SMU SINGAPORE MANAGEMENT UNIVERSITY | School of Computing and Information Systems

FlowPG: Action-constrained Policy Gradient with Normalizing Flows

Janaka Chathuranga Brahmanage, Jiajing Ling, Akshat Kumar
School of Computing and Information Systems
Singapore Management University
{janakat.2022, jjling.2018}@phdcs.smu.edu.sg, akshatkumar@smu.edu.sg

Presented by: Janaka Chathuranga Brahmanage

Motivating Examples in Constrained RL

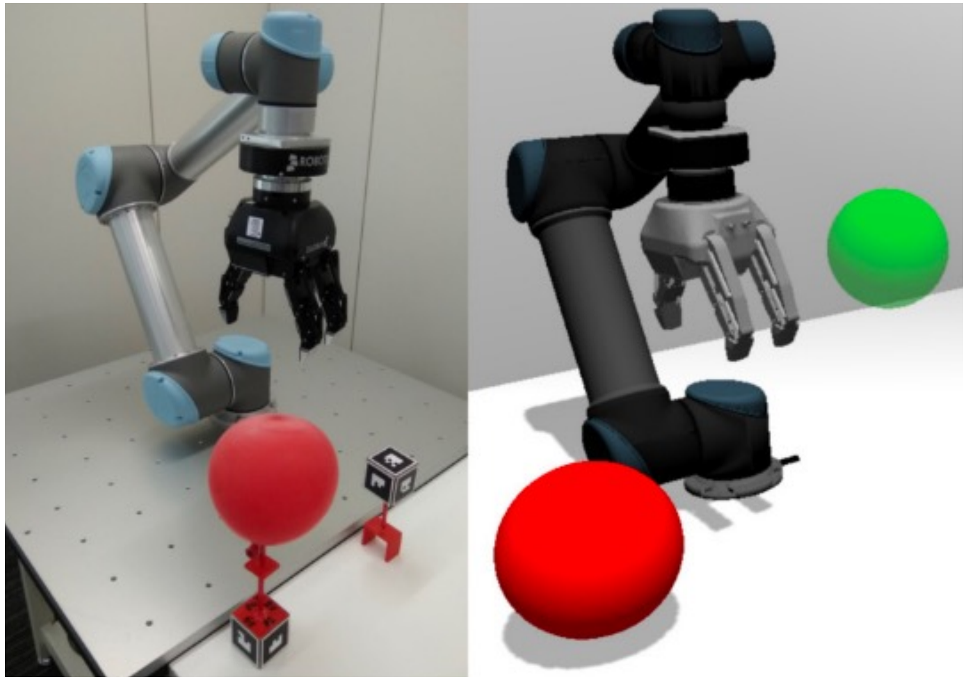


Image Source: Pham, T.-H., De Magistris, G., & Tachibana, R. (2018). OptLayer—Practical Constrained Optimization for Deep Reinforcement Learning in the Real World (arXiv:1709.07643). arXiv. http://arxiv.org/abs/1709.07643

