

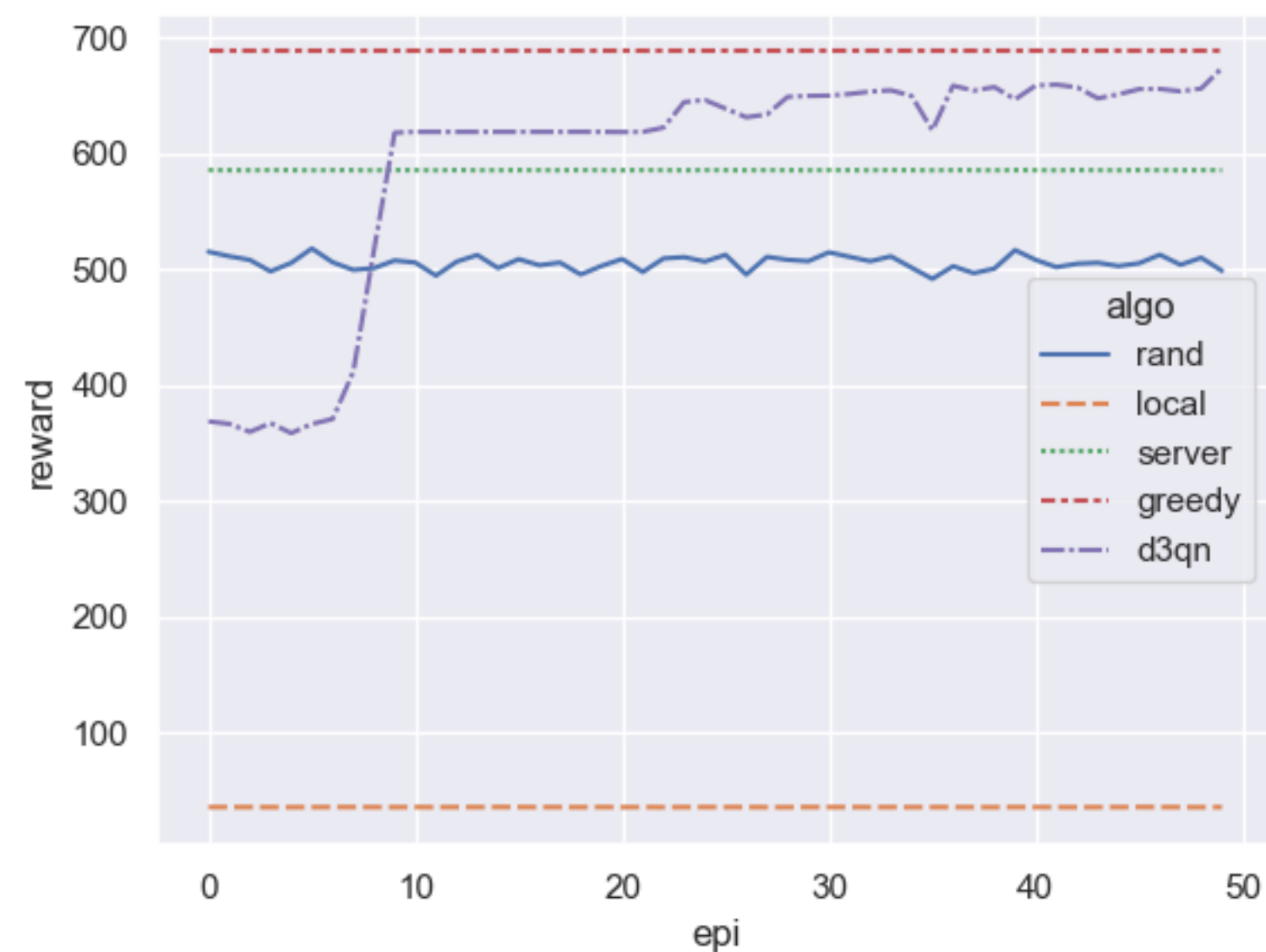
# 进度报告&论文分享

DRL与元学习、分布式相结合

范也 2021.10.27

# 研究进度

1. 使用ddpg，效果不好
2. 换成discrete的D3QN算法，勉强可以



# 后续

1. 加入网络传输延迟
2. 设计公式、数据
3. 尝试与分布式、元学习等结合

