PPO×Family

第三讲：表征多模态观察空间

主办 中国计算机学会 CCF

承办 上海人工智能实验室 Shanghai Artificial Intelligence Laboratory

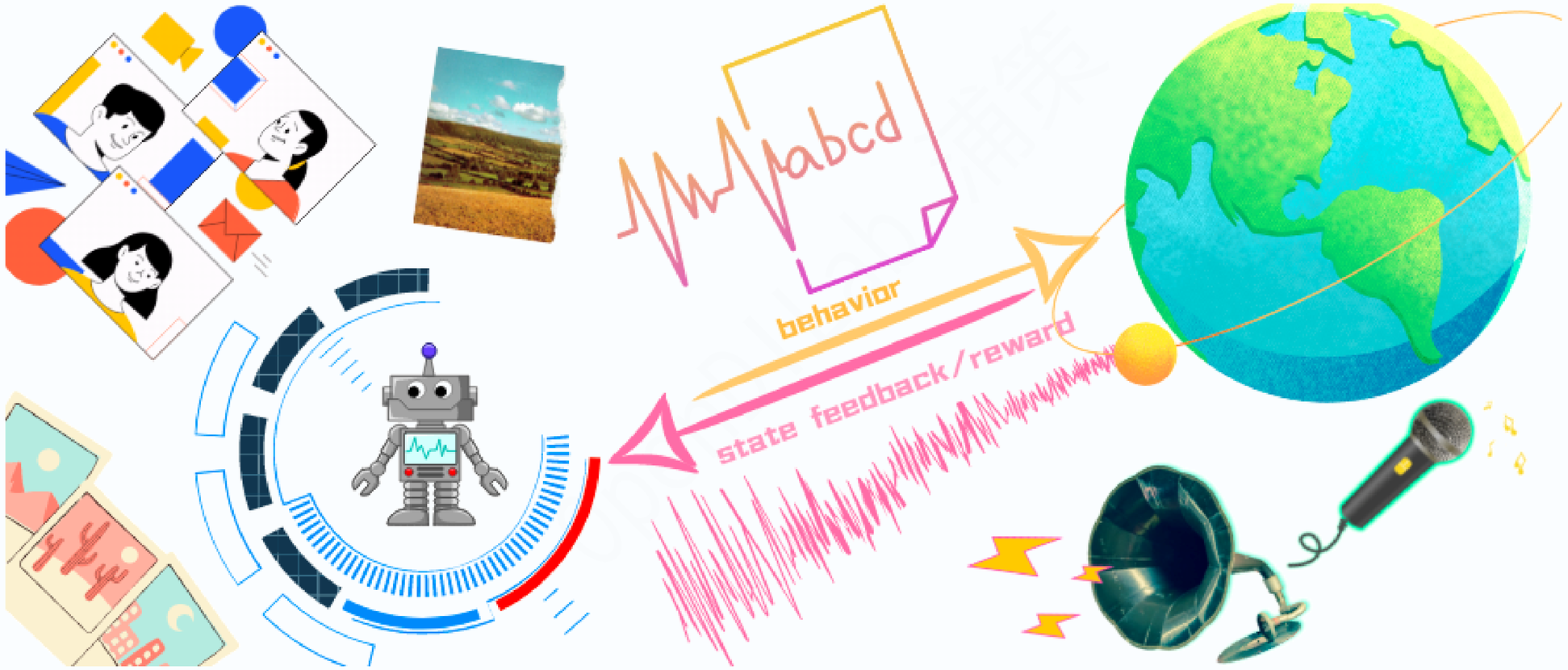
协办 GAIAA 北京大学人工智能研究院 INSTITUTE FOR ARTIFICIAL INTELLIGENCE, PEKING UNIVERSITY 浙江大学上海高等研究院 SHANGHAI INSTITUTE FOR ADVANCED STUDY ZHEJIANG UNIVERSITY 商汤 sensetime

支持 知乎 机器之心 Synced 智海 新一代人工智能科教平台 paperweekly

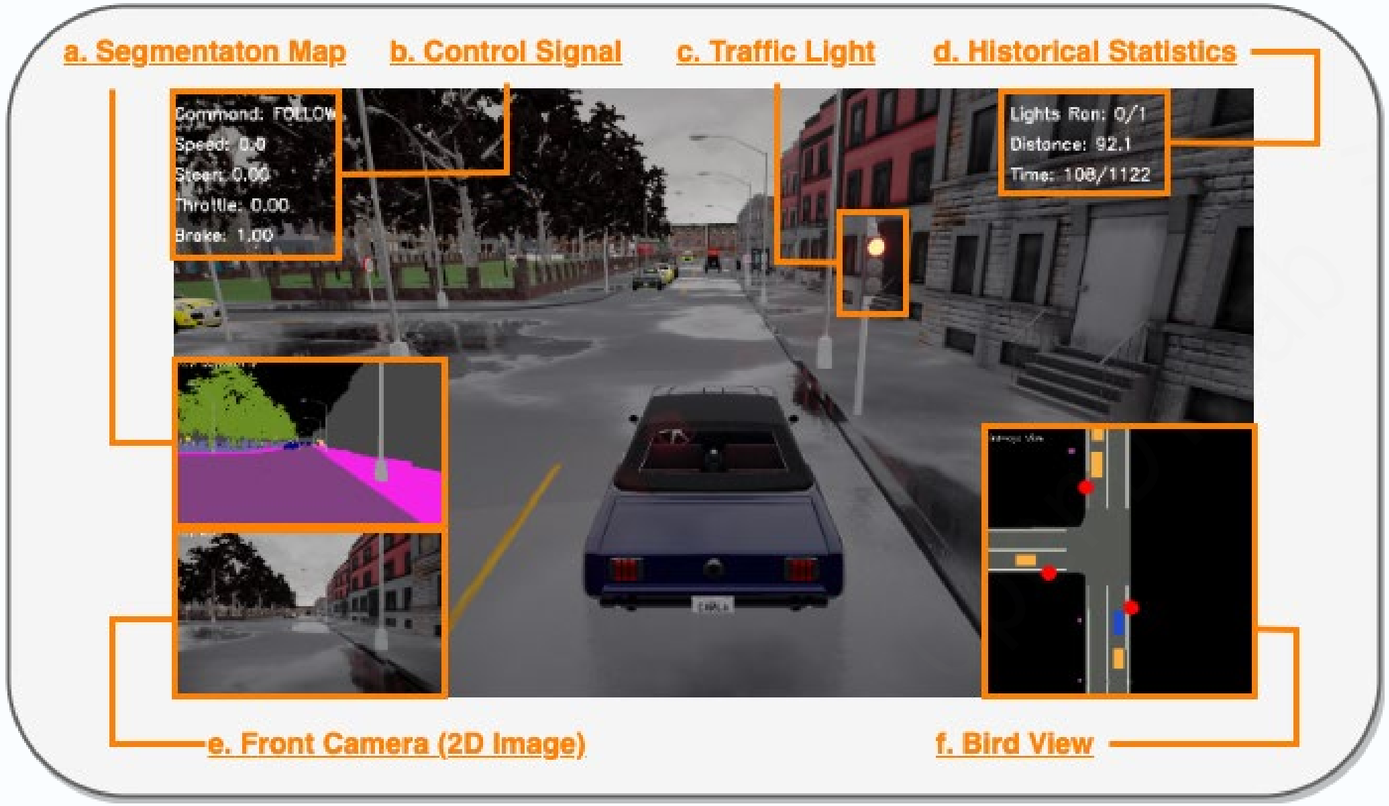
观察空间三部曲

Obs Space Overview

观察空间概述



常见的观察空间类型



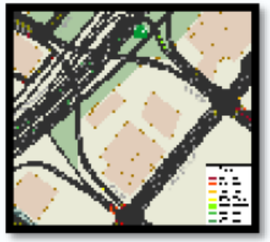
特点：

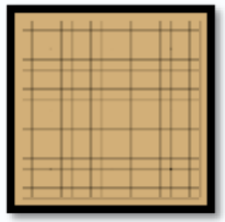
- 特征预处理可以事半功倍
- 高维的观察信息更需要专门设计的神经网络架构
- 复杂的决策问题往往包含多种模态的观察信息

Encoder

编码器

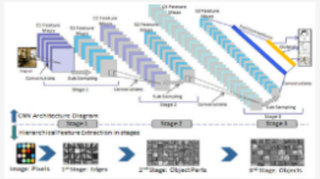
多模态观察空间



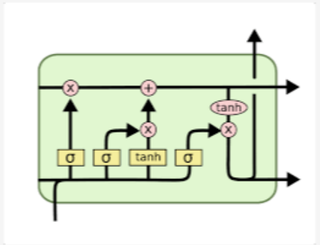


表征编码器（Encoder）

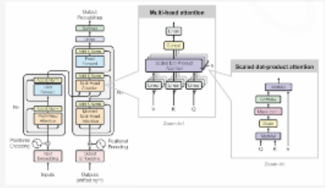
Convolution



LSTM



Transformer



决策预测器（Head）

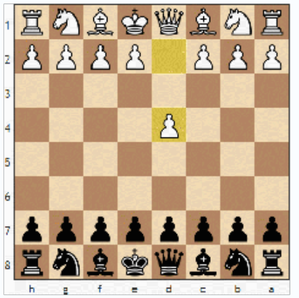
- 离散动作预测
- 连续动作回归
- 连续动作重参数化
- 多维离散
- 层次结构化动作
- 自回归预测

复杂动作空间

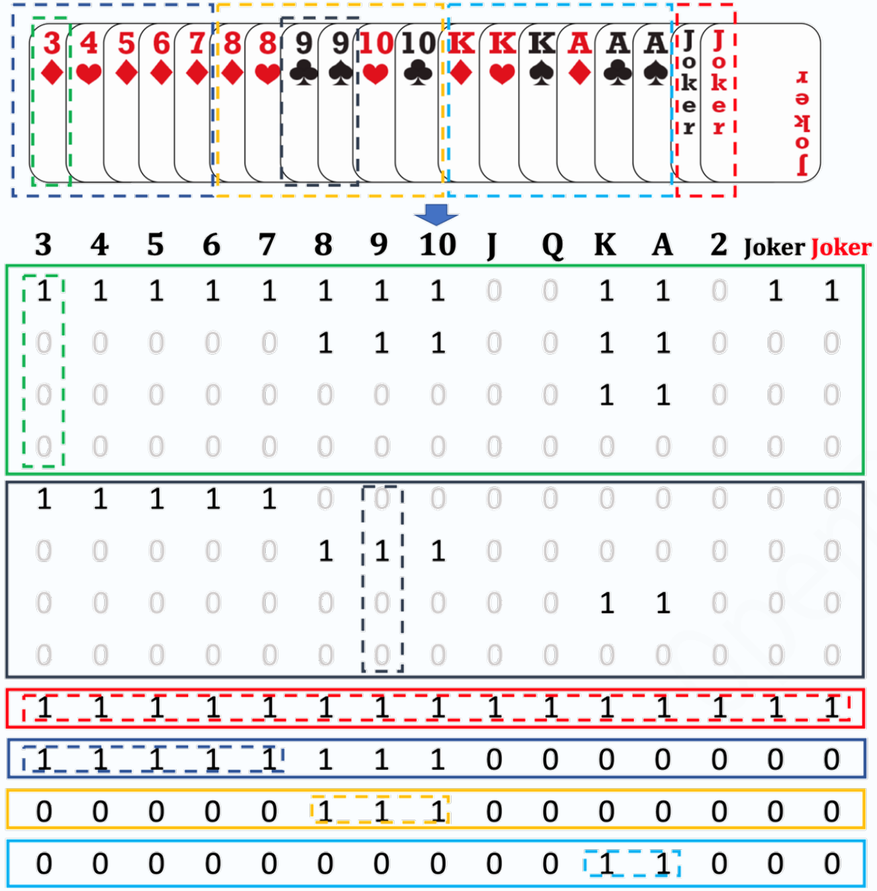
Turn Right

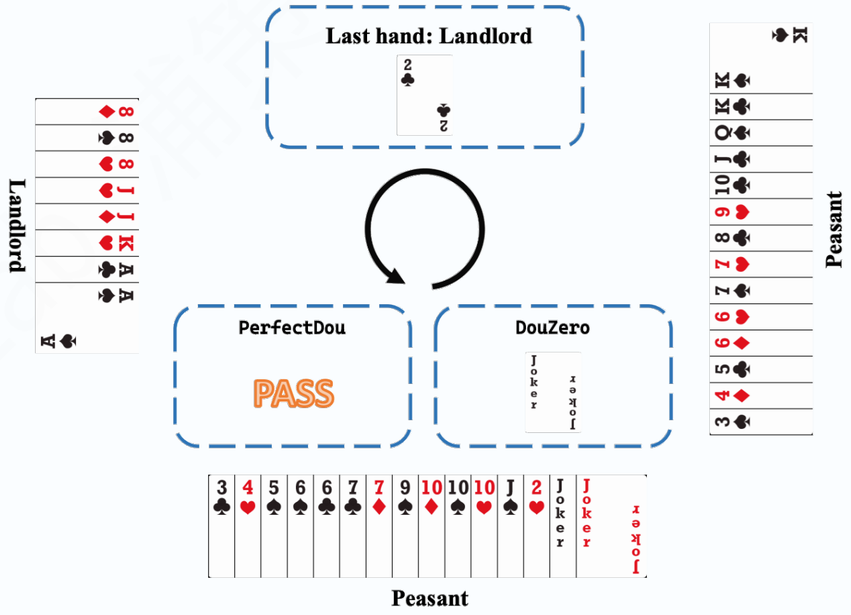
Shoot

Move



向量观察空间

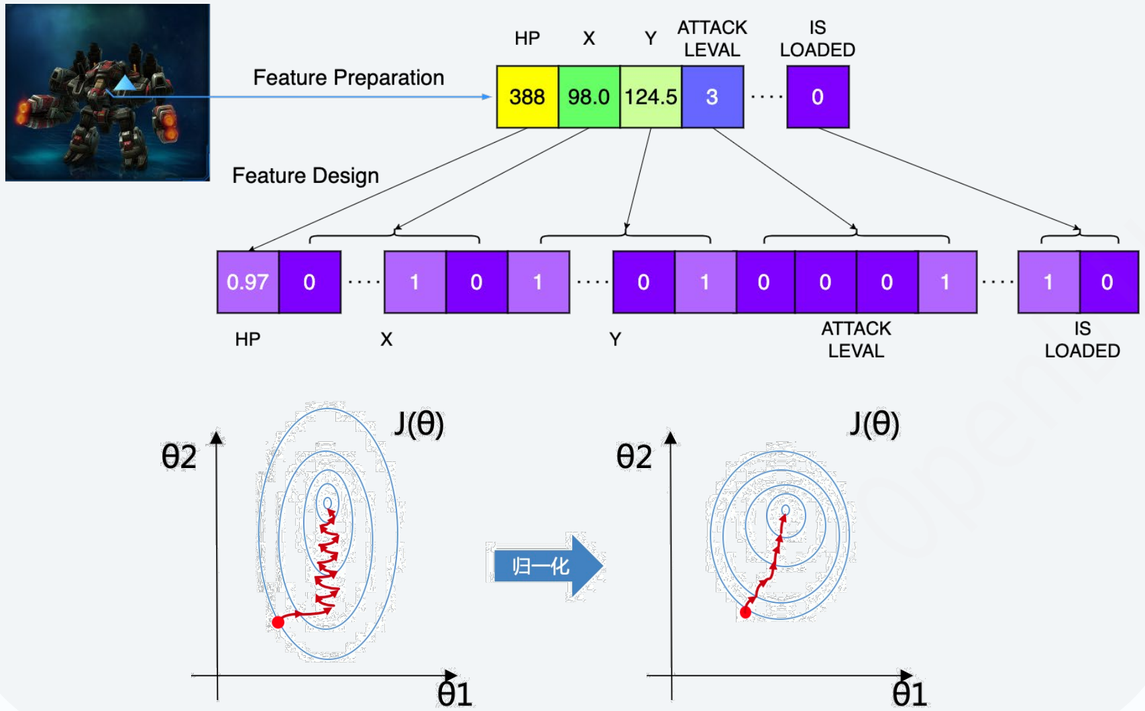




PerfectDou: https://arxiv.org/pdf/2203.16406.pdf

理论：PPO+向量观察预处理

统一量纲



编码设计

One-hot encoding

Color
can
I
eat
the
pizza

One-hot encoding

d0	d1	d2	d3	d4
1	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	0	0	0	1
0	0	0	1	0