To avoid collisions in robot arms

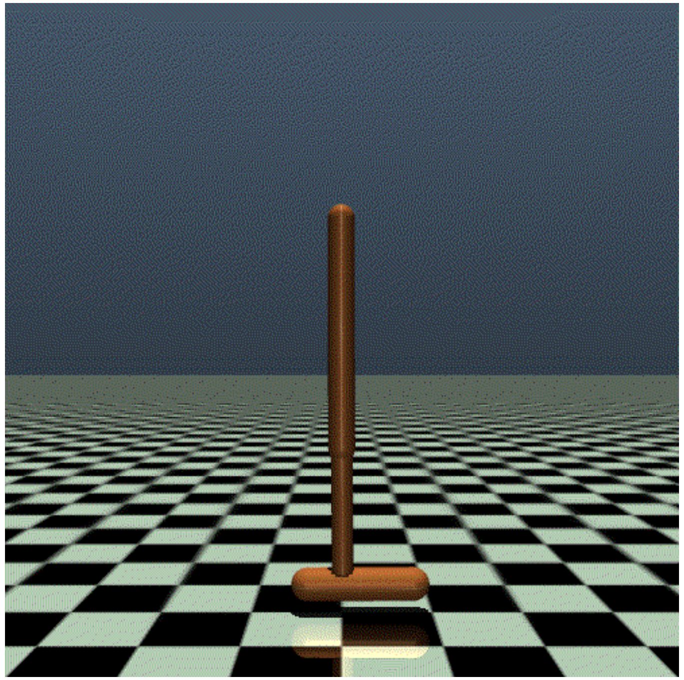
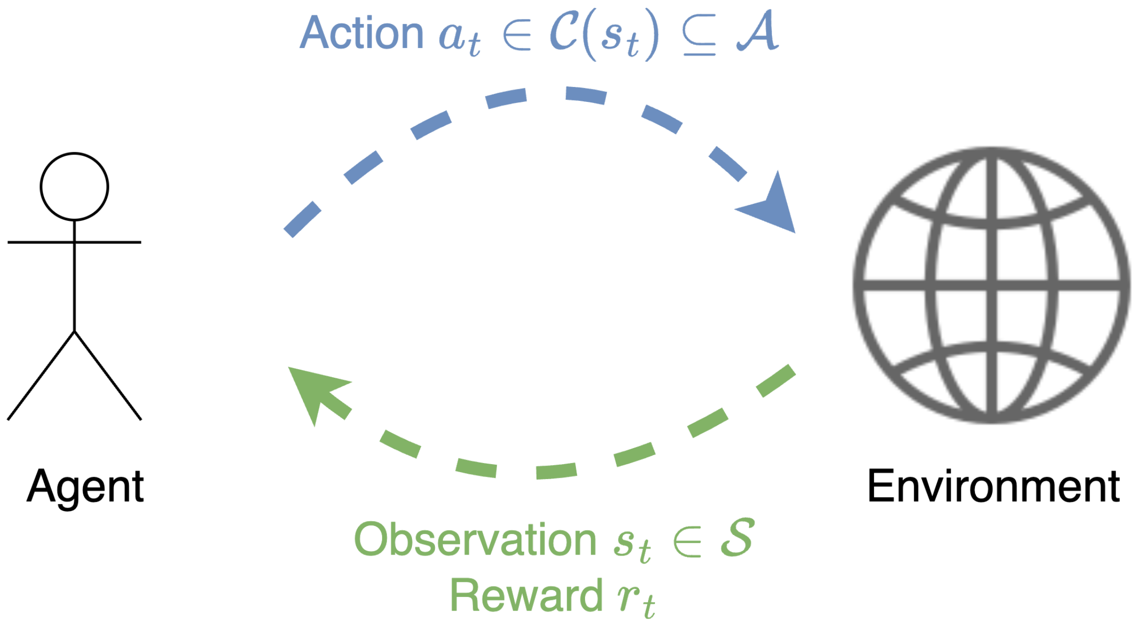


Image Source: Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. (2016). OpenAI Gym (arXiv:1606.01540). arXiv. https://doi.org/10.48550/arXiv.1606.01540

To describe physical limitations in simulators

Action Constrained Reinforcement Learning (ACRL)

