arXiv:2304.02324v1 [eess.SY] 5 Apr 2023

基于凸优化的策略自适应：补偿分布偏移

Navid Hashemi[1], Justin Ruths[2] 和 Jyotirmoy Deshmukh[1]

[1] 美国加利福尼亚州洛杉矶南加州大学[2] 美国德克萨斯州达拉斯达拉斯德克萨斯大学

摘要。 许多现实世界系统通常包含物理组件或运行环境，其动力学具有高度非线性且存在不确定性。假设能够获得对实际系统具有足够高保真度的模型，可以采用多种不同的控制算法，为此类系统设计最优控制器。然而，在设计最优控制器时对模型随机动力学所作的假设，在系统部署到现实世界后可能不再成立。本文探讨的问题是：假设我们在训练环境中通过求解控制问题获得了一条最优轨迹，如何确保现实世界中的系统轨迹在部署环境中跟踪这条最优轨迹，并将误差降至最低？换言之，我们希望了解如何使训练得到的最优策略适应环境中的分布偏移。分布偏移会给安全关键系统带来问题，因为训练得到的策略在部署期间可能导致不安全的结果。我们表明，这一问题可以表述为非线性优化问题，并可使用粒子群优化（PSO）等启发式方法求解。然而，如果改为考虑该问题的凸松弛，我们就能学习到跟踪最优轨迹的策略，且误差性能更佳、计算速度更快。我们通过使用 Dubins 汽车模型跟踪最优路径，以及使用线性和非线性模型进行自适应巡航控制中的碰撞规避，展示了该方法的有效性。

1 Introduction

在高度不确定的环境中运行的系统通常被建模为满足马尔可夫假设的随机动态系统，即马尔可夫决策过程（MDP）。给定状态 s_t （即时间 t 的状态），离散时间 MDP 定义了在给定 s_t 和时间 t 的控制动作（记为 a_t）的条件下，s_{t+1} 的一个分布。我们将这一分布称为状态转移动力学。对于此类系统，人们已经探索了多种基于模型和数据驱动的控制设计方法，以学习一个最优策略（即从状态集合映射到动作集合的函数），使某个控制理论代价函数 [14] 最小化。基于模型的方法显式地假设状态转移动力学服从特定分布，而数据驱动的方法则隐式地作出这一假设。然而，当系统部署到现实世界中时，该分布可能发生变化；这种分布变化称为 分布偏移。

本文所要解决的根本问题，是对预先学习的控制策略进行自适应，以补偿分布偏移。尽管可以在新环境中重新训练控制策略，但学习最优控制策略通常代价高昂。然而，我们的一项关键观察是：虽然学习最优策略成本很高，但学习一个合理且高保真的状态转移动力学模型可能是可行的。我们将这种学习得到的模型称为代理模型。本文表明，在某些类型的分布偏移下，将现有最优策略自适应到新的部署环境这一问题，可以表述为一个关于最优训练轨迹和代理模型的非线性优化问题。此外，我们还表明，如果代理模型是一个神经网络（采用基于整流线性单元或 ReLU 的激活函数），则原始优化问题存在一个凸松弛。该凸松弛能够高效地求得修改后的动作，使最优训练轨迹与部署环境中的系统轨迹之间的误差最小化。最后，我们通过实验证明，如果训练轨迹满足预期的安全目标，那么这种策略自适应便能够为部署环境提供安全保障。

我们工作的主要技术思想受到 [8], 中近期研究的启发，作者在其中提出了一种高效方法，可在输入服从高斯分布的条件下为神经网络的输出提供概率界。我们展示了如何利用这一结果来传播分布偏移的影响。然而， [8] 中的结果并未考虑寻找最优动作的问题（这是一个非凸问题）。在本研究中，我们提出一种方法，将该结果凸化，以寻找最优动作。我们在使用Dubin汽车模型的跟踪问题以及采用自适应巡航控制的避撞问题上演示了我们的技术。

本文其余部分组织如下。在第2节中，我们讨论预备知识、术语和技术符号。在第3节中，我们讨论我们的策略自适应方法，并在第4节给出实验结果。我们在第5节以相关工作作为结语。

2 预备知识

符号说明。多元高斯分布记为 $\mathcal{N}(\mu, \Sigma)$，其中 μ和 Σ 分别表示均值向量和协方差。对于高斯分布随机向量 $r \in \mathbb{R}^n$，其均值记为 μ_r ，协方差记为 Σ_r。令 $c \in \mathbb{R}^n$，则以 c 为中心、形状矩阵为 形状矩阵 Ω的椭球记为 $\mathcal{E}(c, \Omega)$，即 $\mathcal{E}(c, \Omega) = \left\{x \middle| (x-c)^\top \Omega^{-1}(x-c) \leq 1\right\}$。给定一个非凸集 $\mathcal{Y}$，我们用记号 $\mathcal{H}(\mathcal{Y})$ 表示包含 $\mathcal{Y}$的椭球集合，即 $\{\mathcal{E}(c, \Omega) \,|\, \mathcal{Y} \subseteq \mathcal{E}(c, \Omega)\}$。

马尔可夫决策过程、最优策略。现在，我们将本文所研究的随机动力系统类型形式化为马尔可夫决策过程。

定义 1（马尔可夫决策过程（MDP））。马尔可夫决策过程是一个元组 $M = (S, A, T, \iota)$，其中 S 和 A 分别表示状态集合和

动作集合， $T(s' \mid s, a)$ 表示在给定当前状态和动作的条件下，下一状态的概率分布； ι 是定义在 S 上的分布，从中采样以确定 *MDP*[3] 的初始状态。

在我们的方法中，我们关注从 MDP 的状态转移动力学中采样得到的有限-时域轨迹。MDP 的策略 $\pi(a \mid s)$是在给定当前状态的条件下，对动作集合上的一个分布。给定 MDP 的固定策略，长度为 T 的轨迹 （记为 τ），或称 MDP 的行为，是由状态 s_0、.....、s_T构成的序列，满足 $s_0 \sim \iota$，并且对于所有 $t \in [0, T-1]$， $s_{t+1} \sim T(s' \mid s_t, \pi(s_t))$。在控制设计问题中，我们假设存在一个代价函数 J，定义在轨迹空间上，并将每条轨迹映射为一个实数值。最优策略 π^* 定义为使从状态 s_0 开始的轨迹上的代价函数期望值最小的策略，其中状态 s_0 根据初始分布ι采样得到。

分布偏移。对于底层状态转移动力学高度非线性的 MDP，求得最优控制策略通常计算开销很大。为解决这类系统的最优控制问题，已有多种设计方法，包括基于模型的方法，如模型预测控制 [10], 随机最优控制 [2], ，以及无模型的方法，如数据预测控制 [12] 和深度强化学习[17] 。无论采用基于模型还是无模型的方法，这些方法都显式或隐式地假设了一个模型，或假设了由环境状态转移动力学编码的分布。一个关键问题是，系统部署到现实世界后，这种分布可能会发生变化。为区分训练 环境和部署 环境，我们分别用 T_{trn} 和 T_{dpl}表示状态转移分布。

