PlayVirtual: Augmenting Cycle-Consistent Virtual Trajectories for Reinforcement Learning

Tao Yu[1] **Cuiling Lan**[2] **Wenjun Zeng**[2] **Mingxiao Feng**[1] **Zhibo Chen**[1]

[1]University of Science and Technology of China [2]Microsoft Research Asia

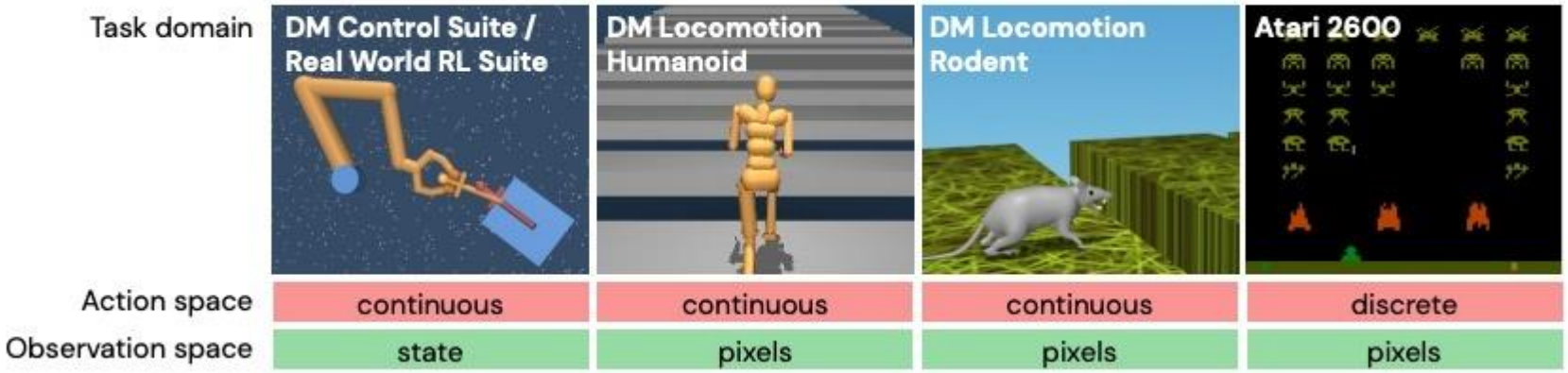
yutao666@mail.ustc.edu.cn, {culan,wezeng}@microsoft.com

fmxustc@ustc.edu.cn, chenzhibo@ustc.edu.cn

NeurIPS 2021

Motivation

☐ Reinforcement Learning from Pixels



Challenge:

High dimensional observation

Limited interaction with the environment

→ Lower sample efficiency

Motivation

◻ Previous Sample-Efficient RL Methods

1.Auxiliary task:

- contrastive learning (e.g., CURL [1])
- Predicting future states (e.g., SPR [2])
- Maximize the mutual information (e.g., CPC [3])
- Image reconstruction (e.g., Yarats et al. [4])

Help state representation learning

Problem: Limited experience is still deficient in the representation learning.

2.Image augmentation(e.g., RAD [5], DrQ [6]):

- Crop, shift, intensity, …

Increase data diversity from appearance

Problem: It cannot enrich the experienced trajectories (state-action sequences) in training.

[1] Laskin M, Srinivas A, Abbeel P. Curl: Contrastive unsupervised representations for reinforcement learning. ICML 2020.
[2] Schwarzer M, Anand A, Goel R, et al. Data-efficient reinforcement learning with self-predictive representations. ICLR 2021.
[3] Oord A, Li Y, Vinyals O. Representation learning with contrastive predictive coding. arXiv:1807.03748, 2018.
[4] Kostrikov D Y A Z I, Fergus B A J P R. Improving Sample Efficiency in Model-Free Reinforcement Learning from Images[J]. AAAI 2021.
[5] Laskin M, Lee K, Stooke A, et al. Reinforcement learning with augmented data. NeurIPS2020.
[6] Yarats D, Kostrikov I, Fergus R. Image Augmentation Is All You Need: Regularizing Deep Reinforcement Learning from Pixels. ICLR 2021.

Method:Overall Framework

- Idea:

Create some virtual "experience"

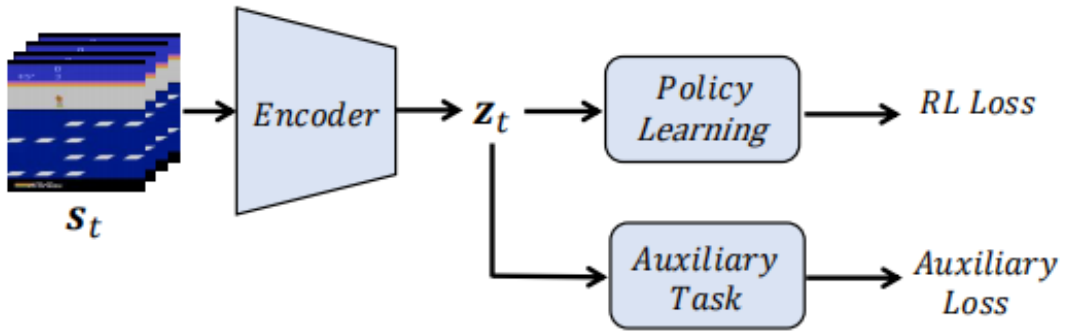
↓

Boost representation learning

↓

Sample efficiency ↑↑

- Overall Framework:



(a)

