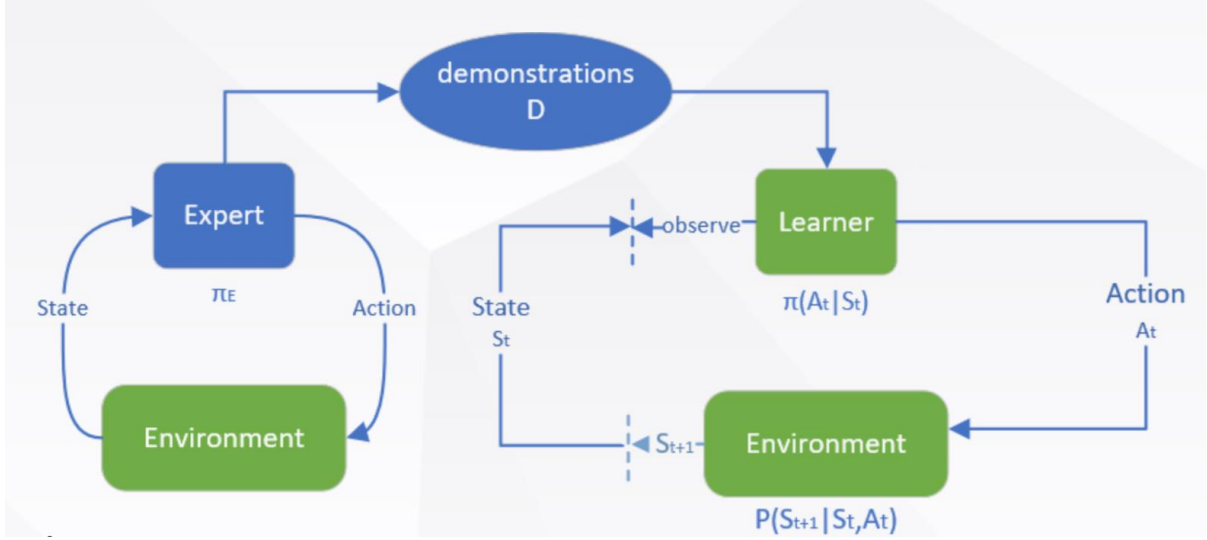
Adversarial Imitation Learning with Trajectorial Augmentation and Correction

Dafni Antotsiou, Carlo Ciliberto and Tae-Kyun Kim

ICRA 2021

Motivation

Imitation Learning



Deep Imitation Learning requires a large number of expert demonstrations, which are not always easy to obtain, especially for complex tasks. However, data augmentation cannot be easily applied to control tasks due to the sequential nature of the problem.

Method

Imitation using trajectorial data augmentation

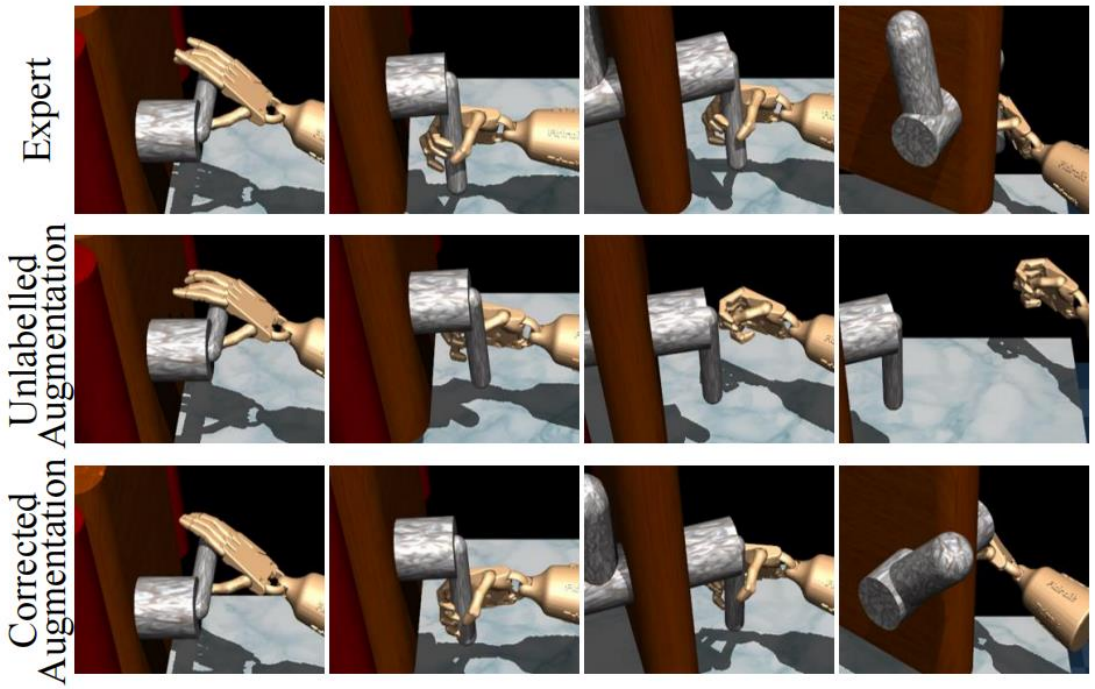


Fig. 1. Example of our proposed method for trajectorial data augmentation. **Top**: the original expert trajectory. **Middle**: the expert trajectory distorted by noise. The distortion makes the trajectory unsuccessful. This result is not guaranteed, therefore this augmentation is unlabelled. **Bottom**: our correction network modifies the unlabelled augmentation to produce a successful trajectory, different from the expert.

