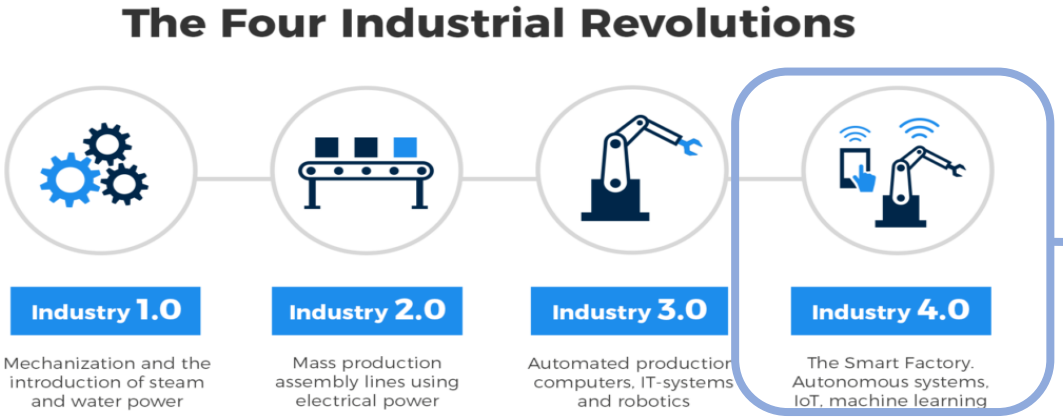
Real-time Update in Robot Imitation Learning

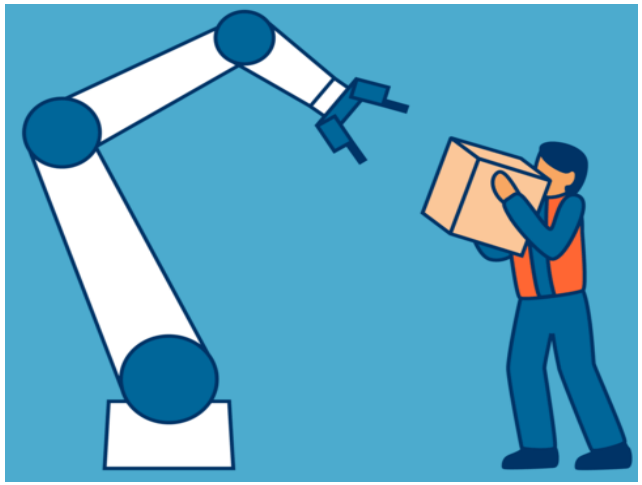
Outline

- Background
- Motivation
- Model 1 – Triple GAIL
- Model 2– Human-in-the-loop IL
- Conclusion

Background

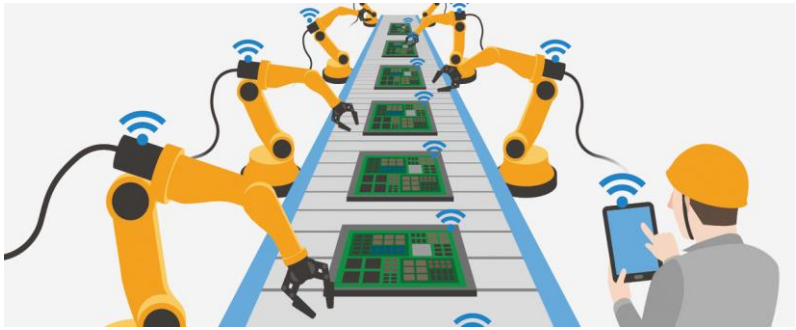
Our term project: the application of reinforcement learning model in Industry 4.0, mostly in human robot collaboration





Background

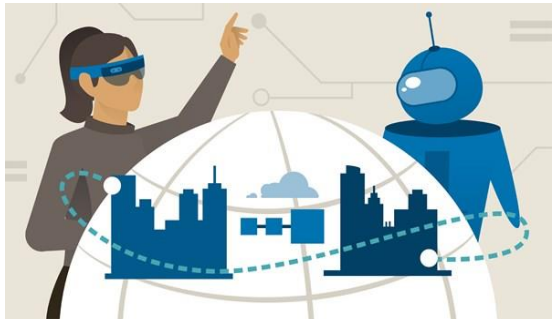
Expectation: robot being **able** to execute by themselves while human supervise



Solutions:

Conditional Programming

Imitation Learning



Q: How to make robot learn human's knowledge and skills?

Not for chaotic sites

Versatile

Today's presentation: technical progress in imitation learning model

Outline

- Background
- Motivation
- Model 1 – Triple GAIL
- Model 2– Human-in-the-loop IL
- Conclusion

Motivation

Imitation learning (IL): learning from demonstrations for complex robot manipulation skills

- **Cons 1:** models trained by IL can suffer from **covariate shift**.
  → re-label dataset samples (infeasible works for robotic tasks)
- **Cons 2**: models cannot recover failure cases
- In robot manipulation (6-DoF), **small inaccuracies** in the output actions can make IL agents **susceptible to making mistakes.**