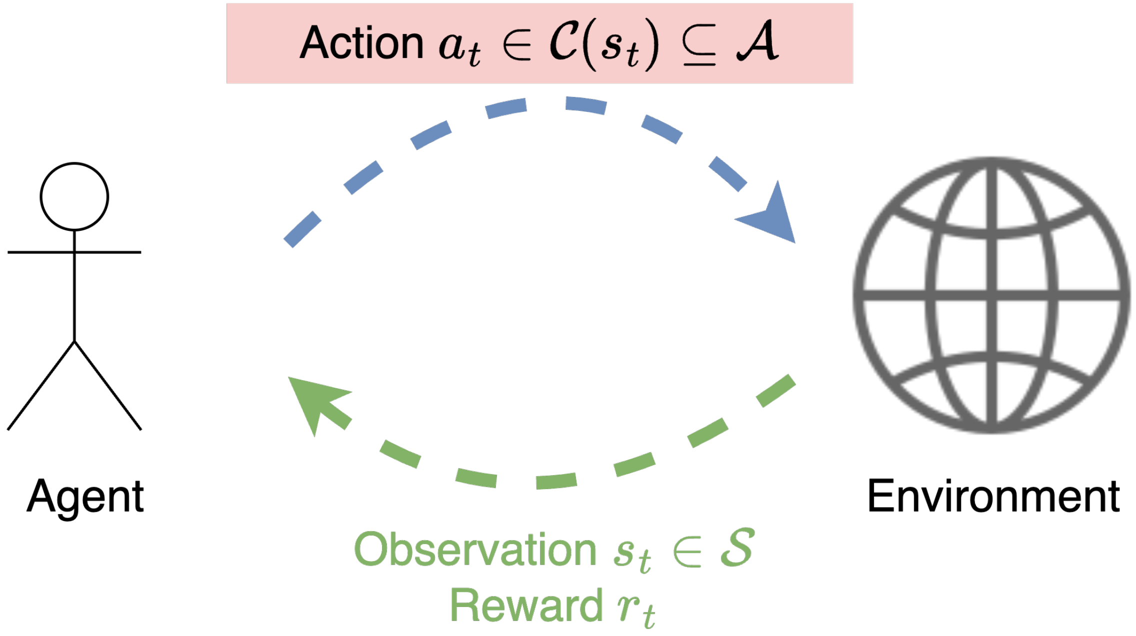
- Agents can only take actions from feasible subset of all actions based on the current state.



SMU SINGAPORE MANAGEMENT UNIVERSITY | School of Computing and Information Systems

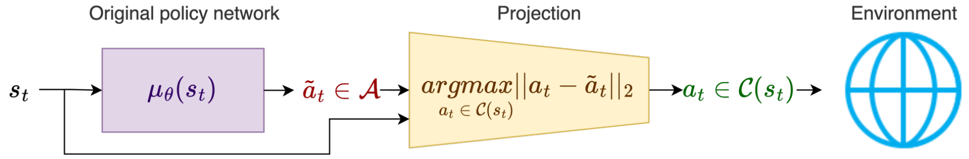
Action Constrained Reinforcement Learning (ACRL)

- Agents can only take actions from feasible subset of all actions based on the current state.



Previous Work

- A Projection Layer
  - Projection the action into the nearest action in the feasible region



  - Challenge: Zero gradient problem

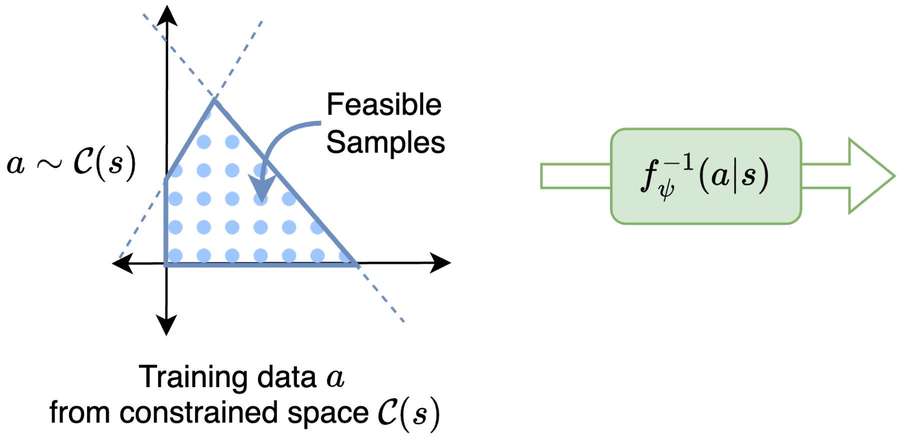
- NFWPO
  - Use Frank-Wolfe algorithm to update
  - Challenge: Significantly high run-time overhead due to Frank-Wolfe direction finding subproblem in the policy update.

