arXiv:2301.02099v1 [cs.RO] 5 Jan 2023

利用自监督奖励塑形离线学习目标条件策略

Lina Mezghani Meta AI，Inria*
linamezghani@fb.com

Sainbayar Sukhbaatar Meta AI

Piotr Bojanowski Meta AI

Alessandro Lazaric Meta AI

Karteek Alahari Inria*

摘要：
在机器人学中，如何从预先收集的数据集中学习，使智能体能够执行多种技能，是一个重要问题，因为与环境进行在线交互极其耗时。此外，为每一种期望的技能手动设计奖励函数也难以承受。先前研究 [1, 2] 通过事后重标记，从离线数据集中学习目标条件策略，无需手动指定奖励，以应对这些挑战。这些方法受到奖励稀疏性问题的困扰，在长时域任务中表现不佳。在本研究中，我们提出一种新颖的自监督学习阶段，利用预先收集的数据集理解模型的结构和动力学，并塑造稠密奖励函数，以便离线学习策略。我们在三项连续控制任务上评估了所提方法，结果表明，该模型显著优于现有方法 [1, 2], ，尤其是在涉及长期规划的任务中。

关键词：离线强化学习、自监督学习、目标条件强化学习

1 引言

尽管要实现通用自主智能体，就必须掌握大量多样化的技能，但若采用标准强化学习（RL）框架，逐项专注于每种技能来实现这一目标，代价将十分高昂。这主要是因为每种技能都需要人工设计奖励函数，并与环境进行交互。无监督强化学习开辟了一条途径，使智能体无需监督（即，无需手工设计奖励）便能学会执行多种能力，之后还能通过少样本或零样本泛化适配下游任务 [3, 4, 5, 6]。然而，使用这类方法学习策略并不适用于真实机器人，因为在线训练需要进行数百万次交互。

近期，出现了一系列研究，利用预先收集的轨迹数据集离线训练策略（即，无需与环境进行额外交互）[7, 8]。更具体地说，给定一个不含奖励的轨迹数据集，以及一个为解决特定任务而设计的奖励函数，智能体通过使用该奖励函数为数据集中的转移重新标注奖励，从而进行离线学习。这种设定对于机器人学尤其重要，因为数据收集极其耗时：在这一情境下，将数据收集与策略学习解耦，可以加快策略迭代。然而，这仍需要为每项任务单独设计一个奖励函数并学习一个策略。

因此，要扩展离线机器人学习，一个重要问题是找到从已收集的数据集中学习多任务策略的方法。近期研究 [1, 9, 10], 从目标条件视角着手解决这一问题：给定一个由先前收集的轨迹组成的数据集，目标是学习一个能够到达数据集中任意状态的目标导向智能体。此类表述有两方面优势：首先，它便于理解技能；其次，它不需要在

*格勒诺布尔-阿尔卑斯大学、Inria、法国国家科学研究中心、格勒诺布尔国立综合理工学院、LJK，法国格勒诺布尔 38000。项目页面：https://linamezghani.github.io/go-fresh

第六届机器人学习会议（CoRL 2022），新西兰奥克兰。

测试时进行任何适应。要使这一框架实现无监督学习，就需要摆脱手工设计的奖励，正如 Chebotar 等人提出的。 [1], 他们通过事后重标记，在离线环境中学习目标条件策略 [2]。然而，他们的方法容易陷入从稀疏奖励中学习的困境，在长时域任务中也可能效率低下。

在这项工作中，我们提出了一种自监督奖励塑形方法，使得能够构建带有密集奖励的离线数据集。为此，我们设计了一个自监督学习阶段，旨在训练策略之前学习环境的结构和动力学。在此阶段，我们：(i) 训练一个可达性网络 [11] ，以估计状态空间 $\mathcal{S}$中的局部距离；(ii) 提取一组能够覆盖 $\mathcal{S}$的代表性状态；最后，(iii) 在这组状态上构建一张图，以近似 $\mathcal{S}$ 中的全局距离。训练目标条件策略时，我们以两种方式使用该图：通过最短路径距离计算奖励，并生成沿着通往目标的路径上难度适中的转移。

我们在复杂的连续控制任务上评估了该方法，并将其与以往最先进的离线 [1, 2] 方法进行了比较。结果表明，我们基于图的奖励方法能够利用以往经验数据集中的转移，学得良好的目标条件策略，既不需要与环境进行额外交互，也不需要人工设计奖励。此外，我们还表明，与使用通过监督奖励训练的策略所收集数据集的先前工作 [1], 不同，我们的方法甚至能够从质量较差的数据集中学习目标条件策略，例如，包含由随机策略采样得到的轨迹。由此，我们的工作首次实现了在没有任何监督的情况下，从离线数据集中学习目标条件策略，因为在任何阶段——数据收集、策略训练和评估——都不需要任何手工设计的奖励函数。

2 相关工作

目标条件强化学习。 在最初的设定中，研究者采用了多种方法来解决目标条件强化学习问题 [12, 13, 2, 14]。这些工作中的策略学习过程受到监督：训练时，评估目标集合以及引导智能体到达目标的奖励函数都是已知的。多项研究提出了在训练目标条件策略时自动生成目标的方案，包括自我博弈 [15, 16, 17], 和对抗式学生-教师策略 [18]。近期的一系列研究 [19, 20, 21, 22, 23, 24, 25, 26] 着重于以无监督方式学习目标条件策略。其目标是在训练时不提供任何监督（奖励、目标到达函数），训练出能够在环境中到达任意目标状态的通用智能体。具体而言，Mendonca 等人 [25] 训练了一个基于模型的智能体，该智能体借助探索者模型发现新目标，并通过想象 rollout 使用达成者策略到达这些目标。

离线强化学习。 在研究如何从预先收集的数据集中训练策略时，数据收集技术是一个重要方面。在这一背景下，早期研究假设可以使用以任务特定奖励训练的策略 [27, 28]。近来，有研究提出利用无监督探索来收集用于离线强化学习的数据集 [7, 8]。具体而言，Yarats 等人 [7] 在 DeepMind 控制套件 [29] 上创建了一个预先收集的轨迹数据集 ExoRL，其中的轨迹均在没有任何人工设计奖励的情况下生成。与 URLB [30], 类似，ExoRL 对多种探索算法 [3, 6, 31, 5], 进行基准测试，并评估在相应的、以任务特定奖励重新标记的离线数据集上训练的策略的性能。

多任务离线强化学习。 近期研究提出了从预先收集的数据集中学习多个任务的方法，首先是通过自监督方式生成目标以改进离线数据收集流程的方法 [32] 。这种联系也在监督学习设定 [9, 33] 以及学习分层策略时 [10]得到了研究。在一个与我们工作密切相关的设定中，可操作模型 [1]研究了在不与环境交互且没有任务特定奖励的情况下，从离线数据集中学习目标条件策略的问题。该方法采用目标条件 Q 学习，并结合事后重标记 [2]。与依赖稀疏奖励进行学习的相关工作不同，我们提出利用自监督训练阶段来密集塑造奖励。

3 预备知识

令 $\mathcal{E} = (\mathcal{S},\ \mathcal{A},\ P,\ p_0,\ \gamma,\ T)$ 表示一个无奖励马尔可夫决策过程（MDP），其中 $\mathcal{S}$ 和 $\mathcal{A}$ 分别为状态空间和动作空间， P: $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathrm{IR}+$ 是状态转移概率