# Fast Adaptive Task Offloading in Edge Computing based on Meta Reinforcement Learning

期刊: TPDS 2021 Jan

作者: Jin Wang, Jia Hu, Geyong Min, Albert Y. Zomaya

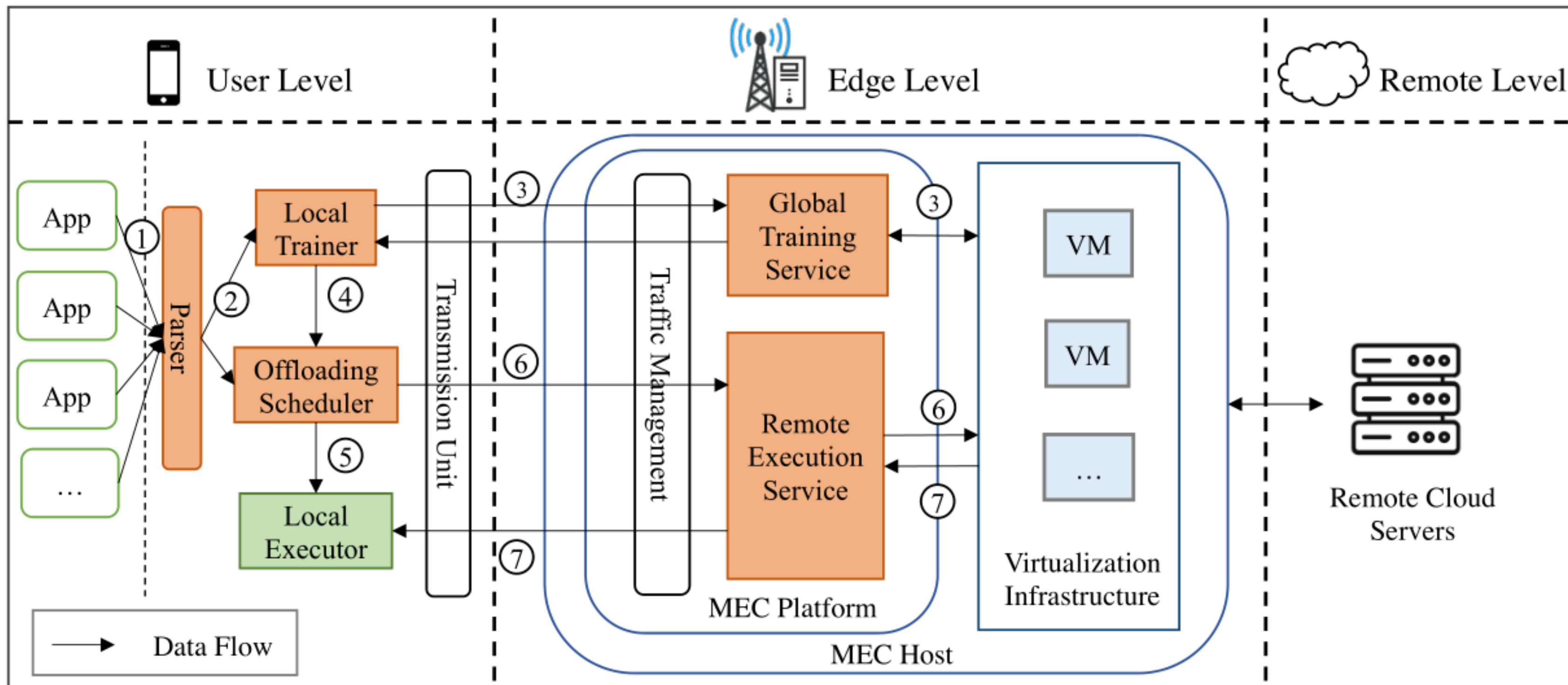
机构: University of Exeter, United Kingdom.

