Learning Dexterous In-Hand Manipulation

- Research Background 机器人控制的现状
- Previous Research 难以迁移、行为学习受限
- Author's Contribution 可迁移的控制策略
- Learning Control Policies From State 控制策略的学习
- Distributed Implementation of PPO 大规模分布式训练
- State Estimation from Vision 基于视觉感知的状态估计
- Qualitative Results 自发的操控行为
- Quantitative Results 可迁移性、视觉感知可行性
- Ablation of Randomizations 消融实验
- Effect of Memory in Policies 记忆增强的迁移策略
- Distributed Training 最佳的训练配置
- Vision Performance 基于视觉的状态估计器

学习灵活的手部操作——Robotics

汇报人：李志豪

时间：2023.11.16

难以灵活操控物体

- 现在的机器人专门面向特定任务而设计，工作在受约束的环境中
- 现在的机器人难以利用复杂的末端执行器，以实现抓取、拾取或操纵其他物体

Inspiration Source

- 人类能够在复杂多样的环境中，执行一系列灵活的操控任务，因此模拟人类设计机器人手爪能够增强机器人控制的能力

研究的问题

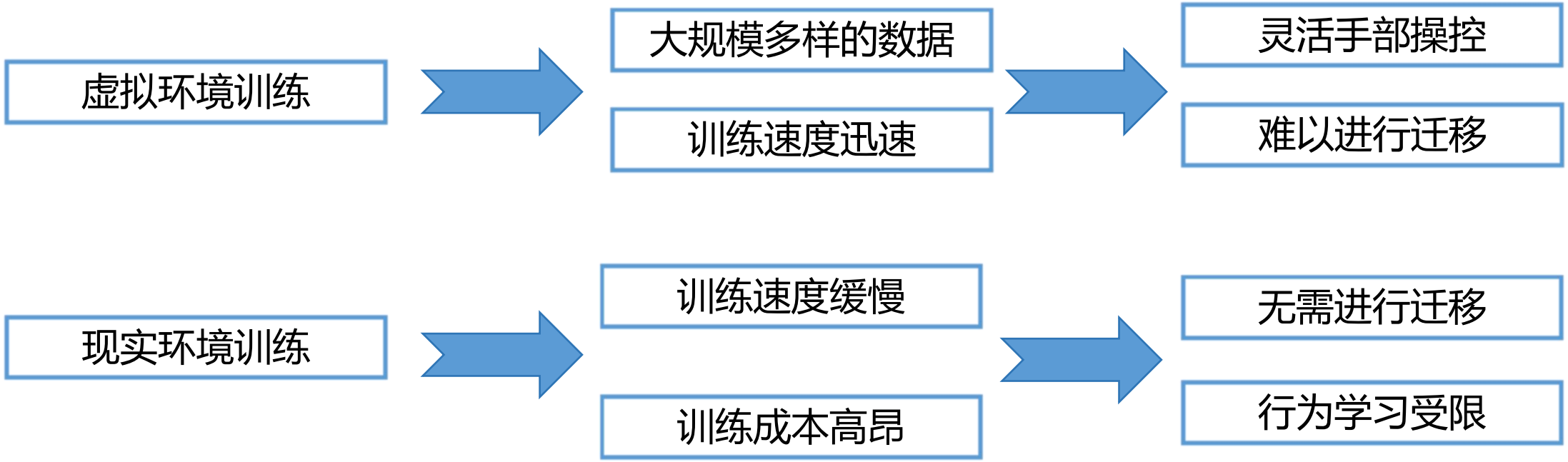
- 机器人手掌中物体的重定向[1]

