arXiv:2303.04431v1 [cs.RO] 8 Mar 2023

通过神经网络修复实现辅助设备中的安全机器人学习

Keyvan Majd[1†], **Geoffrey Clark**[1], **Tanmay Khandait**[1], **Siyu Zhou**[1],
Sriram Sankaranarayanan[2], **Georgios Fainekos**[3], **Heni Ben Amor**[1]

[1]亚利桑那州立大学， [2]科罗拉多大学博尔德分校， [3] Toyota NA-R&D

[†]majd@asu.edu

摘要： 由于需要个性化设计，且人机交互动力学难以建模，辅助机器人设备是神经网络（NN）应用中尤具发展前景的领域。然而，基于神经网络的估计器和控制器可能会对先前未见过的数据点产生潜在不安全的输出。本文介绍一种算法，用于更新神经网络控制策略，使其满足给定的一组形式化安全约束，同时优化原始损失函数。给定一组混合整数线性约束，我们将神经网络修复问题定义为混合整数二次规划（MIQP）。通过大量实验，我们展示了该修复方法在为小腿假肢生成安全策略方面的有效性。

关键词： 模仿学习、辅助机器人技术、安全性、假肢

1 引言

机器人学习有望彻底改变可穿戴机器人设备领域，例如假肢、矫形器和外骨骼 [1]。这类设备通常依据平均人群统计数据建立模型，并采用“通用尺寸”设计。然而，机器学习技术有助于根据穿戴者的个体特征、运动模式或生物力学特性，自动调整控制参数，从而显著提升人体工学舒适度和生活质量。尽管有这些益处，机器学习，尤其是深度学习 [2] 在该领域的应用仍然有限，主要原因是安全方面的顾虑。例如，对于假肢，可能需要确保在一组测试条件下，生成的控制值不会超过最大阈值。其他安全约束还包括对速度和关节角度的限制，以及对控制输入变化幅度的限制。

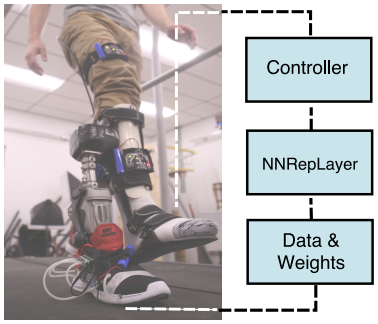


图1：运行神经网络进行控制的小腿假肢。神经网络修复过程可确保满足安全约束。

本文提出了一种算法，用于训练神经网络控制器，使其在拟合给定训练/测试数据的同时，满足指定的安全规范。具体而言，我们运用该方法为机器人小腿假肢设计控制器，使其满足基本安全条件（见图 1）。我们提出的方法以一个经过训练的网络（例如，使用训练数据通过反向传播得到的网络）以及一项规范作为输入，该规范对给定输入集所对应的可能输出施加限制。随后，该方法通过确定性全局优化生成一组修改后的权重，使输出满足所需的安全约束。我们为该技术提供了最优性理论保证（假设不存在数值误差）。此外，我们通过真实世界的机器人实验表明，所提出的方法能够为小腿假肢生成安全的神经策略，并满足多种约束。

第六届机器人学习会议（CoRL 2022），奥克兰，新西兰。

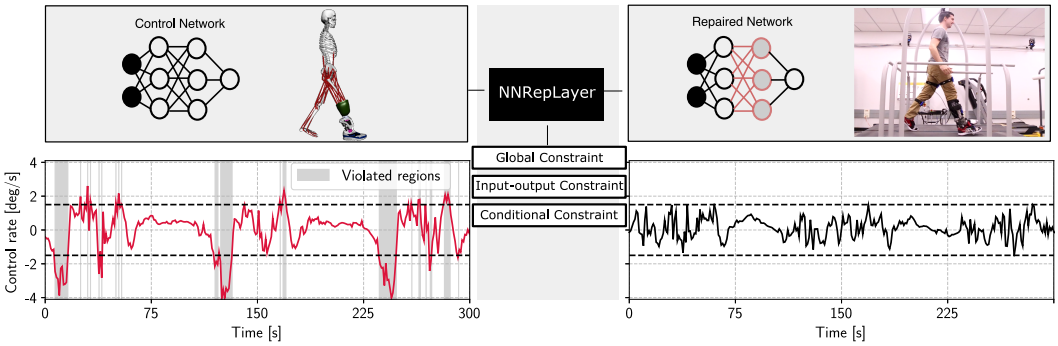


图2：我们方法的概述。左：给定一个用于控制假肢的（不安全）神经网络。然而，其输出违反了形式化安全约束。右：使用我们的神经网络修复策略，我们确定一组经过调整的神经网络权重，从而消除所有违规，同时仍保持控制器的底层行为。

贡献。 所提出的方法可以修复网络中的任意层，从而扩大解空间，提升成功修复的可行性。更重要的是，该方法在成功修复所有已发现的不安全样本方面具有理论保证。与重新训练或微调方法相比，它还有两项显著优势：（1）无需修改训练数据来使其满足约束；（2）不使用梯度优化方法，因为这类方法即使面对已发现的不安全数据点，也无法保证约束满足。最后，与其他属性驱动修复工作相比，即 [3, 4, 5, 6, 7], 该方法首次应用于真实物理系统，即动力假肢设备，而非模型。

相关工作。 假肢的基于学习的控制源于对人-假肢动力学建模所面临的挑战，该动力学呈现时变行为。文献中提出了若干利用基于学习的方法控制假肢设备的不同方案 [8, 9]。不过，这篇简短综述仅关注基于神经网络（NN）的方法。Gao 等人 [10] 提出了一种基于循环神经网络的假肢控制器，并表明这类控制器能够有效地减小动力假肢设备上期望轨迹与实际轨迹之间的差异。另一方面，Keles、、Yucesoy [11] 以及 Vonsevych 等人[12] 着重利用肌电图（EMG）信号预测假肢踝关节和手的控制参数。我们的工作采用类似的方法，利用神经网络预测踝关节控制参数，再通过比例-微分控制器驱动假肢。神经网络验证已得到广泛研究，旨在检查给定神经网络作为独立组件 [13, 14, 15, 16]（综述见 [17]）或作为闭环系统一部分 [18, 19, 20]时是否满足规范。测试技术可以为各种基于神经网络的应用生成反例（对抗样本），例如 [21, 14, 22, 23, 24, 25]。确保神经网络满足给定安全属性的一种方法，是基于反例进行重新训练和微调 [26, 27, 7]。然而，这种方法存在若干缺陷。首先，需要有反例对应的标签，这可能需要进一步进行数据收集，而这通常十分繁琐。此外，基于梯度下降的优化方法无法保证结果满足所提供的约束。相比之下，我们的方法无需生成对抗样本，也无需执行梯度下降。因此，“修复”网络的问题——即以最小幅度修改网络权重以满足某些安全属性——比仅仅用包含反例的额外数据重新训练网络更为复杂。文献中的一些方法 [3, 4, 5, 6] 已朝着修复神经网络的目标取得进展。Goldberger 等人 [3] 的方法只能修复输出层，这大幅缩小了可能成功修复的范围（而且在许多情况下甚至无法修复）。Fu 和 Li [4] 的方法会针对输入空间的某些划分生成补丁，并修改相应的输出。对神经网络输出添加补丁的思路很有前景；然而，由于该方法需要将输入空间划分为仿射子区域，因此其在高维输入下的扩展能力尚不明确。文献 [6] 中的作者利用解耦深度神经网络架构，在故障网络的权重空间中修复与每个故障样本相关的线性区域。然而，该方法会导致修复后的网络不连续。因此，它不适用于本文的目标应用——机器人学习和控制。此外，该方法也不适用于输入数量超过三个的网络。

