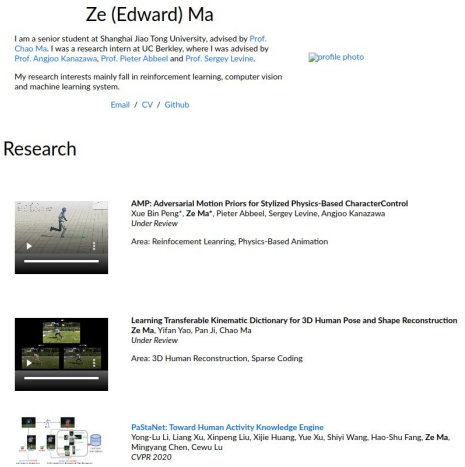
Adversarial Motion Priors for Stylized Physics-Based Character Control

Presented by Yigit YILDIRIM

Authors

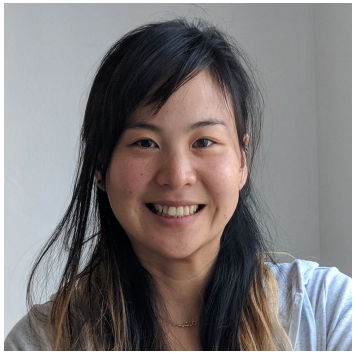
Xue Bin Peng (1)



Ze Ma (2)

Pieter Abbeel (1)

Sergey Levine (1)



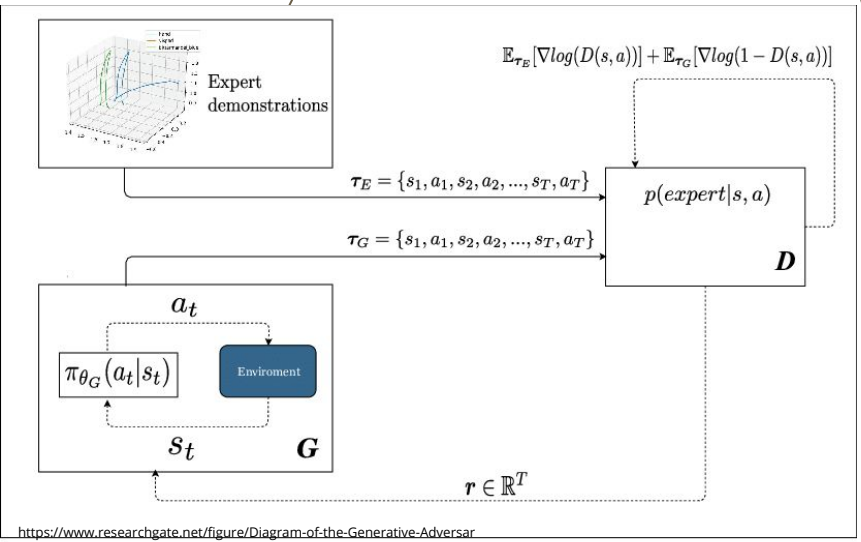
Angjoo Kanazawa (1)

Sim-to-real transfer of robotic control with dynamics randomization XB Peng, M Andrychowicz, W Zaremba, P Abbeel 2018 IEEE international conference on robotics and automation (ICRA), 3803-3810	537	2018
Deepmimic: Example-guided deep reinforcement learning of physics-based character skills XB Peng, P Abbeel, S Levine, M van de Panne ACM Transactions on Graphics (TOG) 37 (4), 1-14	384	2018
Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning XB Peng, G Berseth, KK Yin, M Van De Panne ACM Transactions on Graphics (TOG) 36 (4), 1-13	371	2017
Terrain-adaptive locomotion skills using deep reinforcement learning XB Peng, G Berseth, M Van de Panne ACM Transactions on Graphics (TOG) 35 (4), 1-12	220	2016
Variational discriminator bottleneck: Improving imitation learning, inverse rl, and gans by constraining information flow XB Peng, A Kanazawa, S Toyer, P Abbeel, S Levine arXiv preprint arXiv:1810.00821	101	2018
Sfv: Reinforcement learning of physical skills from videos XB Peng, A Kanazawa, J Malik, P Abbeel, S Levine ACM Transactions On Graphics (TOG) 37 (6), 1-14	99	2018
Learning locomotion skills using deeprl: Does the choice of action space matter? XB Peng, M van de Panne Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation	73	2017

