets share the same state and action representations but are collected with different reward functions, as discussed in Section 3.1. The source critic (Q_θ, V_ψ) is trained exclusively on $\mathcal{D}^{\mathtt{sim}}$ and remains frozen throughout adaptation. Several natural strategies for incorporating $\mathcal{D}^{\mathtt{tgt}}$ are problematic: training on target data alone is unstable when $N_{\mathtt{tgt}}$ is small; pooling the two datasets risks distorting the critic when rewards differ in scale and $N_{\mathtt{sim}} \gg N_{\mathtt{tgt}}$; and adding a raw reward bonus from a target predictor can overweight out-of-distribution source transitions when the predictor lacks calibrated uncertainty. These considerations motivate a conservative design in which the target-domain bonus is large only when the predictor is both confident and optimistic, and defaults to zero otherwise, preserving the source critic's long-horizon judgment.

5.1 Conservative Target-Domain Reward Estimation

The core idea of A^{2}IQL is to keep the source critic frozen and use target-domain data only to identify which source transitions are likely to be high-value under the target objective. To do this, we need a way to estimate, for each source transition, what reward it would have received under the target reward function, along with a measure of confidence in that estimate.

We train a Bayesian reward predictor on $\mathcal{D}^{\texttt{tgt}}$ and apply it to score transitions from $\mathcal{D}^{\texttt{sim}}$. We adopt Bayesian Linear Regression (BLR) [32] over input features (F, z): the predictor is fit on target-domain transitions $\{(F_j, z_j, r_j^{\texttt{tgt}})\}_{j=1}^{N_{\texttt{tgt}}}$, and then queried on each source transition $(F_i, z_i) \in \mathcal{D}^{\texttt{sim}}$ to obtain a predictive distribution over the target reward:

$$\hat{r}_i^{\texttt{tgt}} \mid F_i, z_i \sim \mathcal{N}\big(\mu_\eta(F_i, z_i),\ \sigma_\eta^2(F_i, z_i)\big), \tag{5}$$

where μ_η estimates what target-domain reward the source transition would have received. BLR models $r^{\texttt{tgt}} = \mathbf{w}^\top \mathbf{h}(F, z) + \epsilon$ with prior $\mathbf{w} \sim \mathcal{N}(0, \alpha_0^{-1} I)$ and noise precision ν, and μ_η and $\sigma_\eta^2 = \nu^{-1} + \mathbf{h}^\top \Sigma \mathbf{h}$ are the posterior predictive moments in closed form, with $\Sigma = (\alpha_0 I + \nu \mathbf{H}^\top \mathbf{H})^{-1}$ for the target design matrix $\mathbf{H}$. The epistemic term $\mathbf{h}^\top \Sigma \mathbf{h}$ grows as $\mathbf{h}$ leaves the span of the target features, so σ_η measures how well the scarce target data determine the prediction at a source input. σ_η therefore accounts for the source-target gap through target-data coverage alone.

Since source transitions are generally out-of-distribution with respect to the target training data, the predictor's mean μ_η alone is unreliable: a high predicted reward is meaningful only when accompanied by low uncertainty. We therefore construct a conservative lower bound that credits a source transition only when the predictor is both optimistic and confident:

$$b_\eta^{\texttt{safe}}(F, z) = \mu_\eta(F, z) - \kappa\, \sigma_\eta(F, z), \tag{6}$$

where $\kappa > 0$ controls the degree of conservatism. Under the Gaussian predictive distribution, $b_\eta^{\texttt{safe}}$ is the lower $(1 - \alpha)$ quantile of the predicted target reward for $\kappa = \Phi^{-1}(\alpha)$ and confidence level $\alpha \in (0.5, 1)$. It also equals the conditional value-at-risk (CVaR) [33], the mean of the lowest $(1 - \alpha')$ fraction of the predictive distribution, at the level α' with $\varphi(\Phi^{-1}(\alpha'))/(1 - \alpha') = \kappa$, where φ is the standard normal density. For example, $\kappa = 1.0$ corresponds to approximately 84% confidence that the true target reward exceeds $b_\eta^{\texttt{safe}}$ and to the CVaR at $\alpha' \approx 0.62$.

5.2 Adaptive Actor Extraction

Given the conservative bound $b_\eta^{\texttt{safe}}$ from the previous section, we now incorporate it into actor extraction. A^{2}IQL leaves the source critic frozen and modifies only the AWR weights used to train the actor: a source transition receives an additional weight only where its conservative target bound $b_\eta^{\texttt{safe}}(F_i, z_i)$ exceeds a threshold τ_b.

We augment the AWR weight with a rectified conservative bonus:

$$\tilde{A}_i = A^{\texttt{sim}}(F_i, z_i) + \lambda\, b_i, \qquad b_i = \frac{1}{s_b} \max\big(b_\eta^{\texttt{safe}}(F_i, z_i) - \tau_b,\ 0\big), \tag{7}$$

$$w_i^{\texttt{adapt}} = \exp\Big(\tilde{A}_i / T\Big), \tag{8}$$

where $A^{\texttt{sim}}(F_i, z_i) = Q_\theta(F_i, z_i) - V_\psi(F_i)$ is the source-domain advantage under the frozen critic, τ_b is a fixed percentile of $b_\eta^{\texttt{safe}}$ over the source transitions, $s_b > 0$ sets the bonus scale, and $\lambda \geq 0$ controls the strength of target-domain preference during policy extraction. The actor objective becomes

$$\mathcal{L}_{\text{Actor}}^{\texttt{adapt}}(\phi) = \mathbb{E}_{\mathcal{D}^{\texttt{sim}}}\left[w_i^{\texttt{adapt}} \left(1 - \frac{\langle \tilde{\hat{\mathbf{y}}}, \tilde{\mathbf{y}} \rangle}{\|\tilde{\hat{\mathbf{y}}}\|_2 \|\tilde{\mathbf{y}}\|_2} \right) \right]. \tag{9}$$

The rectification sets the bonus to zero wherever $b_\eta^{\texttt{safe}} \leq \tau_b$, so these transitions keep the unnormalized weight that their source advantage gives them. Where the bonus fires, a bonus that is large relative to the spread of $A^{\texttt{sim}}$ concentrates the weights on the firing transitions, which is why Equation (7) divides the bonus by its scale s_b. With $\lambda = 0$, A^{2}IQL is advantage-weighted extraction against the frozen source critic. Since λ modulates only the supervised actor loss and not the critic or Bellman

Table 1: Single-domain warehouse throughput (↑) over 40 epochs, mean ± std over 10 training seeds for the learned methods (behavior policy: over its logged runs), with the difference to the behavior policy in parentheses. BC clones the logged rankings, and %BC only the top-advantage logged days.

	Behavior Policy	BC	%BC	AIQL ($C = 4$)
Throughput	68.38 ± 0.95	67.2 ± 0.16 (-1.18)	71.2 ± 1.84 $(+2.82)$	$\mathbf{81.59 \pm 0.69}$ $(+13.21)$

target, A^{2}IQL requires no cross-domain Bellman backup, no joint critic optimization, and no online interaction with either domain.

In summary, A^{2}IQL decomposes cross-domain adaptation into two roles: the source critic provides stable long-horizon value estimates from abundant source data, while the Bayesian target predictor is designed to act as a conservative preference filter that steers actor extraction toward transitions confidently aligned with the target objective. Algorithm 2 trains the critic and the actor in two separate phases. Phase 1 trains the IQL source critic on $\mathcal{D}^{\mathtt{sim}}$ without an actor and freezes it. Phase 2 trains a fresh actor by advantage-weighted regression with the weights of Equation (8).

6 Experiments

The warehouse experiments (Sections 6.2 and 6.3) are conducted entirely on a proprietary simulator, and no policies were deployed in a live environment.

6.1 Experimental Setup

Horizon. Each decision epoch corresponds to one simulated day, and each evaluation rollout spans 40 epochs.

Offline data collection. We collect offline trajectories by rolling out a rule-based heuristic policy. Each decision epoch produces one logged transition $(F, z, \mathbf{x}, \mathbf{y}, r, F')$. This yields $1{,}230$ source-domain transitions in total. The heuristic serves as both the data-generating behavior policy and the primary baseline.

Training. All networks are small MLPs optimized with Adam [34] on an NVIDIA Tesla T4 GPU. Each table caption gives the number of training seeds, and spreads are standard deviations over these seeds. A difference between two methods is the mean of their per-seed differences, and SE is its standard error.

6.2 Single-Domain Results (AIQL)

We train AIQL with throughput as the sole reward. As shown in Table 1, AIQL improves throughput by 19% over the behavior policy from offline logs alone. We compare with behavior cloning (BC) of all logged rankings and with %BC [9], which clones only the top-advantage logged days. BC lands at the behavior policy it imitates, and the margin of AIQL over %BC (10.39) exceeds that of %BC over BC (4.0), which indicates that the gain is not explained by upper-tail imitation of the logs. Every tier count of Table 7 keeps AIQL above the behavior policy with a peak at $C = 3$ to 4, and a 12-dimensional state gives no gain. Robustness on unseen datasets is evaluated in Appendix E.1.

6.3 Cross-Domain Results (A^{2}IQL)

We evaluate A^{2}IQL in a setting where the source and target domains differ in reward objective. The source-domain training dataset $\mathcal{D}^{\mathtt{sim}}$ contains $1{,}230$ transitions collected under a throughput-only reward. The target dataset $\mathcal{D}^{\mathtt{tgt}}$ contains only 41 transitions collected under a target reward that adds a fixed multiple of a second operational metric to throughput. We also evaluate a constraint-penalized objective that subtracts a fixed multiple of a constraint metric from the target reward. A rollout ends early if the policy drives the floor into an infeasible state, and constraint-satisfied days count the days a rollout completes before it ends.

Baselines. We compare A^{2}IQL against six methods, all using the same AIQL backbone for policy learning. **Source-Only** trains on $\mathcal{D}^{\mathtt{sim}}$ only, ignoring target data. **Target-Only** trains on $\mathcal{D}^{\mathtt{tgt}}$ only (41

Table 2: Cross-domain transfer on the warehouse over 5 seeds. Target reward (↑, 40-epoch average, 106.88 for the behavior policy) and penalized objective (↑, first $H = 23$ epochs) of the best checkpoint per seed, mean ± std (superscript: seeds with a completed rollout). Constraint-satisfied days (↑, of 40) under the penalized objective, averaged over all evaluated checkpoints, without a spread (Appendix E.3). Differences to the best baseline use unrounded values.