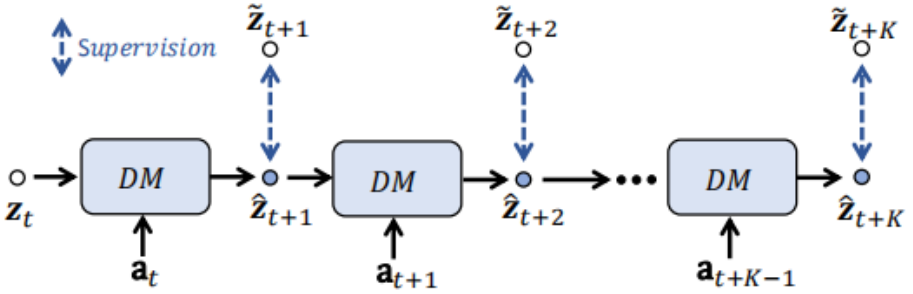
Method:Auxiliary task

Dynamics Model(DM)

real trajectory: $(s_t, a_t, \ldots, a_{t+k-1}, s_{t+k})$

$$\hat{\mathbf{z}}_{t+k+1} = \begin{cases} h(\mathbf{z}_{t+k}, \mathbf{a}_{t+k}) & \text{if } k = 0 \\ h(\hat{\mathbf{z}}_{t+k}, \mathbf{a}_{t+k}) & \text{if } k = 1, 2, \cdots, K-1. \end{cases}$$

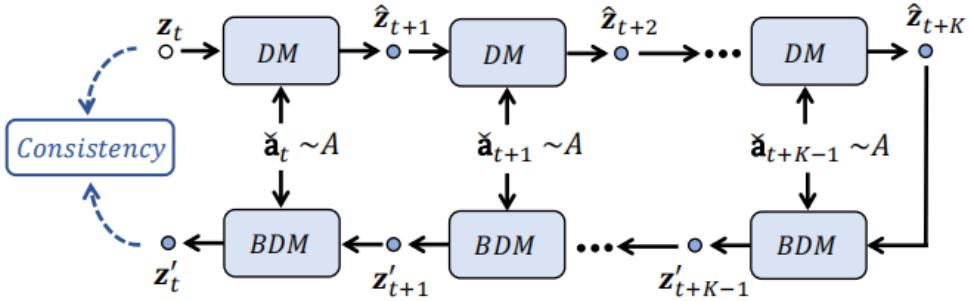
$$\mathcal{L}_{pred} = \sum_{k=1}^{K} d(\hat{\mathbf{z}}_{t+k}, \tilde{\mathbf{z}}_{t+k})$$



(b)

Backward Dynamics Model(BDM) & Cycle Consistency Constraint

$$\mathbf{z}'_{t+k-1} = \begin{cases} b(\hat{\mathbf{z}}_{t+k}, \check{\mathbf{a}}_{t+k-1}) & \text{if } k = K \\ b(\mathbf{z}'_{t+k}, \check{\mathbf{a}}_{t+k-1}) & \text{if } k = K-1, K-2, \cdots, 1. \end{cases}$$



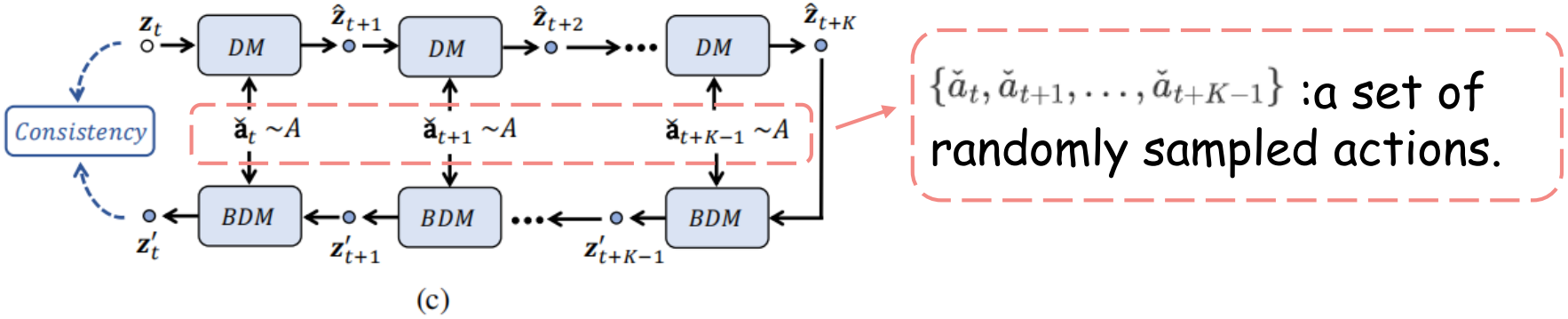
(c)

Method

Dynamics Model(DM) $h(\cdot,\cdot): (\mathbf{z}_t, \mathbf{a}_t) \rightarrow \mathbf{z}_{t+1}$ $d_{\mathcal{M}}$:A distance metric on space $\mathcal{M}$

Backward Dynamics Model(BDM) $b(\cdot,\cdot): (\mathbf{z}_{t+1}, \mathbf{a}_t) \rightarrow \mathbf{z}_t$ M :Sample M sets of actions in the action space

Backward Dynamics Model(BDM) & Cycle Consistency Constrain



(c)

Cycle Consistency loss:

$$forward: \quad \hat{\mathbf{z}}_t = \mathbf{z}_t,\ \hat{\mathbf{z}}_{t+k+1} = h(\hat{\mathbf{z}}_{t+k}, \mathbf{a}_{t+k}),\ \ for\ k = 0, 1, \cdots, K-1,$$

$$backward: \quad \mathbf{z}'_{t+K} = \hat{\mathbf{z}}_{t+K},\ \mathbf{z}'_{t+k} = b(\mathbf{z}'_{t+k+1}, \mathbf{a}_{t+k}),\ \ for\ k = K-1, K-2, \cdots, 0.$$

$$\mathcal{L}_{cyc} = \frac{1}{M} \sum_{m=1}^{M} d_{\mathcal{M}}(\mathbf{z}'_t, \mathbf{z}_t)$$