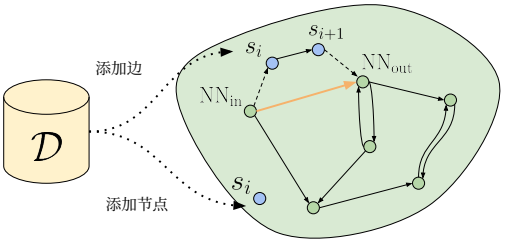


图1：图构建算法概览。给定一个转移 $(s_i,\ s_{i+1}) \in \mathcal{D}$，如果 s_i 与图中现有节点的距离足够远，我们就将其添加为节点。此外，我们在 s_i 的入向最近邻与 s_{i+1}的出向最近邻之间添加一条边。

函数，$p_0 : \mathcal{S} \to \mathrm{IR}+$ 是初始状态分布，γ 是折扣因子，T 是任务时域。在目标条件设置中，目标是学习一个策略 $\pi : \mathcal{S} \times \mathcal{G} \to \mathcal{A}$，使其最大化目标分布上的累积回报期望，其中 $\mathcal{G}$ 表示目标空间。这里，我们采用一个常见假设，即状态和目标以相同的形式定义，*ie.*，$\mathcal{G} \subset \mathcal{S}$。

我们假设可以访问数据集 $\mathcal{D}$，其中包含使用 $\mathcal{E}$中的任意数据收集算法生成的预收集回合。每个回合在 $\mathcal{D}$ 中存储为一系列 (s, a, s') 元组，其中 s、$s' \in \mathcal{S}$和 $a \in \mathcal{A}$。在 Yarats 等人 [7], 提出的一般离线形式中，数据集 $\mathcal{D}$ 可以通过在 $\mathcal{D}$中的每个元组处计算任意奖励函数 $r : \mathcal{S} \times \mathcal{A} \to \mathrm{IR}$ 来重新标记，并将由此得到的元组 $(s, a, r(s, a)\ s')$ 添加到重标记数据集 $\mathcal{D}_r$中。我们可以通过考虑目标空间中的目标分布 $p_{\mathcal{G}}$ 以及任意目标条件奖励函数 $r : \mathcal{S} \times \mathcal{A} \times \mathcal{G} \to \mathrm{IR}$，将这一流程扩展到目标条件设置。给定 $\mathcal{D}$中的一个元组 (s, a, s')，我们通过采样一个目标$g \sim p_{\mathcal{G}}$来重新标记它，计算 $r(s, a, g)$，并将由此得到的元组 $(s, a, g, r(s, a, g)\ s')$ 添加到重标记数据集$\mathcal{D}_r$中$_{p_{\mathcal{G}}}$。

生成重标记数据集$\mathcal{D}_{r,\ p_{\mathcal{G}}}$ 后，我们可以通过执行任意离线强化学习算法来学习目标条件策略。该算法完全离线运行，通过从$\mathcal{D}_{r,\ p_{\mathcal{G}}}$ 中采样元组，且不与环境进行任何交互。随后，在 $\mathcal{E}$ 中在线评估目标条件策略，评估使用一组固定的评估目标，这些目标在训练期间未知。

4 自监督奖励塑形

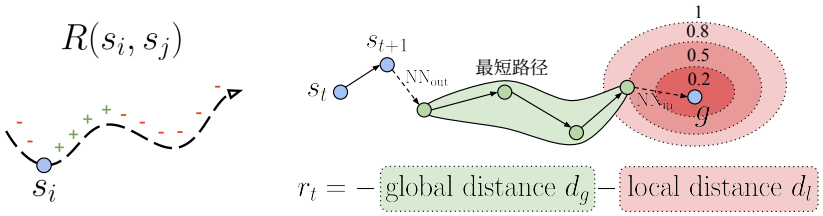
现在，我们介绍自监督奖励塑形方法。该方法包含三个阶段，下面将逐一详述。第一阶段，我们在 $\mathcal{D}$中的轨迹上训练可达性网络（RNet） [11]，以预测两个状态是否彼此可达。第二阶段是构建有向图 $\mathcal{M}$，其节点是 $\mathcal{D}$中一部分状态，边则连接可达状态。我们使用 RNet 作为判据，避免向 $\mathcal{M}$ 中添加相似状态，从而使其节点均匀覆盖 $\mathcal{D}$ 中的状态。最后一个阶段是在从 $\mathcal{D}$采样的转移和目标上训练目标条件策略。训练时使用密集奖励，该奖励由全局距离项（基于 $\mathcal{M}$中的图距离）和局部距离项（基于 RNet）相加得到。我们方法的重要之处在于，整个训练过程仅使用预先收集的数据集 $\mathcal{D}$ 中的轨迹，不会在环境中执行任何动作。下面我们将更详细地介绍各个组成部分。

4.1 可达性网络

为了学习 $\mathcal{D}$ 中状态之间良好的局部距离，我们采用了可达性网络（RNet） [11] 的非对称版本。RNet 的基本思想是通过随机策略从一个状态到达另一个状态所需的平均步骤数，来近似环境中状态之间的距离。我们对原始表述做了两处修改：首先，我们使用来自 $\mathcal{D}$ 的探索轨迹，而不是随机轨迹；其次，我们利用时间方向，因为一个状态可以从另一个状态到达，但反之未必成立。设 $(s_1^a, \ldots,\ s_T^a)$ 表示 $\mathcal{D}$ 中的一条轨迹，其中 a 是轨迹索引。我们为每一对观测 (s_i^a, s_j^b) 定义一个 可达性标签y_{ij}^{ab}，其定义如下

$$y_{ij}^{ab} = \begin{cases} 1 & \text{if } a = b \text{ and } 0 \le j - i \le \tau_{\text{reach}}, \\ 0 & \text{otherwise}, \end{cases} \qquad \text{for } 1 \le i, j \le T, \tag{1}$$

其中，可达性阈值 τ_{reach} 是一个超参数。仅当状态位于同一条轨迹中，且从 s_i^a到 s_j^b 的步骤数小于 τ_{reach}时，可达性标签才等于 1，具体如图所示



(a) RNet 的训练标签　　(b) 奖励计算的可视化

图 2：我们提出的密集奖励塑形方法的可视化。(a) 展示了如何生成用于训练 RNet 的训练标签：给定一个状态 s_i，在阈值 τ_{reach} 范围内，从同一条轨迹中采样正样本对，其余轨迹部分则构成负样本对。(b) 展示了奖励如何由全局距离项（绿色）和局部距离项（红色）组合而成：全局距离项通过计算图中状态 s_{t+1} 的出向最近邻（NN_{out}）与目标的入向最近邻（NN_{in}）之间的最短路径得到；局部距离项则利用 RNet 计算 NN_{in} 与 g 之间的值。