问题定义。假设我们有一个系统，并已在状态转移动力学 T_{trn}下训练出最优策略 π^* ；对于从 ι中采样得到的给定初始状态 s_0，我们使用策略 π^*为该系统采样一条最优轨迹 τ_{opt} 。我们将其记为 $\tau \sim (\iota, \pi^*)$。令 $\tau = (s_0, s_1, \ldots, s_T)$。令 $\tau(t)$作为 s_t的简写。对于一条从相同初始状态出发的轨迹（但处于部署环境中），我们希望找到适配后的策略 $\hat{}\pi$，使得在时刻 $t \in [1, T]$， τ_{opt} 与在 T_{dpl} 动力学下的轨迹之间的误差较小。形式化地，

$$\hat{\pi} = \arg\min_{\pi} \mathop{\mathbb{E}}_{\substack{a_t \sim \pi(a|s_t), \\ s_{t+1} \sim T_{dpl}(s'|s_t, a_t)}} \|\tau_{opt}(t+1) - s_{t+1}\| \tag{1}$$

求解式 (1) 中优化问题的一项关键挑战是未知的 T_{dpl} 。本文提出学习一个用于部署阶段状态转移动力学的代理模型。本质上，代理模型是一种数据驱动的

[3] 严格来说，这一定义涉及随机动力系统的转移结构。通常，动力系统通过描述状态变量随时间演化的差分方程或微分方程来定义。我们假设 $T(s'|s, a)$ 因而是与任意给定系统动力学相一致的转移的无限集合。

模型，能够以合理的精度近似实际的系统动力学。代理模型有多种选择，包括高斯过程 [1],概率集成模型 [4], 和深度神经网络（NN）。本文聚焦于神经网络代理模型，因为它们使我们能够考虑策略自适应问题的凸松弛。

基于代理模型的策略自适应。现在我们说明，基于代理模型的策略自适应可以表述为一个非线性优化问题。首先，我们明确寻找优良代理模型的问题。回想一下，$T_{dpl}(s_{t+1} \mid s_t, a_t)$ 被假定为时不变高斯分布，其均值为 $\mu(s,a)$，协方差为 $\Sigma(s,a)$。状态转移动力学的代理模型是一个元组 $(\mu_{NN}(s,a;\theta_\mu)\ \Sigma_{NN}(s,a;\theta_\Sigma))$，其中 μ_{NN} 和 Σ_{NN} 是深度神经网络，其参数分别为 θ_μ 和 θ_Σ。我们可以通过最小化以下损失函数来训练此类神经网络：

$$\mathcal{L}_\mu(\theta_\mu) = \mathop{\mathbb{E}}_{s\sim S, s'\sim T_{dpl}(s'|s,a)} \|\mu_{NN}(s,a;\theta_\mu) - s'\| \tag{2}$$

$$\mathcal{L}_\Sigma(\theta_\Sigma) = \mathop{\mathbb{E}}_{s\sim S} \|\Sigma_{NN}(s,a;\theta_\Sigma) - \Sigma_s(s')\| \tag{3}$$

在上述公式中，期望通过标准的基于蒙特卡洛的采样计算。在第二个公式中，Σ_s 表示相对于样本均值的 s' 的样本协方差。

假设我们已经学习到精度达到预期水平的代理模型，下一步是将策略自适应表述为一个非线性优化问题。我们针对从最优策略中采样得到的一条特定最优轨迹 τ_{opt} 来陈述该问题（不过，该问题可推广至从任意初始状态采样得到的任意最优轨迹）。注意，$\tau_{opt}(0) = s_0$。

$$\forall t \in [0, T-1] : a_t = \arg\min_{a\in A} \|\tau_{opt}(t+1) - \mu_{NN}(s_t, a_t; \theta_\mu)\| \tag{4}$$

我们注意到，由于上面的方程由一个神经网络组成，因此它是一个高度非线性优化问题。在下一节中，我们将展示如何将这个问题凸化。

3 策略自适应

解决方案概述。式 (4) 中被最小化的量在每个时间t都是残差误差，即最优轨迹与部署环境根据其状态和动作预测的均值状态之间的误差。令$r_{t+1} = \tau_{opt}(t+1) - \mu_{NN}(s_t, a_t; \theta_\mu)$。我们的主要思路是：

1. 在任意时刻 t，假设状态 s_t 位于由椭球 $\mathcal{E}(\mu_{s_t}, \Omega_{s_t})$ 描述的置信集中；2. 假设动作 a_t 位于同样由椭球 $\mathcal{E}(\mu_{a_t}, \Omega_{a_t})$ 描述的置信集中；3. 证明残差误差 r_t 可以由一个椭球界定，该椭球的中心和形状矩阵取决于动作 a_t。

4. 通过凸优化找到使残差误差最小的动作 a_t

现在，我们依次解释这些步骤。首先，我们说明为什么需要考虑置信集。假设系统从状态 s_0 开始，那么状态 s_1 服从部署环境的状态转移动力学。实际上，我们只关心以至少达到概率阈值 p的可能性出现的下一状态。对于多元高斯分布，这对应于该分布逆累积分布函数的次水平集；根据以下引理，该集合可以用一个椭球来描述：

引理 1. 具有高斯分布 $r \sim \mathcal{N}(\mu,\ \Sigma)$的随机向量 $r \in \mathbb{R}^n$满足：

$$\Pr\left[\frac{1}{\rho_n}(r-\mu)^\top \Sigma^{-1}(r-\mu) \leq 1\right] = p, \tag{5}$$

其中，$\rho_n = \Gamma^{-1}(\frac{n}{2},\ \frac{p}{2})$和 $\Gamma^{-1}(.,.)$表示 n 维下不完全伽马函数。

上述引理使我们能够使用截断高斯分布定义椭球置信集。中心为 μ 、形状矩阵为 $\rho_n\Sigma$ （其中 ρ_n 如引理 **1** 所定义）的椭球置信区域定义了一个概率测度为 p的集合。

现在，由于我们考虑的策略是随机的（我们也将其建模为高斯分布），因此可采取的动作由条件高斯分布描述。设 μ_{a_t} 为时间 t动作分布的均值，则所有概率大于 p 的动作都可由椭球置信集 $\mathcal{E}(\mu_{a_t}, \Omega_{a_t})$描述。

由于状态转移动力学的分布可能已经发生偏移，继续应用相同的动作 $\pi^*(s_t)$可能会产生不可接受的残差误差 r_{t+1}。因此，我们希望找到一个能够减小残差误差的新动作 a_t。我们假设，通过选取概率大于固定阈值 p 的动作，a_t 位于一个椭球不确定集内。需要指出的是，动作对应的椭球集的中心或形状矩阵未知，但它们是优化问题中的决策变量。