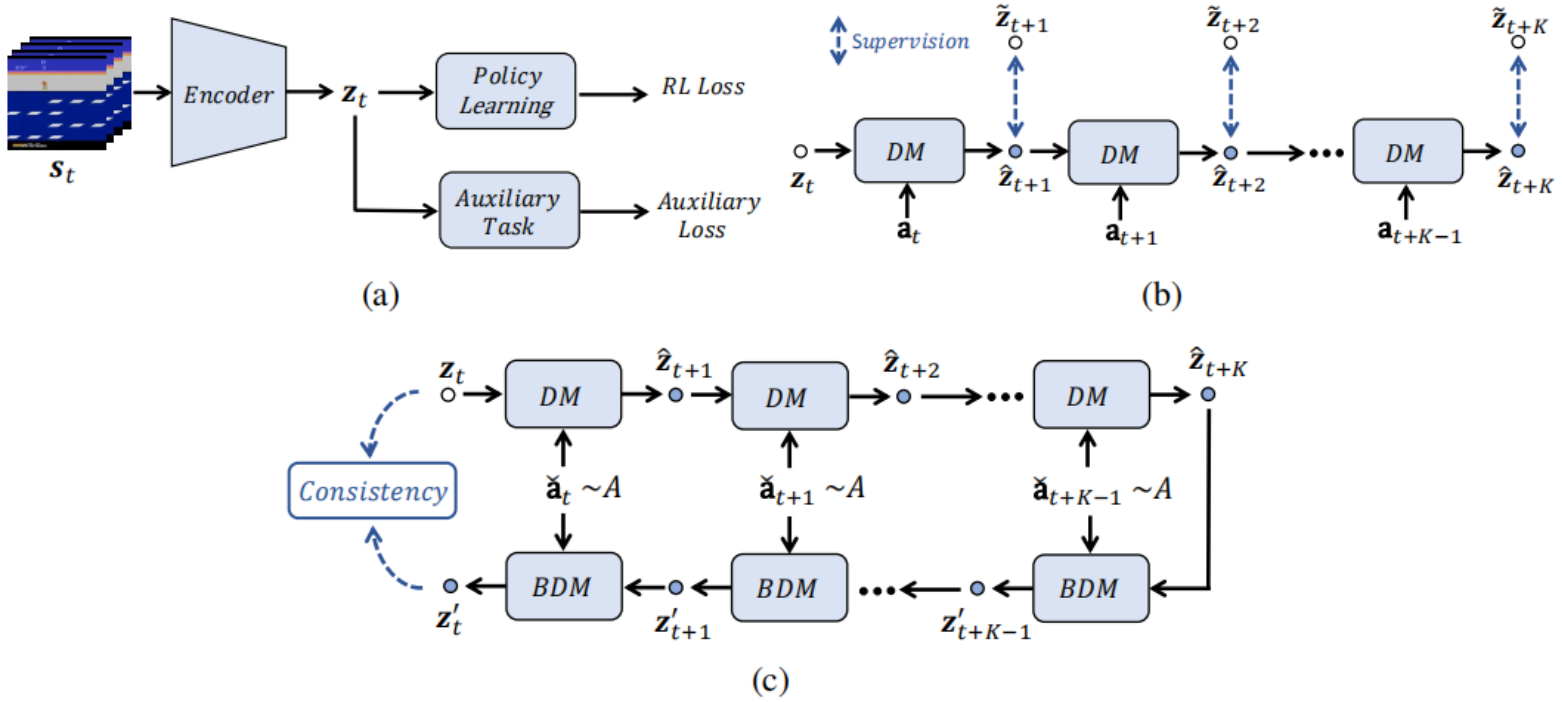
Method:Overall Objective

The loss of Rainbow or SAC.

$$\mathcal{L}_{total} = \mathcal{L}_{rl} + \lambda_{pred}\mathcal{L}_{pred} + \lambda_{cyc}\mathcal{L}_{cyc}$$

$$\mathcal{L}_{pred} = \sum_{k=1}^{K} d(\hat{\mathbf{z}}_{t+k}, \tilde{\mathbf{z}}_{t+k})$$

$$\mathcal{L}_{cyc} = \frac{1}{M}\sum_{m=1}^{M} d_{\mathcal{M}}(\mathbf{z}'_t, \mathbf{z}_t)$$



Algorithm

$$\mathcal{L}_{total} = \mathcal{L}_{rl} + \lambda_{pred}\mathcal{L}_{pred} + \lambda_{cyc}\mathcal{L}_{cyc}$$

