

Appendix

496

A Limitations and Future Work

498 Despite its strong empirical performance, IOSTOM has some limitations. First, like most LfO
499 methods, it assumes access to high-quality consecutive state pairs (s, s') , which may not always
500 be available in real-world datasets. Second, we assume that actions are fully observable in the
501 sub-optimal dataset, which might not hold in practice. While these limitations are beyond the scope
502 of this work, they highlight important directions for future research.

B Missing Proofs and Derivations

B.1 Proof of Proposition 1

503 **Proposition.** *The joint visitation distribution $d^g(s, s')$ can be expressed as:*

$$d^g(s, s') = (1 - \gamma)d_0(s)g(s' | s) + \gamma g(s' | s) \sum_{\bar{s}} d^g(\bar{s}, s) \quad (10)$$

506 *Proof.* We recall that $d^\pi(s, a) = (1 - \gamma)d_0(s)\pi(a | s) + \gamma \sum_{s', a'} d^\pi(s', a')\mathcal{T}(s | s', a')\pi(a | s)$.
507 Therefore, we have:

$$\begin{aligned} d^g(s, s') &= \sum_a d^\pi(s, a)\mathcal{T}(s' | s, a) \\ &= \sum_a (1 - \gamma)d_0(s)\pi(a | s)\mathcal{T}(s' | s, a) + \sum_a \gamma \mathcal{T}(s' | s, a) \sum_{\bar{s}, \bar{a}} d^\pi(\bar{s}, \bar{a})\mathcal{T}(s | \bar{s}, \bar{a})\pi(a | \bar{s}) \\ &= (1 - \gamma)d_0(s)g(s' | s) + \gamma g(s' | s) \sum_{\bar{s}, \bar{a}} d^\pi(\bar{s}, \bar{a})\mathcal{T}(s | \bar{s}, \bar{a}) \\ &= (1 - \gamma)d_0(s)g(s' | s) + \gamma g(s' | s) \sum_{\bar{s}} d^g(\bar{s}, s). \end{aligned}$$

508 as desired. □

B.2 Proof of Proposition 2

510 **Proposition.** *The constrained optimization problem in Equation (2) is equivalent to the following*
511 *unconstrained max-min problem:*

$$\begin{aligned} \max_g \min_{Q(s, s')} &\left\{ \alpha(1 - \gamma) \mathbb{E}_{(s, s') \sim d_0} [Q(s, s')] + \mathbb{E}_{(s, s') \sim d_{mix}^{E, I}} [f^* (\gamma \mathbb{E}_{s'' \sim g(\cdot | s')} [Q(s', s'')] - Q(s, s'))] \right. \\ &\quad \left. - (1 - \alpha) \mathbb{E}_{(s, s') \sim d^I} [\gamma \mathbb{E}_{s'' \sim g(\cdot | s')} [Q(s', s'')] - Q(s, s')] \right\}, \end{aligned}$$

512 where $Q(s, s')$ are the Lagrange multipliers and f^* denotes the convex conjugate of a chosen convex
513 function f .

514 *Proof.* We recall that the primal formulation in Equation (2) is as follows:

$$\begin{aligned} \max_{g, d \geq 0} & -\mathbb{D}_f(d_{mix}^I(s, s') \| d_{mix}^{E, I}(s, s')) \\ \text{s.t.} & d(s, s') = (1 - \gamma)d_0(s)g(s' | s) + \gamma g(s' | s) \sum_{\bar{s}} d(\bar{s}, s) \end{aligned}$$

515 We first apply duality on the inner maximization problem of the above formulation:

$$\begin{aligned} & \max_{g, d \geq 0} \min_{Q(s, s')} -\mathbb{D}_f(d_{mix}^I(s, s') \| d_{mix}^{E, I}(s, s')) \\ & + \alpha \sum_{s, s'} Q(s, s') \left((1 - \gamma) d_0(s) \cdot g(s' | s) + \gamma g(s' | s) \sum_{\bar{s}} d(\bar{s}, s) - d(s, s') \right) \end{aligned} \quad (11)$$

$$\begin{aligned} & = \max_{\pi, d \geq 0} \min_{Q(s, s')} \alpha(1 - \gamma) \mathbb{E}_{d_0(s), g(s' | s)} [Q(s, s')] \\ & + \alpha \mathbb{E}_{s, s' \sim d} \left[\gamma \sum_{s''} g(s'' | s') Q(s', s'') - Q(s, s') \right] - \mathbb{D}_f(d_{mix}^I(s, s') \| d_{mix}^{E, I}(s, s')) \end{aligned} \quad (12)$$

516 Step (11) to (12) is equivalent to changing the following order of summation:

$$\begin{aligned} & \sum_{s, s'} Q(s, s') g(s' | s) \sum_{\bar{s}} d(\bar{s}, s) \\ & = \sum_{\bar{s}, s} d(\bar{s}, s) \sum_{s'} Q(s, s') g(s' | s) \\ & = \sum_{s, s'} d(s, s') \sum_{s''} Q(s', s'') g(s'' | s') \end{aligned}$$

517 By adding and subtracting another term below, (12) becomes:

$$\begin{aligned} & = \max_{g, d \geq 0} \min_{Q(s, s')} \alpha(1 - \gamma) \mathbb{E}_{d_0(s), g(s' | s)} [Q(s, s')] \\ & + \alpha \mathbb{E}_{s, s' \sim d} \left[\gamma \sum_{s''} g(s'' | s') Q(s', s'') - Q(s, s') \right] \\ & + (1 - \alpha) \mathbb{E}_{s, s' \sim d^I} \left[\gamma \sum_{s''} g(s'' | s') Q(s', s'') - Q(s, s') \right] \\ & - (1 - \alpha) \mathbb{E}_{s, s' \sim d^I} \left[\gamma \sum_{s''} g(s'' | s') Q(s', s'') - Q(s, s') \right] \\ & - \mathbb{D}_f(d_{mix}^I(s, s') \| d_{mix}^{E, I}(s, s')) \end{aligned} \quad (13)$$