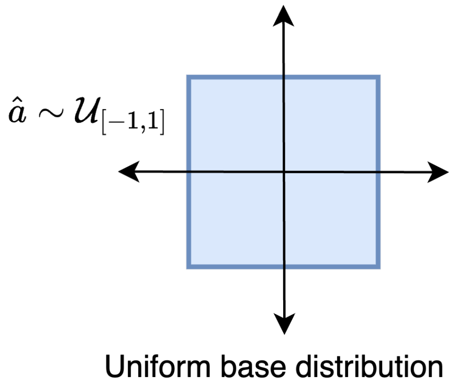
Our Contributions

- We develop multiple methods to generate samples from constrained space
  - Hamiltonian Monte-Carlo
  - Probabilistic decision diagrams (PSDD)
- We utilize normalizing flows, a type of generative model to learn a differentiable, invertible mapping between the samples from constrained space and a simple latent distribution.
- We propose a method to integrate normalizing-flow model with deep RL algorithms such as DDPG

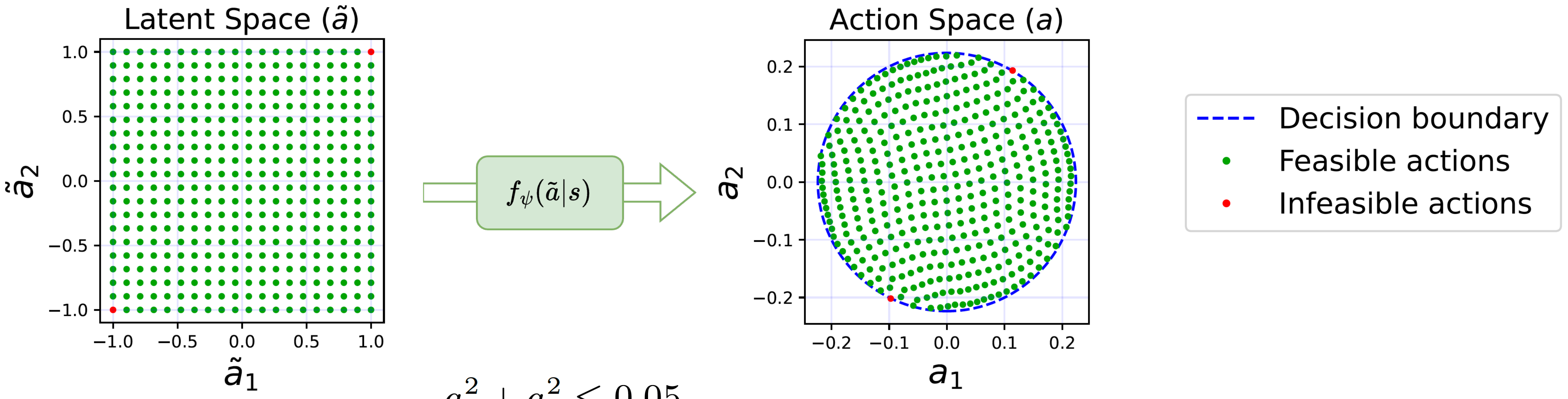
Training a Normalizing-Flow model to approximate the constrained space

- We first generate samples (s, a) from the feasible region,
  - Rejection Sampling
  - Hamiltonian Monte Carlo
- Map them to a **uniform base distribution** using a normalizing flow model and maximize log-likelihood





After training, the model serves as a differentiable, bijective mapping from a simple uniform distribution to the feasible region.

