

Appendix

A SURROGATE DEMONSTRATOR CLASSIFICATION TRAINING

We provide more details about the algorithm and theory of Surrogate Demonstrator Classification training.

Algorithm 3 Learning scoring model

```

1: Input:  $\{(i, D_i)\}_{i=1}^m, \{\alpha_i\}_{i=1}^m$ , and  $\alpha'$ 
2: Initialization: scoring model  $f_\phi(s, a)$ 
3: Compute of  $T(\cdot) = [T_1(\cdot), \dots, T_m(\cdot)]$  using Eq. 5 based on  $\{\alpha_i\}_{i=1}^m$  and  $\alpha'$ 
4: for  $t = 1, 2, \dots$  do
5:   Sample examples:  $\bar{z}, (s, a) \sim \{(i, D_i)\}_{i=1}^m$ 
6:   Update  $f_\phi$  by minimizing the empirical risk from Eq. 6
7: end for
8: Return  $f_\phi$ 

```

We consider demonstrator heterogeneity, i.e., there $\exists i, j \in [m], i \neq j$, such that $\alpha_i \neq \alpha_j$. According to Lu et al. (2021), $f_{\phi_{\text{sur}}^*}$ that recovered by removing transformation layer $T(\cdot)$ of the optimal surrogate classifier f_{ϕ^*} (the minimizer of $\mathbb{E}_{(s,a) \sim \rho} [\ell(\bar{f}_\phi(s, a), \bar{z})] = \mathbb{E}_{(s,a) \sim \rho} [\ell(T(f_\phi(s, a)), \bar{z})]$) is identical to the optimal classifier f_{ϕ^*} that minimize the risk $\mathbb{E}_{(s,a) \sim \rho} [\ell(f_\phi(s, a), z)]$, which is not possible if the heterogeneity setting is invalid. Thus, through the transformation function transformation $T(\cdot)$, the optimality scoring model f_{ϕ^*} can be learned by minimizing the surrogate classification risk in Eq. 6.

B PROOF OF THEOREM 1

Proof. We first prove that the sequence $\{\mathcal{L}(\phi^t, \alpha^t)\}$ is non-increasing. If $\mathcal{L}(\phi, \alpha)$ is lower-bounded (e.g., by zero in many loss formulations), then by the monotone convergence theorem, the sequence must converge.

$$\mathcal{L}(\phi, \alpha) = \mathbb{E}_{(s,a) \sim \rho} [\ell(T_\alpha(f_\phi(s, a)), \bar{z})] \quad (9)$$

Minimizing the cross-entropy loss in Eq. 9 is equivalent to maximizing a log likelihood (Shangnan & Wang, 2021). We define a log likelihood function through the conditional distribution $p(\bar{z}|(s, a))$ in the form of discriminative training. With scoring model parameters and hidden parameters of expertise levels, the log likelihood (ll) can be formulated as

$$ll(\phi, \alpha) := \ln p(\bar{z}|(s, a), \phi) = \sum_{\alpha} p(\alpha) \ln p(\bar{z}|(s, a), \phi) \quad (10)$$

To prove that $\{\mathcal{L}(\phi^t, \alpha^t)\}$ is monotonically non-increasing, we only have to prove $\{ll(\phi^t, \alpha^t)\}$ is monotonically non-decreasing, i.e., $ll(\phi^{t+1}, \alpha^{t+1}) \geq ll(\phi^t, \alpha^t)$.

We define $L(q, \phi) := \sum_{\alpha} q(\alpha) \ln \frac{p(\bar{z}, \alpha|(s, a), \phi)}{q(\alpha)}$, $q := q(\alpha)$, and $p := p(\alpha|\bar{z}, (s, a), \phi)$

$$\begin{aligned}
L(q, \phi) &= \sum_{\alpha} q(\alpha) \ln \frac{p(\bar{z}, \alpha|(s, a), \phi)}{q(\alpha)} \\
&= \sum_{\alpha} q(\alpha) [\ln p(\alpha|\bar{z}, (s, a), \phi) + \ln p(\bar{z}|(s, a), \phi) - \ln q(\alpha)] \\
&= \sum_{\alpha} q(\alpha) \ln \frac{p(\alpha|\bar{z}, (s, a), \phi)}{q(\alpha)} + \sum_{\alpha} q(\alpha) \ln p(\bar{z}|(s, a), \phi) \\
&= -KL(q||p) + \ln p(\bar{z}|(s, a), \phi)
\end{aligned} \quad (11)$$

Then, the likelihood $ll(\phi, \alpha)$ can be represented as

$$\begin{aligned} \ln p(d|(s, a), \phi) &= L(q, \phi) + KL(q||p), \\ &\geq L(q, \phi) \end{aligned} \quad (12)$$

Since $KL(q||p) \geq 0$, $L(q, \phi)$ is the lower bound of $ll(\phi, \alpha)$.