Algorithm 1 Training Algorithm for PlayVirtual

Require: denote parameters of an encoder f, a dynamics model h, a backward dynamics model b and a policy learning head π, as θ_f, ξ_h, ξ_b and ω, respectively;

denote the number of prediction steps as K, the number of virtual trajectories as M;
denote the prediction loss weight and the predefined maximum weight for cycle consistency loss as λ_{pred} and λ_{cyc}^{max}, respectively;
denote the warmup end iteration as i_{end};.
denote the replay buffer as $\mathcal{D}$;
denote the interaction step index for Atari and the environment step index for DMControl as i;
randomly initialize all network parameters and make the reply buffer empty.
while *train* **do**
  determine the action $\mathbf{a} \sim \pi(f(\mathbf{s}))$ (based on policy) and interact with environment
  record/collect experience $\mathcal{D} \leftarrow \mathcal{D} \cup (\mathbf{s}, \mathbf{a}, \mathbf{s}_{next}, r)$
  sample a sequence of $(\mathbf{s}, \mathbf{a}, \mathbf{s}_{next}, r) \sim \mathcal{D}$
  $\mathcal{L}_{cyc} \leftarrow 0$; $\mathcal{L}_{pred} \leftarrow 0$; $\mathcal{L}_{rl} \leftarrow 0$
  $\mathbf{z}_t \leftarrow f(\mathbf{s}_t)$
  for $j = 1, 2, ..., M$ **do**
    $\{\check{\mathbf{a}}_t^{(j)}, \check{\mathbf{a}}_{t+1}^{(j)}, \ldots, \check{\mathbf{a}}_{t+K-1}^{(j)}\} \sim \mathcal{A}$ ▷ randomly sample a sequence of actions
    $\hat{\mathbf{z}}_t^{(j)} \leftarrow \mathbf{z}_t$
    for $k = 0, 1, ..., K-1$ **do**
      $\hat{\mathbf{z}}_{t+k+1}^{(j)} \leftarrow h(\hat{\mathbf{z}}_{t+k}^{(j)}, \check{\mathbf{a}}_{t+k}^{(j)})$ ▷ (forward) dynamics prediction
    end for
    $\mathbf{z}'^{(j)}_{t+K} \leftarrow \hat{\mathbf{z}}_{t+K}^{(j)}$
    for $k = K-1, K-2, ..., 0$ **do**
      $\mathbf{z}'^{(j)}_{t+k} \leftarrow b(\mathbf{z}'^{(j)}_{t+k+1}, \check{\mathbf{a}}_{t+k}^{(j)})$ ▷ backward dynamics prediction
    end for
    $\mathcal{L}_{cyc} \leftarrow \mathcal{L}_{cyc} + d(\mathbf{z}_t'^{(j)}, \mathbf{z}_t^{(j)})$ ▷ calculate cycle-consistency loss
  end for
  $\mathcal{L}_{cyc} \leftarrow \mathcal{L}_{cyc}/M$
  calculate the forward prediction loss $\mathcal{L}_{pred}$ according to Eq. (2)
  calculate the RL loss $\mathcal{L}_{rl}$
  warmup λ_{cyc} based on $\lambda_{cyc}^{max}, i_{end}, i$
  $\mathcal{L}_{total} \leftarrow \mathcal{L}_{rl} + \lambda_{pred}\mathcal{L}_{pred} + \lambda_{cyc}\mathcal{L}_{cyc}$
  $\theta_f, \xi_h, \xi_b, \omega \leftarrow Optimize((\theta_f, \xi_h, \xi_b, \omega), \mathcal{L}_{total})$
end while

collect interaction data

Cycle Consistency

train the DM and policy

Experiments:Comparison on Atari

Human-Normalized Score(HNS): $\frac{S_A - S_R}{S_H - S_R}$

Table 1: Scores achieved by different methods on Atari-100k. We also report median HNS. We run our PlayVirtual with 15 random seeds given that this benchmark is susceptible to high variance across multiple runs. Note that here we report the results of SPR [40] copied from their paper (*i.e.*, 41.5%), which is much higher than our reproduced results using their released source code (*i.e.*, 37.1%).

Game	Human	Random	SimPLe[61]	DER[45]	OTR[23]	CURL[29]	DrQ[50]	SPR[40]	PlayVirtual
Alien	7127.7	227.8	616.9	739.9	824.7	558.2	771.2	801.5	**947.8**
Amidar	1719.5	5.8	88.0	**188.6**	82.8	142.1	102.8	176.3	165.3
Assault	742.0	222.4	527.2	431.2	351.9	600.6	452.4	571.0	**702.3**
Asterix	8503.3	210.0	**1128.3**	470.8	628.5	734.5	603.5	977.8	933.3
Bank Heist	753.1	14.2	34.2	51.0	182.1	131.6	168.9	**380.9**	245.9
Battle Zone	37187.5	2360.0	5184.4	10124.6	4060.6	14870.0	12954.0	**16651.0**	13260.0
Boxing	12.1	0.1	9.1	0.2	2.5	1.2	6.0	35.8	**38.3**
Breakout	30.5	1.7	16.4	1.9	9.8	4.9	16.1	17.1	**20.6**
Chopper Command	7387.8	811.0	**1246.9**	861.8	1033.3	1058.5	780.3	974.8	922.4
Crazy Climber	35829.4	10780.5	**62583.6**	16185.3	21327.8	12146.5	20516.5	42923.6	23176.7
Demon Attack	1971.0	152.1	208.1	508.0	711.8	817.6	1113.4	545.2	**1131.7**
Freeway	29.6	0.0	20.3	**27.9**	25.0	26.7	9.8	24.4	16.1
Frostbite	4334.7	65.2	254.7	866.8	231.6	1181.3	331.1	1821.5	**1984.7**
Gopher	2412.5	257.6	771.0	349.5	**778.0**	669.3	636.3	715.2	684.3
Hero	30826.4	1027.0	2656.6	6857.0	6458.8	6279.3	3736.3	7019.2	**8597.5**
Jamesbond	302.8	29.0	125.3	301.6	112.3	**471.0**	236.0	365.4	394.7
Kangaroo	3035.0	52.0	323.1	779.3	605.4	872.5	940.6	**3276.4**	2384.7
Krull	2665.5	1598.0	**4539.9**	2851.5	3277.9	4229.6	4018.1	3688.9	3880.7
Kung Fu Master	22736.3	258.5	**17257.2**	14346.1	5722.2	14307.8	9111.0	13192.7	14259.0
Ms Pacman	6951.6	307.3	**1480.0**	1204.1	941.9	1465.5	960.5	1313.2	1335.4
Pong	14.6	-20.7	**12.8**	-19.3	1.3	-16.5	-8.5	-5.9	-3.0
Private Eye	69571.3	24.9	58.3	97.8	100.0	**218.4**	-13.6	124.0	93.9
Qbert	13455.0	163.9	1288.8	1152.9	509.3	1042.4	854.4	669.1	**3620.1**
Road Runner	7845.0	11.5	5640.6	9600.0	2696.7	5661.0	8895.1	**14220.5**	13534.0
Seaquest	42054.7	68.4	**683.3**	354.1	286.9	384.5	301.2	583.1	527.7
Up N Down	11693.2	533.4	3350.3	2877.4	2847.6	2955.2	3180.8	**28138.5**	10225.2
Median HNS (%)	100	0	14.4	16.1	20.4	17.5	26.8	41.5	**47.2**