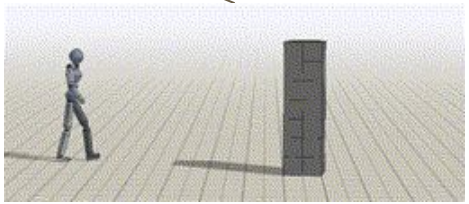
(1): UC Berkeley
(2): Shanghai Jiao Tong

Introduction

Adversarial Motion Priors for Stylized Physics-Based Character Control



https://www.researchgate.net/figure/Diagram-of-the-Generative-Adversarial-Imitation-approach-for-learning-robotics-tasks-from_fig8_334983907



Style: General characteristics of a motion presented in the dataset

Motivation

Aim: Synthesize natural and life-like motions on virtual agents.

Physics-based methods

- Utilize equations of motions
- (Disadv.) Cannot generate motion as natural as kinematic methods
- (Adv.) Works well in case of novel situations

Kinematic methods

- Utilize datasets of motion clips
- (Adv.) Generative models (GP, NN) can generate realistic motions
- (Disadv.) Cannot deal with novel situations when data is scarce

AMP

Related Work

- Physics-based methods:
  - Optimization techniques: Increase (optimize for) naturalness, (such as RL)
    - Quantitative metrics of naturalness is difficult to identify
    - There are heuristics: symmetry, stability, effort minimization

- Kinematic methods:
  - Data-driven techniques: Exploit motion clips from real-life actors
    - Imitation learning: imitation objective implemented as tracking objective
      - Minimize pose error
      - Phase variable to synchronize. Cannot follow multiple motions
    - Recent methods explicitly provide target poses to allow the imitation of different motions
      - Require motion planner to select motion-clips (trajectories) to be followed
      - Planner requires annotation, processing of clips

Related Work

- IRL: Reward learning from expert demonstrations.
  - Learning imitation objectives from data
  - Can be notoriously unstable

- Latent Space Models:
  - An alternative method to generate real-life behaviors are latent space models.
  - Specify controls through a learned latent representation
  - Latent representation is learned through pretraining
    - Using supervised learning or RL to encode behaviors
    - After pretraining, these representations are used to create hierarchical controllers

Background

- Main goal: Can we train agents to achieve goal-directed tasks while maintaining a certain style?

- This system combines classical policy gradient RL technique with Generative Adversarial Networks (GAN)

So it has 2 parts:

1. RL for achieving the goal
2. GAN for adhering to the style given in the dataset

Background - Goal-conditioned RL

Goal-conditioned RL

- Objective: $$J(\pi) = \mathbb{E}_{p(\mathrm{g})} \mathbb{E}_{p(\tau|\pi,\mathrm{g})} \left[\sum_{t=0}^{T-1} \gamma^t r_t \right]$$

- $g \in G$, $g \sim p(g)$
- Given the goal and current policy, agent samples an action $a_t \in A$ from policy $a_t \sim \pi(a_t | s_t, g)$
- $p(\tau | \pi, g)$ represents the likelihood of the trajectory
  - $\tau = \{(s_t, a_t, r_t)^{T-1}, s_T\}$
- Agent tries to maximize expected discounted return $J(\pi)$
- Let us call this reward r^G

Background - Goal-conditioned RL

Goal-conditioned RL: r^G examples

- Striking

$$r_t^G = \begin{cases} 1, & \text{target has been hit} \\ 0.3\, r_t^{\text{near}} + 0.3, & ||\mathrm{x}^* - \mathrm{x}_t^{\text{root}}|| < 1.375m \\ 0.3\, r_t^{\text{far}}, & \text{otherwise} \end{cases}$$

- Target location

$$r_t^G = 0.7 \exp\left(-0.5||\mathrm{x}^* - \mathrm{x}_t^{\text{root}}||^2\right) + 0.3 \exp\left(-\left(\max\left(0,\, v^* - \mathrm{d}_t^* \cdot \dot{\mathrm{x}}_t^{\text{com}}\right)\right)^2\right)$$

Background - Adversarial Motion Priors

Generative Adversarial Imitation Learning (GAIL):

AMP uses GAIL framework with some modifications.