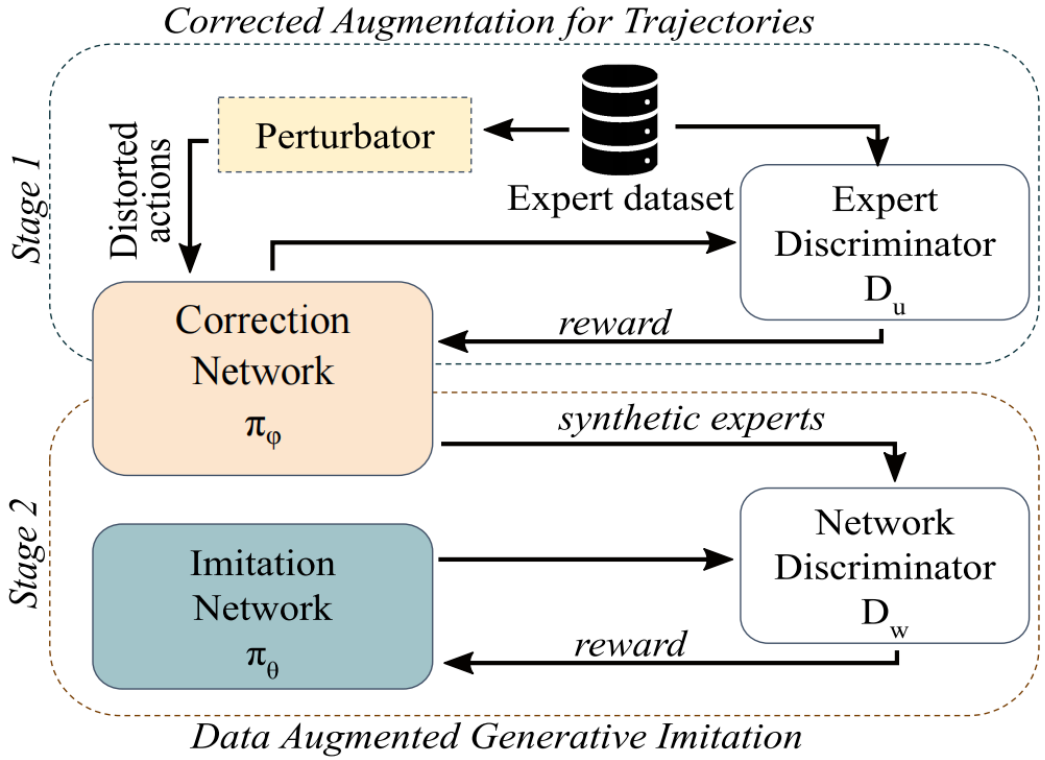


Fig. 2. Flow chart of our system, which performs imitation using trajectorial data augmentation. Stage 1 presents data augmentation by correcting distorted trajectories, while stage 2 presents data augmented imitation.

Method

$$q = \{a'_1, a'_2, a'_3, \dots\}, \quad \text{where} \quad a'_t = a_{E_t} + \nu \quad \text{and} \quad \tau_E = \{(s_{E_1}, a_{E_1}), (s_{E_2}, a_{E_2}), \dots\}. \tag{3}$$

$$L_{\mathbf{u}} = -\mathbb{E}_{\pi_E}[\log D_{\mathbf{u}}(s,a)] - \mathbb{E}_{\pi_\phi}[\log(1 - D_{\mathbf{u}}(s,a))], \tag{1}$$

$$L_\phi = \mathbb{E}_{\pi_\phi}[\log(1 - D_u(s,a))] + \lambda ||a - a'||_2^2. \tag{5}$$

Stage1 : Corrected Augmentation for Trajectories (CAT)