我们注意到，a_t 与 r_{t+1} 之间的关系高度非线性。然而，我们将展示如何对这个问题进行凸化。

在介绍优化问题的凸化之前，我们需要先引入残差值可达集的概念。我们将其称为残差可达集。形式上，给定状态 s_t 的椭球置信区域 $\mathcal{E}(\mu_{s_t}, \Omega_{s_t})$，以及动作 a_t 的椭球置信区域 $\mathcal{E}(\mu_{a_t}, \Omega_{a_t})$，残差可达集 $\mathcal{R}_{t+1}$ 定义如下：

$$\begin{aligned}\mathcal{R}_{t+1}(\mu_{a_t}, \Omega_{a_t}) = \{&\mu_{NN}(s_t, a_t; \theta_\mu) - \tau_{opt}(t+1) \mid \ s_t \in \mathcal{E}(\mu_{s_t}, \Omega_{s_t}),\\ &a_t \in \mathcal{E}(\mu_{a_t}, \Omega_{a_t})\}\end{aligned} \tag{6}$$

在上述方程中，我们注意到残差可达集由 a_t 和 Ω_{a_t}参数化，并希望找到使残差可达集大小最小的 a_t 和 Ω_{a_t} 的取值。然而，残差可达集是一个非凸集。为了使优化问题成为凸优化问题，我们基本上用

集合 $\mathcal{H}(\mathcal{R}_{t+1}(\mu_{a_t}, \Omega_{a_t}))$（所有椭球上界构成的集合）中的一个椭球上界来近似残差可达集。

现在，我们可以将寻找最佳适应后的动作分布的问题表述为以下优化问题：

$$\begin{aligned}(\hat{\mu}_{a_t}, \hat{\Omega}_{a_t}, \hat{\Omega}_{\mathcal{R}_{t+1}}) = \arg\min_{\mu_{a_t}, \Omega_{a_t}} & \mathbf{Logdet}(\Omega_{\mathcal{R}_{t+1}}) \\ \text{s.t.} & \mathcal{R}_{t+1}(\mu_{a_t}, \Omega_{a_t}) \subset \mathcal{E}(\mathbf{0}, \Omega_{\mathcal{R}_{t+1}})\end{aligned} \tag{7}$$

假设我们将残差可达集的椭球界中心设为 0。这是基于我们需要尽可能减小残差的目标。公式 (7) 选择最佳动作 $\hat{a}_t \in \mathcal{E}(\hat{\mu}_{a_t}, \hat{\Omega}_{a_t})$，使得椭球$\mathcal{E}(\mathbf{0}, \hat{\Omega}_{\mathcal{R}_{t+1}})$ 是包围残差可达集的最小椭球。

给定单个椭球置信区域时，神经网络可达集的椭球界构造方法见 [7]。作者随后在 [11] 的定理1中将该方法推广到多个椭球置信区域。我们在引理 2 中结合本文背景重新表述了这些论文的关键结果。

引理 2. 假设 $s_t \in \mathcal{E}(\mu_{s_t}, \Omega_{s_t})$，$a_t \in \mathcal{E}(\mu_{a_t}, \Omega_{a_t})$。则，当(约束 满足时，残差可达集 $\mathcal{R}_{t+1}(\mu_{a_t}, \Omega_{a_t})$ 由 $\mathcal{E}(\mathbf{0}, \Omega_{\mathcal{R}_{t+1}})$ 上界限定（如(7)中所定义）。下文中，$(b_\ell, W_\ell) \in \theta_\mu$ 表示 μ_{NN} 中最后一层的偏置向量和权重。

$$\tau_1 M_{s_t} + \tau_2 M_{a_t} + M_\phi - M_{out} \leq 0, \quad \textit{for some } \tau_1, \tau_2 \geq 0. \tag{8}$$

这里，M_ϕ 是 *[7],* 中提出的一个二次约束，用于表示神经网络中的*ReLU*隐藏层，且[4]，

$$\begin{aligned}M_{s_t} &= \frac{1}{\rho_n} E_1^\top \begin{bmatrix} -\Sigma_{s_t}^{-1} & \Sigma_{s_t}^{-1}\mu_{s_t} \\ \mu_{s_t}^\top \Sigma_{s_t}^{-1} & -\mu_{s_t}^\top \Sigma_{s_t}^{-1}\mu_{s_t} + \rho_n \end{bmatrix} E_1, \\ M_{a_t} &= E_2^\top \begin{bmatrix} -\Omega_{a_t}^{-1} & \Omega_{a_t}^{-1}\mu_{a_t} \\ \mu_{a_t}^\top \Omega_{a_t}^{-1} & -\mu_{a_t}^\top \Omega_{a_t}^{-1}\mu_{a_t} + 1 \end{bmatrix} E_2\end{aligned}$$

$$E_1 = \left[\begin{array}{cc|c|c} I_n & 0_{n\times m} & 0_{n\times(\sum_{i=2}^{\ell+1} N_i)} & 0_{n\times 1} \\ \hline 0_{1\times n+m} & & 0_{1\times(\sum_{i=2}^{\ell+1} N_i)} & 1 \end{array}\right],$$

$$E_2 = \left[\begin{array}{cc|c|c} 0_{m\times n} & I_m & 0_{m\times(\sum_{i=2}^{\ell+1} N_i)} & 0_{m\times 1} \\ \hline 0_{1\times n+m} & & 0_{1\times(\sum_{i=2}^{\ell+1} N_i)} & 1 \end{array}\right]$$

$$M_{out} = \begin{bmatrix} C & b \\ 0 & 1 \end{bmatrix}^\top \begin{bmatrix} -\Omega_{\mathcal{R}_{t+1}}^{-1} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} C & b \\ 0 & 1 \end{bmatrix},$$

以及 $C = \begin{bmatrix} 0\, 0 & \cdots & W_\ell \end{bmatrix}$，$b = b_\ell - \tau_{opt}\,(t+1)$，

[4] 参数 N_i 在变换矩阵 E_1、E_2 中表示 μ_{NN} 的第 i 层中的 ReLU 激活数量。

在上述引理中，假设适应后的动作及其协方差的形状矩阵已知，因此 M_{a_t} 是一个固定矩阵；然而，在我们希望求解的优化问题中，相应矩阵M_{a_t}、 μ_{a_t} 和 Ω_{a_t} 将作为变量出现，这会使问题变为非线性。我们可以通过进行两次变换来解决这一问题。第一次变换通过变量替换，将非线性集中到 M_{a_t} 的一个标量元素中。我们令 $U_{a_t} = \tau_2\Omega_{a_t}^{-1}$、 $V_{a_t} = \tau_2\Omega_{a_t}^{-1}\mu_{a_t}$，所得 M_{a_t} 如下所示：

$$\begin{aligned} M_{a_t}^* &= E_2^\top \begin{bmatrix} -U_{a_t} & V_{a_t} \\ V_{a_t}^\top & -\left(\tau_2\mu_{a_t}^\top\Omega_{a_t}^{-1}\right)\left(\frac{\Omega_{a_t}}{\tau_2}\right)\left(\tau_2\Omega_{a_t}^{-1}\mu_{a_t}\right)+\tau_2 \end{bmatrix} E_2 \\ &= E_2^\top \begin{bmatrix} -U_{a_t} & V_{a_t} \\ V_{a_t}^\top & -V_{a_t}^\top U_{a_t}^{-1}V_{a_t}+\tau_2 \end{bmatrix} E_2 \end{aligned} \tag{9}$$