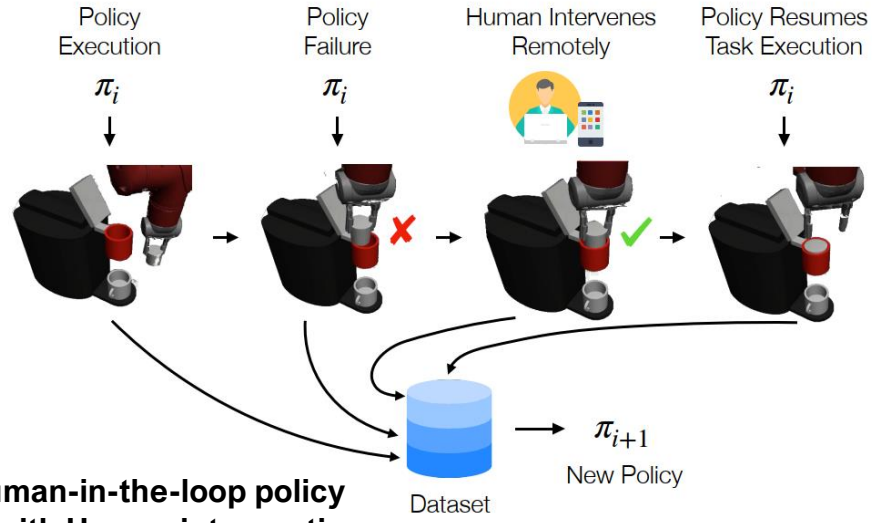


Fig. 1: Human-in-the-loop policy learning with Human interventions

Outline

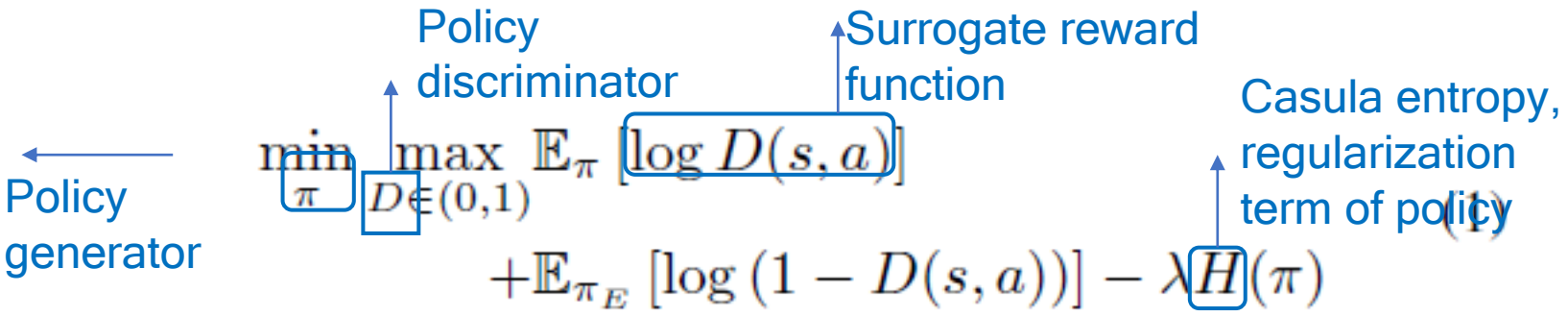
- Background
- Motivation
- Model 1 – Triple GAIL
- Model 2– Human-in-the-loop IL
- Conclusion

Imitation Learning

1. Behavior Cloning: replicate expert policy, large amount of data needed and cause compounding errors;
2. Inverse Reinforcement Learning: learns expert reward function and use this model to train new agent, indirect and slow
3. Generative Adversarial Imitation Learning (GAIL): optimizes a policy directly from expert demonstrations without estimating the corresponding reward function, reduced compounding error

GAIL – Notation and Intro

- **Generator:** serves as a policy to imitate expert behavior by matching the state-action (s, a) distribution of demonstrations.
- **Discriminator:** plays a role of surrogate reward to measure the similarity between the **generated data** and **demonstration data**.



Model update: trust region policy optimization (TRPO)

GAIL – Notation and Intro

- Training Architecture

•The discriminator used the same architecture as the generator (policy).

•Two hidden layers of 100 units each with *tanh* nonlinearities in between.

•The networks were always initialized randomly at the start.

- Model update: trust region policy optimization (TRPO)

Triple GAIL

Motivation:

1. Single modality of single GAIL models
2. Needs to involve real-time experience of agents

Triple GAIL – Letter Notation

C (superscript) - drawn from expert data

e (superscript) – from encoder

g (superscript) - from generator

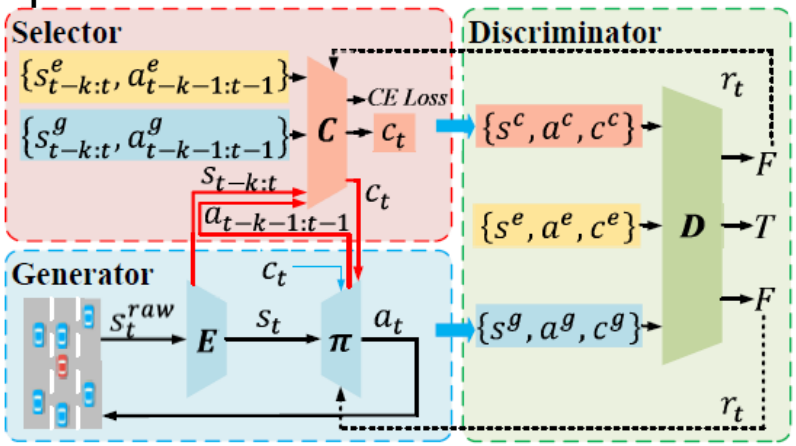
c (normal letter) - label of skills

a – action

s - state

t - time sequences

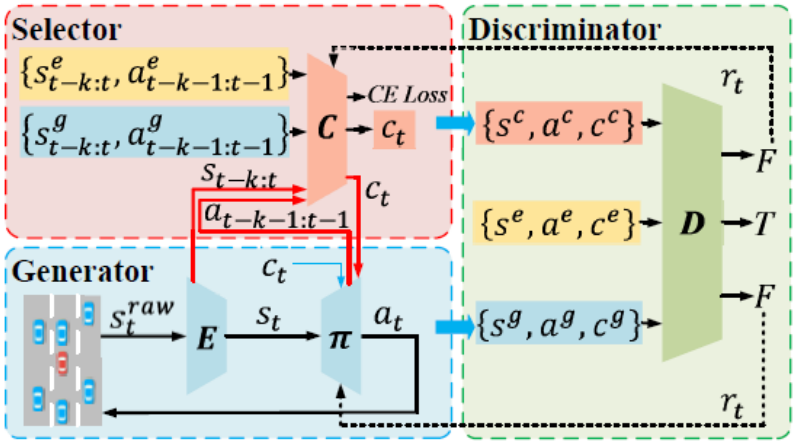
$p_{\pi E}$ - distribution of expert data



All these data sequences are sent to the discriminator together with expert data

Triple GAIL - Color Notation

- Blue line - drawn from expert data
- Red line – drawn from inference model while training
- Red block - conditional distribution of $p_{C\alpha}$
- Blue block - conditional distribution of $p_{\pi\theta}$