Method	Target reward	Constraint-satisfied days	Penalized objective
Source-Only	100.93 ± 10.63	37.8	77.2 ± 27.0
Target-Only	101.35 ± 47.53^{4}	37.8	46.5 ± 40.0
Pooled	105.32 ± 11.37	37.8	81.9 ± 26.8
DARA-style	105.61 ± 10.90	37.5	81.1 ± 27.0
OTDF-style	101.62 ± 42.60^{4}	37.6	50.6 ± 44.0
IGDF-style	98.52 ± 50.12^{3}	37.5	47.5 ± 41.9
A^{2}IQL (minus best baseline)	$\mathbf{115.04 \pm 16.86}$ $(+9.43)$	$\mathbf{38.9}$ $(+1.0)$	$\mathbf{87.4 \pm 28.9}$ $(+5.5)$

transitions). **Pooled** trains on $\mathcal{D}^{\mathtt{sim}} \cup \mathcal{D}^{\mathtt{tgt}}$. We also adapt three representative off-dynamics transfer strategies. The **DARA-style** baseline [20, 23] penalizes each source reward by $2(1 - p_{\text{tgt}})$, with p_{tgt} the target probability of a domain classifier on (F, z). **OTDF-style** [21] and **IGDF-style** [22] filtering keep the 30% of source transitions nearest to the target data in normalized (F, z) or in a learned domain-discriminative embedding.

Cross-Domain Results. In Table 2, A^{2}IQL attains the highest mean target reward, 9.43 above the best baseline, and is the top method on 4 of 5 seeds (Appendix E.3), with a larger spread than Source-Only, Pooled and the DARA-style baseline. Under the penalized objective it keeps the constraints satisfied longer on average and has the highest penalized objective. Its lead in days comes from two of the five seeds (Table 8). For every method the constraint metric meets its threshold on about 0.6 of the first $H = 23$ epochs of the selected rollouts. Target-Only and the filtering baselines learn from fewer transitions and vary widely across seeds, and the other baselines learn mostly from the source reward, while A^{2}IQL concentrates its actor on the source days whose inputs align most with the target data (Appendix E.5).

6.4 Synthetic Ranking Benchmark

The synthetic benchmark of Appendix F can be rebuilt from its specification. At each of 20 steps the policy ranks 52 to 60 items of three categories, the top 8 are processed, and the critic sees a 20-dimensional aggregate. The logging policy is deliberately suboptimal and category-agnostic, so cloning it stays suboptimal. Over 20 seeds (Table 10), AIQL exceeds %BC by 13.09 (SE 1.05). SetEnc, which replaces z by a learned DeepSets embedding [35], is above AIQL by 5.21 (SE 1.55), and Asym-CQL, which adds a CQL-style penalty to the aggregate critic, by 1.11 (SE 0.58). This indicates that the gain over imitation comes from the decomposition, with IQL as one instantiation. When an item attribute that z omits enters the reward with weight $\eta \leq 4$, the AIQL return varies by at most 1.04 while the oracle gains up to 162.39, so their ratio falls (Table 12). Under the reward shift with $3{,}000$ source and 40 target transitions, A^{2}IQL exceeds Target-only by 28.37 (SE 6.93) over 15 seeds (Table 10(b)) and every other baseline by at least 56.89. The domains differ only in their rewards, so the baselines other than Target-only learn mostly from the source reward, whose category values the target reverses, while A^{2}IQL weights all $3{,}000$ source transitions with a reward model fitted to the 40 target transitions.

6.5 Reward Shift on MuJoCo HalfCheetah

We test the adaptation step on MuJoCo HalfCheetah [36, 37] with D4RL [38] and Minari [39] data (Appendix G). The target reward $-|v - v^{\star}| - 0.05\|\mathbf{a}\|^2$ favors the cruising speed $v^{\star}$ of a slower gait in the data, and the 200 target transitions come from the same data, so only the reward shifts. Methods are tuned on separate seeds and reported on held-out seeds, with a tuning budget that favors A^{2}IQL (20 and 6 configurations against 5 values per baseline). On D4RL (Table 18), A^{2}IQL is ahead of five of the six baselines on 20 of 20 held-out seeds and ties the tuned IGDF-style filter, ahead by 359 (SE 256) in the mean and on only 8 of 20 seeds. A bonus-only ablation without the source advantage is above A^{2}IQL by 195 (SE 44). On Minari, A^{2}IQL is above all six baselines. The native reward favors

the fastest gait, which is why Source-only, Pooled and A^{2}IQL with $\lambda = 0$ stay far below A^{2}IQL, whose bonus carries the target preference for the slower gait into the actor weights. Results under a shift in the transition dynamics alone, which A^{2}IQL does not target, are in Tables 17 and 19.

7 Conclusion

Our results indicate that offline RL under a reward or objective shift is viable even under extreme data imbalance when target-domain influence is confined to a conservative, uncertainty-penalized bonus during actor extraction. On the synthetic benchmark (75:1) the bonus accounts for the whole gain over Source-only (Table 16). A^{2}IQL outperforms naive data combination and off-dynamics baselines on average on the warehouse (30:1), the synthetic benchmark and Minari, and ties the best filtering baseline on D4RL. The asymmetric decomposition underlying AIQL, a per-candidate actor extracted against a critic over an order-aware aggregate of the ranking, makes offline learning over large, variable-cardinality ranking action spaces tractable, and IQL is one instantiation of it. Natural extensions include replacing the BLR predictor with a Variational Bayesian Last Layer (VBLL) [40] to learn features end-to-end with closed-form uncertainty, distributionally robust criteria for a shift of the target state distribution, and learned set encoders [16, 35].

Limitations. The aggregate z is manually engineered, may be specific to consolidation and discards within-tier information, and its scope is the condition of Proposition 4.3. The warehouse results rest on a proprietary simulator with $1{,}230$ source and 41 target transitions, and the synthetic benchmark is the reproducible substitute. A non-imitation baseline, a learned aggregate and more data on the warehouse, and a stronger behavior policy and a variant that stresses within-tier resolution on the synthetic benchmark remain future work. The default synthetic setting places little demand on this resolution, which η controls. The Bayesian reward predictor uses fixed features. On MuJoCo the tuning budget favored A^{2}IQL. On the warehouse, the reward model fitted to the 41 target transitions is dominated by its prior, so its bonus selects source days by their alignment with the target inputs (Appendix E.5).

Code. The synthetic benchmark and the code of AIQL, A^{2}IQL and the baselines on it will be released with the arXiv version of this paper, and Appendix F specifies the benchmark in full. The warehouse simulator and data are proprietary.

Acknowledgements and Disclosures

Funding. This work was funded by Amazon.

Competing interests. G. Liu, A. Angiuli, M. Ristivojevic, J. W. Durham and M. Caldara are employees of Amazon. M. M. Zavlanos holds concurrent appointments as an Amazon Scholar and as a Professor of Mechanical Engineering and Materials Science at Duke University, and this paper describes work performed at Amazon.

References

[1] Amazon. *Amazon announces new fulfillment center robots, Sequoia and Digit*. `https://www.aboutamazon.com/news/operations/amazon-introduces-new-robotics-solutions`. Accessed: 2026-05-04. Oct. 2023.

[2] I. Kostrikov, A. Nair, and S. Levine. "Offline reinforcement learning with implicit q-learning". In: *arXiv preprint arXiv:2110.06169* (2021).

[3] S. Levine, A. Kumar, G. Tucker, and J. Fu. "Offline reinforcement learning: Tutorial, review, and perspectives on open problems". In: *arXiv preprint arXiv:2005.01643* (2020).

[4] R. F. Prudencio, M. R. Maximo, and E. L. Colombini. "A survey on offline reinforcement learning: Taxonomy, review, and open problems". In: *IEEE transactions on neural networks and learning systems* 35.8 (2023), pp. 10237–10257.

[5] A. Kumar, J. Fu, M. Soh, G. Tucker, and S. Levine. "Stabilizing off-policy q-learning via bootstrapping error reduction". In: *Advances in neural information processing systems* 32 (2019).

[6] Y. Wu, G. Tucker, and O. Nachum. "Behavior regularized offline reinforcement learning". In: *arXiv preprint arXiv:1911.11361* (2019).

[7] A. Kumar, A. Zhou, G. Tucker, and S. Levine. "Conservative q-learning for offline reinforcement learning". In: *Advances in neural information processing systems* 33 (2020), pp. 1179–1191.

[8] S. Fujimoto and S. S. Gu. "A minimalist approach to offline reinforcement learning". In: *Advances in neural information processing systems* 34 (2021), pp. 20132–20145.

[9] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch. "Decision transformer: Reinforcement learning via sequence modeling". In: *Advances in neural information processing systems* 34 (2021), pp. 15084–15097.

[10] X. B. Peng, A. Kumar, G. Zhang, and S. Levine. "Advantage-weighted regression: Simple and scalable off-policy reinforcement learning". In: *arXiv preprint arXiv:1910.00177* (2019).

[11] G. Dulac-Arnold, R. Evans, H. van Hasselt, P. Sunehag, T. Lillicrap, J. Hunt, T. Mann, T. Weber, T. Degris, and B. Coppin. "Deep reinforcement learning in large discrete action spaces". In: *arXiv preprint arXiv:1512.07679* (2015).

[12] Y. Chandak, G. Theocharous, J. Kostas, S. Jordan, and P. Thomas. "Learning action representations for reinforcement learning". In: *International conference on machine learning*. PMLR. 2019, pp. 941–950.

[13] S. Tang, M. Makar, M. Sjoding, F. Doshi-Velez, and J. Wiens. "Leveraging factored action spaces for efficient offline reinforcement learning in healthcare". In: *Advances in neural information processing systems* 35 (2022), pp. 34272–34286.

[14] Y. Bengio, A. Lodi, and A. Prouvost. "Machine learning for combinatorial optimization: a methodological tour d'horizon". In: *European Journal of Operational Research* 290.2 (2021), pp. 405–421.

[15] I. Bello, H. Pham, Q. V. Le, M. Norouzi, and S. Bengio. "Neural combinatorial optimization with reinforcement learning". In: *arXiv preprint arXiv:1611.09940* (2016).