## 摘要:

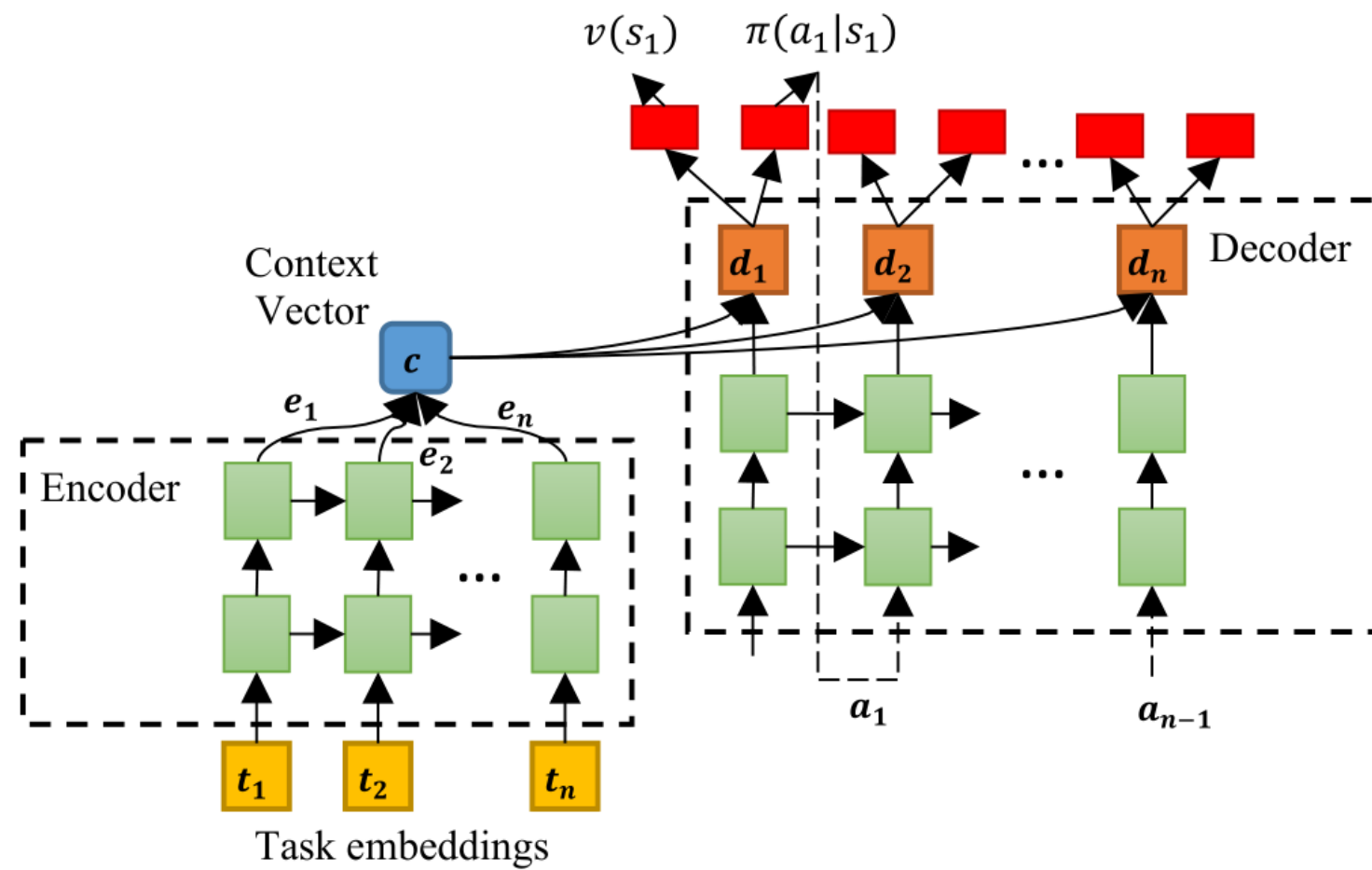
训练drl算法需要的时间比较长, 对于不同的环境还需要分别训练。

1. 使用meta-rl的方法, 仅需较少次数的更新, 便可以适应新的环境。
2. 使用seq2seq编码DAG, 最小化延迟, 快速适应新的环境。

# 架构



# 网络

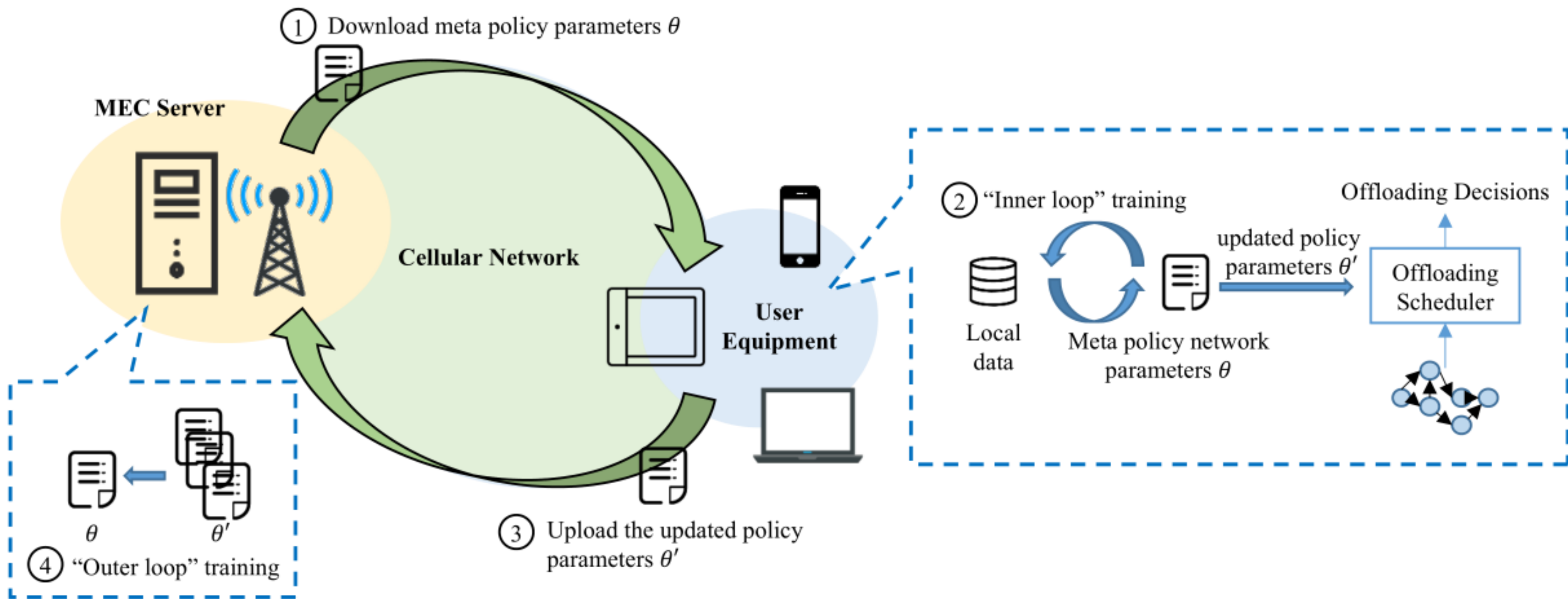


- encoder: task embeddings
- decoder: offloading decisions, value function

# 训练

1. inner loop: 使用meta-policy初始化本地网络, 在本地抽样训练
2. outer loop: 使用本地的参数更新meta-policy

# 元强化学习 (Meta Reinforcement Learning)



# 实验

Testing dataset	Heuristic Algorithms			Fine-tuning DRL		MRLCO	
	Optimal	HEFT-based	Greedy	update steps(20)	update steps(100)	update steps(20)	update steps(100)
Toploogy I	679.31	800.75	847.73	812.32	789.92	791.03	722.63
Toploogy II	555.46	802.46	848.43	688.05	636.49	651.42	601.93
Toploogy III	605.05	814.39	859.03	778.52	712.79	729.63	641.92
$n = 20$	689.21	838.31	893.62	818.14	802.50	802.41	743.42
$n = 30$	N/A	1222.93	1276.70	1185.47	1152.07	1174.55	1098.43
$n = 40$	N/A	1527.47	1589.66	1493.11	1432.41	1472.53	1397.63
$R_{ul} = R_{dl} = 5.5$ Mbps	770.10	929.79	990.58	945.82	901.36	897.73	831.58
$R_{ul} = R_{dl} = 8.5$ Mbps	628.21	757.99	763.49	736.81	701.75	703.38	674.93
$R_{ul} = R_{dl} = 11.5$ Mbps	524.14	649.15	684.97	589.33	570.19	567.88	548.26

