D2SR: Transferring Dense Reward Function to Sparse by Network Resetting

Yongle Luo[a,b], Yuxin Wang[a,b], Kun Dong[a,b], Yu Liu[a,b], Zhiyong Sun[a,b], *Member, IEEE*, Qiang Zhang[a,b] and Bo Song[#,a,b,c], *Member, IEEE*

***Abstract*— 在强化学习中，大多数算法使用固定的奖励函数，很少有研究讨论在学习过程中如何转换奖励函数。实际上，不同类型的奖励函数具有不同的特点。一般来说，具有形状丰富的奖励函数具有快速引导智能体到达高价值状态的优点，但缺点是难以设计一个完美的塑性函数并且容易受到噪音的干扰。稀疏奖励的优点是鲁棒性强，与任务一致，但在探索的早期效率较低。因此，本文提出了一种名为Dense2Sparse by Network Resetting（D2SR）的算法，同时满足了密集奖励函数的效率和稀疏奖励的鲁棒性。具体而言，D2SR方法通过重置网络参数和将经验转化为稀疏奖励，从而挽救了智能体受到次优密集奖励误导的情况，从而在指向全局最优的方向上实现了显著的改进。在本研究中，通过一系列关于具有挑战性的机器人操作任务的消融实验，我们发现D2SR能够降低对密集奖励函数设计的要求，同时在具有噪音奖励的任务中兼顾了效率和性能。**

I. Introduction

深度强化学习 (RL) 在各种决策领域取得了显著的进展，如围棋 [1]、游戏 [2]、自动驾驶 [3] 和机器人操作任务 [4], [5]等。大多数RL算法需要一个奖励函数来提供反馈 [6]。RL agent的学习表现在很大程度上取决于奖励函数的设计。一般来说，大多数RL算法在整个学习周期中只使用固定的奖励函数 [7]–[9]，而很少有文章讨论在agent的学习过程中切换不同的奖励函数 [10], [11]。

本研究主要关注RL中的奖励函数切换问题，特别针对标准操作任务，提出了一种强大而有效的通用切换方案：通过网络重置实现从密集到稀疏的奖励切换(D2SR)。

奖励函数可以大致分为两种类型：密集和稀疏。密集奖励函数通常在每个状态下为agent提供具体且特定的反馈 [12]，使其能够区分哪些动作更好。然而，设计一个全局最优的密集奖励函数是具有挑战性的，而且密集奖励更容易受到噪声信号的干扰，这些信号可以通过Bellman方程传播和放大 [13]–[15]。另一方面，稀疏奖励函数使用二进制奖励表示任务完成，失败时总是获得惩罚奖励，成功时才有正面奖励 [4], [16], [17]。这种类型的奖励不太容易受到噪声的影响，如果有足够的成功样本，agent才可以学习到全局最优策略 [18]。然而，从零开始学习的agent在学习过程中可能会面临一个悖论：“agent需要一个好的策略来完成任务，如果没有完成任务，它只能接收到相同的惩罚奖励样本，而Agent几乎无法从这些低质量数据中学习到好的策略”。

这两种奖励函数的优缺点非常明显且互补。密集奖励函数在早期探索中具有高效性，但设计一个全局最优的密集奖励函数很难，并且容易受到噪声的影响。另一方面，稀疏奖励函数在早期探索中效率较低，但是如果提供了有价值的样本，由稀疏奖励引导的agent在原则上可以与任务更加一致，并且对噪声更具鲁棒性。

因此，所提出的解决方案旨在结合这两种奖励函数的优点，同时避免它们的缺点。之前的提议“Dense2Sparse”（D2S）算法 [11]在有噪声的操作任务中利用密集奖励在早期阶段获取有价值的数据，然后将其转化为稀疏奖励函数，进行后续学习的调优。然而，在复杂操作任务中，D2S这种奖励函数转换方案，也只是一个次优的解决方案。与单纯利用密集奖励相比，其调节效果相对有限。

This work is supported in part by NSFC (61973294), Anhui Provincial Key R&D Program (2022i01020020), and the University Synergy Innovation Program of Anhui Province, China (GXXT-2021-030).