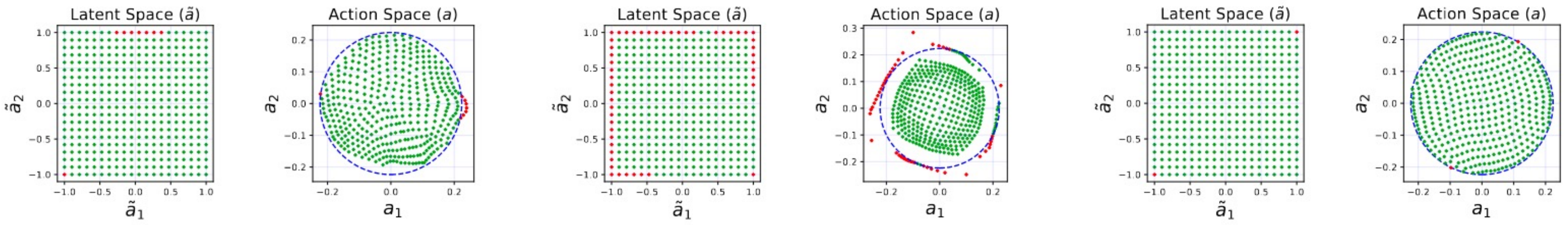


$$a_1^2 + a_2^2 \leq 0.05$$

Example Mapping: Reacher Env

Quality of the Trained Flow

- For RL-agent to converge to better returns, the model should approximate the constrained space well.
  - The model should produce actions only in feasible region (**Accuracy**)
  - The model should be able to cover most of the feasible region (**Recall**)

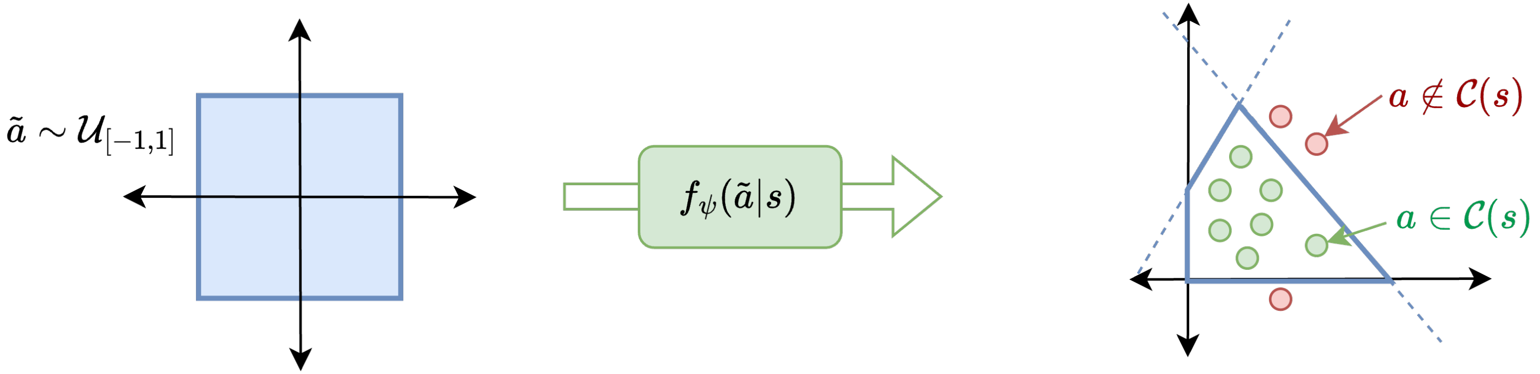


(a) WGAN (b) VAE (c) Normalizing Flow (Ours)

Figure 3: Mapping between a uniform distribution and action space of Reacher with constraint $a_1^2 + a_2^2 \leq 0.05$

Measuring Accuracy

- Generate sample from using the model and measure what percentage of them are placed in the feasible region.



