满足形式　　通过无模型强化学习尽快满足形式化规范　　g

Mengyu Liu[1*]、**Pengyuan Lu**[2*]、**Xin Chen**[3] , **Fanxin Kong**[1] , **Oleg Sokolsky**[2] 和**Insup Lee**[2]雪城大学[2]宾夕法尼亚大学[3]代顿大学 mliu71@syr.edu、pelu@seas.upenn.edu、xchen4@udayton.edu、fkong03@syr.edu、{sokolsky、lee}@seas.upenn.edu

arXiv:2304.12508v1 [cs.LG] 25 Apr 2023

摘要

我们提出了一种无模型强化学习解决方案，即ASAP-Phi框架，以鼓励智能体尽快满足形式化规范。该框架采用分段奖励函数：对于不满足规范的轨迹，分配定量语义奖励；对于其余轨迹，则分配较高的常数奖励。随后，使用基于演员-评论家的算法训练智能体，例如软演员-评论家（SAC）或深度确定性策略梯度（DDPG）。此外，我们证明了 ASAP-Phi 生成的策略会优先尽快满足规范。我们在先进基准测试上开展了大量实验，包括消融实验。结果表明，我们的框架能够为高达 97% 的测试用例找到足够快速的轨迹，并击败基线方法。

1 引言

无模型强化学习的一项重要用途，是在不了解环境具体信息的情况下，训练智能体完成形式化规范。然而，有时我们需要智能体完成当前最先进的规范语言无法表达的任务。例如，智能体需要尽快（ASAP）到达目标，这一需求源于实时信息物理系统从攻击中恢复的场景。不幸的是，尽管“尽快完成任务”通常是实时控制中的标准要求，但如何基于数据驱动的动力学或黑箱动力学开发控制策略，仍未得到充分研究。

形式语言常用于描述控制目标。例如，离散系统和混合系统中的到达-规避问题可以分别用线性时序逻辑（LTL）[Vardi,1996] 以及信号时序逻辑（STL）[Raman 等，2014；Deshmukh 等， 2017] 来定义。现有的大多数控制器综合方法都需要系统动力学的形式模型 [Sadraddini and Belta, 2018；Donz´e 等，2015；Chang 等， 2019], 模拟器 [Ljung，1998；Degris 等，2012；Ma 等， 2021], 或对系统动力学的辨识 [Qin 等，2022] 。尽管在黑箱系统的控制器综合中使用强化学习（RL）已得到广泛研究 [Ding 等，2011；Fu and Topcu，2014；Li 等，2017；Li等，2018；Hamilton等， 2022],但很少有研究关注 ASAP 属性。

我们的目标是为黑箱系统开发能够尽快满足形式化规范的实时控制器。这一解决方案将为控制器综合过程提供指导，使其能够应对形式语言无法表达任务某些信息的情况，在我们的案例中，即尽快完成任务的要求。此外，一旦找到该解决方案，便可将其应用于各种场景。例如，当环境动力学未知时，通过解决实时恢复问题，它可以增强信息物理系统的安全性。恢复问题要求智能体找到一段控制序列，在最短时间内将系统引导至预定义的目标状态。现有方法利用系统动力学的形式化模型（例如常微分方程（ODE））来综合控制器 [Kong等，2018年；Zhang等，2020年；Zhang 等， 2021], 但据我们所知，迄今尚无工作处理黑箱动力学。

我们的框架称为 *ASAP-Phi*。该框架训练智能体，使其在 ASAP 要求下满足给定的 STL性质。此外，该框架包括：（1）构建一个特定的奖励函数，以鼓励尽快满足规范；（2）使用该奖励函数，通过基于演员-评论家的无模型强化学习算法训练控制策略。我们证明了 ASAP-Phi框架能够鼓励尽快满足性质，因为其生成的任何最优策略都必须是 ASAP 的；同时，我们设计了一种算法，将该框架应用于恢复问题，以便在避开障碍物的同时尽快到达目标集合。我们基于各种最先进的基准测试评估了该方法的有效性，例如 OpenAI Gym [Raffin 等、2021], 以及消融实验，以将 ASAP-Phi 与用于推导 ASAP 性质的替代方法进行比较。总体而言，我们的贡献包括

1. 一种名为 ASAP-Phi 的无模型强化学习框架，该框架接收状态空间、动作空间以及我们希望智能体 ASAP 满足的规范信息，并在不了解环境动力学的情况下输出控制策略，以实现该规范。2. 一项证明，表明 ASAP-Phi 生成的任何最优策略确实都能实现 ASAP 属性。3. 实验表明，ASAP-Phi 在需要 ASAP 行为的任务中最高可达到 97% 的成功率，并优于基线方法；消融实验还展示了其相较于替代方法的优势。

[*]这些作者对本工作的贡献相同

法，同时还进行了消融研- 究，结果表明其优于替代方法。

2 预备知识

2.1 符号说明

在本文中，我们使用 $\mathcal{S}$ 表示系统的 状态空间。轨迹（或轨迹记录）是状态的有限序列$\bar{s} = s_1$，……， s_k，满足对于所有 $i = 1$，……， k均有 $s_i \in \mathcal{S}$。我们只考虑系统的离散轨迹，有时使用 s_t 表示时间 t 时的系统状态。符号 $\mathcal{A}$ 表示可应用于系统的动作集合。其余符号将在文中首次出现时介绍。

2.2 演员-评论家强化学习

演员-评论家强化学习是一类无模型强化学习算法，它维护并更新两个模型，分别表示学习到的策略（演员） π 和估计的价值函数（评论家） v。训练过程旨在寻找一个估计期望价值的最优 v^*，以及一个使估计价值最大化的最优 π^*。形式化地，

$$\begin{aligned} v^*_\pi(s_t) &= \mathbb{E}_\pi[r(s_t) + \gamma v^*(s_{t+1})|s_t] \\ \pi^*(s_t) &= \arg\max_\pi v^*_\pi(s_t). \end{aligned} \tag{1}$$

其中， r 是奖励函数， s_t， s_{t+1} 分别是当前和下一个系统状态。 $\mathbb{E}_\pi$ 表示策略 π下的期望， $0 < \gamma < 1$ 是折扣因子。求解方程 (1) 是演员-评论家强化学习的目标 [Peters等， 2005]。这里，最优 v^* 是贝尔曼递归方程的一个解，取决于所选策略；最优 π^* 旨在最大化期望。

实现演员-评论家强化学习的方法多种多样。例如，软演员-评论家（SAC）利用分布的熵来正则化价值估计 [Haarnoja 等， 2018]。深度确定性策略梯度（DDPG）将策略视为确定性的，并使用演员和评论家模型的延迟副本来提高对训练数据方差的鲁棒性 [Silver 等， 2014]。我们可以在 Stable Baselines 模块中找到成熟的方法 [Raffin 等， 2021]。

2.3 形式化规范与STL

在形式化方法领域，形式化规范是对我们希望智能体完成的任务所作的明确描述。它们可以用多种形式语言表达，例如线性时序逻辑（LTL [Vardi,1996] 和信号时序逻辑（STL） [Raman 等，2014；Deshmukh 等， 2017]。本文聚焦于STL。STL公式可以使用以下语法定义：

$$\varphi ::= \top \mid f(\bar{s}) < 0 \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 U_{[t_1,t_2]} \varphi_2 \tag{2}$$

使得 $\top$ 是重言式，而 f 是一个将轨迹（在 STL 语境中等价地称为信号） $\bar{s}$ 映射为实数的函数。 U 是直到算子，例如， $\varphi_1 U_{[t_1,t_2]} \varphi_2$ 表示 φ_2 必须在某个时刻 $t \in [t_1, t_2]$ 成立，并且在此之前 φ_1必须始终成立。时间区间 $[t_1, t_2]$ 解释为 $\{t_1, t_1 + 1, \ldots, t_2\}$。基于这一语法结构，STL 已在最优控制领域中指导了控制综合 [Sadraddini and Belta, 2018年；Donz´e 等人，2015年；Raman 等人， 2014]。然而，我们必须承认，STL 和其他形式语言的表达能力存在一定局限，例如，它们无法表达一些人们常见的要求，如“尽可能快”。