518 We can swap \max_d and \min_Q in (13) due to strong duality.

$$\begin{aligned} (13) & = \max_g \min_{Q(s, s')} \max_{d_{mix}^I(s, s') \geq 0} \alpha(1 - \gamma) \mathbb{E}_{d_0(s), g(s' | s)} [Q(s, s')] \\ & + \mathbb{E}_{s, s' \sim d_{mix}^I} \left[\gamma \sum_{s''} g(s'' | s') Q(s', s'') - Q(s, s') \right] - \mathbb{D}_f(d_{mix}^I(s, s') \| d_{mix}^{E, I}(s, s')) \\ & - (1 - \alpha) \mathbb{E}_{s, s' \sim d^I} \left[\gamma \sum_{s''} g(s'' | s') Q(s', s'') - Q(s, s') \right] \end{aligned} \quad (14)$$

$$\begin{aligned} & = \max_g \min_{Q(s, s')} \alpha(1 - \gamma) \mathbb{E}_{d_0(s), g(s' | s)} [Q(s, s')] \\ & + \mathbb{E}_{s, s' \sim d_{mix}^{E, I}} \left[f^* \left(\gamma \sum_{s''} g(s'' | s') Q(s', s'') - Q(s, s') \right) \right] \\ & - (1 - \alpha) \mathbb{E}_{s, s' \sim d^I} \left[\gamma \sum_{s''} g(s'' | s') Q(s', s'') - Q(s, s') \right] \end{aligned} \quad (15)$$

where f^* is convex conjugate of convex f -divergence function. (14) to (15) can be proved by the following equation using the interchangeability principle [5]:

$$\begin{aligned}
& \max_{d_{mix}^I(s, s') \geq 0} \mathbb{E}_{s, s' \sim d_{mix}^I} \left[\gamma \sum_{s''} g(s'' | s') Q(s', s'') - Q(s, s') \right] - \mathbb{D}_f(d_{mix}^I(s, s') \| d_{mix}^{E, I}(s, s')) \\
&= \max_{d_{mix}^I(s, s') \geq 0} \mathbb{E}_{s, s' \sim d_{mix}^{E, I}} \left[\frac{d_{mix}^I(s, s')}{d_{mix}^{E, I}(s, s')} \left(\gamma \sum_{s''} g(s'' | s') Q(s', s'') - Q(s, s') \right) - f \left(\frac{d_{mix}^I(s, s')}{d_{mix}^{E, I}(s, s')} \right) \right] \\
&= \mathbb{E}_{s, s' \sim d_{mix}^{E, I}} \left[f^* \left(\gamma \sum_{s''} g(s'' | s') Q(s', s'') - Q(s, s') \right) \right]
\end{aligned}$$

Finally, the objective (15) is the unconstrained dual problem of Equation (2). \square

B.3 Proof of Proposition 3

Proposition. $\max_g \{V_Q^g(s)\}$ can be approximated via the following Extreme-V objective:

$$\min_V \left\{ J(V | Q) = \mathbb{E}_{(s, s') \sim d_{mix}^{E, I}} [\exp(\omega(s, s')) + \omega(s, s') - 1] \right\}.$$

where $\omega(s, s') = (Q(s, s') - V(s))/\beta$.

Proof. Recall that:

$$V_Q^g(s) = \mathbb{E}_{s' \sim g(\cdot | s)} \left[Q(s, s') - \beta \log \frac{g(s' | s)}{\mu(s' | s)} \right],$$

which is the expected reward under transition distribution $g(\cdot | s)$, regularized by the KL divergence from a reference distribution $\mu(\cdot | s)$. Moreover, the problem $\max_g \{V_Q^g(s)\}$ is a classic entropy-regularized expected reward maximization problem. The optimal solution has a closed form:

$$\max_g \{V_Q^g(s)\} = \beta \log \sum_{s'} \mu(s' | s) \exp \left(\frac{Q(s, s')}{\beta} \right). \quad (16)$$

We now write the function $J(V | Q)$ as:

$$J(V | Q) = \sum_{(s, s')} \mu(s' | s) \left[\exp \left(\frac{Q(s, s') - V(s)}{\beta} \right) + \frac{Q(s, s') - V(s)}{\beta} - 1 \right].$$

For any state s , and fixed Q , the function $J(V | Q)$ is convex in $V(s)$ because:

- The exponential function $\exp \left(\frac{Q(s, s') - V(s)}{\beta} \right)$ is convex in $V(s)$,
- The linear term $(Q(s, s') - V(s))/\beta$ is also convex (affine),
- The sum and non-negative weights preserve convexity.

To find the minimum of $J(V | Q)$ with respect to V , we take the derivative with respect to $V(s)$ and set it to zero:

$$\frac{\partial J(V | Q)}{\partial V(s)} = \sum_{s'} \mu(s' | s) \left[-\frac{1}{\beta} \exp \left(\frac{Q(s, s') - V(s)}{\beta} \right) - \frac{1}{\beta} \right] = 0.$$

Rewriting:

$$\sum_{s'} \mu(s' | s) \exp \left(\frac{Q(s, s') - V(s)}{\beta} \right) = \sum_{s'} \mu(s' | s).$$

We have $\sum_{s'} \mu(s' | s) = 1$, this gives:

$$\sum_{s'} \mu(s' | s) \exp \left(\frac{Q(s, s') - V(s)}{\beta} \right) = 1.$$

538 Bringing the constant outside the exponential:

$$\begin{aligned} & \exp\left(-\frac{V(s)}{\beta}\right) \sum_{s'} \mu(s' | s) \exp\left(\frac{Q(s, s')}{\beta}\right) = 1, \\ \Rightarrow \exp\left(-\frac{V(s)}{\beta}\right) &= \frac{1}{\sum_{s'} \mu(s' | s) \exp\left(\frac{Q(s, s')}{\beta}\right)}, \end{aligned}$$

540 Taking the logarithm of both sides and solving for $V(s)$, we obtain the closed-form solution to
541 $\min_V J(V|Q)$ as:

$$V^*(s) = \beta \log \sum_{s'} \mu(s' | s) \exp\left(\frac{Q(s, s')}{\beta}\right). \quad (17)$$

542 Combined (16) and (17) we get:

$$V^*(s) = \max_g \{V_Q^g(s)\}$$

543 as desired. \square

544 **B.4 Proof of Proposition 4**

545 **Proposition.** *Under any convex function f , $L(Q, V)$ is concave in Q , and the Extreme-V loss*
546 *$J(V | Q)$ is convex in V .*

547 *Proof.* We rewrite the objective $L(Q, V)$ as:

$$\begin{aligned} \min_Q L(Q, V) &= \alpha(1 - \gamma) \mathbb{E}_{(s, s') \sim \mathcal{D}_0} [Q(s, s')] + \alpha \mathbb{E}_{(s, s') \sim \mathcal{D}^E} [f^*(\gamma V(s') - Q(s, s'))] \\ &\quad + (1 - \alpha) \mathbb{E}_{(s, s') \sim \mathcal{D}^I} [\widetilde{f^*}(\gamma V(s') - Q(s, s'))]. \end{aligned}$$

548 We now analyze the convexity of $L(Q, V)$ with respect to Q . Note the following:

- 549 • The first term, $\mathbb{E}_{\mathcal{D}_0} [Q(s, s')]$, is linear in Q , and hence convex.
- 550 • The functions f^* and $\widetilde{f^*}$ are convex (as they are convex conjugates of proper convex
551 functions).
- 552 • The composition of a convex function with an affine function (i.e., $\gamma V(s') - Q(s, s')$) is
553 convex in Q .
- 554 • Expectations of convex functions preserve convexity.