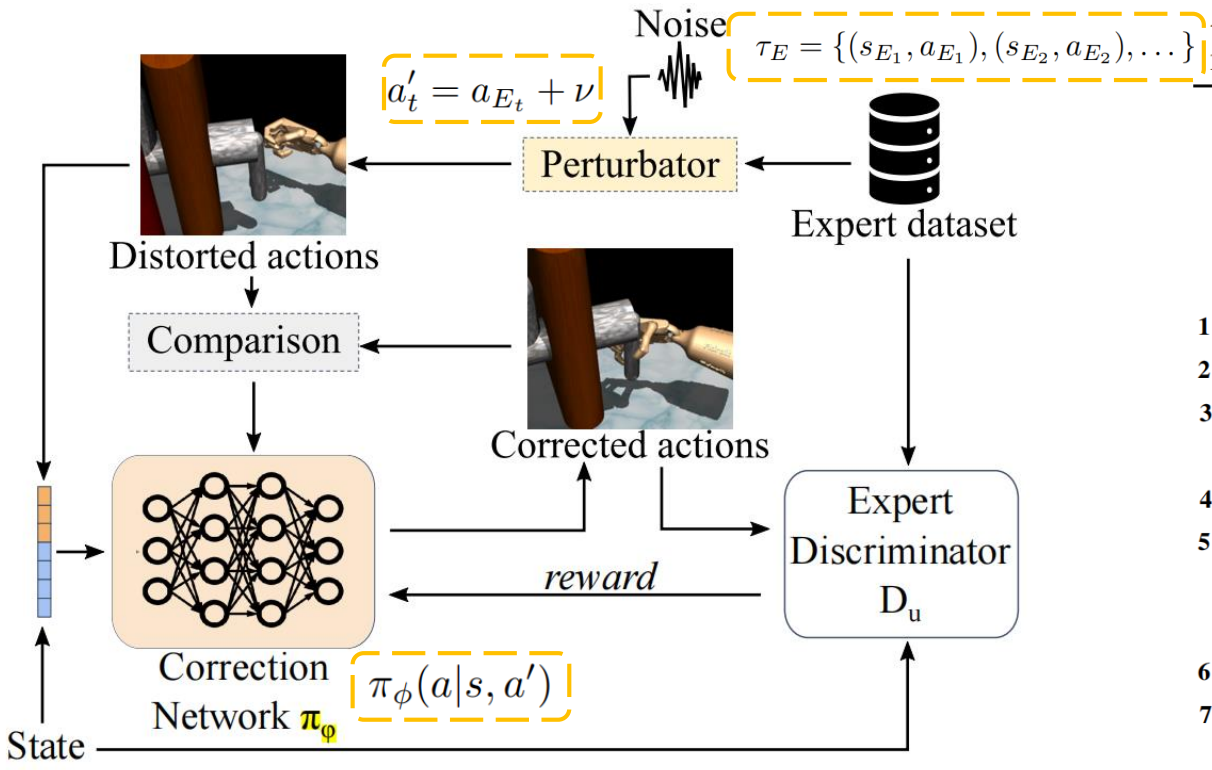


Fig. 3. Detailed overview of stage 1, which performs Corrected Augmentation For Trajectories. The architecture is semi-supervised since it is guided by unlabelled distorted actions.

Algorithm 1: Corrected Augmentation for Trajectories Algorithm

Input: Set of expert trajectories $\mathcal{T}_E$, regularisation λ noise σ, initial policy ϕ_0 and discriminator u_0, N number of perturbed action sequences.

// produce randomly perturbed augmented action sequences $\mathcal{Q}$

$\mathcal{Q} = \{\}$
for each τ_E **in** $\mathcal{T}_E$ **do**
  Generate N perturbed action sequences $\mathcal{Q}' = \{q_1, \dots, q_N\}$ from τ_E according to (3).
  $\mathcal{Q} = \mathcal{Q} \bigcup \mathcal{Q}'$.
end

// correct augmented trajectories so they are successful

for $i = 0, 1, 2, \dots$ **do**
  Sample trajectory $\tau_i \sim \pi_{\phi_i}(q_i)$, with q_i sampled uniformly from $\mathcal{Q}$.
  $u_{i+1} \leftarrow u_i - \nabla_u L_{u_i}$ using GAIL's (1).
  $\phi_{i+1} \leftarrow \phi_i - \nabla_\phi L_{\phi_i}$ using (5).
end

Method

☐ Stage2 : Data Augmented Generative Imitation (DAugGI)

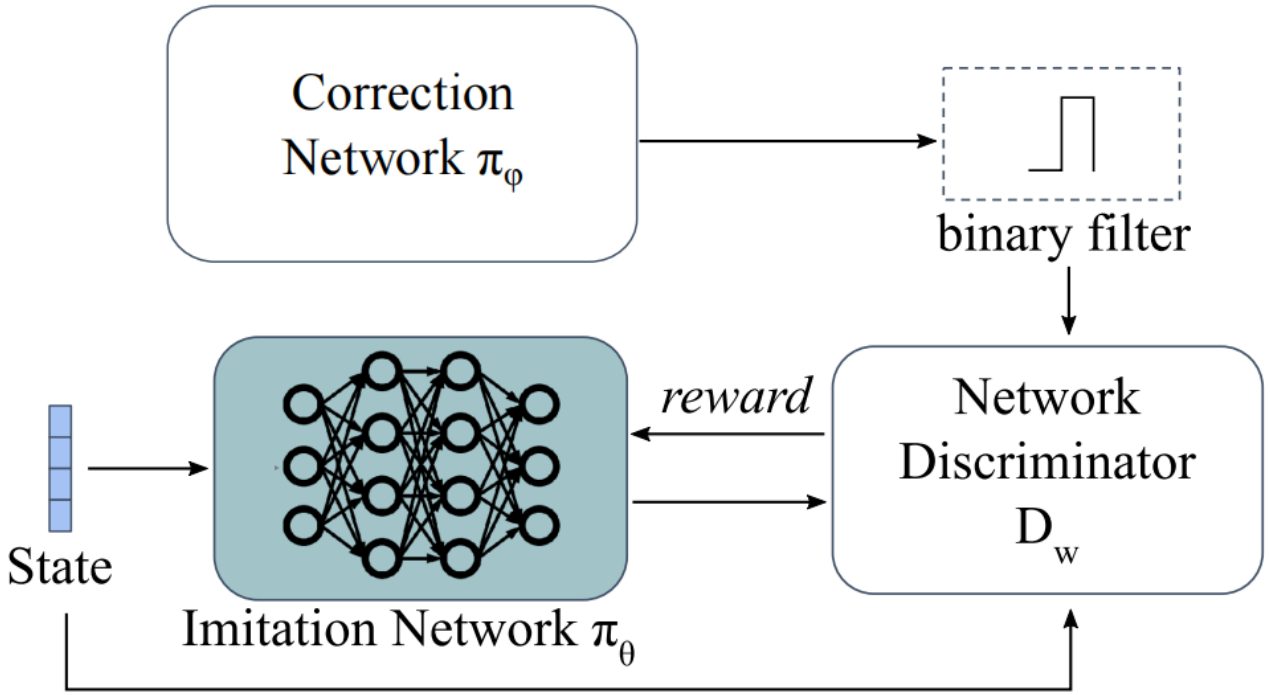


Fig. 4. Detailed overview of stage 2, which performs Data Augmented Generative Imitation, including the success filtering mechanism.