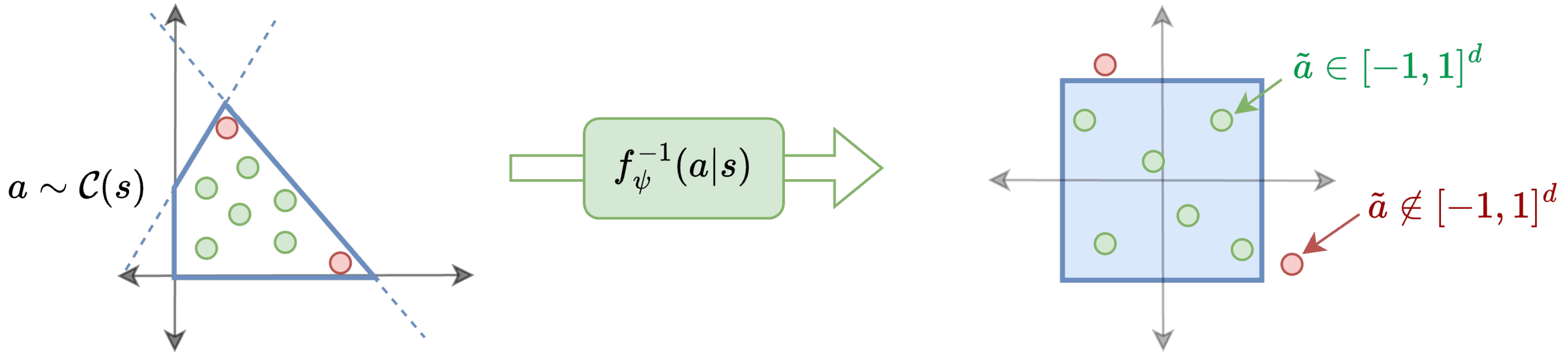
Uniform base distribution

Constrained space $\mathcal{C}(s)$

Measuring Recall (Coverage)

- Generate samples from the feasible region using a known technique and map them back to the latent space to measure recall.

$$recall(s) = \frac{\sum_{a \in \mathcal{C}(s)} \mathbb{I}_{dom_{f_\psi}} f_\psi^{-1}(a, s)}{|\mathcal{C}(s)|}$$

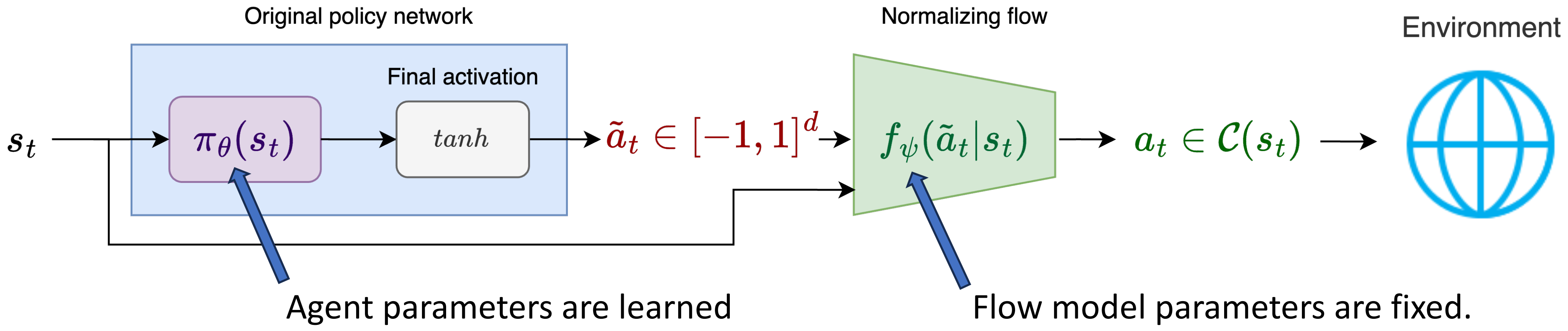


Constrained space $\mathcal{C}(s)$

Uniform base distribution

Integration with the RL agent

- Inverse of the trained model is incorporated with RL agent's policy network to output feasible actions as the final output