Experiments:Comparison on DMControl

Table 2: Scores (mean and standard deviation) achieved by different methods on the DMControl-100k and DMControl-500k. We run our PlayVirtual with 10 random seeds. Note that SPR [40] is originally designed only for discrete control. For the continuous-control environments, we extend SPR to a new version named SPR†.

100k Step Scores	**PlaNet**[16]	**Dreamer**[17]	**SAC+AE**[49]	**SLAC**[30]	**CURL**[29]	**DrQ** [50]	**SPR†**[40]	**PlayVirtual**
Finger, spin	136 ± 216	341 ± 70	740 ± 64	693 ± 141	767 ± 56	901 ± 104	868 ± 143	$\mathbf{915 \pm 49}$
Cartpole, swingup	297 ± 39	326 ± 27	311 ± 11	-	582 ± 146	759 ± 92	799 ± 42	$\mathbf{816 \pm 36}$
Reacher, easy	20 ± 50	314 ± 155	274 ± 14	-	538 ± 233	601 ± 213	638 ± 269	$\mathbf{785 \pm 142}$
Cheetah, run	138 ± 88	235 ± 137	267 ± 24	319 ± 56	299 ± 48	344 ± 67	467 ± 36	$\mathbf{474 \pm 50}$
Walker, walk	224 ± 48	277 ± 12	394 ± 22	361 ± 73	403 ± 24	$\mathbf{612 \pm 164}$	398 ± 165	460 ± 173
Ball in cup, catch	0 ± 0	246 ± 174	391 ± 82	512 ± 110	769 ± 43	913 ± 53	861 ± 233	$\mathbf{926 \pm 31}$
Median Score	137.0	295.5	351.0	436.5	560.0	685.5	719.0	**800.5**
500k Step Scores								
Finger, spin	561 ± 284	796 ± 183	884 ± 128	673 ± 92	926 ± 45	938 ± 103	924 ± 132	$\mathbf{963 \pm 40}$
Cartpole, swingup	475 ± 71	762 ± 27	735 ± 63	-	841 ± 45	868 ± 10	$\mathbf{870 \pm 12}$	865 ± 11
Reacher, easy	210 ± 390	793 ± 164	627 ± 58	-	929 ± 44	942 ± 71	925 ± 79	$\mathbf{942 \pm 66}$
Cheetah, run	305 ± 131	570 ± 253	550 ± 34	640 ± 19	518 ± 28	660 ± 96	716 ± 47	$\mathbf{719 \pm 51}$
Walker, walk	351 ± 58	897 ± 49	847 ± 48	842 ± 51	902 ± 43	921 ± 45	916 ± 75	$\mathbf{928 \pm 30}$
Ball in cup, catch	460 ± 380	879 ± 87	794 ± 58	852 ± 71	959 ± 27	963 ± 9	963 ± 8	$\mathbf{967 \pm 5}$
Median Score	405.5	794.5	764.5	757.5	914.0	929.5	920.0	**935.0**

Experiments:Ablation Studies

Table 3: Effectiveness of PlayVirtual on top of *Baseline*, which is SPR [40] for discrete control on Atari, and SPR† for continual control on DMControl. "w/o Pred" denotes disabling future prediction in *Baseline*. *Baseline+BDM* denotes the scheme that a BDM is incorporated into *Baseline*.

Model	Atari-100k	DMControl-100k
Baseline w/o Pred	33.4	680.0
Baseline	37.1	728.0
Baseline+BDM	38.4	741.0
PlayVirtual	**47.2**	**797.0**

Table 4: Influence of prediction steps K for our PlayVirtual and the baseline scheme SPR/SPR†.

Benchmark	Model	K=0	K=3	K=6	K=9	K=12
Atari-100k	SPR	33.4	33.9	35.2	37.1	34.9
	PlayVirtual	33.4	34.8	39.2	**47.2**	43.1
DMC-100k	SPR†	664.0	725.0	723.0	728.0	721.5
	PlayVirtual	664.0	775.5	**797.0**	795.0	794.5

Table 5: Impact of the augmentation of cycle-consistent virtual trajectories on feature representation learning. *PlayVirtual-ND* denotes that we do not use the cycle consistency loss over virtual trajectories to update the dynamic model.

Model	Atari-100k	DMControl-100k
Baseline	37.1	723.0
PlayVirtual-ND	44.0	777.5
PlayVirtual	**47.2**	**797.0**

Experiments:Ablation Studies

Table 6: Influence of distance metric space $\mathcal{M}$. $\mathcal{M}_{latent}$ and $\mathcal{M}_{proj}$ denote the use of the latent feature space and the "projection" space, respectively.