$$L_\theta = \mathbb{E}_{\pi_\theta}\left[\log(1 - D_w(s,a))\right]$$

$$L_w = -\mathbb{E}_{\pi_\phi}\left[\log D_w(s,a)\right] - \mathbb{E}_{\pi_\theta}\left[\log(1 - D_w(s,a))\right]$$

Experiments : CAT Evaluation

$$\overline{dtw}_n(\mathcal{T}_g) = \frac{\overline{dtw}(\mathcal{T}_g)}{\overline{dtw}(\mathcal{T}_E)},$$

where $\overline{dtw}(\mathcal{T}) = \frac{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} dtw(\tau_{z_i}, \tau_{z_j})}{(N-1)N/2}$.

TABLE I

CAT EVALUATION AND DATASET DIVERSITY

Task	CAT Success %		Dataset Diversity score $\overline{dtw}_n$		
	Random Aug/tion	Corrected Aug/tion	CAT / original	DAugGI / original	GAIL / original
HalfCheetah	0.7	**97.6**	0.54	0.68	0.73
Inv. Pendulum	4	**100**	0.91	**1.13**	1.11
Door	21	**56**	**1.11**	**0.26**	0.25
Pen	58	46	0.93	**1.12**	1.06
Hammer	62	**63**	**1.00**	**0.26**	0.24

Experiments : DAugGI Evaluation

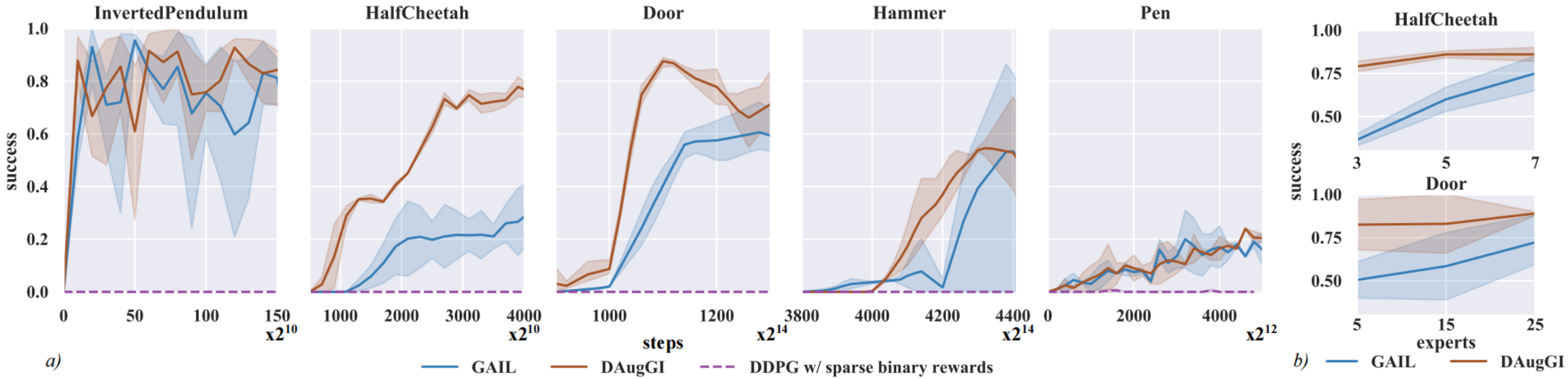


Fig. 5. a) Performance results of various tasks at different training steps. It includes easier OpenAI tasks (HalfCheetah and InvertedPendulum) with 3 experts, as well as more challenging dexterous manipulation tasks (Door, Hammer, Pen) with 25 experts. DAugGI is trained using the augmented CAT trajectories and generally outperforms GAIL, trained with the original limited trajectories. b) Ablation studies with different number of experts for HalfCheetah and Door tasks. DAugGI consistently outperforms GAIL, especially when the expert dataset is limited.

Augmenting Policy Learning with Routines Discovered from a Single Demonstration

Zelin Zhao * [1] , **Chuang Gan** [2], **Jiajun Wu** [3], **Xiaoxiao Guo** [2], **Joshua Tenenbaum** [4]

[1] Shanghai Jiao Tong University, [2] MIT-IBM Watson AI Lab
[3] Stanford University, [4] Massachusetts Institute of Technology
sjtuytc@sjtu.edu.cn, ganchuang@csail.mit.edu, jiajunwu@cs.stanford.edu,
Xiaoxiao.Guo@ibm.com, jbt@mit.edu