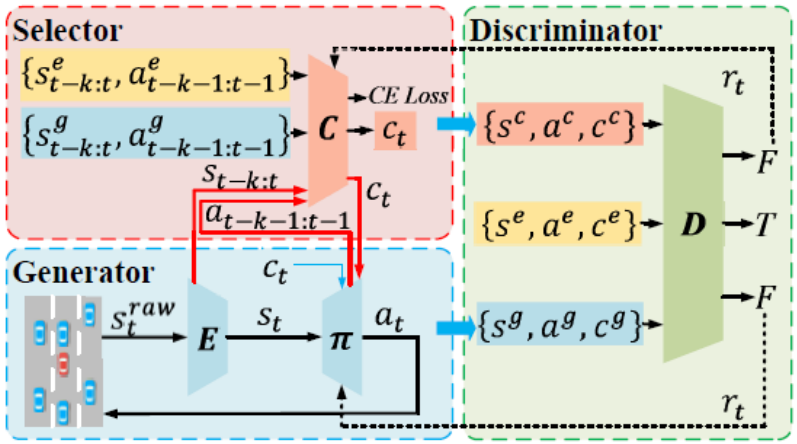


Triple GAIL

- Three-player game, the generator and the selector work cooperatively against the discriminator
- Min-max optimization:

$$\min_{\alpha,\theta}\max_{\psi}\mathbb{E}_{\pi_E}\left[\log\left(1-D_\psi(s,a,c)\right)\right] + \omega\mathbb{E}_{\pi_\theta}\left[\log D_\psi(s,a,c)\right] + (1-\omega)\mathbb{E}_{C_\alpha}\left[\log D_\psi(s,a,c)\right] - \lambda_H H\left(\pi_\theta\right) \tag{4}$$

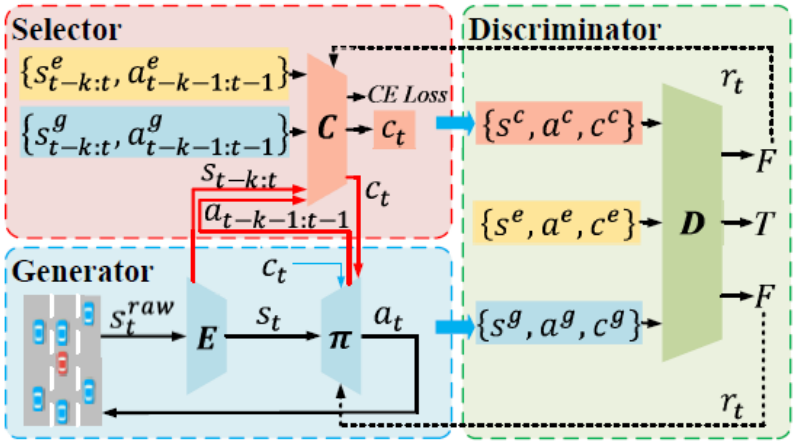
where $\mathbb{E}_{\pi_E}$, $\mathbb{E}_{\pi_\theta}$ and $\mathbb{E}_{C_\alpha}$ denote $\mathbb{E}_{(s,c,a)\sim p_{\pi_E}(s,c,a)}$, $\mathbb{E}_{(s,c,a)\sim p_{\pi_\theta}(s,c,a)}$, $\mathbb{E}_{(s,c,a)\sim p_{C_\alpha}(s,c,a)}$ respectively, $\omega \in (0,1)$ is a hyper-parameter that balances the weights of policy generation and skill selection, and $H(\pi_\theta)$ is the policy casual entropy defined as $\mathbb{E}_{\pi_\theta}[-\log \pi_\theta(a|s,a)]$ with hyper-parameter $\lambda_H > 0$.



Lemma 1. For any fixed generator and selector, the optimal form of the discriminator is denoted as:

$$D_{\psi^*} = \frac{\omega p_{\pi_\theta} + (1-\omega) p_{C_\alpha}}{p_{\pi_E} + \omega p_{\pi_\theta} + (1-\omega) p_{C_\alpha}} = \frac{p_\omega}{p_{\pi_E} + p_\omega} \quad (6)$$

where p_{π_E}, p_{π_θ} *and* p_{C_α} *denote* $p_{\pi_E}(s,a,c)$, $p_{\pi_\theta}(s,a,c)$ *and* $p_{C_\alpha}(s,a,c)$ *respectively, and* p_ω *is defined as* $\omega p_{\pi_\theta} + (1-\omega) p_{C_\alpha}$.



Triple GAIL

Lemma 2. The min-max game in Eqn. (4) can achieve the multiple equilibrium that $p_{\pi E} = p_w$ where variable w is a mixing coefficient between $p_{C\alpha}$ (s, a, c) and $p_{\pi\theta}$ (s, a, c) .

$$R_E = \mathbb{E}_{\pi_E}\left[-\log p_{C_\alpha}(c|s,a)\right]$$
$$\approx -\frac{1}{N}\sum_{i=0}^{N}\frac{1}{T}\sum_{t=1}^{T} c_{i,t}^{e}\log p_{C_\alpha}\left(c_{i,t}^{c}|s_{i,t}^{e}, a_{i,t-1}^{e}\right) \quad (7)$$

$$R_G = \mathbb{E}_{\pi_\theta}\left[-\log p_{C_\alpha}(c|s,a)\right]$$
$$\approx -\frac{1}{N}\sum_{i=0}^{N}\frac{1}{T}\sum_{t=1}^{T} c_{i,t}^{g}\log p_{C_\alpha}\left(c_{i,t}^{c}|s_{i,t}^{g}, a_{i,t-1}^{g}\right) \quad (8)$$

Triple GAIL

Theorem 1. Eqn. (5) ensures the existence and uniqueness of the global equilibrium, which is achieved if and only if

$p_{\pi\theta}$(s; a; c) = $p_{C\alpha}$ (s; a; c) = $p_{\pi E}$ (s; a; c).

$$\begin{aligned}\min_{\alpha,\theta}\max_{\psi}&\mathbb{E}_{\pi_E}\left[\log\left(1-D_\psi(s,a,c)\right)\right]\\&+\omega\mathbb{E}_{\pi_\theta}\left[\log D_\psi(s,a,c)\right]\\&+(1-\omega)\mathbb{E}_{C_\alpha}\left[\log D_\psi(s,a,c)\right]\\&+\lambda_E R_E+\lambda_G R_G-\lambda_H H\left(\pi_\theta\right)\end{aligned} \tag{5}$$

where λ_E and λ_G weigh the relative importance of two supervised loss.