[a]Institute of Intelligent Machines, Hefei Institute of Physical Science, CAS, Hefei, 230031, China

[b]University of Science and Technology of China, Hefei 230026, China

[b]Jianghuai Frontier Technology Coordination and Innovation Center, Hefei, 230088, China

[#]Corresponding author, email: songbo@iim.ac.cn

最近，Nikishin等人 [19] 指出DRL算法容易受到"首因偏差"的影响，即在初期低质量样本中，网络过度拟合，从而阻碍了agent在后续经验中进一步优化策略。他们的工作中提供了一种减轻"首因偏差"的有效机制，即重置网络。通过重置网络的参数，agent可以恢复可塑性并消除从次优数据中获得的先验偏差。

受到上述研究的启发，我们重新思考了D2S的问题。尽管后续阶段，D2S采用了稀疏奖励的样本进行微调，但从早期阶段使用有噪声的密集奖励样本训练网络产生的偏差，阻碍了网络的后续学习。因此，我们的目标是利用上述"硬重置"技术来减轻这种偏差，期望能够有效地利用密集和稀疏奖励的优点。

具体来说，本研究首先使用非全局最优的密集奖励函数来引导agent进行快速探索。在收集了一定数量的高价值样本后，将奖励函数转换为稀疏奖励函数。此外，我们需要重置网络参数，将网络参数重新随机初始化，并使用稀疏奖励函数，重新计算收集到的所有样本的奖励。所有后续的学习和探索过程都将在稀疏奖励函数的指导下进行。我们将这种方法称为"通过网络重置实现的从密集到稀疏的切换(D2SR)"。与之前的D2S方案相比，D2SR只需要重置网络参数，并使用稀疏奖励函数重新计算过去的经验。然而，这些修改极大的提高了整体性能。本研究首先证明D2SR能够消除密集奖励样本的偏差，并最终高效地让策略接近全局最优，相比D2S在机器人操作任务中具有明显的优势。通过进一步的详细消融实验，对D2SR的每个组成部分的影响进行了全面评估。四种不同的密集奖励函数的实验结果表明，D2SR能够减少对密集奖励函数的设计要求。最后，在具有噪声奖励信号的任务中，进一步研究了D2SR方案的鲁棒性。总体而言，D2SR为利用不同奖励函数提供了一个有希望的路径。

II. Related Work

一些深度强化学习算法讨论了切换不同奖励函数的问题。基于后继的算法 [10], [20]将值函数分解为预测表示和奖励函数，通过更新奖励模型可以快速适应新的奖励函数。然而，这些研究并没有考虑用新奖励函数带来额外的性能提升。最相关的工作是D2S算法 [11]，它旨在融合密集和稀疏奖励函数的优点。然而，这些研究都没有考虑先前奖励函数对网络所引入的偏差，这可能会影响新奖励函数的有效性。

最近的研究从不同的角度来解决这个问题，包括容量损失 [21]、可塑性丧失 [22]和首因偏差 [19]。这些文献指出，在学习先前数据之后，深度强化学习会丧失一些学习新数据的能力。为了解决这个问题，本文采用了 [19]中提出的"硬重置"方案，以消除密集奖励样本引入的偏差，从而更好地适应学习新的稀疏奖励函数。最后，关于稀疏奖励数据的收集，与基于示教的方法 [18]相比，D2SR的高价值样本不由人类示教者提供，而只需为RL智能体设计一个要求较低的密集奖励函数，由其自行收集即可。

III. Background

本文主要关注于用于操作任务的off-policy Actor-Critic 结构深度强化学习算法的奖励转换。因此，本节介绍了一些背景知识，包括强化学习和经典的Actor-Critic算法。

