混合动作空间表征学习方法介绍（HyAR）

近些年来，深度强化学习的研究者们将目光投向了**更通用**的混合动作空间建模方法，开始尝试设计额外的表征学习模块来获得**更紧致（compact）、高效的动作表示，**从而拓展强化学习在复杂动作空间上的应用。在本小节中，我们将会介绍相关工作之一：HyAR[1]。

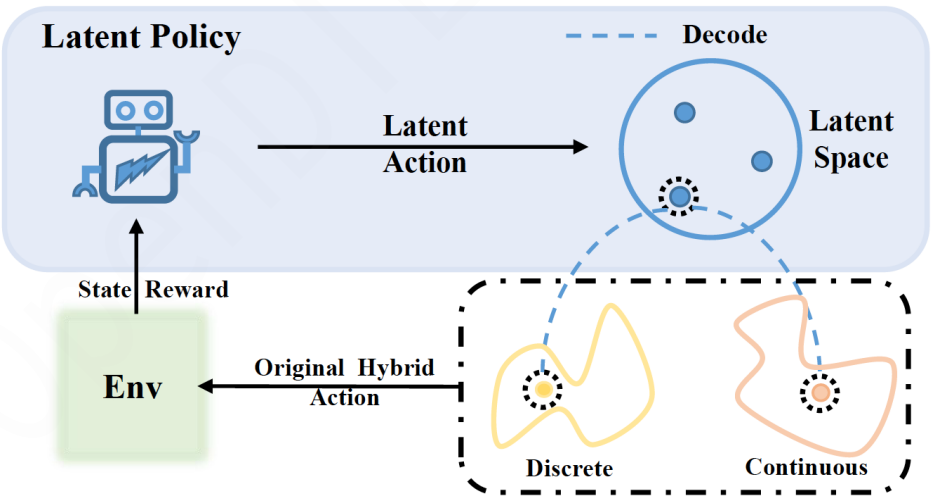
问题概述

- **研究背景：**

  针对决策问题输出成混合动作空间的问题（这里特指参数化动作空间，即离散动作类型和对应的连续参数），最直接的方法是通过简单离散化或连续化将原始空间转换为统一的同质动作空间，但显然这种做法忽略了混合动作空间的底层内在结构，存在严重的扩展性和训练稳定性问题，具体参考下例：

  - 例如，原始混合动作空间为 m 维离散动作和 n 维连续动作，如果对这 n 维连续动作的每一维离散为 K 个 bin，则变换后的总的离散动作空间为 $m + K^n$ 维。当 n 较大时该值会很大，如果用类似 DQN 算法来求解会导致 Q 函数学习负担比较大，不能很好的拟合每一个动作的 Q 值，从而不能扩展到高维空间上。

  为了更好地建模混合动作空间的内在结构并扩展到高维情形，HyAR 便应运而生，具体概览如下：



（图1：HyAR 算法概览图：将原始混合动作映射到一个隐式表征空间 (latent space)，RL 智能体在隐空间上训练学习一个 latent policy。而在与环境交互时，智能体选择的 latent action 会通过 decoder 解码回原始混合动作，然后执行收集数据与评估的后续流程。）