[16] J. Lee, Y. Lee, J. Kim, A. Kosiorek, S. Choi, and Y. W. Teh. "Set transformer: A framework for attention-based permutation-invariant neural networks". In: *International conference on machine learning*. PMLR. 2019, pp. 3744–3753.

[17] M. Landers, T. W. Killian, H. Barnes, T. Hartvigsen, and A. Doryab. "BraVE: Offline Reinforcement Learning for Discrete Combinatorial Action Spaces". In: *The Thirty-ninth Annual Conference on Neural Information Processing Systems*. 2026. URL: `https://openreview.net/forum?id=Oj5tVkjbHD`.

[18] W. Zhao, J. P. Queralta, and T. Westerlund. "Sim-to-real transfer in deep reinforcement learning for robotics: a survey". In: *2020 IEEE symposium series on computational intelligence (SSCI)*. IEEE. 2020, pp. 737–744.

[19] B. Eysenbach, S. Asawa, S. Chaudhari, S. Levine, and R. Salakhutdinov. "Off-dynamics reinforcement learning: Training for transfer with domain classifiers". In: *arXiv preprint arXiv:2006.13916* (2020).

[20] J. Liu, Z. Hongyin, and D. Wang. "DARA: Dynamics-Aware Reward Augmentation in Offline Reinforcement Learning". In: *International Conference on Learning Representations*.

[21] J. Lyu, M. Yan, Z. Qiao, R. Liu, X. Ma, D. Ye, J.-W. Yang, Z. Lu, and X. Li. "Cross-domain offline policy adaptation with optimal transport and dataset constraint". In: *The Thirteenth International Conference on Learning Representations*. 2025.

[22] X. Wen, C. Bai, K. Xu, X. Yu, Y. Zhang, X. Li, and Z. Wang. "Contrastive representation for data filtering in cross-domain offline reinforcement learning". In: *Proceedings of the 41st International Conference on Machine Learning*. 2024, pp. 52720–52743.

[23] J. Lyu, K. Xu, J. Xu, M. Yan, J. Yang, Z. Zhang, C. Bai, Z. Lu, and X. Li. "Odrl: A benchmark for off-dynamics reinforcement learning". In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 59859–59911.

[24] H. Niu, Y. Qiu, M. Li, G. Zhou, J. Hu, X. Zhan, et al. "When to trust your simulator: Dynamics-aware hybrid offline-and-online reinforcement learning". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 36599–36612.

[25] J. Liu, Z. Zhang, Z. Wei, Z. Zhuang, Y. Kang, S. Gai, and D. Wang. "Beyond ood state actions: Supported cross-domain offline reinforcement learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 12. 2024, pp. 13945–13953.

[26] Z. Qiao, R. Yang, J. Lyu, C. Bai, X. Li, Z. Yang, S. Gao, and S. Qiu. "Cross-Domain Offline Policy Adaptation with Dynamics-and Value-Aligned Data Filtering". In: *arXiv preprint arXiv:2512.02435* (2025).

[27] L. Li, T. J. Walsh, and M. L. Littman. "Towards a unified theory of state abstraction for MDPs". In: *International Symposium on Artificial Intelligence and Mathematics (ISAIM)*. 2006.

[28] B. Ravindran and A. G. Barto. "SMDP homomorphisms: An algebraic approach to abstraction in semi-Markov decision processes". In: *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*. 2003.

[29] N. Ferns, P. Panangaden, and D. Precup. "Metrics for finite Markov decision processes". In: *arXiv preprint arXiv:1207.4114* (2012).

[30] G. G. Chowdhury. *Introduction to modern information retrieval*. Facet publishing, 2010.

[31] K. Pearson. "VII. Note on regression and inheritance in the case of two parents". In: *proceedings of the royal society of London* 58.347-352 (1895), pp. 240–242.

[32] C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.

[33] R. T. Rockafellar and S. Uryasev. "Conditional value-at-risk for general loss distributions". In: *Journal of Banking and Finance* 26.7 (2002), pp. 1443–1471.

[34] D. P. Kingma and J. Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[35] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. Smola. "Deep sets". In: *Advances in neural information processing systems* 30 (2017).

[36] E. Todorov, T. Erez, and Y. Tassa. "MuJoCo: A physics engine for model-based control". In: *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE. 2012, pp. 5026–5033.

[37] M. Towers, A. Kwiatkowski, J. Balis, G. De Cola, T. Deleu, M. Goulão, K. Andreas, M. Krimmel, A. Kg, R. Perez-Vicente, et al. "Gymnasium: A standard interface for reinforcement learning environments". In: *Advances in Neural Information Processing Systems* 38 (2026).

[38] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine. "D4RL: Datasets for deep data-driven reinforcement learning". In: *arXiv preprint arXiv:2004.07219* (2020).

[39] O. G. Younis, R. Perez-Vicente, J. U. Balis, W. Dudley, A. Davey, and J. K. Terry. *Minari*. Zenodo. Version 0.5.0. 2024. DOI: 10.5281/zenodo.13767625.

[40] J. Harrison, J. Willes, and J. Snoek. "Variational Bayesian last layers". In: *arXiv preprint arXiv:2404.11599* (2024).

[41] M. Kearns and S. Singh. "Near-optimal reinforcement learning in polynomial time". In: *Machine learning* 49.2 (2002), pp. 209–232.

[42] D. Brandfonbrener, W. Whitney, R. Ranganath, and J. Bruna. "Offline RL without off-policy evaluation". In: *Advances in neural information processing systems* 34 (2021), pp. 4933–4946.

A Empirical Evaluation of Sufficient Action Statistics

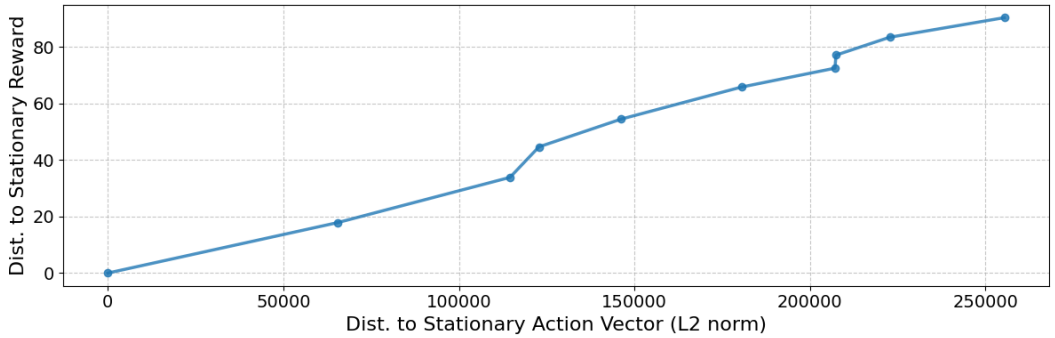

Figure 3: Sensitivity of average reward to deviations in the aggregate action statistic z. The reference point (distance 0) corresponds to the stationary action.

Table 3: Relative differences under score permutations preserving (F, z) on the warehouse simulator, mean and standard deviation over 4,950 pairwise comparisons. Lower values mean that z is closer to sufficient.

Metric	Mean Relative Diff.	Std. Dev.
Next-State ($\delta_{F'}$)	0.0006	0.0004
Reward (δ_r)	0.0125	0.0222

To empirically validate Assumption 4.1, we test whether outcomes depend on the container-level score vector $\mathbf{y}$ beyond its aggregate statistic $z = \zeta(\mathbf{y})$. For fixed (F, z) pairs, we generate multiple score vectors $\mathbf{y}$ that induce the same z by permuting $\mathbf{y}$ while preserving tier-level statistics. For any two permutations $(\mathbf{y}_i, \mathbf{y}_j)$ sharing the same (F, z), we measure relative differences in next-state representations and rewards:

$$\delta_{F'} = \frac{\|F'_i - F'_j\|_2}{\frac{1}{2}(\|F'_i\|_2 + \|F'_j\|_2)}, \tag{10}$$

$$\delta_r = \frac{|r_i - r_j|}{\frac{1}{2}(|r_i| + |r_j|)}. \tag{11}$$

Across 4,950 pairwise comparisons, both $\delta_{F'}$ and δ_r remain negligible (Table 3), with next-state variation on the order of 10^{-3} and reward differences around 1% on average. These magnitudes are small relative to the 19% throughput gain of AIQL over the behavior policy (Table 1). For this environment, the results indicate that conditional on (F, z), residual variation due to permutations of $\mathbf{y}$ is negligible, providing empirical support that z is sufficient for outcome prediction.

This empirical sufficiency is enabled by structural properties of the consolidation problem, such as long candidate lists and aggregate-driven floor evolution.

To further confirm that z is informative rather than degenerate, we examine whether perturbations in z induce systematic reward changes. For controlled deviations from a stationary reference action, we compute the ℓ_2 distance between aggregate statistics and the corresponding change in average reward. As shown in Figure 3, reward differences increase monotonically with distance in z, demonstrating that z parametrizes a meaningful direction of behavioral variation. Together, permutation invariance (within fixed (F, z)) and monotonic sensitivity (across varying z) provide empirical evidence that, for this environment, z captures the outcome-relevant information of $\mathbf{y}$.

This test uses the warehouse simulator, where outcomes depend mainly on aggregate inventory statistics. The synthetic benchmark covers the other case, a reward that depends with weight η on a per-item attribute that z omits (Table 12). Proposition 4.3 bounds the loss when the variation within an aggregate is not zero.

B Proofs

Proof of Proposition 4.2. By the Bellman optimality equation,

$$Q^*(F, \mathbf{y}) = \mathbb{E}\left[r + \gamma \max_{\mathbf{y}'} Q^*(F', \mathbf{y}') \mid F, \mathbf{y} \right]. \tag{12}$$

Under Assumption 4.1, the conditional distribution of (r, F') given $(F, \mathbf{y})$ equals that given (F, z). Therefore the right-hand side depends on $\mathbf{y}$ only through $z = \zeta(\mathbf{y})$. Hence $Q^*(F, \mathbf{y})$ is constant on equivalence classes $\{\mathbf{y} : \zeta(\mathbf{y}) = z\}$, and may be written as $Q^*(F, z)$.