555 Therefore, each term in $L(Q, V)$ is convex in Q , and the entire objective $L(Q, V)$ is convex in Q , as
556 desired.

557 The convexity of $J(V | Q)$ in V follows directly from the discussion in the proof of Proposition (3).
558 \square

559 **B.5 Proof of Proposition 5**

560 **Proposition.** *The following Q -weighted behavior cloning objective returns the same optimal implicit*
561 *policy as the original advantage-weighted BC formulation:*

$$\max_g \mathbb{E}_{(s, s') \sim \mathcal{D}^I} [\exp(\tau Q(s, s')) \log g(s' | s)]$$

562 *Proof.* We write the objective function as:

$$F(g) = \sum_{(s, s')} \mu^I(s' | s) \exp(\tau Q(s, s')) \log g(s' | s),$$

where $\mu^I(s' | s)$ is the state-to-state transition probability (i.e., the implicit behavior policy) for the dataset \mathcal{D}^I . For each fixed state s , the expression

$$\sum_{s'} \mu^I(s' | s) \exp(\tau Q(s, s')) \log g(s' | s)$$

is a weighted log-likelihood, where the weights $\mu^I(s' | s) \exp(\tau Q(s, s'))$ are known. Maximizing this with respect to $g(\cdot | s)$ under the constraint that $g(\cdot | s)$ is a valid probability distribution (i.e., $\sum_{s'} g(s' | s) = 1$) leads to a standard result from maximum likelihood estimation with importance weights. The closed-form solution is:

$$g^*(s' | s) = \frac{\mu^I(s' | s) \exp(\tau Q(s, s'))}{\sum_{s''} \mu^I(s'' | s) \exp(\tau Q(s, s''))}.$$

We now consider the advantage-weighted behavior cloning objective:

$$\max_g \mathbb{E}_{(s, s') \sim \mathcal{D}^I} [\exp(\tau(Q(s, s') - V(s))) \log g(s' | s)],$$

In a similar fashion to soft behavior cloning, this yields the following closed-form optimal “implicit policy”:

$$g^{**}(s' | s) = \frac{\mu^I(s' | s) \exp(\tau(Q(s, s') - V(s)))}{\sum_y \mu^I(y | s) \exp(\tau(Q(s, y) - V(s)))},$$

where $V(s)$ appears in both the numerator and denominator and thus cancels out. This simplifies the expression and leads to:

$$g^*(s' | s) = g^{**}(s' | s),$$

indicating the equivalence between the advantage-weighted behavior cloning and the Q -weighted behavior cloning formulations.

□

B.6 Proof of Proposition 6

Proposition. *The objective $F(\pi)$ is lower-bounded by the following surrogate function $\tilde{F}(\pi)$, up to an additive constant: $\tilde{F}(\pi) = \mathbb{E}_{(s, s') \sim \mathcal{D}^I} [\exp(\tau Q(s, s')) \sum_a \mathcal{I}(a | s, s') \log \pi(a | s)]$.*

Proof. We write the objective function as:

$$F(\pi) = \mathbb{E}_{(s, s') \sim \mathcal{D}^I} \left[\exp(\tau Q(s, s')) \log \left(\sum_a \mathcal{T}(s' | s, a) \pi(a | s) \right) \right].$$

Given that the logarithm function is concave, we apply Jensen’s inequality. Define:

$$\Delta(s, s') = \sum_a \mathcal{T}(s' | s, a),$$

Then we have:

$$\log \left(\sum_a \mathcal{T}(s' | s, a) \pi(a | s) \right) = \log \left(\sum_a \frac{\mathcal{T}(s' | s, a)}{\Delta(s, s')} \pi(a | s) \right) + \log \Delta(s, s') \quad (18)$$

$$\geq \sum_a \frac{\mathcal{T}(s' | s, a)}{\Delta(s, s')} \log \pi(a | s) + \log \Delta(s, s') \quad (19)$$

$$= \sum_a \mathcal{I}(a | s, s') \log \pi(a | s) + \log \Delta(s, s'). \quad (20)$$

Substituting this back into the original objective yields the lower bound:

$$F(\pi) \geq \mathbb{E}_{(s, s') \sim \mathcal{D}^I} \left[\exp(\tau Q(s, s')) \sum_a \mathcal{I}(a | s, s') \log \pi(a | s) \right] + \mathbb{E}_{(s, s') \sim \mathcal{D}^I} [\exp(\tau Q(s, s')) \log \Delta(s, s')].$$

584 The second term is independent of π and can be treated as a constant during training. Therefore, we
 585 can optimize the surrogate lower bound:

$$\tilde{F}(\pi) = \mathbb{E}_{(s,s') \sim \mathcal{D}^I} \left[\exp(\tau Q(s, s')) \sum_a \mathcal{I}(a \mid s, s') \log \pi(a \mid s) \right].$$

586

□

587 B.7 Complete Derivation of $L(Q, V)$ using Pearson χ^2 divergence

588 We recall that the objective function $L(Q, V)$ has the following form:

$$\begin{aligned} \min_Q L(Q, V) &= \alpha(1 - \gamma) \mathbb{E}_{(s,s') \sim d_0} [Q(s, s')] + \alpha \mathbb{E}_{(s,s') \sim d^E} [f^*(\gamma V(s') - Q(s, s'))] \\ &\quad + (1 - \alpha) \mathbb{E}_{(s,s') \sim d^I} [\widetilde{f^*}(\gamma V(s') - Q(s, s'))] \end{aligned} \quad (21)$$