Word Embedding

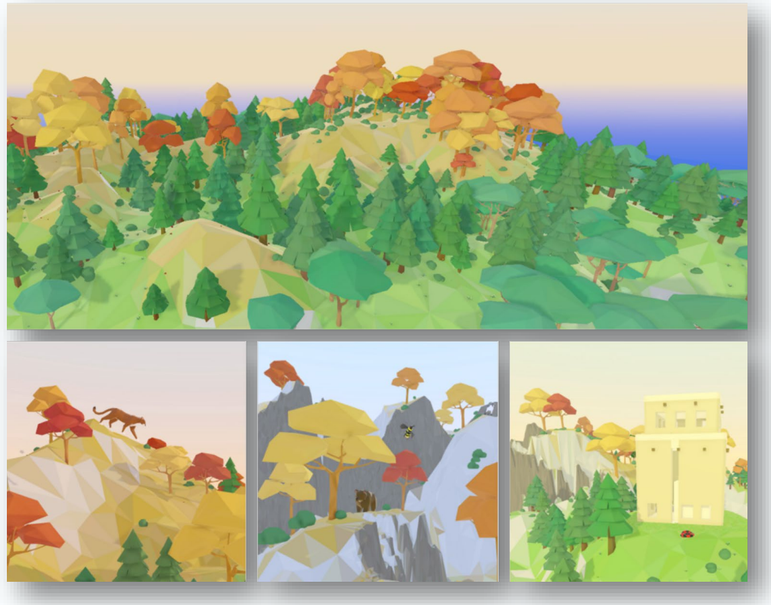
Color
Man
Woman
King
Queen
Orange
Apple
...

Word Embedding

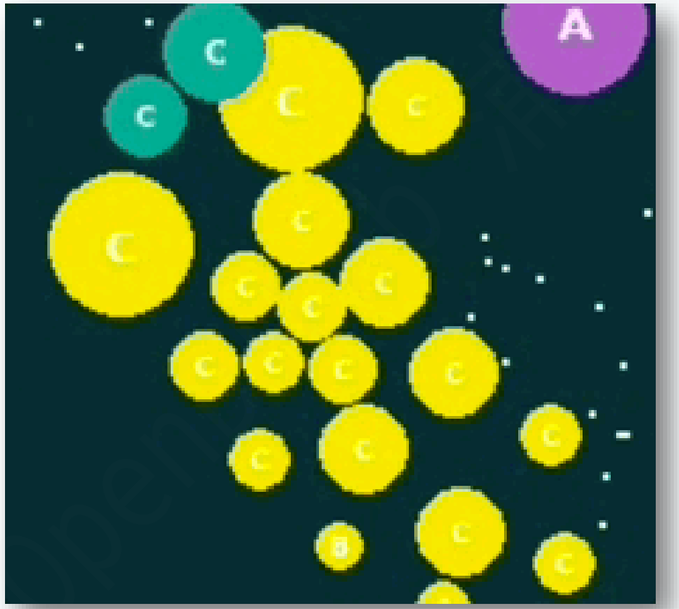
d0	d1	d2	d3	d4
-1.00	0.01	0.03	0.04	0
1.00	0.02	0.02	0.01	0
-0.95	0.93	0.70	0.02	0
0.97	0.95	0.69	0.01	0
0.00	-0.01	0.03	0.95	0
0.01	0.00	-0.02	0.97	0
...	...	...	...	...

理论：PPO+向量观察 不变性

平移不变性



旋转不变性



排列不变性

不变性（invariance） $F(X)=F(T(X))$

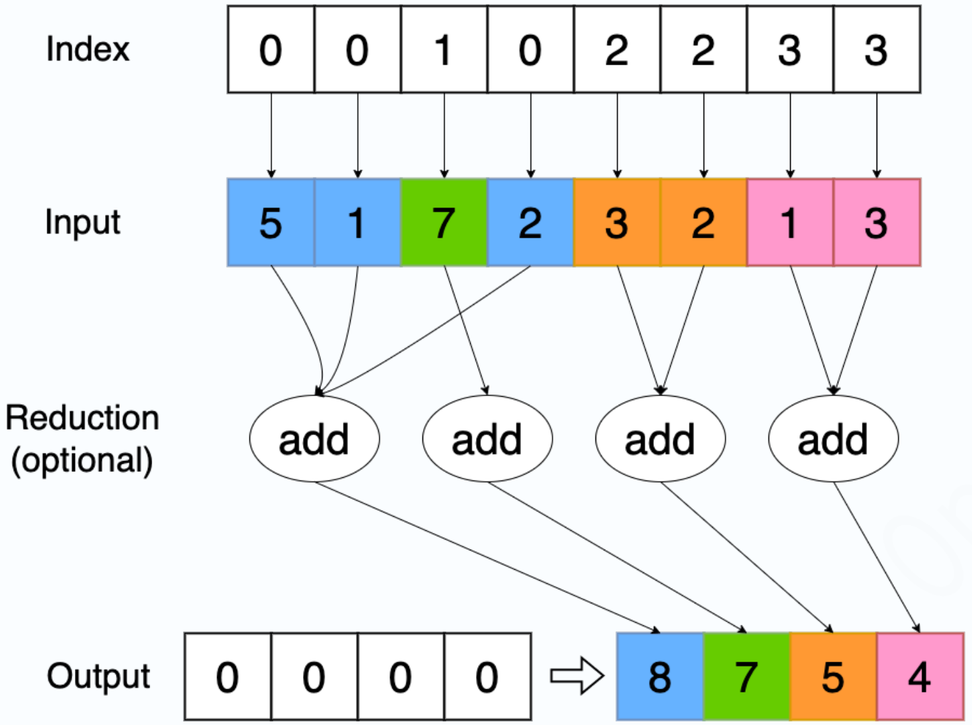
等变性（equivariance） $T(F(X))=F(T(X))$

F指特征编码或神经网络，T指变换，例如平移，旋转，改变顺序

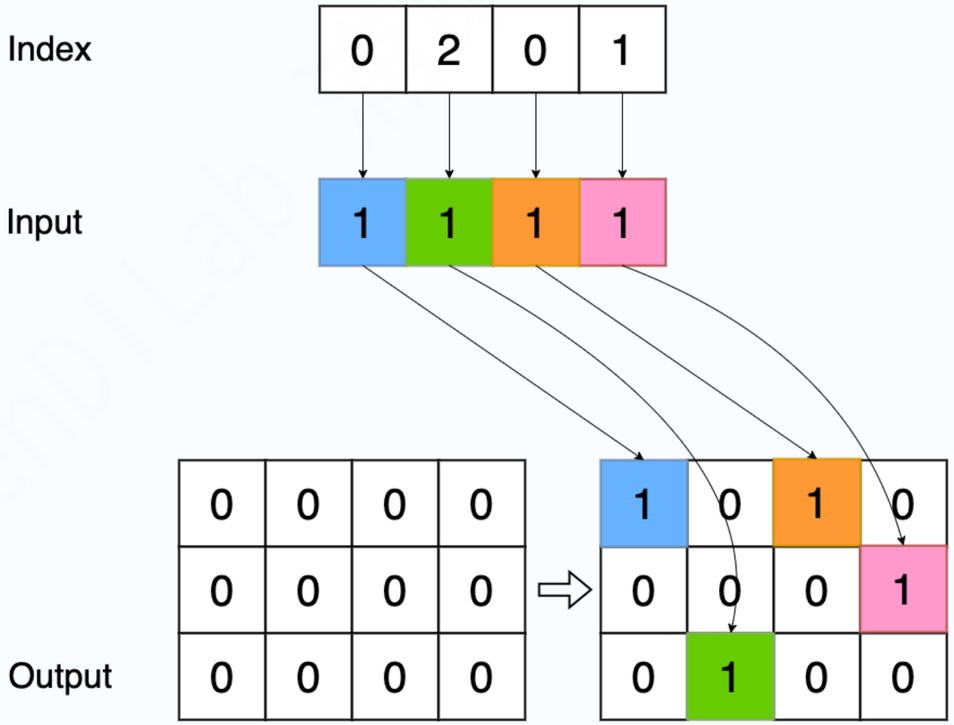
GoBigger: https://github.com/opendilab/GoBigger
ACE + SMAC: https://github.com/opendilab/ACE