Proof of Proposition 4.3. Write $r(F, \mathbf{y}) = \mathbb{E}[r \mid F, \mathbf{y}]$ and $P(\cdot \mid F, \mathbf{y})$ for the next-state distribution. Let $\omega(\cdot \mid F, z)$ be the fixed mixture over $\{\mathbf{y} : \zeta(\mathbf{y}) = z\}$ that defines the aggregated model, with $\hat{r}(F, z) = \mathbb{E}_{\mathbf{y} \sim \omega(\cdot \mid F, z)}[r(F, \mathbf{y})]$ and $\hat{P}(\cdot \mid F, z) = \mathbb{E}_{\mathbf{y} \sim \omega(\cdot \mid F, z)}[P(\cdot \mid F, \mathbf{y})]$. The absolute difference and the total variation $\mathrm{TV}(P, P') = \sup_B |P(B) - P'(B)|$ are convex, so every $\mathbf{y}$ with $\zeta(\mathbf{y}) = z$ satisfies $|r(F, \mathbf{y}) - \hat{r}(F, z)| \le \varepsilon_r$ and $\mathrm{TV}\big(P(\cdot \mid F, \mathbf{y}), \hat{P}(\cdot \mid F, z)\big) \le \varepsilon_P$. For a ranking policy π, let $\zeta_{\#}\pi$ be the aggregated policy that draws $\mathbf{y} \sim \pi(\cdot \mid F)$ and plays $\zeta(\mathbf{y})$, and let $\hat{V}$ denote values in the aggregated model. Subtracting the Bellman equations of V^π and $\hat{V}^{\zeta_{\#}\pi}$ and using $|\int f\, dP - \int f\, dP'| \le \mathrm{TV}(P, P')\, \mathrm{span}(f)$ gives, for every F,

$$\left| V^\pi(F) - \hat{V}^{\zeta_{\#}\pi}(F) \right| \le \varepsilon_r + \gamma\, \varepsilon_P\, \mathrm{span}(V^\pi) + \gamma \left\| V^\pi - \hat{V}^{\zeta_{\#}\pi} \right\|_\infty.$$

Rewards lie in $[0, R_{\max}]$, so $\mathrm{span}(V^\pi) \le R_{\max}/(1-\gamma)$, and the supremum over F gives

$$\left\| V^\pi - \hat{V}^{\zeta_{\#}\pi} \right\|_\infty \le \delta, \qquad \delta = \frac{\varepsilon_r}{1-\gamma} + \frac{\gamma\, \varepsilon_P R_{\max}}{(1-\gamma)^2}.$$

For π with $\zeta_{\#}\pi = \hat{\pi}$ and every ranking policy π', $V^{\pi'}(F) \le \hat{V}^{\zeta_{\#}\pi'}(F) + \delta \le \hat{V}^{\hat{\pi}}(F) + \delta \le V^\pi(F) + 2\delta$, where the middle inequality uses the optimality of $\hat{\pi}$ in the aggregated model. Taking the supremum over π', with $V^* = \sup_{\pi'} V^{\pi'}$, proves the claimed bound. Under Assumption 4.1 both ε_r and ε_P are zero. The argument is the simulation lemma [41] applied to an approximate aggregation of actions [27, 28, 29]. With $\varepsilon_P = 0$ the bound reduces to $2\varepsilon_r/(1-\gamma)$. □

C Algorithms

Algorithm 1 summarizes the training procedure for AIQL. Learning is performed entirely offline from a fixed dataset of historical consolidation trajectories. At each iteration, the critic updates operate on a fixed-dimensional aggregate action representation z, enabling stable value estimation without enumerating or optimizing over the underlying combinatorial container-level action space. The state-value function is learned via expectile regression, which biases value estimates toward the upper tail of observed returns, while the action-value function is trained using standard Bellman regression with no explicit maximization. Actor updates are decoupled from critic learning and performed in score space: the policy predicts container-level scores that are trained to match the relative ordering of historical scores using a weighted, centered cosine distance. Advantage-based weights emphasize high-value historical decisions while remaining strictly within the support of the logged data. This asymmetric design allows AIQL to extract and recombine high-quality sub-trajectories from offline logs, yielding strong performance improvements without any simulator interaction or online exploration.

Algorithm 1: Asymmetric Implicit Q-Learning (AIQL) for warehouse consolidation

Input: Offline dataset $\mathcal{D} = \{(F, z, \mathbf{x}, \mathbf{y}, r, F')\}$, discount γ, expectile τ, temperature T
Result: Trained scoring policy π_ϕ
Initialize V_ψ, Q_θ, Actor π_ϕ;
Initialize target networks $\bar{Q}_\theta \leftarrow Q_\theta$, $\bar{V}_\psi \leftarrow V_\psi$;
for *each gradient step* **do**

1. Sample minibatch: $(F, z, \mathbf{x}, \mathbf{y}, r, F') \sim \mathcal{D}$;
2. Critic update (aggregate action z):;
Compute TD target: $y_Q = r + \gamma \bar{V}_\psi(F')$;
Update Q_θ by MSE: $\mathcal{L}_Q(\theta) = \big(y_Q - Q_\theta(F, z)\big)^2$;
Update V_ψ by expectile regression: $\mathcal{L}_V(\psi) = L_2^\tau\big(\bar{Q}_\theta(F, z) - V_\psi(F)\big)$;
3. Advantage weights:;
$A(F, z) = \bar{Q}_\theta(F, z) - \bar{V}_\psi(F), \quad w = \exp\big(A(F, z)/T\big)$;
4. Actor update (container-level scores):;
Predict scores: $\hat{\mathbf{y}} = \pi_\phi(F, \mathbf{x})$;
Center scores: $\tilde{\mathbf{y}} = \mathbf{y} - \mathrm{mean}(\mathbf{y}),\ \tilde{\hat{\mathbf{y}}} = \hat{\mathbf{y}} - \mathrm{mean}(\hat{\mathbf{y}})$;
Update π_ϕ minimizing weighted centered cosine distance: $\mathcal{L}_\pi(\phi) = w\Big(1 - \frac{\langle \tilde{\hat{\mathbf{y}}}, \tilde{\mathbf{y}} \rangle}{\|\tilde{\hat{\mathbf{y}}}\|_2 \|\tilde{\mathbf{y}}\|_2}\Big)$;
5. Target update:;
$\bar{Q}_\theta \leftarrow (1 - \rho)\bar{Q}_\theta + \rho Q_\theta,\ \bar{V}_\psi \leftarrow (1 - \rho)\bar{V}_\psi + \rho V_\psi$;
end

Algorithm 2 extends AIQL to the cross-domain setting in two phases. The source critic is trained once on $\mathcal{D}^{\mathtt{sim}}$ with the critic updates of Algorithm 1, without an actor, and remains frozen. A Bayesian reward predictor is fit on the scarce target-domain data, and its conservative lower confidence bound modifies only the AWR weights during the extraction of a fresh actor. Phase 2 is a one-step policy extraction against a fixed critic, as in [42]. Their fixed critic is the behavior Q-function, and phase 2 uses the IQL expectile critic.

Algorithm 2: Adaptive Asymmetric IQL (A^2IQL) for cross-domain consolidation

Input: Source dataset $\mathcal{D}^{\mathtt{sim}}$, target dataset $\mathcal{D}^{\mathtt{tgt}}$, discount γ, expectile τ, adaptation strength λ, conservatism κ, threshold percentile p, temperature T
Result: Adapted scoring policy π_ϕ
Phase 1: source critic.;
Initialize Q_θ, V_ψ and their target networks;
for *each critic gradient step* **do**
  Sample $(F, z, r, F') \sim \mathcal{D}^{\mathtt{sim}}$ and apply steps 2 and 5 of Algorithm 1;
end
Freeze Q_θ, V_ψ and their target networks;
Phase 2: adaptive actor extraction.;
Fit BLR on $\{(F_j, z_j, r_j^{\mathtt{tgt}})\}_{j=1}^{N_{\mathtt{tgt}}}$ to obtain $\mu_\eta(\cdot)$, $\sigma_\eta(\cdot)$;
for *each* $(F_i, z_i) \in \mathcal{D}^{sim}$ **do**
  $b_\eta^{\mathtt{safe}}(F_i, z_i) = \mu_\eta(F_i, z_i) - \kappa\, \sigma_\eta(F_i, z_i)$;
end
$\tau_b \leftarrow$ the p-th percentile of $\{b_\eta^{\mathtt{safe}}(F_i, z_i)\}_{i=1}^{N_{\mathtt{sim}}}$;
$b_i \leftarrow \max\big(b_\eta^{\mathtt{safe}}(F_i, z_i) - \tau_b,\ 0\big)/s_b$ for every i, with s_b from Appendix D.2;
Initialize a fresh Actor π_ϕ;
for *each actor gradient step* **do**
  Sample minibatch: $(F, z, \mathbf{x}, \mathbf{y}) \sim \mathcal{D}^{\mathtt{sim}}$ with their bonuses b;
  Source advantage from the frozen target networks: $A^{\mathtt{sim}} = \bar{Q}_\theta(F, z) - \bar{V}_\psi(F)$ (other testbeds in Appendix D.3);
  Adapted advantage: $\tilde{A} = A^{\mathtt{sim}} + \lambda\, b$;
  Adapted weight: $w^{\mathtt{adapt}} = \exp\big(\tilde{A}/T\big)$;
  Update π_ϕ by minimizing the $w^{\mathtt{adapt}}$-weighted actor loss of step 4 of Algorithm 1;
end

D Implementation Details

This section states what the code computes on the synthetic and MuJoCo testbeds. All A^2IQL results use the two-phase schedule of Algorithm 2, with $\kappa = 1$ except in the κ sweep of Table 16, and prior precision $\alpha_0 = 1$.

D.1 Reward Model

Synthetic. The features are $\mathbf{h} = [1, s, z]$ with raw inputs. The target rewards are standardized with their mean and standard deviation. The noise precision is $\nu = 1/\max(10^{-3}, v_{\mathrm{res}})$, where v_{res} is the residual variance of an ordinary least-squares fit to the standardized targets, and μ_η and σ_η are mapped back to reward units. **MuJoCo.** The same estimator with $\mathbf{h} = [1, \tilde{s}, a]$, where $\tilde{s}$ is the observation standardized with the dataset statistics.