For likelihood function $ll(\phi^t, \alpha^t)$ at iteration t ,

$$\begin{aligned} ll(\phi^t, \alpha^t) &= \ln p(\bar{z}|(s, a), \phi^t) \\ &= L(q(\alpha^t), \phi^t) + KL(q(\alpha^t)||p(\alpha|\bar{z}, (s, a), \phi^t)) \end{aligned} \quad (13)$$

Since $L(q(\alpha^t), \phi^t) = \ln p(\bar{z}|(s, a), \phi^t) - KL(q(\alpha^t)||p(\alpha|\bar{z}, (s, a), \phi^t))$ is the lower bound $ll(\phi^t, \alpha^t)$. We update α^t by letting $q(\alpha^t) = p(\alpha|\bar{z}, (s, a), \phi^t)$ to maximize the lower bound, where $p(\alpha|\bar{z}, (s, a), \phi^t)$ can be calculated based on the second equation in Eq.7. Thus, $KL(q(\alpha^{t+1})||p(\alpha|\bar{z}, (s, a), \phi^t)) = 0$. Thus, $ll(\phi^t, \alpha^{t+1})$ can be expressed as

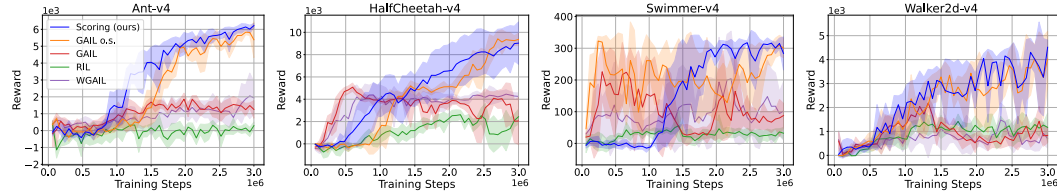
$$\begin{aligned} ll(\phi^t, \alpha^{t+1}) &= \ln p(\bar{z}|(s, a), \phi^t) \\ &= L(q(\alpha^{t+1}), \phi^t) + KL(q(\alpha^{t+1})||p(\alpha|\bar{z}, (s, a), \phi^t)) \\ &= L(q(\alpha^{t+1}), \phi^t) \end{aligned} \quad (14)$$

By updating α , we increase the lower bound of $ll(\phi^t, \alpha^{t+1})$ equals to $\ln p(\bar{z}|(s, a), \phi^t)$. Then we maximize $ll(\phi^t, \alpha^{t+1})$ with regard to ϕ using a gradient decent method, $ll(\phi^{t+1}, \alpha^{t+1})$ must be not smaller than $ll(\phi^t, \alpha^{t+1})$. Finally, we have $ll(\phi^{t+1}, \alpha^{t+1}) \geq ll(\phi^t, \alpha^{t+1}) = \ln p(\bar{z}|(s, a), \phi^t) = ll(\phi^t, \alpha^t)$

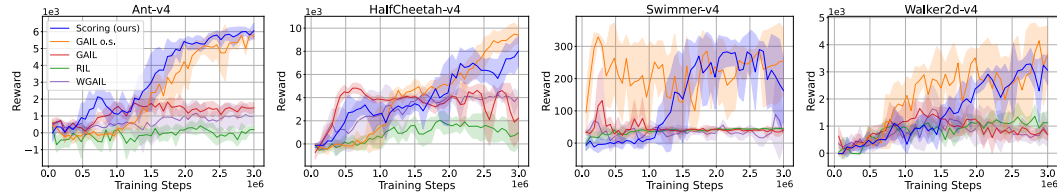
□

C EXPERIMENT DETAILS

C.1 DETAILED RESULTS



(a) General expertise test



(b) Low expertise test

Figure 3: Performance comparison (mean \pm std, shown as shaded region) with baselines. Supplementary results for Sec. 4.2

Table 5: Performance (reward) of checkpoints used as optimal and suboptimal policies across environments. Values are reported as mean \pm standard deviation.

