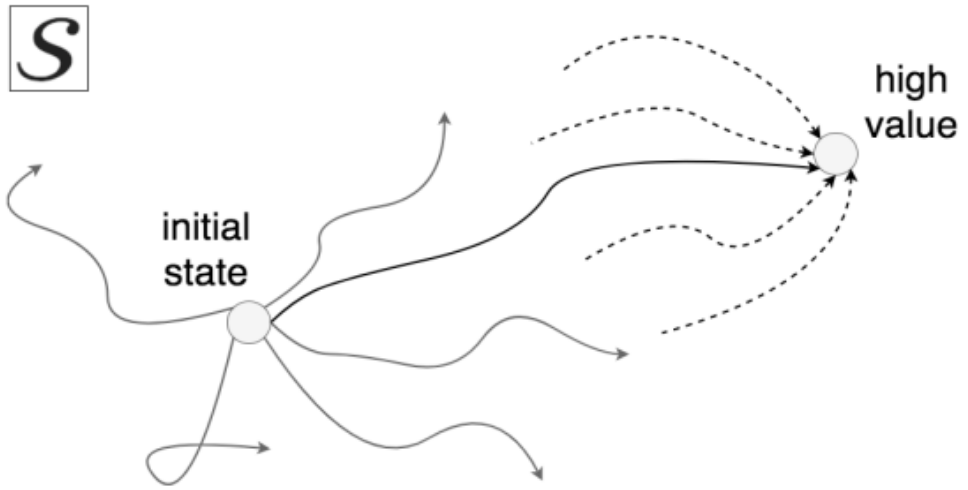
Recall Traces: Backtracking Models for Efficient Reinforcement Learning

Anirudh Goyal[1], Philemon Brakel[2], William Fedus[1], Soumye Singhal[1,4], Timothy Lillicrap[2], Sergey Levine[5], Hugo Larochelle[3], Yoshua Bengio[1]

ICLR 2019

[1]Mila, University of Montreal, [2] Google Deepmind, [3] Google Brain, [4] IIT Kanpur, [5] University of California, Berkeley. Corresponding author :anirudhgoyal9119@gmail.com

Motivation



In many environments only a tiny subset of all states yield high reward. We may want to preferentially train on those high-reward states and the probable trajectories leading to them.

sample efficiency

Producing Intended High Value States

- Method 1

Rely on picking the most valuable states stored in the replay buffer $\mathcal{B}$.

- Method 2

$V^{\pi}(s)$ (off-policy method)

reward (on-policy method)

Goal GAN[1]: goal states g are produced via a Generative Adversarial Network.

Algorithm 2 Produce High Value States

Require: Critic $V(s)$
Require: D; transformation 'decoder' from g to s
Require: Experience buffer $\mathcal{B}$ with tuples (s_t, a_t, r_t, s_{t+1})
Require: gen_state; boolean whether to generate states
Require: GAN, some generative model trained to model high-value goal states

```
if gen_state then
    g ~ GAN
    D : g ↦ s
    Return s
else
    Return argmax(V(s)) ∀s ∈ B
end if
```

[1] Florensa C, Held D, Geng X, et al. Automatic goal generation for reinforcement learning agents[C]//International conference on machine learning. PMLR, 2018: 1515-1528.

Generating Recall Traces

- Backtracking Model $p(s_t, a_t | s_{t+1})$

  Predict the preceding states that terminate at a given high-reward state.

Start from a high value state (or one that is estimated to have high value), predict and sample which (state,action)-tuples may have led to that high value state.

Recall Traces

use these traces to improve a policy.