- **核心思路：**

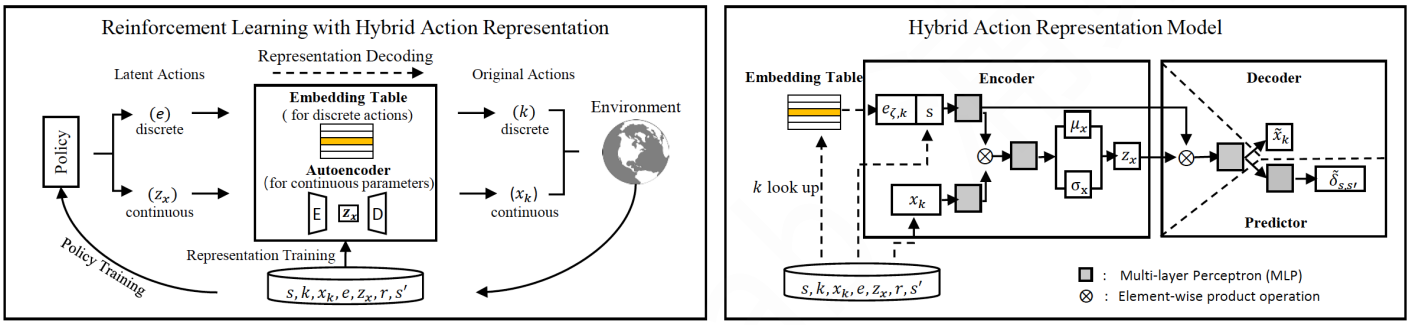
  - 提出了一种混合动作表征（Hybrid Action Representation，HyAR）方法，将原始混合动作空间转换为一个紧凑的、可编解码的隐式表征空间。

- 通过条件变分自编码器（conditional VAE）和可学习的嵌入表（embedding table），对原始动作的离散部分和连续部分之间的依赖性进行建模，构造出一个优质的隐式动作空间。
- 为了进一步提高模型的有效性，作者利用无监督的环境动力学预测（unsupervised environmental dynamics prediction）方法，使得训练得到的动作表征更适用于 RL 训练。
- 最后，应用常规强化学习算法在这个学习得到的动作表征空间上优化策略，通过动作表征的 decoder 将动作表征解码回原始动作空间执行与环境交互收集数据与评估的后续流程。

- **实验评估**：

  在 gym-hybrid 等混合动作空间 Benchmark 环境上对比了 HyAR 与其他基线混合动作空间算法，展现出更好的最终性能和收敛速度。另外，进一步的实验表明，在高维动作空间上，HyAR 的优势会更加明显。

核心算法设计



（图2: (左) 混合动作空间表征与强化学习结合的流程图。(右) 混合动作表征模型的结构：包括条件变分自编码器 (conditional VAE) 和可学习的嵌入表 (embedding table)）

📌 符号说明：

- s ： 当前状态
- s' ： 下一时刻状态
- r ： 奖励
- k ： 原始混合动作的离散部分 (或简称为离散动作)
- x_k ： 原始混合动作的连续部分 (或简称为连续动作)
- e ： 离散动作的隐式编码
- z_x ： 连续动作的隐式编码
- Embedding Table： $E_\zeta \in \mathbb{R}^{K \times d_1}$ ，即维度为 K x d_1 的矩阵表，每个离散动作 k 对应 d_1 维的编码，可训练参数记为 ζ 。
- $e_{\zeta,k}$ ： 离散动作 k 在 Embedding Table E_ζ 下的隐编码，即 E_ζ 的第 k 行向量: $e_{\zeta,k} = E_\zeta(k)$ 。

具体的算法设计需要解决两方面的问题：

- 建模混合动作中各部分之间的关系。
- latent policy 输出的 latent action 可以便利地能够解码回原始混合动作，以便与环境交互。

为此，如图2所示，HyAR提出了 dependence-aware 的混合动作编解码框架，关键点如下：

- Embedding Table对应离散动作的隐空间 $e \in \mathbb{R}^{d_1}$, 条件 VAE 对应连续动作的隐空间 $z_x \in \mathbb{R}^{d_2}$ ；
- Embedding Table和条件 VAE 联合构建了一个可解码的混合动作表征空间($\in \mathbb{R}^{d_1+d_2}$), 远比之前工作中的混合动作空间 $\mathbb{R}^{K+\sum_k |\mathcal{X}_k|}$ 小 (尤其是在 K 或 $\sum_k |\mathcal{X}_k|$ 特别大的时候)。
- 动作表征网络的编解码过程，数学表达式为：
  - 编码(Encooding): $e_{\zeta,k} = E_\zeta(k), z_x \sim q_\phi(\cdot \mid x_k, s, e_{\zeta,k}) \quad \text{for } s, k, x_k$
  - 解码(Decoding): $k = g_E(e) = \arg\min_{k' \in \mathcal{K}} \|e_{\zeta,k'} - e\|_2, \quad x_k = p_\psi(z_x, s, e_{\zeta,k}) \text{ for } s, e, z_x$