代码：特征工程中的张量操作技巧

Scatter: $\text{out}_i = \text{out}_i + \sum_j \text{in}_j$



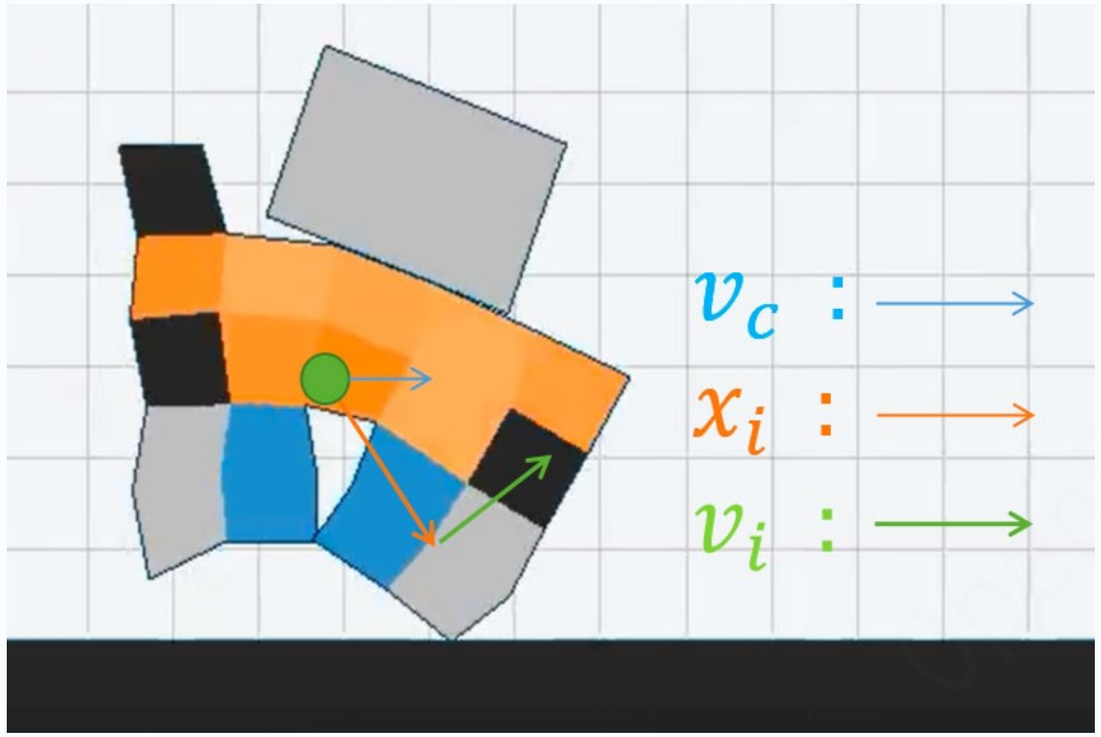
One-Hot Encoding By Scatter

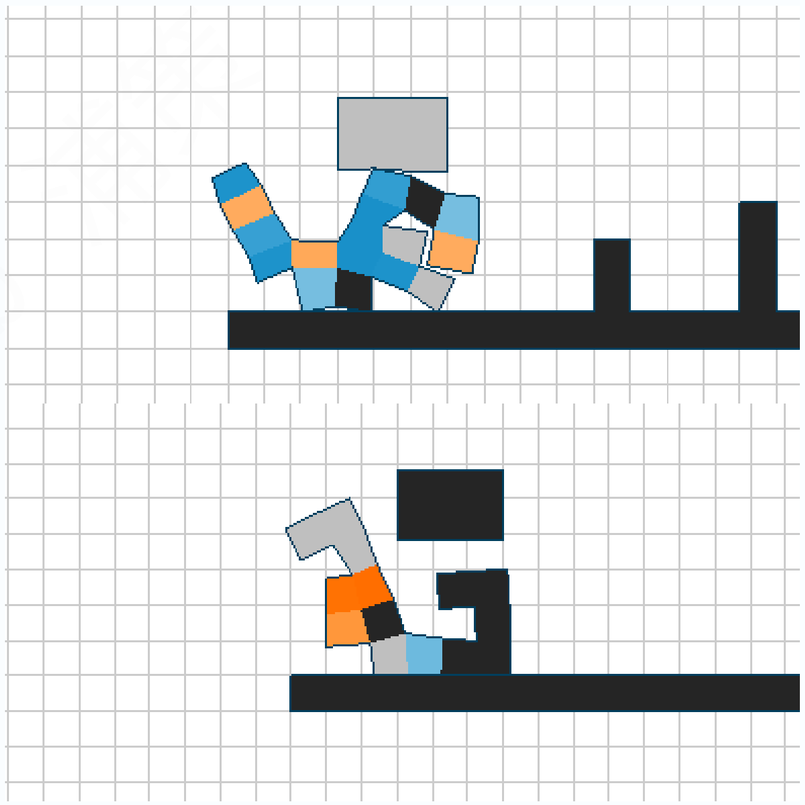


Scatter: https://github.com/rusty1s/pytorch_scatter

完整示例：https://opendilab.github.io/PPOxFamily/

实践：PPO+软体机器人



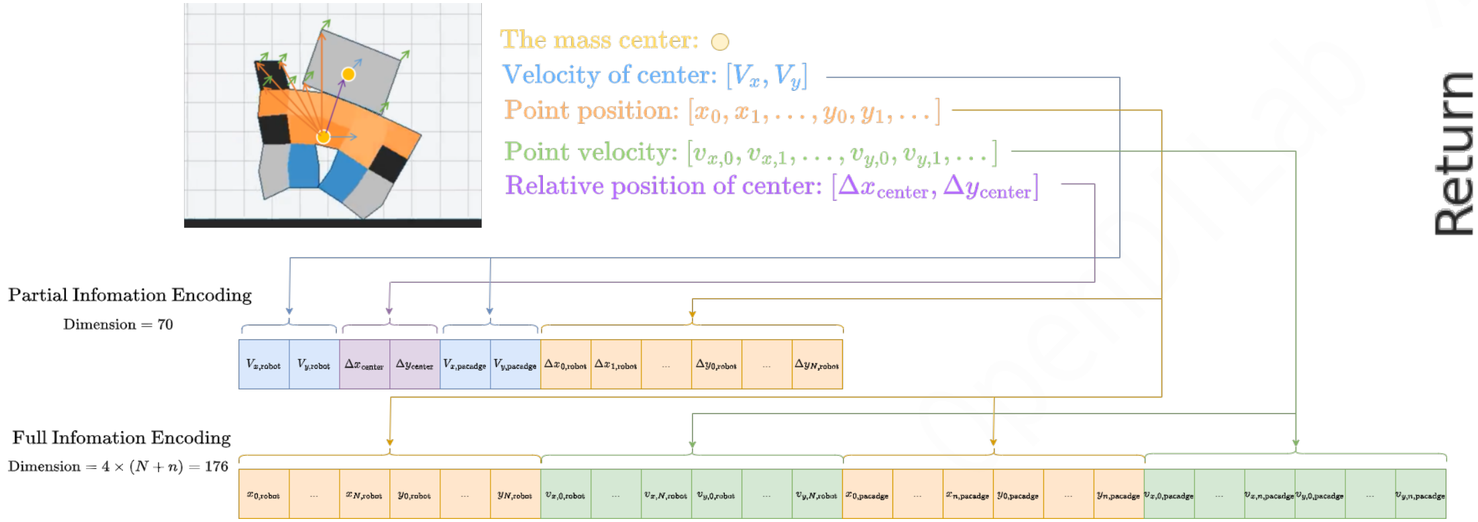


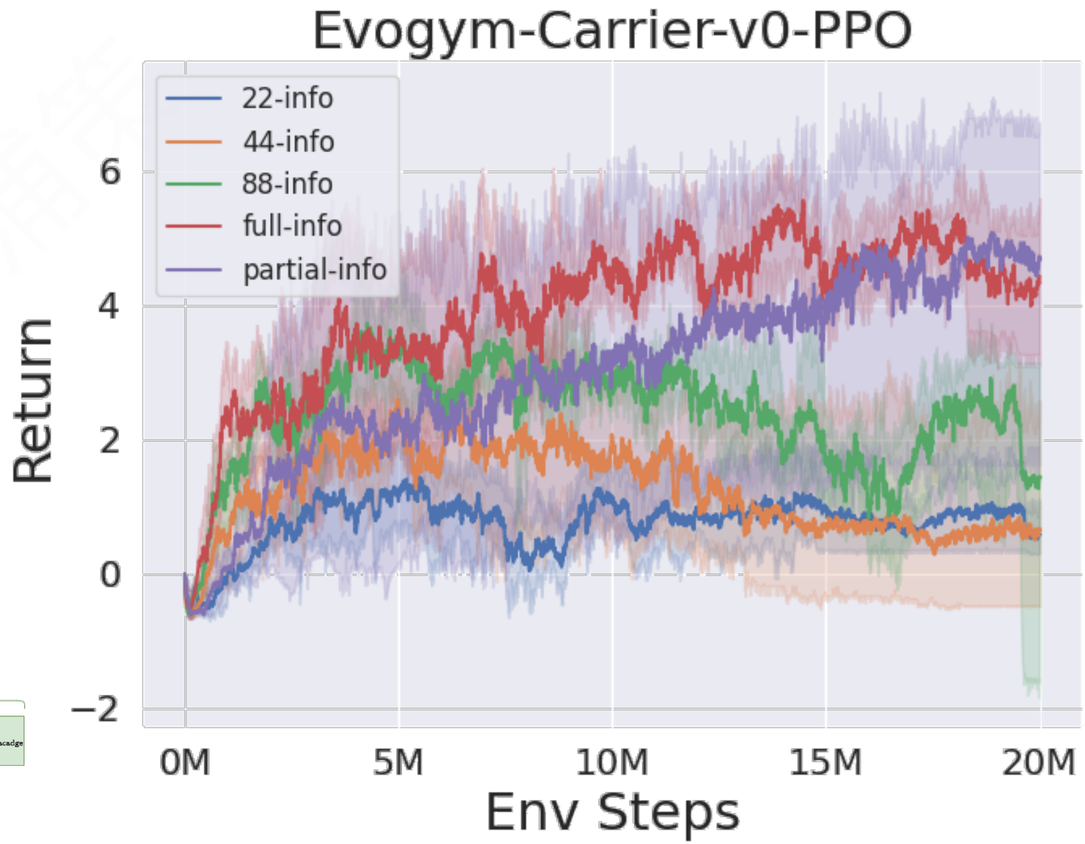
Evogym: https://github.com/EvolutionGym/evogym
DI-engine + Evogym doc: https://di-engine-docs.readthedocs.io/en/latest/13_envs/evogym.html

实践：PPO+软体机器人

观察空间：

机器人全局和局部信息（例如位置，速度）**的编码**

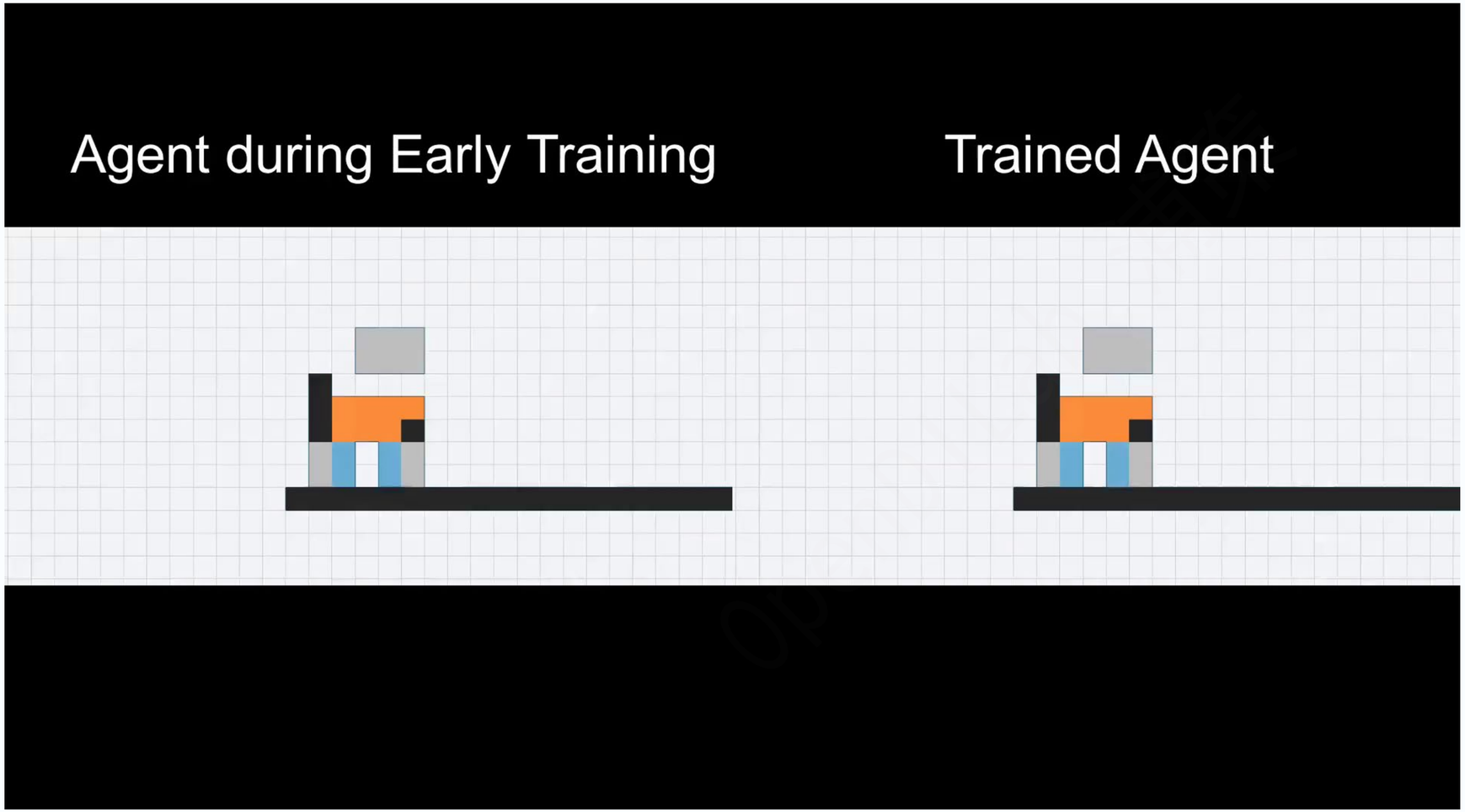




实验细节：https://github.com/opendilab/PPOxFamily/issues/8

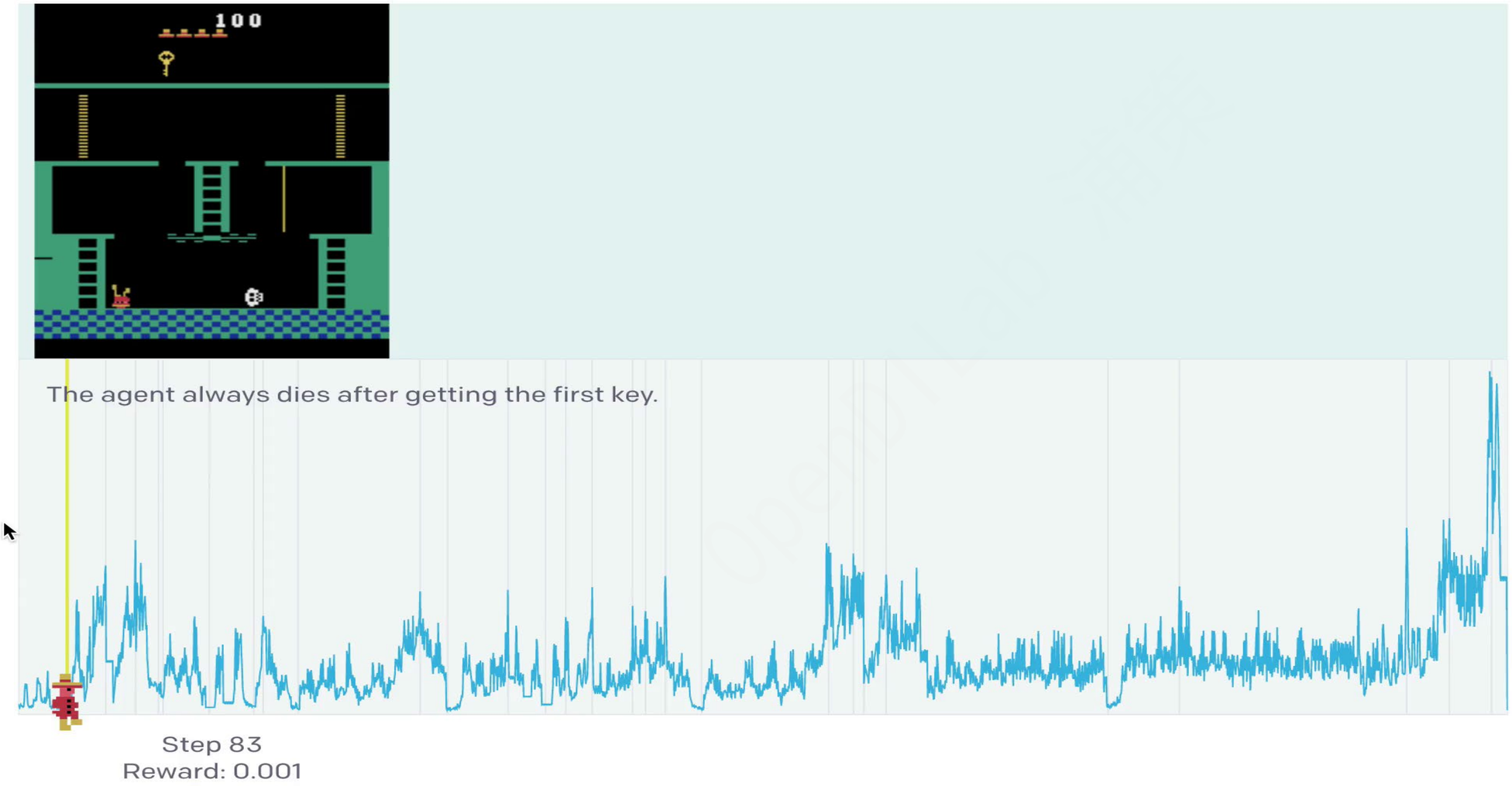
P.S. 88/44/22实验组指只选取软体机器人中部分数量小块的观察信息，数字代表数量

实践：PPO+软体机器人



视频样例：https://github.com/opendilab/PPOxFamily/issues/8

图片观察空间



RND: https://openai.com/blog/reinforcement-learning-with-prediction-based-rewards/