Generating Recall Traces

- Backtracking Model

$$B_\phi = q_\phi(s_t, a_t | s_{t+1})$$

backward policy

$$\pi_b = q(a_t | s_{t+1})$$

state generator

$$q(\Delta s_t | a_t, s_{t+1})$$

$$\Delta s_t = s_t - s_{t+1}$$

for training stability with continuous-valued states

$$q_\phi(\Delta s_t, a_t | s_{t+1}) = q(\Delta s_t | a_t, s_{t+1}) q(a_t | s_{t+1})$$

- Generating Recall Traces

$$a_t \sim q(a_t | s_{t+1})$$

$$\Delta s_t \sim q(\Delta s_t | a_t, s_{t+1})$$

$$s_t = \Delta s_t + s_{t+1}$$

Improving The Policy From The Recall Traces

Algorithm 1 Improve Policy via Recall Traces and Backtracking Model

Require: RL algorithm with parameterized policy (i.e. TRPO, Actor-Critic)
Require: Agent policy $\pi_\theta(a|s)$
Require: Backtracking model $B_\phi = q_\phi(\Delta s_t, a_t|s_{t+1})$
Require: Critic $V(s)$
Require: k quantile of best state values used to train backtracking model, k_{traj} number of trajectories filtered by returns.
Require: N; number of backward trajectories per target state
Require: α, β; forward, backward learning rates

Randomly initialize agent policy parameters θ
Randomly initialize backtracking model parameters ϕ
for $t = 1$ to K **do**
    Execute policy to produce trajectory τ
    Add trajectory $\tau = (s_1, a_1, r_1, \cdots, s_T, a_T, r_T)$ in $\mathcal{B}$
    Estimate $\nabla_\theta R(\pi_\theta)$ from RL algorithm
    $\theta \leftarrow \theta + \alpha \nabla_\theta R(\pi_\theta)$
    Compute $\mathcal{L}_\mathcal{B}$ via Equation 2, using top $k\%$ valuable states from top k_{traj} trajectories in $\mathcal{B}$
    $\phi \leftarrow \phi + \beta \nabla_\phi \mathcal{L}_\mathcal{B}$
    Obtain target high value state s (see Algorithm 2 for details)
    Generate N recall traces $\tilde{\tau}$ for s using $B_\phi(s)$
    Compute imitation loss $\mathcal{L}_\mathcal{I}$ via Equation 3
    $\theta \leftarrow \theta + \alpha \nabla_\theta \mathcal{L}_\mathcal{I}$
end for

Improving The Policy From The Recall Traces

Algorithm 1 Improve Policy via Recall Traces and Backtracking Model

Require: RL algorithm with parameterized policy (i.e. TRPO, Actor-Critic)
Require: Agent policy $\pi_\theta(a|s)$
Require: Backtracking model $B_\phi = q_\phi(\Delta s_t, a_t|s_{t+1})$
Require: Critic $V(s)$
Require: k quantile of best state values used to train backtracking model, k_{traj} number of trajectories filtered by returns.
Require: N; number of backward trajectories per target state
Require: α, β; forward, backward learning rates

Randomly initialize agent policy parameters θ
Randomly initialize backtracking model parameters ϕ
for $t = 1$ to K **do**
  Execute policy to produce trajectory τ
  Add trajectory $\tau = (s_1, a_1, r_1, \cdots, s_T, a_T, r_T)$ in $\mathcal{B}$
  Estimate $\nabla_\theta R(\pi_\theta)$ from RL algorithm
  $\theta \leftarrow \theta + \alpha \nabla_\theta R(\pi_\theta)$
  Compute $\mathcal{L}_\mathcal{B}$ via Equation 2, using top $k\%$ valuable states from top k_{traj} trajectories in $\mathcal{B}$
  $\phi \leftarrow \phi + \beta \nabla_\phi \mathcal{L}_\mathcal{B}$
  Obtain target high value state s (see Algorithm 2 for details)
  Generate N recall traces $\tilde{\tau}$ for s using $B_\phi(s)$
  Compute imitation loss $\mathcal{L}_\mathcal{I}$ via Equation 3
  $\theta \leftarrow \theta + \alpha \nabla_\theta \mathcal{L}_\mathcal{I}$
end for

RL Training

Backtracking Model Training

Generating Recall Traces

Imitation Learning

$$\mathcal{L}_\mathcal{B} = \log q_\phi(\tau) = \log \prod_{t=0}^{T} q(\Delta s_t, a_t|s_{t+1})$$
$$= \sum_{t=0}^{T} \log q(a_t|s_{t+1}) + \log q(\Delta s_t|a_t, s_{t+1}), \quad (2)$$

$$\mathcal{L}_\mathcal{I} = \sum_{t=0}^{T} \log p(a_t|s_t) = \sum_{t=0}^{T} \log \pi_\theta(a_t|s_t), \quad (3)$$