2.4 规范引导的强化学习

强化学习领域的现有研究尝试了多种利用形式化规范来指导智能体训练的方法。具体而言，当前最先进的方法主要有两种。首先，一些方法构造乘积马尔可夫决策过程（MDP），将底层MDP与规范相结合，并将规范转换为有限状态机 [Fu and Topcu，2014；Ding 等， 2011]。另一种方法的保证性较低，但更加灵活，即基于形式化规范构造奖励函数。例如，定量语义是一个实值度量 ρ，用于衡量信号 $\bar{s}$ 在时间 t [上满足STL性质 φ 的程度Hamilton 等， 2022]。我们使用本文的符号重新陈述其定义如下。

$$\begin{aligned} &\rho(\bar{s}, t, (f(\bar{s}) < d)) = d - f(s_t) \\ &\rho(\bar{s}, t, \neg\varphi) = -\rho(\bar{s}, \varphi, t) \\ &\rho(\bar{s}, t, \varphi_1 \wedge \varphi_2) = \min(\rho(\bar{s}, t, \varphi_1), \rho(\bar{s}, t, \varphi_2)) \\ &\rho(\bar{s}, t, \varphi_1 \vee \varphi_2) = \max(\rho(\bar{s}, t, \varphi_1), \rho(\bar{s}, t, \varphi_2)) \\ &\rho(\bar{s}, t, F_{[t_1,t_2]}\varphi) = \max_{t' \in [t'+t_1, t'+t_2]} \rho(\bar{s}, t', \varphi) \\ &\rho(\bar{s}, t, G_{[t_1,t_2]}\varphi) = \min_{t' \in [t'+t_1, t'+t_2]} \rho(\bar{s}, t', \varphi) \\ &\rho(\bar{s}, t, \varphi_1 U_{[t_1,t_2]} \varphi_2) \\ &= \max_{t' \in [t'+t_1, t'+t_2]} \Big(\min(\rho(\bar{s}, t', \varphi_2), \min_{t'' \in [t, t')} \rho(\bar{s}, t'', \varphi_1)) \Big) \end{aligned} \tag{3}$$

在此，为方便起见，我们介绍 finally（F）和 globally（G）算子，它们可以使用 U 定义，即 $F_I\varphi = \top U_I \varphi$ 以及 $G_I\varphi = \neg(\top U_I \neg\varphi)$，其中 I 是一个时间区间。根据公式（3）的构造，我们能够评估当前状态截至目前的性质满足程度实值分数。因此，人们一直通过各种方式将该分数用于奖励函数，以指导强化学习 [Hamilton 等，2022年；Li 等，2017年；Li 等， 2018]。然而，由于这些指标源自 STL 满足度，因此在以与 STL 相同的方式描述某些任务时，表达能力仍然不足。对此，研究人员提出了复杂的奖励塑形技术，使该分数能够更准确地鼓励和抑制智能体行为[Berducci等， 2021]。然而，据我们所知，针对 ASAP 的表达研究仍然很少，而 ASAP 对下一小节介绍的 CPS恢复问题至关重要。

2.5 实时信息物理系统恢复

信息物理系统（CPS）是由软件控制、与物理世界交互的智能体（通常具有实时性）。例如，自动驾驶车辆、机器人、智能电网和医疗设备都属于 CPS。近年来，研究人员广泛讨论了一个问题，即 CPS 恢复。具体而言，当 CPS 发现自身处于

不期望状态时，如何引导其回到预先定义的安全目标状态集合?

研究人员讨论了这一问题的一些更困难的版本。例如，有些研究讨论了智能体传感器遭受攻击时的引导问题，此时观测状态不可靠 [Kong 等，2018年；Zhang 等，2020年；Zhang等， 2021]。然而，大多数研究假设已知环境动力学，因此可以通过最优控制算法综合恢复控制器。我们关注一种更严重的情况：环境动力学是未知的黑盒，但我们能够从中采样数据点，用于智能体训练。总体而言，我们的工作探讨了在这一受现实世界启发的场景中，如何弥补形式化规范表达能力的不足。

3 问题 формα化

我们的问题受到第2节所定义的实时CPS恢复问题的启发。也就是说，在某些场景中，我们希望一个当前未满足某些属性（例如ASAP）的智能体执行恢复。例如，在恢复问题中，我们希望智能体尽快到达预先定义的目标状态集。

具体而言，我们希望构建一个框架，该框架的输入为状态空间 $\mathcal{S}$、动作空间 $\mathcal{A}$ 和指定的 STL性质 φ。该框架输出一个控制策略 $\pi(a_t|s_t)$，即给定状态下动作的概率分布，使得满足 φ 的时间尽可能短。

给定某个 $k \geq 0$ 的轨迹 $\bar{s} = s_0 s_1 \dots s_k$，我们使用 $(\bar{s}, t) \models \varphi$，其中 $0 \leq t \leq k$ 表示状态 s_t 满足性质 φ。然后，我们定义函数 $t_\varphi(\cdot)$，用于计算信号首次满足某个性质的时间步，如下所示。

$$t_\varphi(\bar{s}) = \begin{cases} \infty, & \nexists t \in \{0, \dots, k\}, , (\bar{s}, t) \models \varphi \\ \arg\min_{t \in \{0,\dots,k\}}((\bar{s}, t) \models \varphi), & \text{otherwise.} \end{cases} \tag{4}$$

直观而言，如果不存在满足 φ 的时间步，则 $t_\varphi(s) = \infty$。否则，它就是首次满足的时间步。

我们称策略 π^* 为 ASAP 策略，是指在未知概率系统模型，即状态转移分布 $p(s_{t+1}|s_t, a_t)$ 下，该策略以更高的概率采样到一条在更短时间内满足给定性质的轨迹。

定义 1 (尽快策略)。 在状态空间 $\mathcal{S}$、动作空间 $\mathcal{A}$ 和底层系统模型 p上，如果对于从任意初始状态 s_0出发的任意两条轨迹 $\bar{s}$ 和 $\bar{s}'$，策略π^* 满足以下条件，则称其为 *ASAP*:

$$t_\varphi(\bar{s}) < t_\varphi(\bar{s}') \implies \Pr_{p,\pi^*}(\bar{s}'|s_0) < \Pr_{p,\pi^*}(\bar{s}|s_0) \tag{5}$$

我们提出一种基于以下假设寻找此类最优策略的方法。

假设1： 系统一旦到达某项性质，就始终能够维持该性质。当系统满足给定性质 φ 时，通过使用某些控制输入，它以最高概率持续满足该性质。请注意，这一假设是合理的，因为恢复问题中的目标集合通常是稳定平衡点 [Abad 等，2016；Abdi等， 2018]的邻域。一旦进入该邻域，智能体的行为将趋于稳定，并且我们可以假设未来具有恒定期望累积奖励，即对于任意策略 π，在任意轨迹 $\bar{s}'$ 上，该轨迹持续

在某个 s 上，

$$(|\bar{s}| = t) \wedge ((\bar{s}, t) \models \varphi) \implies \mathbb{E}_\pi[\sum_{t'=t+1}^{\infty} \gamma^{t'} r(s'_{t'}) | \bar{s}'_{0:t} = \bar{s}] = M \tag{6}$$

其中， M 是一个常数， $\bar{s}'_{0:t}$ 表示 $\bar{s}'$从时间步 0 到 t 的子轨迹。换言之，如果 $\bar{s}$ 在其最后一个时间步满足该性质，那么由于稳定性，所有从其延伸出的轨迹此后都具有固定的期望累积奖励。

假设2： 轨迹的长度和语义分数始终有界。 首先，我们的强化学习框架仅考虑长度有界的系统轨迹。我们将其界记为 k_{min} 和 k_{max}。 其次，我们假设所有系统轨迹的定量语义分数都有界。我们将其界记为 ρ_{min} 和 ρ_{max}。

假设3： 黑盒（未知）系统执行为马尔可夫决策过程。具体而言，系统执行的任何动作都仅基于当前状态。因此，系统生成的所有轨迹均具有彼此独立的概率。

我们旨在解决以下两个问题:

问题1： 给定一个黑盒系统及其状态空间 $\mathcal{S}$ 和动作空间 $\mathcal{A}$。我们如何学习关于STL性质 φ的ASAP策略 π^*？

问题2： 我们如何利用通过解决问题1获得的ASAP策略，并使用它解决系统恢复问题?

4 ASAP-Phi框架

4.1 工作流程

针对问题1，我们提出了以下框架，即 ASAP-Phi。该框架包括生成分段奖励函数，并在基于演员-评论家的无模型强化学习算法上进行训练，同时训练需要从底层转移分布 $p(s_{t+1}|s_t, a_t)$ 中进行采样。

首先，我们设计了如下确定性分段奖励函数r_{asap}。

$$r_{\text{asap}}(\bar{s}, t, \varphi) = \begin{cases} \rho(\bar{s}, t, \varphi), & \text{if } (\bar{s}, t) \not\models \varphi \\ r_{\text{sat}} >> \rho_{\max}, & \text{otherwise} \end{cases} \tag{7}$$

公式（7）定义了一个奖励函数，用于根据 STL性质 φ 和最大值时间索引 t 评估轨迹 $\bar{s}$。该函数将奖励分为两种情况：如果信号直到 t 都不满足给定性质，则使用第 2.4 节定义的定量语义分数 ρ 来评估性质满足程度；否则，选取一个大常数 r_{sat}，该常数大于前一种情况下所有可能的奖励。请注意，为了计算该奖励，我们需要保留完整轨迹 $\bar{s}$，以检查性质满足情况。下一小节将讨论这一大常数。

通常，定量语义会产生稀疏奖励，因为它只在回合结束时计算鲁棒性程度，这也称为稀疏奖励 [Hamilton等， 2022]。回报是整个回合的鲁棒性程度。另一种选择是，定量语义也可以采用稠密奖励，即在每个时间步计算鲁棒性程度。稠密奖励的回报表示