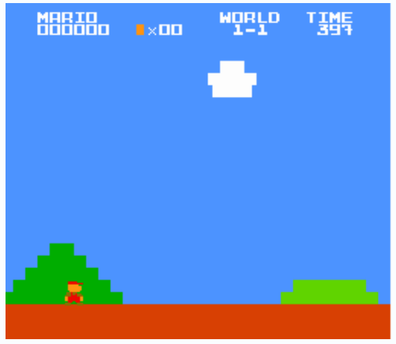
理论：PPO+图片观察预处理

问题：图片数据维度很高但有很多冗余，因此信息密度很低，

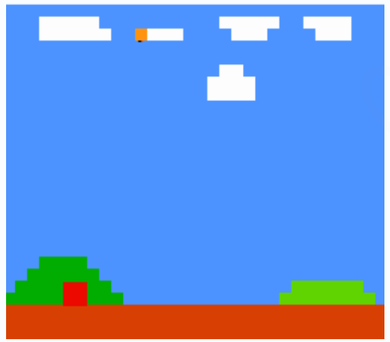
Origin

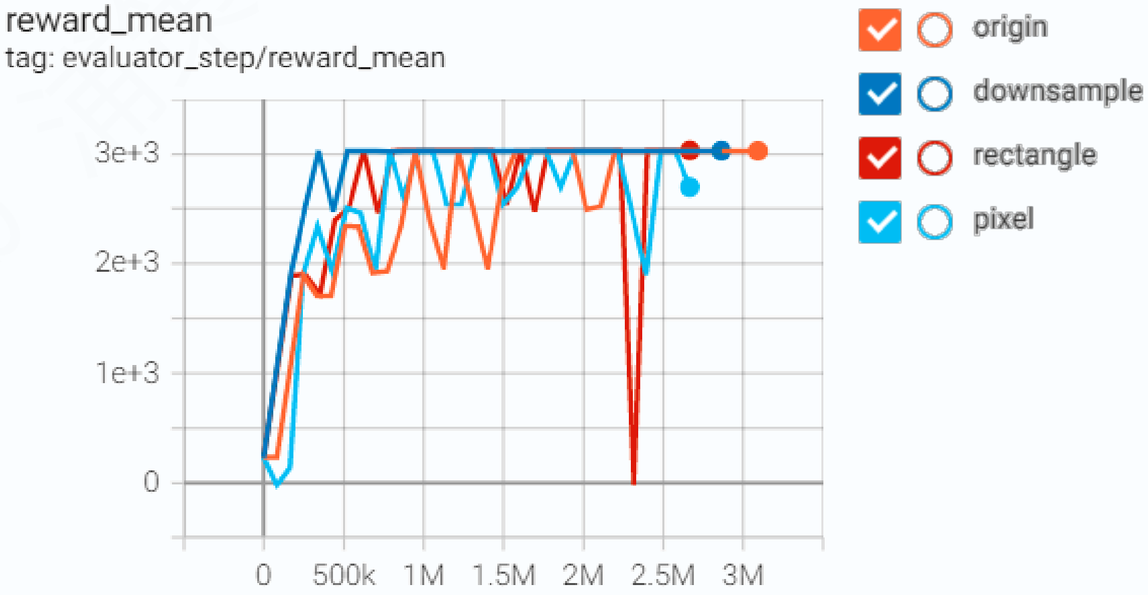
Downsample

Pixel



Rectangle





downsample > pixel >= rectangle >= origin

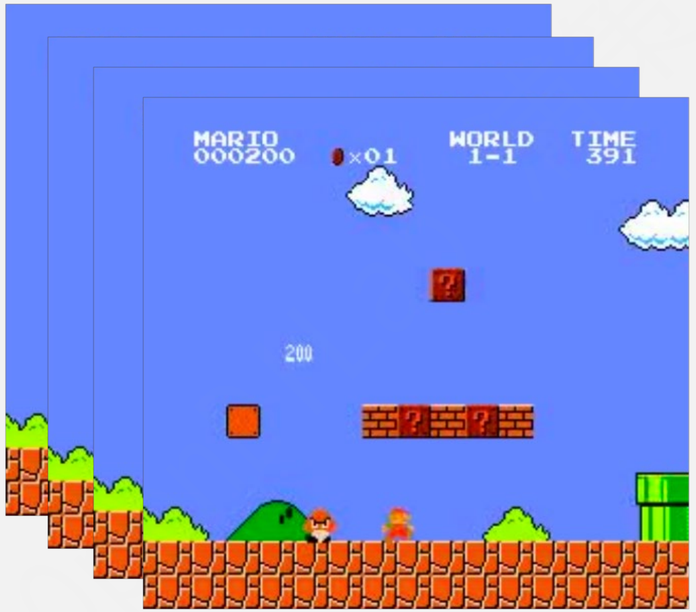
Super Mario Bros: https://github.com/Kautenja/gym-super-mario-bros
DI-engine + Mario Doc: https://di-engine-docs.readthedocs.io/zh_CN/latest/13_envs/gym_super_mario_bros_zh.html
GradCAM: https://github.com/frgfm/torch-cam

理论：PPO+图片观察 运动信息

单帧

State Shape: (1, 84, 84)

叠帧



State Shape: (4, 84, 84)

光流

State Shape: (3, 84, 84)

问题：单帧图片无法表示运动信息（例如方向，速度）

解决思路：堆叠连续4帧环境帧为一个动作帧 or 提取光流

Optical Flow: https://en.wikipedia.org/wiki/Optical_flow

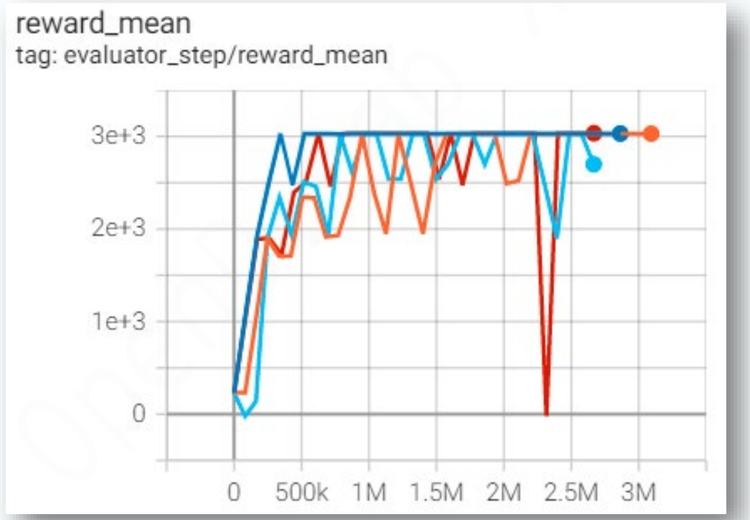
P.S. 运动信息问题进一步延伸可以拓展到 POMDP，相关内容将会在课程第五讲详细展开

理论：PPO+图片观察 运动信息

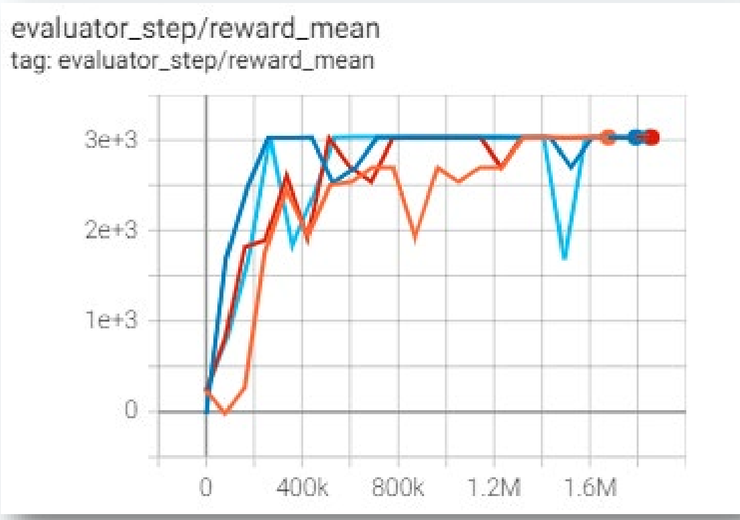
origin
downsample
rectangle
pixel

单帧

叠帧

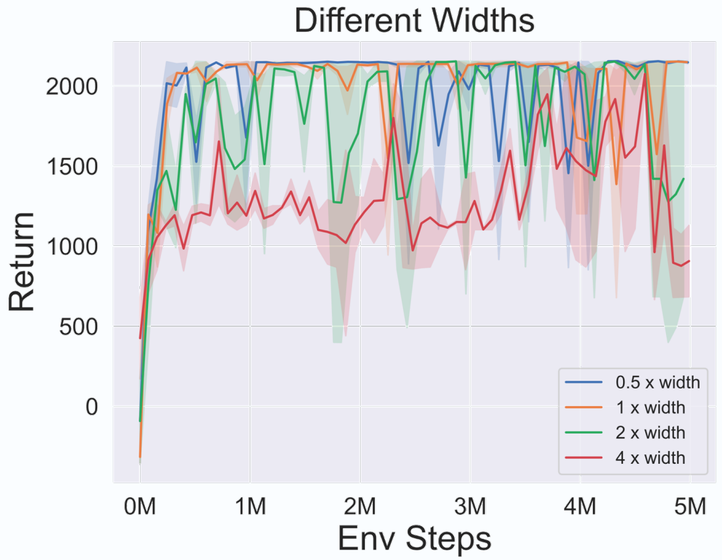


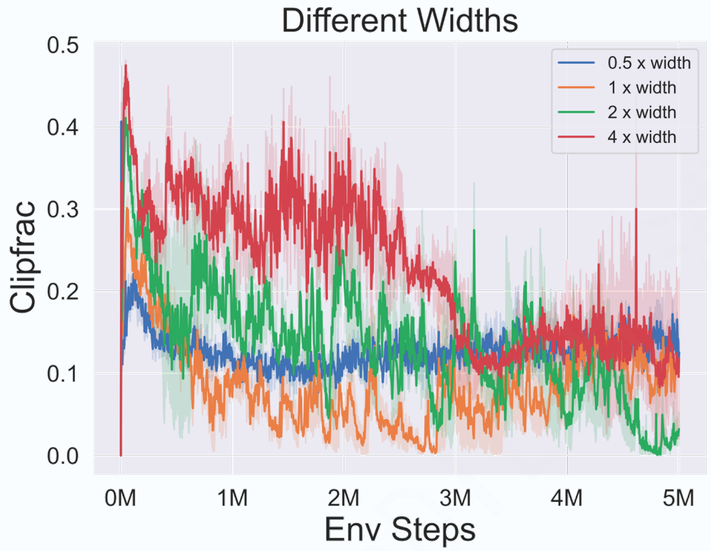
光流

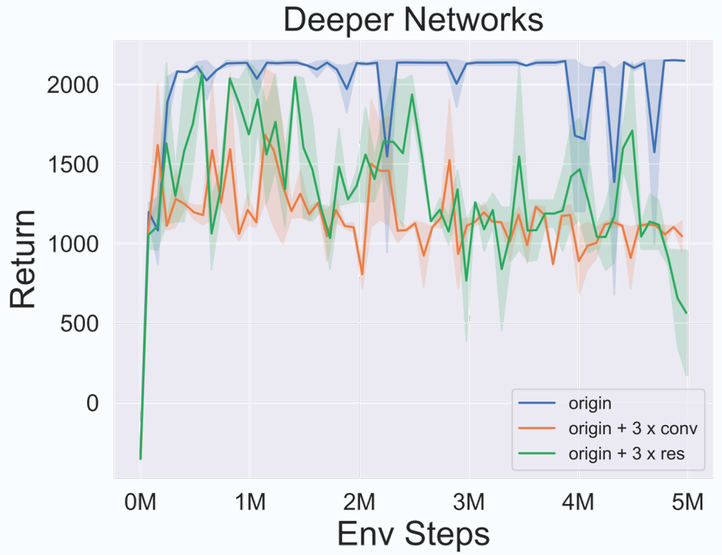


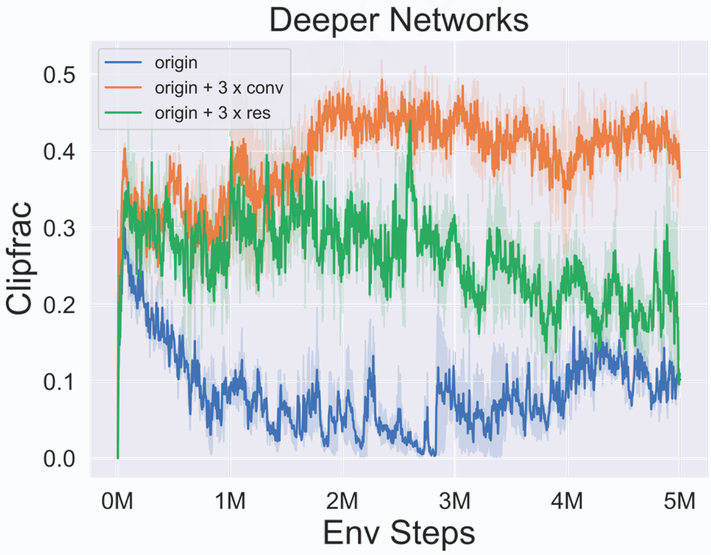
实验细节：https://github.com/opendilab/PPOxFamily/issues/8

理论：PPO+图片观察 网络架构







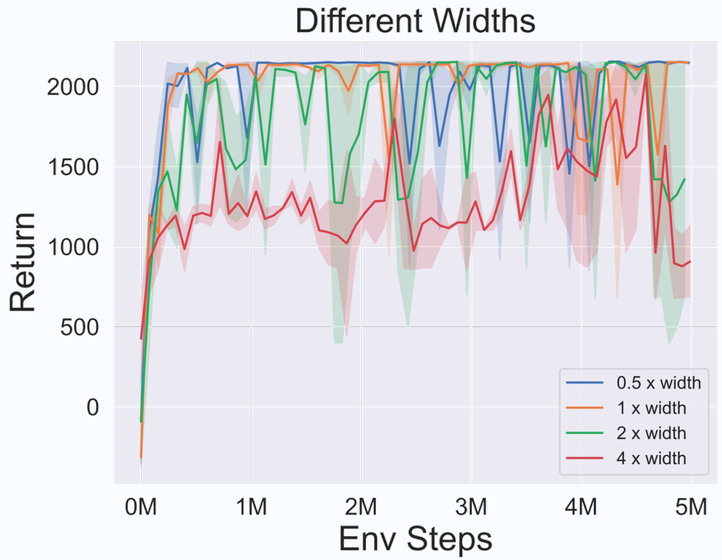


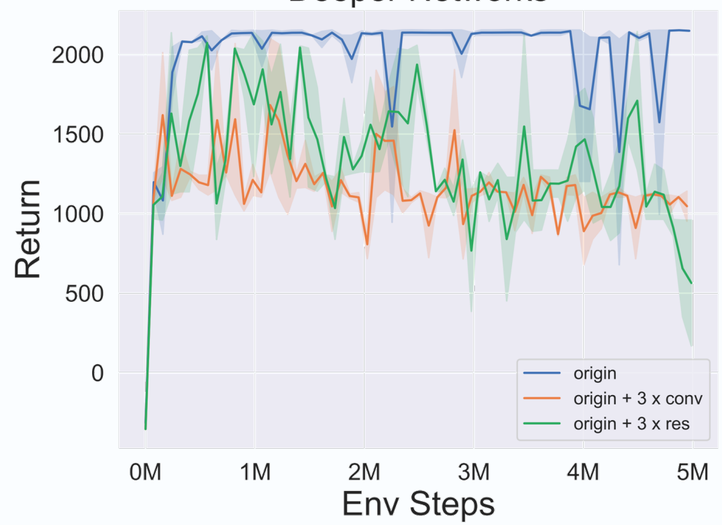
Key Facts:

- 单纯的加深 or 加宽网络并不总是有效的

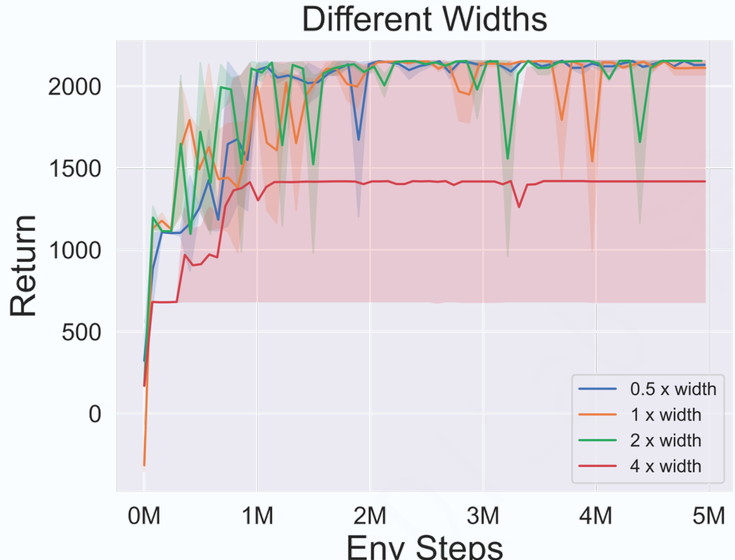
理论：PPO+图片观察 网络架构

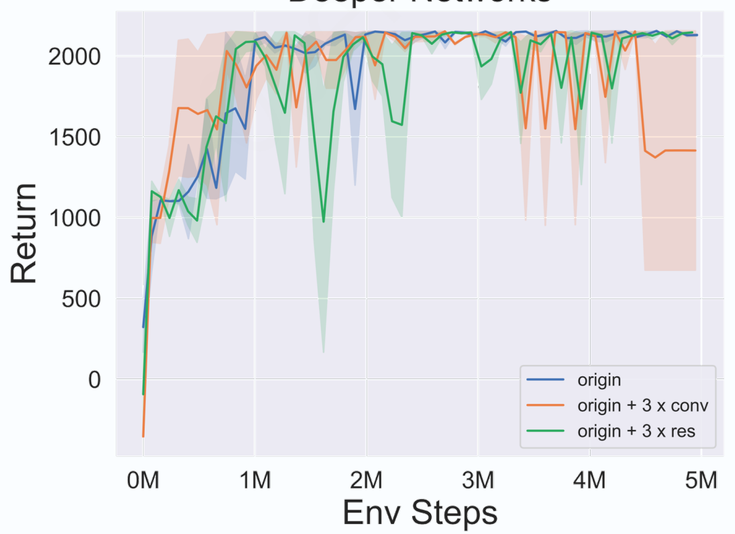
without LN





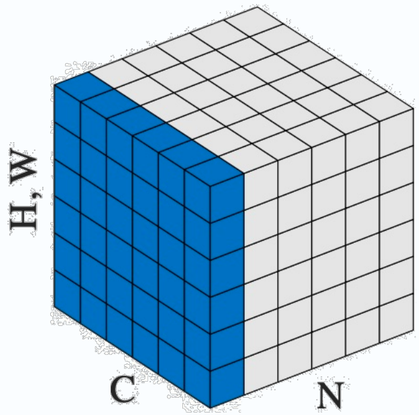
with LN





Key Facts:

- 网络一旦变大，就需要类似 LN 这样的模块来稳定训练



代码：巧用 Env Wrapper

Stars 1.1k | bilibili video course | Follow @opendilab 44

View code on GitHub

PyTorch 代码实现：用于 PPO 训练的超级马里奥兄弟仿真环境设置

Related Link

概述

创建超级马里奥环境，并使用一系列 Env Wrapper 来实现各种预处理变换

```python
from ding.envs import DingEnvWrapper
from ding.envs.env_wrappers import MaxAndSkipWrapper, WarpFrameWrapper, ScaledFloatFrameWrapper, FrameStackWrapper, \
    EvalEpisodeReturnEnv
import gym_super_mario_bros
from nes_py.wrappers import JoypadSpace


def wrapped_mario_env():
```

转换为 DI-engine 的环境接口格式

```python
    return DingEnvWrapper(
```

将动作空间简化为2维的离散空间（向右走，向右上跳）

```python
        JoypadSpace(gym_super_mario_bros.make("SuperMarioBros-1-1-v0"), [["right"], ["right", "A"]]),
```

添加各类 Env Wrapper 的相关配置

```python
        cfg={
            'env_wrapper': [
```

添加环境帧采样 Wrapper，每隔四帧采样一帧作为智能体的观测信息

```python
                lambda env: MaxAndSkipWrapper(env, skip=4),
```

添加图片格式和尺寸统一化 Wrapper

```python
                lambda env: WarpFrameWrapper(env, size=84),
```

完整示例：https://opendilab.github.io/PPOxFamily/

实践：PPO+超级马里奥

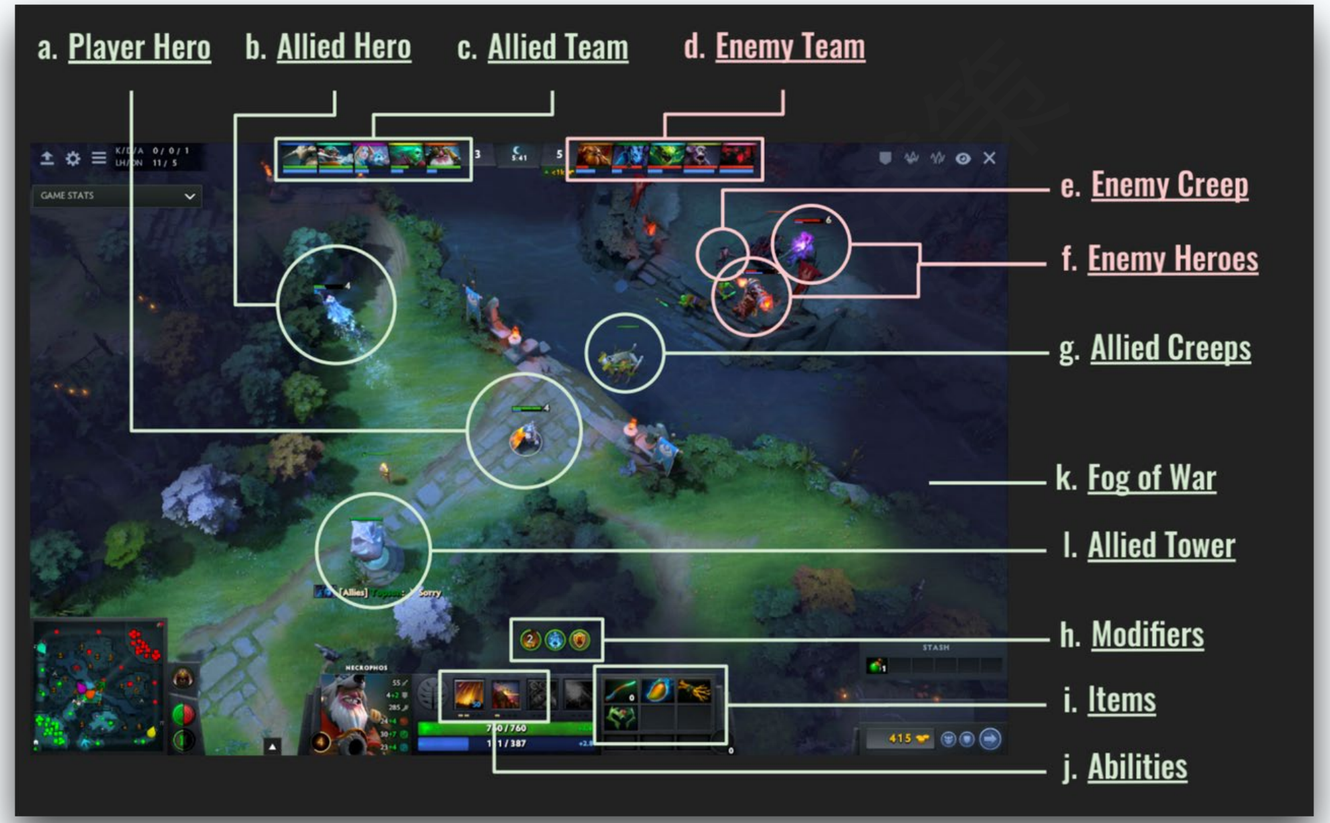
Level 1-1

Level 1-4

Super Mario Bros: https://github.com/Kautenja/gym-super-mario-bros
DI-engine + Mario Doc: https://di-engine-docs.readthedocs.io/zh_CN/latest/13_envs/gym_super_mario_bros_zh.html
视频样例：https://github.com/opendilab/PPOxFamily/issues/8

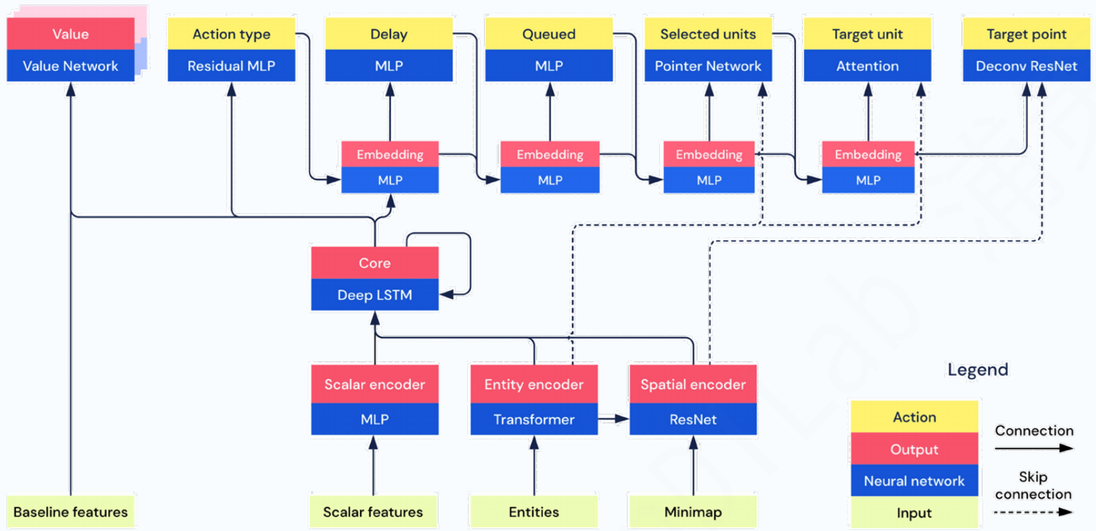
Struct

复杂结构化观察空间



OpenAI Five: https://openai.com/five/

理论：PPO+结构化观察 概述

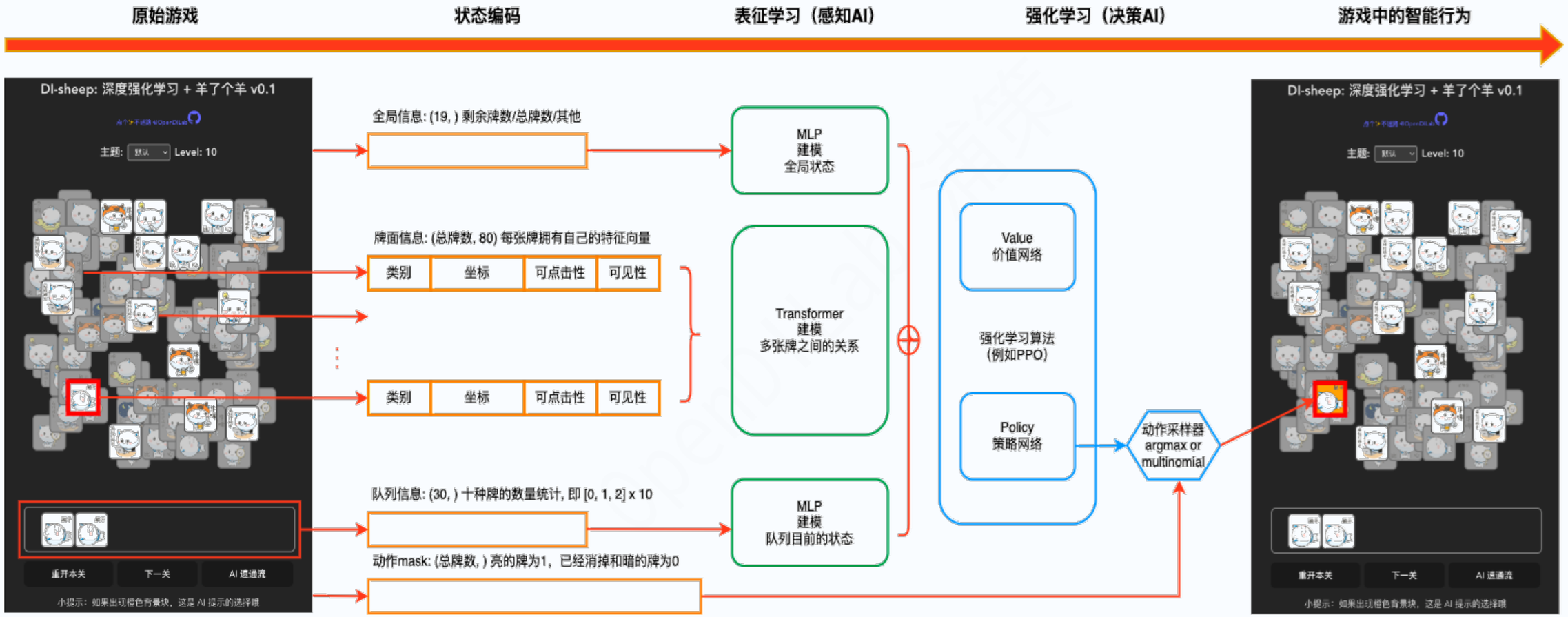


解决方案：

- 不同模态的数据需要结合对应领域经典的建模方法和网络结构设计
- 重要的信息利用 skip connection 进行补充
- 要根据决策任务的特点灵活组合各种网络结构设计方法