Variational Interpretation

- Variational Inference (RL Model: θ , Backtracking Model: ϕ)

$$p(R > L|\theta) = \frac{p(R > L, \tau|\theta)}{p(\tau|R > L, \theta)}$$

$$\log p(R > L|\theta) = \log \frac{p(R > L, \tau|\theta)}{q(\tau)} - \log \frac{p(\tau|R > L, \theta)}{q(\tau)}$$

$$\text{左边} = \int q(\tau) \log p(R > L|\theta)\, \mathrm{d}\tau = \log p(R > L|\theta)$$

$$\text{右边} = \int q(\tau) \log \frac{p(R > L, \tau|\theta)}{q(\tau)}\, \mathrm{d}\tau - \int q(\tau) \log \frac{p(\tau|R > L, \theta)}{q(\tau)}\, \mathrm{d}\tau$$
$$= \mathrm{ELBO} + \mathrm{KL}(q(\tau)||p(\tau|R > L, \theta))$$

$$\begin{aligned} \therefore \log p(R > L|\theta) &= \mathrm{ELBO} + \mathrm{KL}(q(\tau)||p(\tau|R > L, \theta)) \\ &\geq \mathrm{ELBO} \end{aligned}$$

Variational Interpretation

EM Algorithm (RL Model: θ , Backtracking Model: ϕ)

(1) E-step：固定θ，优化ϕ，通过最小化KL来最大化ELBO

$$\begin{aligned}\phi &= \underset{\phi}{argmin}\,\mathrm{KL}(q_\phi(\tau)||p(\tau|R>L,\theta)) \\ &= \underset{\phi}{argmin}\,\mathrm{KL}(p(\tau|R>L,\theta)||q_\phi(\tau))\end{aligned}$$

$$\begin{aligned}\because \mathrm{KL}(p(\tau|R>L,\theta)||q_\phi(\tau)) &= \int p(\tau|R>L,\theta)\log\frac{p(\tau|R>L,\theta)}{q_\phi(\tau)}\,\mathrm{d}\tau \\ &= \int p(\tau|R>L,\theta)\log p(\tau|R>L,\theta)\,\mathrm{d}\tau - \int p(\tau|R>L,\theta)\log q_\phi(\tau)\,\mathrm{d}\tau\end{aligned}$$

$$\begin{aligned}\therefore \phi &= \underset{\phi}{argmax}\int p(\tau|R>L,\theta)\log q_\phi(\tau)\,\mathrm{d}\tau \\ &= \underset{\phi}{argmax}\,E_{\tau\sim p(\tau|R>L,\theta)}[\log q_\phi(\tau)]\end{aligned}$$

Variational Interpretation

EM Algorithm (RL Model: θ , Backtracking Model: ϕ)

(2) M-step：固定ϕ，优化θ，最大化ELBO

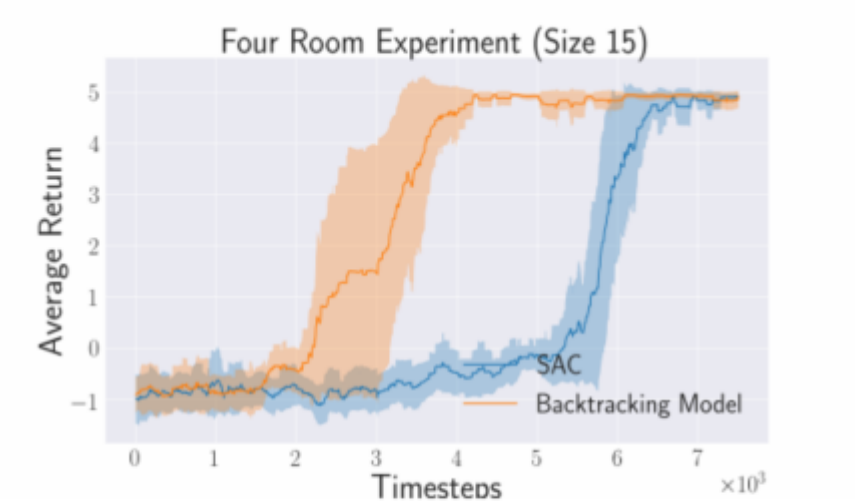
$$\begin{aligned}\mathrm{ELBO} &= \int q_\phi(\tau) \log \frac{p(R > L, \tau|\theta)}{q_\phi(\tau)} \, \mathrm{d}\tau \\ &= \int q_\phi(\tau) \log p(R > L, \tau|\theta) \, \mathrm{d}\tau - \int q_\phi(\tau) \log q_\phi(\tau) \, \mathrm{d}\tau\end{aligned}$$