研究的目标

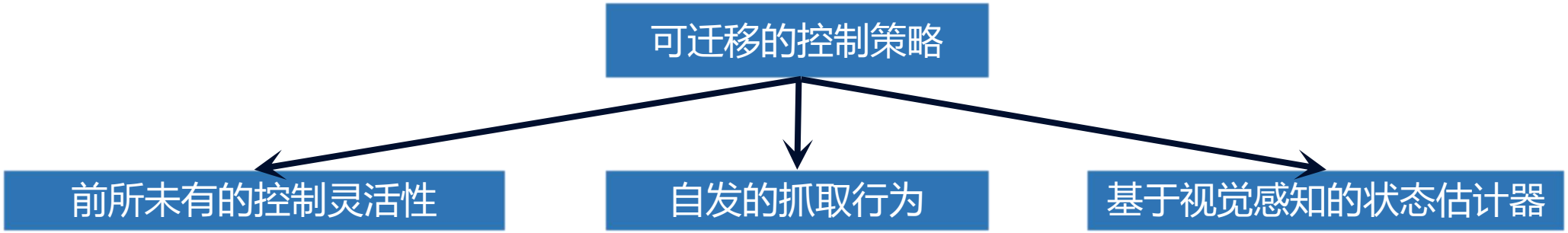
- 能够使得机器人将物体重定向为指定的目标状态

[1]物体重定向，指在机器人或人类手中改变物体的方向或位置的过程

难以迁移、行为学习受限



之前的研究工作特点分析



- 在模拟环境中的各影响因素进行广泛的随机化，并且附加额外的效果
  - Physical parameters
  - Observation noise
  - PhaseSpace tracking errors
  - Action noise and delay
  - Timing randomization
  - Backlash model
  - Random forces on the object
  - Randomized vision appearance
- 采用内存增强的控制策略使得学习适应性行为和系统环境识别成为可能
  - LSTM Policy And LSTM Value Function
- 利用大规模分布式强化学习训练，可能增加模型学习的效果和速度
  - Distributed Training with Rapid，distributed implementation of PPO

Learning Control Policies From State

控制策略的学习

西安电子科技大学 XIDIAN UNIVERSITY

Policy Architecture

Actions And Rewards

Proximal Policy Optimiztion (PPO)

- Continuous Action Spaces
- Discrete Action Spaces ✓

Policy Network

$Map\ \pi:\ S \rightarrow A$

LSTM+RNN

Value Network

$$V_\pi(s) = \sum_{a \in A} \pi(a|s)(R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a V_\pi(s'))$$

Actor-Critic Method
With Asymmetric Inputs

$$\nabla_\theta J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) A(s_t, a_t)$$

$$= \frac{1}{m} \sum_{i=1}^{m} \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) (Q(s_t, a_t) - V_\phi(s_t))$$

Actor Critic

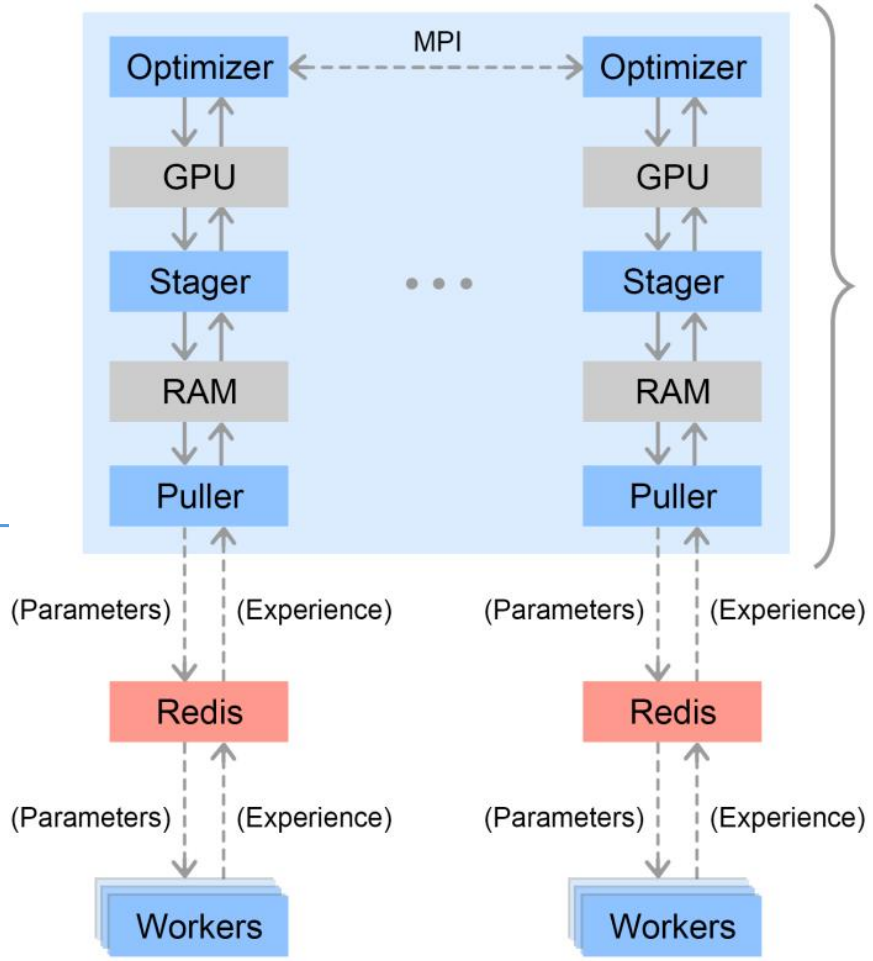
$$r_t = d_t - d_{t+1}$$

Achieve A Goal: +5

Drop Object: -20

Distributed Implementation of PPO

大规模分布式训练



Proximal Policy Optimiztion (PPO)