Model	Atari-100k	DMControl-100k
Baseline	37.1	728.0
PlayVirtual($\mathcal{M}_{latent}$)	44.8	**798.5**
PlayVirtual($\mathcal{M}_{proj}$)	**47.2**	797.0

Table 7: Influence of the number of virtual trajectories M.

Atari-100k					
M	0	$\|\mathcal{A}\|$	$2\|\mathcal{A}\|$	$3\|\mathcal{A}\|$	
Median HNS(%)	37.1	39.5	**47.2**	42.5	
DMControl-100k					
M	0	1	10	20	30
Median Score	723.0	770.5	797.0	**806.0**	792

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]

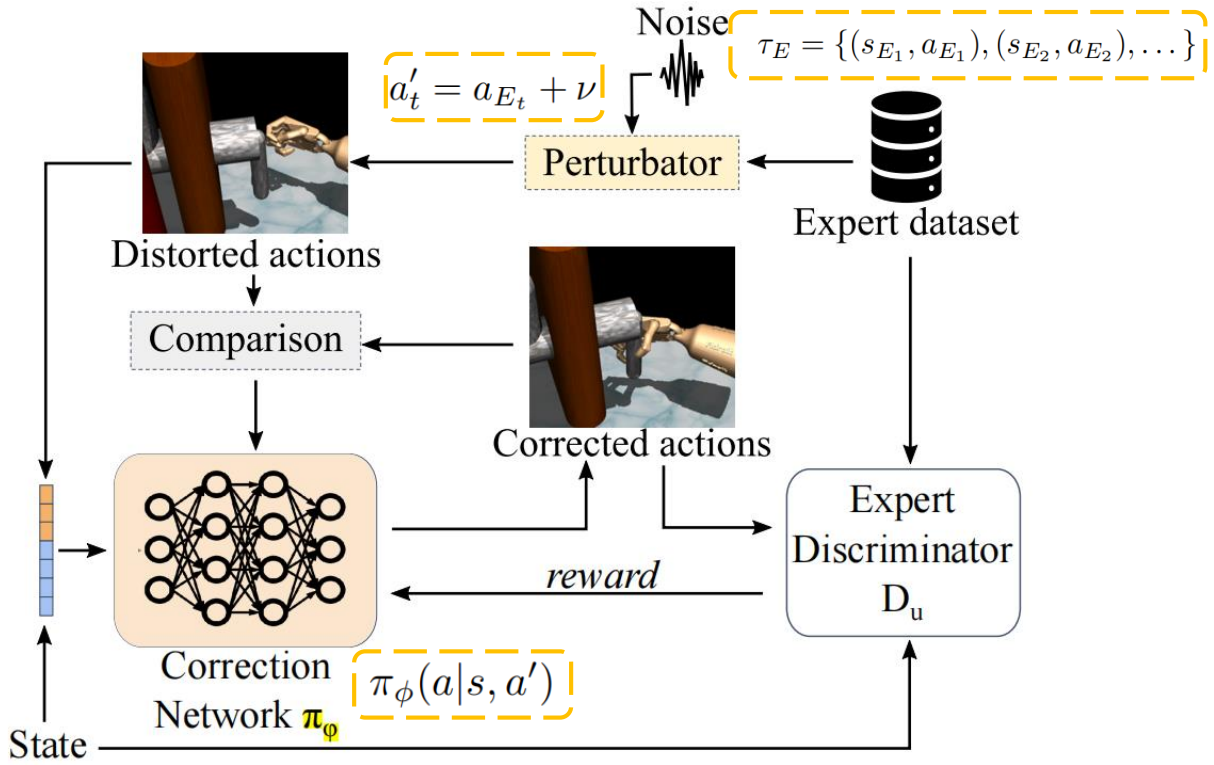
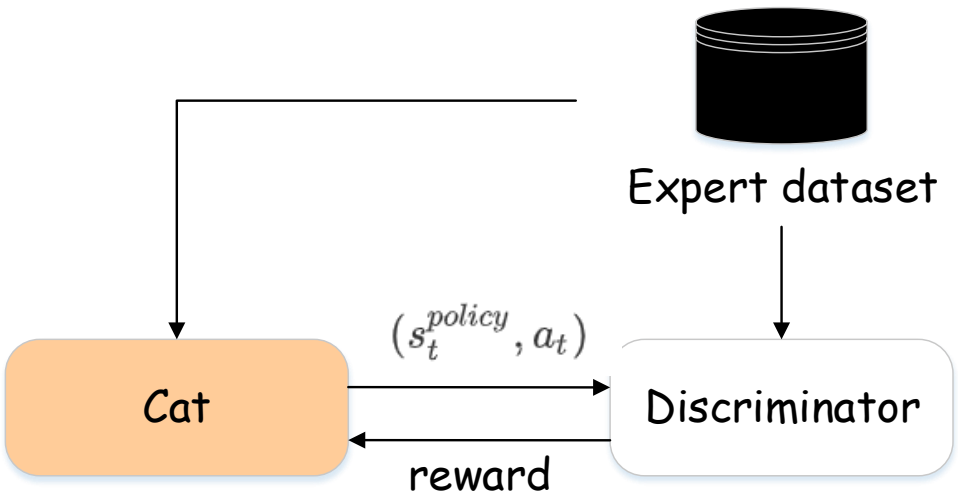