$$\therefore \theta = \underset{\theta}{argmax}\, \mathrm{ELBO} = \underset{\theta}{argmax} \int q_\phi(\tau) \log p(R > L, \tau|\theta) \, \mathrm{d}\tau$$

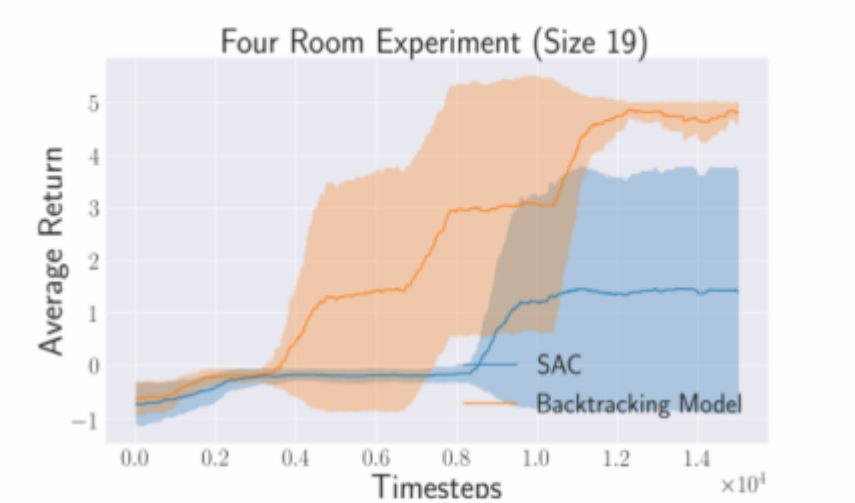
$$\because q_\phi(\tau) = p(\tau|R > L, \theta_t)$$

$$\begin{aligned}\therefore \theta &= \underset{\theta}{argmax}\, E_{\tau \sim q_\phi(\tau)} [\log p(R > L, \tau|\theta)] \\ &= \underset{\theta}{argmax}\, E_{\tau \sim q_\phi(\tau)} [\log p(\tau|\theta)]\end{aligned}$$

Experiments : Access To True Backtracking Model



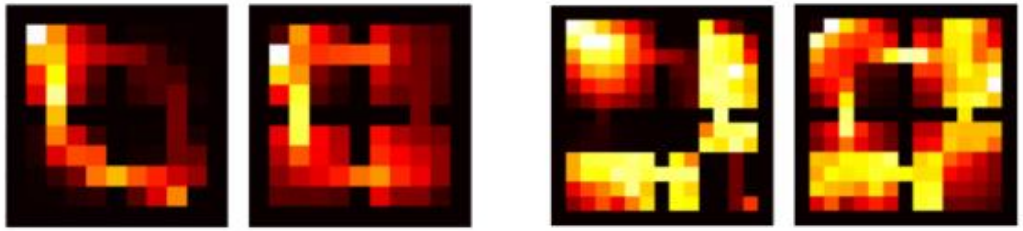
(a) 15-Dimension Environment



(b) 19-Dimension Environment

Figure 2: Training curves from the Four Room Environment for the Actor-Critic baseline (blue) and the backtracking model augmented Actor-Critic (orange). For the size-19 environment, several of the Actor-Critic baselines failed to converge, whereas the augmented recall trace model always succeeded in the number of training steps considered. For additional results see Figure 9 in Appendix.

Experiments : Comparison With Prioritized Experience Replay (PER)



(a) 11x11 environment (b) 15x15 environment

Figure 3: Visitation count visualization of trained policies for PER (left) and Recall Traces (right) for two 4-room grid sizes.