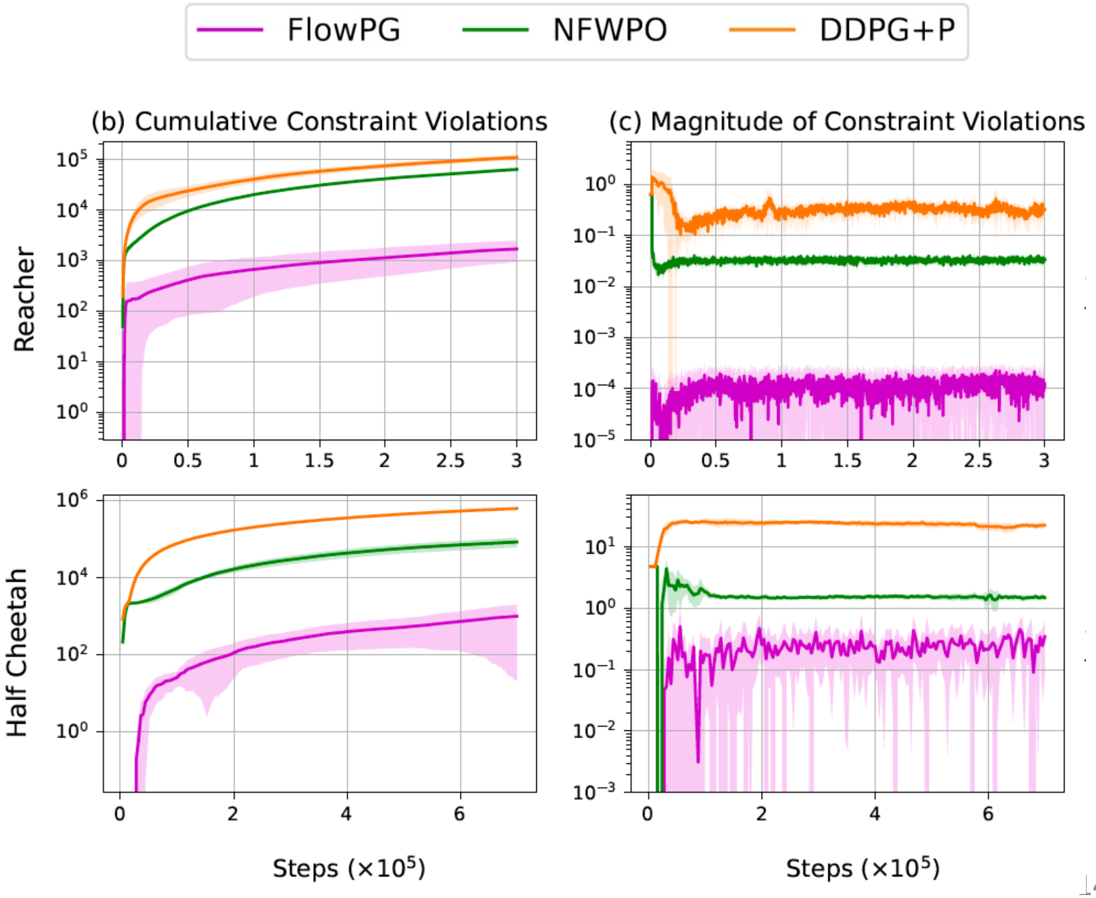


Evaluation

- Environments
  - 4 continues action environments
    - Reacher
    - Half Cheetah
    - Hopper
    - Walker2d
  - 1 combinatorial action environment
    - Bike Sharing System (BSS)
- Algorithms
  - NFWPO
  - DDPG+Projection
  - FlowPG (Ours)