A. Reinforcement Learning

强化学习（RL）由一个与环境进行交互的智能体组成。智能体通过环境的反馈学习，优化基于最大化总期望奖励的策略 [13]。把标准RL形式建模为马尔科夫决策过程（MDP），MDP由状态集合S中的状态s，动作集合A中的动作a，动力学$p(s'|s,a)$，奖励函数$R(s,a)$和折扣因子$\gamma \in (0,1)$定义。在每个时间步t，智能体根据当前状态s_t执行动作a_t。随后，智能体接收到奖励信号$r_t = R(s_t,a_t)$和下一个状态s_{t+1}，其由分布$p(s'_{t+1}|s_t,a_t)$采样得到。强化学习智能体从策略$\pi(s_t)$中选择动作a_t，该策略直接将状态映射到动作。强化学习的最终目标是通过最大化期望回报$R_t = \sum_{i=t}^{T} \gamma^{i-t} r_i$来学习最优策略$\pi^*$，其中$T$是时间范围。

B. Deep Deterministic Policy Gradient

深度确定性策略梯度（Deep Deterministic Policy Gradient, DDPG） [7] 是一种经典的无模型Off-Policy强化学习算法，在连续控制任务中表现出卓越的性能。作为一个Actor-Critic框架，DDPG利用Critic网络来指导策略的学习：使用一个近似策略π_θ的Actor网络和一个学习动作价值函数Q_ϕ的Critic网络。

为了稳定性，DDPG采用了一组目标网络π'_θ和Q'_ϕ，这些网络具有相同的结构，但对更新进行延迟。评价

网络的参数 ϕ 通过最小化贝尔曼误差来更新，损失函数定义如下：

$$Loss_\phi = |Q_\phi(s_t, a_t) - y_t|^2, \tag{1}$$

目标值$y_t = r_t + \gamma Q'_\phi(s_{t+1}, a_{t+1})$和动作$a_{t+1}$由$\pi'_\theta(s_{t+1})$生成。

基于Q函数，使用梯度下降法训练策略网络参数θ，损失函数在(2)中定义：

$$Loss_\theta = -\mathbb{E}_{s_t} Q[s_t, \pi_\theta(s_t)]. \tag{2}$$

DDPG算法还需要维护一个经验池，用来存储先前经历过的transition(s_t, a_t, s_{t+1}, r_t)。这使得智能体能够基于一组不相关的transition来更新自身，从而通过打破更新中的时间相关性来稳定训练过程。

众所周知，off-policy强化学习算法Soft Actor-Critic（SAC）[23]和延迟双重DDPG（TD3）[8]是DDPG的改进版本。在本工作中，我们使用TD3作为骨干算法。

IV. Method

与仅使用单一奖励函数的标准RL方法相比，本文介绍了一种名为D2SR的方法，可以使RL agent更有效地利用多个奖励函数进行学习。

具体来说，本文研究了两种不同的奖励函数：密集奖励和稀疏奖励。密集奖励被认为可以促进快速学习，但容易陷入局部最优解。相反，稀疏奖励在初期学习时较慢，但更有可能收敛到全局最优解。为了演示如何利用这两种奖励函数的优点，本文以标准操作任务作为案例进行研究，展示了使用D2SR训练agent，可以实现效率和性能的提高。

直观的解决方案是先使用一个简单的密集奖励函数来引导agent收集足够数量的高价值样本，然后切换到稀疏奖励函数进行性能调优，就像D2S中提到的那样。然而，这种方案忽略了早期密集奖励样本产生的两个隐含问题。

一个问题是密集奖励样本仍然保留在经验经验池中，在后续更新过程中，可能会继续引入偏见。直接删除这些样本也是不合理的，因为这会导致经验池中丢失已经搜集到的高价值数据，从而会对网络梯度产生剧烈改变，并最终导致策略崩溃。因此，我们提出使用更稳健的稀疏奖励函数，对过去样本的奖励函数重新计算。这种奖励的重新计算不会改变环境的动力学，从而避免引入任何新的偏差。

第二个问题是在用密集奖励样本训练之后，网络参数会存在“首因偏差”，这会影响后续稀疏奖励样本的学习。目前，还没有同时能够保持网络策略性能、并恢复其作为随机初始化的可塑性的正则化方法。为了解决这个问题，D2SR采用了 [19]中概述的硬重置技术来减轻首因偏差。虽然这种方法会导致短暂的性能下降，但经验池中的数据，可以被视为世界的非参数模型，允许agent快速从短暂的在线交互和正常的Off-policy更新中恢复。此外，在D2SR框架中，经过缓解首因偏差之后，高价值的稀疏奖励样本可以极大地提高性能，远远超过以前的密集奖励样本中训练的性能。在某种程度上，这是一种“以退为进”的策略。