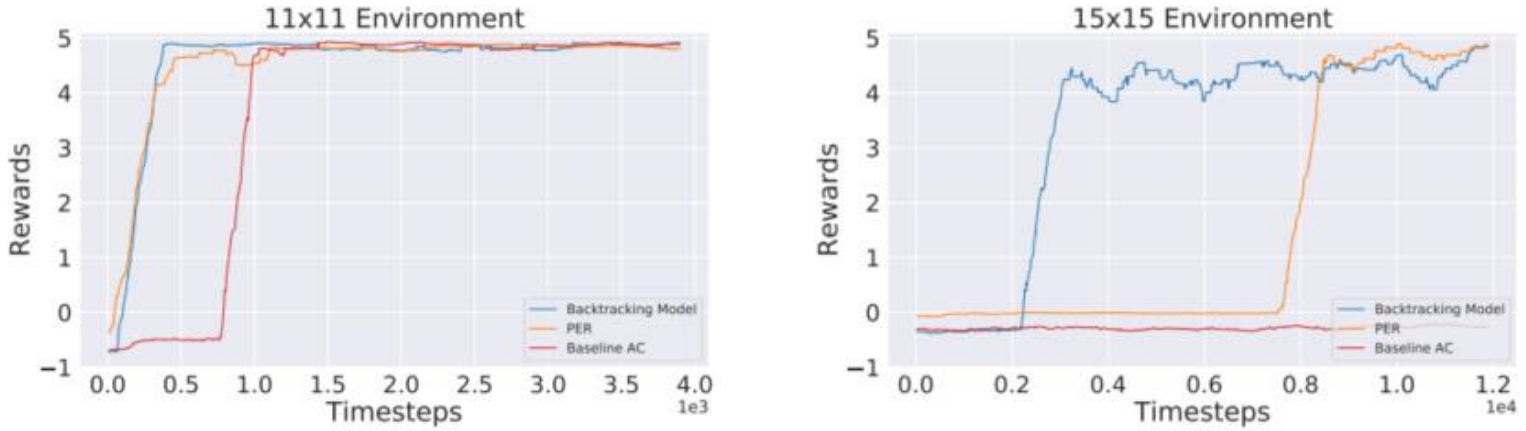
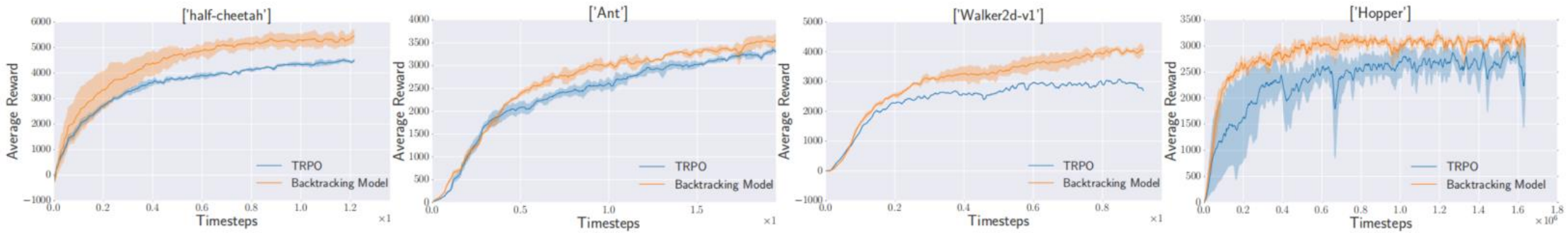


Figure 4: Plots for reward vs. time steps, comparing the performance of recall traces (labeled BacktrackingModel), PER and baseline Actor Critic (AC).

Experiments : Learning Backtracking Model



(a) Half-Cheetah (b) Ant (c) Walker (d) Hopper

Figure 6: Our model as compared to TRPO. For TRPO baselines, except walker, we ran with 5 different random seeds. For our model, we ran with 5 different random seeds.

(a) Humanoid (b) Ant (c) Walker (d) Hopper

Figure 7: Our model as compared to SAC. We ran SAC baselines with 2 different random seeds. For our model, we ran with 5 different random seeds.