Environment	Optimal policy π^*	Suboptimal policy π^{sub}
Ant-v4	6766.26 \pm 109.70	1266.37 \pm 533.96
HalfCheetah-v4	15947.53 \pm 39.08	4030.23 \pm 169.82
Swimmer-v4	339.80 \pm 1.15	41.60 \pm 2.34
Walker-v4	5638.94 \pm 49.76	554.38 \pm 267.99

Table 6: True expertise levels $\{\alpha_i\}_{i=1}^m$ for different numbers of demonstrators across two tests. Expertise levels are uniformly distributed within each range.

Demonstrator	1	2	3	4	5	6	7	8
General expertise test								
2 demonstrators	0.100	0.900	–	–	–	–	–	–
4 demonstrators	0.100	0.366	0.633	0.900	–	–	–	–
6 demonstrators	0.100	0.260	0.420	0.580	0.740	0.900	–	–
8 demonstrators	0.100	0.214	0.329	0.443	0.557	0.671	0.786	0.900
Low expertise test								
2 demonstrators	0.050	0.150	–	–	–	–	–	–
4 demonstrators	0.050	0.083	0.117	0.150	–	–	–	–
6 demonstrators	0.050	0.070	0.090	0.110	0.130	0.150	–	–
8 demonstrators	0.050	0.064	0.079	0.093	0.107	0.121	0.136	0.150

Table 7: Estimated expertise levels $\{\hat{\alpha}_i\}_{i=1}^m$, showing the mean estimate and the mean \pm standard deviation of error across five random seeds. True values are shown separately. Supplementary results for Sec. 4.3

Demonstrator	1	2	3	4	5	6	Mean error \pm std
General expertise test							
Ant-v4	0.100	0.260	0.420	0.580	0.740	0.900	$(2.67 \pm 1.28) \times 10^{-4}$
HalfCheetah-v4	0.100	0.260	0.420	0.580	0.740	0.900	$(2.00 \pm 2.00) \times 10^{-5}$
Hopper-v4	0.101	0.259	0.418	0.575	0.734	0.892	$(3.83 \pm 2.73) \times 10^{-3}$
Swimmer-v4	0.100	0.260	0.420	0.580	0.740	0.900	$(2.00 \pm 2.00) \times 10^{-5}$
Walker2d-v4	0.100	0.260	0.420	0.581	0.741	0.900	$(3.27 \pm 3.12) \times 10^{-4}$
True expertise levels	0.100	0.260	0.420	0.580	0.740	0.900	
Low expertise test							
Ant-v4	0.148	0.170	0.187	0.207	0.222	0.246	$(9.65 \pm 0.24) \times 10^{-2}$
HalfCheetah-v4	0.086	0.105	0.127	0.146	0.167	0.190	$(3.67 \pm 0.15) \times 10^{-2}$
Hopper-v4	0.156	0.176	0.195	0.216	0.231	0.257	$(1.05 \pm 0.02) \times 10^{-1}$
Swimmer-v4	0.077	0.100	0.118	0.139	0.158	0.177	$(2.81 \pm 0.11) \times 10^{-2}$
Walker2d-v4	0.179	0.206	0.219	0.248	0.263	0.287	$(1.34 \pm 0.04) \times 10^{-1}$
True expertise levels	0.050	0.070	0.090	0.110	0.130	0.150	

Table 8: Classification / Labeling results (mean \pm std). Stage 1: labeling using the pretrained scoring model. Stage 2: relabeling using the fine-tuned scoring model. TP (true positive), TN (true negative), FP (false positive), FN (false negative). Improvement (%) shows the Stage 1 \rightarrow Stage 2 percentage change of accuracy and precision after relabeling. Supplementary results for Sec. 4.4