589 where f^* is the convex conjugate of divergence function f and $\widetilde{f^*}(x) = f^*(x) - x$. Under Pearson
 590 χ^2 divergence, its convex conjugate $f^*(x) = \frac{x^2}{4} + x$ and the associated $\widetilde{f^*}(x) = \frac{x^2}{4}$. The objective
 591 (21) with Pearson χ^2 divergence becomes:

$$\begin{aligned} \min_Q L(Q, V) &= \alpha(1 - \gamma) \mathbb{E}_{(s,s') \sim d_0} [Q(s, s')] + \alpha \mathbb{E}_{(s,s') \sim d^E} [\gamma V(s') - Q(s, s')] \\ &\quad + \frac{\alpha}{4} \mathbb{E}_{(s,s') \sim d^E} [(\gamma V(s') - Q(s, s'))^2] + \frac{1 - \alpha}{4} \mathbb{E}_{(s,s') \sim d^I} [(\gamma V(s') - Q(s, s'))^2] \end{aligned} \quad (22)$$

$$\Leftrightarrow \min_Q L(Q, V) = (1 - \gamma) \mathbb{E}_{(s,s') \sim d_0} [Q(s, s')] + \mathbb{E}_{(s,s') \sim d^E} [\gamma V(s') - Q(s, s')] \quad (23)$$

$$\begin{aligned} &\quad + \frac{1}{4\alpha} \alpha \mathbb{E}_{(s,s') \sim d^E} [(\gamma V(s') - Q(s, s'))^2] + \frac{1}{4\alpha} (1 - \alpha) \mathbb{E}_{(s,s') \sim d^I} [(\gamma V(s') - Q(s, s'))^2] \\ &\Leftrightarrow \min_Q L(Q, V) = (1 - \gamma) \mathbb{E}_{(s,s') \sim d_0} [Q(s, s')] + \mathbb{E}_{s \sim d^E} [\gamma V(s)] - \mathbb{E}_{(s,s') \sim d^E} [Q(s, s')] \\ &\quad + \frac{1}{4\alpha} \mathbb{E}_{s,s' \sim d_{mix}^{E,I}} [(\gamma V(s') - Q(s, s'))^2] \end{aligned} \quad (24)$$

592 B.8 Complete Derivation of $\tilde{L}(Q, V)$ for bounded Q-learning

593 The operator $\min_Q -\alpha \mathbb{E}_{(s,s') \sim \mathcal{D}^E} [Q(s, s')]$ in (24) which effectively encourages maximizing the
 594 Q -values of expert transitions can lead to *unbounded* growth in Q , potentially resulting in learning
 595 instability. To address this issue, we adapt a technique from [2] that bounds the expert Q -values. First,
 596 looking at the objective 23, Let's define $r_Q^E(s, s') = Q(s, s') - \gamma V(s') \forall s, s' \sim d^E$. The training

597 objective becomes:

$$\begin{aligned}
\min_Q L(Q, V) &= (1 - \gamma) \mathbb{E}_{(s, s') \sim d_0} [Q(s, s')] + \mathbb{E}_{(s, s') \sim d^E} [-r_Q^E(s, s')] \\
&+ \frac{1}{4\alpha} \alpha \mathbb{E}_{(s, s') \sim d^E} [r_Q^E(s, s')^2] + \frac{1}{4\alpha} (1 - \alpha) \mathbb{E}_{(s, s') \sim d^I} [(\gamma V(s') - Q(s, s'))^2] \\
&\Leftrightarrow \min_Q L(Q, V) = (1 - \gamma) \mathbb{E}_{d_0(s, s')} [Q(s, s')] + \frac{1 - \alpha}{4\alpha} \mathbb{E}_{s, s' \sim d^I} [(\gamma V(s') - Q(s, s'))^2] \\
&+ \left[\mathbb{E}_{s, s' \sim d^E} [-r_Q^E(s, s')] + \frac{1}{4} \mathbb{E}_{s, s' \sim d^E} [r_Q^E(s, s')^2] \right] \\
&\Leftrightarrow \min_Q L(Q, V) = (1 - \gamma) \mathbb{E}_{d_0(s, s')} [Q(s, s')] + \frac{1 - \alpha}{4\alpha} \mathbb{E}_{s, s' \sim d^I} [(\gamma V(s') - Q(s, s'))^2] \\
&+ \frac{1}{4} [\mathbb{E}_{s, s' \sim d^E} [-4r_Q^E(s, s')] + \mathbb{E}_{s, s' \sim d^E} [r_Q^E(s, s')^2] + 4] - 1 \\
&\Leftrightarrow \min_Q L(Q, V) = \min_{Q(s, s')} (1 - \gamma) \mathbb{E}_{d_0(s, s')} [Q(s, s')] + \frac{1 - \alpha}{4\alpha} \mathbb{E}_{s, s' \sim d^I} [(\gamma V(s') - Q(s, s'))^2] \\
&+ \frac{1}{4} \mathbb{E}_{s, s' \sim d^E} [(r_Q^E(s, s') - 2)^2] \tag{25}
\end{aligned}$$

$$\begin{aligned}
&\Leftrightarrow \min_Q L(Q, V) = (1 - \gamma) \mathbb{E}_{d_0(s, s')} [Q(s, s')] + \frac{1 - \alpha}{4\alpha} \mathbb{E}_{s, s' \sim d^I} [(\gamma Q(s', g) - Q(s, s'))^2] \\
&+ \frac{1}{4} \mathbb{E}_{s, s' \sim d^E} [(Q(s, s') - (\gamma V(s') + 2))^2] \tag{26}
\end{aligned}$$

598 Following [2], the minimum of the third term in (25) is reached when $r_Q^E(s, s') = 2$. This will lead to
599 $Q(s, s') = \sum_{t=0}^{\infty} \gamma^t 2 = \frac{2}{1-\gamma} \forall s, s' \sim d^E$. Therefore, we can replace the target $\gamma V(s') + 2$ in (26)
600 with fixed target $\frac{2}{1-\gamma}$ to have the following modified objective with bounded Q .

$$\begin{aligned}
\min_Q \tilde{L}(Q, V) &= (1 - \gamma) \mathbb{E}_{d_0(s, s')} [Q(s, s')] + \frac{1 - \alpha}{4\alpha} \mathbb{E}_{s, s' \sim d^I} [(\gamma V(s') - Q(s, s'))^2] \\
&+ \frac{1}{4} \mathbb{E}_{s, s' \sim d^E} \left[\left(Q(s, s') - \frac{2}{1 - \gamma} \right)^2 \right]
\end{aligned}$$

601 C Experimental and Implementation Details

602 Our method is implemented in JAX version 0.5.3 (with CUDA 12 capabilities). We conduct our
603 experiments using a computing cluster with 8 NVIDIA RTX 3090 GPUs. For each IOSTOM run,
604 five distinct training seeds are processed simultaneously on a shared hardware set comprising a single
605 GPU, 32 CPU cores, and 128 GB of RAM. This parallel execution on shared resources enables the
606 completion of 1 million training steps for all five seeds in about 60-90 minutes.