Discussion

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

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

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

Corrected Augmentation for Trajectories (CAT)[2]

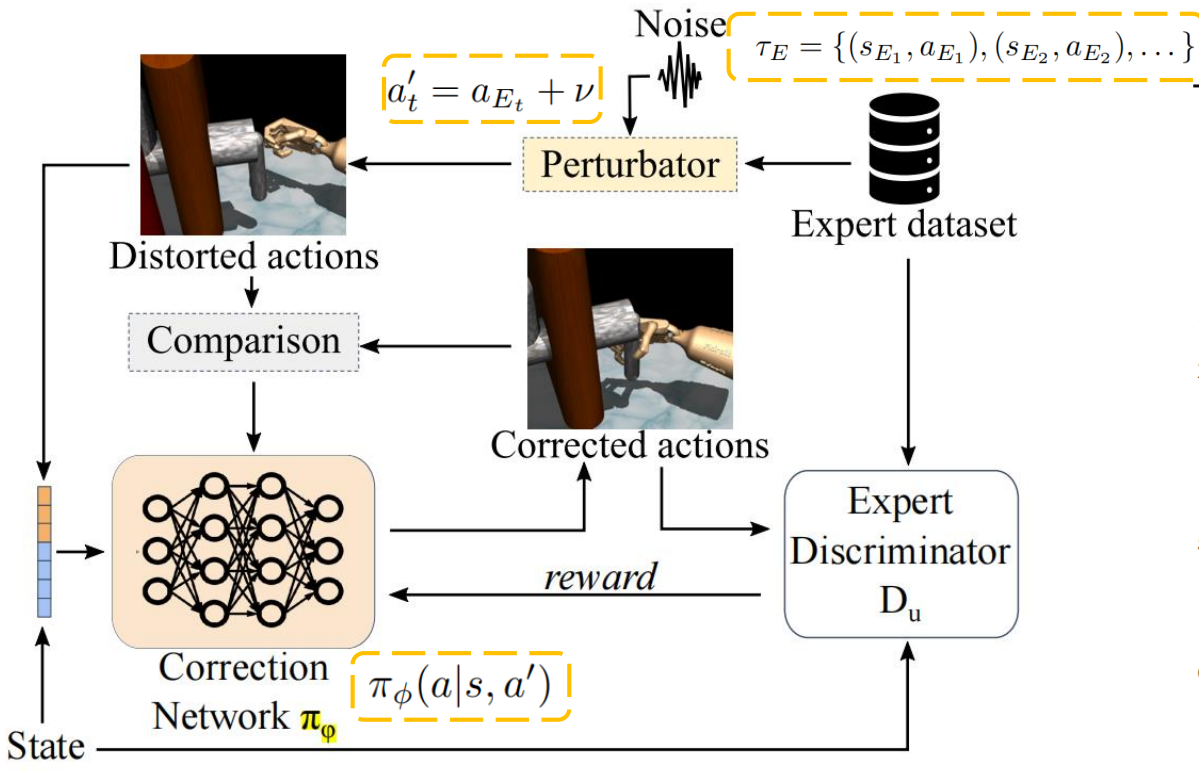


Fig. 3. Detailed overview of stage 1, which performs Corrected Augmentation For Trajectories. The architecture is semi-supervised since it is guided by unlabelled distorted actions.