训练过程

- 连续动作的隐空间的构建依赖于离散动作，如图2右边所示, 原始混合动作的离散部分为 k ，连续部分为 x_k , 通过 Embedding Table 进行查找得到隐动作编码 $e_{\zeta,k}$ ，然后将离散动作编码 $e_{\zeta,k}$ ，状态 s ，连续动作 x_k ，一起输入到 conditional VAE 的 encoder 得到连续动作的编码 z_x （具体地，建模为高斯分布， encoder 输出均值 μ_x 和标准差 σ_x ，从中采样得到 z_x ）。
- 给定数据样本为 $(s, k, x_k, e, z_k, r, s')$, 从数据 buffer 中采样得到一个小批次 (minibatch) 的样本 $\{(s, k, x_k, e, z_k, r, s')\}$ 后，Embedding Table和conditional VAE一起联合训练， 训练的损失函数定义为:

$$L_{VAE}(\phi, \psi, \zeta) = \mathbb{E}_{s,k,x_k \sim \mathcal{D}, z \sim q_\phi}\left[\|x_k - \hat{x}_k\|_2^2 + D_{KL}\left(q_\phi(\cdot \mid x_k, s, e_{\zeta,k} \| \mathcal{N}(0, I))\right)\right]$$

- 其中第1项为 L_2 重建损失，第2项为隐表征变量 z 的变分后验和高斯先验之间的KL散度。

推理过程

- latent policy 模型分别输出离散和连续动作的隐编码 e 和 z_x (实际采用 TD3 policy, 都为连续向量), 通过动作表征的 decoding 过程得到环境能够执行的原始混合动作: k 和 x_k 。
- 具体来说， 如图2右边所示, 给定 e 和 z_x , 首先通过 Embedding Table 进行最近邻查找，得到最近邻的隐式动作编码 $e_{\zeta,k}$ 以及其在表中的索引 k ，然后将编码 $e_{\zeta,k}$ 与状态 s , 连续动作的编码 z_x ，一起输入到 conditional VAE的 encoder 和 decoder 得到连续动作的解码 $\hat{x}_k$ 。

环境动态正则化损失

只使用 VAE 的重建损失，会导致学习到的混合动作表征缺少类似“*这个动作会对环境产生什么样的影响*”这样的信息，这样的混合动作表征空间在学习 RL 策略和价值函数时可能是无效的，因为后面这些函数高度依赖于环境动态 (dynamics)的知识。为了充分利用环境动态，作者提出了一个**基于预测动态的无监督损失**来进一步细化了混合动作表征的学习。

直观上理解，能够预测动态的表征语义上也会更加平滑，也就是说，2个混合动作表征向量在表征空间中越接近，也就意味着他们对应的原始混合动作对环境的影响也越相似。因此，从原理上看，这种**正则化后的表征空间会更适合** RL 策略和值函数的近似和泛化。

具体来说，如图2右边所示，HyAR 采用了一个子网络接在条件VAE解码器后面，用于预测状态残差。其中状态残差定义为 $\delta_{s,s'} = s' - s$，其预测的状态残差为:

$$\tilde{\delta}_{s,s'} = p_\phi(z_x, s, e_{\zeta,k}) \text{ for } s, e, z_x$$

然后使用以下的 L2 损失作为正则化项：

$$L_{\text{Dyn}}(\phi, \psi, \zeta) = \mathbb{E}_{s,k,x_k,s'}\left[\left\|\tilde{\delta}_{s,s'} - \delta_{s,s'}\right\|_2^2\right]$$