$$L_{\text{PPO}} = \mathbb{E}\min\left(\frac{\pi(a_t|s_t)}{\pi_{\text{old}}(a_t|s_t)}\hat{A}_t^{\text{GAE}},\ \text{clip}\left(\frac{\pi(a_t|s_t)}{\pi_{\text{old}}(a_t|s_t)},\ 1-\epsilon,\ 1+\epsilon\right)\hat{A}_t^{\text{GAE}}\right)$$

Generalized Advantage Estimator (GAE)

$$\hat{V}_t^{(k)} = \sum_{i=t}^{t+k-1}\gamma^{i-t}r_i + \gamma^k V(s_{t+k}) \approx V^{\pi}(s_t, a_t) \qquad \hat{V}_t^{\text{GAE}} = (1-\lambda)\sum_{k>0}\lambda^{k-1}\hat{V}_t^{(k)} \approx V^{\pi}(s_t, a_t)$$

$$\hat{A}_t^{\text{GAE}} = \hat{V}_t^{\text{GAE}} - V(s_t) \approx A^{\pi}(s_t, a_t)$$

WorkFlows

- 在每一轮开始，Worker Machines 从优化器获取最新的策略参数，启动训练阶段
- 在分布的随机化环境中，将执行当前策略所得的经验传回优化器
- 优化器与 Workers 之间通过内存数据区中的 Redis 进行的，通过设置多个 Redis 来分担 Worker 的训练任务
- 优化器从 Redis 获取生成的策略经验，然后将其加载到对应的GPU内存中进行处理；在计算完参数的梯度后，平均分配到各线程中用于更新模型参数即控制策略

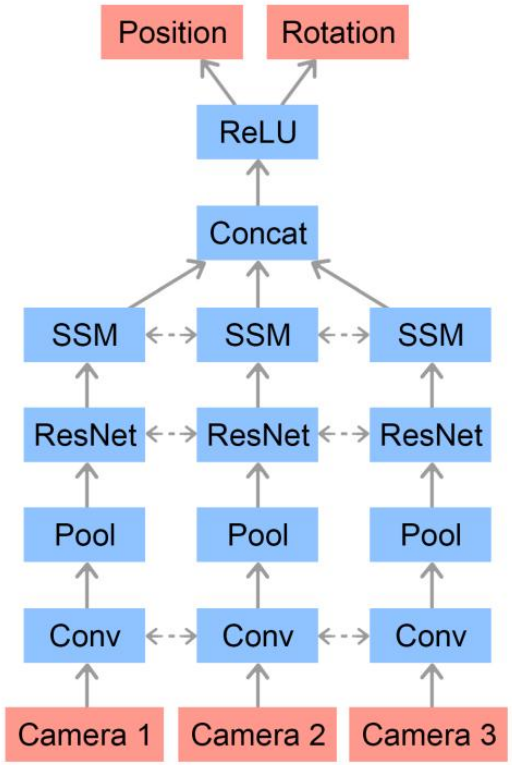


Table 12: Hyperparameters for the vision model architecture.

Layer	Details
Input RGB Image	$200 \times 200 \times 3$
Conv2D	32 filters, 5×5, stride 1, no padding
Conv2D	32 filters, 3×3, stride 1, no padding
Max Pooling	3×3, stride 3
ResNet	1 block, 16 filters, 3×3, stride 3
ResNet	2 blocks, 32 filters, 3×3, stride 3
ResNet	2 blocks, 64 filters, 3×3, stride 3
ResNet	2 blocks, 64 filters, 3×3, stride 3
Spatial Softmax	
Flatten	
Concatenate	all 3 image towers combined
Fully Connected	128 units
Fully Connected	output dimension (3 position + 4 rotation)