从起点到当前时间步的子轨迹的鲁棒性程度。本文提出了稠密奖励的一种变体：有限稠密奖励，用于评估轨迹中最近 d 个时间步的鲁棒性程度， d 是一个超参数。

我们在算法 1 中介绍我们的强化学习框架。主循环会迭代生成长度处于特定范围内的随机轨迹。通过调用外部基于演员-评论家的算法（记为 AC），逐步更新由 θ 参数化的策略（演员）模型 π_θ 以及由 w 参数化的价值（评论家）模型 v_w。该算法可以是软演员-评论家（SAC）[Haarnoja 等人， 2018] 或深度确定性策略梯度（DDPG）[Silver 等人， 2014]。

算法 1ASAP-Phi（同策略版本）

输入：奖励函数 r_{asap}、状态空间 $\mathcal{S}$、动作空间 $\mathcal{A}$、STL性质 φ、演员-评论家训练算法 AC、系统模型 p（该模型未知但允许采样）、训练样本数量 m、回合长度的下界和上界 $k_{\min}$ 以及 $k_{\max}$

输出：控制策略 π_θ1：初始化策略模型 π_θ 和价值模型 v_w2： $m_{train} \leftarrow 0$3：当 $m_{train} < m$时执行 4：从 $[k_{\min}, k_{\max}]$中均匀采样$k \leftarrow$ 5：从 $\mathcal{S}$中均匀采样$s_0 \leftarrow$ 6：缓存信号 $\bar{s} \leftarrow [s_0]$ 7：对于 $t = 0, \ldots, k$ 执行 8：$r \leftarrow r_{\text{asap}}(\bar{s}, t, \varphi)$ 9：从 $\pi_\theta(\cdot|s_t)$中采样 $a_t \leftarrow$ 10：从 $p(\cdot|s_t, a_t)$中采样 $s_{t+1} \leftarrow$ 11：将 s_{t+1} 追加到 $\bar{s}$ 12：$\pi_\theta, v_w \leftarrow AC(s_t, a_t, s_{t+1}, r)$ 13：$m_{train} \leftarrow m_{train} + 1$14：结束 for 15：结束 while

算法 1 在 m 个数据点上训练策略和价值模型，这些数据点组合成多个长度介于 $k_{\min}$ 和 $k_{\max}$之间的轨迹回合。具体而言，第 3 行至第 14 行的外层循环表示针对所有回合的完整过程，而第 7 行至第 13 行的内层循环表示一个回合。训练发生在第 11 行，演员-评论家算法使用 $\bar{s}_t$、 a_t 和 $\bar{s}_{t+1}$，基于我们设计的奖励函数 r_{asap}更新这两个模型。请注意，该版本的算法是同策略的。也就是说，每当有新的数据点到达时，模型就会更新一次。我们还可以通过在每个回合结束时执行模型更新来利用异策略训练。

4.2 正确性证明

在本小节中，我们证明 ASAP-Phi 的正确性，即如果 ASAP-Phi 返回一个最优策略，那么该策略也是 ASAP 策略。

定理 2 (ASAP-Phi正确性)。 从 *ASAP-Phi* 框架中学习到的最优策略是一个 *ASAP* 策略。

证明：我们通过反证法证明该定理。假设存在一个由 ASAP-Phi 学习得到的最优策略 π^*，且该策略不是 ASAP。我们旨在证明 也不是最优策略，这与我们的假设相矛盾。

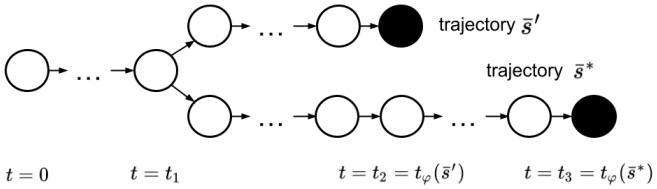


图1：在假设的 π^* 下存在的两条轨迹。空心圆表示截至目前的轨迹不满足 φ，实心圆表示满足。

，这与我们的假设相矛盾。

由于 π^* 是最优的但不是 ASAP，因此存在两条轨迹 $\bar{s}^*$ 和 $\bar{s}'$，它们都从一个初始状态 s_0 出发，并以到达目标状态作为最后状态。我们还知道 $\Pr_{p,\pi^*}(\bar{s}'|s_0) < \Pr_{p,\pi^*}(\bar{s}^*|s_0)$，并且根据假设， $|\bar{s}'| < |\bar{s}^*|$，即 $\bar{s}^*$ 比 $\bar{s}'$ 长，但使用 π^* 时，前者被执行的概率高于后者。图1展示了这两条轨迹。这里， t_1 是这两条轨迹自开始以来完全相同的最后时刻； t_2 是 $\bar{s}'$ 首次满足 φ 的时刻，而 t_3 是 $\bar{s}^*$ 首次满足的时刻。

根据假设1，轨迹到达目标集合后可以继续保持在其中。因此，我们在 $\bar{s}'$ 之后填入满足条件的状态，直至时间 t_3，从而构造一条轨迹。将这条轨迹记为 $\bar{s}''$。于是有，$\Pr_{p,\ \pi^*}(\bar{s}''|s_0) \leq \Pr_{p,\ \pi^*}(\bar{s}'|s_0) < \Pr_{p,\ \pi^*}(\bar{s}^*|s_0)$，因为 $\bar{s}''$ 是 $\bar{s}'$ 的扩展。

由于 v^* 也是最优的，因此它是贝尔曼方程 (1) 的一个解:

$$v^*_\pi(s_0) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_{\text{asap}}(\bar{s}, t, \varphi)|s_0] = \mathbb{E}_\pi[\sum_{t=0}^{t_3} \gamma^t r_{\text{asap}}(\bar{s}, t, \varphi)|s_0] + \underbrace{\mathbb{E}_\pi[\sum_{t=t_3+1}^{\infty} \gamma^t r_{\text{asap}}(\bar{s}, t, \varphi)|s_0]}_{\text{denote this term } D_\pi} \tag{8}$$

因此

$$\begin{aligned} v^*_{\pi^*}(s_0) &= \mathbb{E}_{\pi^*}[\sum_{t=0}^{t_3} \gamma^t r_{\text{asap}}(\bar{s}, t, \varphi)|s_0] + D_{\pi^*} \\ &= p_1 \underbrace{\sum_{t=0}^{t_3} \gamma^t r_{\text{asap}}(\bar{s}^*, t, \varphi)}_{\text{denote this term } A} + p_2 \underbrace{\sum_{t=0}^{t_3} \gamma^t r_{\text{asap}}(\bar{s}'', t, \varphi)}_{\text{denote this term } B} \\ &\quad + \underbrace{\mathbb{E}_{\pi^*}[\sum_{t=0}^{t_3} \gamma^t r_{\text{asap}}(\bar{s}, t, \varphi)|s_0, \bar{s} \neq \bar{s}^* \wedge \bar{s} \neq \bar{s}'' \wedge |\bar{s}| = t_3)]}_{\text{denote this term } C_{\pi^*}} + D_{\pi^*} \\ &= p_1 A + p_2 B + C_{\pi^*} + D_{\pi^*} \end{aligned} \tag{9}$$

其中 $p_1 = \Pr_{p,\ \pi^*}(\bar{s}^*|s_0)$、 $p_2 = \Pr_{p,\ \pi^*}(\bar{s}''|s_0)$。也就是说，分别取轨迹 $\bar{s}^*$ 和 $\bar{s}''$的概率。术语C_{π^*} 表示在策略 π^*下，从 s_0 出发，采取任意其他长度为 t_3 的轨迹所能获得的价值。

我们现在通过交换两条轨迹上的概率 p_1 和 p_2，构造另一种策略 π'。由于假设3指出，马尔可夫状态转移使所有轨迹彼此独立，因此我们可以这样做。于是得到以下新的价值估计：

$$v^*_{\pi'}(s_0) = p_2 A + p_1 B + C_{\pi'} + D_{\pi'} \tag{10}$$

由于长度为 t_3 的所有轨迹中，除 $\bar{s}^*$ 和 $\bar{s}''$ 之外的轨迹在 π^* 和 π'中的概率均相同，因此$C_{\pi'} = C_{\pi^*}$。此外，根据假设1，由于 $\bar{s}''$ 和 $\bar{s}^*$ 均在时间 t_3进入稳定邻域，我们可以利用公式（6）得到

$$\begin{aligned}
D_{\pi'} &= \mathbb{E}_\pi[\sum_{t=t_3+1}^{\infty} \gamma^t r_{\text{asap}}(\bar{s}, t, \varphi) | s_0] \\
&= \mathbb{E}_\pi[\sum_{t=t_3+1}^{\infty} \gamma^t r_{\text{asap}}(\bar{s}, t, \varphi) | s_0, \bar{s}_{0:t_3} = \bar{s}^*] \\
&+ \mathbb{E}_\pi[\sum_{t=t_3+1}^{\infty} \gamma^t r_{\text{asap}}(\bar{s}, t, \varphi) | s_0, \bar{s}_{0:t_3} = \bar{s}''] \\
&+ \mathbb{E}_\pi[\sum_{t=t_3+1}^{\infty} \gamma^t r_{\text{asap}}(\bar{s}, t, \varphi) | s_0, \bar{s} \neq \bar{s}^* \wedge \bar{s} \neq \bar{s}''] \\
&= 2M + \mathbb{E}_\pi[\sum_{t=t_3+1}^{\infty} \gamma^t r_{\text{asap}}(\bar{s}, t, \varphi) | s_0, \bar{s}_{0:t_3} \neq \bar{s}^* \wedge \bar{s}_{0:t_3} \neq \bar{s}''] \\
&= D_{\pi^*}
\end{aligned} \tag{11}$$