2 问题形式化

不失一般性，我们以学习用于小腿假肢的安全机器人控制器为例，阐述并讨论我们的方法。这项任务的目标是学习一种策略 π_θ，根据一组传感器值生成动力假肢踝关节角度的控制值。然而，最关键的是，策略 π_θ 必须满足一组安全约束 Ψ。图 2 概述了我们应对这一挑战的方法。目前，我们假设可能存在一个不安全的先验策略网络，见图 2 （左）。网络参数 θ 可以通过模仿学习 [10], 、强化学习 [28] 或任何常见的机器学习范式学习得到。策略可以针对任务效率进行优化，例如实现稳定且低耗力的步态，但可能尚未满足任何安全约束。我们的目标是找到一组调整后的网络参数，使其满足所有此类约束。例如，我们现在可以选择将控制速率限制在特定范围内。在验证数据集的输入值上运行网络后，可以发现某些时间步存在违规，如图 2所示。我们的方法称为**NNRepLayer**（逐层神经网络修复），它接收原始网络参数 θ 和谓词 Ψ，并生成不会导致任何违规的更新后参数。

符号约定。 我们用 $\{a_n\}_{n=1}^N$表示变量集合 $\{a_1, a_2, \cdots, a_N\}$。设 π_θ 是具有 L 个隐藏层的策略网络。每一层 $l \in \{l\}_{l=0}^L$ 的节点表示为 x^l，其中$|x^l|$ 表示层 l 的维度（x^0 表示网络的输入）。网络输出也表示为 y 或 $\pi_\theta(x^0)$。我们考虑具有权重和偏置项 $\{(\theta_w^l, \theta_b^l)\}_{l=1}^{L+1}$ 的全连接策略网络。由 N 个输入 x_n^0 和目标输出 t_n 组成的训练数据集表示为$\{(x_n^0, t_n)\}_{n=1}^N$，这些数据从输入输出空间 $\mathcal{X} \times \mathcal{T} \subseteq \mathbb{R}^{|x^0|} \times \mathbb{R}^{|t|}$中采样。我们用 x_n^l 表示样本 n 在第 l 层的节点向量。本文关注使用修正线性单元（ReLU）激活函数 $R(z) = \max\{0, z\}$的策略网络。因此，给定 n^{th} 样本，在第 l^{th} 个隐藏层中，有 $x^l = R\left(\theta_w^l x^{l-1} + \theta_b^l\right)$。最后一层不应用激活函数，即 $y = \theta_w^{L+1} x^L + \theta_b^{L+1}$。

问题陈述（修复问题）。设 π_θ 是在训练输入-输出空间 $\mathcal{X} \times \mathcal{T} \subseteq \mathbb{R}^{|^0|} \times \mathbb{R}^{|t|}$ 上训练好的策略网络，$\Psi(y, x^0)$ 是网络输出 y 上针对一组感兴趣输入 $x^0 \in \mathcal{X}_r \subseteq \mathcal{X}$ 的谓词。修复问题是修改 π_θ 的权重 θ_w 和偏置项θ_b，使修复后的策略 π_{θ_r} 满足 $\Psi(y, x^0)$。

策略网络的修复不仅应满足谓词 $\Psi(y, x^0)$，还应保持原始策略的性能。为满足后者，[3] 中提出的方法通过将最后一层权重的偏差限制在最小范围内，确保谓词得到满足。然而，正如我们随后在实验结果中所示，仅修复最后一层不一定可行，也不一定足以在与原始参数的偏差最小的情况下满足这些谓词。此外，与原始权重的最小偏差并不能充分保证网络保持原有性能。众所周知，权重的细微变化可能导致网络性能显著偏离原有水平 [29]。因此，修复后的策略 π_{θ_r} 还应尽量降低相对于其原始训练数据的损失。我们提出 NNRepLayer（逐层神经网络修复），通过修复策略网络的特定层来满足谓词 $\Psi(y, x^0)$，同时尽量降低训练损失。

3 NNRepLayer

我们可以将框架表述为如下最小化问题：最小化损失函数 $E(\theta_w, \theta_b)$，同时满足 $(x^0, t) \in \mathcal{X} \times \mathcal{T}$和 $\Psi(y, x^0)$，其中 $x^0 \in \mathcal{X}_r$。然而，由于 ReLU激活函数的非凸性，以及权重/偏置变量相关项相乘所产生的高阶非线性约束，最终得到的优化问题是非凸的，难以求解。我们的方法通过只专注于修复单个层来获得次优解。因此，我们修改单个层的权重和偏置项，以调整预测结果，从而最小化 $E(\theta_w, \theta_b)$ 并满足 $\Psi(y, x^0)$。由此，我们求解以下问题

问题 1. 令 π_θ 表示一个训练好的策略网络，它在训练输入-输出空间 $\mathcal{X} \times \mathcal{T} \subseteq \mathbb{R}^{|x^0|} \times \mathbb{R}^{|t|}$ 上具有 L 个隐藏层；令 $\Psi(y, x^0)$ 表示一个谓词，用于描述对感兴趣输入集合 $x^0 \in \mathcal{X}_r \subseteq \mathcal{X}$中 π_θ 的输出 y 的约束。
NNRepLayer 修改一

层 $l \in \{1, \cdots, L+1\}$ 中的权重，使得新网络 π_{θ_r} 满足 $\Psi(y, x^0)$，同时最小化网络 $E(\theta_w^l, \theta_b^l)$ 相对于其原始训练集的损失。