图像维度

Input RGB Image	$200 \times 200 \times 3$	ResNet 1	$23 \times 23 \times 16$	Spatial Softmax	$3 \times 3 \times 64$
Conv2D	$196 \times 196 \times 32$	ResNet 2	$9 \times 9 \times 32$	Flatten	1×576
Conv2D	$194 \times 194 \times 32$	ResNet 3	$5 \times 5 \times 64$	Concatenate	1×1728
Max Pooling	$64 \times 64 \times 32$	ResNet 4	$3 \times 3 \times 64$		

Qualitative Results 自发的操控行为

无直接激励

无人工演示：作者没有使用人类提供的任何演示或示例来训练如何操作方块

不对先验知识进行编码：作者也没有将任何先验知识或先前存在的信息编码到学习过程中使用的激励函数中

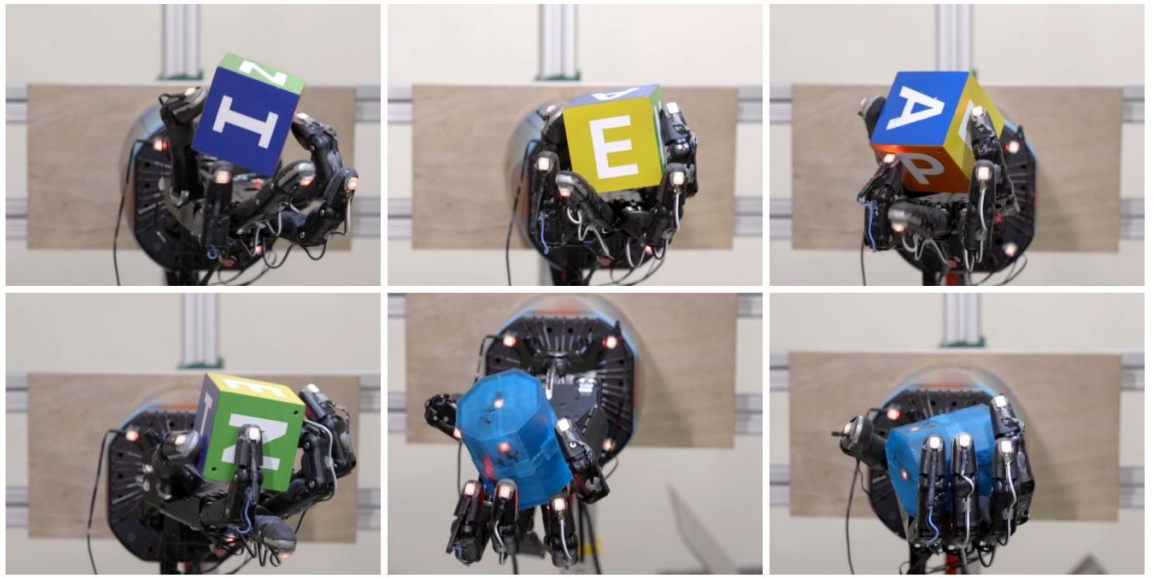


Figure 7: Different grasp types learned by our policy. From top left to bottom right: Tip Pinch grasp, Palmar Pinch grasp, Tripod grasp, Quadpod grasp, 5-Finger Precision grasp, and a Power grasp. Classified according to [18].

Adaptive Learning

- **自适应迁移**：倾向于使用更加灵活的小拇指而非食指和中指操控物体
- **成熟的策略**：能够利用手指的远端关节进行物体的旋转

Quantitative Results 可迁移性、视觉感知可行性

Simulated task	Mean	Median	Individual trials (sorted)
Block (state)	43.4 ± 13.8	50	-
Block (state, locked wrist)	44.2 ± 13.4	50	-
Block (vision)	30.0 ± 10.3	33	-
Octagonal prism (state)	29.0 ± 19.7	30	-
Physical task			
Block (state)	18.8 ± 17.1	13	50, 41, 29, 27, 14, 12, 6, 4, 4, 1
Block (state, locked wrist)	26.4 ± 13.4	28.5	50, 43, 32, 29, 29, 28, 19, 13, 12, 9
Block (vision)	15.2 ± 14.3	11.5	46, 28, 26, 15, 13, 10, 8, 3, 2, 1
Octagonal prism (state)	7.8 ± 7.8	5	27, 15, 8, 8, 5, 5, 4, 3, 2, 1