607 C.1 Mujoco tasks

608 We use the same offline LfO benchmark from DILO [37], which utilizes datasets derived from
609 the D4RL [7] framework, and tests on Mujoco environments. The state-only expert dataset in all
610 tasks includes only one expert trajectory. In terms of locomotion tasks, suboptimal datasets, labeled
611 ‘random+expert’, ‘random+few-expert’, ‘medium+expert’, and ‘medium+few-expert’, are generated
612 by mixing expert trajectories with lower-quality trajectories from D4RL’s ‘random-v2’ and ‘medium-
613 v2’ datasets, respectively. The ‘random+expert’ and ‘medium+expert’ datasets combine 200 expert
614 trajectories with roughly 1 million transitions from the corresponding ‘random-v2’ or ‘medium-v2’
615 dataset. The ‘x+few-expert’ variants are similar but incorporate only 30 expert trajectories. In
616 manipulation environments, all suboptimal ‘x+expert’ datasets are formed using 30 expert trajectories
617 mixed with the complete ‘x’ D4RL dataset. We also use ‘-v0’ variant of D4RL datasets for all
618 manipulation tasks. Table 4 gives an overview about our LfO Mujoco tasks.

Task	State Dim	Action Dim	Horizon	Suboptimal Dataset	Data Points
Hopper	11	3	1000	random+expert medium+expert random+few-expert medium+few-expert	1e6 random transitions + 200 expert trajectories 1e6 medium transitions + 200 expert trajectories 1e6 random transitions + 30 expert trajectories 1e6 medium transitions + 30 expert trajectories
Walker2d	17	6	1000	random+expert medium+expert random+few-expert medium+few-expert	1e6 random transitions + 200 expert trajectories 1e6 medium transitions + 200 expert trajectories 1e6 random transitions + 30 expert trajectories 1e6 medium transitions + 30 expert trajectories
Halfcheetah	17	6	1000	random+expert medium+expert random+few-expert medium+few-expert	1e6 random transitions + 200 expert trajectories 1e6 medium transitions + 200 expert trajectories 1e6 random transitions + 30 expert trajectories 1e6 medium transitions + 30 expert trajectories
Ant	27	8	1000	random+expert medium+expert random+few-expert medium+few-expert	1e6 random transitions + 200 expert trajectories 1e6 medium transitions + 200 expert trajectories 1e6 random transitions + 30 expert trajectories 1e6 medium transitions + 30 expert trajectories
Pen	45	24	100	cloned+expert human+expert	5e6 cloned transitions + 30 expert trajectories 5000 human transitions + 30 expert trajectories
Door	39	28	200	cloned+expert human+expert	1e6 cloned transitions + 30 expert trajectories 6729 human transitions + 30 expert trajectories
Hammer	46	26	200	cloned+expert human+expert	1e6 cloned transitions + 30 expert trajectories 11310 human transitions + 30 expert trajectories
Kitchen	59	9	280	partial+expert mixed+expert	136950 partial transitions + 1 expert trajectories 136950 mixed transitions + 1 expert trajectories

Table 4: Overview of D4RL tasks and their repsective suboptimal dataset we use in LfO setting

C.2 Maritime Navigation task

The Maritime Navigation task was created using historical data from a hotspot in the Singapore Strait. We selected the area with the highest traffic density and collision risk—where numerous ships cross paths, as shown in Figure 2—as our planning region. We collect large amount of historical navigation data (~ 2 years) of vessels operating in this hotspot region recorded in the Automatic Identification System (AIS) from MarineTraffic¹. The AIS data of each vessel contains two types of information: static and dynamic. The static data contains some information like width, length, type, and ID of vessel. Other vessel movement information like latitude, longitude, speed, heading and course-over-ground are included in dynamic data. To generate trajectory data, we selected tankers and cargo vessels as they represent the riskiest class due to their larger size (200-300 meters) and lower navigational agility. All trajectories were then interpolated at 10-second intervals. The final dataset comprises approximately 125,000 trajectories, totaling around 14 million environment transitions. The average trajectory length in dataset is around 100-150. We used 80% of the data for training and reserved the remaining 20% for evaluation.

The **observation space** is defined from the perspective of the ego agent (the vessel being controlled). At any given time, the agent observes a historical sequence of its own trajectory and those of the 10 closest nearby ships (each over a configurable number of past steps). For the ego agent and nearby ships, and for each historical point, the available features include the x and y coordinates, speed v , and heading angle h . Additionally, the agent observes its goal location. Observing past states and nearby ship information helps capture multi-ship interactions and provides context for decision-making. For simplicity, all algorithms used the same neural network architecture to process the observation space. We did not use any complex structures; instead, we flattened the observation space and provided it as input to the neural network.

The **action space** is modeled as a straightforward, 3-dimensional continuous space. An action is defined as $\langle d_x, d_y, d_h \rangle$, representing the changes in the x and y coordinates and the change in heading h , respectively. The vessel’s speed at the next time step is derived from the distance traveled (calculated from $v_{t+1} = \sqrt{d_x^2 + d_y^2} / \delta_T$) divided by the time interval δ_T , which is set to 10 seconds. This is also known as a delta action space [12] and can be used for any moving object. Because action

¹<https://www.marinetraffic.com/>



Figure 2: The red region is used as the environment area. The gray areas indicate anchorage zones, the green areas represent landmasses, and the arrows and regions with dark blue borders represent the Maritime Traffic Separation Scheme (TSS). The high density of crossing points in the red area makes it a more challenging region for navigation, providing a suitable setting for testing advanced planning techniques.

in our environment represents the difference between some state features of current and next timestep, we can have a simple Inverse Kinematics Model computing this difference to infer action between two consecutive states of state-only trajectories in datasets.