由于 $\mathcal{X}_r$ 和 $\mathcal{X}$ 不一定是凸的，我们在数据集$\{(x_n^0, t_n)\}_{n=1}^N \sim \mathcal{X} \times \mathcal{T} \cup \mathcal{X}_r \times \tilde{\mathcal{T}}$上构造 NNRepLayer，其中 $\tilde{\mathcal{T}}$是 $\mathcal{X}_r$中输入所对应的原始目标值集合。在 $x^0 \in \mathcal{X}_r$ 上定义的谓词 $\Psi(x^0, y)$ 不一定与$\mathcal{T}$中的目标值兼容。这意味着，该谓词可能会约束 $\mathcal{X}_r$ 输入空间中的神经网络输出，从而使输入 $x^0 \in \mathcal{X}_r$ 无法达到其在 $\tilde{\mathcal{T}}$中的目标值。这是许多应用中的一种自然约束。例如，出于安全约束的考虑，对于给定的不安全输入状态集合，我们可能不允许神经网络控制器遵循其原始控制参考。对于给定的层 l，我们还将 $E(\theta_w^l, \theta_b^l)$定义为平方和损失 $E(\theta_w^l, \theta_b^l) = \sum_{n=1}^N \|y_n(x_n^0, \theta_w^l, \theta_b^l) - t_n\|_2^2$的形式，其中 $\|\cdot\|_2$ 表示欧几里得范数。这里，由于我们只修复目标层 l的权重和偏置项，损失项 E 分别是 θ_w^l 和 θ_b^l的函数。因此，除目标层 l 之外的所有层的权重和偏置项均固定不变。我们的优化形式定义如下。

NNRepLayer优化形式。设 π_θ 是一个具有 L 个隐藏层的神经网络，$\Psi(y, x^0)$ 是一个谓词，且 $\{(x_n^0, t_n)\}_{n=1}^N$ 是从集合 $\mathcal{X}$、$\mathcal{X}_r$、$\mathcal{T}$ 和 $\mathcal{T}$ 上的 $(\mathcal{X} \times \mathcal{T}) \cup \tilde{(}\mathcal{X}_r \times \mathcal{T})$ 中采样得到的输入输出数据集，所有定义均同问题 1。NNRepLayer 通过修改 θ_w^l 和 θ_b^l，在满足约束 (2)–(5) 的条件下，最小化损失 (1)。

这里，约束（2）表示网络最后一层的线性前向传播。约束（3）表示从第l层开始的隐藏层前向传播。除了 l^{th} 层的权重和偏置项，即 θ_w^l 和 θ_b^l之外，后续各层 $\{(\theta_w^i, \theta_b^i)\}_{i=l+1}^{L+1}$ 的权重和偏置项均固定。对于所有 N 个样本$\{n\}_{n=1}^N$，x_n^{l-1}的样本值由其从 x_n^0 开始的前序层节点的加权和得到。每个 ReLU 节点 x^l 均通过大 M 公式进行建模 [30, 31]：

$$
\begin{array}{lll}
(1) & \min\limits_{\theta_w^l,\ \theta_b^l,\ \delta,\ y_n,\ \{x_n^i\}_{i=l}^L,\ \{\phi_n^i\}_{i=l}^L} & E(\theta_w^l,\ \theta_b^l) + \delta, \\
& \text{s.t.} & \\
(2) & y_n = \theta_w^{L+1} x_n^L + \theta_b^{L+1}, & \\
(3) & x_n^i = R(\theta_w^i x_n^{i-1} + \theta_b^i), & \text{对于 } \{i\}_{i=l}^L \\
(4) & \Psi(y_n,\ x_n^0), & \text{对于 } x_n^0 \in \mathcal{X}_r \\
(5) & \delta \geq \|\theta_w^l - \theta_w^{l,init}\|_\infty,\ \|\theta_b^l - \theta_b^{l,init}\|_\infty & \geq 0.
\end{array}
$$

$x^l \geq \theta_w^l x_n^{l-1} + \theta_b^l$，$x^l \leq \left(\theta_w^l x_n^{l-1} + \theta_b^l\right) - lb\,(1-\phi)$，以及 $x^l \leq ub\,\phi$，其中 $x^l \in [0, \infty)$，而$\phi \in \{0, 1\}$ 决定节点 x^l的激活状态。边界 lb、$ub \in \mathbb{R}$ 称为大 M 系数，$\theta_w^l x_n^{l-1} + \theta_b^l \in [lb, ub]$，为提升混合整数二次规划求解器的性能，这些系数应尽可能紧。我们使用区间算术（IA）方法 [32, 14] 为 ReLU 节点求取紧致边界（有关 IA 的更多细节，请参阅补充材料第 ?? 节）。约束（4）是定义在 $x^0 \in \mathcal{X}_r$上的 y 的给定谓词。NNReplayer 处理形式为$\bigvee_{c=1}^C \psi_c\,(x^0,\ y)$ 的谓词，其中 C 表示析取命题的数量，ψ_i 是关于 x^0 和 y的仿射函数。最后，约束（5）将权重和偏置项 θ_w^l、θ_b^l与原始的 $\theta^l{}_w^{init}$和 $\theta^l{}_b^{init}$ 之间的逐项最大范数误差限制在 δ以内。考虑到二次损失函数 $E\,(\theta_w^l,\ \theta_b^l)$ 以及 $\Psi\,(y_n,\ x_n^0)$ 和 $R\,(\theta_w^i x_n^{i-1} + \theta_b^i)$ 的仿射析取形式，我们将 NNRepLayer 求解为混合整数二次规划（MIQP）。

定理 *1.* 给定谓词 $\Psi(y, x^0)$，以及从 $(\mathcal{X} \times \mathcal{T}) \cup (\mathcal{X}_r \times \tilde{\mathcal{T}})$ 中抽取的输入-输出数据集 $\{(x_n^0, t_n)\}_{n=1}^N$，其定义在集合 $\mathcal{X}$、$\mathcal{X}_r$、$\mathcal{T}$ 和 $\tilde{\mathcal{T}}$ 上（如问题 *1* 中所定义），假设 θ_w^l 和 θ_b^l 是 *(1)-(5)* 的可行解。那么，对于所有输入样本 x_n^0，$\Psi(\pi_{\theta_r}(x_n^0)\ x_n^0)$ 均满足。

证明。由于可行解 θ_w^l 和 θ_b^l 满足针对修复数据集（4）的硬约束$\{(x_n^0, t_n)\}_{n=1}^N$，$\Psi(\pi_{\theta_r}(x_n^0)\ x_n^0)$ 得以满足。 □

由定理 1可知，下面的推论显而易见。

推论 *1.* 给定谓词 $\Psi(y, x^0)$，以及从 $(\mathcal{X} \times \mathcal{T}) \cup (\mathcal{X}_r \times \tilde{\mathcal{T}})$ 中采样得到的输入输出数据集 $\{(x_n^0, t_n)\}_{n=1}^N$，其定义在问题 *1* 所述的集合 $\mathcal{X}$、$\mathcal{X}_r$、$\mathcal{T}$ 和 $\tilde{\mathcal{T}}$ 上。假设 $\theta_w^{l^*}$ 和 $\theta_b^{l^*}$ 是 *NNRepLayer* 的最优解。那么，对于来自 $\{(x_n^0, t_n)\}_{n=1}^N$ 的所有输入样本 x_n^0，$\Psi(\pi_{\theta_r}(x_n^0)\ x_n^0)$ 均满足。