见 图 2a。注意，$y_{ij}^{ab} \neq y_{ji}^{ab}$。我们训练了一个孪生神经网络 R，即 RNet，用于预测可达性标签 y_{ij}^{ab}，该标签由一对观测 $(s_i^a,\ s_j^b)$ 在 $\mathcal{D}$中确定。RNet 由一个嵌入网络 g和一个用于比较嵌入的全连接网络 f组成，即，例如，

$$R(s_i^a, s_j^b) = \sigma\left[f(g(s_i^a), g(s_j^b))\right], \tag{2}$$

其中 σ 是一个 Sigmoid函数。较高的 R 值表示两个状态可以通过随机游走轻松到达，因此可视为在环境中彼此接近。更准确地说，R 的取值范围为 (0，1)，且当 $R(s,\ s') \geq 0.5$ 时，s' 可从 s 到达。RNet 以自监督方式进行学习，因为训练网络所需的真实标签是自动生成的。

4.2 有向图

在下一阶段，我们使用 $\mathcal{D}$ 中的轨迹构建一个有向图 $\mathcal{M}$，以捕捉环境的高层动态，如图1所示。我们希望 $\mathcal{M}$ 中的节点能够均匀地表示 $\mathcal{D}$中的状态。为此，我们对 $\mathcal{D}$中的状态进行筛选：只有当某个状态与 $\mathcal{M}$ 中的所有其他节点都相距足够远时，才会将其添加到 $\mathcal{M}$中。更准确地说，仅当 $s \in \mathcal{D}$ 时，才会将状态 $\mathcal{M}$ 添加到

$$R(s, n) < 0.5 \text{ and } R(n, s) < 0.5, \quad \text{for all } n \in \mathcal{M}. \tag{3}$$

请注意，我们要求两个方向都具有新颖性。这种筛选通过防止将相似状态添加到记忆中来避免冗余。它还具有平衡效果，因为即使智能体在 $\mathcal{D}$中多次访问某个区域，也会限制可从该区域添加的状态数量。

选定节点后，我们将可从一个节点到达另一个节点的节点对连接起来。为此，我们使用 $\mathcal{D}$ 中的轨迹，因为其中包含实际可行的转移。给定 $\mathcal{D}$中的一个转移 $s_i \to s_j$，若可从节点 n_i 到达 s_i，且可从s_j到达节点 n_j，则添加边 $n_i \to n_j$。这样，我们就得到一条链 $n_i \to s_i \to s_j \to n_j$，并可假定 n_i可从 n_j 到达。具体而言，我们选择节点 n_i 作为指向 s_i的入向最近邻（NN_{in}），并选择节点 n_j 作为从 s_j出发的出向最近邻（NN_{out}），即也，

$$n_i = \text{NN}_{\text{in}}(s_i) = \operatorname*{argmax}_{n \in \mathcal{M}} R(n, s_i), \quad n_j = \text{NN}_{\text{out}}(s_j) = \operatorname*{argmax}_{n \in \mathcal{M}} R(s_j, n). \tag{4}$$

对 $\mathcal{D}$ 中的所有转移执行此操作后，我们便将 $\mathcal{M}$ 转换为有向图，其中的边表示从一个节点到另一个节点的可达性。

4.3 用于策略训练的距离函数

随后，我们使用得到的有向图来计算状态空间中的全局距离。实际上，RNet 会预测 s_i 与 s_j 之间的可达性，因此我们可以直接将其用作距离度量

$$d_l(s_i, s_j) = 1 - R(s_i, s_j), \quad \forall s_i, s_j \in \mathcal{S}. \tag{5}$$

然而，这种可达性度量受限于某个阈值，因此无法保证 RNet 的预测具有良好的全局性质。

相比之下，有向图 $\mathcal{M}$ 捕捉了环境的高层全局动力学。只要图是连通的，我们就可以通过计算该图中最短路径的长度，轻松推导出任意一对节点之间的距离函数 $d_{\mathcal{M}}(n_i, n_j)$。在实践中，如有必要，我们可以采用一种技巧将图连通起来：在来自不同连通分量且 RNet 值最大的那对节点之间添加一条边。此外，我们还可以将该距离 $d_{\mathcal{M}}$扩展为状态空间 $\mathcal{S}$ 中的全局距离函数 d_g：对于 $\mathcal{S}$ 中的任意一对 s_i 和 s_j，找到它们在相应方向上的最近邻。更确切地说，

$$d_g(s_i, s_j) = d_{\mathcal{M}}(\text{NN}_{\text{out}}(s_i), \text{NN}_{\text{in}}(s_j)), \quad \forall s_i, s_j \in \mathcal{S}. \tag{6}$$

状态空间中两个状态之间的距离 d_g 等于它们各自在图中最近节点之间最短路径的长度。此过程如 图 2b所示，将 RNet 良好的局部特性传播开来，从而为距离较远的状态构造出形状良好的距离函数。由于 d_g 捕捉全局距离，而 d_l 捕捉局部细粒度距离，因此我们将二者组合用作最终的距离函数： $\forall s_i, \ s_j \in \mathcal{S}, \quad d(s_i, \ s_j) = d_g(s_i, \ s_j) + d_l(s_i, \ s_j)$。

4.4 策略训练

我们的方法的最后一个阶段是离线训练目标条件策略。在此，我们创建一个离线回放缓冲区 $\mathcal{B}$，其中填入了重新标记的数据。我们从 $\mathcal{D}$ 中随机采样一个转移 $(s_t, \ a_t, \ s_{t+1})$，并采样一个目标 g，然后用奖励 $r_t = -d(s_{t+1}, \ g)$ 对该转移重新标记。接着，我们将重新标记后的转移 $(s_t, \ a_t, \ g, \ r_t, \ s_{t+1})$ 推入 $\mathcal{B}$。为了创建一种课程，以人为方式引导智能体朝目标前进，我们尝试了两种不同的转移增强技术：

子目标增强。设 $(s_t, a_t, g, r_t, s_{t+1})$ 表示一个重新标记后的转移，$(n_0, ..., n_{P-1})$ 表示图 $\mathcal{M}$ 中 $n_0 = \text{NN}_{\text{out}}(s_t)$ 与 $n_{P-1} = \text{NN}_{\text{in}}(g)$ 之间的最短路径。该数据增强技术包括：对所有 $i \in \{0, P-1\}$，将每个转移 $(s_t, a_t, n_i, r_t^i, s_{t+1})$ 添加到回放缓冲区，其中 $r_t^i = -d(s_{t+1}, n_i)$。换言之，给定一个转移 (s_t, a_t, s_{t+1}) 和来自 $\mathcal{D}$ 的目标 g，我们会将一组重新标记后的转移推入回放缓冲区，这些转移的目标涵盖从 s_t 到 g 的最短路径上的所有目标（以及相应的奖励）。

边增强。与子目标增强技术类似，我们考虑一个重新标记的转移 $(s_t, a_t, g, r_t, s_{t+1})$ 及其对应的最短路径 $(n_0, ..., n_{P-1})$。这次，对于每个增强后的转移，我们都保留相同的目标 g，但对于每条边 (n_{i-1}, n_i) $i \in \{1, P-1\}$，我们将重新标记的转移 $(s_t^i, a_t^i, g, r_t^i, s_{t+1}^i)$ 添加到 $\mathcal{B}$，其中 $(s_t^i, a_t^i, s_{t+1}^i) \in \mathcal{D}$，$\text{NN}_{\text{out}}(s_t^i) = n_{i-1}$，$\text{NN}_{\text{in}}(s_{t+1}^i) = n_i$，且 $r_t^i = -d(s_t^i, g)$。请注意，构造方式保证了 $\mathcal{D}$ 中存在这样的转移：当且仅当 $\mathcal{D}$ 中存在一个转移，其对应的最近邻节点按相同顺序分别为这两个节点时，才会在图中添加一条从一个节点指向另一个节点的边。