我们提出的矩阵 $M_{a_t}^*$ 是非线性的，其非线性体现在标量变量 $V_a^\top U_a^{-1}V_a$ 中。

我们还注意到，适应后的动作应满足执行器约束界 $[\ell,\mathbf{u}]$，我们将其作为如下凸约束：

$$U_{a_t}\ell \le V_{a_t} \le U_{a_t}\mathbf{u}, \tag{10}$$

我们将证明推迟到附录 B。陈述最终定理之前，我们先对式 (7) 作一番观察。在没有额外约束的情况下，式 (7) 的最优解总是返回 Ω_{a_t}，使得 $\mathbf{tr}(\Omega_{a_t}) = 0$（证明见附录 A）。这会导致数值错误，因为引理 2 中出现了该矩阵的逆，而该逆未定义。为避免这一问题，我们对该矩阵的迹施加一个极小的下界。

最后，鉴于 $\mathcal{E}(\mu_{a_t},\Omega_{a_t})$ 的优化所需的所有约束均已给出，我们可以将它们汇总为一个凸优化问题，并由此得到修改后的动作集。以下定理阐明了修改后的动作的正确性及其保守性。证明见附录。

定理1。给定调节因子 $\delta > 0$，并定义 $\Omega = \Omega_{\mathcal{R}_{t+1}}^{-1}$，假设决策变量为 τ_1，$\tau_2 \ge 0$。则以下凸优化，

$$\begin{cases} \min\limits_{M_\phi, V_{a_t}, U_{a_t}, \tau_1, \tau_2} -\mathbf{Logdet}(\Omega) \\ \mathbf{s.t.}\ -\tau_1 M_{s_t} - E_2^\top \begin{bmatrix} -U_{a_t} & V_{a_t} \\ V_{a_t}^\top & \tau_2 \end{bmatrix} E_2 - M_\phi + M_{out} \ge 0, \\ U_{a_t}\boldsymbol{\ell} \le V_{a_t} \le U_{a_t}\boldsymbol{u}, \quad \mathbf{tr}(U_{a_t})\delta \le \tau_2. \end{cases} \tag{11}$$

得到值 $(V_{a_t},\ U_{a_t})$，使得修改后的确定性决策 $\hat{a}_t^c$ 可以用 $\hat{a}_t^c = \hat{}\mu_{a_t} = U_{a_t}^{-1}V_{a_t}$进行近似。

针对模型可能存在的鲁棒性问题（对抗样本），作为一项预防措施，我们需要比较 π^* 和 $\hat{a}_t^c$，并通过以下方式选出修改后的动作 $\hat{a}_t$的最佳选择：

$$\hat{a}_t = \underset{a\in\{\pi^*,\ \hat{a}_t^c\}}{\arg\min} \|\tau_{opt}(t+1) - \mu_{NN}(s_t,a;\theta_\mu)\|_2. \tag{12}$$

算法1： 用于补偿分布偏移的过程

输入： s_1 以及训练好的自主智能体，π^*

结果： 残差对应的小型可达集，其中心更接近原点。

1 • 采样最优轨迹 τ_{opt}。

2 **foreach**时间步 t **执行**

3 • **如果**$(t=1)$ $[\hat{a}_1 \leftarrow \pi^*]$ **否则**$(t \geq 2)$ $\begin{cases} 1 \text{ - 应用定理} & 1 \quad a_t^c \text{ 并计算 ˆ - 根据 (12)，在 ˆ 和 } \pi^* \text{之间选择最佳动作，并返回} \hat{a}_t. \\ 2 & a_t^c \end{cases}$

4 • 利用观测 s_t 和 $\hat{a}_t$ 刻画置信区域 S_{t+1} 使用深度代理模型（见附录 D）进行部署环境。

5 • 记录通过对 $\hat{a}_t$ 施加作用所产生的观测 s_{t+1} 到环境

6 **结束**

更多详情请参见附录 D。我们在算法1中总结了所提方法的主要步骤。

3.1 可扩展性

[7] 中引入的紧椭球界近似的保守性会随着神经网络结构复杂度的增加而增加，并导致定理1的解不准确。然而，对于高度非线性的部署环境，有必要为代理模型训练一个深度神经网络。针对这一问题（类似于 [16]），我们采用嵌入网络 $\mathcal{M}_p$，将状态 s_t 映射到另一空间 $s'_t \in \mathbb{R}^{n'}$（$s'_t = \mathcal{M}_p(s_t;\ \theta_p)$），使得$s'_t$ 在训练方面比 s_t 更易处理。接下来，我们基于 s'_t 定义代理模型及其参数 θ_μ 如下，

$$\mu_{s_{t+1}} = \mu_{NN}(s'_t,\ \hat{a}_t; \theta_\mu).$$

在这一针对高度非线性部署环境的设定下，神经网络 μ_{NN} 不一定是深度神经网络。因此，给定二元组（s_t，a_t）作为输入向量、s_{t+1} 作为输出，我们为该函数安排训练过程，

$$\mu_{s_{t+1}} = \mu_{NN}\left(\mathcal{M}_p(s_t; \theta_p),\ a_t; \theta_\mu\right)$$

共同学习参数 θ_μ、θ_p，并在凸规划中使用 θ_μ。换句话说，给定 s_t 的分布和参数 θ_p，我们可以利用高斯混合模型技术 [20]近似 s'_t 的分布，并将其置信区域引入凸优化（通过 μ_{NN}）以进行策略修改。

4 实验结果

与 *PSO* 的比较。我们假设简单汽车环境，并比较我们的凸规划技术与粒子群优化（PSO） [13][5], 在求解优化问题（4）时的性能。PSO 在低维环境中表现良好。因此，我们计划展示，即使可扩展性并非重要问题，我们的凸规划技术仍优于 PSO。该环境表示以下简单汽车模型:

$$\begin{aligned} &\dot{x} = u\mathbf{cos}(\theta), \quad \dot{y} = u\mathbf{sin}(\theta), \quad \dot{\mathbf{sin}}(\theta) = \frac{u}{\ell}\mathbf{tan}(\phi)\mathbf{cos}(\theta), \\ &\dot{\mathbf{cos}}(\theta) = -\frac{u}{\ell}\mathbf{tan}(\phi)\mathbf{sin}(\theta) \end{aligned} \tag{13}$$

该系统描述一辆长度为 ℓ 、以恒定速度 u 行驶并由控制动作 ϕ驱动的汽车。训练环境的特征为 ℓ =2.5 和 $u = 4.9$ ，而部署环境略有不同，为$\ell = 2.1$ 和 $u = 5.1$。我们以 0.01 秒的时间步长，从部署环境收集训练数据集。