- GAIL adapts techniques from GAN to imitation learning
- GAIL objective:

$$\arg\min_{D} \; -\mathbb{E}_{d^{\mathcal{M}}(\mathrm{s},\mathrm{a})}\left[\log\left(D(\mathrm{s},\mathrm{a})\right)\right] - \mathbb{E}_{d^{\pi}(\mathrm{s},\mathrm{a})}\left[\log\left(1 - D(\mathrm{s},\mathrm{a})\right)\right]$$

real data

fake data

- The policy is trained using

$$r_t = -\log\left(1 - D(\mathrm{s}_t, \mathrm{a}_t)\right)$$

Background - Adversarial Motion Priors

Modifications:

1. AMP uses raw motion clips that do not contain actions. So the discriminator (AMP) does not use (s, a) pairs but (s, s') pairs.

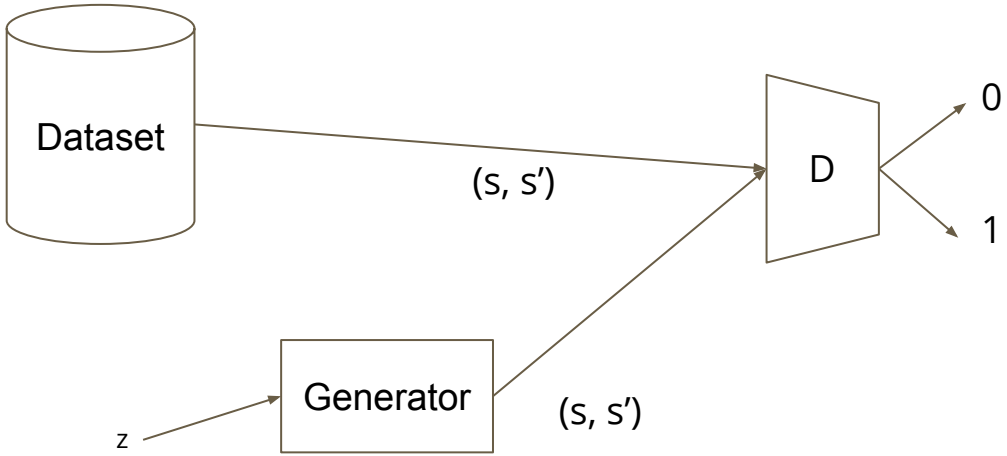
$$\arg\min_D -\mathbb{E}_{d^{\mathcal{M}}(s,s')}\left[\log\left(D(s,s')\right)\right] - \mathbb{E}_{d^{\pi}(s,s')}\left[\log\left(1-D(s,s')\right)\right]$$

2. Due to vanishing gradients problem in the standar GAN, more stable LSGAN is chosen. So the objective is changed into:

$$\arg\min_D \ \mathbb{E}_{d^{\mathcal{M}}(s,s')}\left[\left(D(s,s')-1\right)^2\right] + \mathbb{E}_{d^{\pi}(s,s')}\left[\left(D(s,s')+1\right)^2\right]$$

Background - Adversarial Motion Priors

Modifications:



The discriminator is trained to tell apart the transitions from the dataset and the one generated by the policy

Generator is not a standard one. It's the policy, interacting with the env.

Background - Adversarial Motion Priors

Modifications:

3. Discriminator is not given the raw states. A feature-mapping is applied to states. Features include:

1. Linear and angular velocity of root
2. Local rotation of each joint
3. Local velocity of each joint
4. 3D positions of end-effectors

$$\begin{aligned}\arg\min_D \quad & \mathbb{E}_{d^{\mathcal{M}}(s,s')}\left[\left(D(\Phi(s),\Phi(s'))-1\right)^2\right] \\ & + \mathbb{E}_{d^{\pi}(s,s')}\left[\left(D\left(\Phi(s),\Phi(s')\right)+1\right)^2\right] \\ & + \frac{w^{\mathrm{gp}}}{2}\,\mathbb{E}_{d^{\mathcal{M}}(s,s')}\left[\left\|\nabla_{\phi} D(\phi)\Big|_{\phi=(\Phi(s),\Phi(s'))}\right\|^2\right]\end{aligned}$$

4. Gradient penalty (?): To improve training stability, nonzero gradient on real data are penalized.

Background - Adversarial Motion Priors

Style Reward:

- Since the overall system is non-differentiable, output of the discriminator is used as the reward for training the policy.
- Calculated as follows:

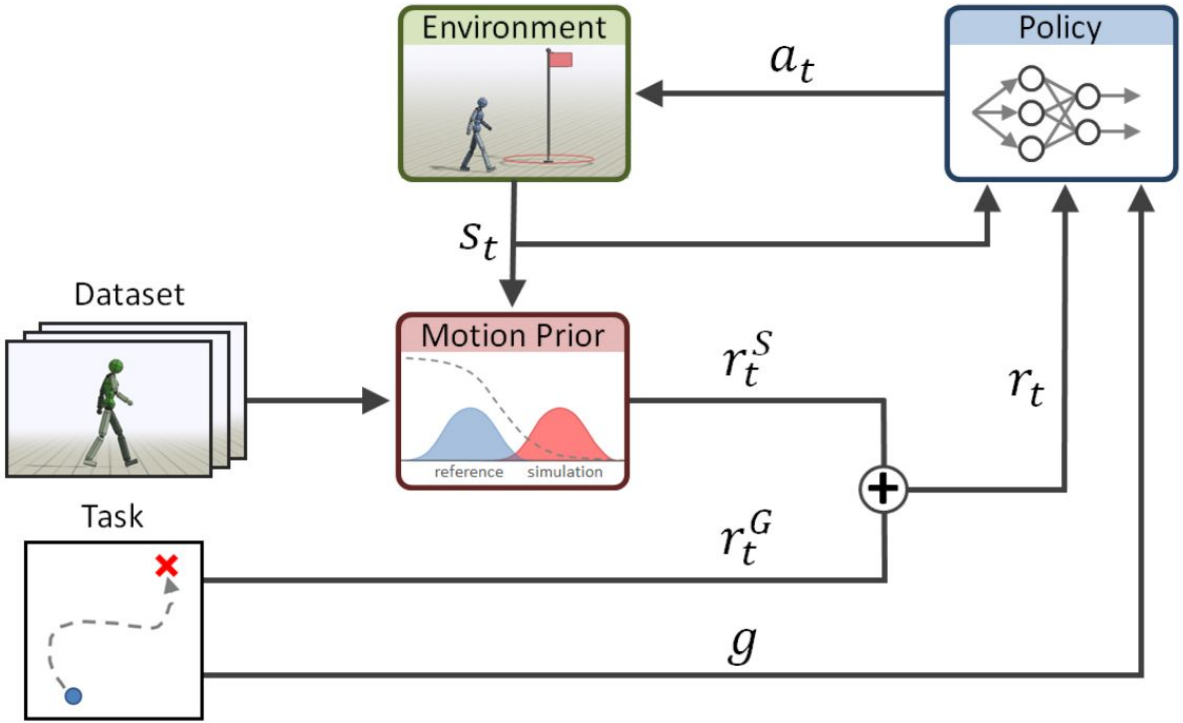
$$r(s_t, s_{t+1}) = \max\left[0,\ \ 1 - 0.25(D(s_t, s_{t+1}) - 1)^2\right]$$

- Let us call this reward r^S . The combined reward is calculated as a linear combination of r^S and r^G:

$$r(s_t, a_t, s_{t+1}, g) = w^G r^G(s_t, a_t, s_t, g) + w^S r^S(s_t, s_{t+1})$$

The overall system is as follows:

Background


Environment
Policy
a_t
s_t
Dataset
Motion Prior
reference
simulation
r_t^S
r_t
Task
r_t^G
g

Background

ALGORITHM 1: Training with AMP

input $\mathcal{M}$: dataset of reference motions
$D \leftarrow$ initialize discriminator
$\pi \leftarrow$ initialize policy
$V \leftarrow$ initialize value function
$\mathcal{B} \leftarrow \emptyset$ initialize reply buffer

while not done **do**
  for trajectory $i = 1, ..., m$ **do**
    $\tau^i \leftarrow \{(s_t, a_t, r_t^G)_{t=0}^{T-1}, s_T^G, g\}$ collect trajectory with π
    for time step $t = 0, ..., T-1$ **do**
      $d_t \leftarrow D(\Phi(s_t), \Phi(s_{t+1}))$
      $r_t^S \leftarrow$ calculate style reward according to Equation 7 using d_t
      $r_t \leftarrow w^G r_t^G + w^S r_t^S$
      record r_t in τ^i
    end for
    store τ^i in $\mathcal{B}$
  end for