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

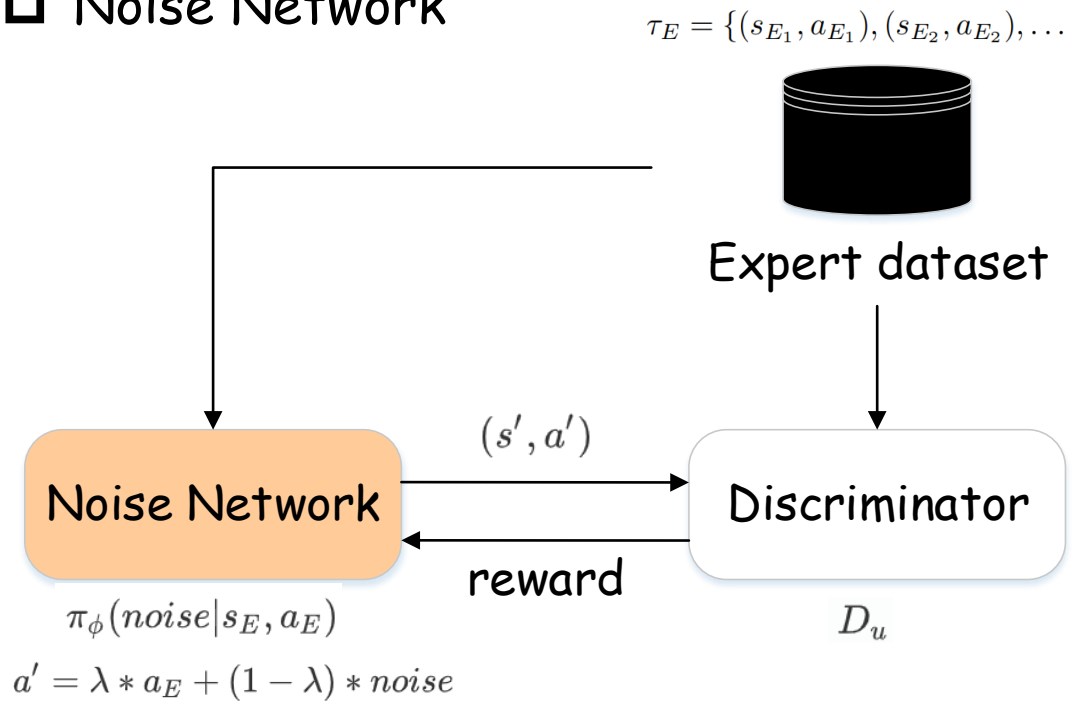
for $i = 0, 1, 2, \dots$ **do**
  Sample trajectory $\tau_i \sim \pi_{\phi_i}(q_i)$, with q_i sampled uniformly from $\mathcal{Q}$.
  $u_{i+1} \leftarrow u_i - \nabla_u L_{u_i}$ using GAIL's (1).
  $\phi_{i+1} \leftarrow \phi_i - \nabla_\phi L_{\phi_i}$ using (5).
end

[2] Antotsiou D, Ciliberto C, Kim T K. Adversarial Imitation Learning with Trajectorial Augmentation and Correction[J]. arXiv preprint arXiv:2103.13887, 2021.

Discussion

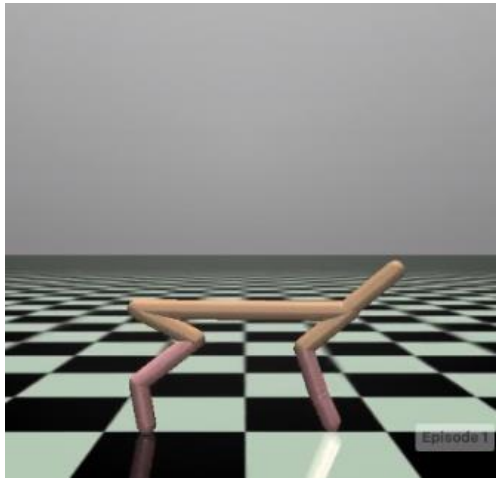
$$L_u = -E_{\pi_E}[logD_u(s,a)] - E_{\pi_\phi}[log(1 - D_u(s,a))]$$
$$L_\phi = E_{\pi_\phi}[log(1 - D_u(s,a))]$$