AAAI 2021

Motivation

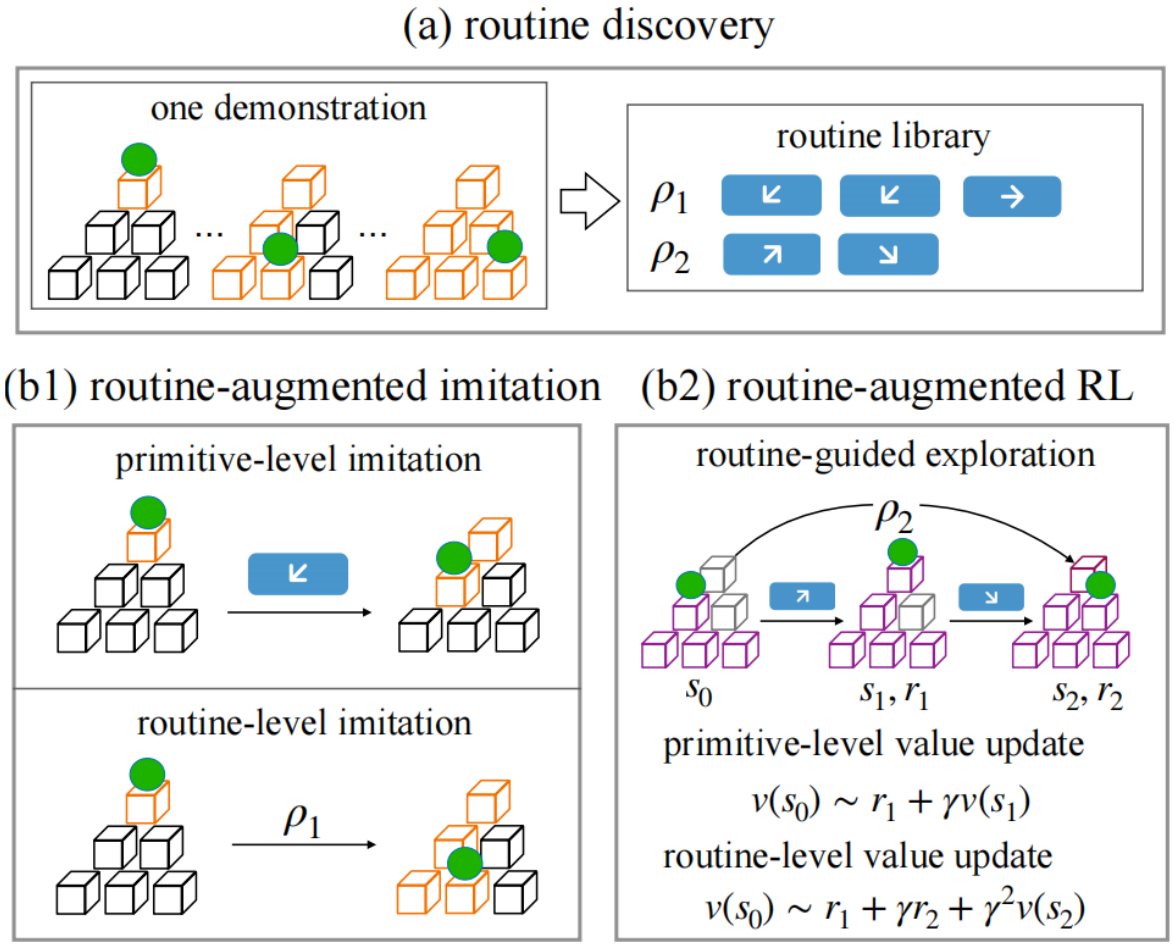
Humans can abstract prior knowledge from very little data and use it to boost skill learning.

Discover routines composed of primitive actions from a single demonstration and use discovered routines to augment policy learning

Method

focus on MDPs with high-dimensional states and discrete actions

Routine-Augmented Policy Learning (RAPL)



routine ρ : a sequence of primitive actions $(a^{(1)}, a^{(2)}, \ldots, a^{(|\rho|)})$

Method

☐ Part 1 : Routine Discovery

Phase 1 : Routine Proposal

Action Sequence — abcdbcabcd

↓ Sequitur Algorithm [Nevill-Manning and Witten 1997]

Grammer — S->CAC, A->bc, C->aAd

↓ Transfer non-terminal symbols to routines

Routine Candidates — A:bc, C:abcd

Phase 2 : Routine Selection

Routine Candidates

↓ Compute scores of each routine

$$f(\rho) + \lambda^{\text{length}} |\rho|$$

$f(\rho)$: the occurrence time of one routine ρ in the demonstrated action sequence

$|\rho|$: routine length

↓

Select K best routines

Method

☐ Part 2 : Routine Policy Learning

RAPL-SQIL : using routines to augment imitation learning

Idea : imitate the expert's experiences at multiple temporal scales.

$$\mathcal{L}^{\mathcal{SR}} = \delta^2(\mathcal{D}_{\text{prim}} \cup \mathcal{D}_{\text{routine}}, 1) + \lambda_{\text{sample}}\delta^2(\mathcal{D}_{\text{sample}}, 0)$$

$$\delta^2(\mathcal{D}, r) = \frac{1}{|\mathcal{D}|} \sum_{(s_t, \widetilde{\rho}, s_{t+|\widetilde{\rho}|}) \in \mathcal{D}} \left(Q_\theta(s_t, \widetilde{\rho}) - Q_{\text{target}}(\widetilde{\rho}, s_{t+|\widetilde{\rho}|}, r)\right)^2, \qquad Q_{\text{target}}(\widetilde{\rho}, s_{t+|\widetilde{\rho}|}, r) = R_{sq}(\widetilde{\rho}, r) + \Gamma(\widetilde{\rho}) \log\left(\sum_{\widetilde{\rho}' \in \widetilde{\mathcal{L}}} \exp\left(Q_\theta\left(s_{t+|\widetilde{\rho}|}, \widetilde{\rho}'\right)\right)\right),$$

D_{prim} : the experience from primitive-level demonstration. (s_t, a, s_{t+1})

D_{routine} : the demonstration explained by the abstracted routines. $(s_t, \rho, s_{t+|\rho|})$

D_{sample} : the experiences collected via interaction with the environments.