最后，D2SR方案可以重置网络并重新计算收集样本的奖励值，以兼顾密集奖励函数的高效性和稀疏奖励的性能优势。因此，D2SR降低了设计复杂的、专家级密集奖励函数的要求。

V. Experimental setup

在本节中，我们首先介绍了在我们的实验中使用的机器人模拟环境。随后，我们描述了在我们的研究中使用的奖励函数，包括四个针对操作任务特点，量身定制的密集奖励函数，以及一个通用的稀疏奖励函数。

A. Task setup

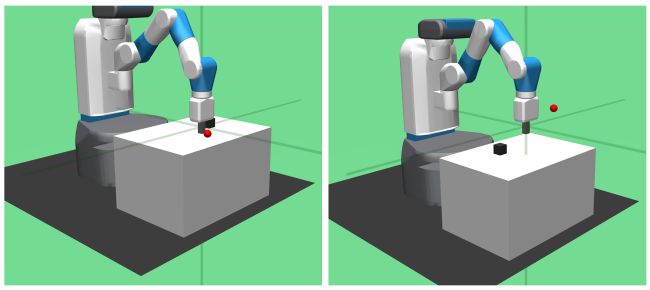

Fig. 1. 机器人模拟环境：FetchPush、FetchPickAndPlace。

我们实验中使用的仿真环境由OpenAI Gym提供[26]，它提供了一套复杂的机器人连续控制任务。在我们的实验中，我们使用一个七自由度的Fetch机械臂，配备了一个双指并行夹爪，作为我们的机器人智能体。机器人的仿真采用MuJoCo物理引擎 [27]。我们在两个具有挑战性的任务中评估方法D2SR，包括FetchPush和FetchPickAndPlace，如图1所示。在这

些任务中，状态空间包括夹爪和末端的位置以及线速度，物体的位置、旋转、角速度以及相对于夹爪的位置和线速度。动作空间是4维的，其中3个维度控制夹爪的移动增量，第四个维度控制夹爪的张合。作为标准的目标条件任务，这些任务的目标，是让物体达到所需位置，即图1中的红点位置。

B. Reward functions

为了证明D2SR与各种密集奖励函数的兼容性，在本节中，我们介绍了四个不同的密集奖励函数，用于所选取的操作任务。随后，我们介绍了对这些操作任务采用的稀疏奖励函数。

在 [4]中，通过目标重标记技术，证明了DDPG与HER可以成功地在标准操作任务中，学得良好的策略，但论文声称，无法使用密集奖励函数获得好的策略。HER论文中的密集奖励函数R_d^1表示如下：

$$R_d^1(s_t, a_t) = -d_2, \tag{3}$$

其中，d_2表示物体与物体目标之间的距离，单位为米（m）。

正如 [28], [29]所指出的，对于学习操作任务来说，学会接触物体，对于智能体的策略更新非常重要。然而，原始的密集奖励函数（R_d^1）只考虑了物体和物体目标之间的距离，无法有效地指导智能体完成任务。为了解决这个问题，我们设计了三个额外的奖励函数，考虑了夹爪与物体之间的距离附加项。它们中的第一个是R_d^2。

$$R_d^2(s_t, a_t) = -d_1 - d_2, \tag{4}$$

其中d_1表示夹爪和物体之间的距离，单位为米（m）。

第二个是用R_d^3将每个项映射到区间$(0, 1)$。

$$R_d^3(s_t, a_t) = 1 - tanh(d_1) + 1 - tanh(d_2). \tag{5}$$

肚脐在 [30]指出，正向奖励可能会对深度强化学习网络的探索能力产生负面影响。因此，为了研究奖励偏移对性能的影响，第三个密集奖励函数R_d^4通过向R_d^2添加一个偏移量来增加奖励的绝对值。