D.2 Bonus Threshold and Scale

The threshold τ_b is a percentile of $b_\eta^{\mathtt{safe}}$ over all source transitions. The synthetic benchmark uses the 75th percentile by default, and Table 16 varies it. On MuJoCo the threshold is the 50th percentile for the D4RL reward shift, selected on tuning seeds 0 to 9, and the default 75th percentile on Minari and in the dynamics-shift runs of Table 19. The single-setting check of Appendix G.4 uses the mean of $b_\eta^{\mathtt{safe}}$. On the synthetic and MuJoCo testbeds, s_b is the population standard deviation of the rectified bonus over the source transitions, zeros included, when this standard deviation exceeds 10^{-8}, and $s_b = 1$ otherwise. The synthetic code adds 10^{-8} to this standard deviation.

D.3 Advantage, Weights and Schedule

Table 4 gives the advantage and the actor weight as computed, where $\overline{(\cdot)}_{\mathcal{B}}$ is the minibatch mean. The TD target of $\mathcal{L}_Q$ and the advantages in Sections 4 and 5 are written with the online networks for

readability. Algorithms 1 and 2 use the Polyak target networks $\bar{Q}_\theta$ and $\bar{V}_\psi$ in the TD target and in the advantage on the warehouse. On the synthetic and MuJoCo testbeds only Q has a target network, and the TD target uses V_ψ. On the synthetic benchmark every method trains its critic first and its actor second, so A^{2}IQL with $\lambda = 0$ equals Source-only. On the warehouse and MuJoCo, Source-only trains its actor during critic training against the evolving critic, so there A^{2}IQL with $\lambda = 0$ is a different procedure (Table 18 reports it for MuJoCo).

Table 4: Advantage and actor weight per testbed, with $\tilde{A} = A^{\texttt{sim}} + \lambda b$ for A^{2}IQL and $\tilde{A} = A$ for AIQL and the baselines.

Testbed	Advantage	Weight
Synthetic	$Q_\theta - V_\psi$	$\hat{A} = \tilde{A}$ standardized over the data, $\hat{w} = \min(e^{3\hat{A}}, 100)$, $w = \hat{w}/\overline{\hat{w}}_{\mathcal{B}}$
MuJoCo	$\min(\bar{Q}_1, \bar{Q}_2) - V_\psi$	$w = \min(e^{3\tilde{A}}, 100)$

D.4 Hyperparameters

Table 5: Network architectures and training hyperparameters of the synthetic and MuJoCo experiments.

Component	**Synthetic**	**MuJoCo**
Inputs	$s \in \mathbb{R}^3$, $x \in \mathbb{R}^5$, $z \in \mathbb{R}^{20}$	$s \in \mathbb{R}^{17}$, $a \in \mathbb{R}^6$
Q-network	$[23, 128, 128, 1]$	twin, $[23, 256, 256, 1]$
V-network	$[3, 128, 128, 1]$	$[17, 256, 256, 1]$
Actor	per item, $[8, 64, 64, 1]$	Gaussian, $[17, 256, 256, 6]$
Optimizer, learning rate	Adam, 3×10^{-4}	Adam, 3×10^{-4}
Batch size	256 (critic), 512 items (actor)	256
Training length	4,000 critic, 2,500 actor steps (6,000, 4,000 in Tables 10 and 17)	300k steps
Target update (ρ)	0.005 on Q	0.005 on Q
Discount, expectile	0.99, 0.7	0.99, 0.7
Actor loss	per-item cross-entropy on the selection	Gaussian log-likelihood
λ, τ_b percentile	3, 75	D4RL reward shift 16, 50; Minari and dynamics shift 4, 75
Evaluation	80 episodes (sweeps 60, dynamics 50)	50 episodes

Spreads in Tables 2, 10 and 18 and in Appendices F.3 and G are population standard deviations over seeds, except Table 17, which uses the sample standard deviation. Paired standard errors are the sample standard deviation of the per-seed differences divided by $\sqrt{n}$. Synthetic and MuJoCo results use the final policy, and Appendix E.3 gives the warehouse selection protocol.

E Additional Warehouse Results

E.1 Robustness on Unseen Datasets

A key concern for offline RL is whether the learned policy generalizes beyond the specific data it was trained on. To test this, we train AIQL on one dataset and evaluate the resulting policy on three other datasets that differ in their initial item distribution across containers. As shown in Table 6, AIQL maintains throughput within $\pm 0.25\%$ of its value on the training dataset. This stability indicates that the learned ranking strategy captures general consolidation structure rather than memorizing properties of a particular initial state.

E.2 Tier Count and State Features

Every tier count in Table 7 stays above the behavior policy. At $C = 1$ the aggregate holds only the mean and variance over all candidates, which do not change with the ordering, so the value cannot separate a good ranking from a bad one. Many thin tiers over-fragment the summary and add estimation noise to the value on a fixed dataset, and $C = 3$ to 4 resolves the ranking without this

Table 6: AIQL robustness on unseen warehouse datasets. The entries are the relative change of the 40-epoch throughput of AIQL versus its training dataset, where values near 0% mean stable transfer.

Dataset	Training	Unseen 1	Unseen 2	Unseen 3
Relative Throughput Change (vs. training)	0%	-0.13%	+0.24%	+0.01%

Table 7: Warehouse single-domain ablations, 40-epoch throughput (↑), mean ± std over 10 training seeds. (a) Number of tiers C in the aggregate z. (b) Six or twelve global state features at $C = 4$. The behavior policy that generated the logs reaches 68.38.

(a)

C	1	2	3	4	6	8
Throughput	77.40 ± 0.89	77.68 ± 0.89	80.92 ± 1.11	81.59 ± 0.69	78.87 ± 0.58	78.65 ± 0.62

(b)

State	6-dim (default)	12-dim (higher-order features added)
Throughput	81.59 ± 0.69	80.92 ± 2.40

fragmentation. The peak is more pronounced than on the synthetic benchmark (Table 13), which is consistent with a warehouse reward that depends on finer ranking structure. The 12-dimensional state adds the skewness and the 25th and 75th percentiles of the per-container item count and volume, and it gives no improvement. For $C \neq 4$ each ranked list is split into C contiguous tiers of near-equal size, while $C = 4$ keeps the rank ranges of the default z.

E.3 Cross-Domain Protocol and Per-Seed Results

Each checkpoint is scored by one deterministic 40-epoch rollout. Under the target reward, the checkpoint with the highest target reward within a fixed window of training epochs is selected for each method and seed, among the rollouts that complete 40 epochs. Under the constraint-penalized objective, the checkpoint with the highest penalized objective over the first $H = 23$ epochs is selected per seed from a fixed set of checkpoints. $H = 23$ is the shortest rollout over all runs. All methods are evaluated at the same training epochs. Constraint-satisfied days average the completed days over all evaluated checkpoints. Seeds 1 to 3 have six evaluated checkpoints and seeds 4 and 5 have three, so the days of Table 2 pool these 24 checkpoints and differ from the mean of the per-seed values of Table 8(b). Table 8 gives the per-seed values of both experiments. Under the target reward A^{2}IQL is the top method on 4 of 5 seeds. Under the penalized objective every baseline keeps the constraints for 39 to 40 days on seeds 1 to 3. A^{2}IQL is one day behind them on seed 2, and its lead comes from seeds 4 and 5, where it stays satisfied longer than every baseline.

E.4 Source Advantage under Growing Reward Mismatch

Table 9 answers how the source advantage relates to the target outcome as the reward mismatch grows on the warehouse. Throughput dominates r_β on the logged days, so the source advantage keeps ranking the days by the realized target reward at every β. Its rank correlation with the state-adjusted residual e_β is weak at every β and grows in magnitude with β. The Source-only weights therefore raise the weighted mean of r_β at every β. The A^{2}IQL weights lower it, because their bonus has a rank correlation between -0.753 and -0.736 with r_β. The synthetic reward-shift sweep of Table 14 covers a larger mismatch, under which the target return of Source-only falls by 112.7 from $\xi = 0$ to $\xi = 1$.

E.5 Reward-Model and Weight Diagnostics

The warehouse reward model has no intercept and regresses the raw target rewards. Its posterior is set by the prior in all but one direction, since the eigenvalues of $\mathbf{H}^\top \mathbf{H}$ divided by the noise variance have a maximum of 3.05 and a median of 2.14×10^{-6}, against a prior precision of 1. Its predicted mean on the source days therefore ranks like the inner product with the mean target input (rank correlation 0.9995), so the bonus selects source days by their alignment with the target inputs. In actor weights recomputed with a phase-1 critic, the 5.6% to 7.4% of the transitions in a minibatch that reach the

Table 8: Warehouse cross-domain results per seed, 5 seeds. (a) Target reward (↑) of the selected checkpoint, with – where no rollout in the selection window completes 40 epochs. The last column gives the mean paired difference A^{2}IQL minus method over the seeds where both have a value, with the number of seeds on which A^{2}IQL is ahead. (b) Constraint-satisfied days (↑, of 40) averaged over the evaluated checkpoints of each seed, with the mean paired difference A^{2}IQL minus method and the number of seeds on which A^{2}IQL is ahead and tied.

Seed	1	2	3	4	5	Paired comparison
(a) Target reward						A^{2}IQL minus (ahead)
A^{2}IQL	129.55	89.97	100.01	131.68	124.00	–
Source-Only	119.88	92.51	96.57	104.83	90.87	+14.11 (4/5)
Target-Only	93.16	157.14	28.94	126.16	–	+11.45 (3/4)
Pooled	122.67	94.24	99.21	114.85	95.61	+9.72 (4/5)
DARA-style	122.19	95.50	99.72	114.85	95.79	+9.43 (4/5)
OTDF-style	90.98	158.98	41.09	115.44	–	+11.18 (3/4)
IGDF-style	96.71	160.79	38.06	–	–	+7.99 (2/3)
(b) Constraint-satisfied days						A^{2}IQL minus (ahead, tied)
A^{2}IQL	40.0	39.0	40.0	37.0	36.0	–
Source-Only	40.0	40.0	39.5	36.3	26.7	+1.9 (3, 1)
Target-Only	40.0	40.0	40.0	31.0	31.3	+1.9 (2, 2)
Pooled	40.0	40.0	40.0	34.7	28.0	+1.9 (2, 2)
DARA-style	40.0	40.0	40.0	34.7	25.3	+2.4 (2, 2)
OTDF-style	40.0	40.0	40.0	32.3	28.7	+2.2 (2, 2)
IGDF-style	40.0	40.0	40.0	28.0	31.7	+2.5 (2, 2)

Table 9: Source advantage and realized target reward on the 1,230 logged source days. r_β is the target reward of Table 2 with the weight of the second operational metric set to β, so $\beta = 0$ is the source reward, and e_β is the residual of r_β after a linear fit on the six state features. A comes from the frozen phase-1 critics of 5 seeds, and correlations with A are mean ± std over these seeds. Δr_β and Δe_β are the weighted minus the plain mean under the actor weights of Source-only and of A^{2}IQL, with the normalization over all source days, means over 5 seeds. The plain mean of r_β is 67.83 at $\beta = 0$ and 106.88 at $\beta = 200$, and that of e_β is 0.