Noise Network



$$a' = \lambda * a_E + (1 - \lambda) * noise$$

HalfCheetah-v2



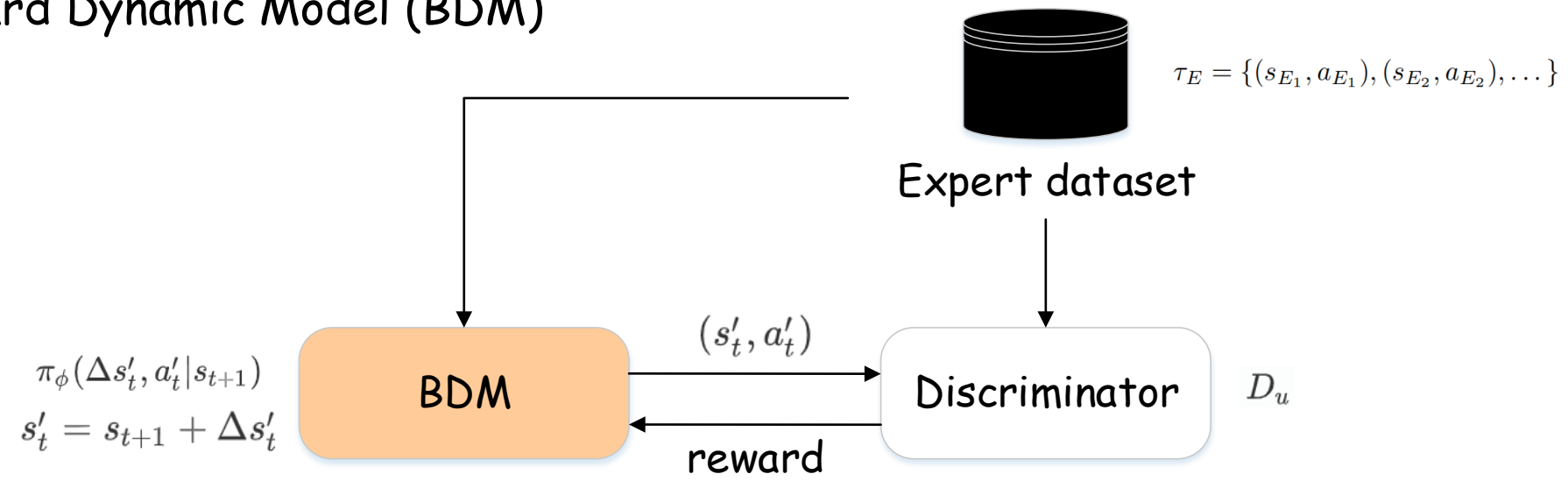
方法	增广轨迹条数	增广avgret	BC训练是否使用专家轨迹	BC训练后验证100条轨迹的avgret
expert	0	2439.8168651239403	是	991.2917141601323
cat	497	2221.0755368804244	否	**2906.3392535252096**
noise network($\lambda = 0.9$)	497	1172.8915687165966	否	1341.52380444484
mixup($\alpha = 0.1$)	42	1124.827231400378	否	1240.4956561966935

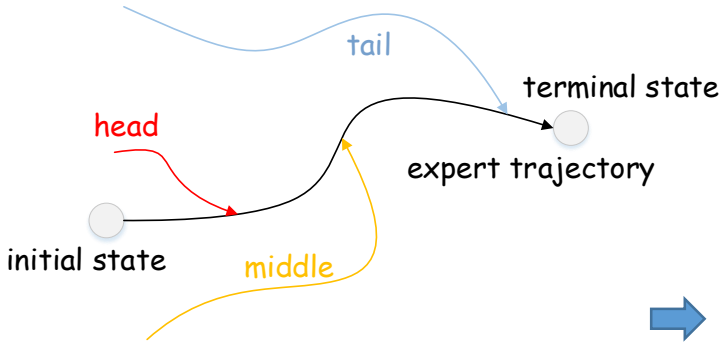
$\lambda \sim Beta(\alpha, \alpha); \lambda' = max(\lambda, 1 - \lambda)$
$a' = \lambda' a_1 + (1 - \lambda') a_2;$

Discussion

$$L_u = -E_{\pi_E}[logD_u(s,a)] - E_{\pi_\phi}[log(1-D_u(s,a))]$$
$$L_\phi = E_{\pi_\phi}[log(1-D_u(s,a))]$$

Backward Dynamic Model (BDM)





方法	增广轨迹条数	增广avgret	BC训练是否使用专家轨迹	BC训练后验证100条轨迹的avgret
expert	0	2439.8168651239403	是	991.2917141601323
cat	497	2221.0755368804244	否	**2906.3392535252096**
noise network($\lambda = 0.9$)	497	1172.8915687165966	否	1341.52380444484
mixup($\alpha = 0.1$)	42	1124.827231400378	否	1240.4956561966935
BDM	497	——	否	319.0855923529666
BDM_CL	497	——	否	1095.62363152283

Thanks