$$R_d^4(s_t, a_t) = 1 - d_1 - d_2. \tag{6}$$

在本研究中，我们利用标准稀疏奖励函数R_s来进行性能调优，这与原始的HER算法一致。稀疏奖励函数的定义如下：

$$R_s(s_t, a_t) = -\mathbb{I}(d_2 < 0.05), \tag{7}$$

其中，$\mathbb{I}(\cdot)$是指示函数，当其参数为真时返回1，否则返回0。该奖励函数会在任务未完成时将奖励分配为-1，在任务完成时将奖励分配为0。任务完成的条件被定义为物体与目标之间的距离d_2小于0.05米。

VI. Experimental Results and Analysis

本节对比了基准模型D2S与提出的D2SR方法在两个标准操作任务上的性能。随后，我们进行了全面的消融研究，评估了D2SR中引入的新操作对性能的影响。此外，我们展示了D2SR对各种奖励函数的兼容性，从而减少了对设计密集奖励函数的专业知识要求。最后，我们评估了D2SR在有噪声奖励信号任务中的鲁棒性。

A. Compared to Baselines

为了评估所提出方法的性能，我们将TD3作为基础强化学习算法，利用HER方法中引入的目标重标记技术，来增加目标条件任务中的成功样本。评估包括四种方法，即D2SR、D2S、Dense和Sparse。Dense和Sparse方案，表示在整个训练过程中全程使用密集或者稀疏的奖励函数（分别为R_d^3和R_s）。

本文采用通用的设置，选用HER中的未来策略（future-strategy），以及设置重标记数k = 4，将轨迹中的原始目标替换为已实现的目标，并在所有方法中的经验回放过程中使用。我们比较了这四种方法在五个固定的随机种子上的平均测试成功率，其中阴影区域表示标准差。智能体在FetchPush和FetchPickAndPlace环境中使用一个CPU核心进行300个Epoch的训练，每个Epoch包含50个回合。D2SR和D2S在第150个Epoch将奖励函数从R_d^3转换为R_s。隐藏层由256个单元组成，最后输出单元的数量是任务的状态和期望目标的维度总和。TD3网络的参数与DDPG+HER（OpenAI Baselines）中的参数相同。我们使用大小为1e6的经验池，并使用批量大小为2048来训练策略。

学习曲线如图2所示。实证结果清楚地表明，当使用一个单一固定的奖励函数时，两个任务都会面临次

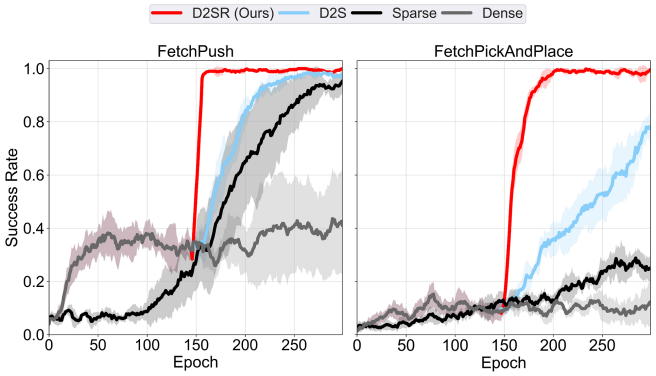

Fig. 2. 在所有两个环境中，对D2SR与其他方法的测试成功率进行比较。

优性能的困扰。具体而言，在FetchPickAndPlace任务中，稀疏奖励函数R_s遇到了探索困境，而密集奖励函数R_d^3会引导智能体进入次优性能，这些现象在强化学习中都很常见。尽管之前的方法D2S在150个epochs之后通过引入新的稀疏样本略微提升了性能，但这种改进并不显著。相反，我们提出的D2SR方法在两个任务中都实现了显著的改进，在仅进行少数交互后成功率接近100%。尽管D2SR由于重置网络参数，而在短期内经历了性能下降，但由于经验池中保留了高质量稀疏奖励样本，从而使得智能体能够迅速恢复，并超越以前密集奖励的性能瓶颈，进而获得性能大幅提升。此外，相比其他三种方法，D2SR的曲线在150个epochs之后的阴影区域更少，表明本方法更加稳定，且受到随机种子的影响更小。这些结果强有力地证明了，D2SR能够仅仅通过网络重置和重新计算奖励这两个操作，就可以同时融合密集和稀疏奖励函数的优势，从而实现令人印象深刻的性能提升。