β	0	25	50	100	200	400	800
Rank corr. (throughput, r_β)	+1.000	+1.000	+0.999	+0.996	+0.991	+0.981	+0.967
Rank corr. (A, r_β)	$+0.885 \pm 0.026$	$+0.888 \pm 0.026$	$+0.890 \pm 0.026$	$+0.891 \pm 0.025$	$+0.891 \pm 0.025$	$+0.888 \pm 0.024$	$+0.884 \pm 0.022$
Rank corr. (A, e_β)	-0.064 ± 0.006	-0.087 ± 0.008	-0.108 ± 0.009	-0.133 ± 0.010	-0.153 ± 0.011	-0.160 ± 0.012	-0.161 ± 0.013
Δr_β, Source-only weights	+29.11	+33.42	+37.73	+46.35	+63.58	+98.05	+167.00
Δr_β, A^{2}IQL weights	−20.68	−22.58	−24.47	−28.15	−34.88	−48.58	−77.92
Δe_β, Source-only weights	+2.28	+2.95	+3.62	+4.95	+7.62	+12.95	+23.62
Δe_β, A^{2}IQL weights	−2.05	−3.00	−3.26	−3.47	−5.96	−8.93	−19.97

weight cap hold 79% to 80% of the normalized weight mass. The A^{2}IQL results of Table 2 use this reward model, and the synthetic and MuJoCo experiments test the mechanism of Section 5 with the estimator of Appendix D.1. With that estimator on inputs z-scored with the source statistics and the bonus scale of Appendix D.2, A^{2}IQL on the warehouse reaches a target reward of 117.97 ± 13.32, a penalized objective of 79.2 ± 26.3 and 37.79 constraint-satisfied days over 5 seeds with the protocol of Appendix E.3 (baselines in Table 2).

E.6 Notes on the Reported Runs

In the target-reward experiment, the baseline runs of seeds 1 to 3 were resumed from a checkpoint during training, with the target networks reset to the online networks and a fresh Adam optimizer. The phase-1 critic of A^{2}IQL matches the critic of the reported Source-Only runs exactly at every saved epoch in the penalized experiment and for seeds 4 and 5 of the target-reward experiment. For seeds 1 to 3 the two critics agree up to the resume point, and after it no weight differs by more than 1.017×10^{-1}. The container features of the two domains follow different definitions, since only the source features include a later correction, and 10 of the 54 reward-model inputs differ between the domains by more than three target standard deviations.

F Synthetic Ranking Benchmark

F.1 Specification

The state is a backlog $s_t \in [0,3]^m$ over $m = 3$ item categories, with $s_0 \sim \mathcal{U}(0.3, 1.2)^m$. At each step $N_t \sim \mathcal{U}\{52, \dots, 60\}$ candidate items arrive. Item i has a category c_i drawn uniformly from $\{1, \dots, m\}$, a size $v_i \sim \mathcal{U}(0.1, 1)$ and a detail attribute $d_i \sim \mathcal{U}(0, 1)$, and its feature vector is the one-hot category, v_i and d_i. The policy ranks the items, and the top $K = 8$ form the selected set $\mathcal{S}_t$, which gives $\binom{60}{8} \approx 2.6 \times 10^9$ possible selections at the largest candidate count. The reward and the backlog update are

$$r_t = \textstyle\sum_{i \in \mathcal{S}_t} \left[\vartheta(c_i)\left(1 + s_t(c_i)\right) v_i + \eta\,(d_i - 0.5)\right], \tag{13}$$

$$s_{t+1}(c) = \min\Big(3,\ \max\Big(0,\ s_t(c) + a_t(c) - \textstyle\sum_{i \in \mathcal{S}_t :\, c_i = c} v_i\Big)\Big), \tag{14}$$

with arrivals $a_t(c) = \max\left(0,\ \bar{a}_c\,(1 + 0.6\,(u_{t,c} - 0.5))\right)$, $u_{t,c} \sim \mathcal{U}(0,1)$, and arrival means $\bar{a}_c = 0.15 + 0.10\,\mathcal{U}(0,1)$ drawn once per environment seed, and each episode lasts 20 steps. The source values are $\boldsymbol{\vartheta}^{\mathtt{src}} = (1.5, 0.5, 0.1)$, and the target values are $\boldsymbol{\vartheta}^{\mathtt{tgt}} = (1-\xi)\boldsymbol{\vartheta}^{\mathtt{src}} + \xi\,(0.1, 0.5, 1.5)$ with shift magnitude $\xi = 1$ by default. Both domains share the dynamics, and $\eta = 0$ by default.

The aggregate $z \in \mathbb{R}^{20}$ splits the ranked list into three contiguous tiers of near-equal size. Each tier contributes the mean one-hot category, the mean size and the fraction of selected items, and a fourth block holds the mean one-hot category and the mean size of the selected items and K/N_t. The aggregate omits d_i, so η sets how much the reward depends on information outside z. The behavior policy scores item i by $(1 + s_t(c_i))\,v_i + \epsilon_i$ with $\epsilon_i \sim \mathcal{N}(0, 0.6^2)$. It ignores the category values and is the same in both domains, so it is deliberately suboptimal and cloning it stays suboptimal. The source data are 150 episodes (3,000 transitions) and the target data 2 episodes (40 transitions), a 75:1 ratio. At the default arrival rates about 90% of the category backlogs are empty in 50 behavior-policy episodes in the source dynamics, which makes the default task close to one-step. In the transition-only shift, both domains use $\boldsymbol{\vartheta}^{\mathtt{src}}$ and the target arrival means are $g\,\bar{a}_c$ with $g \in \{1, 2, 5, 10\}$.

F.2 Methods

The critic and the actor follow Tables 4 and 5, and every method trains its critic first and its actor second. The actor scores each item from s and the item features and is trained with the advantage-weighted cross-entropy of the selection indicator. A^{2}IQL uses $\lambda = 3$, $\kappa = 1$ and τ_b at the 75th percentile. BC weights all transitions equally, and %BC keeps the 25% of transitions with the highest advantage. Asym-CQL adds the penalty $\log\left(e^{Q(s,z)} + e^{Q(s,z')}\right) - Q(s,z)$ with weight 0.5 to the IQL critic, where z' is the aggregate of another transition in the minibatch. SetEnc replaces z by a mean-pooled embedding of the item features and selection flags with 64 hidden units, trained jointly with the IQL value objective. The DARA-style baseline subtracts $1 - p_{\text{tgt}}$ from each source reward. The OTDF-style baseline keeps the 30% of source transitions closest to any target transition in standardized (s, z), and the IGDF-style baseline the 30% closest to the target centroid in a learned classifier embedding. Each of these trains AIQL on the modified source data and the target data. The oracle ranks items by their immediate value under the evaluation reward, including d_i. Training seeds set the network initialization, the generated data and the environment constants.

F.3 Additional Results

AIQL stays above %BC at every expectile in Table 11, including $\tau = 0.5$, where V_ψ regresses to the mean of Q. In Table 12 the ratio falls because the reward moves to information that z omits. In Table 13 the ratio varies by 0.030 across C. $C = 1$ is marginally best because the default reward is determined by the coarse composition of the selected set, which a single tier and the selected-set block already capture. Together with the mostly empty backlog, this makes the default setting a weak test of within-tier resolution, and η is the knob that stresses it. In Table 14 A^{2}IQL is above Target-only at every shift magnitude and has the highest mean except at $\xi = 0.25$, where the DARA-style baseline is 0.2 higher. In Table 15 Target-only reaches the level of A^{2}IQL with 20 target episodes. In Table 16 A^{2}IQL is below Target-only for $\lambda \leq 2$ and above it for $\lambda \geq 3$, it rises with κ, and the default 75th-percentile threshold is not the best value. With $\lambda = 0$, A^{2}IQL equals Source-only because both use the same critic and the same weights. In Table 17 the shift barely changes the task at $g \leq 2$, where

Table 10: Synthetic ranking benchmark, episodic return (↑), mean ± std over seeds, with the paired per-seed difference and its standard error. (a) Single-domain return over 20 training seeds. (b) Reward shift with 3,000 source and 40 target transitions, evaluated under the target reward over 15 seeds. The oracle ranks items by their immediate true value.

(a) Single domain

Method	Return	Minus AIQL (SE)
Oracle	196.85 ± 0.72	–
SetEnc	168.42 ± 10.47	$+5.21$ (1.55)
Asym-CQL	164.32 ± 9.84	$+1.11$ (0.58)
AIQL	163.21 ± 9.24	–
%BC	150.12 ± 8.22	-13.09 (1.05)
BC	110.60 ± 4.69	-52.61 (2.37)
Behavior policy	85.01 ± 0.74	–

(b) Reward shift

Method	Return	A^2IQL minus (SE)
Oracle	196.98 ± 0.61	–
A^2IQL	136.76 ± 16.73	–
Target-only	108.39 ± 17.92	$+28.37$ (6.93)
IGDF-style	79.87 ± 32.30	$+56.89$ (8.34)
OTDF-style	78.79 ± 23.72	$+57.97$ (6.48)
Pooled	60.40 ± 11.07	$+76.36$ (4.39)
Source-only	60.18 ± 7.56	$+76.58$ (4.22)
DARA-style	57.45 ± 10.07	$+79.31$ (4.56)

Table 11: Synthetic single domain, expectile sweep, episodic return (↑), mean ± std over 10 seeds. The oracle reaches 196.41 ± 0.44 at every expectile.