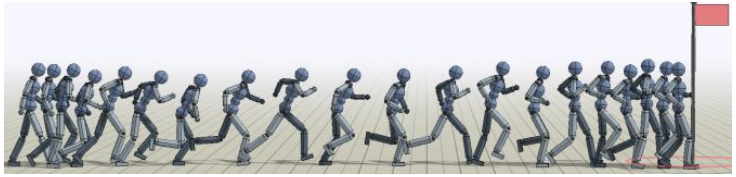
  for update step $= 1, ..., n$ **do**
    $b^{\mathcal{M}} \leftarrow$ sample batch of K transitions $\{(s_j, s'_j)\}_{j=1}^K$ from $\mathcal{M}$
    $b^{\pi} \leftarrow$ sample batch of K transitions $\{(s_j, s'_j)\}_{j=1}^K$ from $\mathcal{B}$
    update D according to Equation 8 using $b^{\mathcal{M}}$ and b^{π}
  end for

  update V and π using data from trajectories $\{\tau^i\}_{i=1}^m$
end while

Model Representation

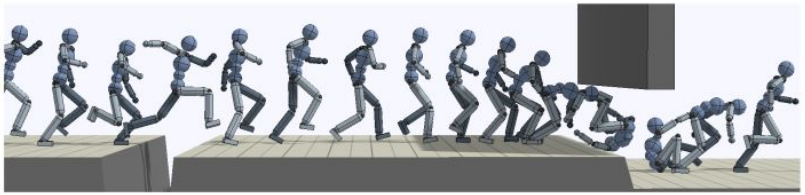
- s_t : Includes a set of features, all in the local frame (wrt root (pelvis))
  - Linear and angular velocity of root
  - Local rotation of each joint
  - Local velocity of each joint
  - 3D positions of end-effectors
- a_t : Specifies target positions for PD controller on each joint
- Networks:
  - Policy, value and discriminator functions are simple multilayer perceptrons.
  - MLP with 2 hidden layers (1024, 512 ReLU nodes)

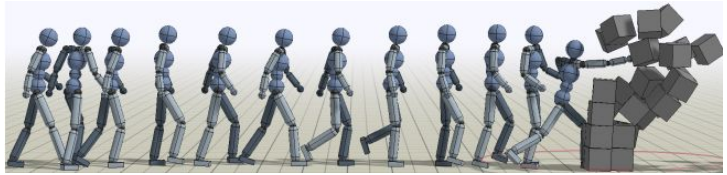
Results



Tasks:

1. Target heading: move along the target direction with a certain speed
2. Target location: move to a target location
3. Dribbling: dribble a soccer ball to a target location
4. Strike: punch a target with hands
5. Obstacles: traverse an obstacle-filled environment