4 评估

我们探讨该框架在满足以下三类约束方面的适用性。**全局约束** 用于对网络输出施加全局界限，即 $y \in [y_{min}, y_{max}]$。**输入输出约束**确保网络输出 y 相对于网络输入保持在一定范围内 $\{\psi_c\ x^0\ y \leq 0\}_{c=0}^{C}$，即(，)，其中是约束的数量，是关于 x^0 和 y 的仿射函数。最后，**条件约束**用于编码如下所述的 if-then-else 约束 $\{\psi_c(x^0,\ y) \leq 0$，如果 $x^0,\ y \in S_c\}_{c=0}^{C}$，其中 C 指定条件的数量，ψ_c 是关于 x^0 和 y的仿射函数，且 $S_c \subseteq \mathcal{X} \times \mathcal{T}$。我们设计了多项实验，以验证我们的修复框架能否成功将这些约束应用于策略网络。遵循我们的研究动机，所有实验均针对图 2 中介绍的假肢步态生成任务进行。通过这些实验，我们旨在回答以下问题：(1) 我们的方法能否使假肢装置处理上述三种类型的约束？(2) 能否

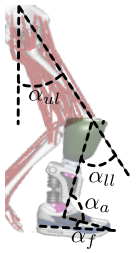


图3：假肢装置模型。

修复后的控制器能否成功应用于真实行走场景？（3）通过我们的方法修复后的策略，对未见过的违反约束的样本有多强的鲁棒性？

实验设置。在我们的实验中，我们训练了一个策略网络 π_θ 来控制假肢，随后对其进行修复，以确保其符合安全约束。为此，我们首先采用模仿学习 [33] 策略训练模型。为收集数据，我们开展了一项经机构审查委员会（IRB）批准的研究，记录一名未佩戴任何假肢的健康受试者的步态。步行数据包括三个通过绑带固定在大腿（股骨）、小腿（胫骨）和足部的惯性测量单元（IMU）。这些 IMU 以 100Hz 采集各肢体部分在世界坐标系中的角度和角速度。踝关节角度 α_a 由足部和下肢惯性测量单元的数据经后处理计算得出。随后，我们训练神经网络，根据大腿和小腿惯性测量单元的传感器值生成踝关节角度。具体而言，神经网络模型接收来自大腿和小腿传感器的角度与速度（网络输入 x^0），α_{ul}、$\dot{\alpha}_{ul}$、α_{ll}、$\dot{\alpha}_{ll}$，以预测踝关节角度 α_a（网络输出y）；该输出随后用作假肢上比例-微分控制器的控制参数。图 3 展示了各传感器读数。我们使用输入变量的滑动窗口，记为 dt（在所有实验中$dt = 10$），以考虑控制参数受到的时间影响，并适应传感器读数中的噪声。因此，网络的输入为 $dt \times |x^0|$，更具体地说，是当前及之前的 dt 个传感器读数。在所有实验中，我们训练了一个具有三个隐藏层的深度策略网络，每个隐藏层包含 32 个 ReLU 节点。网络完成训练后，我们评估策略是否违反约束，并收集样本供 NNRepLayer 使用。我们在策略网络的最后一层和倒数第二层上测试 NNRepLayer，使用原始训练数据的一个子集（包含对抗样本和非对抗样本）来满足约束。在所有实验中，我们在 NNRepLayer 中使用 150 个样本，并保留 2000 个样本作为测试集。最后，我们将经过修复以满足全局约束和输入-输出约束的策略应用于假肢装置，在其上步行 10 分钟进行测试，见图 5。具体而言，我们为同一名健康受试者安装了踝部旁路装置；该装置是一个模制于小腿上的碳纤维结构，其设计允许安装假肢踝关节，使健全受试者能够穿戴假肢行走，如图 2 所示。装置增加的重量及其偏轴定位会使受试者倾向于采用更慢且不对称的步态，产生偏离原始训练分布 [34, 10] 的步幅。随后，请参与者再次步行 10 分钟，以评估约束是否得到满足。

为了评估修复能否泛化到未见对抗样本，我们分析了违反程度。违反程度是通过网络输出相对于约束集的距离来衡量的。对于每项实验，我们都会说明如何计算输出与约束集之间的距离。我们将自己的框架与重新训练、微调[26, 35, 27, 7], 以及 [4] 中的基于补丁的修复方法（REASSURE）进行了比较。修复数据集中的对抗样本经过人工标注，以便用于微调和重新训练，使目标输出满足给定谓词。在微调中，按照 [26, 35], 提出的方法，我们使用收集到的对抗数据集，通过梯度下降并采用较小的学习率（10^{-4}），训练原始策略的所有参数。为避免对抗数据集上的过拟合，我们先训练顶层的权重，随后再对其余层进行少量训练轮次的微调。同样的人工标注

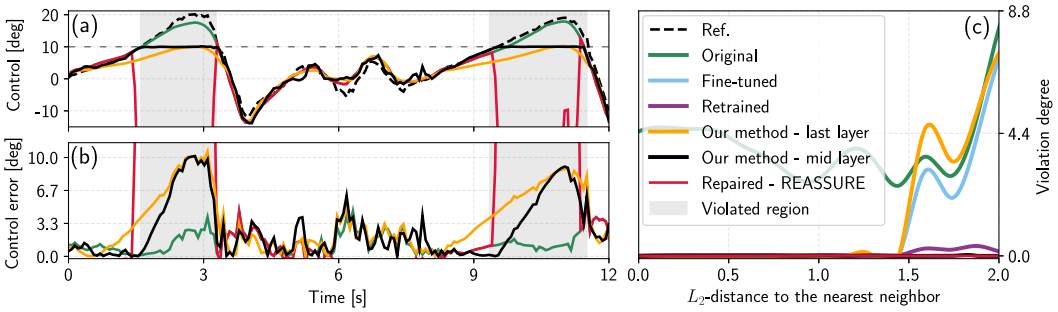


图 4：全局约束：(a) 踝关节角度，α_a；(b) 预测控制量与参考控制量之间的误差；(c) 违反程度与测试输入和修复样本输入之间的 L_2 距离的关系。

策略也应用于重新训练，但重新训练是针对所有原始训练样本从头开始训练一个新策略。在这两种方法中，我们都对策略进行训练，直到修复数据集中的所有对抗样本都满足网络输出上的给定谓词。我们的代码可在 GitHub 上获取：`https://github.com/k1majd/NNRepLayer.git`。