定量指标：连续成功旋转达成目标次数

中断标志

- **物体掉落：** Reward = -20
- **时间限制：** 80s内达不成目标
- **次数限制：** 50次旋转仍达不成目标

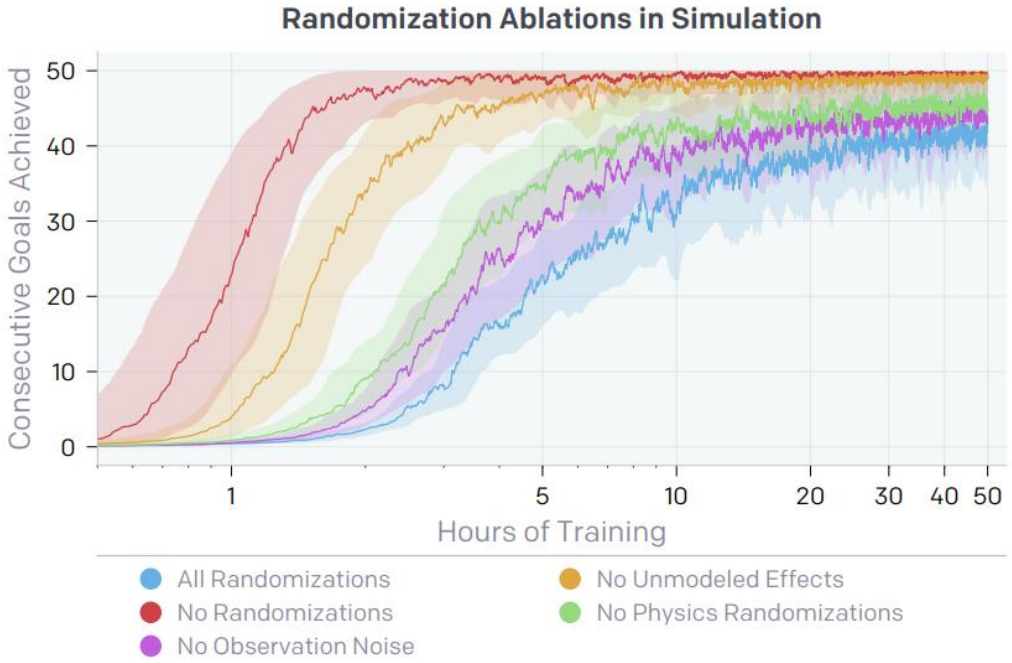
实验配置

- 训练环境：加入各种随机化因素并进行校准
- 试验次数：虚拟环境中试验 100 次/组，物理环境中试验 10 次/组
- state：利用训练的PhaseSpace Markers进行目标姿势估计
- vision：利用训练的Vision Network进行目标姿势估计

实验结论

- 迁移差距缩小：虚拟环境迁移到物理环境中，仍然存在一定的差距
- Vision Model：利用Vision Model略微有些精度损失，但是已经足够实现较好的迁移性能，同时能够有望摆脱实验室环境进行应用
- Object Generation：利用方块的控制策略进行八棱柱试验，虽然能够学习到迁移的控制策略，但性能上有所差距，模型的泛化能力需要进一步调整

Ablation of Randomizations 消融实验



指数移动平均EMA

$$EMA_t = \alpha \cdot X_t + (1-\alpha) \cdot EMA_{t-1}$$

其中 α 表示平滑因子，采用 EMA 可以分析和预测每组实验结果的趋势

90% 置信区间

$$Lower\ Bound = \overline{X} - Z \cdot SE, Upper\ Bound = \overline{X} + Z \cdot SE$$