回放缓冲区 $\mathcal{B}$ 填满后，就可以使用任意离策略算法训练目标条件策略。在我们的实现中，我们选择了软演员-评论家 [34], ，因为众所周知，它所需的超参数调优较少，而且在相关文献中应用广泛。

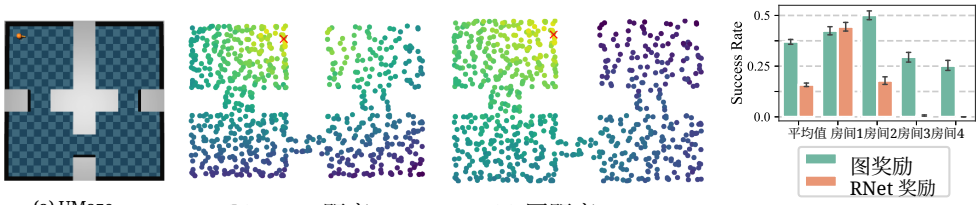
5 实验

5.1 环境与数据收集

我们在三个具有基于状态的输入的连续控制任务上进行实验。

UMaze [35]。第一个环境如图3a所示，是一个二维 U 形迷宫，具有连续动作空间和固定的初始位置。我们通过在迷宫中部署起始位置随机的随机策略，为该环境生成训练数据。我们收集了 10k 条长度为 1k 的轨迹。我们通过为目标条件智能体随机采样一个环境中的目标，并计算最终欧几里得距离来评估该智能体。

RoboYogaWalker [25]。 该项具有挑战性的 RoboYoga 基准由 Mendonca等人 [25], 提出，基于 DeepMind 控制套件中的 Walker 领域 [29], ，包含 12 个目标



(a) UMaze (b) RNet 距离 (c) 图距离 (d) RNet 与图奖励

图 3：(a) UMaze 环境；使用 RNet (b) 和图距离 (c) 计算得到的奖励热力图；以及 (d) 在 UMaze 上使用 RNet 和基于图的奖励训练的目标条件策略的性能。在 (b) 和 (c) 中，高奖励以黄色显示，低奖励以黑色显示。

这些目标对应于受瑜伽启发的身体姿势（例如，躺下、抬起一条腿或保持平衡）。我们考虑任务的基于状态的版本，并使用 Yarats 等人提出的任务无关数据集 [7]，该数据集由无监督探索策略生成，包含 10k 条长度为 1k 的轨迹，这些轨迹是在 Walker 领域中运行“proto” [5] 算法获得的。评估策略的成功度量指标根据回合结束时人形机器人的姿势进行评定。

Pusher [20]。 我们还将我们的方法应用于*Pusher*，这是一个逼真的机器人环境，如图 7 （左）所示，其中机械臂（红色）需要将冰球（蓝色）推到桌面上的指定位置。为了构建离线数据集，我们生成了 1 万条长度为 200 的随机轨迹。与先前研究类似 [20, 22, 26]，，我们在状态空间中随机生成了 500 个目标，并以冰球与其目标位置之间最终的欧几里得距离来衡量性能。

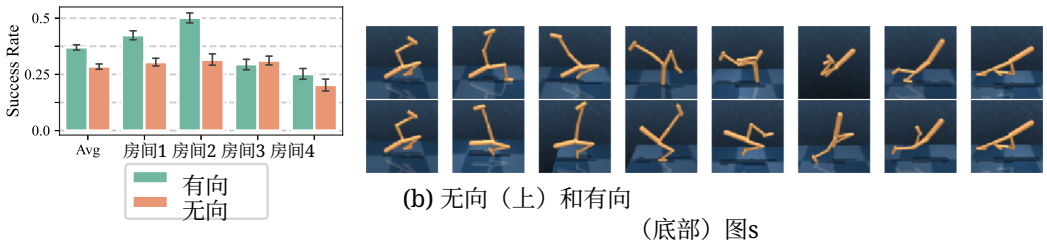
5.2 消融与设计选择

我们首先展示图结构对于长期规划是必不可少的。接着，我们说明图的直接性对于具有非对称行为的任务为何重要。最后，我们展示在为目标条件策略标注数据时，转移增强技术所产生的影响。

基于图的奖励的必要性。 我们的方法的一个重要组成部分是构建图 $\mathcal{M}$，以便计算具有良好全局属性的距离。为实证验证这一假设，我们比较了使用 RNet 奖励训练的目标条件策略（即，使用方程（5）中的距离 d_l ）与使用两个距离项作为奖励训练的策略。我们在 UMaze 环境中进行了这项实验，结果见 图 3d。我们注意到，使用图奖励训练的模型整体上优于使用 RNet 奖励训练的模型，尤其是在目标较远时（即房间 3 和 4）。我们还发现，对于靠近初始位置的目标，使用 RNet 奖励训练的模型略胜一筹。这凸显了 RNet 在估计局部距离方面的优势。 图 3b & 3c中的定性可视化证实了这一观察结果，因为图中显示第一和第四个房间内的状态之间具有较低的值。

图的有向性的重要性。随后，我们探究 RNet 的非对称性和图的有向性的重要性。为此，我们实现了所提方法的无向版本，其中 RNet 是对称的，图是无向的。所提方法的其他所有组成部分均保持不变。首先，我们比较两种变体在 UMaze 任务中的表现，如图 4a所示，并注意到，我们的方法采用非对称 RNet 和有向图，显著提升了目标条件策略的表现（+11% 成功率），尤其是对靠近初始位置的目标而言，即，也就是房间 1 和 2 中的目标。随后，我们分析了 RoboYoga 任务中无向图与有向图的最短路径定性可视化结果，如图 4b所示。在无向情况下，人形机器人仿佛违背了重力定律，会向后翻倒，头朝下站立，这可能极为困难，甚至无法实现。在有向情况下，最短路径会促使智能体先重新站稳，然后再向前倾。在这个例子中，重力使环境的动力学呈现非对称性且无法完全逆转，这也证明了我们的方法所述的有向形式是合理的。

转移采样策略。 作为最后一项消融研究，我们考察 第 4.4 节中所述转移数据增强技术的效用。我们评估了方法的四种可能变体：



(a) UMaze 任务上的比较

(b) 无向（上）和有向（底部）图s

Fig图 4：图的有向性对 (a) UMaze 任务和 (b) RoboYoga Walker 任务的影响 .