Triple GAIL - Training Procedure

Algorithm 1 The Training Procedure of Triple-GAIL

Input: The multi-intention trajectories of expert τ_E;

Parameter: The initial parameters θ_0, α_0 and ψ_0

1: **for** $i = 0, 1, 2, \cdots$ **do**

2: **for** $j = 0, 1, 2, \cdots, N$ **do**

3: Reset environments by the demonstration episodes with fixed label c_j;

4: Run policy $\pi_\theta(\cdot|c_j)$ to sample trajectories: $\tau_{c_j} = (s_0, a_0, s_1, a_1, ...s_{T_j}, a_{T_j}|c_j)$

5: **end for**

6: Update the parameters of π_θ via TRPO with rewards: $r_{t_j} = -\log D_\psi(s_{t_j}, a_{t_j}, c_j)$

7: Update the parameters of D_ψ by gradient ascending with respect to:

$$\nabla_\psi \frac{1}{N_e}\sum_{n=1}^{N_e}\log(1-D_\psi(s_n^e, a_n^e, c_n^e)) + \frac{1}{N}\sum_{j=1}^{N}\left[\frac{\omega}{T_j}\sum_{t=1}^{T_j}\log D_\psi(s_t^g, a_t^g, c_j^g) + \frac{1-\omega}{T_j}\sum_{t=1}^{T_j}\log D_\psi(s_t^c, a_t^c, c_j^c)\right] \quad (9)$$

8: Update the parameters of C_α by gradient descending with respect to:

$$\nabla_\alpha \frac{1}{N}\sum_{j=1}^{N}\left[\frac{1-\omega}{T_j}\sum_{t=1}^{T_j}\log D_\psi(s_t^c, a_t^c, c_j^c) - \frac{\lambda_E}{T_j}\sum_{t=1}^{T_j}c_j^e \log p_{C_\alpha}(c_t^c|s_t^e, a_{t-1}^e) - \frac{\lambda_G}{T_j}\sum_{t=1}^{T_j}c_j^e \log p_{C_\alpha}(c_t^c|s_t^g, a_{t-1}^g)\right] \quad (10)$$

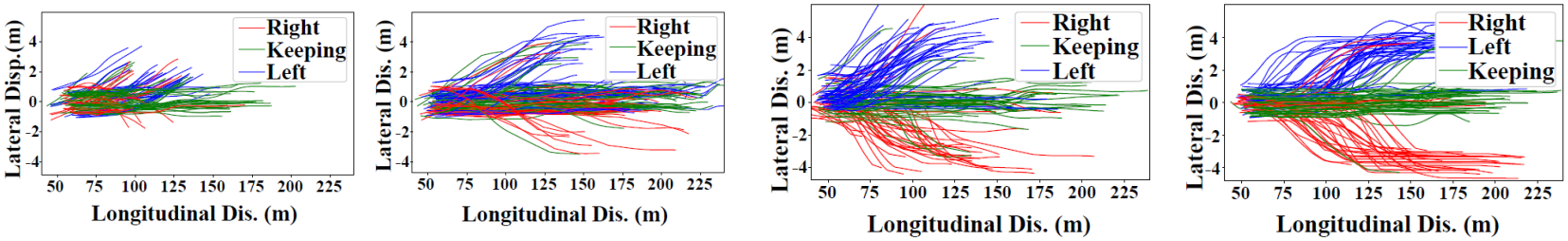
9: **end for**

Triple GAIL – Experiments and Results

Experiment: testing in driving learning and dense traffic driving

Algorithms	*Success Rate* (%)	*Mean Distance* (m)	*KL Divergence*		
			Lane-change Left	Lane-keeping	Lane-change Right
BC	6.8 ± 3.2	81.7 ± 3.1	3827 ± 358	4008 ± 486	2581 ± 371
GAIL	73.9 ± 1.3	168.4 ± 5.2	1764 ± 279	1893 ± 378	606 ± 278
CGAIL	65.5 ± 0.9	149.8 ± 7.2	1297 ± 255	1892 ± 279	977 ± 109
Triple-GAIL	$\mathbf{80.9 \pm 1.2}$	$\mathbf{179.6 \pm 3.6}$	$\mathbf{447 \pm 122}$	$\mathbf{685 \pm 214}$	$\mathbf{392 \pm 127}$
Expert	100	210 ± 2.1	0	0	0

TABLE I: The *Success Rate*, *Mean Distance* and *KL Divergence* of different algorithms.



(a) BC (b) GAIL (c) CGAIL (d) Triple-GAIL

Fig. 2: The visualization of trajectories. The trajectories of lane change right, lane keeping and lane-change left are represented by red, green and blue lines respectively. Dis. denotes displacement.

Triple GAIL – Conclusion and Advantages

- Solved single modality problem
- Improved model performance by adding real-time experience to the training

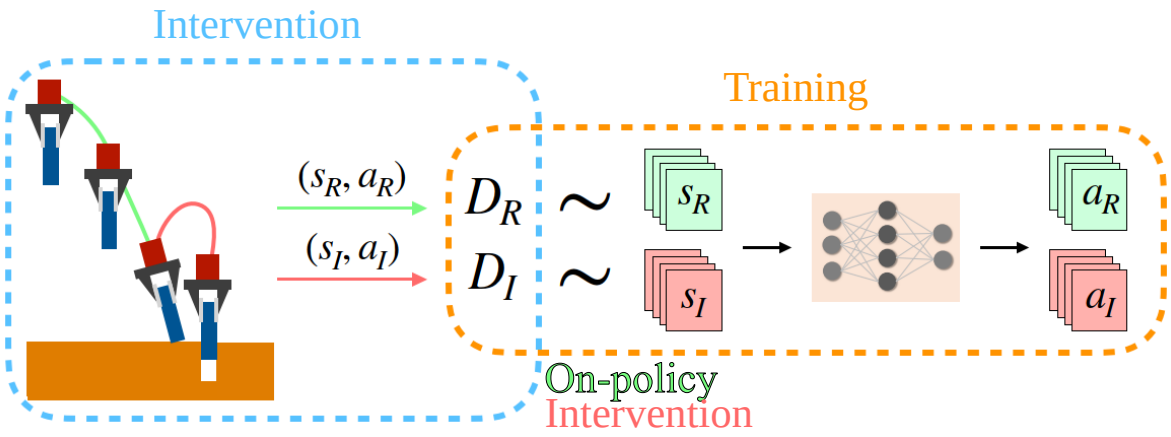
Outline

- Background
- Motivation
- Model 1 – Triple GAIL
- Model 2– Human-in-the-loop IL
- Conclusion

Outline – Model 2

- A. Intervention-based Policy learning
- B. Intervention-Weighted Regression (IWR)