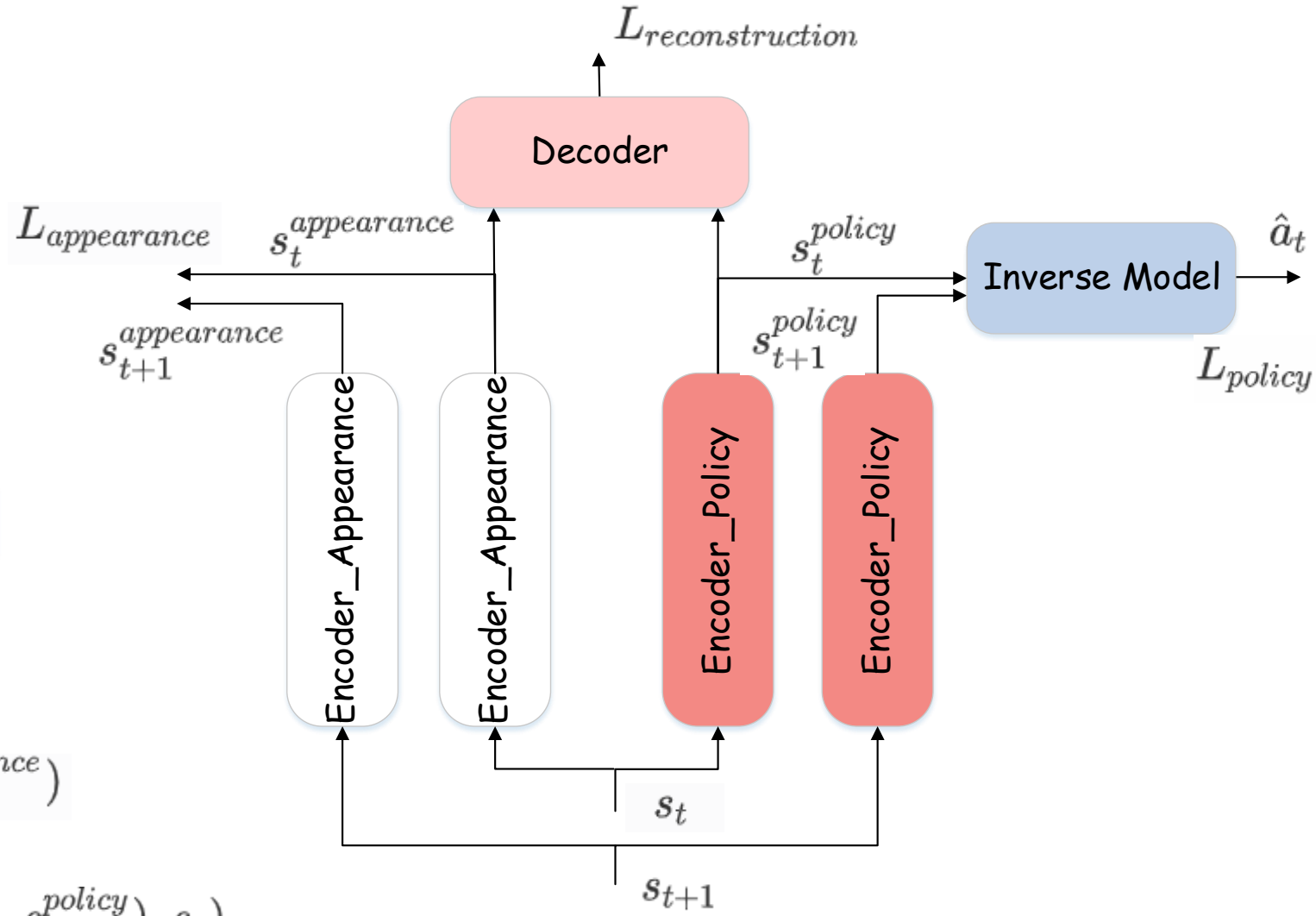
Fig. 3. Detailed overview of stage 1, which performs Corrected Augmentation For Trajectories. The architecture is semi-supervised since it is guided by unlabelled distorted actions.

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

Discussion

- EncDecInv-Cat





$$L_{appearance} = cosin(s_t^{appearance}, s_{t+1}^{appearance})$$

$$L_{policy} = mse(\hat{a}_t, a_t)$$

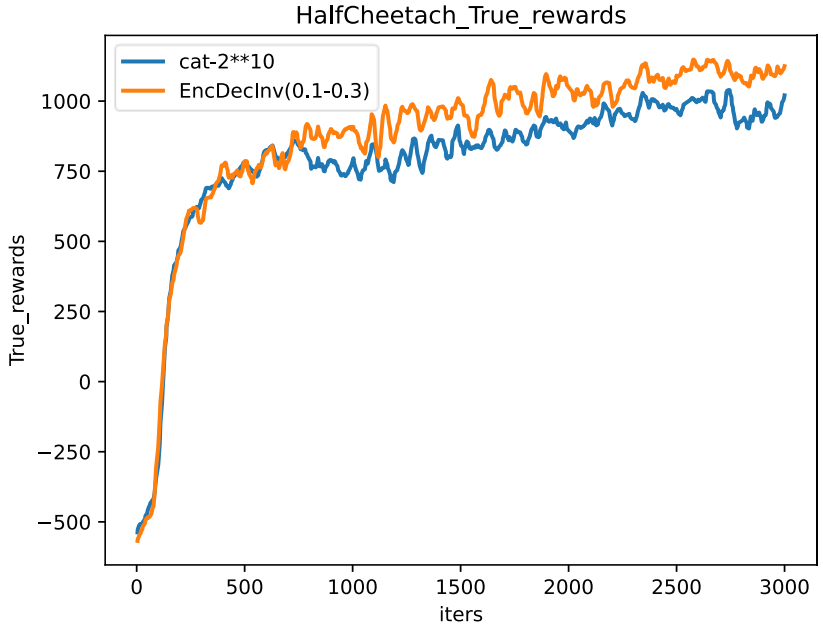
$$L_{reconstruction} = L_1(Decoder(s_t^{appearance}, s_t^{policy}), s_t)$$