B. Ablation Studies

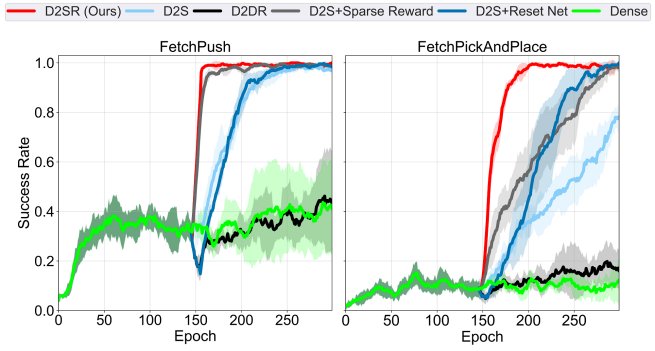

Fig. 3. 两个标准机器人环境上的消融结果。

在上述实验中，D2SR表现出比D2S更显著的性能提升。为了进一步探究D2SR中每个组成部分的贡献，我们进行了详细的消融实验。图3展示了D2S+Sparse Reward（不重置网络的D2SR）的学习曲线。结果表明，D2S+Sparse Reward在FetchPickAndPlace任务中，性能曲线上升明显较慢，相比D2SR速度慢很多。然而，D2S+Sparse Reward在两个任务中仍优于之前的方法D2S，从而进一步证实经验池中具有密集奖励的样本，确实阻碍了学习性能的提升。

相反，D2S+Reset Net方法（仅重置网络），不用稀疏奖励函数重新计算经验池中的样本。实验结果显示，在两个任务中，经过150个Epochs后性能显著下降，并且恢复速度比D2SR慢，但最终性能优于D2S。这些发现揭示了重置网络，可以提高智能体学习新数据的能力。然而，当存在具有密集奖励样本的干扰时，网络重置带来的优势也会被限制。因此，通过重置网络、并使用稀疏奖励函数重新计算所有收集到的样本，D2SR不仅恢复了网络的学习能力，还增强了已收集样本的鲁棒性，最终实现了性能的显著提升。

C. Different Dense Reward Functions

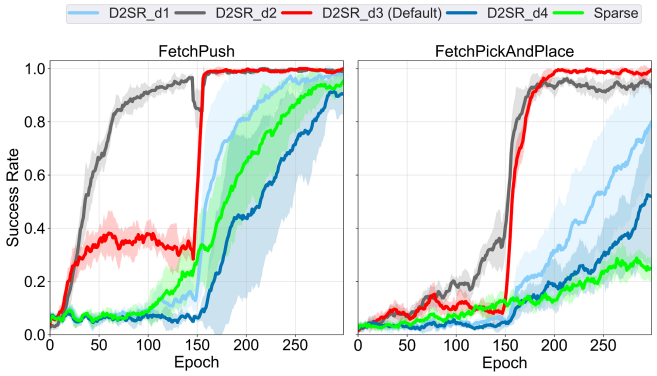

Fig. 4. 在两个标准的机器人环境中，四个不同的密集奖励函数。

本小节评估了使用不同密集奖励函数（即D2SR_d1，D2SR_d2，D2SR_d3和D2SR_d4）的D2SR算法的性能。学习曲线如图4所示。四个不同的密集奖励函数在前150个Epoch中的每个任务的学习性能方面有显著差异。整体实验结果表明，当利用密集奖励函数学习到的性能越高，D2SR的最终性能改进也会越好。然而，即使在150个Epoch时成功率低于10%，相比原始的密集奖励函数，D2SR仍然表现出了相当大的性能提升。对于那些低于纯稀疏奖励的D2SR曲线，我们应该可以通过多次重置来获得性能改进，这也将是我们未来工作的方向。在本研究中，我们仅在固定的Epoch切换奖励函数，并将成功