SQIL: Imitation Learning via Regularized Behavioral Cloning

Method

☐ Part 2 : Routine Policy Learning

RAPL-A2C : using routines to augment reinforcement learning

Idea : conduct value approximation and policy optimization at multiple temporal scales.

$$\mathcal{L}^{\text{policy}} = -A_{\text{routine}} \log \pi(\widetilde{\rho}_t | s_{t_0}; \theta_\pi),$$

$$\mathcal{L}^{\text{entropy}} = \sum_{\widetilde{\rho}} \pi(\widetilde{\rho} | s_{t_0}; \theta_\pi) \log \pi(\widetilde{\rho} | s_{t_0}; \theta_\pi).$$

$$\mathcal{L}^{\mathcal{AR}} = \mathbb{E}(\mathcal{L}^{\text{policy}} + \lambda^{\text{entropy}} \mathcal{L}^{\text{entropy}} + \lambda^{\text{value}} (\|A_{\text{routine}}\|^2 + \lambda^{\text{prim}} \|A_{\text{prim}}\|^2))$$

$A_{\text{routine}} = \sum_{i=0}^{N-1} \gamma^{t_i - t_0} R_{t_i} + \gamma^{t_N - t_0} V(s_{t_N}) - V(s_{t_0})$: routine-level n-step bootstrap advantage function

$A_{\text{prim}} = \sum_{i=0}^{N-1} \gamma^i r_{t_j+i} + \gamma^N V(s_{t_j+N}) - V(s_{t_j})$: primitive-level n-step boostrap advantage function

Experiments : RAPL-SQIL

Table 1: Comparing with several imitation learning baselines on 33 Atari games. We shown both alignment scores (defined in Eq. 10) and mean of human-normalized scores (Mnih et al. 2015) which indicates the alignment performance with regarding to the demonstration. Each number in the table is averaged over five random seeds.

	Alignment ($\pm$ std)	Mean ($\pm$ std)
BC	0.18 ($\pm$ 0.03)	18.3% (2.1%)
GAIL	0.16 ($\pm$ 0.08)	26.4% (1.6%)
SQIL	0.28 ($\pm$ 0.07)	29.4% (3.2%)
RAPL-SQIL	**0.34 ($\pm$ 0.07)**	**36.1% ($\pm$ 3.6%)**

$$s = 1 - \frac{D(\iota_d, \iota_t)}{|\iota_d|}, \qquad (10)$$

ι_d : the demonstrated action trajectory
ι_t : the action trajectory produced by the trained agent
D : Levenshtein distance