# Distributed and Collective Deep Reinforcement Learning for Computation Offloading: A Practical Perspective

期刊: TPDS 2021 May

作者: Xiaoyu Qiu, Weikun Zhang, Wuhui Chen, Zibin Zheng

机构: 中山大学

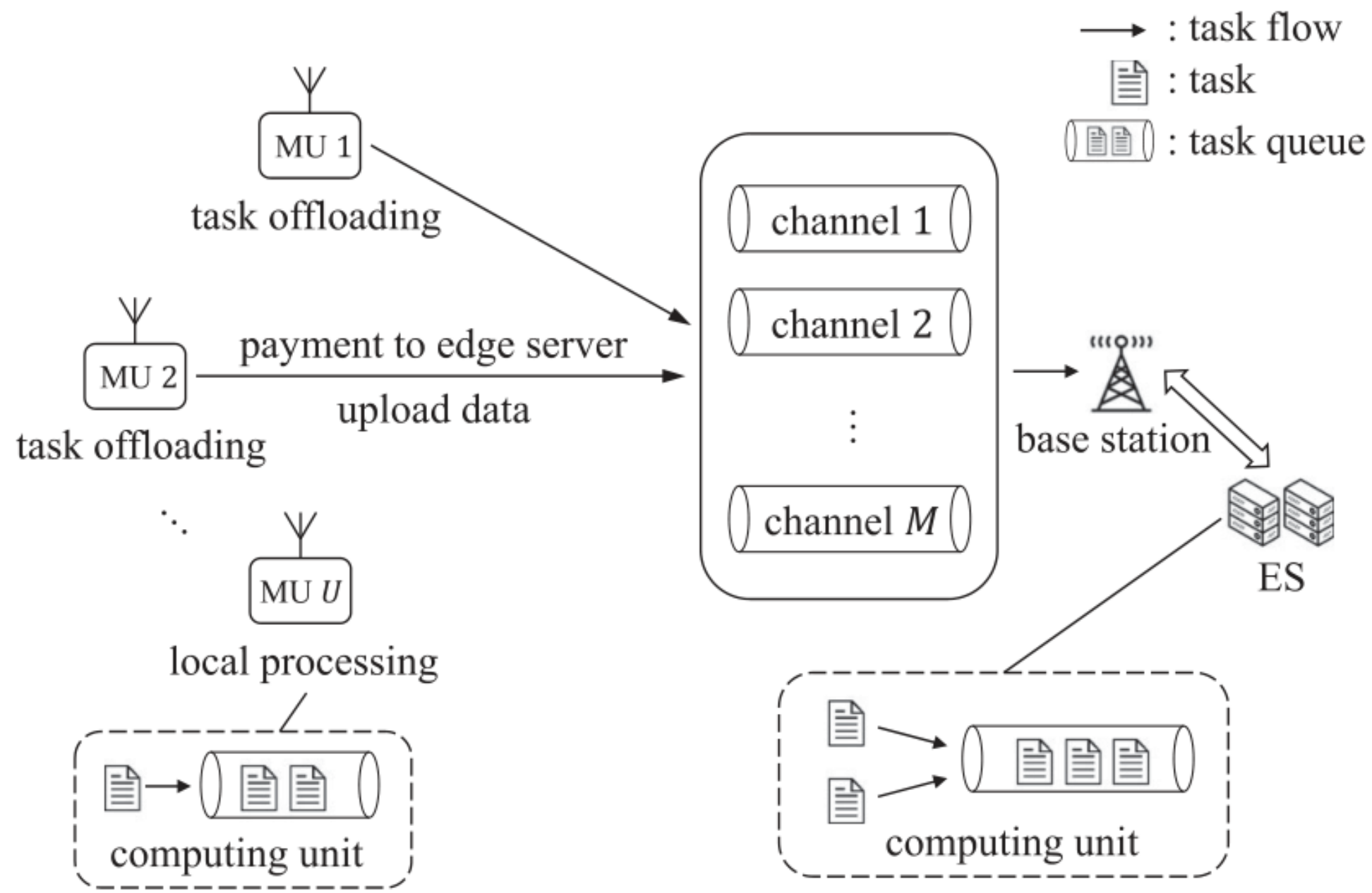
## 摘要

1. 使用分布式、协同方法训练，增多数据量，提升泛化性
2. adaptive n-step learning提升了训练的效率。
3. 结合deep neuroevolution和policy gradient