Following **vessel-specific metrics** are used to evaluate navigation policies, comparing learned agent behavior to human expert data.

Goal-Conditioned ADE (GC-ADE) measures the average displacement between the learned policy’s trajectory and the original historical trajectory in the 2D plane. Given τ_m of length T_m and τ_p of length T_p , GC-ADE computes the error over the minimum of the two lengths.

$$\text{GC-ADE} = \frac{1}{\min(T_m, T_p)} \sqrt{\sum_{t=1}^{\min(T_m, T_p)} (x_t^m - x_t^p)^2 + (y_t^m - y_t^p)^2}$$

Goal Rate is the percentage of times the ego agent successfully reaches its designated goal location. Success is defined as coming within a radius of 200 meters of the goal.

Near Miss Count represents the average number of time-steps per episode during which the ego agent approaches another vessel within a distance of 3 cable lengths (555 meters), which is considered a near-miss by domain experts. The ‘near-miss’ metric is interpreted broadly as a proxy for high traffic density and increased potential for navigation risk; it does not always imply that vessels in ‘near-miss’ situation were about to collide.

C.3 Architecture and Hyperparameters

Our implementation builds upon the official implementations of ReCOIL [38] and XQL [9]. We keep most of their parameters and network settings as shown in Table 5. We also add Layer Normalization [3] in V-function network to improve training stability as suggested in XQL. The regularization β was tuned by searching over [3, 5, 7, 10, 15, 20]. For locomotion tasks, we set $\beta = 20$ for standard LfO setting, and $\beta = 15$ for subsampled setting. In terms of manipulation tasks, $\beta = 10$ works best in most cases except ‘pen-cloned’ setting where β is set to 3. The policy temperature τ is often set to 3 in previous works [22, 38]. However, we find that this value results in very bad performance for IOSTOM because we do not use advantage for updating policy. We tune τ via hyper-parameter sweeps over [0.01, 0.04, 0.08, 0.1, 0.2]. $\tau = 0.04$ is the best-performing hyperparameter in most tasks except for the ‘human’ Adroit and ‘mixed’ Franka Kitchen manipulation tasks, where $\tau = 0.01$ was used. For maritime navigation task, we set $\beta = 20$ and $\tau = 0.04$ which is similar to LfO setting of locomotion tasks.

Type	Hyperparameter	Value
Actor	Network Size	[256, 256]
	Activation Function	ReLU
	Learning Rate	3e-4
	Weight Decay	1e-3
	Training Length	1M steps
	Batch Size	512
	Optimizer	Adam
	Dropout Rate	0.1
Critic	LR decay schedule	cosine
	Network Size	[256, 256]
	Activation Function	ReLU
	Learning Rate	3e-4
	Training Length	1M steps
	Batch Size	512
	Optimizer	Adam
	Mixture Ratio α	0.5
	Polyak Update Rate λ	0.005
	Discount Factor γ	0.99

Table 5: Hyperparameters of IOSTOM

675 C.4 Baselines

676 To evaluate the performance of our approach, we conduct comparative evaluations against three
677 established state-of-the-art (SOTA) techniques: SMODICE [27], PW-DICE [45], and DILO [37].
678 The SMODICE and PW-DICE algorithms both operate by training a discriminator to guide the
679 learned policy. Their fundamental difference lies in the divergence measure employed: SMODICE
680 seeks to minimize the KL-divergence between the state occupancies of the learner and the expert,
681 while PW-DICE alternatively uses the Wasserstein distance for this alignment. DILO offers a
682 distinct, more recent SOTA paradigm for LfO, notable for its discriminator-free learning process.
683 For all comparative methods, we utilize the publicly accessible codebases provided by their authors.
684 To ensure fair comparisons, we use the hyperparameter settings recommended in their original
685 publications or the default configurations within their code. The only exception is DILO where we
686 can not reproduce consistent results as reported in the paper using their default parameters. After some
687 tuning effort, we find that using Layer Normalization [3] can help to improve DILO performance.
688 However, the training still diverges in some tasks as shown in Figures 3 and 4