其中，$\bar{s}_{0:t_3}$ 表示轨迹 $\bar{s}$从时间步0到 t_3 的子轨迹。因此，我们还可以得到 $D_{\pi'} = D_{\pi^*}$。那么我们只需比较 $p_1 A + p_2 B$ 和 $p_2 A + p_2 B$。

根据（7）中r_{asap} 的定义，我们有

$$\begin{aligned}
A &= \underbrace{\sum_{t=0}^{t_1} \gamma^t \rho(\bar{s}^*, t, \varphi)}_{A_1} + \underbrace{\sum_{t=t_1+1}^{t_2} \gamma^t \rho(\bar{s}^*, t, \varphi)}_{A_2} + \underbrace{\sum_{t=t_2+1}^{t_3} \gamma^t \rho(\bar{s}^*, t, \varphi)}_{A_3} \\
B &= \underbrace{\sum_{t=0}^{t_1} \gamma^t \rho(\bar{s}'', t, \varphi)}_{B_1} + \underbrace{\sum_{t=t_1+1}^{t_2} \gamma^t \rho(\bar{s}'', t, \varphi)}_{B_2} + \underbrace{\sum_{t=t_2+1}^{t_3} \gamma^t r_{\text{sat}}}_{B_3}
\end{aligned} \tag{12}$$

由于两条轨迹 $\bar{s}^*$ 和 $\bar{s}''$ 从 $t = 1$ 到$t = t_1$完全相同，因此我们有 $A_1 = B_1$。此外，由于 $r_{\text{sat}} > \rho_{\max}$，我们有 $A_3 < B_3$。对于最后一对 A_2 和 B_2，我们有

$$\begin{aligned}
A_2 - B_2 &= \sum_{t=t_1+1}^{t_2} \gamma_t (\rho(\bar{s}^*, t, \varphi) - \rho(\bar{s}'', t, \varphi)) \\
&\leq \sum_{t=t_1+1}^{t_2} \gamma_t (\rho_{\max} - \rho_{\min}) \leq (\rho_{\max} - \rho_{\min}) \sum_{t=0}^{k_{\max}} \gamma_t \\
&\leq (\rho_{\max} - \rho_{\min}) k_{\max}
\end{aligned} \tag{13}$$

因此，如果我们设定 $r_{\text{sat}} > \rho_{max} + (\rho_{\max} - \rho_{\min}) k_{\max}$，就可以保证 $A_2 - B_2 < B_3 - A_3$。于是，我们有$A < B$。回想一下，$p_1 > p_2$，因此 $p_2 A + p_1 B > p_1 A + p_2 B$，即 $v^*_{\pi'}(s_0) > v^*_{\pi^*}(s_0)$。因此，$\pi'$ 优于 π^*，或者说 π^*不是最优策略。 □

4.3 ASAP-Phi 在恢复问题中的应用

ASAP-Phi 是一种通用解决方案，可以将系统从偏离状态（不满足某项性质）引导至目标状态（满足该性质）。虽然我们的解决方案并非专门针对安全性，但它能够有效增强系统抵御故障和攻击的韧性。针对特定的故障或攻击场景，可能需要对我们的解决方案进行某些调整。作为一个用例，我们提出如下使用 ASAP-Phi 的方法，以解决 CPS恢复问题并回答问题2。

在恢复问题中，我们有一个状态目标集合$\mathcal{S}_T \subset \mathcal{S}$，希望智能体尽快到达该集合。可选地，还可能存在某些状态 $\mathcal{S}_O \subset \mathcal{S}$，它们表示我们希望避开的障碍或危险区域，$\mathcal{S}_T \cap \mathcal{S}_0 = \emptyset$。假设目标集合和障碍集都是有界的，因此我们可以高效地检查状态 s_t 是否包含在其中。因此，我们使用 STL性质来描述这两个要求：

$$\varphi_T = s_t \in \mathcal{S}_T, \quad \varphi_O = s_t \notin \mathcal{S}_O \tag{14}$$

换句话说，我们希望学习一个满足φ_T ASAP且始终满足 φ_O的策略。因此，我们需要将 φ_T 的鲁棒性程度作为训练的奖励。此外，我们还需要进行奖励塑形，以确保最优策略能够满足 φ_O。为此，我们采用算法 1中的 ASAP-Phi 训练：在 φ_T 上进行常规的策略和价值函数更新，并在采样轨迹违反 φ_O时提前停止。为方便起见，我们将每个回合的 ASAP-Phi 初始化封装为一个规范子程序 $ASAP\ phi\ ep\ init$()（算法 1的第 4-5 行），并将内循环中的每个更新步骤封装为 $ASAL\ phi\ step$(φ)（算法 1的第 8-12 行），其奖励函数基于性质φ进行设计。该方案在算法 2中给出。

算法 2用于恢复的 ASAP-Phi

输入：属性 φ_T、 φ_O 以及其他所需内容
通过算法 1
输出：恢复控制策略 π_θ

```
1: 初始化策略模型 π_θ 和价值模型 v_w
2: m_train ← 0
3: 当 m_train < m时
4:     ASAP_phi_ep_init()
5:     对于 t = 0, ..., |s̄| − 1 执行
6:         如果 (s̄, t) ⊭ φ_O 则
7:             中断
8:         结束如果
9:         π_θ, v_w ← ASAL_phi_step(φ_T)
10:        m_train ← m_train + 1
11:    结束 for 循环
12: 结束 while 循环
```

请注意，关键思想是搜索一个能够尽快满足 φ_T并在违反 φ_O时提前停止的最优策略。这样做的原因是防止轨迹在进入障碍状态时获得高奖励 r_{sat}，从而使智能体学会采取替代路径，尽快到达目标状态。我们在实验部分展示了该方法的有效性。

5 实验评估

5.1 设置

我们针对恢复问题设置了多组实验进行评估。实验在四个基准测试上实现：直流电机位置、自行车、姿态控制和 Mujoco 游泳者。直流电机位置是控制领域和 CPS 领域中的经典系统，目标是通过直流电 [Zhang等， 2020] 控制电机的旋转角。自行车基准测试模拟了具有两个控制输入的自行车动力学：转向角和加速度 [Kong 等， 2015], 目标是使自行车以不同的速度和方向行驶。姿态控制基准测试来自连续与混合系统应用验证研讨会（ ARCH-COMP'22）的人工智能类别 [Lopez 等， 2022], 目标是控制刚体以避开不安全区域。最后，Mujoco 游泳者基准测试是强化学习中广为人知的 Mujoco 环境之一 [Brockman 等，2016；Todorov 等， 2012], 旨在控制双关节系统并尽可能快速地向右移动。附录中给出了这些基准测试的更详细描述。尽管其中一些基准测试具有常微分方程等形式化模型，但我们的方法只将它们的模拟器作为黑箱使用。

我们选择了经典的演员-评论家训练算法，包括 SAC、DDPG、A2C、TD3 和 PPO。为避免这些算法的实现导致性能下降，我们采用了由 OpenAI [Raffin等人开发和维护的 stable baselines 3 实现， 2021]。对于每个基准测试，都给定一个目标集合和一个不安全集。为简化起见，这两个集合均定义为球形区域。我们训练智能体寻找策略，使系统尽快（ASAP）到达目标集合，同时不触碰不安全集。

尽管我们已在第 4.2 节中给出证明，但通常很难通过训练找到最优策略，因此我们寻求次优策略，即在选定的最大时间容差内完成恢复；这与先前工作中恢复任务采用的评估指标类似 [Zhang 等人，2020年；Zhang 等人，2021年；Kong 等人， 2018]。我们考虑两种类型的恢复任务：$Reach$和 $Reach$ &$Avoid$。对于 $Reach$ 任务，不存在不安全的障碍状态。对于 $Reach$&$Avoid$ 任务，系统应避开预定义的障碍集，如第 4.3 节所述。我们在 1000 个随机初始点上测试训练好的智能体，并统计在最大时间容差内能够到达目标集合的轨迹数量。如果轨迹触碰障碍物，就会终止，并在 $Reach$&$Avoid$ 任务中被视为失败。我们计算策略生成轨迹的成功率，以评估其性能。

我们将 ASAP-Phi 与其他基线奖励的性能进行比较，结果如下。此外，我们还对 ASAP-Phi 进行了消融实验，以展示 ASAP-Phi 对训练算法的不敏感性。有关实验的更多细节，请参见附录。