我们为部署环境 μ_{NN}, μ^*_{NN}训练了两个代理模型。前者用于优化（11），是一个维度为 $[5, 8, 4]$的 ReLU 神经网络。该 ReLU 神经网络由第 3.1 节提出的方法构建。后者用于比较，相关讨论见公式（12）和附录 D；它是一个维度为 $[5, 200, 200, 200, 200, 200, 200, 200, 4]$的深度 tanh() 神经网络。策略修改的结果如图 1 所示。图中以绿色显示最优轨迹，τ_{opt}。该轨迹通过对训练环境采用最优控制和基于模型的算法计算出的高保真代理模型进行仿真得到。蓝色曲线表示算法1运行（500 步）的结果，在所有三个状态 x, y, θ上都紧密跟踪最优轨迹。红色曲线表示未进行策略修改时部署环境中的轨迹。在个人笔记本电脑上使用 YALMIP 和 MOSEK 求解器进行凸规划的运行时间为$[0.005, 0.027]$ 。因此，我们将 PSO 的运行时间限制为 0.027，并将其用于策略修改。在一次尝试中，我们在同一模型上使用 PSO 求解优化问题（4），该模型也采用凸规划， μ_{NN} 所得轨迹以黑色显示。另一次尝试中，我们在深度模型 μ^*_{NN} 上使用 PSO，所得轨迹以洋红色显示。结果清楚表明，在这两种情况下，我们的凸规划都优于 PSO。

[5] 众所周知，非线性优化技术缺乏保证，且可能陷入局部极小值，但粒子群等技术在实践中表现良好，尤其是在低维系统中。因此，我们进行这一比较，以表明凸化在残差最小化方面优于最先进的全局优化方法。

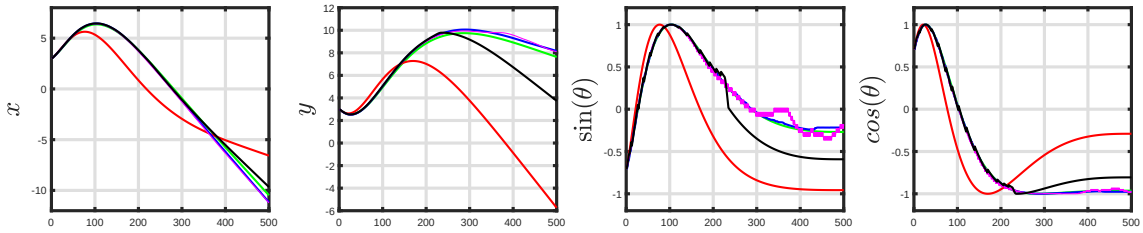

图1：展示了PSO与我们的凸规划之间的比较。绿色曲线和蓝色曲线分别是算法1和最优轨迹的结果。红色曲线表示在没有策略修改时部署环境的轨迹。我们在与凸规划相同的模型上利用PSO进行优化（4），μ_{NN} 所得轨迹以黑色显示。我们还在深度模型上利用PSO，μ_{NN}^{*} 所得轨迹以品红色显示。

汽车线性环境。训练环境是如下随机线性动力学：

$$\mathbf{x}_{t+1} = \begin{bmatrix} 1 & 0.1 & 0.0047 \\ 0 & 1 & 0.0906 \\ 0 & 0 & 0.8187 \end{bmatrix} \mathbf{x}_t + \begin{bmatrix} 0.003 \\ 0.0094 \\ 0.1813 \end{bmatrix} u_t + \nu_t,$$
$$\nu_t \sim \mathcal{N}\left(\begin{bmatrix} 0 & 0 & 0.2 \end{bmatrix}^\top, \exp(-8) I_3 \right)$$

部署环境同样是如下随机线性动力学：

$$\mathbf{x}_{t+1} = \begin{bmatrix} 1 & 0.1 & 0.0046 \\ 0 & 1 & 0.0885 \\ 0 & 0 & 0.7788 \end{bmatrix} \mathbf{x}_t + \begin{bmatrix} 0.004 \\ 0.0115 \\ 0.2212 \end{bmatrix} u_t + \eta_t,$$
$$\eta_t \sim \mathcal{N}\left(\mathbf{0}, \exp(-8) I_3\right)$$

其中，采样时间为 $t_s = 0.1\ s$。状态 $\mathbf{x}_t \in \mathbb{R}^3$ 定义为 $\mathbf{x}_t = [x_t,\ v_t,\ a_t]^\top$，分别表示汽车的位置、速度和加速度。标量动作 u_t 也限制在 $u_t \in [-3,\ 3]$范围内。由于环境是线性的，因此无需使用嵌入网络。因此，我们只针对部署环境 μ_{NN} 训练一个代理模型，采用维度为 $[4, 10, 5, 3]$ 的两层隐藏层 ReLU 神经网络。我们还可以使用训练环境的模型和训练好的最优反馈策略。因此，我们通过算法1对部署环境进行策略修改，结果如图2所示。图中，绿色曲线表示仿真的最优轨迹。蓝色和红色曲线分别表示进行和未进行策略修改时部署环境的轨迹。该图表明，算法1促使部署环境跟踪规划器 τ_{opt}，策略修改过程取得了成功。

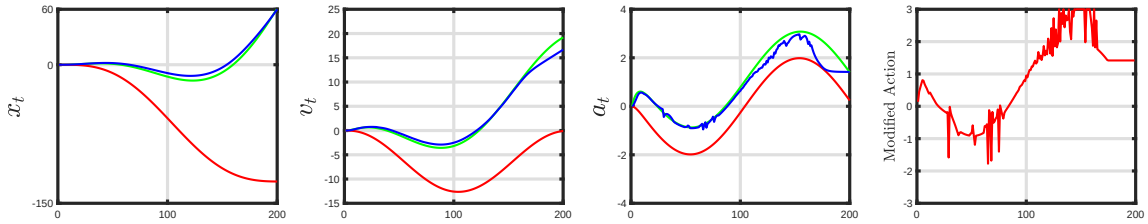

图2：展示了在汽车的随机线性环境中进行策略修改的结果。在该图中，绿色曲线表示模拟最优轨迹。蓝色和红色曲线分别表示存在和不存在策略修改时部署环境的轨迹。

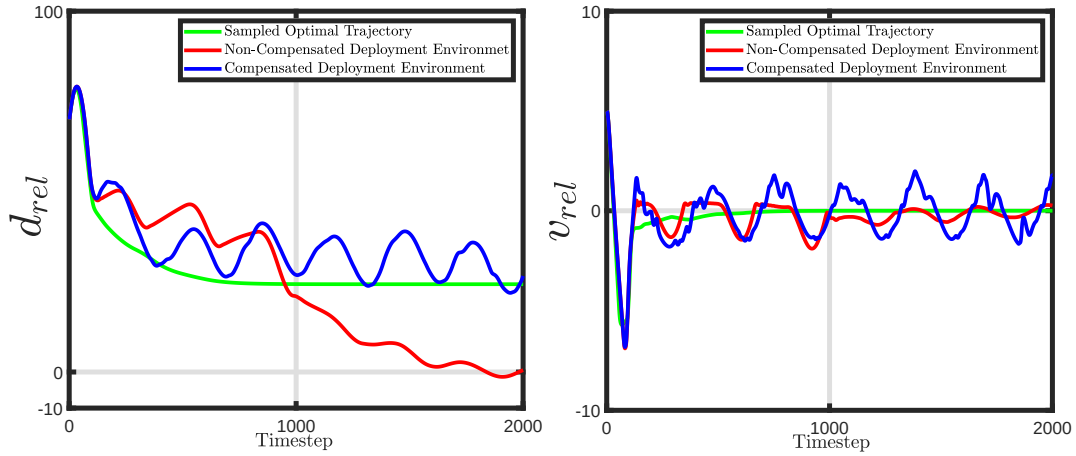

图3：绿色曲线表示 v_{rel} 和 d_{rel}的最优轨迹，而红色和蓝色曲线分别表示未进行策略自适应和进行策略自适应的部署环境轨迹。

自适应巡航控制。考虑 MATLAB 文档 [6] 中用于自适应巡航控制的 Simulink 环境。我们考虑这个经过训练的反馈控制器，并假设可以使用训练环境的模型。随后，我们使用该模型和控制器仿真最优轨迹 τ_{opt}。该控制器经过 14 小时的训练，这清楚地表明，训练新的控制器可能代价高昂，也说明了我们技术的贡献。训练好的控制器的输入是向量 $\mathbf{x} = [\int v_{err},\ v_{err},\ v_{ego}]^\top$。因此，我们将该向量作为环境的状态 [7]。这意味着前车、自车和信号