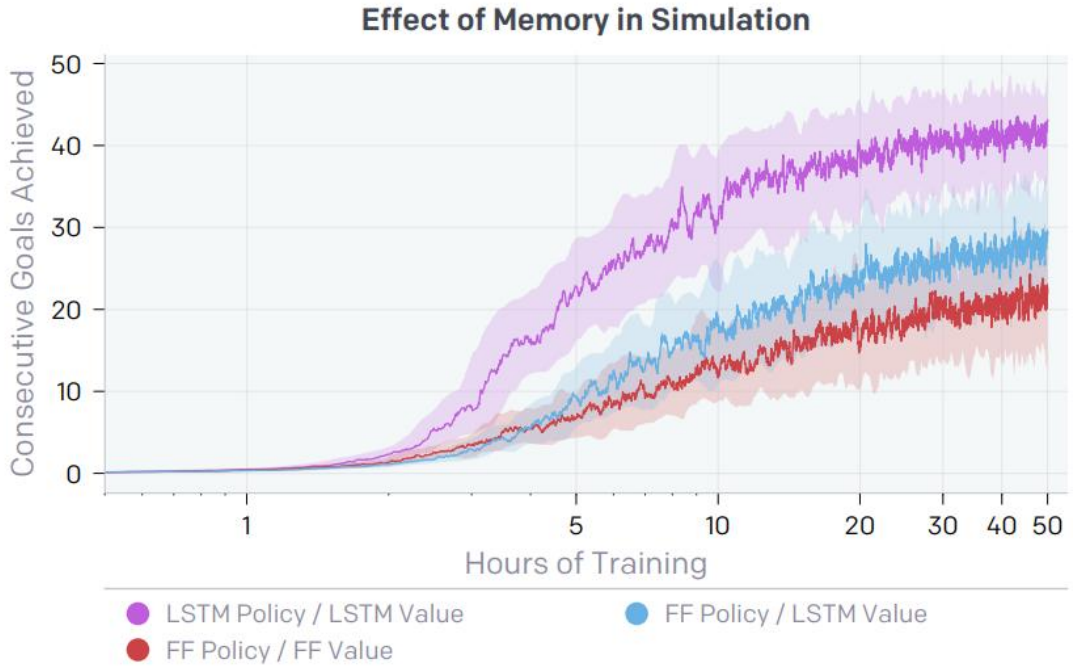
其中 $\overline{X}$ 表示序列的均值，SE 表示序列的方差，Z 取1.645.

实验结论

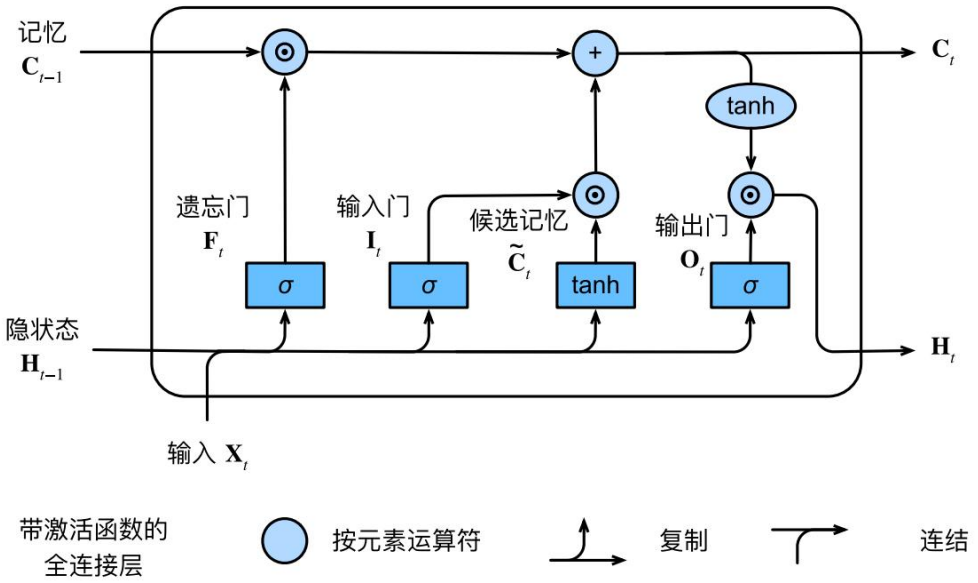
- All Randomizations：
  收敛速度较慢，需要获取更多的计算和仿真经验但是迁移效果更好
- No Observation Noise：动作捕捉系统存在一定噪声，加入噪声训练使得效果改善较为明显

Training environment	Mean	Median	Individual trials (sorted)
All randomizations (state)	18.8 ± 17.1	13	50, 41, 29, 27, 14, 12, 6, 4, 4, 1
No randomizations (state)	1.1 ± 1.9	0	6, 2, 2, 1, 0, 0, 0, 0, 0, 0
No observation noise (state)	15.1 ± 14.5	8.5	45, 35, 23, 11, 9, 8, 7, 6, 6, 1
No physics randomizations (state)	3.5 ± 2.5	2	7, 7, 7, 3, 2, 2, 2, 2, 2, 1
No unmodeled effects (state)	3.5 ± 4.8	2	16, 7, 3, 3, 2, 2, 1, 1, 0, 0
All randomizations (vision)	15.2 ± 14.3	11.5	46, 28, 26, 15, 13, 10, 8, 3, 2, 1
No observation noise (vision)	5.9 ± 6.6	3.5	20, 12, 11, 6, 5, 2, 2, 1, 0, 0