AlphaStar: https://www.deepmind.com/blog/alphastar-grandmaster-level-in-starcraft-ii-using-multi-agent-reinforcement-learning

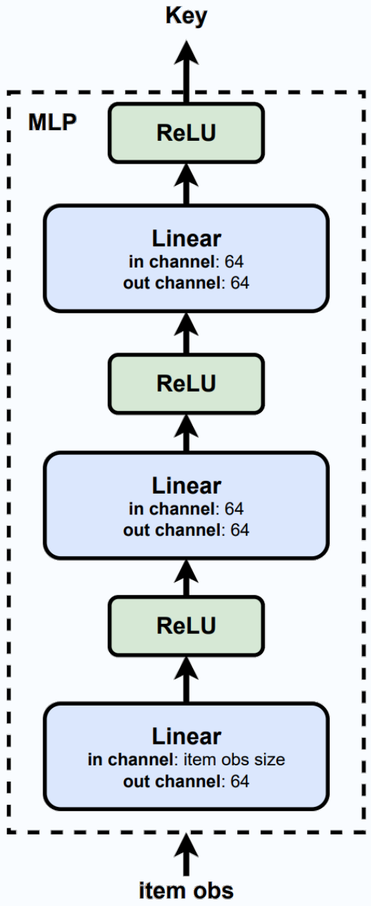
实践：PPO+羊了个羊



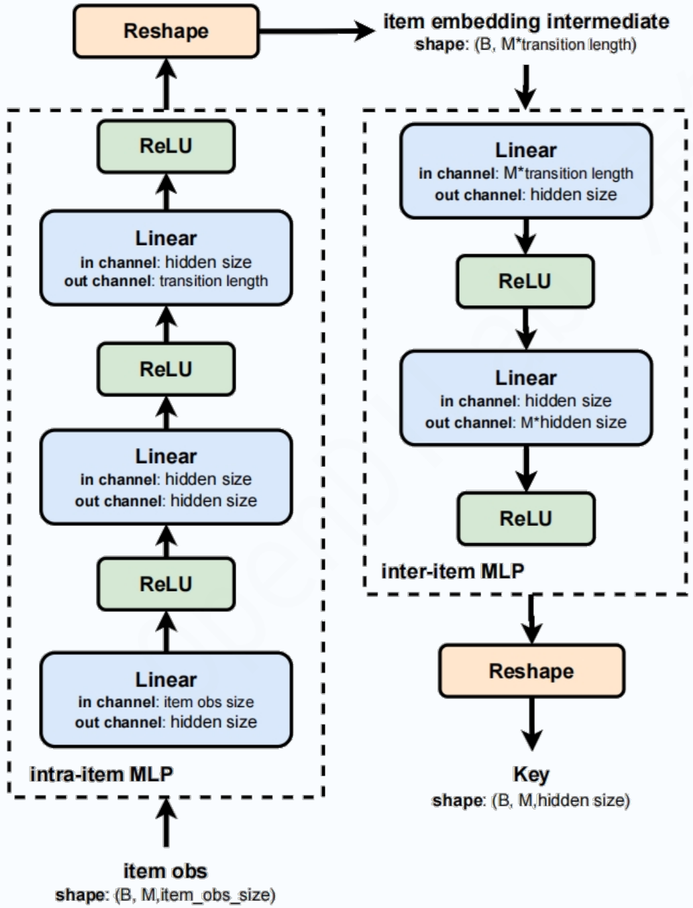
DI-sheep: https://github.com/opendilab/DI-sheep

实践：PPO+羊了个羊

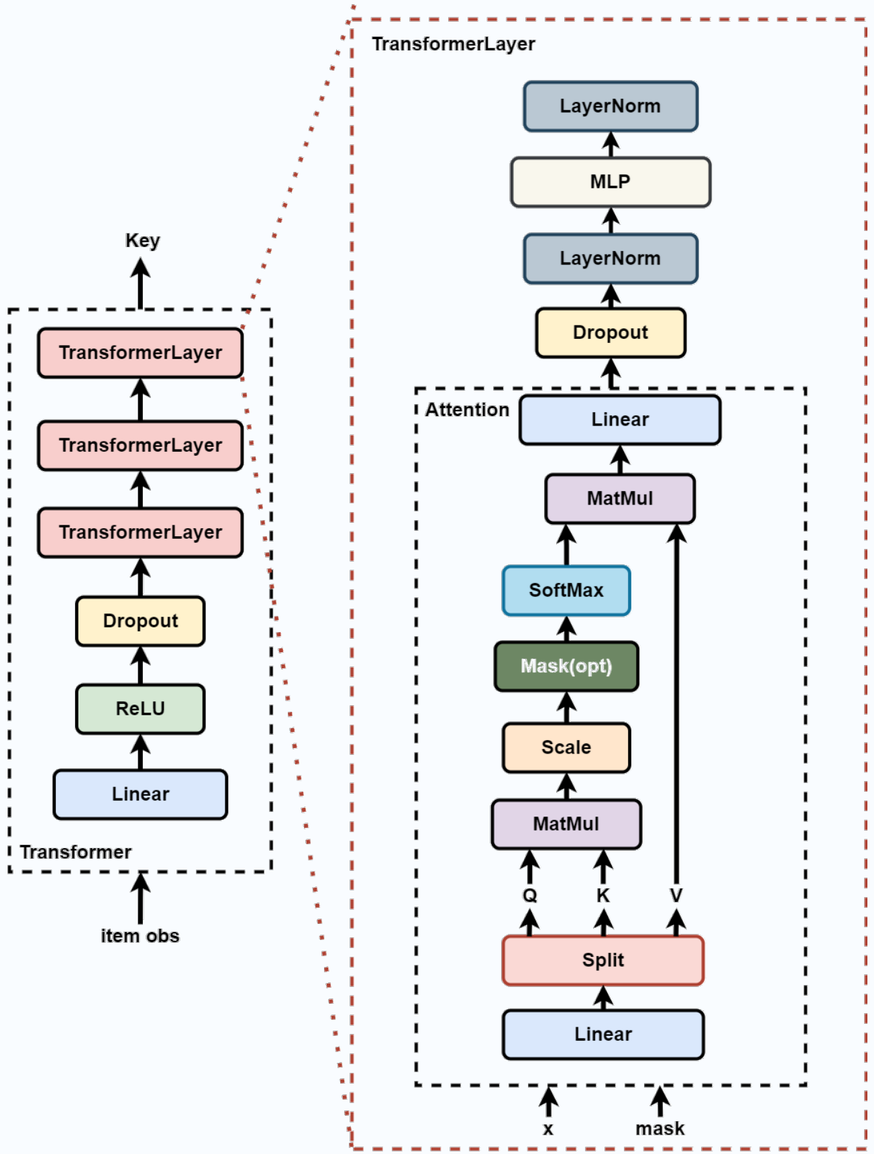
不同的牌面信息编码器



MLP



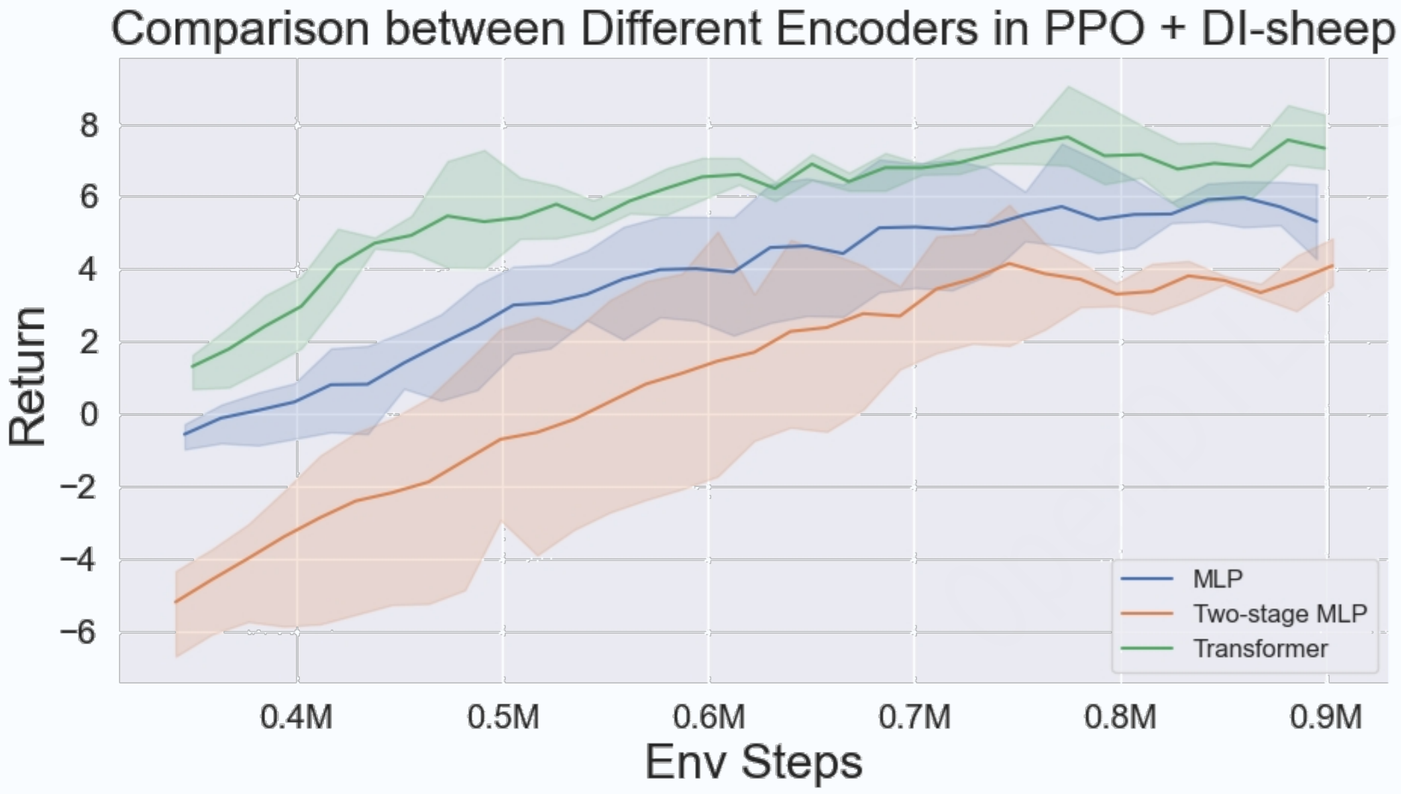
Two-Stages MLP



Transformer

实践：PPO+羊了个羊

（实验对比）不同的牌面信息编码器



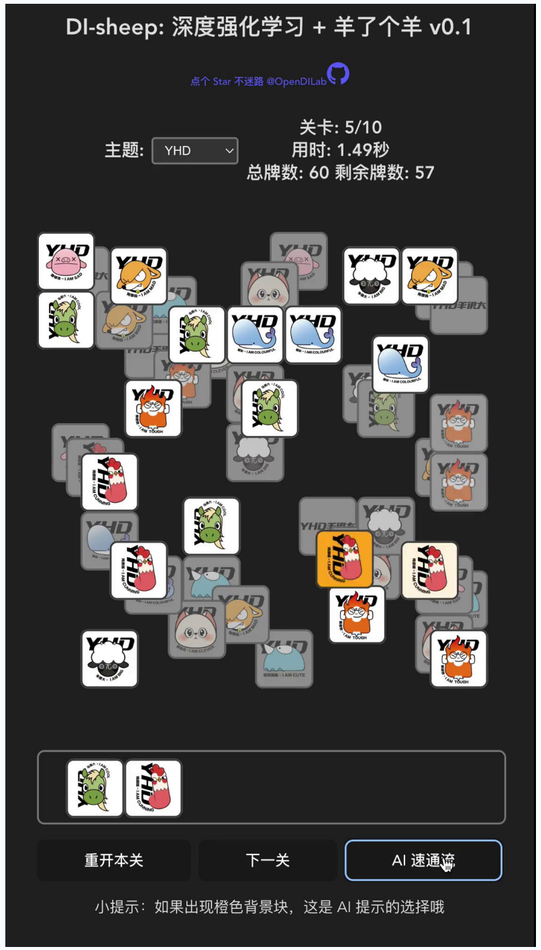
实验细节：https://github.com/opendilab/PPOxFamily/issues/8

Key Facts:

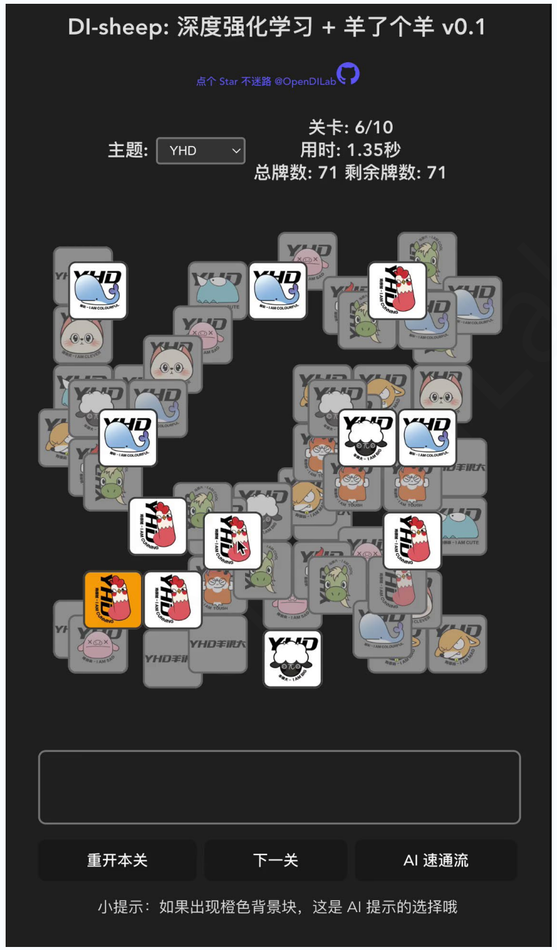
- 环境角度：多张牌之间的关系建模很重要
- 计算角度：参数量大小不一定重要，合理利用参数进行计算很重要
- 算法角度：网络结构的设计代表问题的上限，网络的训练方式决定问题的下限