Environment	Stage	TP	TN	FP	FN	Accuracy / Precision	Improvement (%)
General expertise test							
Ant-v4	1	14988.6 \pm 3.9	14984.2 \pm 10.1	15.8 \pm 10.1	11.4 \pm 3.9	0.9991 / 0.9989	-
	2	14984.0 \pm 5.5	14984.0 \pm 5.5	16.0 \pm 5.5	16.0 \pm 5.5	0.9989 / 0.9989	-0.02 / 0.00
HalfCheetah-v4	1	15000.0 \pm 0.0	14999.4 \pm 0.8	0.6 \pm 0.8	0.0 \pm 0.0	1.0000 / 1.0000	-
	2	15000.0 \pm 0.0	15000.0 \pm 0.0	0.0 \pm 0.0	0.0 \pm 0.0	1.0000 / 1.0000	0.00 / 0.00
Swimmer-v4	1	15000.0 \pm 0.0	14999.4 \pm 1.2	0.6 \pm 1.2	0.0 \pm 0.0	1.0000 / 1.0000	-
	2	14991.4 \pm 16.2	14991.4 \pm 16.2	8.6 \pm 16.2	8.6 \pm 16.2	0.9994 / 0.9994	-0.06 / -0.06
Walker2d-v4	1	14992.0 \pm 2.2	14985.0 \pm 6.1	15.0 \pm 6.1	8.0 \pm 2.2	0.9992 / 0.9990	-
	2	14990.4 \pm 1.6	14990.4 \pm 1.6	9.6 \pm 1.6	9.6 \pm 1.6	0.9994 / 0.9994	0.02 / 0.04
Low expertise test							
Ant-v4	1	2997.6 \pm 1.5	24101.4 \pm 735.5	2898.6 \pm 735.5	2.4 \pm 1.5	0.9033 / 0.5178	-
	2	2947.0 \pm 32.9	26947.0 \pm 32.9	53.0 \pm 32.9	53.0 \pm 32.9	0.9965 / 0.9823	10.32 / 89.71
HalfCheetah-v4	1	3000.0 \pm 0.0	25898.2 \pm 355.8	1101.8 \pm 355.8	0.0 \pm 0.0	0.9633 / 0.7368	-
	2	2997.4 \pm 5.2	26997.4 \pm 5.2	2.6 \pm 5.2	2.6 \pm 5.2	0.9998 / 0.9991	3.79 / 35.60
Swimmer-v4	1	3000.0 \pm 0.0	26156.2 \pm 393.5	843.8 \pm 393.5	0.0 \pm 0.0	0.9719 / 0.7884	-
	2	2637.2 \pm 290.4	26637.2 \pm 290.4	362.8 \pm 290.4	362.8 \pm 290.4	0.9758 / 0.8791	0.40 / 11.50
Walker2d-v4	1	2965.0 \pm 42.7	22953.2 \pm 890.1	4046.8 \pm 890.1	35.0 \pm 42.7	0.8639 / 0.4301	-
	2	2772.0 \pm 170.3	26772.0 \pm 170.3	228.0 \pm 170.3	228.0 \pm 170.3	0.9848 / 0.9240	13.99 / 114.83

Table 9: Ablation results on the impact of relabeling and early stopping in Stage 2 on Ant-v4 task. Reported as mean \pm std of reward across last five checkpoints. Supplementary results for Sec. 4.5

Method	Expertise [0.1 – 0.9]	Expertise [0.05 – 0.15]
Scoring (ours)	6084.6 \pm 146.9	5932.0 \pm 335.2
Scoring (no relabel)	5593.0 \pm 564.2	3316.2 \pm 1207.8
Scoring (no early stop)	2129.8 \pm 2205.8	1018.2 \pm 250.0

Table 10: Ablation study on top-k hyperparameter on Ant-v4 task. Reported as mean \pm std of reward across last five checkpoints. Supplementary results for Sec. 4.5

Top-k fraction	Expertise [0.1 – 0.9]	Expertise [0.05 – 0.15]
0.05	–	5887.6 \pm 263.8
0.1	5403.8 \pm 378.7	5932.0 \pm 335.2
0.15	–	3663.4 \pm 1950.9
0.2	–	1756.6 \pm 902.6
0.3	5809.0 \pm 274.8	–
0.5	6084.6 \pm 146.9	–
0.55	4109.2 \pm 2204.9	–
0.6	3641.0 \pm 1237.7	–
0.7	2383.0 \pm 1059.7	–

Table 11: Estimated expertise levels $\{\hat{\alpha}_i\}_{i=1}^m$ on Ant-v4 task, showing the mean estimate and the mean \pm standard deviation of error across five random seeds. True values are shown separately. Results shown for different numbers of demonstrators (2, 4, 6, 8). Supplementary results for Sec. 4.5