Effect of Memory in Policies 记忆增强的迁移策略



LSTM Policy And Value Function

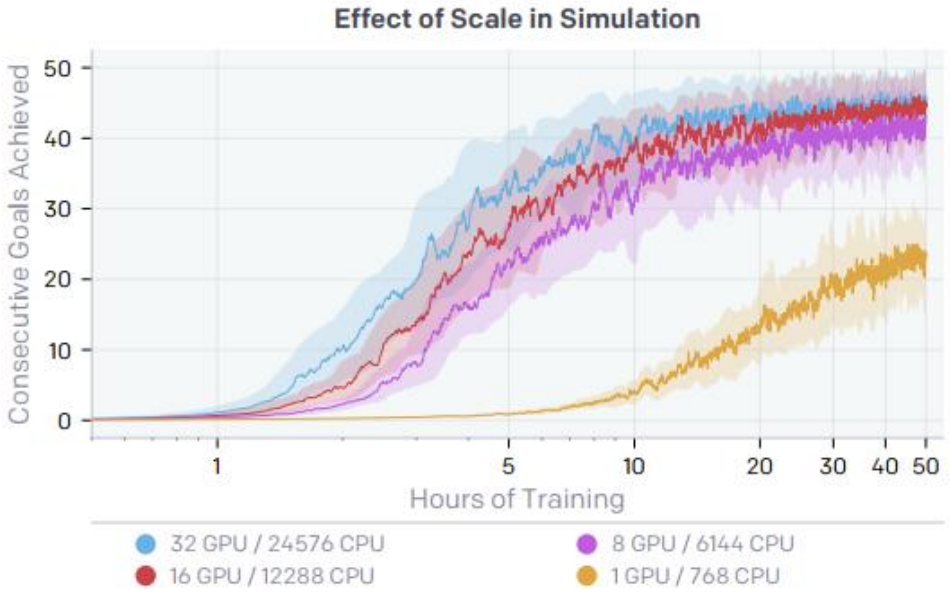


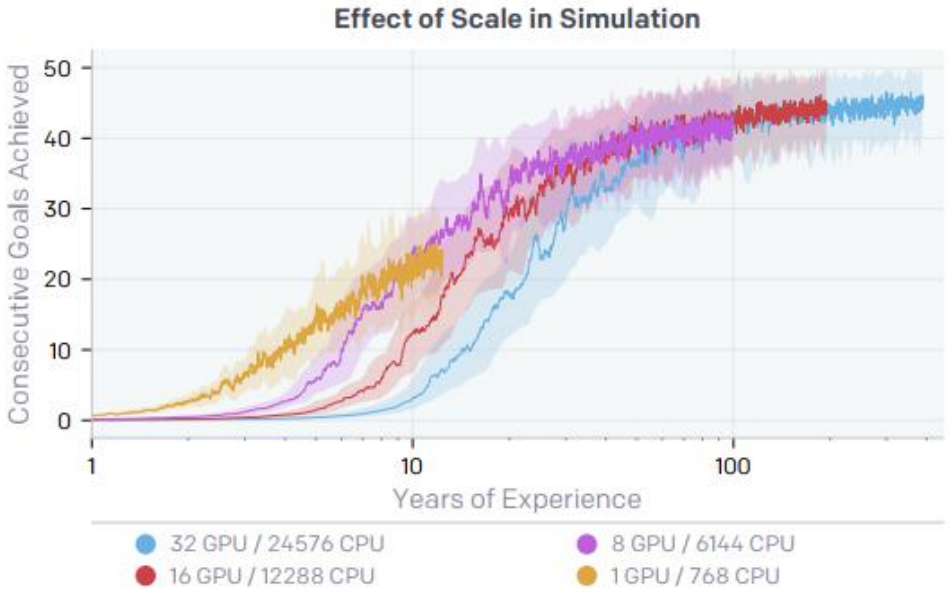
实验结论

- 增强性能：采用具有记忆的策略和值函数使得模型能够适应地迁移，具有更好的性能表现
- 自适应学习：LSTM网络的内部存储器，可用于预测环境随机化的影响，通过与物体 5s 的模拟交互后，在 80% 情况中，LSTM隐藏状态可以预测物体是大于还是小于平均水平