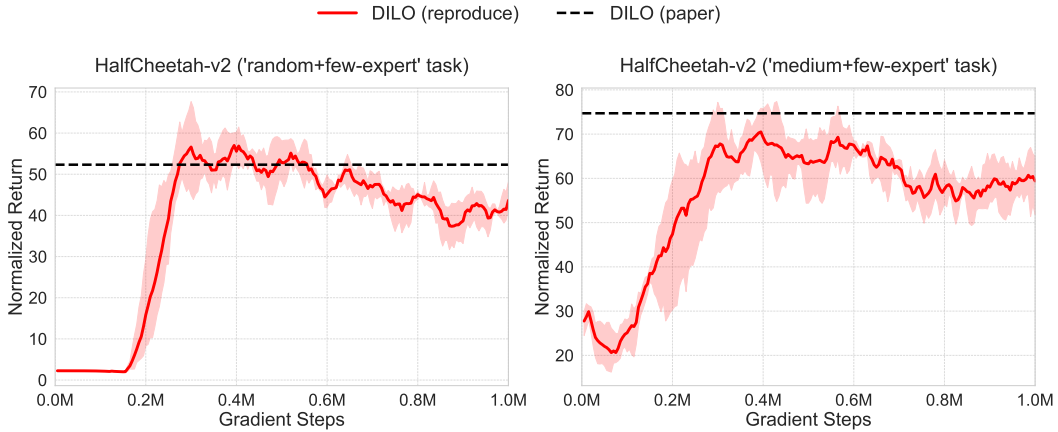


Figure 3: Training divergence of DILO on locomotion tasks

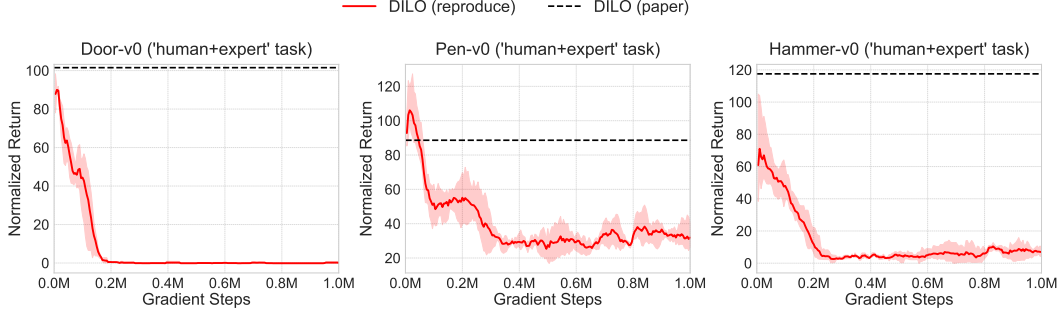


Figure 4: Training divergence of DILO on manipulation tasks

D Additional Experiments

D.1 LfO with mismatched expert

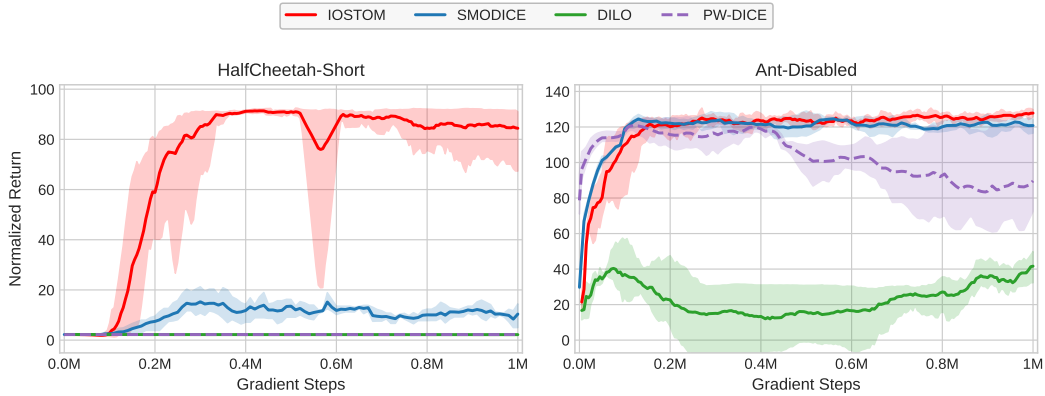


Figure 5: Comparison results for LfO with mismatched experts

In order to evaluate IOSTOM performance when learning from experts of different dynamics, we adopt SMODICE’s mismatched dynamics setting. We test on ‘HalfCheetah-Short’ (halved torso) and ‘Ant-Disabled’ (partially amputated front leg) (See Appendix H of SMODICE [27] for illustration), using one expert trajectory from these modified agents. The suboptimal dataset remains the ‘random+expert’ data (Section 6.1) from the original agents. This setting creates a clear mismatch between expert and interaction datasets. Figure 5 shows IOSTOM outperforming baselines on these challenging tasks, while DILO performs worst. The poor performance of DILO can be due to the use of visitation distribution $d(s, s', a')$ in its objective which matches the wrong a' in the mismatched expert dataset.

D.2 Comparison with other variants of IOSTOM

To validate our algorithmic designs, we compare IOSTOM with other variants: IOSTOM-IDM (Using Inverse Dynamics Model), IOSTOM-Adv (Using advantage instead of Q to update policy), and IOSTOM-IQL (Using Implicit Q Learning [22] objective to train V-function network). Table 6 shows these comparison results. Overall, IOSTOM has the best performance on 17/24 tasks and consistently produces high-quality results compared to other variants. IOSTOM-IQL is the second best method, but its performance is still significantly worse than IOSTOM on ‘cloned’ tasks. The results in ‘few-expert’ setting of IOSTOM-IDM is very bad compared to ‘expert’ setting which clearly shows the weakness of training Inverse Dynamics Model with low-quality data. IOSTOM-ADV has the worst performance in most tasks.

Suboptimal Dataset	Env	IOSTOM-ADV	IOSTOM-IDM	IOSTOM-IQL	IOSTOM	Expert
random+ expert	hopper	55.93 \pm 6.67	97.06 \pm 4.57	109.74 \pm 0.26	109.32 \pm 1.08	111.33
	halfcheetah	6.43 \pm 2.51	80.53 \pm 3.40	92.82 \pm 0.71	93.02 \pm 0.40	88.83
	walker2d	104.20 \pm 4.07	79.70 \pm 29.28	108.43 \pm 0.14	107.98 \pm 0.20	106.92
	ant	120.13 \pm 3.65	128.99 \pm 3.11	128.71 \pm 3.73	128.19 \pm 1.52	130.75
random+ few-expert	hopper	20.47 \pm 10.26	50.63 \pm 32.03	106.59 \pm 3.05	107.28 \pm 3.92	111.33
	halfcheetah	2.13 \pm 0.11	68.79 \pm 4.26	86.44 \pm 1.55	88.77 \pm 1.26	88.83
	walker2d	8.38 \pm 2.84	69.11 \pm 23.28	108.37 \pm 0.05	108.40 \pm 0.21	106.92
	ant	41.18 \pm 12.90	123.86 \pm 1.97	125.15 \pm 4.50	120.09 \pm 5.17	130.75
medium+ expert	hopper	53.46 \pm 13.16	97.62 \pm 7.42	110.71 \pm 0.35	110.20 \pm 0.51	111.33
	halfcheetah	49.00 \pm 3.44	85.21 \pm 1.86	91.45 \pm 1.49	93.12 \pm 0.32	88.83
	walker2d	85.96 \pm 21.33	94.90 \pm 29.34	108.45 \pm 0.17	108.12 \pm 0.13	106.92
	ant	118.74 \pm 4.55	128.32 \pm 0.64	124.66 \pm 4.62	124.72 \pm 3.49	130.75
medium few-expert	hopper	42.79 \pm 2.36	68.32 \pm 17.89	106.02 \pm 3.31	108.96 \pm 1.33	111.33
	halfcheetah	42.31 \pm 0.60	76.48 \pm 5.57	78.18 \pm 2.24	89.47 \pm 0.82	88.83
	walker2d	74.16 \pm 2.14	107.89 \pm 0.28	108.38 \pm 0.16	108.15 \pm 0.43	106.92
	ant	99.89 \pm 1.92	121.80 \pm 1.78	121.64 \pm 2.35	120.36 \pm 1.25	130.75
cloned+expert	pen	41.65 \pm 5.42	63.39 \pm 11.46	10.54 \pm 1.51	82.77 \pm 4.84	106.42
	door	13.87 \pm 8.26	18.68 \pm 12.44	32.25 \pm 13.03	102.77 \pm 0.96	103.94
	hammer	11.77 \pm 16.66	47.83 \pm 8.43	57.04 \pm 6.34	94.59 \pm 9.39	125.71
human+expert	pen	92.73 \pm 3.73	81.72 \pm 5.13	95.26 \pm 10.16	95.77 \pm 8.91	106.42
	door	95.08 \pm 1.90	78.50 \pm 20.55	99.47 \pm 3.67	100.77 \pm 1.68	103.94
	hammer	88.23 \pm 5.72	82.12 \pm 18.51	68.32 \pm 13.66	93.34 \pm 7.41	125.71
partial+expert	kitchen	56.08 \pm 0.29	66.30 \pm 5.70	57.75 \pm 2.00	58.95 \pm 2.27	75.0
mixed+expert	kitchen	49.42 \pm 0.58	28.95 \pm 10.62	47.92 \pm 1.23	46.45 \pm 0.84	75.0