(i) 不使用任何数据增强，(ii) 仅使用边增强，(iii) 仅使用子目标增强，以及 (iv) 同时使用两种增强。我们在 RoboYoga 任务上执行了这项实验，结果如 图 6b所示。我们发现，两种数据增强技术都能提升目标条件智能体的性能，其中子目标增强带来的提升更大。此外，我们注意到，结合两种增强技术能进一步提升性能。在其余实验中，我们均采用这两种数据增强技术。

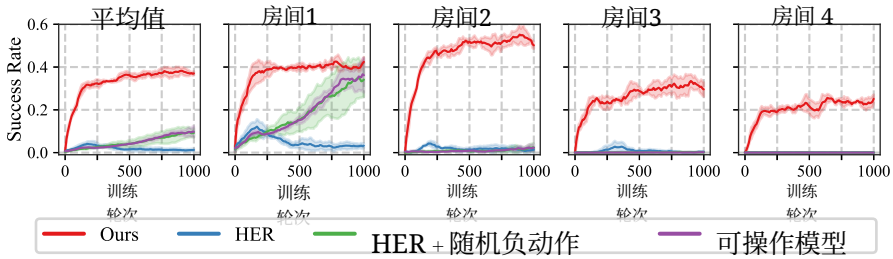


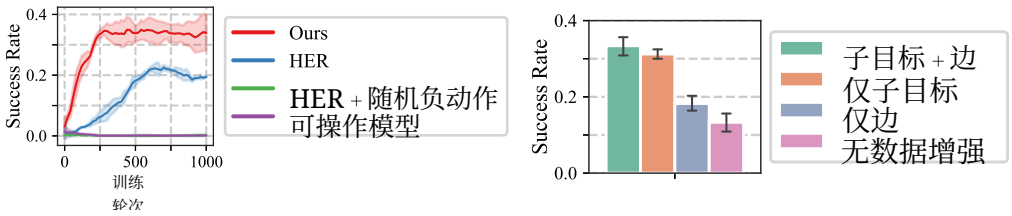
图 5：UMaze 任务的表现。我们展示了在四个房间中分别随机采样目标时的成功率，以及所有房间的平均值。

5.3 与先前工作的比较

基线方法。 我们将我们的方法与先前的无监督目标条件策略学习工作进行比较。为进行公平比较，我们采用与我们的方法相同的学习框架实现基线方法，仅改变奖励重标记过程。我们与以下基线方法进行比较：

- **后见经验回放 [HER] [2]**这是对标准无监督强化学习技术的重新实现，并将其适配到离线设置。更具体地说，我们使用稀疏奖励对来自 $\mathcal{D}$ 的子轨迹进行重标记：只有子轨迹的最后一次转移获得 1，其余部分均为 0。遵循 Chebotar 等人 [1]，的做法，我们还为子轨迹标注了从 $\mathcal{D}$ 中随机采样的目标，并赋予零奖励。
- **HER [2] 带随机负动作** 是 HER 的一种变体：对于 $\mathcal{D}$中的某个转移，我们在动作空间中均匀随机采样一个动作，并为其标记零奖励。这有助于克服 Chebotar 等人 [1] 提到的对未见动作的 Q 值过高估计问题。
- **可操作模型 [1]** 该方法基于带事后重标记的目标条件 Q 学习。我们重新实现了目标重标记过程：该过程使用 $\mathcal{D}$ 中子轨迹最终状态处的 Q 值来实现目标链式连接，同时也采用了负动作采样技巧。

在 UMaze 上的比较。我们在 UMaze 任务上将我们的方法与基线方法进行比较，并在图 5中展示结果。我们观察到，我们的模型总体上优于所有基线方法，并且在远离初始位置、具有挑战性的目标上表现出更大的提升。有趣的是，我们注意到，可操作模型只能到达第一个房间内的目标。这印证了这样一种直觉：稀疏奖励会让策略难以学习长时域任务。



(a) 与基线方法的比较　　(b) 转移增强的影响

图 6：RoboYoga Walker 任务上的表现

在 RoboYoga Walker 上的比较。在第二项实验中，我们在 RoboYoga 任务上将我们的方法与基线方法进行比较，如图 6a所示。同样，我们的方法优于以往工作，而可操作模型相较于 HER 并未取得显著提升。按目标细分的结果见补充材料。总体而言，这些结果表明，我们的稠密奖励塑形方法能够实现更快速、更稳健的离线目标条件策略训练。

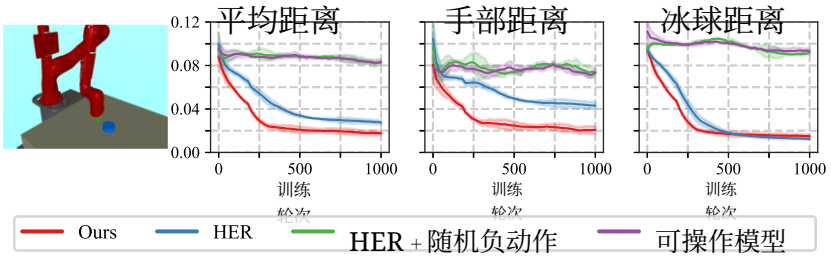


图 7：Pusher 任务上的表现（越低越好）。我们报告模型和所有基线方法在目标处的最终平均距离、手部距离和冰球距离。

Pusher 上的比较。 作为最后一项实验，我们在一个逼真的机器人环境中将我们的方法与先前工作进行了比较，如 图 7 所示。我们通过在状态空间中随机采样一个目标来评估离线训练的策略，并测量三个不同的指标：(i) 手部距离，即机械臂末端与目标之间的最终距离；(ii) 冰球距离，即冰球最终位置与目标之间的距离；以及 (iii) 平均距离，即前两个指标的平均值。我们的方法在此任务上优于基线方法，并且我们的目标条件智能体能够依次将冰球放置在目标位置，然后将手部放置在其目标位置。相比之下，**HER** [2] 能够以与我们的方法相近的表现将冰球放置在目标位置，但手部位置的精度不足。

6 结论：总结与局限性

我们提出了一种方法，能够以离线、无监督的方式从预先生成的数据集中学习多任务策略，即，无需与环境进行任何额外交互，也不需要人工设计奖励。我们的方法利用一个自监督阶段，旨在从离线数据集中学习环境的动力学，并据此构造稠密奖励函数。与基于事后重标记的先前研究相比，该方法取得了显著改进，尤其是在长时域任务中；在这类任务中，密集奖励对于学习优质策略至关重要。

我们的方法的主要局限在于依赖预先收集的轨迹数据集，而且该数据集必须充分覆盖状态空间，才能有效学习策略。尽管这类数据可能已经存在，例如 RoboYoga Walker 任务中的数据；或者也可以像我们在 UMaze 和 Pusher 任务中所做的那样，使用随机策略收集离线数据集，但在其他环境中，这一步可能颇具挑战。另一项局限是，我们仅在模拟环境中评估了该方法，没有在真实机器人上进行任何实验；而对于真实机器人，可能已有包含专家示范的预先收集数据集 [1]。