[6] https://www.mathworks.com/help/reinforcement-learning/ug/train-ddpg-agent-for-adaptive-cruise-control.html [7] 这里的 v_{err} 是一个基于逻辑的函数，取决于 x_{ego}、x_{lead}、v_{ego} 和 v_{lead}。有关详细信息，请参阅 MATLAB 文档。这里，(v_{ego}, v_{lead}) 和 (x_{ego}, x_{lead}) 分别表示自车和前车的速度和位置。

处理模块共同构成了环境。需要注意的是，由于信号处理模块中存在基于逻辑的关系，该环境具有高度非线性。在训练环境中，V_{set} 的设定值为 $30\ m/s$，而在部署环境中却被错误地设为 $34.5\ m/s$。这种差异体现了分布偏移。

图3展示了前车与自车之间相对速度和相对位置 v_{rel}、d_{rel} 的变化。绿线表示在训练环境中应用最优策略 π^* 时，v_{rel}、d_{rel} 的仿真结果。另一方面，蓝线和红线分别表示存在和不存在策略修改时，v_{rel}、d_{rel} 的变化。策略修改过程旨在使部署环境的状态跟踪最优轨迹 τ_{opt}。以红线在 $t = 183$ 秒时的参数 $d_{rel} < 0$ 为例。因此，在没有策略修改的情况下，分布偏移会导致事故。图3表明，我们的策略修改技术能够保护系统免遭潜在事故。

5 结论与相关工作

相关工作。在 [5], 中，作者将学习得到的策略与原始输入和输出分离，以便从仿真中进行迁移。在 [3] 中，作者提出了一种方法，利用学习得到的深度逆动力学模型，判断哪种现实世界动作最适合实现与仿真器中相同的状态。互对齐迁移学习方法通过辅助奖励，在仿真到机器人的迁移过程中，应对系统动力学差异带来的影响（[21]）。诸如 [6] 的方法通过修改奖励函数来弥补动力学差异，使修改后的奖励函数惩罚智能体在源域中访问目标域无法到达的状态以及采取目标域无法执行的动作。在 [19], 中，作者研究了鲁棒对抗强化学习。受 $\|H\|_\infty$ 控制思想的启发，他们将致不稳定对手（如仿真与环境之间的差异）视为不确定因素，并设计了一种考虑最坏情况对手且能抵御该对手的学习算法。此外，研究人员也在多智能体环境中探讨了迁移学习 [15], ，研究如何训练具有连续动作的智能体，以确保即使对手的策略发生变化，训练好的智能体仍能泛化。 [18] 提出使用贝叶斯优化（BO）主动选择环境变量的分布，以最大化策略梯度方法每次迭代带来的改进。与提出奖励修改技术的作者[6] 不同，我们提出了一种策略修改技术，以应对训练期间的环境模型与预期不符的问题。

结论与更广泛的影响。在这项工作中，我们提出了一种用于非线性系统的线性化算法，该系统由配备 ReLU 激活函数的深度神经网络训练；此外，我们提出了一个基于凸优化的框架，用于补偿未知模型上的分布偏移，且该分布偏移本身也未知。这种方法的优势在于计算方便，控制器能够即时响应分布偏移；代价是基于椭球界的线性化会带来少量额外的保守性。

未来工作。算法1目前逐步执行优化，这是一个贪心方法。我们计划推导一种涵盖整个轨迹的单一优化方法，以改善结果。

参考文献

1. Alpaydin, E.：《机器学习导论》。麻省理工学院出版社（2020年） 2. Bertsekas, D., Shreve, S.E.：《随机最优控制：离散时间情形》，第5卷。Athena Scientific（1996年） 3. Christiano, P., Shah, Z., Mordatch, I., Schneider, J., Blackwell, T., Tobin, J., Abbeel, P., Zaremba, W.：通过学习深度逆动力学模型，实现从仿真到现实世界的迁移。arXiv预印本 arXiv:1610.03518（2016年） 4. Chua, K., Calandra, R., McAllister, R., Levine, S.：利用概率动力学模型，仅通过少量试验进行深度强化学习。arXiv预印本 arXiv:1805.12114（2018年） 5. Clavera, I., Held, D., Abbeel, P.：通过模块化和奖励引导实现策略迁移。载于：2017年IEEE/RSJ智能机器人与系统国际会议（IROS）。第1537–1544页。IEEE（2017年） 6. Eysenbach, B., Asawa, S., Chaudhari, S., Salakhutdinov, R., Levine, S.：离线动力学强化学习：利用域分类器进行迁移训练。arXiv预印本 arXiv:2006.13916（2020年） 7. Fazlyab, M., Morari, M., Pappas, G.J.：通过半定规划对神经网络进行概率验证与可达性分析。arXiv预印本 arXiv:1910.04249（2019年） 8. Fazlyab, M., Morari, M., Pappas, G.J.：通过二次约束和半定规划对神经网络进行安全性验证与鲁棒性分析。CoRR abs:1903.01287（2019年） 9. Fazlyab, M., Morari, M., Pappas, G.J.：通过二次约束和半定规划对神经网络进行安全性验证与鲁棒性分析。《IEEE自动控制汇刊》（2020年） 10. Garcia, C.E., Prett, D.M., Morari, M.：模型预测控制：理论与实践综述。《Automatica》**25**（3），335–348（1989年） 11. Hashemi, N., Fazlyab, M., Ruths, J.：神经网络估计器的性能界：在故障检测中的应用。载于：2021年美国控制会议（ACC）。第3260–3266页（2021年）。https://doi.org/10 9482752 12 23919 50511 2021 . / ACC . . . Jain, A., Smarra, F., Mangharam, R.：使用回归树和集成学习进行数据预测控制。载于：2017年IEEE第56届年度决策与控制会议（CDC）。第4446–4451页。IEEE（2017年） 13. Kennedy, J., Eberhart, R.：粒子群优化。载于：ICNN'95国际神经网络会议论文集。第4卷，第1942–1948页。IEEE（1995年） 14. Khalil, I., Doyle, J., Glover, K.：《鲁棒与最优控制》。Prentice Hall（1996年） 15. Li, S., Wu, Y., Cui, X., Dong, H., Fang, F., Russell, S.：通过极小极大深度确定性策略梯度实现鲁棒多智能体强化学习。载于：AAAI人工智能会议论文集。第33卷，第4213–4220页（2019年） 16. Lusch, B., Kutz, J.N., Brunton, S.L.：用于非线性动力学通用线性嵌入的深度学习。《自然·通讯》**9**（1），1–10（2018年） 17. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., 等：通过深度强化学习实现人类水平控制。《自然》**518**（7540），529–533（2015年）