率作为性能指标。在未来的研究中，我们将进一步寻找新的指标来评估样本的质量和切换标准。

此外，在这个实验中还有一个有趣的发现。原版的HER论文里声称，在他们的尝试中，即使在R_d^1奖励函数中，添加一个线性项以鼓励夹爪接近物体，也不能训练出一个成功的策略。然而，我们的实验结果表明，不同的奖励项会导致了显著性能差别，是可以找到一个附加项，**使得HER在dense奖励函数中，也可以学会操作任务**。具体来说，密集奖励函数R_d^2，提供了足够多的信息以指导智能体，让智能体能够在150个Epoch内，学习到FetchPush任务。相比之下，密集奖励函数R_d^4则无法学习到一个良好的策略。我们推测，从探索的角度出发，也许可以提供一个潜在的解释：正面的奖励信号会隐式地妨碍网络的探索 [30]。总之，我们的实验结果证明，在操作任务中，HER算法也可以有效地利用密集奖励函数，甚至比稀疏奖励更高效，而非原版论文里声称的HER无法使用密集奖励函数。

D. Robust to Noisy Rewards

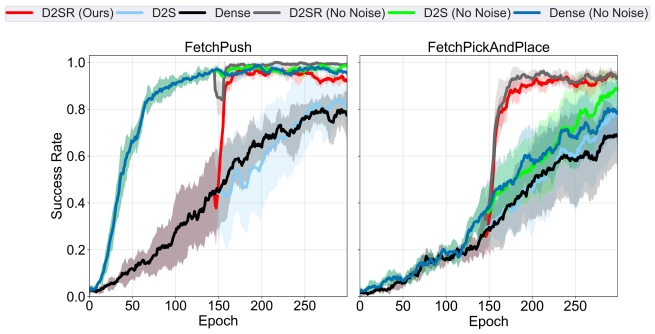

Fig. 5. 在奖励和状态具有随机噪声的任务上，评估D2SR的鲁棒性。

为了评估D2SR在噪声奖励下的鲁棒性，我们在后续实验中，对状态和奖励信号添加了均匀分布[-0.01, 0.01]的噪声。基于第VI.C节中呈现的结果，我们选择了性能最佳的密集奖励函数r_d^2进行进一步评估。三种方法在有噪声和无噪声情况下的性能对比如图5所示。我们观察到，在噪声存在的情况下，Dense with noise的学习曲线性能明显下降，尤其在FetchPush任务中。之前的方法D2S在噪声环境中没有表现出明显的性能提升。相反，D2SR能够利用稀疏奖励函数，显著提高最终性能，并且受到噪声的影响较小。这些实验表明，在存在噪声奖励的情况下，D2SR能够有效地利用稀疏奖励函数的鲁棒性。

VII. Discussion and Conclusion

在本研究中，我们提出了一种简洁高效的奖励函数切换方法，D2SR，该方法融合了密集奖励函数和稀疏奖励函数的优势。通过一系列的操作任务实验，我们证明了所提出的方法D2SR能够快速突破原始密集奖励函数的限制，并避免使用单一固定稀疏奖励函数时的探索困境，实现性能和效率的双重目标。此外，对四种不同密集奖励函数的对比实验表明，D2SR能够降低设计密集奖励函数的要求。此外，噪声实验表明，D2SR能够利用稀疏奖励函数的鲁棒性，为具有噪声奖励的任务提供稳定且改进的性能。D2SR方案使智能体，能够利用密集奖励函数收集高质量样本，并使用稀疏奖励函数进行进一步的性能调优，为利用不同奖励函数提供了一条有前景的道路。在未来的工作中，我们将探索同一奖励函数和不同奖励函数内部的多次转换，以最大化样本效率。

References

[1] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[3] R. Zhang *et al.*, "Residual policy learning facilitates efficient model-free autonomous racing," *IEEE Robotics and Automation Letters*, pp. 1–8, 2022.