致谢

Karteek Alahari 获得了 ANR资助 AVENUE（ANR-18-CE23-0011）的部分支持。

参考文献

[1] Y. Chebotar, K. Hausman, Y. Lu, T. Xiao, D. Kalashnikov, J. Varley, A. Irpan, B. Eysenbach, R. C. Julian, C. Finn 等。可操作模型：机器人技能的无监督离线强化学习。载于国际机器学习会议，第 1518–1528 页。PMLR，2021年。[2] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. To- bin, O. Pieter Abbeel 和 W. Zaremba。后见经验回放。神经信息处理系统进展，第 30 卷，2017年。[3] D. Pathak, P. Agrawal, A. A. Efros 和 T. Darrell。通过自监督预测实现好奇心驱动探索。载于国际机器学习会议，第 2778–2787 页。PMLR，2017年。[4] Y. Burda, H. Edwards, A. Storkey 和 O. Klimov。随机网络蒸馏探索。载于国际学习表征会议，2018年。[5] D. Yarats, R. Fergus, A. Lazaric 和 L. Pinto。基于原型表征的强化学习。载于国际机器学习会议，第 11920–11931 页。PMLR，2021年。[6] B. Eysenbach, A. Gupta, J. Ibarz 和 S. Levine。多样性就是你所需的一切：无需奖励函数即可学习技能。载于国际学习表征会议，2018年。[7] D. Yarats, D. Brandfonbrener, H. Liu, M. Laskin, P. Abbeel, A. Lazaric 和 L. Pinto。不要改变算法，改变数据：用于离线强化学习的探索性数据。载于*ICLR 2022*物理世界可泛化策略学习研讨会，2022年。[8] N. Lambert, M. Wulfmeier, W. Whitney, A. Byravan, M. Bloesch, V. Dasagi, T. Hertweck 和 M. Riedmiller。离线强化学习探索面临的挑战。*arXiv* 预印本 *arXiv:2201.11861*，2022年。[9] R. Yang, Y. Lu, W. Li, H. Sun, M. Fang, Y. Du, X. Li, L. Han 和 C. Zhang。重新思考目标条件监督学习及其与离线强化学习的联系。*arXiv* 预印本 *arXiv:2202.04478*，2022年。

[10] J. Li、C. Tang、M. Tomizuka 和 W. Zhan。通过目标条件离线强化学习进行分层规划。*arXiv* 预印本 *arXiv:2205.11790*，2022年。[11] N. Savinov、A. Raichuk、D. Vincent、R. Marinier、M. Pollefeys、T. Lillicrap 和 S. Gelly。通过可达性实现情节好奇心。载于国际学习表征会议，2018年。[12] L. P. Kaelbling。学习实现目标。载于国际人工智能联合会议，第2卷，第1094–8页。Citeseer，1993年。[13] T. Schaul、D. Horgan、K. Gregor 和 D. Silver。通用价值函数逼近器。载于国际机器学习会议，第1312–1320页。PMLR，2015年。[14] S. Nasiriany、V. Pong、S. Lin 和 S. Levine。使用目标条件策略进行规划。神经信息处理系统进展，第32卷，2019年。

[15] S. Sukhbaatar、Z. Lin、I. Kostrikov、G. Synnaeve、A. Szlam 和 R. Fergus。《通过非对称自我博弈实现内在动机与自动课程》。发表于国际学习表征会议，2018年。

[16] S. Sukhbaatar、E. Denton、A. Szlam 和 R. Fergus。通过自我博弈学习目标嵌入，用于分层强化学习。*arXiv* 预印本 *arXiv:1811.09083*，2018年。

[17] O. OpenAI、M. Plappert、R. Sampedro、T. Xu、I. Akkaya、V. Kosaraju、P. Welinder、R. D'Sa、A. Petron、H. P. d. O. Pinto 等。用于机器人操作中自动发现目标的非对称自博弈。arXiv 预印本 arXiv:2101.04882，2021。[18] A. Campero、R. Raileanu、H. Kuttler、J. B. Tenenbaum、T. Rockt¨aschel 和 E. Grefenstette。通过 AMIGO 学习：对抗性驱动的内在目标。发表于国际学习表征会议，2020年。[19] D. Warde-Farley、T. Van de Wiele、T. Kulkarni、C. Ionescu、S. Hansen 和 V. Mnih。通过非参数判别奖励实现无监督控制。发表于国际学习表征会议，2018年。[20] A. V. Nair、V. Pong、M. Dalal、S. Bahl、S. Lin 和 S. Levine。借助想象目标进行视觉强化学习。神经信息处理系统进展，31：9191–9200，2018年。[21] A. Ecoffet、J. Huizinga、J. Lehman、K. O. Stanley 和 J. Clune。Go-explore：解决高难度探索问题的新方法。arXiv 预印本 arXiv:1901.10995，2019年。[22] V. Pong、M. Dalal、S. Lin、A. Nair、S. Bahl 和 S. Levine。Skew-fit：覆盖状态的自监督强化学习。发表于国际机器学习会议，第7783–7792页。PMLR，2020年。[23] S. Venkattaramanujam、E. Crawford、T. Doan 和 D. Precup。为目标条件强化学习进行距离函数的自监督学习。arXiv 预印本 arXiv:1907.02998，2019年。[24] K. Hartikainen、X. Geng、T. Haarnoja 和 S. Levine。用于半监督和无监督技能发现的动力学距离学习。发表于国际学习表征会议，2019年。[25] R. Mendonca、O. Rybkin、K. Daniilidis、D. Hafner 和 D. Pathak。通过世界模型发现并实现目标。神经信息处理系统进展，34，2021年。[26] L. Mezghani、P. Bojanowski、K. Alahari 和 S. Sukhbaatar。沿着随机游走前进：无需监督地学习发现并到达目标。发表于 ICLR 开放式学习智能体研讨会，2022年。[27] J. Fu、A. Kumar、O. Nachum、G. Tucker 和 S. Levine。D4rl：深度数据驱动强化学习数据集。arXiv 预印本 arXiv:2004.07219，2020年。[28] C. Gulcehre、Z. Wang、A. Novikov、T. Paine、S. G´omez、K. Zolna、R. Agarwal、J. S. Merel、D. J. Mankowitz、C. Paduraru 等。RL Unplugged：离线强化学习基准套件。神经信息处理系统进展，33：7248–7259，2020年。[29] Y. Tassa、Y. Doron、A. Muldal、T. Erez、Y. Li、D. d. L. Casas、D. Budden、A. Abdolmaleki、J. Merel、A. Lefrancq 等。DeepMind 控制套件。arXiv 预印本 arXiv:1801.00690，2018年。[30] M. Laskin、D. Yarats、H. Liu、K. Lee、A. Zhan、K. Lu、C. Cang、L. Pinto 和 P. Abbeel。URLB：无监督强化学习基准。arXiv 预印本 arXiv:2110.15191，2021年。[31] D. Pathak、D. Gandhi 和 A. Gupta。通过分歧实现自监督探索。发表于国际机器学习会议，第5062–5071页。PMLR，2019年。[32] S. Endrawis、G. Leibovich、G. Jacob、G. Novik 和 A. Tamar。用于离线机器人学习的高效自监督数据收集。发表于2021年 IEEE 国际机器人与自动化会议（ICRA），第4650–4656页。IEEE，2021年。[33] Y. J. Ma、J. Yan、D. Jayaraman 和 O. Bastani。我将走多远：通过 f-优势回归进行离线目标条件强化学习。arXiv 预印本 arXiv:2206.03023，2022年。[34] T. Haarnoja、A. Zhou、P. Abbeel 和 S. Levine。软演员-评论家：采用随机演员的离策略最大熵深度强化学习。发表于国际机器学习会议，第1861–1870页。PMLR，2018年。[35] Y. Kanagawa。mujoco-maze。https://github.com/kngwyu/mujoco-maze，2020年。