A. Intervention-based Policy learning



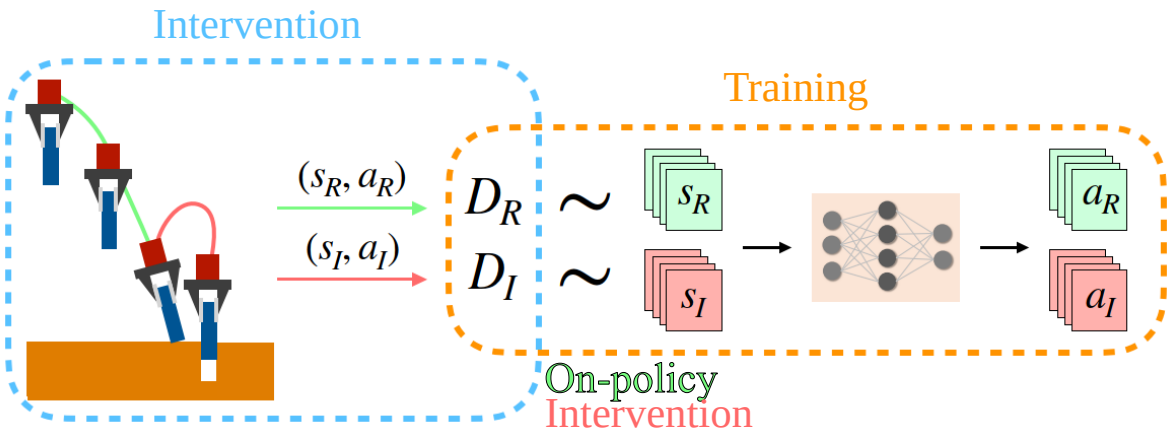
s (states), a (actions),
D (data samples),
R (Robot) and I (intervention)

Fig. 3: Intervention Weighted Regression.

- To sample equal size batches from the non-intervention and intervention datasets

  → **Re-weight** of the data distribution to reinforce intervention actions

A. Intervention-based Policy learning



s (states), a (actions),
D (data samples),
R (Robot) and I (intervention)

Fig. 3: Intervention Weighted Regression.

- Reinforcement learning objective:

$$J(\theta, q) = \mathbb{E}_{q(\tau)}\left[\log R(\tau) + \log p_{\pi_\theta}(\tau) - \log q(\tau)\right] \quad (1)$$

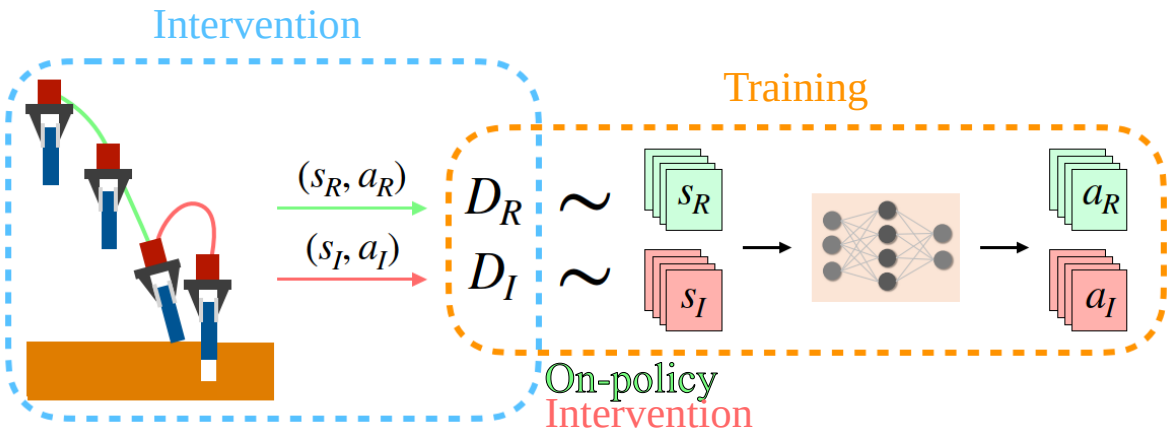
* Eq. (1) can be maximized via Expectation-Maximization.

- Human intervention:

$$q(\tau) = \arg\max \mathbb{E}_{q(\tau)}[\log R(\tau)] - KL\left[q(\tau) \,\|\, p_{\pi_\theta}(\tau)\right] \quad (2)$$

$$\theta = \arg\max \mathbb{E}_{(s,a)\sim q(\tau)}[\log \pi_\theta(a|s)] \quad (3)$$

A. Intervention-based Policy learning



s (states), a (actions),
D (data samples),
R (Robot) and I (intervention)

Fig. 3: Intervention Weighted Regression.

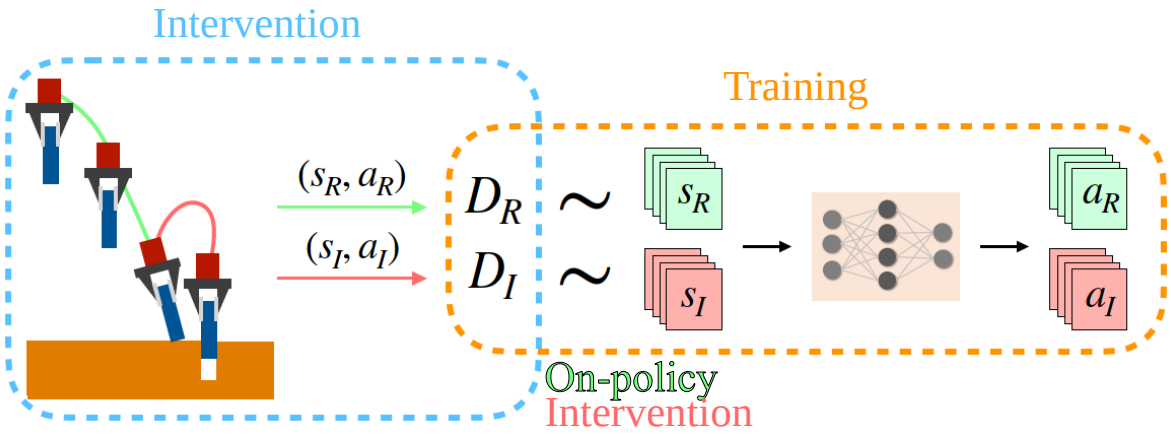
- Reinforcement learning objective to find a policy π_θ:

$$J(\theta, q) = \mathbb{E}_{q(\tau)}\left[\log R(\tau) + \log p_{\pi_\theta}(\tau) - \log q(\tau)\right] \quad (1)$$

θ: policy parameters
τ: a trajectory of states and actions
$q(\tau)$: dataset sample distribution
$p_{\pi_\theta}(\tau)$: the distribution of trajectories induced by policy π_θ

* Eq. (1) can be maximized via Expectation-Maximization.

A. Intervention-based Policy learning



s (states), a (actions),
D (data samples),
R (Robot) and I (intervention)

Fig. 3: Intervention Weighted Regression.