Expectile τ	AIQL	Asym-CQL	SetEnc	%BC	BC
0.5	169.78 ± 10.37	169.55 ± 9.03	171.94 ± 10.19	157.20 ± 6.90	110.43 ± 4.42
0.6	168.33 ± 10.02	168.68 ± 9.71	171.54 ± 10.16	157.11 ± 7.45	110.43 ± 4.42
0.7	167.36 ± 10.62	167.38 ± 9.54	170.62 ± 10.40	156.80 ± 7.15	110.43 ± 4.42
0.8	165.83 ± 10.50	166.31 ± 10.38	169.79 ± 10.62	155.30 ± 7.58	110.43 ± 4.42
0.9	163.52 ± 10.85	163.34 ± 10.26	168.53 ± 11.00	151.73 ± 6.56	110.43 ± 4.42

the backlog stays mostly empty, and the A^2IQL gain at $g = 1$ appears without any shift. At $g = 5$ and 10 A^2IQL gives no gain, while Target-only and Pooled, which train on the shifted transitions, gain up to 96.76. At $g = 10$ the myopic oracle is below Target-only and no longer an upper bound.

F.4 Schedule Check

We reran the 15-seed reward-shift configuration of Table 10(b) with the critic-then-actor schedule of Appendix F.2. It reproduces every logged value, with a maximum absolute difference of 0.00. A concurrent schedule, which trains the actor during critic training and yields identical critics, changes A^2IQL by $+1.96 \pm 2.08$ (mean ± std over 15 seeds). The other synthetic results come from the same code and were not rerun.

G MuJoCo HalfCheetah

G.1 Data and Target Reward

D4RL halfcheetah-medium-expert-v2 [38] has 2,000,000 transitions from 1,000 medium and 1,000 expert episodes, with cruising speeds of 5.30 ± 0.40 and 11.58 ± 0.49 m/s. Its target reward is $-|v - 5.35| - 0.05\|\mathbf{a}\|^2$, where 5.35 m/s is the median cruising speed of the medium half, and policies are evaluated in Gymnasium HalfCheetah-v4 [37]. The Minari [39] data combine the HalfCheetah simple, medium and expert datasets (3,000,000 transitions). There $v^\star = 7.4$ m/s, the cruising speed of the simple policy, and evaluation uses HalfCheetah-v5. On each seed, 200 transitions drawn at random from the pool and relabeled with the target reward form the target data, and all other transitions with the native reward form the source data. Rewards are scaled by 1000 over the range of episode returns, and observations are standardized with the dataset statistics. Replaying the D4RL transitions in HalfCheetah-v4 reproduces the logged next observation with a mean error of 1.2% of the data standard deviation (5.9% at the 99th percentile). Replaying the Minari transitions reproduces them to float64 round-off.

Table 12: Synthetic single domain, sufficiency sweep over the weight η of the omitted detail attribute, episodic return (↑), mean ± std over 10 seeds.

η	AIQL	Oracle	Behavior policy	AIQL / oracle
0	167.36 ± 10.62	196.41 ± 0.44	85.69 ± 0.57	0.852
1	167.11 ± 11.26	209.14 ± 0.59	85.44 ± 0.59	0.799
2	167.28 ± 11.56	246.98 ± 0.79	85.19 ± 0.75	0.677
4	166.32 ± 13.52	358.80 ± 1.15	84.69 ± 1.25	0.464

Table 13: Synthetic single domain, number of tiers C in z, episodic return of AIQL (↑), mean ± std over 10 seeds, and its ratio to the oracle at $\eta = 0$ (oracle 196.41) and $\eta = 1$ (oracle 209.14).

C	1	2	3	4	6	8
Dimension of z	10	15	20	25	35	45
AIQL, $\eta = 0$	170.85 ± 8.46	169.02 ± 9.53	167.36 ± 10.62	168.04 ± 8.92	167.72 ± 10.07	164.95 ± 8.76
AIQL / oracle, $\eta = 0$	0.870	0.861	0.852	0.856	0.854	0.840
AIQL / oracle, $\eta = 1$	0.809	0.801	0.799	0.803	0.803	0.787

G.2 Methods

The backbone is IQL with twin Q-networks, a value network and a Gaussian policy (Table 5), trained for 300k steps with inverse temperature 3 and weight cap 100. Phase 1 is IQL on the source data. Its actor is the Source-only policy, and its final critic is frozen for phase 2, which trains a fresh Gaussian actor with the weights of Table 4. The reward model is the BLR of Appendix D.1. Pooled trains IQL on the union of the source and target data. The DARA-style baseline subtracts $c\,(1 - p_{\text{tgt}})$ from the source rewards, with p_{tgt} from a domain classifier. The OTDF-style and IGDF-style baselines keep a fraction of the source transitions nearest to the target data in the standardized inputs or in a learned domain-discriminative embedding. The bonus-only ablation drops $A^{\texttt{sim}}$ from Equation (7) and keeps λb. The target oracle is a diagnostic that trains IQL on the source transitions relabeled with the true target reward. All results use the final policy, evaluated over 50 episodes.

G.3 Tuning Protocol

Each tunable method is selected by mean target return on tuning seeds and reported on held-out seeds. On D4RL the tuning seeds are 0 to 9 and the held-out seeds 10 to 29, and on Minari the tuning seeds are 0 to 2 and the test seeds 10 to 19. The IGDF-style and OTDF-style keep fractions range over $\{0.05, 0.1, 0.2, 0.3, 0.5\}$ and the DARA-style penalty c over $\{0.5, 1, 2, 4, 8\}$, with defaults 0.3 and 2. The selected values are 0.2, 0.2 and 4 on D4RL and 0.3, 0.5 and 0.5 on Minari. A^{2}IQL was selected from 20 configurations of the reward-model features, λ, κ and τ_b on D4RL and from 6 configurations of the reward-model features and λ on Minari, so the tuning budget favors A^{2}IQL. Source-only, Target-only and Pooled have no tunable parameter.

G.4 Results

On D4RL the tuned IGDF-style filter fails on a seed-dependent subset (Table 18), with a worst seed of -5055 against -1756 for A^{2}IQL. The bonus-only ablation is above A^{2}IQL on both benchmarks, and dropping the source advantage improves A^{2}IQL by 175 (SE 36) over all 30 D4RL seeds. On D4RL the Source-only policy scores 82.1 normalized points on the source reward, against 86.7 reported for IQL trained for 1M steps [2]. On Minari the Source-only policy scores only 30.4 normalized points.

One setting for both benchmarks. With $\lambda = 4$ and τ_b at the mean of $b_\eta^{\texttt{safe}}$, A^{2}IQL reaches -1335 ± 244 on D4RL and -1499 ± 141 on Minari. It ties the tuned IGDF-style filter on D4RL, $+213$ (SE 243), and is above every tuned baseline on Minari. The mean threshold was chosen after its results on D4RL seeds 0 to 9 and Minari seeds 10 to 19 were known, so only the D4RL held-out numbers evaluate this choice on unseen seeds.

Table 14: Synthetic reward-shift sweep over the shift magnitude ξ, target return (↑), mean ± std over 10 seeds, with $4{,}000$ critic and $2{,}500$ actor steps. The myopic oracle reaches 196.4, 152.6, 121.0, 153.0 and 197.1 at these shifts.

ξ	A^{2}IQL	Source-only	Target-only	Pooled	DARA-style	OTDF-style	IGDF-style
0	172.1 ± 5.0	167.4 ± 10.6	129.7 ± 15.0	165.5 ± 7.1	168.6 ± 7.2	153.3 ± 15.7	150.7 ± 13.3
0.25	139.5 ± 2.5	139.2 ± 6.2	117.0 ± 9.8	138.1 ± 3.7	139.7 ± 3.6	130.4 ± 8.7	129.5 ± 7.5
0.5	115.6 ± 1.6	111.0 ± 2.2	109.0 ± 5.9	110.8 ± 1.6	111.0 ± 1.5	108.8 ± 3.9	107.7 ± 4.6
0.75	112.7 ± 4.9	82.9 ± 3.6	106.0 ± 10.4	83.4 ± 4.4	82.2 ± 5.2	86.7 ± 8.3	86.5 ± 6.6
1	138.0 ± 12.9	54.7 ± 7.9	105.5 ± 22.2	56.6 ± 7.1	53.5 ± 9.5	65.4 ± 13.1	66.3 ± 11.7

Table 15: Synthetic reward shift, number of target episodes (20 transitions each), target return (↑), mean ± std over 10 seeds. Source-only reaches 54.69 ± 7.86 and the oracle 197.05 ± 0.37.

Target episodes	A^{2}IQL	Target-only	Pooled	DARA-style	OTDF-style	IGDF-style
1	134.12 ± 22.62	109.11 ± 16.38	57.61 ± 6.00	54.96 ± 6.20	67.61 ± 23.21	74.24 ± 20.01
2	138.00 ± 12.91	105.53 ± 22.24	56.57 ± 7.05	53.46 ± 9.47	65.37 ± 13.11	66.28 ± 11.69
5	142.27 ± 11.05	122.51 ± 24.60	57.80 ± 11.67	54.66 ± 10.70	68.74 ± 18.53	76.04 ± 13.60
20	142.90 ± 11.28	142.94 ± 21.16	60.78 ± 6.39	59.32 ± 5.78	98.46 ± 12.69	101.17 ± 14.63

G.5 Dynamics Shift

The source data are all D4RL transitions with the native reward, and the target environment scales gravity by 1.25 or 1.5 under the same reward. The $2{,}000$ target transitions per seed come from the 1M-step Source-only policy with Gaussian action noise of standard deviation 0.3. The DARA-style, OTDF-style and IGDF-style baselines act on (s, a, s'). Neither A^{2}IQL row of Table 19 differs from Source-only by two standard errors, and the bonus has no measurable effect, with A^{2}IQL minus $\lambda = 0$ at $+108$ (SE 166) and -57 (SE 516). This is consistent with one check at gravity $\times 1.5$ on seed 3, in which the reward model fitted to the target rewards correlates 0.996 with the source reward. At $\times 1.5$, Pooled and the IGDF-style and OTDF-style baselines exceed Source-only by $1{,}223$ to $1{,}660$, and Target-only and the DARA-style baseline fall below it. The collector reaches 3317.8 ± 219.0 in the $\times 1.5$ environment, so part of this gain may come from the collector's actions.