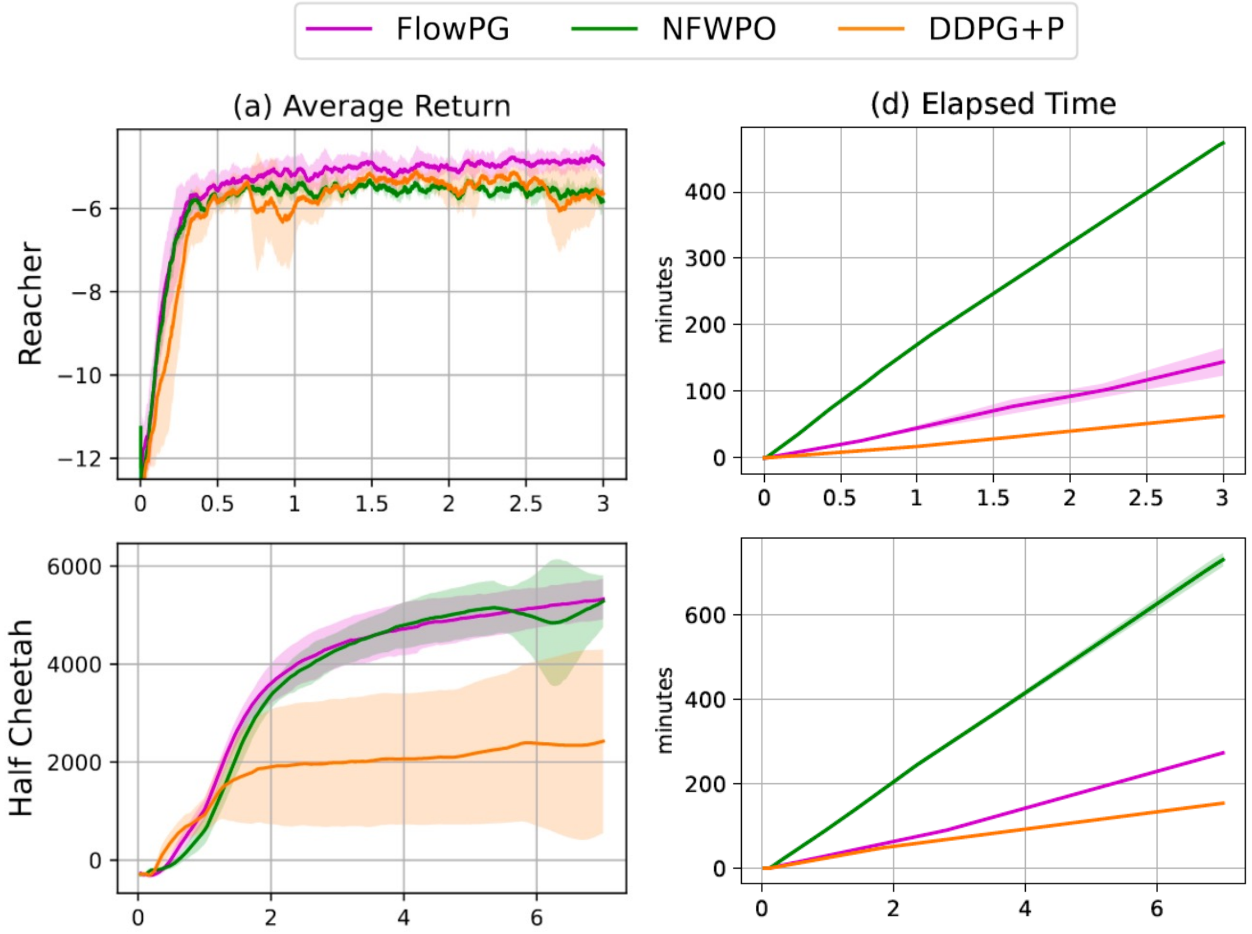
Fewer constraint violations

- Our approach produce fewer constrain violations and the magnitude of the constraint violations are low.



Better returns and faster run time

- In all cases our approach produce similar or better returns with a significantly faster run times than NFWPO.



SMU SINGAPORE MANAGEMENT UNIVERSITY | School of Computing and Information Systems

Thank you

FlowPG: Action-constrained Policy Gradient with Normalizing Flows

Janaka Chathuranga Brahmanage, Jiajing Ling, Akshat Kumar
School of Computing and Information Systems
Singapore Management University
{janakat.2022, jjling.2018}@phdcs.smu.edu.sg, akshatkumar@smu.edu.sg