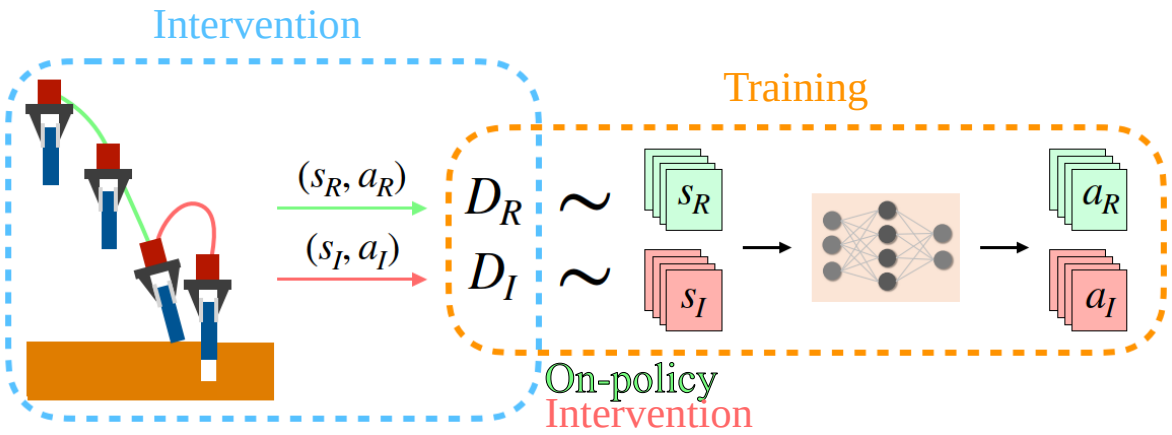
- Reinforcement learning objective to find a policy π_θ:

$$J(\theta, q) = \mathbb{E}_{q(\tau)}\left[\log R(\tau) + \log p_{\pi_\theta}(\tau) - \log q(\tau)\right] \quad (1)$$

- Human intervention:

$$q(\tau) = \arg\max \mathbb{E}_{q(\tau)}[\log R(\tau)] \; - KL\left[q(\tau) \parallel p_{\pi_\theta}(\tau)\right] \quad (2)$$

A. Intervention-based Policy learning



s (states), a (actions),
D (data samples),
R (Robot) and I (intervention)

Fig. 3: Intervention Weighted Regression.

- Reinforcement learning objective to find a policy π_θ:

$$J(\theta, q) = \mathbb{E}_{q(\tau)}\left[\log R(\tau) + \log p_{\pi_\theta}(\tau) - \log q(\tau)\right] \quad (1)$$

- Human intervention:

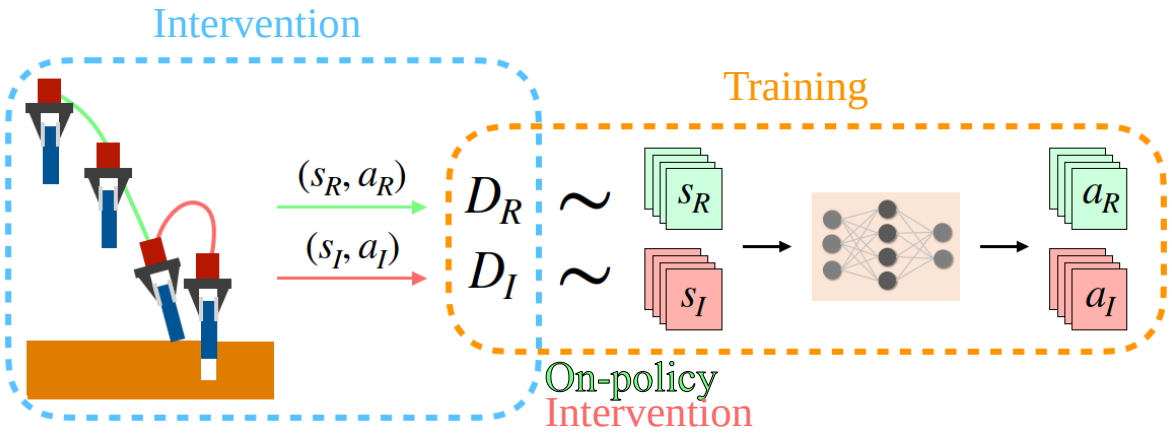
Dataset sample distribution

$$q(\tau) = \arg\max \mathbb{E}_{q(\tau)}[\log R(\tau)] - KL\left[q(\tau) \,\|\, p_{\pi_\theta}(\tau)\right] \quad (2)$$

- $R(\tau)$ ← Human tries to maximize return via interventions
- $KL\left[q(\tau) \,\|\, p_{\pi_\theta}(\tau)\right]$ ← On-policy regularization

*$KL[q \,\|\, p]$ denotes the Kullback-Leibler diverence between q (the variational trajectory distribution) and p (the one induced by the current policy).

A. Intervention-based Policy learning



s (states), a (actions),
D (data samples),
R (Robot) and I (intervention)

Fig. 3: Intervention Weighted Regression.

- Reinforcement learning objective to find a policy π_θ:

$$J(\theta, q) = \mathbb{E}_{q(\tau)}\left[\log R(\tau) + \log p_{\pi_\theta}(\tau) - \log q(\tau)\right] \quad (1)$$

- Human intervention:

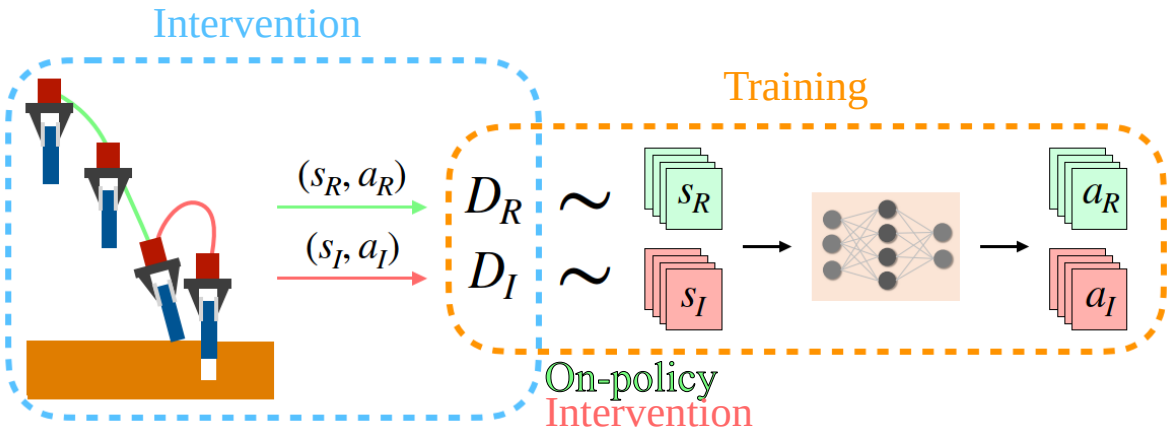
Dataset sample distribution

$$q(\tau) = \arg\max \mathbb{E}_{q(\tau)}[\log R(\tau)] - KL\left[q(\tau) \,\|\, p_{\pi_\theta}(\tau)\right] \quad (2)$$