综上，混合动作表征训练时总的损失函数为：

$$L_{\text{HyAR}}(\phi, \psi, \zeta) = L_{\text{VAE}}(\phi, \psi, \zeta) + \beta L_{\text{Dyn}}(\phi, \psi, \zeta)$$

混合动作表征学习与 RL 结合

论文中将 TD3 算法和上文提出的表征学习方法相结合，得到了具体的算法实例 HyAR-TD3 ，具体算法伪代码如图3所示：

Algorithm 1: HyAR-TD3

Initialize actor π_ω and critic networks $Q_{\theta_1}, Q_{\theta_2}$ with random parameters $\omega, \theta_1, \theta_2$
Initialize discrete action embedding table E_ζ and conditional VAE q_ϕ, p_ψ with random parameters ζ, ϕ, ψ
Prepare replay buffer $\mathcal{D}$
repeat Stage ❶
  Update ζ and ϕ, ψ using samples in $\mathcal{D}$ ▷ see Eq. 6
until *reaching maximum warm-up training times*;
repeat Stage ❷
  for $t \leftarrow 1$ *to* T **do**
    // select latent actions in representation space
    $e, z_x = \pi_\omega(s) + \epsilon_e$, with $\epsilon_e \sim \mathcal{N}(0, \sigma)$
    // decode into original hybrid actions
    $k = g_E(e), x_k = p_\psi(z_x, s, e_{\zeta,k})$ ▷ see Eq. 3
    Execute (k, x_k), observe r_t and new state s'
    Store $\{s, k, x_k, e, z_x, r, s'\}$ in $\mathcal{D}$
    Sample a mini-batch of N experience from $\mathcal{D}$
    Update $Q_{\theta_1}, Q_{\theta_2}$ ▷ see Eq. 7
    Update π_ω with policy gradient ▷ see Eq. 8
  repeat
    Update ζ and ϕ, ψ using samples in $\mathcal{D}$ ▷ see Eq. 6
  until *reaching maximum representation training times*;
until *reaching maximum total environment steps*;

（图3： HyAR-TD3算法伪代码）

总体上，算法分为2个阶段：

- warm-up 阶段:
  - 首先预收集数据放入 buffer $\mathcal{D}$ 中，(注意收集策略不限，可以结合人类的先验知识，这里通过使用随机策略与环境交互收集)，然后通过最小化混合动作表征总损失函数来学习 embedding table 和 conditional VAE，目的是为后续的联合训练阶段给出一个好的初始化参数。

- training 阶段（强化学习和动作表征学习联合训练阶段）：
  - 强化学习
    - 通过强化学习策略网络给出表征空间的隐动作；
    - 隐动作通过**动作表征网络的解码过程**得到原始的混合动作，与环境交互，将数据样本存入 buffer $\mathcal{D}$ 中；
    - 从 buffer $\mathcal{D}$ 中采样，根据强化学习损失函数训练强化学习的策略网络和值网络。
  - 动作表征学习
    - 从buffer $\mathcal{D}$ 中采样最新的样本，通过最小化混合动作表征总损失函数来更新 embedding table 和 conditional VAE。

实验部分：作者在 Platform，Goal，CatchPoint，HardMove 这4个混合动作空间环境上，对比了 PADDPG，PDQN，HHQN，HPPO 等算法，发现 HyAR-TD3 的最终性能和收敛速度都有大幅提升，具体实验分析可以参见原论文。

后记

在 HyAR 之外，还有很多混合动作空间表征学习方法的相关研究仍在不断探索中，最近也出现了一些基于专家数据预训练的方法[2]，期待能有更多有兴趣的研究者，共同探索这个方向。

参考文献

[1] https://openreview.net/pdf?id=64trBbOhdGU

[2] https://arxiv.org/abs/2110.10149