实践：PPO+羊了个羊

乖巧型



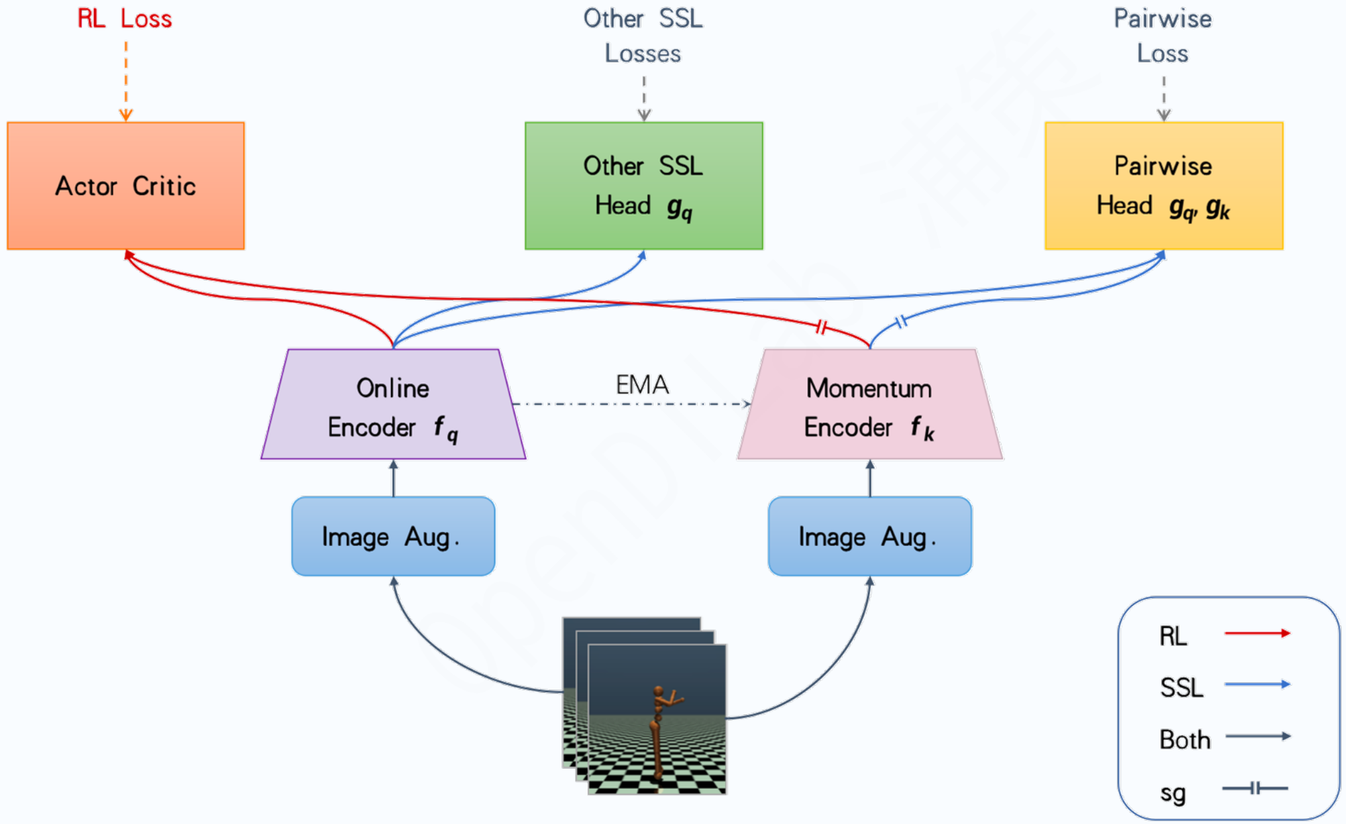
叛逆型



合作型

实验细节：https://github.com/opendilab/PPOxFamily/issues/8 科普文章：https://mp.weixin.qq.com/s/4Z3WtkcWRp6x4x60RVELfQ

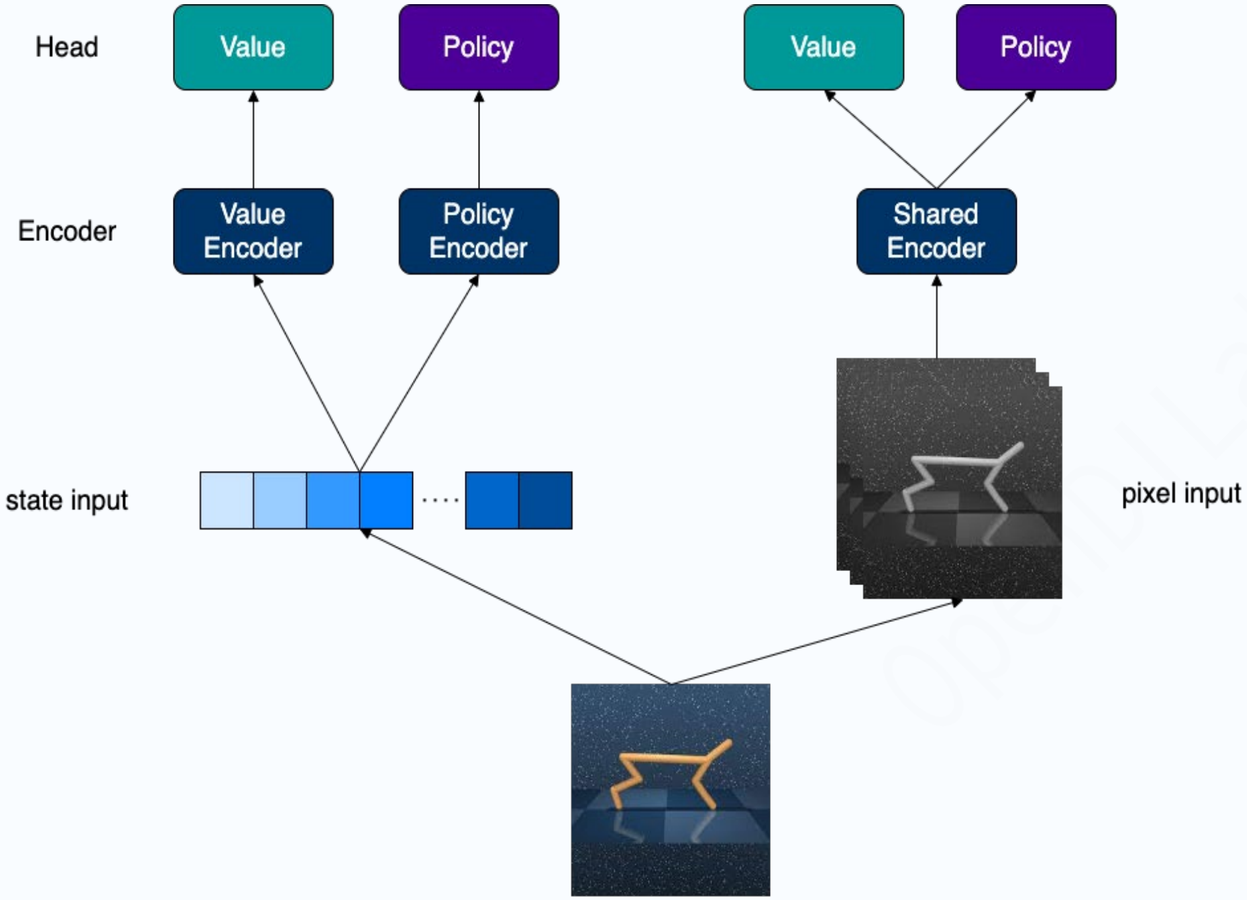
通用观察空间训练方法



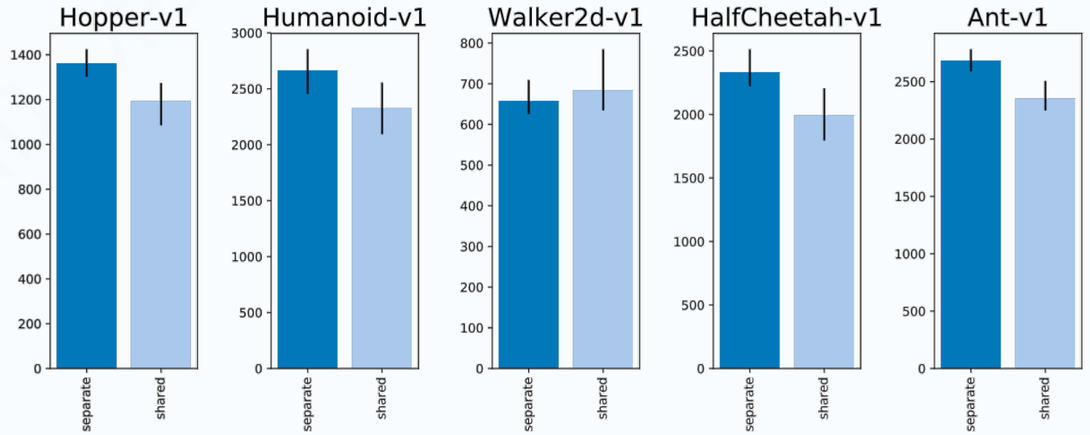
Does Self-supervised Learning Really Improve Reinforcement Learning from Pixels? https://openreview.net/forum?id=fVslVNBfjd8

观察空间表征学习补充材料：https://github.com/opendilab/PPOxFamily/blob/main/chapter3_obs/chapter3_supp_representation.pdf

理论：Actor和Critic是否共享编码器



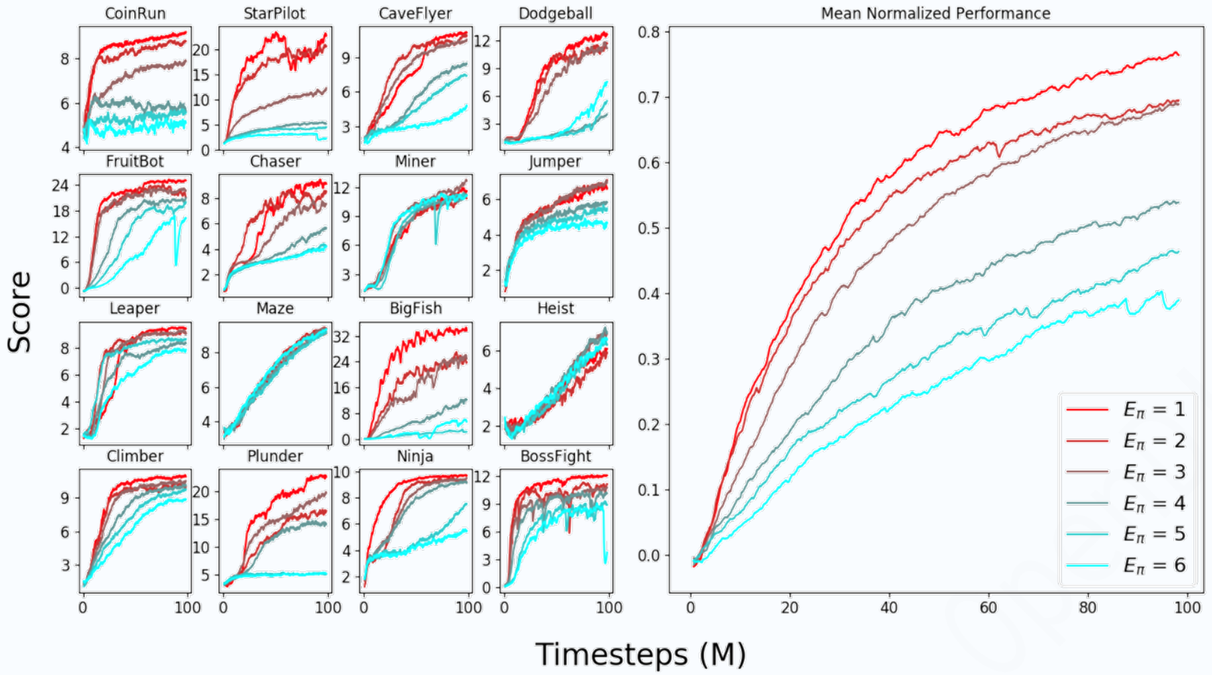
经验原则：高维共享，低维不共享



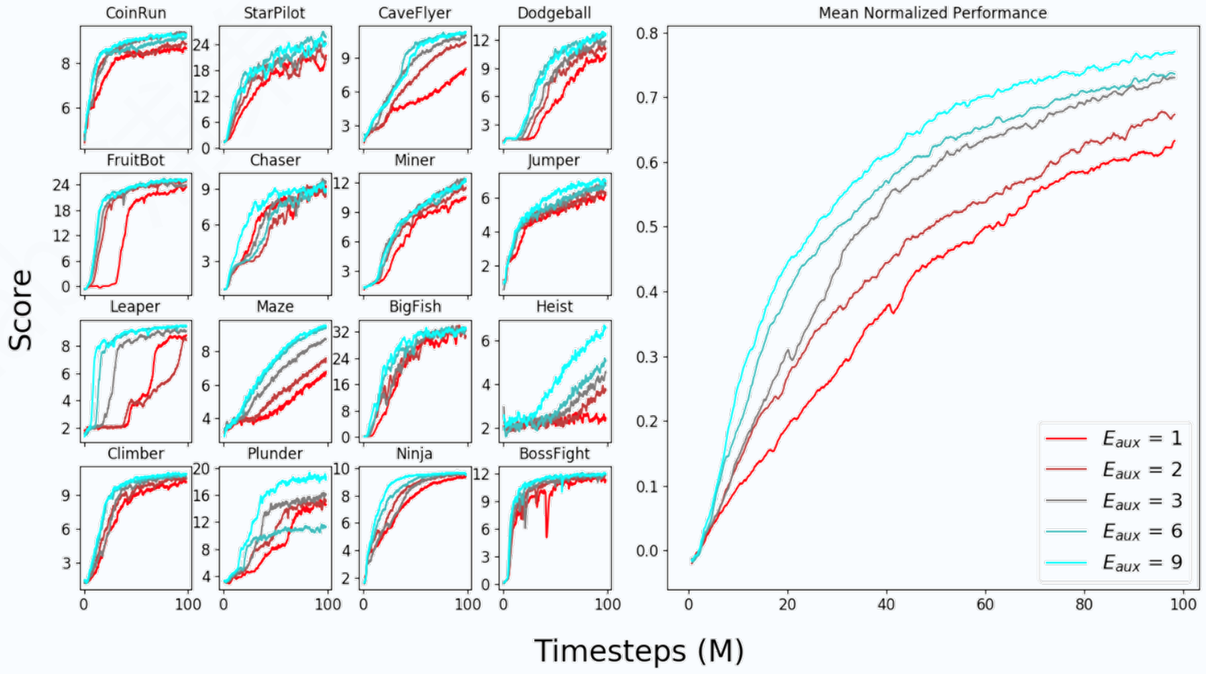
What Matters for On-Policy Deep Actor-Critic Methods? A Large-Scale Study:
https://openreview.net/forum?id=nIAxjsniDzg