Policy parameters

$$\theta = \arg\max \mathbb{E}_{(s,a)\sim q(\tau)}[\log \pi_\theta(a|s)] \quad (3)$$

A. Intervention-based Policy learning



s (states), a (actions),
D (data samples),
R (Robot) and I (intervention)

Fig. 3: Intervention Weighted Regression.

- Reinforcement learning objective to find a policy π_θ:

$$J(\theta, q) = \mathbb{E}_{q(\tau)}\left[\log R(\tau) + \log p_{\pi_\theta}(\tau) - \log q(\tau)\right] \quad (1)$$

- Human intervention:

Dataset sample distribution

$$q(\tau) = \arg\max \mathbb{E}_{q(\tau)}[\log R(\tau)] - KL\left[q(\tau) \,\|\, p_{\pi_\theta}(\tau)\right] \quad (2)$$

Policy parameters

$$\theta = \arg\max \mathbb{E}_{(s,a)\sim q(\tau)}[\log \pi_\theta(a|s)] \quad (3)$$

$$q^{i+1}(\tau) \leftarrow \arg\max_q J(\theta^i, q)$$

$$\theta^{i+1}(\tau) \leftarrow \arg\max_\theta J(\theta, q^{i+1}) = \arg\max \mathbb{E}_{(s,a)\sim q(\tau)}[\log \pi_\theta(a|s)]$$

B. Intervention Weighted Regression

$$q(\tau) = \arg\max \mathbb{E}_{q(\tau)}[\log R(\tau)] - KL\left[q(\tau) \parallel p_{\pi_\theta}(\tau)\right] \quad (2)$$

Optimization in Eq (2)

- Excluding on-policy samples in the dataset distribution can cause the policy trained in the next round to change substantially from the current policy.
- → Human-curated distribution $q(\tau)$ consists of a set of intervention data samples D_I and on-policy data samples collected by the robot D_R.

B. Intervention Weighted Regression

$$q(\tau) = \arg\max \mathbb{E}_{q(\tau)}[\log R(\tau)] - KL\left[q(\tau) \parallel p_{\pi_\theta}(\tau)\right] \quad (2)$$

- **Intervention Weighted Regression** (IWR): To assume that the human is specifying $q(\tau)$ by re-weighting the distribution of data with a parameter α.

$$q(s,a) \propto \alpha\rho_I(s,a) + \rho_R(s,a)$$

$\rho_I(s,a)$: the state-action distributions for the intervention
$\rho_R(s,a)$: the state-action distributions for the non-intervention

B. Intervention Weighted Regression

$$q(\tau) = \arg\max \mathbb{E}_{q(\tau)}[\log R(\tau)] \; - KL\big[q(\tau) \parallel p_{\pi_\theta}(\tau)\big] \qquad (2)$$

- **Intervention Weighted Regression** (IWR): To assume that the human is specifying $q(\tau)$ by re-weighting the distribution of data with a parameter α.

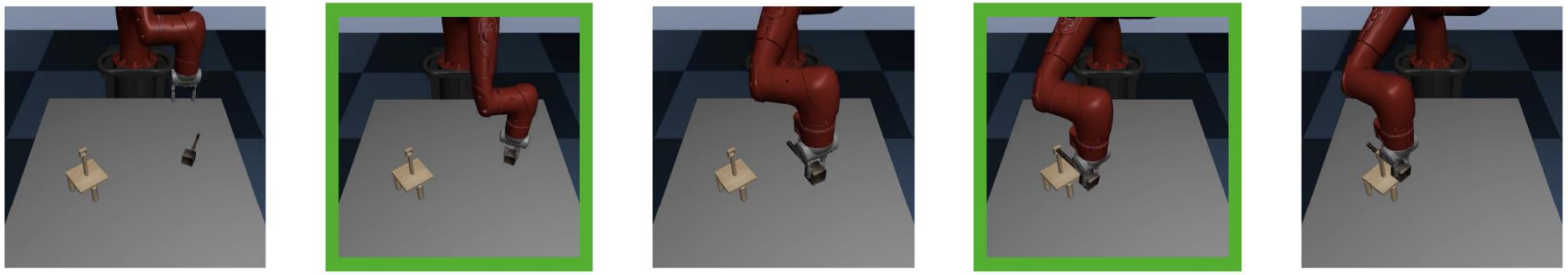
$$q(s,a) \; \propto \; \alpha\rho_I(s,a) \; + \rho_R(s,a)$$

$\rho_I(s,a)$: the state-action distributions for the intervention
$\rho_R(s,a)$: the state-action distributions for the non-intervention

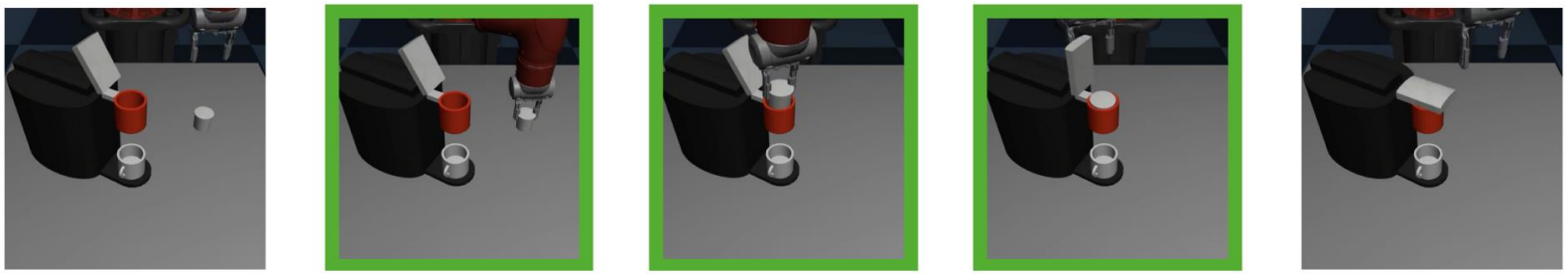
- In this paper, $q(s,a)$ samples from $\rho_I(s,a)$ and $\rho_R(s,a)$ are in equal proportion by choosing $\alpha = \; |D_R|/|D_I|$.