5.2 主要结果

我们选择了两个奖励函数基线方法与 ASAP-Phi 进行比较。其中一个是第 2.4 节重新介绍的定量语义，它表示给定

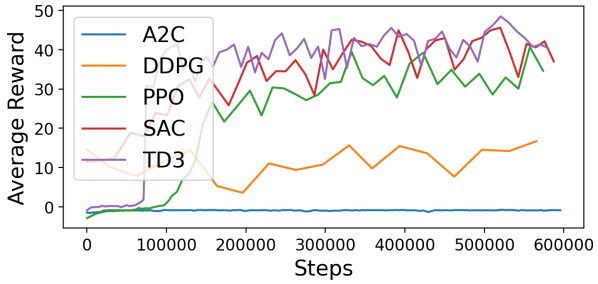


图 2：Swimmer 在 $Reach$&$Avoid$ 任务上的训练历史

STL 规范 [Hamilton 等， 2022]。我们将该奖励记为 ρ-奖励。另一个奖励函数基线方法是启发式距离奖励函数，该方法广泛应用于已有工作中 [Burchfiel 等，2016；Trott 等，2019；Li 等， 2017], 。为避免符号滥用，我们将其重述如下。

$$r_d = -\lambda_T d_T + \lambda_O d_O + r_{base} \tag{15}$$

其中， d_T和 d_O 分别是到目标集合 $\mathcal{S}_T$和不安全障碍集合 $\mathcal{S}_O$的距离， λ_T、 λ_O 和 r_{base} 是超参数。

我们使用 SAC 训练智能体以获得主要结果。从表 1 可以观察到三个主要结论：

1. ASAP-Phi 在大多数情况下优于基线方法。可以看到，在所有情况下，ASAP-Phi 的成功率都超过 50%，并且在大多数试验中击败了两种基线方法。仅在少数情况下，例如自行车基准测试中的 $Reach$&$Avoid$ 任务， ρ奖励的表现略优于 ASAP-Phi。这一微小差异可能源于训练过程中的噪声，因为无法保证能够找到最优策略。

2. ASAP-Phi 的性能比另外两种基线方法稳定得多。例如，当最大时间容差为 15 个时间步时， d-奖励基线在大多数任务上的表现都远逊于其他方法，但在直流电机位置的 $Reach$&$Avoid$ 任务上略优于其他方法。ASAP-Phi 不会出现这种不稳定性。

3. 有时，智能体在 $Reach$&$Avoid$ 任务上的表现可能优于在 $Reach$ 任务上的表现。直观来看，$Reach$&$Avoid$ 任务比 $Reach$ 任务更难。然而，通过训练很难找到最优策略，因此不同任务之间可能存在性能差异。

由于 ASAP-Phi 在大多数情况下都优于基线方法，因此我们认为，ASAP-Phi 在实践中能够有效地实现给定的 ASAP 规范。

5.3 消融实验

我们针对以下两方面开展消融实验：(1) 不同的演员-评论家算法，以及 (2) 除当前奖励函数之外，用于促进 ASAP 的不同方式。这里，我们在表2中展示前者的结果，并得出两个主要观察结果。

1. ASAP-Phi 并非针对某一种特定的训练算法而设计。换言之，表2 展示了其对训练算法的不敏感性。相反，ASAP-Phi

Max 时间容差*	方法	直流电机位置		自行车		姿态控制		游泳者	
		R	*R&A*	*R*	*R&A*	*R*	*R&A*	*R*	*R&A*
30	ASAP-Phi	**97.4%**	**97.8%**	**92.8%**	**91.0%**	**93.9%**	**92.0%**	**62.6%**	**64.1%**
	ρ-奖励	83.5%	82.1%	80.6%	86.4%	75.1%	73.6%	58.4%	57.2%
	d-奖励	35.5%	95.3%	78.4%	27.6%	21.1%	12.7%	42.5%	6.5%
25	ASAP-phi	**95.2%**	**95.3%**	**88.1%**	**89.2%**	**86.5%**	**85.3%**	**57.4%**	**59.3%**
	ρ-奖励	77.3%	79.8%	80.6%	86.1%	72.4%	70.9%	51.5%	50.8%
	d-奖励	32.2%	95.1%	76.2%	27.0%	18.0%	11.9%	39.4%	6.1%
15	ASAP-phi	**89.0%**	84.9%	**74.4%**	65.1%	**55.7%**	**52.0%**	**52.0%**	**53.0%**
	ρ-奖励	59.2%	79.4%	72.6%	**65.5%**	52.2%	51.9%	47.4%	46.0%
	d-奖励	29.2%	**87.0%**	65.4%	25.3%	15.6%	10.7%	34.1%	4.8%

表 1：通过成功恢复率衡量的智能体性能。此处，R 和 $R\&A$ 分别表示 $Reach$ 和 $Reach\&Avoid$任务。
*：对于游泳者，最大时间容差为 50、40、30 个时间步，而不是 30、25、15 个时间步。

	直流电机位置		自行车		姿态控制		游泳者	
	到达	到达与避障	到达	到达与避障	到达	到达与避障	到达	到达与避障
SAC	97.4%	**97.8%**	**83.0%**	78.8%	75.6%	74.1%	62.6%	**64.1%**
DDPG	**99.6%**	95.7%	75.8%	78.2%	62.7%	61.6%	11.5%	11.8%
A2C	56.8%	39.3%	37.8%	22.8%	15.2%	14.3%	0.6%	0.4%
TD3	99.0%	96.3%	76.8%	**81.4%**	**78.4%**	**77.6%**	**62.8%**	54.1%
PPO	37.8%	39.6%	79.9%	80.2%	65.0%	64.8%	15.2%	13.1%

表2：不同算法训练出的策略所生成轨迹的成功率。直流电机位置的最大时间容差设为30个时间步，自行车和姿态控制的最大时间容差设为20个时间步，Swimmer基准测试的最大时间容差设为50个时间步。

兼容各种基于演员-评论家的方法。例如，在测试的五种算法中，SAC 和 TD3 的表现最佳。如图2所示，DDPG 在若干任务上也表现良好。

2. PPO和A2C的性能不稳定。例如，PPO在Swimmer基准测试上的表现较差，而A2C总体上在这两项任务中的表现都逊于其他算法。

在针对鼓励ASAP的不同方式开展的第二项消融研究中，我们发现当前的奖励表现最佳。有关结果及其他细节，请参阅附录。

6 讨论

ASAP-Phi 的优势。通过第 4.2 节的分析，我们证明 ASAP-Phi 仅使用采样训练数据，而不依赖系统动力学知识，寻求一种能够最大化选择满足给定形式化规范 ASAP 的轨迹概率的最优策略。据我们所知，尽管在 CPS 恢复等各种应用场景中确有必要，但现有研究尚未提出此类框架。此外，尽管理论上的最优策略通常难以在实际实现中实现，我们在第5节的实验表明，ASAP-Phi 能够取得较高的成功率，能够在截止时间内恢复智能体，成功率最高达 97%，并且总体上优于最先进的基线方法。

ASAP-Phi 的局限性。 然而，该方法确实存在一些局限性。首先，ASAP-Phi 基于无模型强化学习，这意味着它自然继承了此类学习算法的缺点。例如，它需要收集足够的训练数据，并调整学习超参数。幸运的是，研究人员通过设计稳定的演员-评论家骨干网络，在一定程度上解决了这一问题 [Raffin 等人， 2021]。其次，尽管我们探索了鼓励ASAP的各种方式，奖励函数7未必是最佳方法。因此，如何鼓励ASAP策略的学习仍然是一个开放问题。最后，该框架做出了若干假设。尽管第3节中的假设1至3在当前语境下是合理的，但目前尚不清楚如何在仍能获得ASAP策略的同时消除这些假设。系统动力学虽然是一个黑箱，却被假定为静态的，并且允许任意数量的数据采样，这又是一个有待消除的假设。我们认为，这些问题可以作为未来研究的课题。

7 结论

受未知系统动力学下实时CPS恢复问题的启发，我们提出了ASAP-Phi，一种模型

无关的强化学习框架，旨在鼓励尽快满足STL形式化规范。我们证明了ASAP-Phi学习到的最优策略确实能够实现ASAP，并将其应用回恢复这一启发性案例研究中。实验表明，由ASAP-Phi训练的控制策略在最大时间容差内始终能够实现较高的恢复成功率，并且总体上在最先进的基准测试中击败了基线方法。

参考文献

[Abad等人， 2016] Fardin Abdi Taghi Abad、Renato Mancuso、Stanley Bak、Or Dantsker 和 Marco Caccamo。具有时序安全约束的实时信息物理系统的基于重置的恢复。载于*2016*年*IEEE*第*21*届新兴技术与工厂自动化国际会议（*ETFA*），第1–8页。IEEE，2016年。

[Abdi等人， 2018] Fardin Abdi、Chien-Ying Chen、Monowar Hasan、Songran Liu、Sibin Mohan 和 Marco Caccamo。通过基于重启的设计保障信息物理系统的物理安全。载于*2018*年*ACM/IEEE*第*9*届信息物理系统国际会议（*ICCPS*），第10–21页。IEEE，2018年。

[Berducci等人， 2021] Luigi Berducci、Edgar A Aguilar、Dejan Ni˘ckovi´c 和 Radu Grosu。基于任务规范的分层势函数奖励塑形。*arXiv*电子预印本，页码 arXiv–2110，2021年。

[Brockman 等， 2016] Greg Brockman、Vicki Cheung、Ludwig Pettersson、Jonas Schneider、John Schulman、Jie Tang 和 WojciechZaremba。OpenAI Gym。*arXiv* 预印本 *arXiv:1606.01540*，2016。