4.1 实验与结果

全局约束。 全局约束确保假肢控制（即 α_a）保持在一定范围内，并且不会输出意外的大值，以免扰乱用户的步行平衡。

此外，我们在这些场景中使用的假肢装置包含一个并联柔顺机构。因此，无论人体受试者还是机器人控制器，都可能使该机构达到硬限位，从而造成装置损坏。因此，在步行测试中，我们设定了全局约束，确保踝关节角度始终处于 $[-14, 24]$ [度] 的范围内，无论由人体还是机器人驱动。在仿真实验中，我们对踝关节角度 α_a 施加了人为的严格限位，使其绝不超过 $\alpha_a = 10$ [度]，这是一项更难满足的约束。我们将违反程度定义为

$$0 \quad \alpha_a \in [\alpha_a^{min}, \alpha_a^{max}] \qquad \{|\alpha_a - \text{若}, \alpha_a^{max}|, |\alpha_a - \alpha_a^{min}|\}$$

则为，否则为最小值。如图4(a)-(b)所示，修复后的网络成功满足了原故障区域内的约束，同时保持了跟踪性能

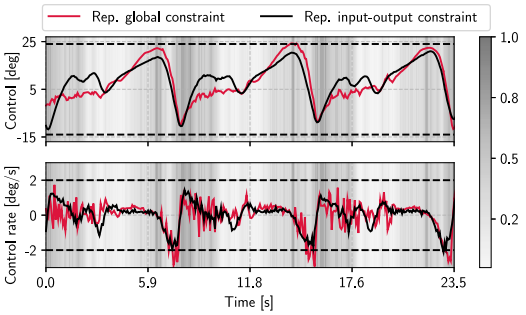


图 5：施加 $[-14, 24]$ 全局约束后的假肢行走测试结果：对控制量施加约束（红色所示），并将控制速率限制在 2 [deg/s]（黑色所示）。色条表示每个测试输入与修复集内最近邻之间的归一化 L_2 距离。

控制器在不受约束区域中的性能。4 （c）表明，在修复中间层后，即使是距离较远的、原本违反约束的数据点，其违反程度也几乎保持为零。图 5 中的红色控制信号也表明，我们的方法成功施加了控制边界$[-14, 24]$ 到实际的假肢行走测试中。

输入-输出约束。 深度神经网络作为高度非线性的函数逼近器，其输出变化速度可能超过机器人假肢或人体受试者所能适应的范围。因此，我们提出一项额外约束，限制控制动作从一个时间步到下一个时间步的变化。该约束既应能在存在传感器噪声时平滑控制动作，也应减少控制动作中的尖峰和振荡。为将此约束表示为输入-输出关系，我们将先前的 dt 控制动作 $\{\alpha_a(i)\}_{i=t-dt}^{t-1}$ 与上下肢传感器的数值一同作为输入，训练策略网络。此例中施加的约束形式为 $|\alpha_a(t) - \alpha_a(t-1)| \leq \Delta\alpha_a^{max}$。在假肢行走测试中，我们将控制速率限制为 $\Delta\alpha_a^{max} = 2$ [度/秒]，；在仿真中，我们测试了 $\Delta\alpha_a^{max} = 1.5$和 $\Delta\alpha_a^{max} = 2$ [度/秒]。我们的

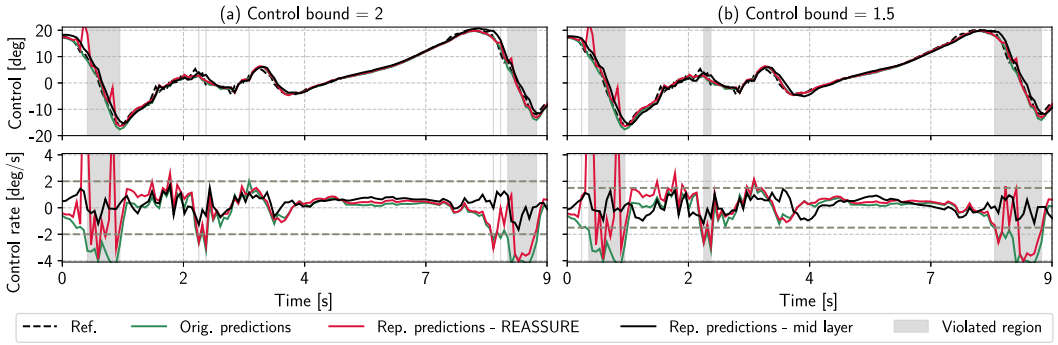


图6：输入输出约束：踝关节角度和踝关节角速度的边界 (a) $\Delta\alpha_a = 2$ 和 (b) $\Delta\alpha_a = 1.5$

图中的仿真结果 6 表明，NNRepLayer 满足控制速率的两个界限，从而使控制输出更加平滑。还可以观察到，NNRepLayer 成功保留了控制器的跟踪性能。在此实验中，我们将违反程度定义为 0，如果 $|\Delta\alpha_a(t)| \leq \Delta\alpha_a^{max}$且 $|\Delta\alpha_a(t) - \Delta\alpha_a^{max}|$；否则为其他值。图 7 (a) 表明，随着测试样本与修复集之间距离的增加，NNRepLayer 的违反程度几乎为零。在实际假肢行走测试中也获得了相同结果，如图 5所示。即使对于远离修复集的样本，也能满足输入输出约束（黑色信号）。最后，在此实验中，将 NNRepLayer 应用于最后一层，甚至无法获得优化问题（1）-（5）的可行解。

Co条件约束。 根据患者的人体工程学需求和病史 t, 主治骨科医生或假肢矫形师可能会识别出某些有害的身体姿势，例如，这些姿势可能会增加患骨关节炎或肌肉骨骼疾病的风险 [36, 37]。基于这一理念，我们定义了一个应当避免的关节角度空间区域 $\mathcal{S}$。图8展示了这样一个区域的示例：踝关节和股骨角度的关节空间中的灰色框 $\mathcal{S} = \{(\alpha_{ul}, \alpha_a) \mid \alpha_{ul} \in [-2, -0.5], \alpha_a \in [1, 3]\}$。为满足这一约束，应调整控制速率，使踝关节和股骨的关节角度保持在集合 $\mathcal{S}$之外。该约束可定义为条件命题 $\alpha_{ul} \in [-2, -0.5] \implies (\alpha_a \in (-\infty, 1]) \vee (\alpha_a \in [3, \infty))$，并可表述为网络输出上的线性不等式析取。对于每个给定的测试输入及其对应输出 α_a，违反程度定义为：若 α_a 位于框外，则为 α_a 到该框的距离；否则为 0。图8展示了使用 NNRepLayer 修复后新策略的输出。如图所示，我们的方法避免踝关节和股骨角度进入不安全区域 S。图7 (b) 也