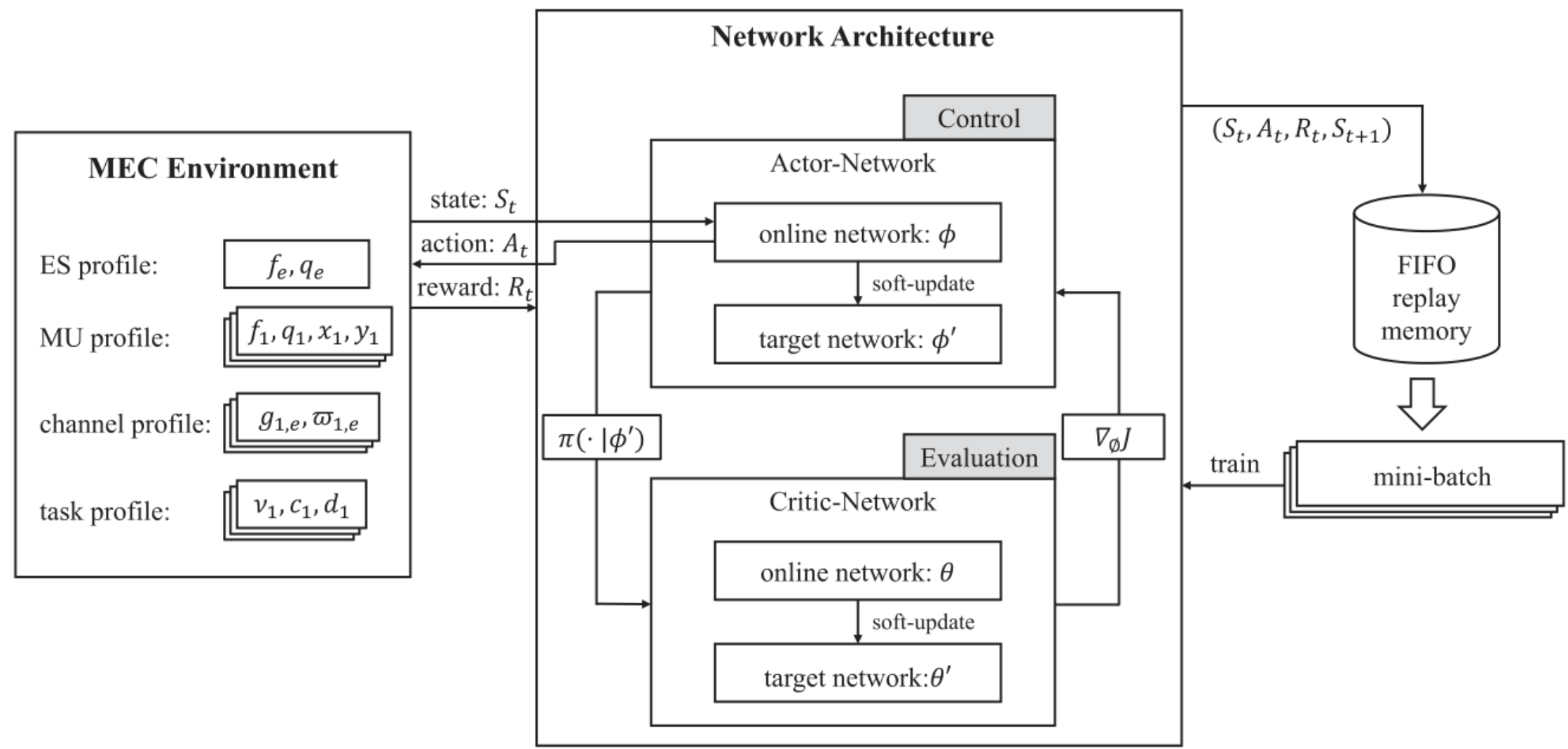
# 系统建模

1. 多UE(用户设备), 单个ES(边缘服务器)
2. 目标: 最小化  $latency + energy$



# 算法

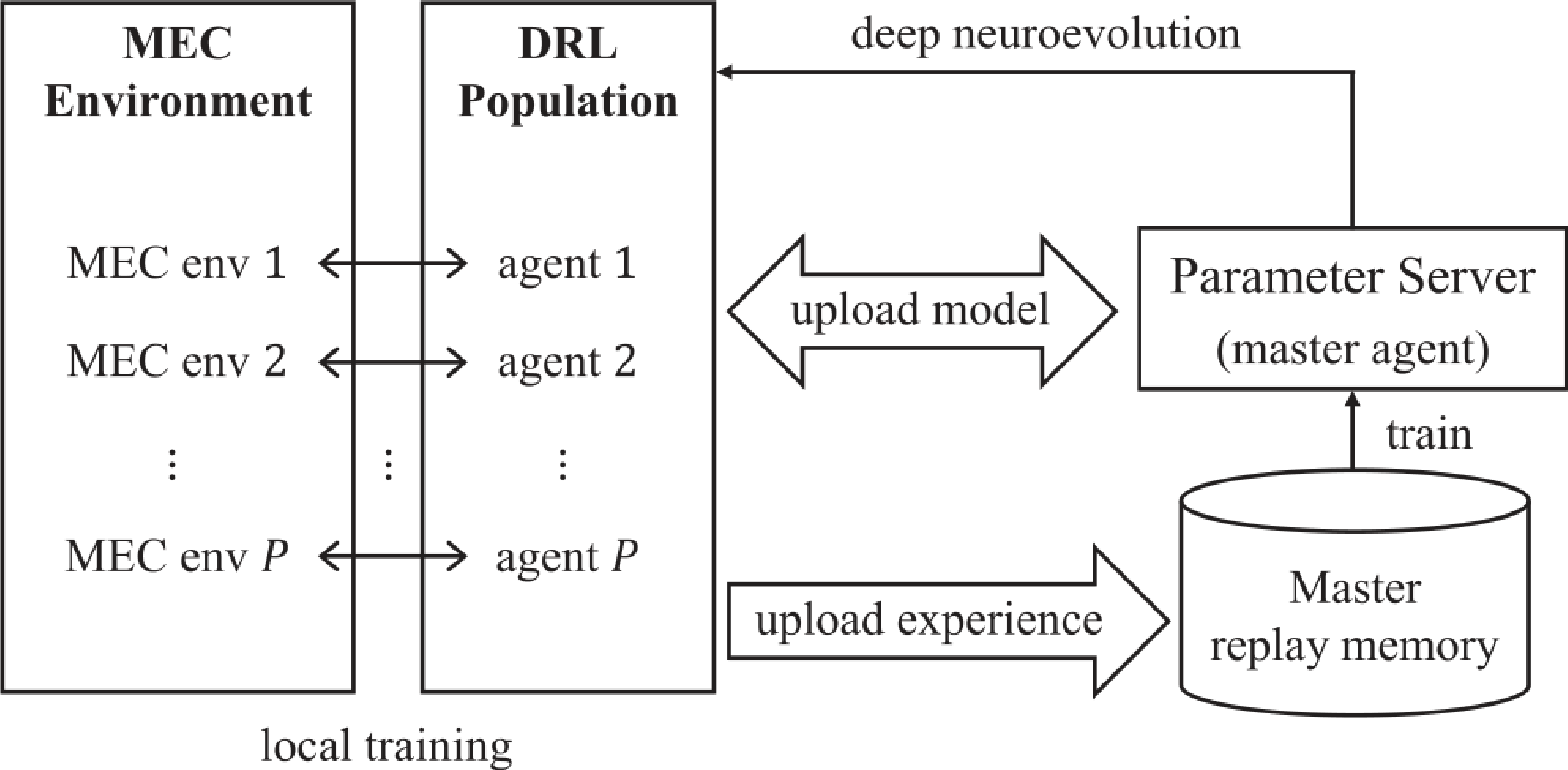
## Actor Critic framework



# 特点

- 1. 长期来看，效果较直接学习q值更好
- 2. 训练时间较长

# 分布式训练



# Adaptive N-Step Learning

G. Barth-Maron et al., “Distributed distributional deterministic policy gradients,” 2018