B. Intervention Weighted Regression

- Tasks: Threading and Coffee Machine



(a) Threading



(b) Coffee Machine

Baselines:

- **IWR** (This paper)
- **IWR-NB**: No dataset balancing takes place.
- **HG-Dagger**: Only the samples where the user was intervening are added to the dataset, while the policy samples are discarded.
- **Full Demos**: A human operator collects full task demonstrations instead of interventions

B. Intervention Weighted Regression

- Experiment setting: initial dataset (30 task demonstrations) and all policies (2-layer LSTMs)

TABLE I: Single-Operator Results on the Threading Task

Model	Round 1	Round 2	Final
Base	-	-	58.0 ± 9.2
Full Demos	-	-	76.7 ± 2.3
HG-DAGGER	57.3 ± 9.5	62.7 ± 5.0	75.3 ± 8.1
IWR-NB	76.0 ± 6.9	72.0 ± 3.5	74.7 ± 1.2
IWR (Ours)	$\mathbf{84.0 \pm 5.3}$	$\mathbf{90.7 \pm 3.1}$	$\mathbf{87.3 \pm 5.0}$

TABLE II: Multi-Operator Results on the Coffee Machine Task

Model	Round 1	Round 2	Final
Base	-	-	52.0 ± 3.5
Full Demos	-	-	64.9 ± 8.3
HG-DAGGER	70.2 ± 15.3	71.1 ± 9.7	69.6 ± 10.1
IWR (Ours)	$\mathbf{79.6 \pm 8.9}$	$\mathbf{79.5 \pm 11.7}$	$\mathbf{87.5 \pm 9.4}$

- **IWR vs Full Demos:** Intervention-based data collection can produce higher quality policies with fewer human-annotated samples.
- **IWR vs HG-DAGGER**: the method IWR is able to leverage intervention data to outperform the Full Demos baseline.
- ➔ **IWR** intelligently leverage both the human intervention and non-intervention samples for learning.

B. Intervention Weighted Regression

- Experiment setting: initial dataset (30 task demonstrations) and all policies (2-layer LSTMs)

TABLE I: Single-Operator Results on the Threading Task

Model	Round 1	Round 2	Final
Base	-	-	58.0 ± 9.2
Full Demos	-	-	76.7 ± 2.3
HG-DAGGER	57.3 ± 9.5	62.7 ± 5.0	75.3 ± 8.1
IWR-NB	76.0 ± 6.9	72.0 ± 3.5	74.7 ± 1.2
IWR (Ours)	$\mathbf{84.0 \pm 5.3}$	$\mathbf{90.7 \pm 3.1}$	$\mathbf{87.3 \pm 5.0}$

TABLE II: Multi-Operator Results on the Coffee Machine Task

Model	Round 1	Round 2	Final
Base	-	-	52.0 ± 3.5
Full Demos	-	-	64.9 ± 8.3
HG-DAGGER	70.2 ± 15.3	71.1 ± 9.7	69.6 ± 10.1
IWR (Ours)	$\mathbf{79.6 \pm 8.9}$	$\mathbf{79.5 \pm 11.7}$	$\mathbf{87.5 \pm 9.4}$

- **IWR vs Full Demos:** Intervention-based data collection can produce higher quality policies with fewer human-annotated samples.
- **IWR vs HG-DAGGER**: the method IWR is able to leverage intervention data to outperform the Full Demos baseline.
- ➔ **IWR** intelligently leverage both the human intervention and non-intervention samples for learning.

B. Intervention Weighted Regression

- Experiment setting: initial dataset (30 task demonstrations) and all policies (2-layer LSTMs)

TABLE III: Single-Operator Comparison across Final Threading Datasets Collected by Each Method

	Final Dataset		
Model	HG-DAGGER	IWR-NB	IWR (Ours)
HG-DAGGER	75.3 ± 8.1	72.0 ± 5.3	81.3 ± 4.2
IWR-NB	80.0 ± 1.4	74.7 ± 1.2	86.0 ± 4.0
IWR (Ours)	$\mathbf{87.3 \pm 6.4}$	$\mathbf{84.7 \pm 6.4}$	$\mathbf{87.3 \pm 5.0}$

TABLE IV: Multi-Operator Comparison across Final Coffee Machine Datasets Collected by Each Method

	Final Dataset	
Model	HG-DAGGER	IWR (Ours)
HG-DAGGER	69.6 ± 10.1	71.6 ± 16.1
IWR (Ours)	$\mathbf{85.6 \pm 6.5}$	$\mathbf{87.5 \pm 9.4}$

- **IWR** method consistently outperforms other baselines on their collected datasets
- **IWR** method can reach a level of performance close to its own collected dataset.
- ➔ Other baselines do not fail purely due to lower quality data or wore base policies at each iteration, but due to the way they leverage the data.

Conclusion

Contribution

- 1. **Development of a system** that enables remote teleoperation **for 6-DoF** robot control
- 2. **Introduction of Interaction Weighted Regression (IWR**): a simple but effective method to learn from human interventions that encourages the policy to learn how to traverse bottlenecks through the interventions.
- 3. Evaluation on **two challenging contact-rich manipulation tasks**.

Outline

- Background
- Motivation
- Model 1 – Triple GAIL
- Model 2– Human-in-the-loop IL
- Conclusion

Conclusion

- Solved single modality problem
- Improved model performance by adding real-time experience to the training

Reference

- Ho, J., & Ermon, S. (2016). Generative adversarial imitation learning. *arXiv preprint arXiv:1606.03476*.
- Fei, C., Wang, B., Zhuang, Y., Zhang, Z., Hao, J., Zhang, H., ... & Liu, W. (2020). Triple-GAIL: a multi-modal imitation learning framework with generative adversarial nets. *arXiv preprint arXiv:2005.10622*.
- Mandlekar, Ajay, et al. (2020). "Human-in-the-Loop Imitation Learning using Remote Teleoperation." *arXiv preprint arXiv:2012.06733.*
- Paine, Tom Le, et al. (2018)."One-shot high-fidelity imitation: Training large-scale deep nets with rl." *arXiv preprint arXiv:1810.05017.*
- Popov, Ivaylo, et al. (2017). "Data-efficient deep reinforcement learning for dexterous manipulation." *arXiv preprint arXiv:1704.03073.*