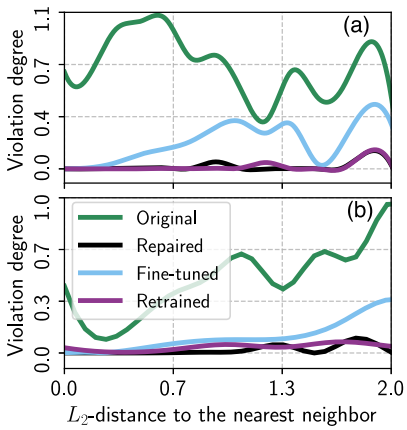


图7：违反程度与 L_2-距离的关系，其中距离是测试输入与修复样本输入之间的距离，分别对应 (a) 输入-输出约束和 (b) 条件约束情况。

展示了修复输入集之外的远距离测试输入样本的低输出违规程度。最后，我们观察到，修复最后一层并不能得到可行解。

与\微调、重新训练和 REASSURE 的比较。在每个实验中，我们展示了与微调、重新训练 [26, 35, 27, 7], 和 REASSURE 相比，我们方法的违反程度和控制信号 [4]。相较于 [4], , 尽管 REASSURE 保证了局部修复线性区域中约束的满足，我们表明该方法显著降低了网络在修复局部区域中的性能，见图4 和6。该方法无法针对给定故障样本处理输入输出约束，并且与我们的技术相比，它引入了多 500 倍的故障样本。REASSURE 也无法处理条件约束。不同于 REASSURE 所保证的

表1：表中报告了以下指标：RT：运行时间；MAE：修复后输出与原始输出之间的平均绝对误差；RE：得到修复的对抗样本百分比（修复效果）；IB：原本安全、但修复后出现故障的测试样本百分比（引入的错误）。各项指标为 50 次运行的平均值。

	NNRepLayer				REASSURE [4]			
	运行时间 [s]	MAE	修复误差 [%]	输入边界 [%]	运行时间 [s]	MAE	修复误差 [%]	输入边界 [%]
全局	233 ± 159	1.4 ± 0.11	99 ± 1	0.09 ± 0.20	14 ± 1	2.3 ± 0.78	97 ± 1	0
输入-输出	112 ± 122	0.5 ± 0.03	98 ± 1	0.19 ± 0.18	30 ± 8	0.6 ± 0.03	19 ± 4	85 ± 5
条件式	480 ± 110	0.35 ± 0.07	93 ± 2	0.11 ± 0.26	不可行	不可行	不可行	不可行
	微调				重新训练			
	运行时间 [s]	MAE	修复误差 [%]	输入边界 [%]	运行时间 [s]	MAE	修复误差 [%]	输入边界 [%]
全局	25 ± 13	1.2 ± 0.03	97 ± 4	0.95 ± 0.45	127 ± 30	1.4 ± 0.08	98 ± 3	0.65 ± 0.40
输入-输出	8 ± 2	0.6 ± 0.03	88 ± 2	2.47 ± 0.49	101 ± 1	0.5 ± 0.04	98 ± 1	0.28 ± 0.32
条件式	18 ± 3	0.7 ± 0.10	72 ± 5	0.27 ± 0.25	180 ± 2	0.31 ± 0.03	76 ± 2	0.12 ± 0.35

与修复样本处于同一线性区域内的样本满足约束，我们的技术仅保证修复样本满足约束。尽管我们通过实验证明了我们的技术在修复样本局部邻域内的泛化能力，但我们的方法并不能从理论上保证未见对抗样本满足约束。我们在补充材料第**??**节中提出了一种可靠算法，该算法可保证所有其他未见样本的安全性。表1 更好地说明了我们的方法在满足约束的同时保持控制性能方面的成功。

如表 1所示，对于全局约束和输入输出约束，重新训练和 NNRepLayer 在保持最小绝对误差以及将约束满足泛化至未见过的测试样本方面都表现出色。然而，对于重新训练和微调而言，满足条件分支约束颇具挑战，因为使用这些技术时，修复效果下降了近 30% 。这也凸显了我们的方法在将条件约束的满足泛化至未见案例方面的优势。有关比较结果的更多细节，请参阅补充材料第 **??** 节。

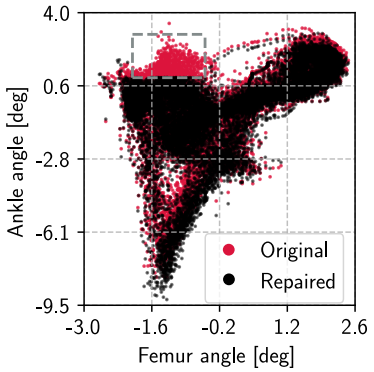


图8：强制满足条件约束，使股骨与踝关节的角度保持在灰色框之外。

5 结论与讨论

本文介绍了一种算法，用于训练满足一组形式化安全约束的安全神经网络控制器。我们提出的方法 NNRepLayer 通过执行全局优化步骤，对神经网络进行逐层修复权重。在真实机器人实验中，我们证明了所提出的方法能够为小腿假肢生成满足各种约束的安全神经策略。我们认为，这类方法对于以人为中心且安全关键型的机器人学习应用至关重要，例如下一代辅助机器人技术。

讨论。 所介绍的方法通常无法确保对于任意输入都满足约束。相反，它只能保证在修复时提供的所有数据点上满足这一性质。因此，必须谨慎确保修复过程涵盖应用领域中各种输入的代表性样本。从计算角度来看，求解 NNRepLayer 所依据的混合整数二次规划是一项计算开销很大的任务，其规模随网络大小而增长。在我们的实验中，我们成功修复了最多包含 256 个神经元的 NN 层，全局优化耗时从数分钟到 10 小时不等（详细实验结果请参阅补充材料第**??**节）。此外，我们在补充材料第**??**节中展示了，修复隐藏层中随机选取的子节点能够在短得多的时间内准确修复网络（速度比完整修复快 12 倍以上）。最后，我们的方法仅限于修复网络中的单个层。迭代修复多个层的初步结果颇具前景，未来将予以报告。不过，我们的方法无法同时修复多个层或整个网络。