A 实现细节

A.1 自监督奖励塑形

我们为所提方法的两个阶段提供了伪代码：图构建过程见算法1，填充回放缓冲区的步骤见算法2。这里我们使用记号 $R(s, \mathcal{M})$ 表示 s 与 $\mathcal{M}$ *i.e.*, 中所有节点之间的 RNet 值的最大值

$$R(s, \mathcal{M}) := \max_{m \in \mathcal{M}} R(s, m)$$

算法1构建有向图 $\mathcal{M}$

输入：预先收集的数据集 $\mathcal{D}$，可达性网络 R初始化：$\mathcal{M} = \{\}$/ * 构建节点集合 * / 对于 $\mathcal{D}$ 中的每个状态 s，执行：如果 $R(s, \mathcal{M}) < 0.5$ 且 $R(\mathcal{M}, s) < 0.5$，则更新 $\mathcal{M} := \mathcal{M} \cup \{s\}$结束 if 结束 for / * 构建边 * / 对于 $\mathcal{D}$ 中的每个转移 (s_t, s_{t+1})，执行：令 $n_t :=$ $\text{NN}_{\text{in}}(s_t) = \text{argmax}_{n \in \mathcal{M}} R(n, s_t)$ 令 $n_{t+1} := \text{NN}_{\text{out}}($ $s_{t+1}) = \text{argmax}_{n \in \mathcal{M}} R(s_{t+1}, n)$ 创建一条从 n_t 到 n_{t+1}的有向边 结束 for

算法2构建离线策略训练的回放缓冲区 $\mathcal{B}$

输入： 预先收集的数据集 $\mathcal{D}$、可达性网络 R、有向图 $\mathcal{M}$

初始化： $\mathcal{B} = \{\}$

当 $\mathcal{B}$ 未满**时执行**

在 $\mathcal{D}$ 中随机采样一个转移 (s_t, a_t, s_{t+1})

在 $\mathcal{D}$ 中随机采样一个目标 g

计算 $d_l(s_{t+1}, g) := 1 - R(s_{t+1}, g)$

令 $n_{t+1} := \text{NN}_{\text{out}}(s_{t+1}) = \text{argmax}_{n \in \mathcal{M}} R(s_{t+1}, n)$

令 $n_g := \text{NN}_{\text{in}}(g) = \text{argmax}_{n \in \mathcal{M}} R(n, g)$

计算 $d_g(s_{t+1}, g) :=$ 最短路径长度(n_{t+1}, n_g)

计算 奖励 $r_t := -(d_l(s_{t+1}, g) + d_g(s_{t+1}, g))$

用目标 g 和奖励 r_t重新标记转移，并将 $(s_t, a_t, g, r_t, s_{t+1})$ 推送到 $\mathcal{B}$

结束循环

A.2 可操作模型基线方法的重新实现

本节详细介绍我们对可操作模型 [1] 和 HER [2]的重新实现。由于这些基线方法与我们的方法在离线策略训练中使用相同的优化算法，唯一的区别在于如何对预先收集的数据集 $\mathcal{D}$ 中的转移重新标记，以构建回放缓冲区 $\mathcal{B}$。

在 HER 中，[2], 想法是在 $\mathcal{D}$ 中随机采样一条轨迹和一个目标 g，并在某个步骤 i 截断轨迹。随后，将轨迹中的每个转移（直到步骤 i）重新标记两次：一次使用目标g，并将所有转移的奖励设为 0；另一次使用目标 s_i（轨迹的最终状态），并将奖励

设为 0，但最后一个转移的奖励为 1。该方法的伪代码如图 8所示。

可操作模型 [1] 依赖于与 HER 中类似的理念 [2], 但包含两个额外步骤，以改进该方法。第一步是一种 目标链式连接 ，其做法是使用轨迹最终状态处的 Q 值来计算最终转移的奖励。第二步旨在平衡未观测到的动作影响，从而对动作空间进行正则化。具体而言，该步骤从策略中采样负动作，并使用这些动作将转移标记为零奖励。两种技巧的实现如 图 8所示。

对于第三种基线方法 HER + 随机负动作，其整体算法与 HER 相同，不同之处在于，我们还会像 Ac- tionable Models 一样生成包含负动作的转移。这一次，负动作不是从策略中采样的，而是在动作空间中均匀随机生成的。

算法 3HER

输入： 数据集 $\mathcal{D}$

初始化： $\mathcal{B} = \{\}$

当 $\mathcal{B}$ 未满**时**
采样轨迹 $\tau \in \mathcal{D}$
示例目标 $g \in \mathcal{D}$
在步骤 i随机截取 τ
对 $j \in \{0, ..., i-2\}$ **执行**
$(s_j, a_j, g, 0, s_{j+1}) \rightarrow \mathcal{B}$
$(s_j, a_j, s_i, 0, s_{j+1}) \rightarrow \mathcal{B}$
结束循环
$(s_{i-1}, a_{i-1}, g, 0, s_i) \rightarrow \mathcal{B}$
$(s_{i-1}, a_{i-1}, s_i, 1, s_i) \rightarrow \mathcal{B}$
结束 while

算法 4可操作模型

输入： 数据集 $\mathcal{D}$,
目标条件评论家网络 Q,
目标条件策略 π
初始化： $\mathcal{B} = \{\}$

当 $\mathcal{B}$ 未满**时**
采样轨迹 $\tau \in \mathcal{D}$
采样目标 $g \in \mathcal{D}$
在步骤 i处随机截断 τ
对 $j \in \{0, ..., i-2\}$ **执行**
$(s_j, a_j, g, 0, s_{j+1}) \rightarrow \mathcal{B}$
$a_1 \sim \pi(s_j, g)$
$(s_j, a_1, g, 0, s_{j+1}) \rightarrow \mathcal{B}$
$(s_j, a_j, s_i, 0, s_{j+1}) \rightarrow \mathcal{B}$
$a_2 \sim \pi(s_j, s_i)$
$(s_j, a_2, s_i, 0, s_{j+1}) \rightarrow \mathcal{B}$
结束循环
$(s_{i-1}, a_{i-1}, g, Q(s_i, a_{i-1}, g) \; s_i) \rightarrow \mathcal{B}$
$a_3 \sim \pi(s_{i-1}, g)$
$(s_{i-1}, a_3, g, 0, s_i) \rightarrow \mathcal{B}$
$(s_{i-1}, a_{i-1}, s_i, 1, s_i) \rightarrow \mathcal{B}$
$a_4 \sim \pi(s_{i-1}, s_i)$
$(s_{i-1}, a_4, s_i, 0, s_i) \rightarrow \mathcal{B}$
当.....时结束循环

图 8：使用 HER 填充回放缓冲区的伪代码 [2] 和可操作模型 [1] 方法。我们通过以下方式比较这两种实现：将与目标链式连接相关的修改标为红色，将与未见动作正则化相关的修改标为蓝色。

A.3 超参数

我们首先列出自监督奖励塑形阶段的超参数，见 表1。 表 2详细列出了使用 SAC 进行离线策略训练阶段的超参数 [34]。对于 Action- able Models [1]和 HER [2]基线方法，我们使用了与我们的方法相同的参数，但这些方法特有的部分参数除外，相关参数见 表3。