Network architecture	Mean	Median	Individual trials (sorted)
LSTM policy / LSTM value (state)	18.8 ± 17.1	13	50, 41, 29, 27, 14, 12, 6, 4, 4, 1
FF policy / LSTM value (state)	4.7 ± 4.1	3.5	15, 7, 6, 5, 4, 3, 3, 2, 2, 0
FF policy / FF value (state)	4.6 ± 4.3	3	15, 8, 6, 5, 3, 3, 2, 2, 2, 0

最佳的训练配置





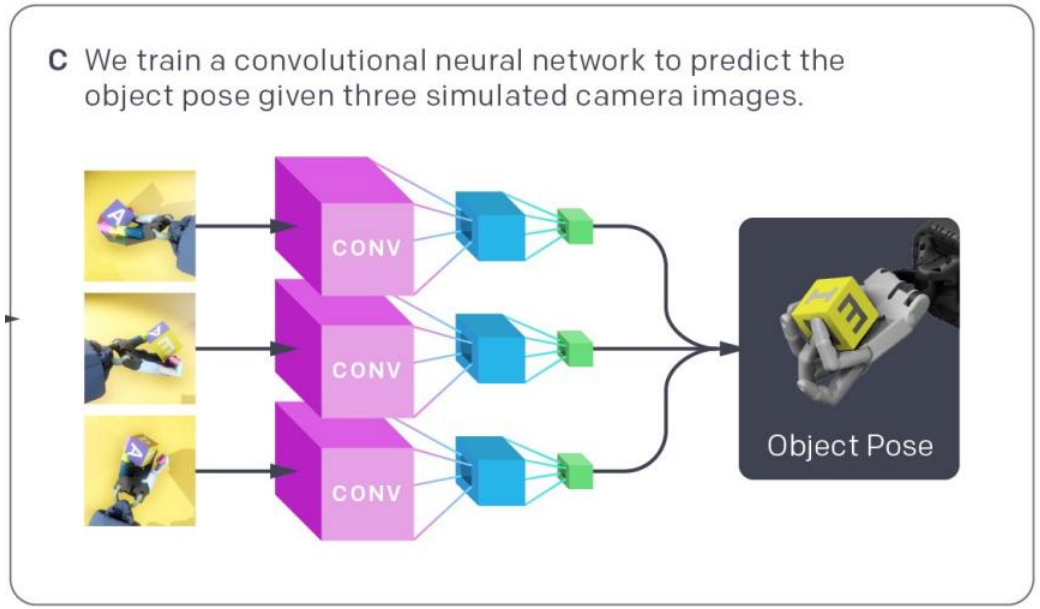
实验配置

- 单个GPU具有固定的批大小，整体的批大小于GPU数量成正比；
- 调整批次大小：改变批量大小意味着更改每次迭代期间一起处理的数据样本的数量，可以控制每个训练步骤中使用的数据量，这可能会影响学习过程的速度和效率，使训练过程控制更加灵活

实验结论

- 收益回减：在探索到最佳训练配置的情况后，继续增加训练配置规模，反而使得训练速度减缓，学习增益减少
- 最佳训练配置：16 GPUs and 12288 CPU cores 使得训练速度接近线性

Vision Performance 基于视觉的状态估计器



实验配置

- Test Set: 在收集得到的测试集上进行实验
- Unity: 用于模拟环境合成数据集，测试模拟环境中基于视觉的状态估计器的估计误差
- MuJoCo: 用于实际环境的真实数据集，测试真实环境中基于视觉的状态估计器的估计误差

实验结论

- 较低的数据误差：在模拟环境中合成的数据上能够实现较低的旋转误差和位置误差
- 成功迁移：虽然利用MuJoCo渲染的图像会增加些许误差，但是能够实现模型成功迁移

Table 6: Performance of a vision based pose estimator on synthetic and real data.

Test set	Rotation error	Position error
Rendered images (Unity)	$2.71° \pm 1.62$	$3.12\text{mm} \pm 1.52$
Rendered images (MuJoCo)	$3.23° \pm 2.91$	$3.71\text{mm} \pm 4.07$
Real images	$5.01° \pm 2.47$	$9.27\text{mm} \pm 4.02$

THANKS!

恳请批评指正

学习灵活的手部操作——Robotics

汇报人：李志豪

时间：2023.11.16