[Burchfiel 等， 2016] Benjamin Burchfiel、Carlo To masi 和 Ronald Parr。通过带评分轨迹进行奖励学习的距离最小化。在*AAAI*人工智能会议论文集，第 30 卷，2016 年。

[Chang 等， 2019] Ya-Chien Chang、Nima Roohi 和 Si-cun Gao。神经李雅普诺夫控制。神经信息处理系统进展，32，2019 年。

[Degris 等， 2012] Thomas Degris、Patrick M Pilarski 和 Richard SSutton。在实践中使用连续动作的无模型强化学习。在*2012*年美国控制会议（*ACC*），第 2177–2182 页。IEEE，2012 年。

[Deshmukh 等， 2017] JyotirmoyV Deshmukh、Alexandre Donz´e、Shromona Ghosh、Xiaoqing Jin、Garvit Juniwal 和 Sanjit A Seshia。信号时序逻辑的鲁棒在线监测。《系统设计中的形式化方法》，51(1):5–30，2017 年。

[Ding 等， 2011] Xu Chu Ding、Stephen L Smith、Calin Belta 和 Daniela Rus。时序逻辑约束下的 MDP最优控制。载于 *2011* 年第 *50* 届 *IEEE*决策与控制会议及欧洲控制会议，第 532–538 页。IEEE，2011 年。

[Donz´e等， 2015] Alexandre Donz´e、Vasumathi Raman、G Frehse 和MAlthoff。BLuSTL：从信号时序逻辑规范综合控制器。*ARCH@ CPSWeek*，34:160–8，2015年。

[Fu and Topcu， 2014] Jie Fu 和 Ufuk Topcu。带有时序逻辑约束的可能近似正确的马尔可夫决策过程学习与控制。*arXiv* 预印本 *arXiv:1404.7073*，2014年。

[Haarnoja 等， 2018] Tuomas Haarnoja、Aurick Zhou、Kristian Hartikainen、George Tucker、Sehoon Ha、Jie Tan、Vikash Kumar、Henry Zhu、Abhishek Gupta、Pieter Abbeel 等。软演员-评论家算法及其应用。*arXiv* 预印本 *arXiv:1812.05905*，2018年。

[Hamilton 等， 2022] Nathaniel Hamilton、Preston K Robi- nette 和 Taylor T Johnson。使用强化学习训练智能体以满足有时限和无时限信号时序逻辑规范。载于国际软件工程与形式化方法会议，第190–206页。Springer，2022年。

[Kong 等， 2015] Jason Kong、Mark Pfeiffer、Georg Schildbach 和 Francesco Borrelli。用于自主驾驶控制设计的运动学与动力学车辆模型。载于 *2015* 年 *IEEE*智能车辆研讨会（*IV*），第 1094–1099 页。IEEE，2015 年。

[Kong 等， 2018] Fanxin Kong、Meng Xu、James Weimer、Oleg Sokolsky 和 Insup Lee。信息物理系统检查点与恢复。载于 *2018*年 *ACM/IEEE* 第*9*届信息物理系统国际会议（*ICCPS*），第22–31页。IEEE，2018年。

[Li等， 2017] Xiao Li、Cristian-Ioan Vasile 和 Calin Belta。带有时序逻辑奖励的强化学习。载于 *2017*年 *IEEE/RSJ*智能机器人与系统国际会议（*IROS*），第3834–3839页。IEEE，2017年。

[Li等， 2018] Xiao Li、Yao Ma 和 Calin Belta。面向时序逻辑指定的强化学习任务的策略搜索方法。载于 *2018*年美国控制会议（*ACC*），第240–245页。IEEE，2018年。

[Ljung， 1998] Lennart Ljung。系统辨识。载于信号分析与预测，第163–173页。Springer，1998年。

[Lopez等， 2022] Diego Manzanas Lopez、Matthias Al- t hoff、Luis Benet、Xin Chen、Jiameng Fan、Marcelo Forets、Chao Huang、Taylor T Johnson、Tobias Ladner、Wenchao Li 等。Arch-comp22 类别报告：连续和混合系统对象的人工智能与神经网络控制系统（ainncs）。载于第九届应用国际研讨会论文集，第90卷，第142–184页，2022年。

[Ma等， 2021] Rui Ma、Sagnik Basumallik、Sara Eftekharnejad 和 Fanxin Kong。在存在潜在虚假数据的情况下，采用数据驱动模型预测控制缓解热过载。*IEEE*工业应用汇刊，57(2)：1872–1881，2021年。

[Peters等， 2005] Jan Peters、Sethu Vijayakumar 和 Stefan Schaal。自然演员-评论家。载于欧洲机器学习会议，第280–291页。Springer，2005年。

[Qin 等， 2022] Zengyi Qin、Dawei Sun 和 Chuchu Fan。Sablas：为黑箱动力

学系统学习安全控制。IEEE机器人与自动化快报，7(2)：1928–1935，2022年。

[Raffin等， 2021] Antonin Raffin、Ashley Hill、Adam Gleave、AnssiKanervisto、Maximilian Ernestus 和 Noah Dormann。《Stable-baselines3：可靠的强化学习实现》。机器学习研究期刊，22(268)：1–8，2021年。

[Raman等， 2014] Vasumathi Raman、Alexandre Donz´e、Mehdi Maasoumy、Richard M Murray、Al- berto Sangiovanni-Vincentelli 和 Sanjit A Seshia。《基于信号时序逻辑规范的模型预测控制》。载于第*53*届*IEEE*决策与控制会议，第81–87页。IEEE，2014年。

[Sadraddini 和 Belta， 2018] Sadra Sadraddini 和 Calin Belta。《数据驱动模型辨识与控制综合中的形式化保证》。载于第*21*届国际混合系统：计算与控制会议论文集（*CPS*周的一部分），第147–156页，2018年。

[Silver 等， 2014] David Silver、Guy Lever、Nicolas Heess、Thomas Degris、Daan Wierstra 和 Martin Riedmiller。确定性策略梯度算法。载于 机器学习国际会议，第 387–395 页。PMLR，2014 年。

[Todorov 等， 2012] Emanuel Todorov、Tom Erez 和 Yu- val Tassa。Mujoco：用于基于模型控制的物理引擎。载于 *2012 IEEE/RSJ* 智能机器人与系统国际会议，第 5026–5033 页。IEEE，2012 年。

[Trott等， 2019] Alexander Trott、Stephan Zheng、Caim- ing Xiong 和 Richard Socher。保持距离：使用自平衡塑形奖励解决稀疏奖励任务。神经信息处理系统进展，32，2019 年。

[Vardi， 1996] Moshe Y Vardi。基于自动机理论的线性时序逻辑方法。并发逻辑，第238–266页，1996 年。

[Zhang 等人， 2020] LinZhang、XinChen、Fanxin Kong 和 Alvaro A Cardenas。《使用线性近似对信息物理系统进行实时攻击恢复》。载于*2020*年 *IEEE*实时系统研讨会（*RTSS*），第 205–217 页。IEEE，2020年。

[Zhang 等， 2021] Lin Zhang、PengyuanLu、Fanxin Kong、Xin Chen、Oleg Sokolsky 和 Insup Lee。使用线性二次调节器实现信息物理系统的实时攻击恢复。*ACM*嵌入式计算系统汇刊（*TECS*），20(5s)：1–24，2021年。

附录A 详细实验设置

本节将展示并解释实验设置的详细信息，包括基准测试信息和框架的超参数。

A.1 基准测试

如表 5 所示，我们已在四个基准测试上实现了该框架。这四个基准测试的状态空间维度各不相同，范围为 3 到 10；动作空间维度则为 1 到 3。根据训练算法的要求，它们全都是连续系统。除直流电机位置基准测试外，其余都是非线性系统。

直流电机位置基准测试的状态包括旋转角、角速度和电流。该基准测试唯一的控制输入是电压。自行车基准测试的状态包括位置坐标、方向和速度。自行车基准测试的控制输入是转向角和加速度。姿态控制基准测试的状态包括三个维度的角速度和罗德里格斯参数。姿态控制基准测试的控制输入是扭矩。游泳者基准测试的状态包括位置坐标、前端角度、转子角度、端点速度、端点角速度和转子的角速度。控制输入是两个转子的扭矩。

Swimmer基准测试是四项基准测试中最复杂的一项。游泳者智能体至少由三个首尾相连的节段组成，每两个相邻节段之间通过铰接关节连接。游泳者位于二维池塘中，其目标是通过对转子施加扭矩并利用流体摩擦，尽可能快速地向右移动。

A.2 超参数

	A2C	PPO
学习率	7e-4	3e-4
更新频率	每 5 步	每 2048 步
伽马	0.99	0.99
价值函数系数	0.5	0.5
RMSProp Epsilon	1e-5	NA
最大梯度范数	0.5	0.5

表3：A2C 和 PPO 的超参数

	DDPG	TD3	SAC
学习率	1e-3	3e-4	3e-4
批量大小	100	100	256
缓冲区大小	1e6	1e6	1e6
伽马	0.99	0.99	0.99
软更新系数	5e-3	5e-3	5e-3