Table 16: Synthetic reward shift, A^{2}IQL hyperparameters with the others at their defaults ($\lambda = 3$, $\kappa = 1$, 75th percentile), target return (↑), mean ± std over 10 seeds. Target-only reaches 105.53 ± 22.24 and Source-only 54.69 ± 7.86 in this setting.

λ	0	1	2	3	5	10
A^{2}IQL	54.69 ± 7.86	71.11 ± 5.16	97.69 ± 11.79	138.00 ± 12.91	161.39 ± 8.47	168.89 ± 4.63
κ	0	0.5	1	2	4	
A^{2}IQL	135.78 ± 13.05	137.01 ± 13.02	138.00 ± 12.91	139.98 ± 12.73	143.58 ± 12.17	
τ_b percentile	0	50	75	90		
A^{2}IQL	147.42 ± 7.44	149.59 ± 9.12	138.00 ± 12.91	109.28 ± 14.88		

Table 17: Synthetic transition-only shift with target arrival means scaled by g and the source reward in both domains. Return in the target dynamics (↑), mean ± sample std over 10 seeds. Below each method, its paired difference to Source-only with SE and the seeds on which the method is ahead. The last rows give the myopic oracle and the share of category backlogs that are empty in 50 behavior-policy episodes in the target dynamics.

g	1	2	5	10
Source-only	163.12 ± 11.12	160.83 ± 12.48	161.36 ± 14.04	219.25 ± 23.09
A^{2}IQL	170.01 ± 5.65	165.84 ± 5.81	158.47 ± 5.62	204.63 ± 19.14
minus Source-only	$+6.89$ (2.36), 9/10	$+5.01$ (2.63), 8/10	-2.89 (3.47), 3/10	-14.62 (6.64), 2/10
Target-only	122.68 ± 17.70	128.01 ± 17.91	183.87 ± 21.64	316.01 ± 29.84
minus Source-only	-40.44 (6.09), 1/10	-32.83 (6.89), 1/10	$+22.51$ (8.24), 7/10	$+96.76$ (7.24), 10/10
Pooled	163.95 ± 11.40	160.88 ± 13.22	168.41 ± 16.68	310.31 ± 28.21
minus Source-only	$+0.84$ (1.77), 7/10	$+0.04$ (1.99), 7/10	$+7.05$ (4.12), 7/10	$+91.06$ (7.25), 10/10
Myopic oracle	196.35 ± 0.64	197.63 ± 0.82	203.93 ± 2.33	245.20 ± 19.12
Empty-backlog share	0.90	0.88	0.66	0.19

Table 18: Offline reward-shift transfer on HalfCheetah, target return (↑, less negative is better), mean ± std over held-out seeds (D4RL 20, Minari 10), with the paired difference A^{2}IQL minus method (SE) and the number of seeds on which A^{2}IQL is ahead. A^{2}IQL uses $\lambda = 16$ and τ_b at the 50th percentile on D4RL and $\lambda = 4$ and the 75th percentile on Minari. Tuned baselines use the values selected on the tuning seeds.

	D4RL medium-expert		Minari simple, medium, expert	
Method	Return	A^{2}IQL minus (SE), ahead	Return	A^{2}IQL minus (SE), ahead
A^{2}IQL	-1189 ± 184	–	-1349 ± 133	–
Bonus only (no $A^{\mathtt{sim}}$)	-994 ± 72	-195 (44), 4/20	-1286 ± 143	-63 (44), 3/10
A^{2}IQL, $\lambda = 0$	-5803 ± 358	$+4614$ (73), 20/20	-4636 ± 735	$+3287$ (251), 10/10
IGDF-style, tuned	-1548 ± 1079	$+359$ (256), 8/20	-4072 ± 1935	$+2723$ (641), 9/10
OTDF-style, tuned	-3512 ± 1023	$+2323$ (239), 20/20	-6371 ± 611	$+5022$ (195), 10/10
DARA-style, tuned	-5642 ± 185	$+4453$ (54), 20/20	-4211 ± 478	$+2862$ (168), 10/10
Pooled	-5953 ± 149	$+4764$ (61), 20/20	-4505 ± 431	$+3155$ (156), 10/10
Source-only	-5968 ± 125	$+4778$ (58), 20/20	-4708 ± 349	$+3359$ (147), 10/10
Target-only	-5174 ± 492	$+3985$ (136), 20/20	-7134 ± 277	$+5784$ (91), 10/10
Target oracle	-836 ± 10	-353 (43), 0/20	-1165 ± 45	-184 (40), 0/10

Table 19: Dynamics shift on HalfCheetah with the D4RL source data, gravity scaled by 1.25 and 1.5 and the native reward. Return (↑), mean $\pm$ std over 10 seeds, with the paired difference to Source-only (SE) and the seeds on which the method is ahead. All baselines use their default parameter values here.

	Gravity $\times 1.25$		Gravity $\times 1.5$	
Method	Return	Minus Source-only (SE), ahead	Return	Minus Source-only (SE), ahead
Source-only	7347 ± 337	–	1830 ± 644	–
Pooled	7469 ± 209	$+122$ (103), 7/10	3319 ± 1235	$+1489$ (390), 9/10
IGDF-style	7585 ± 156	$+238$ (111), 7/10	3490 ± 527	$+1660$ (198), 10/10
OTDF-style	6574 ± 866	-773 (353), 2/10	3053 ± 760	$+1223$ (313), 9/10
DARA-style	6909 ± 345	-438 (177), 2/10	858 ± 932	-972 (186), 1/10
Target-only	1527 ± 739	-5820 (266), 0/10	1298 ± 159	-532 (238), 2/10
A^2IQL ($\lambda = 4$, 75th percentile)	7409 ± 319	$+62$ (172), 7/10	2152 ± 1006	$+321$ (216), 7/10
A^2IQL, $\lambda = 0$	7300 ± 317	-46 (90), 6/10	2209 ± 940	$+379$ (364), 6/10

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and the introduction state the asymmetric decomposition behind AIQL, the adaptation step of A^2IQL for a reward or objective shift, and the three testbeds. The quantitative claims match Tables 1 and 2 and Sections 6.4 and 6.5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The Limitations paragraph of the Conclusion states that the aggregate is manually engineered with Proposition 4.3 as its scope, that the warehouse results rest on a proprietary simulator with 1,230 source and 41 target transitions, which comparisons remain future work, that the Bayesian reward predictor uses fixed features, that the MuJoCo tuning budget favored A^2IQL, and that on the warehouse the reward model is dominated by its prior, so its bonus selects source days by their alignment with the target inputs.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Assumption 4.1, Proposition 4.2 and Proposition 4.3 state their assumptions in full, and Appendix B proves both propositions. Proposition 4.3 bounds the suboptimality when the assumption holds only approximately, and Appendix A tests the assumption on the warehouse simulator.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Sections 4 and 5 and Algorithms 1 and 2 define both methods in full. Appendix D states what the code computes on the synthetic and MuJoCo testbeds, including the reward-model features, the bonus threshold and scale, and the weight caps and normalization, and Table 5 lists the hyperparameters. Each table caption gives the number of seeds. Appendix F specifies the synthetic benchmark in full, and Appendix G gives the MuJoCo protocol with its tuning and held-out seeds. The warehouse simulator and its data are proprietary.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The synthetic benchmark and the code of AIQL, A^{2}IQL and the baselines on it will be released with the arXiv version of this paper, and Appendix F specifies the benchmark in full. The warehouse simulator and data are proprietary and cannot be released. The MuJoCo experiments use the public D4RL and Minari datasets, and Appendix G describes their protocol.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 6 and Appendices D, F and G give the data sizes, selection protocols and seeds of all testbeds and the hyperparameters of the synthetic and MuJoCo experiments. For MuJoCo, Appendix G.3 gives the tuning grids and the selected values, with tuning on seeds that are not used for reporting.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Result tables report the mean and standard deviation over training seeds, with 10 and 5 seeds on the warehouse, 10 to 20 on the synthetic benchmark, and 10 or 20 held-out seeds on MuJoCo. The constraint-satisfied days of Table 2, the per-seed values of Table 8, the unseen-dataset values of Table 6 and the derived ratios and weighted means have no spread, and each caption states what it reports. Paired comparisons report the mean per-seed difference and the number of seeds on which a method is ahead, and Tables 10 and 17 to 19 also report its standard error. Appendix D.4 states how the spreads and standard errors are computed. Training seeds set the network initialization and the minibatch order, and on the synthetic benchmark also the generated data and the environment constants.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Warehouse training ran on one NVIDIA Tesla T4 GPU, and the paper does not report the warehouse simulation resources or run times. The synthetic experiments run on CPU and took 81 and 129 minutes with 12 workers for the two experiment scripts and 5.6 minutes with 16 workers for the transition-only shift. MuJoCo baseline jobs of 300k gradient steps took 83 to 92 minutes each on a lightly loaded machine, and the phase-2 jobs of the single-setting check took 23.0 minutes on average. The tuning campaign took 25.8 hours and the campaign with the dynamics-shift runs 697.8 minutes, with 5 GPU worker processes sharing one NVIDIA Tesla T4 and 10 to 13 CPU worker processes. The full project used more compute than the reported runs, including preliminary and discarded runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research involves optimization of warehouse operations in simulation. No human subjects, personal data, or dual-use concerns are involved.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This work is a methodological contribution to offline reinforcement learning, evaluated entirely in simulation. The method addresses a general problem (offline RL with large ranking action spaces and cross-domain adaptation) and is not tied to a specific deployment. No policies were deployed in a live environment.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

Justification: The code to be released contains a synthetic ranking benchmark and offline RL baselines, which pose no risk of misuse. No warehouse models or data will be released.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We use the D4RL halfcheetah-medium-expert-v2 dataset [38], whose datasets are licensed CC BY 4.0 and code Apache 2.0, the Minari HalfCheetah datasets through the Minari library [39], whose library is licensed MIT, Gymnasium [37] under MIT and MuJoCo [36] under Apache 2.0. All baselines are our implementations of published methods in PyTorch and NumPy. The warehouse simulator and data are proprietary.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The new asset is the synthetic ranking benchmark. Appendix F specifies it in full, and its code will be released with the arXiv version of this paper, with a README that documents the environment, the methods and the commands that reproduce Appendix F.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as a component of the proposed method. Any LLM usage was limited to writing assistance and does not impact the core methodology.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.