18. Paul, S., Osborne, M.A., Whiteson, S.：用于鲁棒强化学习的指纹策略优化。载于：国际机器学习会议。第5082–5091页（2019年） 19. Pinto, L., Davidson, J., Sukthankar, R., Gupta, A.：鲁棒对抗式强化学习。载于：国际机器学习会议。第2817–2826页（2017年） 20. Reynolds, D.A., 等：高斯混合模型。《生物识别百科全书》 **741**（659–663）（2009年） 21. Wulfmeier, M., Posner, I., Abbeel, P.：互对齐迁移学习。载于：机器人学习会议。第281–290页（2017年）

6 附录

附录 A：通用计算误差

我们在附录 E 中简要总结了 [7] 和 [11]，这里则直接聚焦于证明所需的步骤。我们将 $\mathcal{M}_t : (S \times A) \rightarrow \mathcal{R}_{t+1}$记作神经网络，其中 表示残差。该网络等价于 μ_{NN}，但最后一个偏置向量被 $\tau_{opt}(t+1)$ 平移。假设 $\mathcal{R}_{t+1}, \mathcal{R}^c_{t+1}$是残差可达集，当分别引入 $(S_t \times A_t)$ 和 $(S_t \times A^c_t)$ 时，$\mathcal{M}_t$（$A^c_t \subset A_t$）。这意味着，如果 $a_t \in A^c_t$，那么 $a_t \in A_t$，因此，对于$A_t = \mathcal{E}(\mu_{a_t}, \Omega_{a_t})$,

$$\begin{bmatrix} a_t \\ 1 \end{bmatrix}^\top \begin{bmatrix} -\Omega^{-1}_{a_t} & \Omega^{-1}_{a_t}\mu_{a_t} \\ \mu^\top_{a_t}\Omega^{-1}_{a_t} & -\mu^\top_{a_t}\Omega^{-1}_{a_t}\mu_{a_t} + 1 \end{bmatrix} \begin{bmatrix} a_t \\ 1 \end{bmatrix} \geq 0,$$

这足以说明，对于优化问题 (22)，由 $(S_t \times A_t)$ 得到的 $\mathcal{R}_{t+1}$ 的最优上界，实际上是由 $(S_t \times A^c_t)$ 得到的 $\mathcal{R}^c_{t+1}$ 上界的一个可行解（参见附录 E 中对 [7,11] 的总结）。这意味着第二个问题中优化问题 (22) 的最优目标函数值（以 $(S_t \times A^c_t)$ 为输入）小于第一个问题中优化问题 (22) 的最优目标函数值（以 $(S_t \times A_t)$ 为输入）。因此，我们得出 Logdet($\Omega_{\mathcal{R}^c_{t+1}}$) <Logdet($\Omega_{\mathcal{R}_{t+1}}$)。换言之，我们得出，用 A^c_t替换 A_t会使残差可达集的上界更小。

现在假设修改后的动作在优化 (7) 中的最优置信区域 A_t 包含非空子集 $A^c_t \subset A_t$。然而，我们知道，通过将 A_t 替换为 A^c_t，残差可达集的椭球边界仍可进一步缩小。这与我们的结论相矛盾，因为我们已经得出结论，A_t 会使残差可达集的上界最小。因此，最优置信区域 A_t不包含任何子集，而是一个单元素集。换言之，tr(Ω_{a_t}) $= 0$。

附录 B：执行器界的构造

所提出的动作应位于以下超矩形内：$\boldsymbol{\ell} \leq a_t \leq \boldsymbol{u}$。在附录 A 中，我们证明了优化问题 (11) 的解是一个单元素集$A_t = \{a_t\}$，因此忽略形状矩阵 $\tau_2 U^{-1}_{a_t}$，仅对上述超矩形 $\boldsymbol{\ell} \leq U^{-1}_{a_t} V_{a_t} \leq \boldsymbol{u}$中的均值进行界定，这意味着，$U_{a_t}\boldsymbol{\ell} \leq V_{a_t} \leq U_{a_t}\boldsymbol{u}$

附录 C. 定理1的证明：

根据 [7] ，我们知道椭球 $\mathcal{E}(0,\Omega)$ 包含残差可达集的充分条件为:

$$\tau_1 M_{s_t} + E_2^\top \begin{bmatrix} -U_{a_t} & V_{a_t} \\ V_{a_t}^\top & -V_{a_t}^\top U_{a_t}^{-1} V_{a_t} + \tau_2 \end{bmatrix} E_2 + M_\phi - M_{out} \leq 0 \tag{14}$$

我们将线性项移至右侧，并将非线性项保留在左侧，

$$\begin{aligned} E_2^\top \begin{bmatrix} 0 & 0 \\ 0 & -V_{a_t}^\top U_{a_t}^{-1} V_{a_t} \end{bmatrix} E_2 \leq & -\tau_1 M_{s_t} - E_2^\top \begin{bmatrix} -U_{a_t} & V_{a_t} \\ V_{a_t}^\top & \tau_2 \end{bmatrix} E_2 \\ & - M_\phi + M_{out} \end{aligned} \tag{15}$$

不等式左侧的矩阵是负定的，因此，如果我们引入新的约束，

$$-\tau_1 M_{s_t} - E_2^\top \begin{bmatrix} -U_{a_t} & V_{a_t} \\ V_{a_t}^\top & \tau_2 \end{bmatrix} E_2 - M_\phi + M_{out} \geq 0 \tag{16}$$

我们便满足了式 (15) 中所需的约束。根据我们的观测，这个新的线性约束不会带来保守性，因为 $V_{a_t}^\top U_{a_t}^{-1} V_{a_t}$ 的值始终接近于零，因此我们忽略了非线性项的负值。正如前文所述，我们知道 $\Omega_{a_t} = \tau_2 U_{a_t}^{-1}$ 收敛于零，这意味着 U_{a_t} 有可能变得无界，这也是无穷小的 Ω_{a_t} 对凸优化而言会造成问题的原因。为了避免 U_{a_t} 出现无界解，我们为 $\text{tr}(\Omega_{a_t})$ 设定了一个较小的下界。我们知道 Ω_{a_t} 是正定矩阵，因此，如果 $\text{tr}(\Omega_{a_t}^{-1})$ 小于一个较大的数 σ，就足以使 $\text{tr}(\Omega_{a_t})$ 大于一个较小的数（ A_t 大小的下界）。这可以改写为凸约束 $\text{tr}(U_{a_t}) \leq \tau_2 \sigma$，或者换句话说，$\text{tr}(U_{a_t})\delta \leq \tau_2$ ，其中 $\delta = \frac{1}{\sigma}$ 最好是一个较小的数。因此，为了说明凸约束 $\text{tr}(U_{a_t})\delta \leq \tau_2$ 的必要性，我们指出这只是一项预防措施（δ 非常小），用于避免出现无界解。