理论：Actor和Critic训练的不同特性

Policy Reuse



Value Reuse

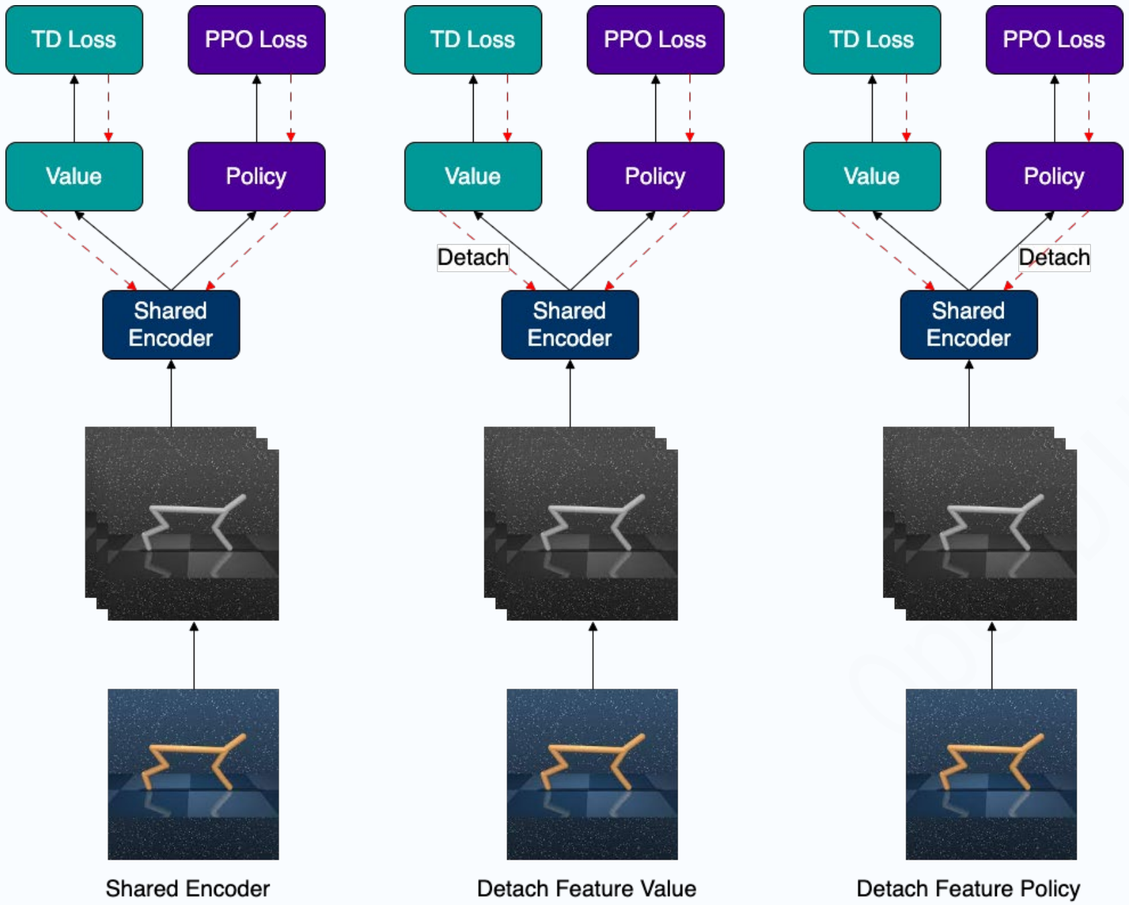


策略函数（Policy）：对数据新旧程度和复用次数很敏感，尽可能每个数据只用一次

价值函数（Value）：需要更高的数据多样性，可以通过复用数据来提升性能

PPG: https://arxiv.org/pdf/2009.04416.pdf
PPG 详解补充材料：https://github.com/opendilab/PPOxFamily/blob/main/chapter3_obs/chapter3_supp_ppg.pdf

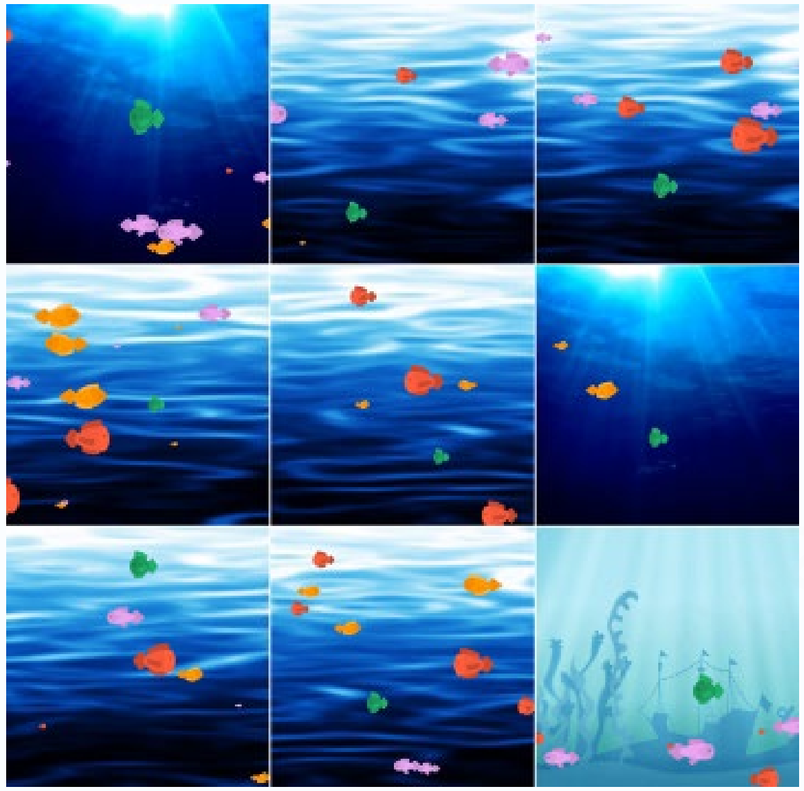
理论：Actor和Critic分离编码器训练

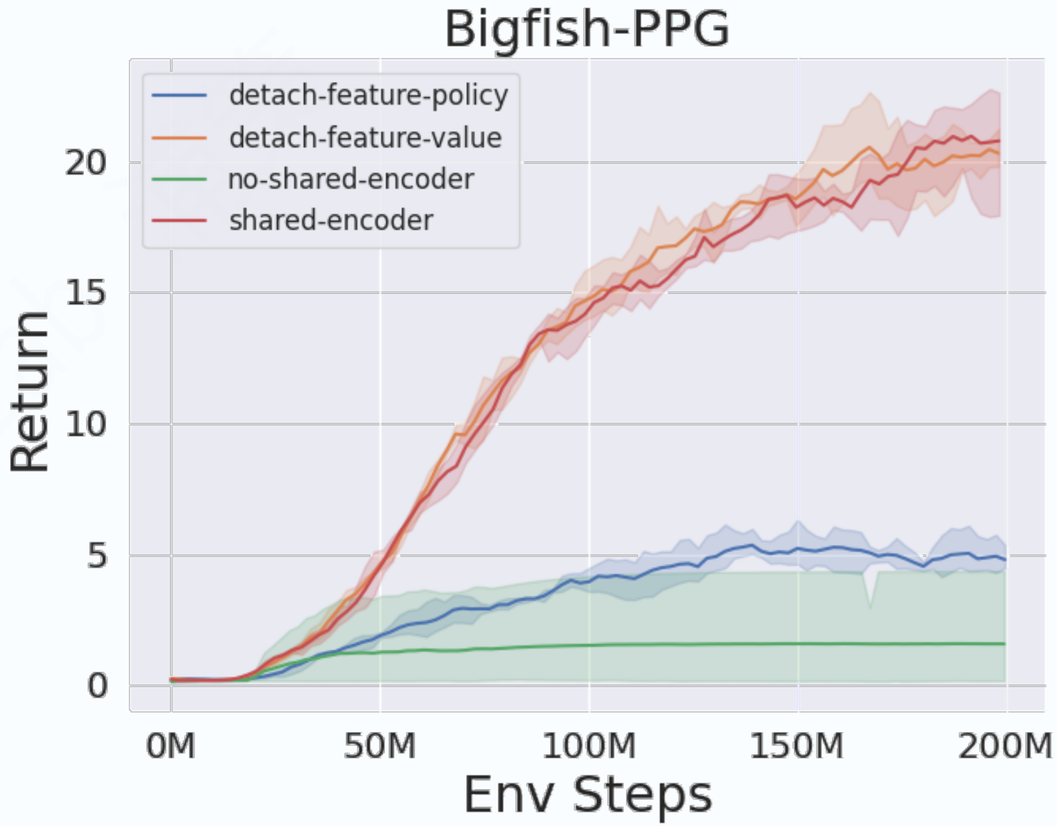


对于高维观察（比如图片）空间：

- Policy 和 Value 共享编码器是非常重要的
- 但 Policy 和 Value 对于编码器具体有什么样的影响关系，二者是缺一不可，还是各有不同的作用？

实践：PPO+大鱼吃小鱼(Procgen)





Procgen: https://openai.com/five/
DI-engine + Procgen Doc: https://di-engine-docs.readthedocs.io/zh_CN/latest/13_envs/procgen_zh.html
实验细节：https://github.com/opendilab/PPOxFamily/issues/8

代码：如何理解计算图中的梯度流动

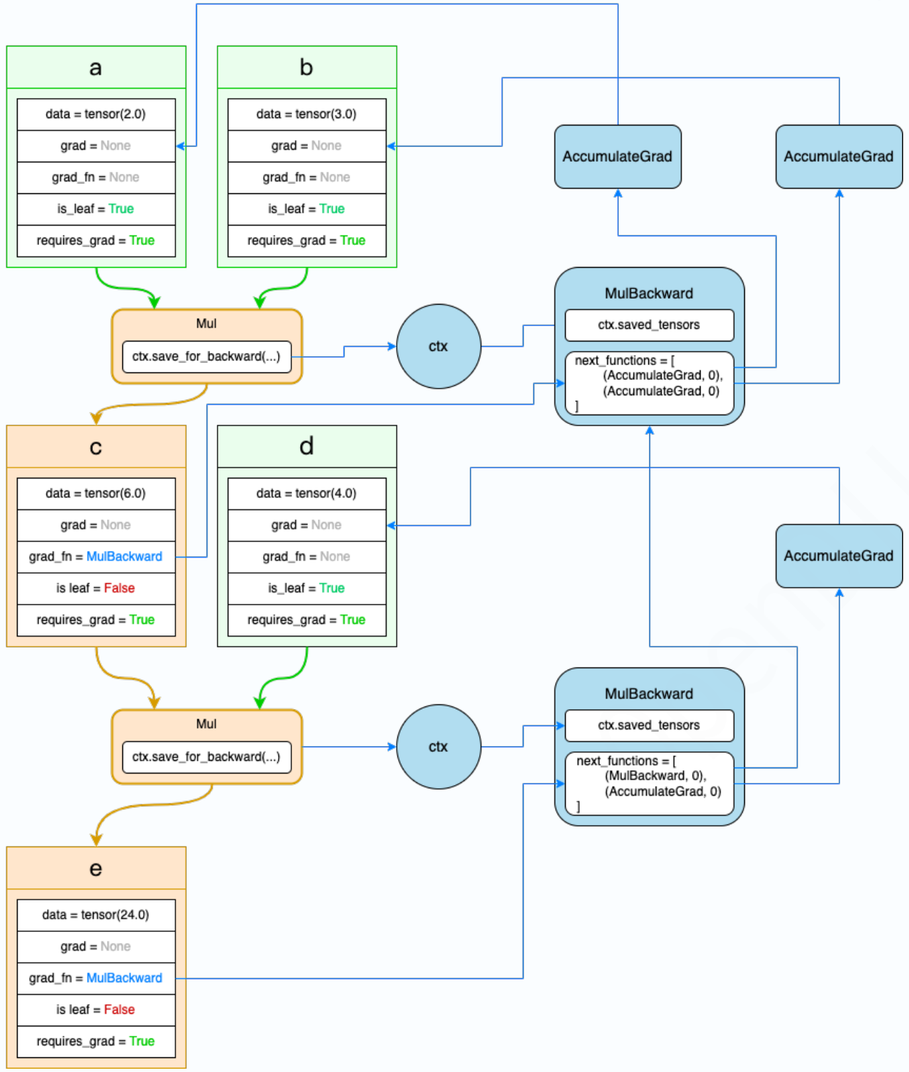
a =torch.tensor(2.0, requires_grad=True)

b= torch.tensor(3.0. requires_grad=True)

c=a* b

d= torch.tensor(4.0, requires_grad=True)

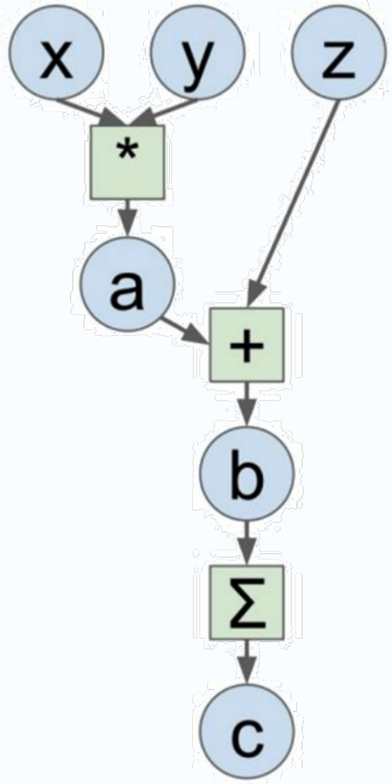
e=c* d




```python
import numpy as np
np.random.seed(0)


# generte data
B, D = 3, 4

x = np.random.randn(B, D)
y = np.random.randn(B, D)
z = np.random.randn(B, D)
# forward
a = x * y
b = a + z
c = np.sum(b)
# backward
grad_c = 1.0
grad_b = grad_c * np.ones((B, D))
grad_a = grad_b.copy()
grad_z = grad_b.copy()
grad_x = grad_a * y
grad_y = grad_a * x
```



Autograd: https://pytorch.org/tutorials/beginner/introyt/autogradyt_tutorial.html

完整代码示例：https://opendilab.github.io/PPOxFamily/

总结：PPO + 观察空间

小节	技术要点	代表决策任务
向量观察空间	• 统一量纲+特征编码设计 • 不变性与等变性概述	机器人控制 交通信号控制
图片观察空间	• 视觉信息预处理+运动信息建模 • 网络架构设计（LN）	视频游戏（超级马里奥） 棋类游戏（围棋）
复杂结构化观察空间	• 应用对应模态数据经典的网络架构 • 灵活利用 skip connection • 根据数据模态特性定制化设计	自动驾驶 三消类游戏（羊了个羊） 即时战略游戏（星际争霸2）
通用训练方法	• Actor和Critic是否共享编码器 • 表征学习 + 强化学习联合优化	/

下节预告

（四）探索之法：解密稀疏奖励空间

- 稀疏奖励空间的万能钥匙
- 多尺度奖励空间的锦囊妙计