致谢

本研究部分获得美国国家科学基金会资助，项目编号为 CNS-1932068、IIS-1749783 和 CNS-1932189。

参考文献

[1] M. Kyrarini、F. Lygerakis、A. Rajavenkatanarayanan、C. Sevastopoulos、H. R. Nambiappan、K. K. Chaitanya、A. R. Babu、J. Mathew 和 F. Makedon。医疗保健机器人综述。《*Technologies*》，9(1):8，2021。[2] Y. LeCun、Y. Bengio 和 G. Hinton。深度学习。*nature*，521(7553):436，2015。[3] B. Goldberger、G. Katz、Y. Adi 和 J. Keshet。利用验证对深度神经网络进行最小修改。发表于第 *23* 届程序设计、人工智能与推理逻辑国际会议，第 73 卷，第 260–278 页，2020。[4] F. Fu 和 W. Li。具有最小性和局部性保证的可靠且完备的神经网络修复。发表于第*10*届学习表征国际会议*(ICLR)*，2021。[5] X. Yang、T. Yamaguchi、H.-D. Tran、B. Hoxha、T. T. Johnson 和 D. Prokhorov。基于可达性分析的神经网络修复。发表于定时系统形式化建模与分析国际会议，第 221–236 页。施普林格，2022。[6] M. Sotoudeh 和 A. V. Thakur。深度神经网络的可证明修复。发表于第 *42* 届 *ACM SIGPLAN* 编程语言设计与实现国际会议，第 588–603 页，2021。[7] G. Dong、J. Sun、X. Wang、X. Wang 和 T. Dai。迈向正确修复神经网络。发表于*IEEE* 第 *21* 届软件质量、可靠性与安全国际会议 *(QRS)*，第 714–725 页，2021。[8] N. Thatte、N. Srinivasan 和 H. Geyer。动力型经股假肢的实时反应式绊倒规避。发表于机器人系统与科学 *(RSS)*，2019。[9] G. Clark、J. Campbell 和 H. B. Amor。用于假肢设备人体工学控制的预测模型学习。发表于 J. Kober、F. Ramos 和 C. Tomlin 编，《style id='26'>机器人学习会议，机器学习研究论文集第 155 卷，第 2268–2278 页。PMLR，2020。[10] C. Gao、R. Gehlhar、A. D. Ames、S.-C. Liu 和 T. Delbruck。利用 edgedrnn 加速器控制混合动力学经股假肢的循环神经网络。发表于*2020* 年 *IEEE*机器人与自动化国际会议 *(ICRA)*，第 5460–5466 页。IEEE，2020。[11] A. D. Keles, 和 C. A. Yucesoy。动力型踝关节假肢的神经网络控制算法开发。《生物力学杂志》，113:110087，2020。[12] K. Vonsevych、M. F. Goethel、J. Mrozowski、J. Awrejcewicz 和 M. Bezuglyi。基于人工神经网络模型的手指运动控制系统。《无线电电子与通信系统》，62(1):23–33，2019。[13] R. Ehlers。分段线性前馈神经网络的形式化验证。发表于第*15*届自动化技术验证与分析国际研讨会 *(ATVA)*，第 269–286 页。施普林格，2017。[14] V. Tjeng、K. Y. Xiao 和 R. Tedrake。利用混合整数规划评估神经网络的鲁棒性。发表于第七届学习表征国际会议 *(ICLR)*，2019。[15] S. Dutta、S. Jha、S. Sankaranarayanan 和 A. Tiwari。深度前馈神经网络的输出范围分析。发表于第*10*届 *NASA*形式化方法国际研讨会 *(NFM)*，LNCS 第 10811 卷，第 121–138 页。施普林格，2018。

[16] G. Katz、D. A. Huang、D. Ibeling、K. Julian、C. Lazarus、R. Lim、P. Shah、S. Thakoor、H. Wu、A. Zelji´c 等。用于深度神经网络验证与分析的 Marabou 框架。发表于计算机辅助验证国际会议，第 443–452 页。施普林格，2019。[17] C. Liu、T. Arnon、C. Lazarus、C. Strong、C. Barrett 和 M. J. Kochenderfer。深度神经网络验证算法。《优化基础与趋势》，4(3-4):244–404，2021。ISSN 2167-3888。[18] X. Sun、H. Khedr 和 Y. Shoukry。神经网络控制自主系统的形式化验证。发表于第22届ACM混合系统国际会议：计算与控制，2019。[19] R. Ivanov、T. J. Carpenter、J. Weimer、R. Alur、G. J. Pappas 和 I. Lee。利用神经网络控制器验证自主系统的安全性。《ACM嵌入式计算系统汇刊》(TECS)，20(1):1–26，2020。[20] S. Dutta、S. Jha、S. Sankaranarayanan 和 A. Tiwari。利用前馈神经网络学习和验证反馈控制系统。发表于《混合系统分析与设计》，2018。[21] X. Huang、M. Kwiatkowska、S. Wang 和 M. Wu。深度神经网络的安全性验证。发表于 R. Majumdar 和 V. Kunˇcak 编，第29届计算机辅助验证国际会议 (CAV)，LNCS 第 10426 卷，第 3–29 页。施普林格，2017。[22] Y. Tian、K. Pei、S. Jana 和 B. Ray。DeepTest：对深度神经网络驱动的自动驾驶汽车进行自动化测试。发表于第40届软件工程国际会议 (ICSE)，2018。[23] T. Dreossi、A. Donze 和 S. A. Seshia。利用机器学习组件对信息物理系统进行组合式证伪。《自动推理杂志》，63:1031–1053，2019。[24] C. E. Tuncali、G. Fainekos、D. Prokhorov、H. Ito 和 J. Kapinski。针对具有机器学习组件的自动驾驶车辆进行需求驱动的测试生成。《IEEE智能车辆汇刊》，5:265–280，2020。doi:10.1109/TIV.2019.2955903。[25] T. Dreossi、S. Ghosh、X. Yue、K. Keutzer、A. L. Sangiovanni-Vincentelli 和 S. A. Seshia。反例引导的数据增强。发表于第二十七届人工智能国际联合会议论文集 (IJCAI)，第 2071–2078 页，2018。[26] A. Sinitsin、V. Plokhotnyuk、D. Pyrkin、S. Popov 和 A. Babenko。可编辑神经网络。发表于国际学习表征会议，2019。[27] X. Ren、B. Yu、H. Qi、F. Juefei-Xu、Z. Li、W. Xue、L. Ma 和 J. Zhao。用于 DNN 修复的少样本引导混合。发表于2020 年 IEEE 软件维护与演化国际会议 (ICSME)，第 717–721 页。IEEE，2020。[28] X. Gao、J. Si、Y. Wen、M. Li 和 H. H. Huang。用于机器人下肢假肢的知识引导强化学习控制。发表于2020 年 IEEE机器人与自动化国际会议 (ICRA)，第 754–760 页，2020。doi:10.1109/ICRA40945.2020.9196749。[29] X. Zeng 和 D. Yeung。多层感知机对输入和权重扰动的敏感性分析。《IEEE神经网络汇刊》，12(6):1358–1366，2001。doi:10.1109/ 72.963772。[30] P. Belotti、L. Liberti、A. Lodi、G. Nannicini、A. Tramontani 等。析取不等式：应用与扩展。《威利运筹学与管理科学百科全书》，2:1441–1450，2011。[31] C. Tsay、J. Kronqvist、A. Thebelt 和 R. Misener。用于训练后的 ReLU 神经网络混合整数优化的基于划分的公式化方法。《神经信息处理系统进展》，34:3068–3080，2021。[32] R. E. Moore、R. B. Kearfott 和 M. J. Cloud。《区间分析导论》/ R. E. Moore、R. Baker Kearfott、Michael J. Cloud。费城，2009。