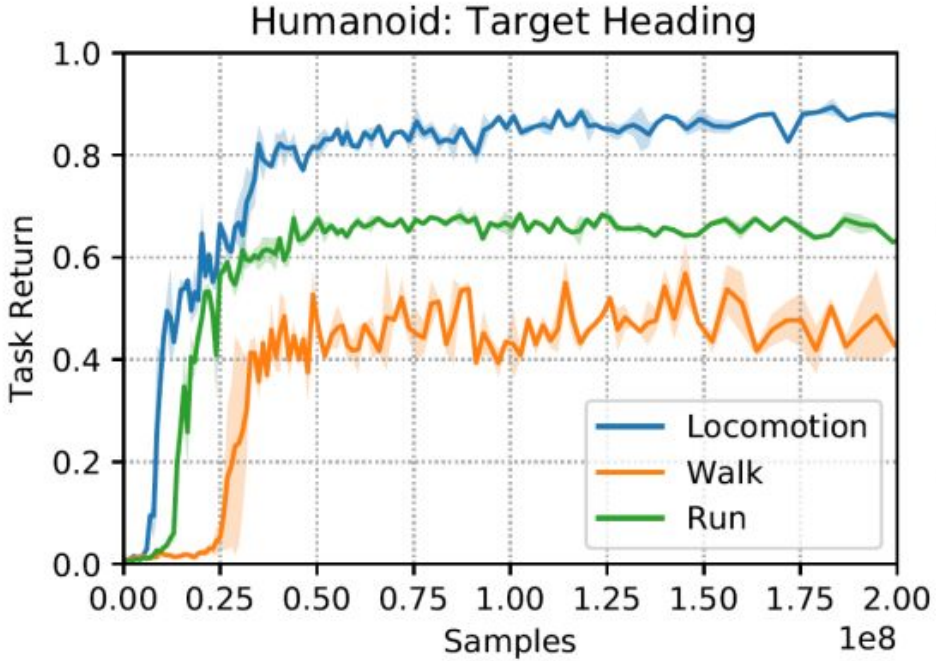
Dataset: Locomotion

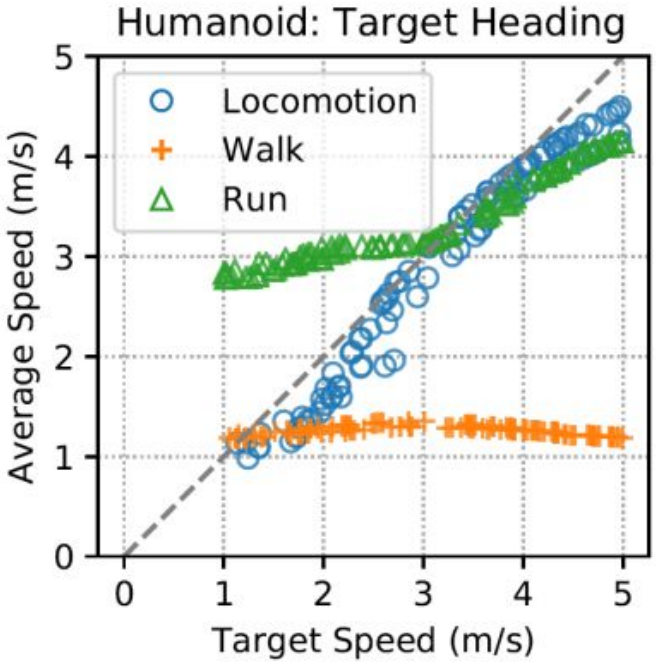
In addition to training with mocap clips
recorded from human actors,

Results - Emerged skills

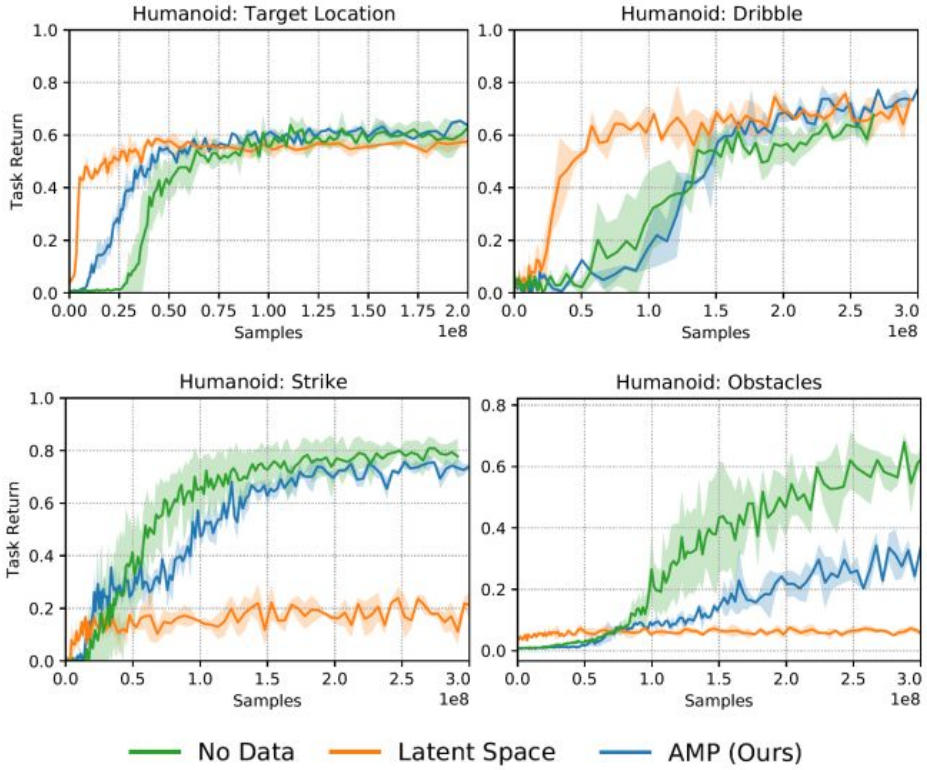
(f) Humanoid: Obstacles (Run + Leap + Roll)

Results


Humanoid: Target Heading
Task Return
1.0
0.8
0.6
0.4
0.2
0.0
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
Samples
1e8
Locomotion
Walk
Run


Humanoid: Target Heading
Average Speed (m/s)
5
4
3
2
1
0
0
1
2
3
4
5
Target Speed (m/s)
Locomotion
Walk
Run

Results - Comparison



Results - Limitations