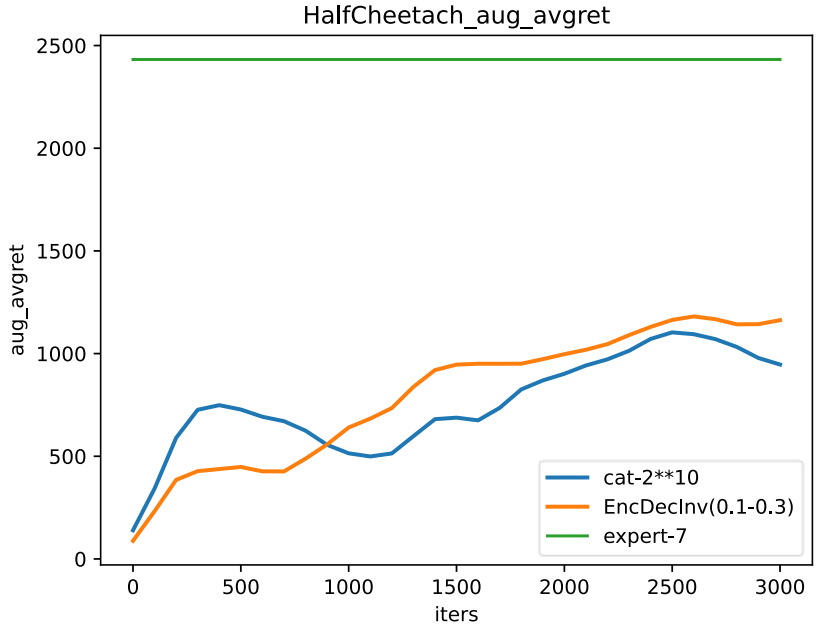
$$L_{total} = L_{policy} + \lambda L_{appearance} + \beta L_{reconstruction}$$

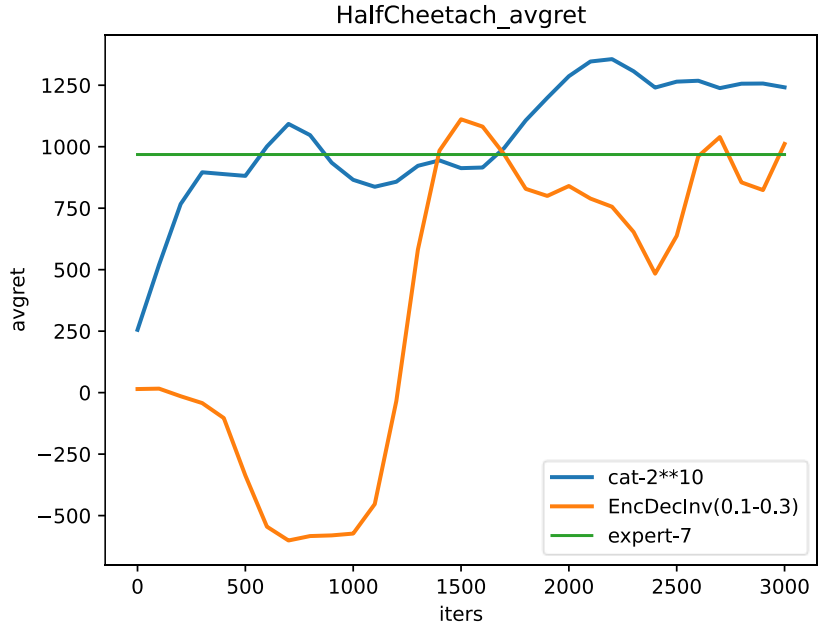
Discussion

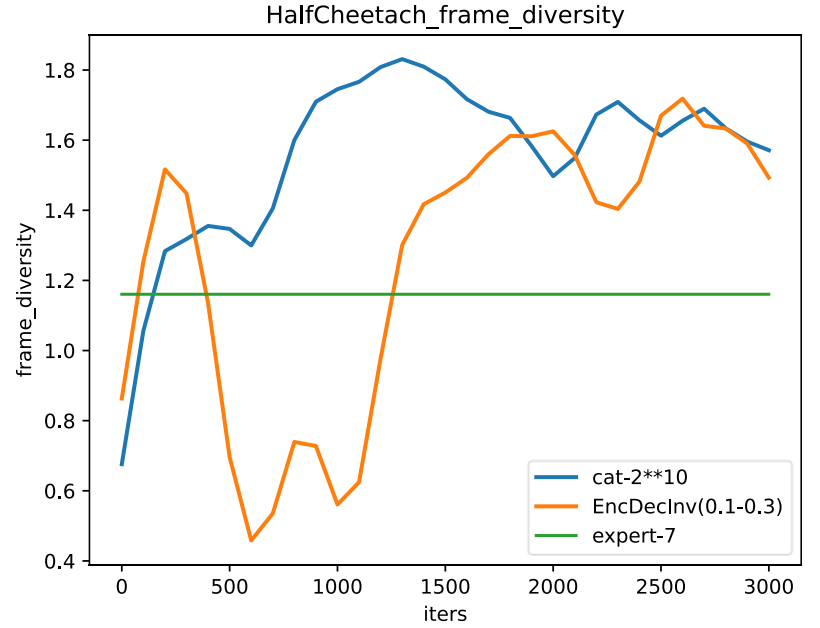
$$\overline{dtw}_n(\mathcal{T}_g) = \frac{\overline{dtw}(\mathcal{T}_g)}{\overline{dtw}(\mathcal{T}_E)},$$

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









Thanks