Demonstrator	Type	1	2	3	4	5	6	7	8	Mean error \pm std
General expertise test										
2 demonstrators	Est.	0.101	0.899	-	-	-	-	-	-	$(9.60 \pm 1.60) \times 10^{-4}$
	True	0.100	0.900	-	-	-	-	-	-	-
4 demonstrators	Est.	0.100	0.366	0.633	0.899	-	-	-	-	$(4.50 \pm 2.96) \times 10^{-4}$
	True	0.100	0.367	0.633	0.900	-	-	-	-	-
6 demonstrators	Est.	0.100	0.260	0.420	0.580	0.740	0.900	-	-	$(2.67 \pm 1.28) \times 10^{-4}$
	True	0.100	0.260	0.420	0.580	0.740	0.900	-	-	-
8 demonstrators	Est.	0.100	0.214	0.329	0.443	0.557	0.671	0.785	0.899	$(2.50 \pm 1.97) \times 10^{-4}$
	True	0.100	0.214	0.329	0.443	0.557	0.671	0.786	0.900	-
Low expertise test										
2 demonstrators	Est.	0.193	0.287	-	-	-	-	-	-	$(1.40 \pm 0.03) \times 10^{-1}$
	True	0.050	0.150	-	-	-	-	-	-	-
4 demonstrators	Est.	0.106	0.143	0.173	0.208	-	-	-	-	$(5.77 \pm 0.16) \times 10^{-2}$
	True	0.050	0.083	0.117	0.150	-	-	-	-	-
6 demonstrators	Est.	0.148	0.170	0.187	0.207	0.222	0.246	-	-	$(9.65 \pm 0.24) \times 10^{-2}$
	True	0.050	0.070	0.090	0.110	0.130	0.150	-	-	-
8 demonstrators	Est.	0.073	0.087	0.101	0.115	0.130	0.144	0.160	0.175	$(2.31 \pm 0.10) \times 10^{-2}$
	True	0.050	0.064	0.079	0.093	0.107	0.121	0.136	0.150	-

Table 12: Ablation on the different number of demonstrators for Ant-v4 task. Reported numbers are mean \pm std of reward across last five checkpoints. Supplementary results for Sec. 4.5

Scoring Samples	Expertise [0.1 – 0.9]	Expertise [0.05 – 0.15]
2	5634.0 \pm 617.8	6050.6 \pm 120.4
4	5934.0 \pm 264.3	5637.2 \pm 772.5
6	6084.6 \pm 146.9	5932.0 \pm 335.2
8	5735.0 \pm 225.6	6146.6 \pm 107.2

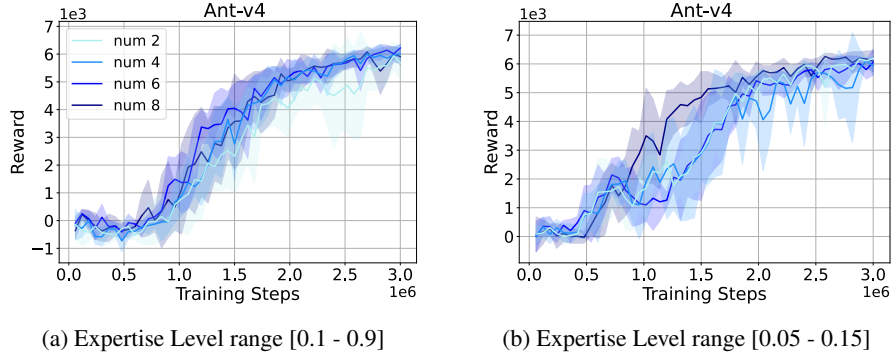


Figure 4: Ablation on the different number of demonstrators for Ant-v4 task (mean \pm std, shown as shaded region). Supplementary results for Sec. 4.5

Number of Demonstrators: The results in Tab. 11 show that the estimation error of expertise levels in low expertise test increase when the number of demonstrators is reduced. However, the reward obtained by the IL agent in Stage 2 (see Tab. 12) does not decrease substantially. These results demonstrate the robustness of our method with respect to varying demonstrator counts in terms of IL performance.