Table 6: Average normalized return over last 10 evaluations of IOSTOM against other variants on the D4RL suboptimal datasets with 1 expert trajectory. The mean and std are obtained over 5 random seeds. LfO methods with avg. perf within the std-dev of the top performing LfO approach is in **bold**.

710 D.3 β Ablation

Suboptimal Dataset	Env	$\beta=3$	$\beta=5$	$\beta=10$	$\beta=15$	$\beta=20$	Expert
random+ expert	hopper	13.72 \pm 3.59	5.56 \pm 1.16	41.39 \pm 39.18	110.36 \pm 0.46	109.32 \pm 1.08	111.33
	halfcheetah	52.40 \pm 11.29	92.64 \pm 0.90	93.10 \pm 0.25	93.18 \pm 0.35	93.02 \pm 0.40	88.83
	walker2d	1.18 \pm 0.29	60.97 \pm 39.06	6.75 \pm 14.23	107.67 \pm 0.14	107.98 \pm 0.20	106.92
	ant	118.94 \pm 7.69	126.17 \pm 2.26	128.02 \pm 3.01	128.20 \pm 3.52	128.19 \pm 1.52	130.75
random+ few-expert	hopper	10.68 \pm 3.56	6.10 \pm 0.76	35.94 \pm 15.77	102.30 \pm 5.12	107.28 \pm 3.92	111.33
	halfcheetah	2.24 \pm 0.01	2.20 \pm 0.05	83.37 \pm 2.08	85.20 \pm 1.61	88.77 \pm 1.26	88.83
	walker2d	1.20 \pm 0.24	11.65 \pm 3.33	29.87 \pm 31.90	108.21 \pm 0.41	108.40 \pm 0.21	106.92
	ant	45.72 \pm 15.95	97.42 \pm 14.16	125.28 \pm 3.05	122.76 \pm 3.64	120.09 \pm 5.17	130.75
medium+ expert	hopper	31.99 \pm 24.17	61.48 \pm 19.37	109.97 \pm 0.42	110.02 \pm 1.00	110.20 \pm 0.51	111.33
	halfcheetah	43.20 \pm 0.47	54.63 \pm 3.08	92.63 \pm 0.21	92.96 \pm 0.29	93.12 \pm 0.32	88.83
	walker2d	71.70 \pm 4.56	107.83 \pm 0.75	108.31 \pm 0.27	108.28 \pm 0.12	108.12 \pm 0.13	106.92
	ant	98.70 \pm 1.81	102.01 \pm 2.96	121.86 \pm 2.49	124.12 \pm 2.93	124.72 \pm 3.49	130.75
medium few-expert	hopper	46.86 \pm 3.74	61.88 \pm 22.27	105.83 \pm 4.07	106.80 \pm 2.29	108.96 \pm 1.33	111.33
	halfcheetah	42.85 \pm 0.27	43.17 \pm 0.12	49.01 \pm 1.15	83.52 \pm 1.72	89.47 \pm 0.82	88.83
	walker2d	66.58 \pm 1.99	69.35 \pm 6.58	108.33 \pm 0.28	108.46 \pm 0.13	108.15 \pm 0.43	106.92
	ant	92.15 \pm 1.80	94.62 \pm 3.44	98.59 \pm 1.67	111.45 \pm 3.09	120.36 \pm 1.25	130.75
cloned+expert	pen	82.77 \pm 4.84	56.05 \pm 7.20	11.30 \pm 2.64	10.33 \pm 2.27	10.13 \pm 2.90	106.42
	door	40.58 \pm 55.52	80.99 \pm 45.33	102.77 \pm 0.96	100.08 \pm 2.59	86.06 \pm 6.14	103.94
	hammer	88.63 \pm 33.94	94.88 \pm 16.51	94.59 \pm 9.39	100.59 \pm 10.60	90.27 \pm 18.70	125.71
human+expert	pen	99.27 \pm 5.22	101.27 \pm 6.45	95.77 \pm 8.91	96.14 \pm 7.88	99.96 \pm 13.06	106.42
	door	99.90 \pm 1.90	102.22 \pm 1.42	100.77 \pm 1.68	99.43 \pm 2.36	101.22 \pm 2.95	103.94
	hammer	87.11 \pm 16.28	102.10 \pm 15.65	93.34 \pm 7.41	97.37 \pm 15.52	93.39 \pm 7.70	125.71
partial+expert	kitchen	49.80 \pm 14.81	61.10 \pm 3.24	57.75 \pm 2.00	63.00 \pm 3.33	59.70 \pm 5.28	75.0
mixed+expert	kitchen	46.75 \pm 1.63	45.80 \pm 2.65	47.92 \pm 1.23	46.85 \pm 1.80	45.40 \pm 3.84	75.0

Table 7: Average normalized return over last 10 evaluations of IOSTOM with different β values on the D4RL suboptimal datasets with 1 expert trajectory. Method with the best avg. perf is in **bold**.

711 This section presents an ablation study to evaluate the impact of the hyperparameter β on IOSTOM's
712 performance. Table 7 summarizes these results. For locomotion tasks (e.g., Hopper, HalfCheetah,
713 Walker2d, Ant), higher β values, typically 15 or 20, generally yield superior scores compared to lower
714 values such as 3 or 5. Conversely, for manipulation tasks (e.g., Pen, Door, Hammer, Kitchen), optimal

715 performance is often achieved with β values of 5 or 10. However, the performance differences across
716 various β settings for these tasks are less pronounced. The only exception is the ‘pen’ environment
717 within the ‘cloned+expert’ dataset, where decreasing β leads to improved results, with $\beta = 3$
718 achieving the highest score.