[4] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. P. Abbeel, and W. Zaremba, "Hindsight experience replay," in *Advances in neural information processing systems*, 2017, pp. 5048–5058.

[5] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray *et al.*, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

[6] M. J. Mataric, "Reward functions for accelerated learning," in *Machine Learning Proceedings 1994*. Elsevier, 1994, pp. 181–189.

[7] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[8] S. Fujimoto, H. Van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," *arXiv preprint arXiv:1802.09477*, 2018.

[9] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[10] A. Zhang, H. Satija, and J. Pineau, "Decoupling dynamics and reward for transfer learning," *arXiv preprint arXiv:1804.10689*, 2018.

[11] K. Dong, Y. Luo, E. Cheng, Z. Sun, L. Zhao, Q. Zhang, C. Zhou, and B. Song, "Balance between efficient and effective learning: Dense2sparse reward shaping for robot manipulation with environment uncertainty," in *2022 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2022, pp. 1192–1198.

[12] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *ICML*, vol. 99, 1999, pp. 278–287.

[13] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine learning*, vol. 3, pp. 9–44, 1988.

[14] J. Wang, Y. Liu, and B. Li, "Reinforcement learning with perturbed rewards," *arXiv preprint arXiv:1810.01032*, 2018.

[15] Q. He and X. Hou, "Wd3: Taming the estimation bias in deep reinforcement learning," in *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, 2020, pp. 391–398.

[16] M. Plappert *et al.*, "Multi-goal reinforcement learning: Challenging robotics environments and request for research," *arXiv:1802.09464*, 2018.

[17] B. Manela *et al.*, "Curriculum learning with hindsight experience replay for sequential object manipulation tasks," *Neural Networks*, vol. 145, pp. 260–270, 2022.

[18] M. Vecerik, T. Hester, J. Scholz, F. Wang, O. Pietquin, B. Piot, N. Heess, T. Rothörl, T. Lampe, and M. Riedmiller, "Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards," *arXiv preprint arXiv:1707.08817*, 2017.

[19] E. Nikishin, M. Schwarzer, P. D' Oro, P.-L. Bacon, and A. Courville, "The primacy bias in deep reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2022, pp. 16 828–16 847.

[20] E. Z. Liu, R. Keramati, S. Seshadri, K. Guu, P. Pasupat, E. Brunskill, and P. Liang, "Learning abstract models for strategic exploration and fast reward transfer," *arXiv preprint arXiv:2007.05896*, 2020.

[21] C. Lyle, M. Rowland, and W. Dabney, "Understanding and preventing capacity loss in reinforcement learning," *arXiv preprint arXiv:2204.09560*, 2022.

[22] S. Dohare, R. S. Sutton, and A. R. Mahmood, "Continual backprop: Stochastic gradient descent with persistent randomness," *arXiv preprint arXiv:2108.06325*, 2021.

[23] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel *et al.*, "Soft actor-critic algorithms and applications," *arXiv preprint arXiv:1812.05905*, 2018.

[24] R. Agarwal, M. Schwarzer, P. S. Castro, A. Courville, and M. G. Bellemare, "Reincarnating reinforcement learning: Reusing prior computation to accelerate progress," *arXiv preprint arXiv:2206.01626*, 2022.

[25] H. P. Van Hasselt, M. Hessel, and J. Aslanides, "When to use parametric models in reinforcement learning?" *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[26] G. Brockman *et al.*, "Openai gym," *arXiv:1606.01540*, 2016.

[27] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.

[28] B. Manela *et al.*, "Bias-reduced hindsight experience replay with virtual goal prioritization," *Neurocomputing*, vol. 451, pp. 305–315, 2021.

[29] Y. Luo, Y. Wang, K. Dong, Q. Zhang, E. Cheng, Z. Sun, and B. Song, "Relay hindsight experience replay: Continual reinforcement learning for robot manipulation tasks with sparse rewards," *arXiv preprint arXiv:2208.00843*, 2022.

[30] H. Sun, L. Han *et al.*, "Exploiting reward shifting in value-based deep rl," *arXiv preprint arXiv:2209.07288*, 2022.