表 4：DDPG、TD3 和 SAC 的超参数

我们的算法中有多个超参数，包括训练和测试。对于所有试验，我们采用以下两个超参数。

（1）训练期间的最大轨迹长度。除游泳者基准测试设置为 50 外，所有基准测试的该超参数均设置为 30，并且在测试阶段保持与最大截止期限相同。为训练选择合适的最大轨迹长度十分重要。一方面，如果长度过短，系统可能没有足够的时间依据物理定律到达目标集合。如果系统在训练期间始终无法触及目标集合，我们的框架将退化，并且无法从 ASAP 获得鼓励。另一方面，如果最大轨迹长度过长，训练需要更多步数，也会变得耗时。在训练步数固定的情况下，最大轨迹长度越长，能够探索的轨迹数量就越少。探索不足可能是性能下降的原因，因为每条轨迹都从状态空间中的一个随机初始点开始，而由于经历的轨迹数量减少，状态空间中被探索的比例也会降低。如果将该参数设置为 40 或 60 等更大的值，智能体在增加训练步数后可以达到类似的趋势。

（2）训练步数总数。对于直流电机位置基准测试，该超参数设置为 1e6；对于自行车基准测试和姿态控制基准测试，设置为 3e6；对于游泳者基准测试，设置为 6e6。直观而言，系统动力学越复杂，收敛所需的训练步数就越多。更多证据可参见附录 B。

随后，我们在表3和表4中列出了实现中训练算法特有的一些重要超参数。具体而言，表3展示了 A2C 和 PPO 算法的超参数，说明如下。

1. 价值函数系数表示用于损失计算的价值函数系数。
2. RMSProp 是 AdaGrad 方法的一种变体，在各种问题上都能取得良好性能，尤其适用于非凸优化。RMSProp Epsilon 用于稳定 RMSProp 更新分母中的平方根计算。3. 最大梯度范数控制梯度裁剪范围。

除此之外，PPO 算法在广义优势估计器中使用了 0.95 这一因子，以权衡偏差和方差。此外，每经过 10 个周期，替代损失都会再次进行优化。

此外，表 4 展示了我们实验中 DDPG、TD3 和 SAC 算法的详细信息，具体说明如下。

1. DDPG 的学习率高于其他算法。这一设置通常用于训练。 2. 对于我们的实验，缓冲区大小大于训练步数。换句话说，在训练过程中，我们不会从回放缓冲区中弹出样本。

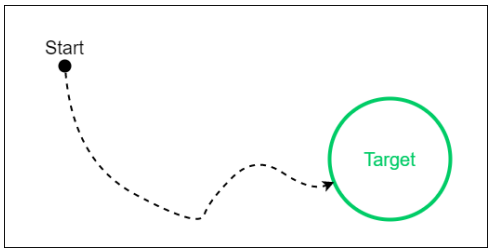
除此之外，所有算法的网络均为多层感知器。

A.3 任务详情

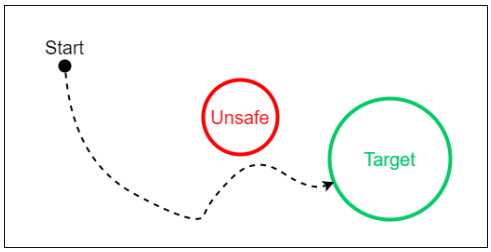
如图3所示，我们在实验中考虑两类任务： $Reach$任务和 $Reach$&$Avoid$ 任务。我们不对初始状态作任何假设。

	直流电机位置	自行车
# 状态维度	3	4
# 动作维度	1	2
线性	线性	非线性
连续性	连续	连续
目标中心	$[\pi/2, 0, 0]$	$[1,1,0, \sqrt{2}]$
目标半径	0.5	0.8
不安全中心	$[\pi/4, 0, 0]$	$[0.5,0.5,0, \sqrt{2}/2]$
不安全区域半径	0.2	0.3
	姿态控制	游泳者
# 状态维度	6	10
# 动作维度	3	2
线性	非线性	非线性
连续性	连续	连续
目标中心	[0,0,0,0,0,0]	[0,9,0,0,0,0,0,0,0,0]
目标半径	0.8	0.8
不安全中心	[0,0,0.2,0,0,0]	[0,0.6,0,0,0,0,0,0,0,0]
不安全区域半径	0.3	0.3

表5：基准测试详情



(a) $Reach$ 任务



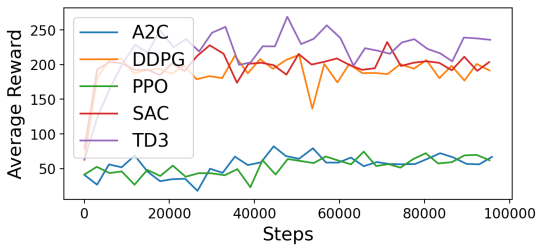
(b) $Reach$&$Avoid$任务

图3：任务示例

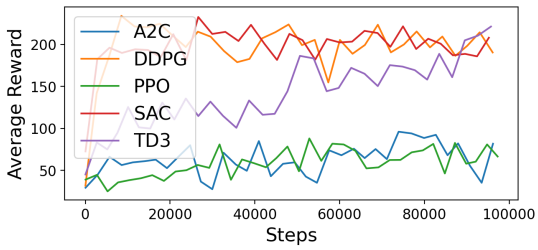
到达与规避任务的 STL 规范格式如下：

$$F_{[0,10]}(Euclid(target) \leq r_t \; and \\ G_{[0,\infty]}(Euclid(unsafe) > r_u) \tag{16}$$

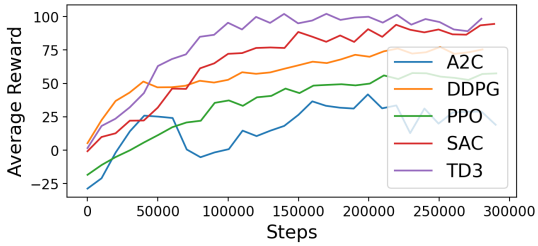
公式的前半部分用于 φ_T 到达任务，后半部分用于 φ_O 规避任务，具体定义见第4.3节。我们希望 φ_T 尽快实现ASAP。 $Euclid$ 是关于该对象的欧氏距离