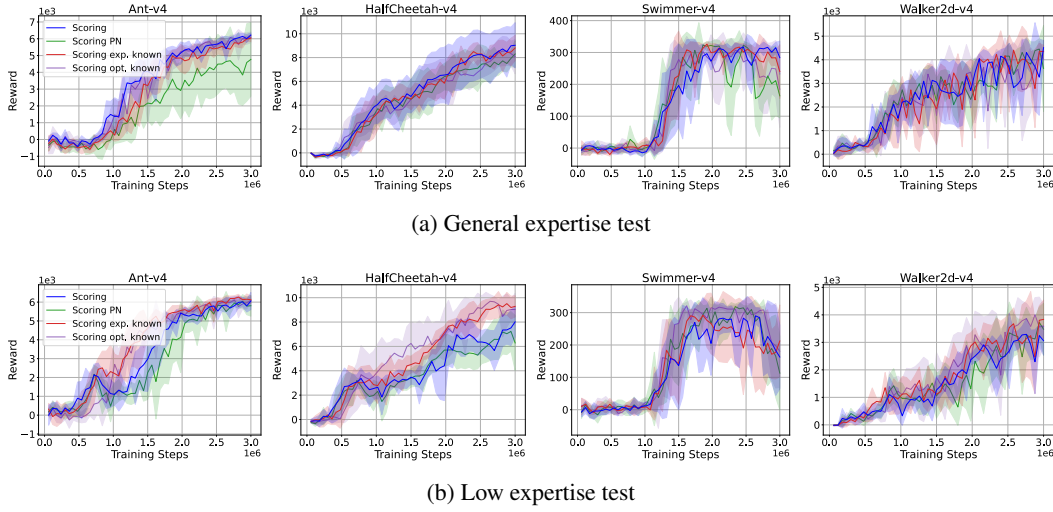


Figure 5: Comparative analysis of our method and its variants.

C.2 ADDITIONAL ANALYSIS OF METHOD VARIANTS

We analyze our algorithm by comparing it against several variants under different settings. (1) **Scoring**: original method; (2) **Scoring PN**: in Stage 2, the first term $\hat{\mathbb{E}}_{(s,a) \sim \rho} [\mathcal{L}(f_\phi(s, a))]$ (demonstration discriminative loss) in Eq. 8 is replaced with a PN loss $\hat{\mathbb{E}}_{(s,a) \sim \hat{\rho}^{\pi^*}} [\mathcal{L}_{PN}(f_\phi(s, a, s', a'))]$, where $\mathcal{L}_{PN}(f_\phi(s, a, s', a')) = \log(1 - f_\phi(s, a)) + \log(f_\phi(s', a'))$, and π^{sub} represent the pseudo-suboptimal data inferred from $D_{\text{subopt}} = D \setminus D_{\text{opt}}$. (3) **Scoring exp. known**: the true expertise levels are assumed to be given in both Stage 1 and 2. (4) **Scoring opt. known**: in Stage 2, D_{opt} is provided based on the true labels.

The results in Fig. 5 show that, overall, all methods achieve comparable performance. **Scoring PN** yields slightly lower rewards in general expertise test on Ant and Swimmer tasks, which may be attributed to the overconfidence issue introduced by self-labeling when using the PN loss. This result highlights the effectiveness of our demonstration discriminative loss. **Scoring exp. known** and **Scoring opt. known** serve as oracle variants (with oracle knowledge of expertise levels and optimality labels, respectively) and achieve only marginally better performance than the original method in the low expertise test on HalfCheetah and Walker2d. This observation highlights the effectiveness of our approach in expertise-level estimation and optimality prediction when learning from imperfect demonstrations.

C.3 REPRODUCIBILITY STATEMENT

To ensure the reproducibility of this work, details of the training parameters are provided in Tab. 13. All reported results are averaged over five random seeds (0–4) per condition/method, with mean and standard deviation reported. The corresponding code will be released as open-source upon publication of this work.

Table 13: Training Parameters

Stage	Parameter	Value
Stage 1	f_ϕ learning rate	1×10^{-4}
	f_ϕ batch size	1000
	f_ϕ optimizer	Adam
	Variance threshold ϵ for expertise levels	0.0005
	Expertise levels estimation frequency	Every 5 epochs
	f_ϕ training epochs	50 (Ant, HalfCheetah, Swimmer) 100 (Walker2d)
Stage 2	π_θ model	SAC (Stable-Baselines3)
	π_θ learning starts from	100 steps
	π_θ batch size	1024
	π_θ learning rate	2×10^{-3}
	π_θ replay buffer size	1×10^6
	π_θ training steps / IL iteration	1.5×10^4
	f_ϕ learning rate	1×10^{-4}
	f_ϕ batch size	1024
	f_ϕ optimizer	Adam
	f_ϕ updates epochs / IL iteration	500
	Δ_t, p for relabeling early stopping	30 (for 5 consecutive steps)
	IL total iterations	200
	checkpoints saving frequency	1 per IL iteration