Experiments : RAPL-A2C

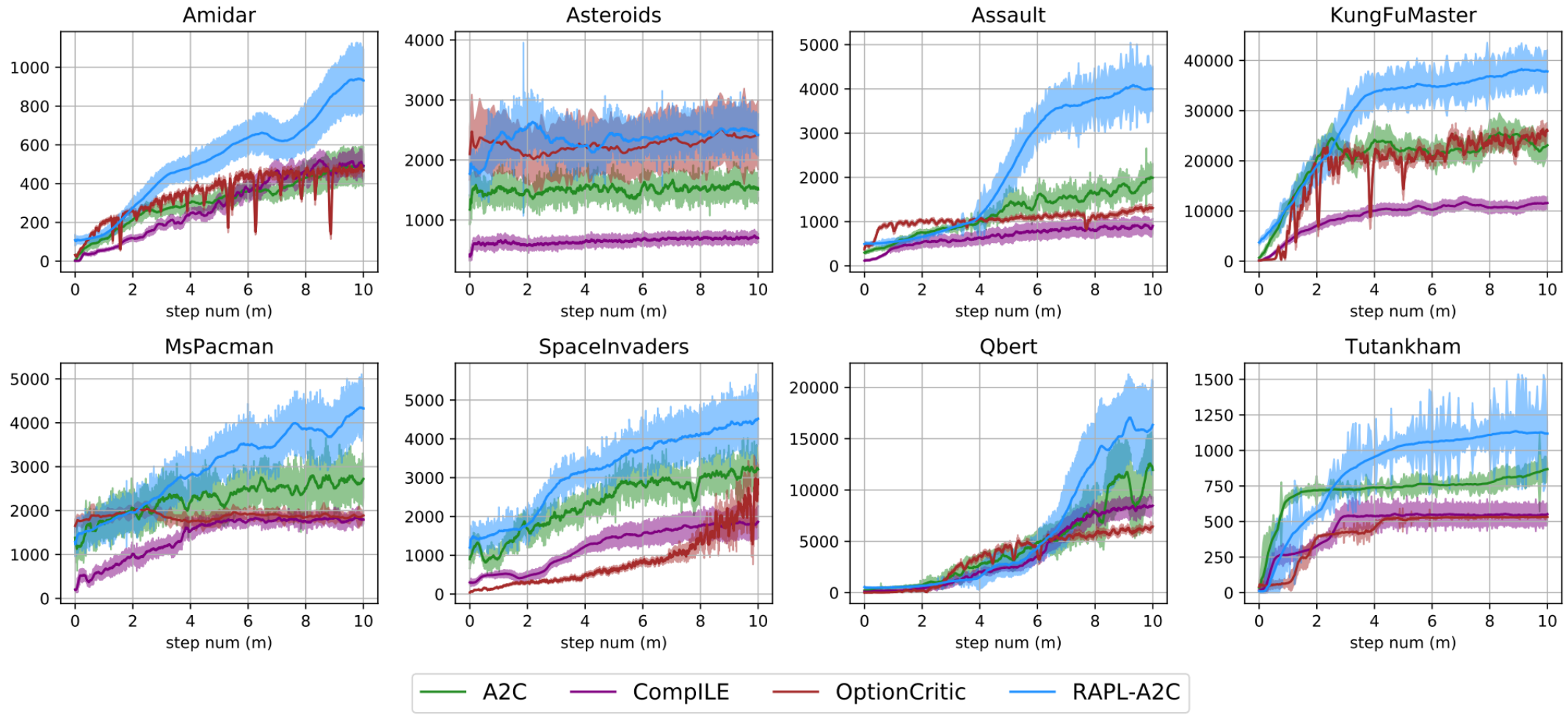


Figure 3: Training curves on eight randomly selected Atari games in comparison with several RL baselines. We plot both the mean and standard deviation in those curves across five agents with random seeds.

CompILE: Compositional Imitation Learning and Execution
The Option-Critic Architecture

Experiments : RAPL-A2C

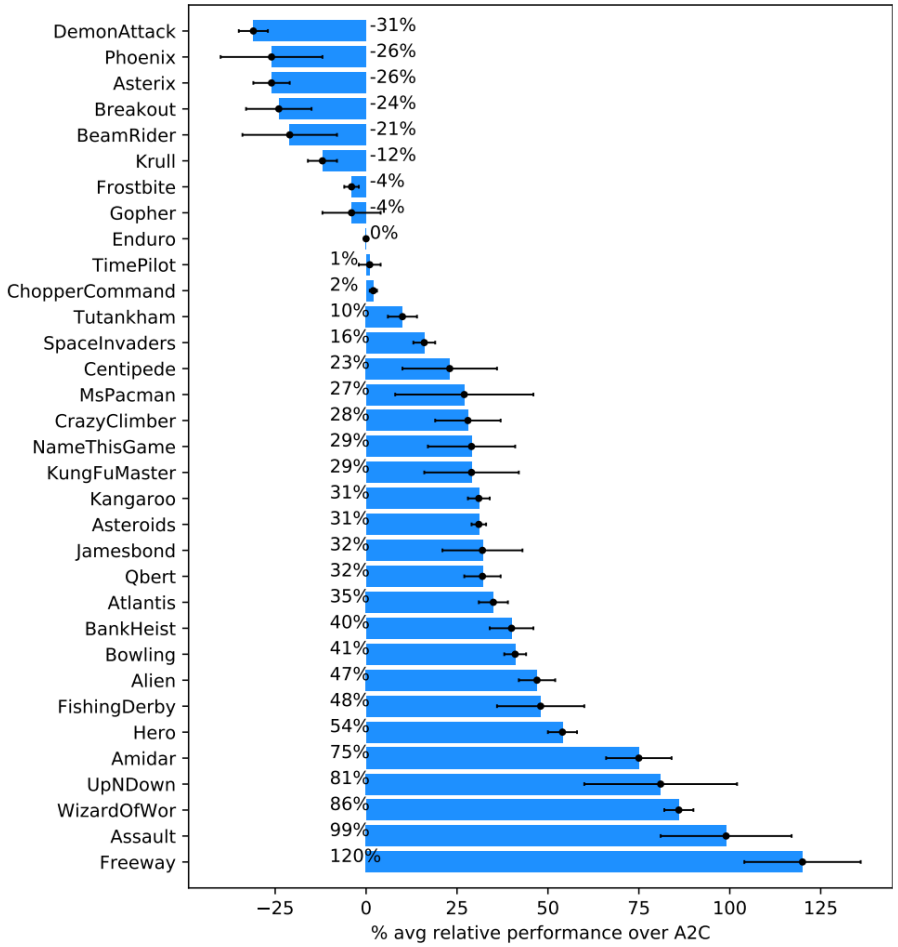
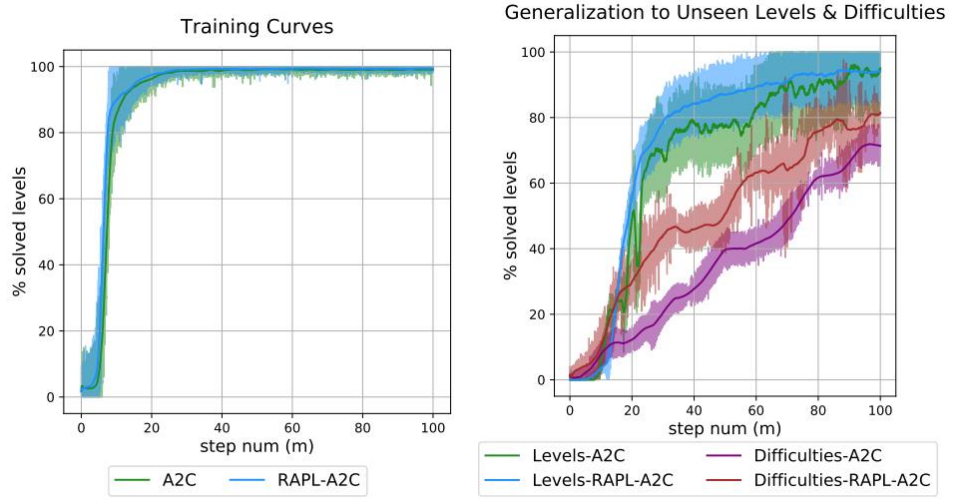
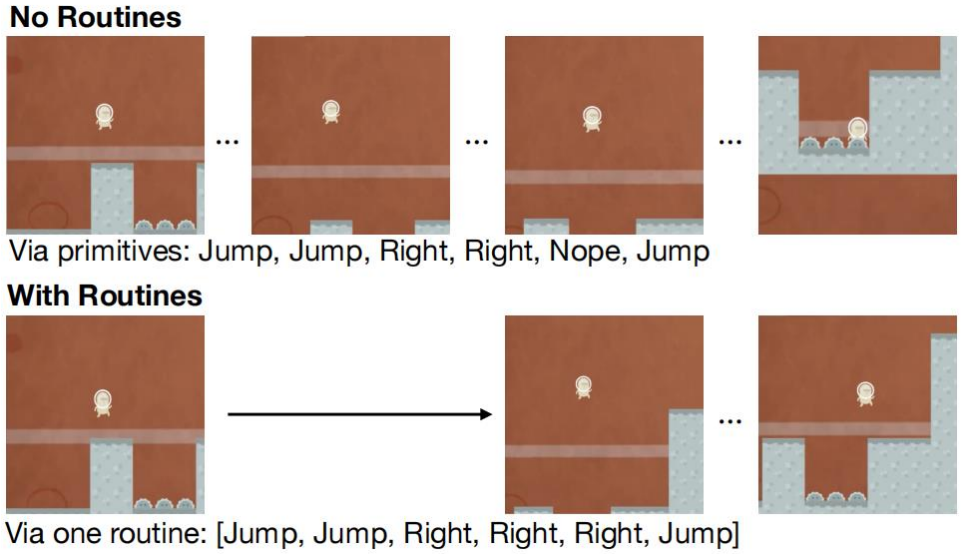