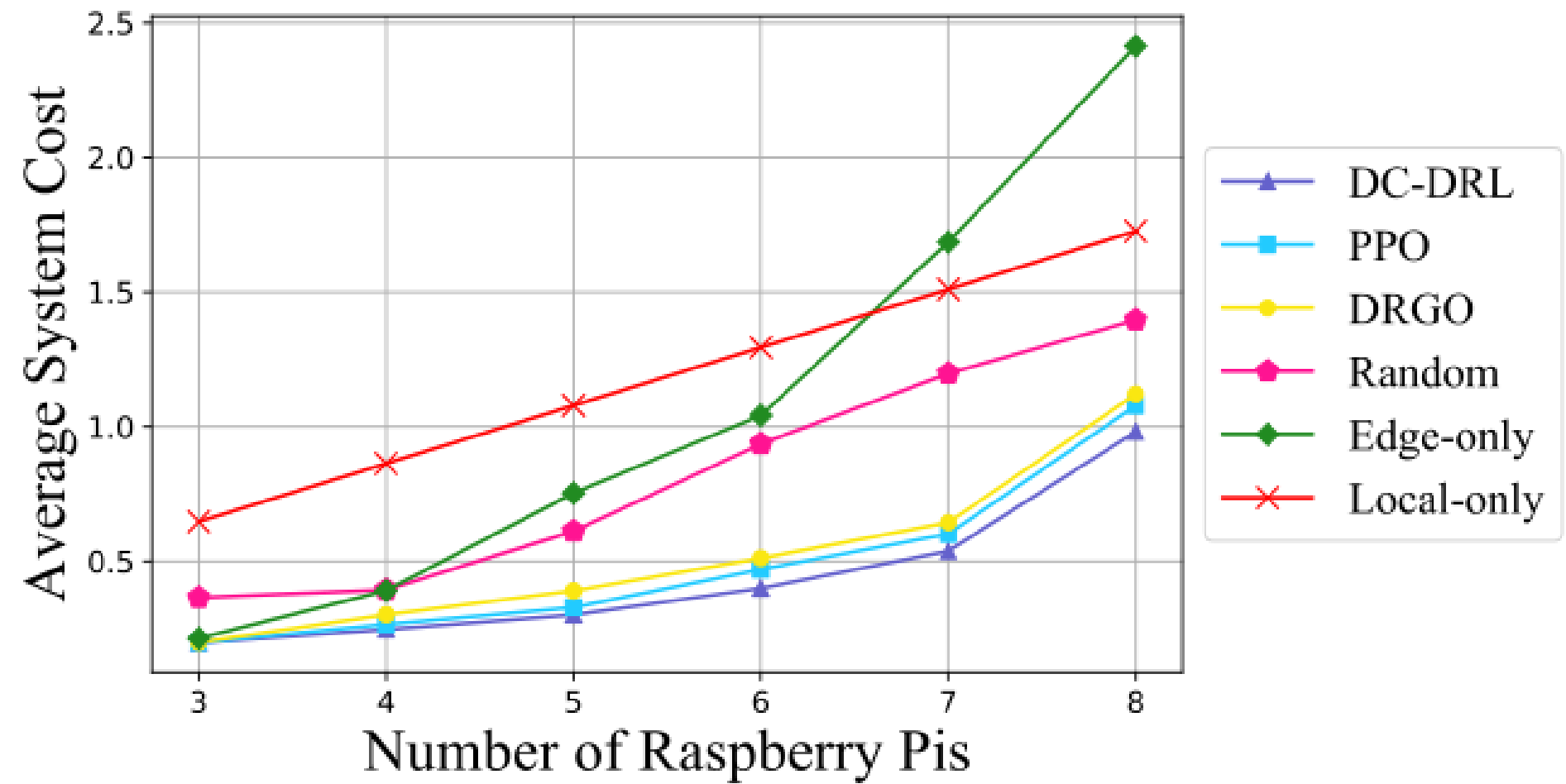
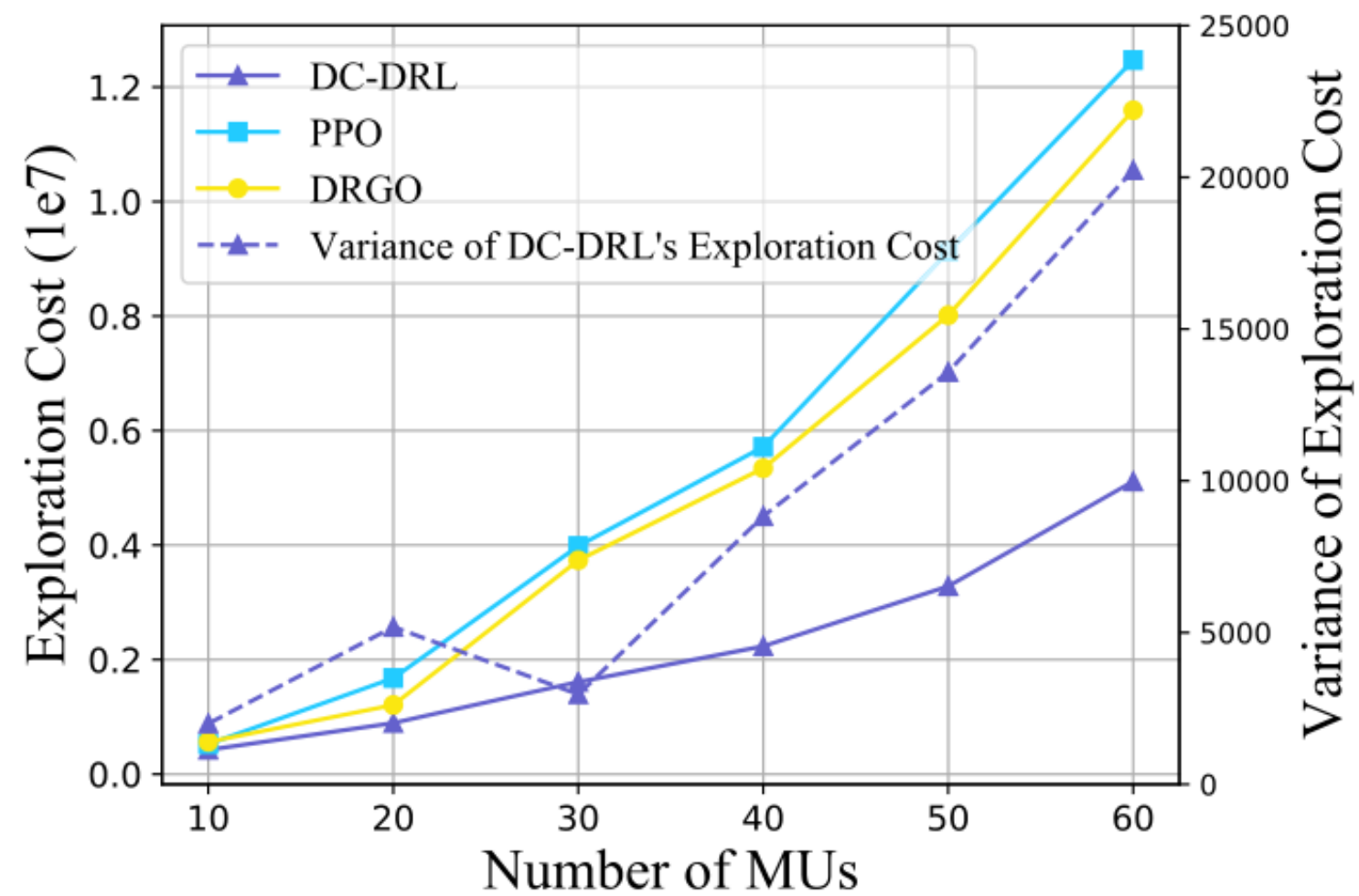
$$\mathcal{T}Q(S_t, A_t \mid \theta) = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \dots + \gamma^{n-1} R_{t+n-1} \\ + \gamma^n \cdot \mathbb{E}[Q(S_{t+n}, \pi(S_{t+n} \mid \phi') \mid \theta')]$$

# Deep Neuroevolution and Policy Gradient

H. Beyer, “Evolution strategies,” Scholarpedia, vol. 2, no. 8, 2007, Art. no. 1965.

1. 与遗传算法相结合，保证收敛。
2. 向输出中添加噪声。
3. 选择其中表现较好的模型。

# 实验



# 总结

1. 深度强化学习训练成本较高
2. 不同环境共同训练一个模型，使用这个模型进行初始化
3. 在训练中使用一些trick（剪裁梯度、启发式等）提高训练效率和泛化能力

# 计划

1. 实现网络层传输
2. 考虑使用分布式或者迭代的方法优化