这些超参数是通过对所有方法在多个参数值上进行随机搜索得到的。本研究中的所有实验均使用 3 个随机种子进行。

通用超参数	数值		
Task	UMaze	RoboYoga	Pusher
训练对数量	5×10^5	5×10^5	5×10^5
负样本比例	0.5	0.5	0.5
来自同一条轨迹的负样本比例	0.5	0.5	0.5
可达性阈值（τ_{reach}）	5	2	10
奖励中局部距离的权重	1	1	100
批次大小	512	512	512
学习率	0.001	0.0003	0.001
权重衰减	0.00001	0.00001	0.00001
训练周期总数	100	100	100
有向图容量	1000	10000	1000

表1：可达性网络训练和有向图构建的超参数。

超参数	值		
Task	UMaze	RoboYoga	Pusher
回放缓冲区容量	10^6	10^6	10^6
批次大小	2048	2048	2048
折扣因子 (γ)	0.90	0.95	0.95
每轮更新次数	1000	1000	1000
总轮数	1000	1000	1000
目标网络更新间隔	1	1	1
软更新系数 (τ)	0.005	0.005	0.005
SAC 熵参数 (α)	0.05	0.01	0.0001
优化器	Adam	Adam	Adam
学习率	0.0003	0.0003	0.0005
动作重复次数	1	2	1
奖励缩放因子	0.1	0.5	0.01

表 2：使用我们的 [34] 方法进行 SAC 离线策略学习的超参数。

超参数	值		
Task	UMaze	RoboYoga	Pusher
折扣因子（γ）	0.99	0.99	0.99
SAC 熵参数（α）	0.01	0.001	0.005
学习率	0.0001	0.0001	0.0001
奖励缩放因子	1	10	1

表3：使用 SAC 进行离线策略学习时的超参数， [34]专用于 Actionable Mod- els [1] 和 HER [2] 基线方法。

A.4 架构细节

可达性网络 [11] RNet 采用孪生架构，包含两个嵌入头（分别对应一对观测中的一个），二者权重共享；此外还包含一个比较器网络，用于比较两个嵌入并返回可达性分数。对于 UMaze 任务，我们使用了一个包含 3 个全连接层的嵌入头，这些层带有批归一化和 Tanh 激活函数，隐藏层维度为 64，嵌入维度为 16。对于 RoboYoga Walker 任务，嵌入网络采用相同的架构，但我们将隐藏层维度和嵌入维度都增至 128。比较器网络也是一个全连接网络，包含批归一化和 ReLU 激活函数。UMaze 任务的隐藏层维度设为 16，而 RoboYoga Walker 任务的隐藏层维度设为 128；两者的层数分别为 2 层和 4 层。

策略网络 目标条件策略网络分别通过两个架构相同但权重独立的分支，将观测和目标作为输入。这两个分支均采用 3 层全连接网络实现，使用 Tanh 激活函数，隐藏层维度为 64，特征维度为 16。两个分支的输出随后拼接，并输入一个宽度为 256 的 2 层全连接网络。评论家网络的观测分支和目标分支采用相同的架构，后接 3 个宽度为 256 的全连接层。

B RoboYoga Walker 任务的完整结果

我们展示了我们的方法与前述基线方法在 RoboYogaWalker 任务的 12 个目标上的比较，结果见 图 9。这些目标在 图 10中展示。可以看到，我们的方法掌握了大多数不需要保持平衡的目标（仰卧与俯卧、抬腿、弓步），并且在侧角式、后仰和桥式等更复杂的目标上表现出色，但在头倒立或阿拉贝斯克等复杂目标上无法取得进展。

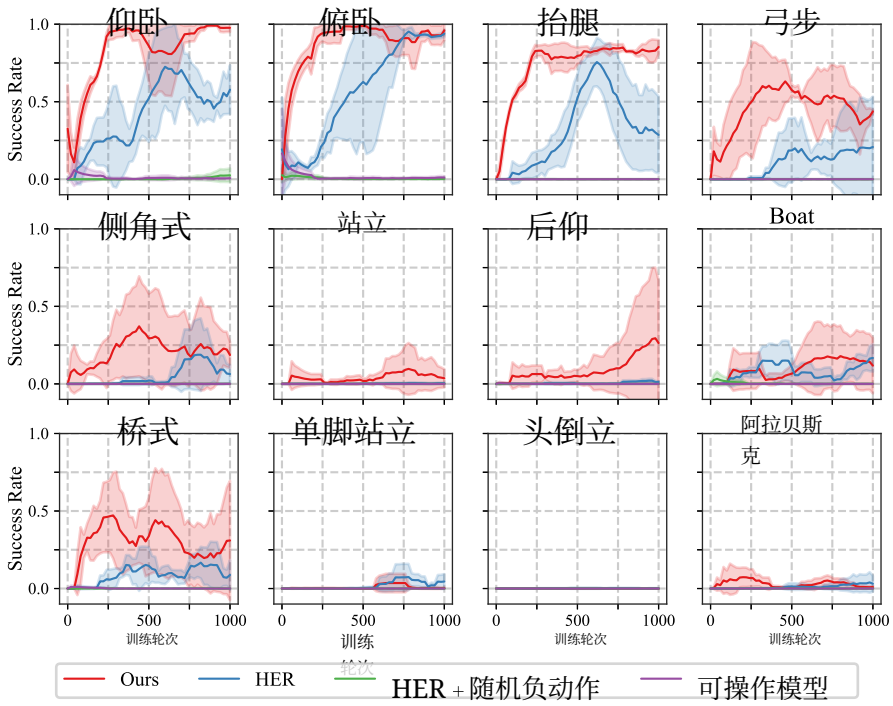


图 9：RoboYoga Walker 任务中 12 个目标各自的表现。

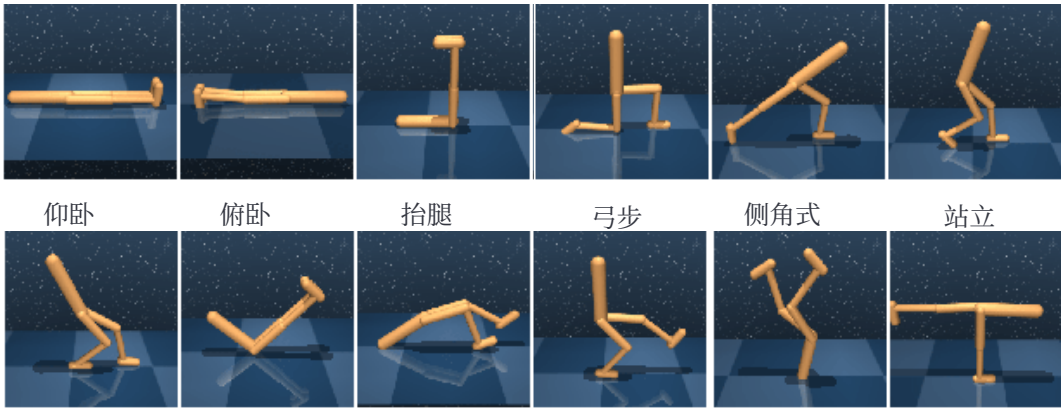


图 10：RoboYoga Walker 任务的 12 个评估目标的可视化。