附录 D. 比较

由于存在对抗样本，我们需要证明修改后的动作在部署环境中的表现优于自主智能体。由于无法直接利用该环境来实现这一目的，我们必须为部署环境构建一个深度代理模型。另一方面，阅读 [9] 可知，尽管深度网络具有较高的准确性，但在定理1的凸规划中会导致明显的保守性。因此，我们在高度非线性的环境中训练两个代理网络。前者通过第3.1节中的嵌入式技术获得，并用于凸规划。后者是一个非常深的网络，可确保对 π^* 和 a_t^c进行准确比较时的可靠性。我们将这个深度神经网络称为 μ_{NN}^*。

附录 E. [7] 的简要总结

为了进一步说明，我们给出针对特定问题的解的概要及其参数。假设状态和动作的置信区域 $s_t \in S_t$、 $a_t \in A_t$ 是固定且已知的。那么， [7,11] 中提供的工具会针对残差的可达集$\mathcal{R}_{t+1}$提出一个凸优化问题，以求得最紧的椭球上界。在本研究中，我们增加了另一个约束，并将上述上界的中心固定在原点，以确保残差的欧几里得范数减小。因此，我们使用工具 [7,11]来求得满足 $\mathcal{R}_{t+1} \subset \mathcal{E}(0, \Omega^{-1}_{\mathcal{R}_{t+1}})$ 的最紧上界。我们知道 $a_t \in A_t := \mathcal{E}(\mu_{a_t}, \Omega_{a_t})$，因此：

$$\begin{bmatrix} a_t \\ 1 \end{bmatrix}^\top \underbrace{\begin{bmatrix} -\Omega^{-1}_{a_t} & \Omega^{-1}_{a_t}\mu_{a_t} \\ \mu^\top_{a_t}\Omega^{-1}_{a_t} & -\mu^\top_{a_t}\Omega^{-1}_{a_t}\mu_{a_t}+1 \end{bmatrix}}_{Q_1} \begin{bmatrix} a_t \\ 1 \end{bmatrix} \geq 0 \tag{17}$$

我们还知道 $s_t \in S_t := \mathcal{E}(\mu_{s_t}, \rho_n \Sigma_{s_t})$，因此：

$$\frac{1}{\rho_n}\begin{bmatrix} s_t \\ 1 \end{bmatrix}^\top \underbrace{\begin{bmatrix} -\Sigma^{-1}_{s_t} & \Sigma^{-1}_{s_t}\mu_{s_t} \\ \mu^\top_{s_t}\Sigma^{-1}_{s_t} & -\mu^\top_{s_t}\Sigma^{-1}_{s_t}\mu_{s_t}+\rho_n \end{bmatrix}}_{Q_2} \begin{bmatrix} s_t \\ 1 \end{bmatrix} \geq 0, \tag{18}$$

在下一次尝试中， [7] 建议我们将残差模型 $\mathcal{M}_t$ 中的所有激活后变量拼接起来，并生成向量 $\mathbf{x} = [z^{1\top}, z^{2\top}, \cdots, z^{\mathcal{L}-1\ \top}]^\top$。随后，他们提出一个满足二次约束的对称矩阵 Q_ϕ，

$$\begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}^\top Q_\phi \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} \geq 0. \tag{19}$$

最终目标是证明残差 $\Delta_{t+1} \in \mathcal{E}(0, \Omega_{\mathcal{R}_{t+1}})$。因此，定义 $\Omega = \Omega^{-1}_{\mathcal{R}_{t+1}}$，我们应提出一个能够推出以下结论的约束：

$$\begin{bmatrix} r_{t+1} \\ 1 \end{bmatrix}^\top \begin{bmatrix} -\Omega & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} r_{t+1} \\ 1 \end{bmatrix} \geq 0 \tag{20}$$

为了提出这样的约束， [7] 中的作者建议定义基向量 $\mathbf{z} = [s_t^\top,\ a_t^\top, \mathbf{x}^\top, 1]^\top$，并将线性变换矩阵 E_1、 E_2、 E_3以及矩阵 C 定义为：

$$\begin{bmatrix} s_t \\ 1 \end{bmatrix} = E_1\mathbf{z},\ \begin{bmatrix} a_t \\ 1 \end{bmatrix} = E_2\mathbf{z},\ \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} = E_3\mathbf{z},\ \begin{bmatrix} r_{t+1} \\ 1 \end{bmatrix} = \begin{bmatrix} C & b \\ 0 & 1 \end{bmatrix}\mathbf{z},$$

且 $C = \begin{bmatrix} 0\ 0\ \cdots\ W_\ell \end{bmatrix}$, $b = b_\ell - \tau_{opt}(t+1)$,$(b_\ell, W_\ell) \in \theta_\mu$ 表示 μ_{NN}中最后一层的偏置向量和权重，并加上方程 (17)、(18)、(19) 的左侧，从而得到以下不等式：

$$\mathbf{z}^\top \left(\tau_1 \underbrace{E_1^\top Q_1 E_1}_{M_{s_t}} + \tau_2 \underbrace{E_2^\top Q_2 E_2}_{M_{a_t}} + \underbrace{E_3^\top Q_\phi E_3}_{M_\phi} \right) \mathbf{z} \geq 0, \tag{21}$$

对于某些 $\tau_1, \tau_2 \geq 0$。因此，如果该不等式，

$$\mathbf{z}^\top \left(\tau_1 M_{s_t} + \tau_2 M_{a_t} + M_\phi\right) \mathbf{z} - \begin{bmatrix} r_{t+1} \\ 1 \end{bmatrix}^\top \begin{bmatrix} -\Omega & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} r_{t+1} \\ 1 \end{bmatrix} \leq 0$$

成立，则约束 (20) 得到满足。该约束可改写为,

$$\mathbf{z}^{\top}\left(\tau_1 M_{s_t}+\tau_2 M_{a_t}+M_{\phi}-\underbrace{\begin{bmatrix} C & b_{\mathcal{L}} \\ 0 & 1 \end{bmatrix}^{\top}\begin{bmatrix} -\Omega & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} C & b_{\mathcal{L}} \\ 0 & 1 \end{bmatrix}}_{M_{out}}\right)\mathbf{z}\leq 0$$

因此，应用假设 $\tau_1 M_{s_t}+\tau_2 M_{a_t}+M_{\phi}-M_{out}\leq 0$ 足以但并非必要地声称（20）得到满足。因此，凸优化:

$$\begin{cases} \min\limits_{M_{\phi},\tau_1,\tau_2} -\mathbf{Logdet}(\Omega) \\ \mathbf{s.t.}\ \tau_1 M_{s_t}+\tau_2 M_{a_t}+M_{\phi}-M_{out}\leq 0 \end{cases} \tag{22}$$

给出了以原点为中心、覆盖残差可达集 $\mathcal{R}_{t+1}$的次优最紧椭球上界。