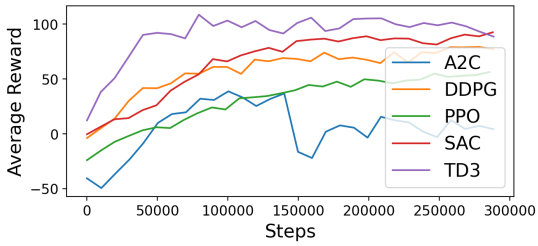
（a）直流电机 *Reach* 任务



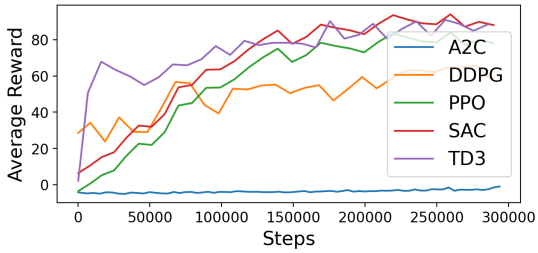
（b）直流电机 *Reach*&*Avoid* 任务



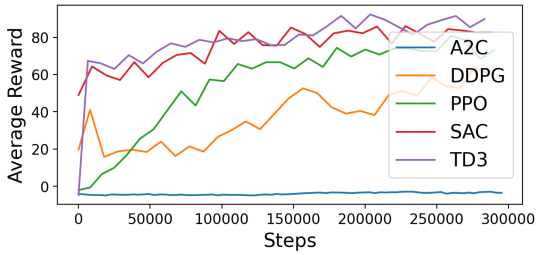
(c) 自行车 *Reach* 任务



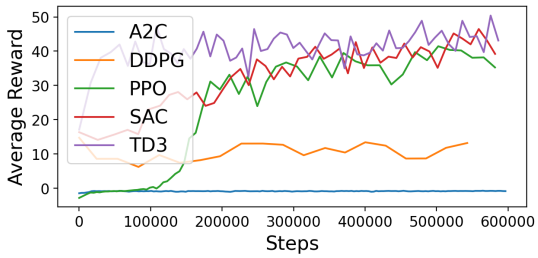
(d) 自行车 *Reach*&*Avoid* 任务



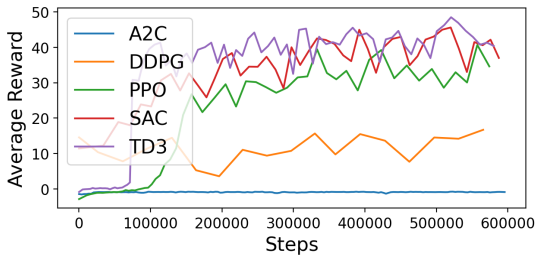
(e) 姿态控制 *Reach* 任务



(f) 姿态控制 *Reach*&*Avoid*任务

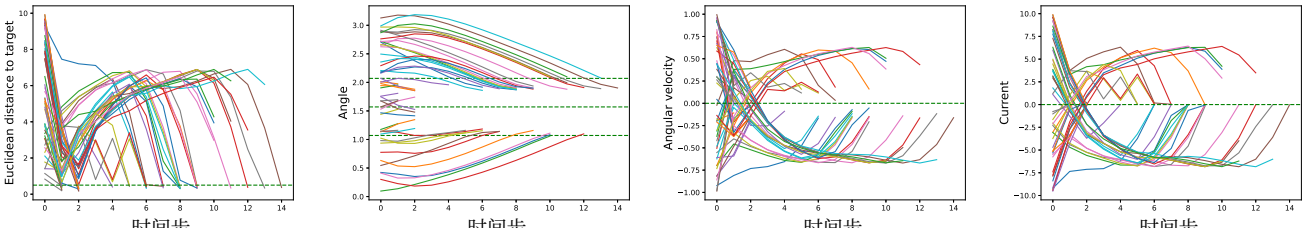


(g) 游泳者 *Reach* 任务



(h) 游泳者 *Reach*&*Avoid* 任务

图4：不同任务训练期间平均奖励的变化。



（a）欧氏距离　（b）角度　（c）角速度　（d）电流

图5： *Reach* 任务在直流电机位置基准测试上的测试轨迹示例

状态空间， r_t 是目标集合的半径， r_u 是不安全集的半径，具体见表 5。

附录 B 详细实验结果

在本节中，我们展示消融实验更详细的实验结果。

B.1 消融研究 1

这一系列图展示了不同算法在不同基准测试的不同任务上的训练历史。

在直流电机位置基准测试中，训练进行了 1e6 步。TD3、SAC 和 DDPG 的表现相近，且优于 PPO 和 A2C。

在姿态控制基准测试和自行车基准测试中，训练需要进行 3e6 步。如图 4 所示，在 1e6 步之后，平均奖励仍在增加。这印证了我们在附录 A 中的直觉。

在游泳者基准测试中，训练需要进行 6e6 步。SAC 和 TD3 的表现优于其他算法。

B.2 消融研究2

在本小节中，我们讨论实现 ASAP 的其他方法及其性能。直观而言，贪心算法可以解决这个问题。我们可以强制系统在每一步都越来越接近目标集合。然而，由于系统动力学遵循物理定律，因此无法保证这种方法有效。

图 5 展示了一个到达任务的示例，其中显示了欧氏距离目标的变化以及系统状态。目标是尽快到达 $[\pi/2, 0, 0, 0]$。每幅图中都有 50 条轨迹；欧氏距离并非单调递减，因为需要通过电流和角速度的变化来驱动角度到达目标，而这取决于系统动力学。相应地，在直流电机位置基准测试中，贪心算法对 $Reach$ 任务的成功率为 67.6%。此外，由于系统动力学未知，很难设计每一步系统接近目标集合的程度，使其相较于前一步有所增加（步进移动常数）。

表达 ASAP 的另一种方式，是将贪心算法转化为 STL 规范，以此鼓励系统不断向目标靠近。然而，STL 无法表达 ASAP；在直流电机位置基准测试中，贪心算法在 $Reach$ 任务上的成功率为 55。3%。

对于到达的 STL 规范，我们使用最终算子（F）及其对应的语义。用自然语言来说，该规范的含义是“最终到达目标集合”。也可以将其改为全局算子（G），训练得到的智能体表现相近。

需要注意的是，STL 的定量语义并非我们的贡献。此外，我们已经测试过，如果在同一任务上将上述替代方法与 ASAP-Phi 结合使用，其成功率可以超过 95%。这一结果部分说明了我们的框架的必要性。

附录 C 主证明的解释

在本附录部分，我们通过图示进一步解释第 4.2 节中定理 2 的证明。

总的来说，该证明旨在表明，ASAP-Phi 策略的最优性意味着它是一个 ASAP 策略。等价地，我们可以证明，存在一个最优但非 ASAP 的策略会导致矛盾。因此，我们假设存在这样一个策略 π^*：由于其非 ASAP 的性质，它倾向于选择轨迹 $\bar{s}^*$，而该轨迹满足性质的时间晚于另一条备选轨迹$\bar{s}'$ 。这两条轨迹已在第 4.2 节中示出，为方便起见，我们在此重新绘制，如图 6 所示。

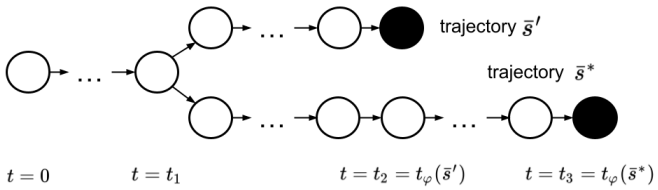


图 6：在假设的 π^* 下存在的两条轨迹。空心圆表示截至目前的轨迹不满足 φ，实心圆则表示满足。

接下来，由于假设1，可以一直维持 $\bar{s}'$ 满足状态，直到时间步 t_3。也就是说，轨迹 $\bar{s}''$ 从 $\bar{s}'$ 延伸至 t_3，并从 t_2 到 t_3 始终保持性质满足。如图 7 所示。

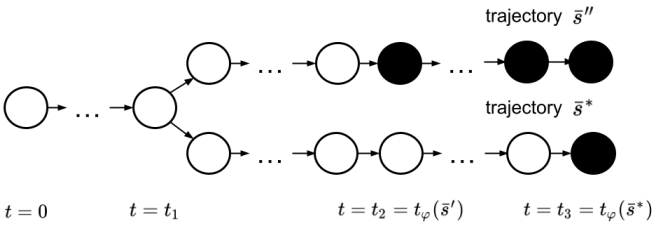


图 7：轨迹 $\bar{s}''$ 从 $\bar{s}'$ 延伸，并从 t_2 到 t_3 保持性质满足。

为了证明当前策略 π^* 比 $\bar{s}^*$ 更偏好$\bar{s}'$（因此也比 $\bar{s}''$ 更偏好），我们需要证明它并未在所有状态下最大化状态价值函数。具体而言，我们选取两条轨迹的共同初始状态 s_0，并计算策略 π^*下该状态的价值。总体而言，该价值可分为四个部分，如图 8 所示。

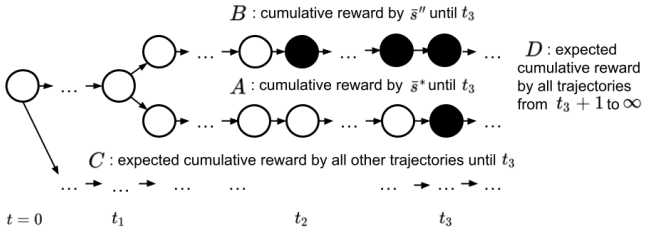


图 8： s_0 处的价值分为四个部分： $p_{\bar{s}^*}A$、 $p_{\bar{s}''}B$、 C 和 D。

这一划分由证明中的公式（8）和（9）推导得出。总体而言，记在某一策略下采取轨迹 $\bar{s}^*$ 和 $\bar{s}''$ 的概率分别为 $p_{\bar{s}^*}$ 和$p_{\bar{s}''}$。无论采用何种策略，这四个部分均为

1. 从起点到 t_3沿轨迹$\bar{s}^*$ 行动所获得的期望累积奖励，即 $p_{\bar{s}^*}A$；2. 从起点到 t_3沿轨迹$\bar{s}''$ 行动所获得的期望累积奖励，即 $p_{\bar{s}''}B$；3. 从起点到 t_3沿除这两条轨迹之外的任意其他轨迹行动所获得的期望累积奖励，即 C；4. 从 t_3+1 到 ∞沿任意轨迹行动所获得的期望累积奖励，即 D。

因此，对于任意策略 π，我们有状态 s_0处的价值为

$$v^*_\pi(s_0) = p_{\bar{s}^*}A + p_{\bar{s}''}B + C_\pi + D_\pi. \tag{17}$$

注意， A 和 B 是确定性项，与任何策略分布无关，但 C 和 D 是期望值，因此它们由分布 π参数化。

在我们假设的最优策略 π^* 下，该策略比 $\bar{s}^*$ 更偏好 $\bar{s}''$，即在 π^* 下，我们有 $p_1 = p_{\bar{s}^*} > p_{\bar{s}''} = p_2$

$$v^*_{\pi^*}(s_0) = p_1A + p_2B + C_{\pi^*} + D_{\pi^*}. \tag{18}$$

找到这个价值划分后，我们寻求一个替代策略 π'，其价值 $v^*_{\pi'}(s_0)$ 高于 $> v^*_{\pi^*}(s_0)$，因此 π^* 在状态 s_0 下并未使价值最大化，也不是最优策略，从而产生矛盾。

如主证明中所述，我们通过交换采取轨迹 $\bar{s}''$ 和 $\bar{s}^*$ 的概率来构造替代策略 π'，而其他所有事项的概率保持不变。这给出了 $p_{\bar{s}''} = p_1$ 和 $p_{\bar{s}^*} = p_2$，即

$$v^*_{\pi'}(s_0) = p_2A + p_1B + C_{\pi'} + D_{\pi'}. \tag{19}$$

从公式（10）到证明结束，我们逐项比较这两个价值。该过程确实表明 $v^*_{\pi'}(s_0) > v^*_{\pi^*}(s_0)$，并得出矛盾。