Figure 2: Relative performance of RAPL-A2C over A2C on Atari. Denote S_R as the score of RAPL-A2C and S_A is the score of A2C. The relative performance is calculated by $(S_R - S_A)/|S_A| \times 100\%$. Each number is averaged over five random agents and we also plot the stand error of the numbers.





Experiments : Effectiveness of Routine Discovery

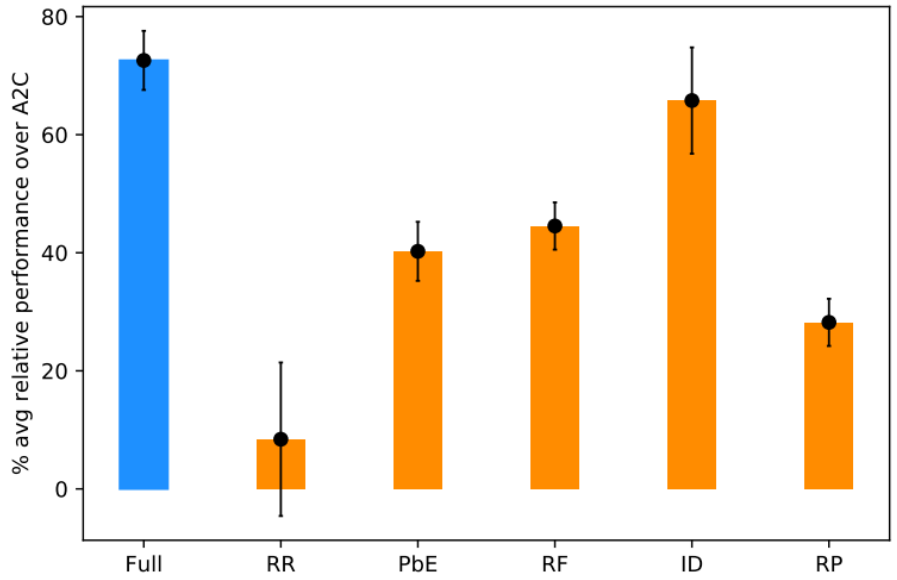


Figure 6: Comparison of ablated routine discovery models on Atari games. Mean and standard error over five random agents are shown in the figure.

(1) Random Routines (RR):each routine is generated randomly;
(2) proposal by Enumeration (PbE):enumerate all the possible combinations of primitive actions to form routine candidates.
(3) Random Fetch (RF):random fetch sub-sequences from the demonstration to form routines.
(4) Imperfect Demonstration (ID):the expert is only trained with 1 million steps.
(5) Repeat (RP):the routines are the repetition of most frequently used atomic actions in the demonstration

Thanks