[33] S. Schaal. 模仿学习是否是通往人形机器人的道路？认知科学趋势，3 (6):233–242，1999.[34] R. J. Cortino, E. Bol´ıvar-Nieto, T. K. Best, and R. D. Gregg. 动力膝踝假肢的上楼梯相位变量控制。载于*2022*年国际机器人与自动化会议（*ICRA*），第5673–5678页。IEEE，2022年。[35] V. Taormina, D. Cascio, L. Abbene, and G. Raso. 微调卷积神经网络用于HEp-2图像分类的性能。应用科学，10(19):6940，2020年。[36] D. C. Morgenroth, A. C. Gellhorn, and P. Suri. 残障人群中的骨关节炎：力学视角。物理医学与康复，4(5):S20–S27，2012年。[37] A. Ekizos, A. Santuz, A. Schroll, and A. Arampatzis. 步行和跑步期间的最大李雅普诺夫指数：不同标记集的可靠性评估。生理学前沿，9:1101，2018年。ISSN 1664-042X。

arXiv:2303.04431v1 [cs.RO] 8 Mar 2023

通过神经网络修复实现辅助设备中的安全机器人学习——补充材料

Keyvan Majd[1†]、**Geoffrey Clark**[1], **Tanmay Khandait**[1], **Siyu Zhou**[1],
Sriram Sankaranarayanan[2], **Georgios Fainekos**[3], **Heni Ben Amor**[1]

[1]亚利桑那州立大学， [2]科罗拉多大学博尔德分校， [3] Toyota NA-R&D

[†]majd@asu.edu

一种区间算术方法

为了说明我们如何为每个 ReLU 激活节点生成紧致的有效界，我们使用了区间算术方法 [1, 2]。区间算术广泛应用于验证，可针对给定的有界输入集，为松弛ReLU激活求得上界和下界。我们采用相同的方法，在假设权重相对于原始权重只能在有界的 l_∞ 误差范围内扰动的情况下，为 ReLU 节点求得紧致界。假设我们将修复层 L 的每个输入变量记为 $x^{L-1}(i)$，将连接 $x^{L-1}(i)$与 $x^L(j)$的权重项记为 $\theta_w^L(ij)$，将节点 $x^L(j)$的偏置项记为 $\theta_b^L(j)$。给定变量 $\theta_w^L(ij) \in \left[\theta_w^L(ij)\ \bar{\theta}_w^L(ij)\right]$ 和$\theta_b^L(ij) \in \left[\theta_b^L(ij)\ \bar{\theta}_b^L(ij)\right]$的界，区间算术可为 $x^L(j)$给出有效上下界，表示为 $\bar{}$

$$\bar{x}^L(j) = \sum_i \left(\bar{\theta}_w^L(ij)\max(0, x^{L-1}(i)) + \underline{\theta}_w^L(ij)\min(0, x^{L-1}(i))\right) + \bar{\theta}_b^L(ij), \text{ and}$$

$$\underline{x}^L(j) = \sum_i \left(\underline{\theta}_w^L(ij)\max(0, x^{L-1}(i)) + \bar{\theta}_w^L(ij)\min(0, x^{L-1}(i))\right) + \underline{\theta}_b^L(ij),$$

分别。后续层中 ReLU 节点的界 $l = L{+}1, \ \cdots N$可按如下方式获得：

$$\bar{x}^l(j) = \sum_i \left(\bar{x}^{l-1}\max(0, \theta_w^l(ij)) + \underline{x}^{l-1}\min(0, \theta_w^l(ij))\right) + \theta_b^l(ij),$$

$$\underline{x}^l(j) = \sum_i \left(\underline{x}^{l-1}\max(0, \theta_w^l(ij)) + \bar{x}^{l-1}\min(0, \theta_w^l(ij))\right) + \theta_b^l(ij).$$

第六届机器人学习会议（CoRL 2022），奥克兰，新西兰。

B NNRepLayer 的可靠性

我们的方法只能保证修复样本满足约束。虽然我们通过实验展示了该方法的泛化能力，但无法保证未见对抗样本满足约束。为解决这一问题，我们提出算法 1 ，该算法保证输入 Ψ 满足约束 $x \in \mathcal{X}_r$。在该算法中，NNReplayer 与可靠验证器 [3, 4] 在循环中配合使用：我们的方法首先返回修复后的网络 π_θ^r。随后，验证器对该网络进行评估。如果算法终止，则可保证该网络对于目标输入空间 $\mathcal{X}_r$中所有其他未见样本都是安全的。否则，该网络无法满足安全要求，验证器会提供新发现的对抗样本 $\mathcal{X}_r$ ，对于这些样本，安全保证不成立。接着，NNRepLayer 使用验证器提供的样本来修复网络。当验证器确认约束得到满足时，该循环终止。

算法 1在带有可靠验证器的循环中的 NNRepLayer

输入： π_θ^o, $\mathcal{X}_r$, Ψ

输出： π_θ^r

$\pi_\theta^r \leftarrow \pi_\theta^o$
当 $\mathcal{X}_r \notin \emptyset$ **时**
  $\pi_\theta^r \leftarrow$ NNRepLayer(π_θ^r, $\mathcal{X}_r$, Ψ)
  $\mathcal{X}_r \leftarrow$ V验证器(π_θ^r)
结束循环

定理 1. 假设 V验证器() 是一个可靠验证器 *[3, 4]*。如果算法 *1* 终止，则谓词 Ψ 由修复后的网络满足 π_θ^r。

证明。 给定可靠验证器 VERIFIER()，如果算法终止，则 $\mathcal{X}_r$ 为空，这意味着 VERIFIER() 未找到其他违反 Ψ的样本。因此，谓词 Ψ 保证由 π_θ^r满足。 □

C 将 NNRepLayer 与其他修复技术进行比较

我们在问题中运行 REASSURE [5] 和 PRDNN [6] 提供的修复代码。为确保使用 [5], 中介绍的技术所构建的修复准确无误，我们联系了作者。将我们的控制任务运行于 [6] 提供的代码时会报错。我们发现，错误是由于 [6] 仅适用于具有一维和二维输入的网络（作者在论文 [6]中也提到了这一点）。如图 **??** （论文中的图）所示，REASSURE [5] 能准确满足边界约束。然而，它在修复区域引入了幅值较大的控制量。图 **??** （论文中的图）显示，REASSURE [5] 无法满足输入输出约束，而我们的方法则能满足这些约束。总的来说，这些修复方法 [5, 6] 要么无法应用于我们所用规模的网络，要么表现不佳，无法处理我们所针对的复杂约束。

表 1 详细比较了我们的方法与文献中的其他修复方法。表 1 以及我们的实证评估表明，我们的方法是文献中唯一能够对网络施加复杂硬约束的修复方法。

表1：将我们的方法与文献中的其他方法进行比较

	训练可满足性保证	泛化	副作用	复杂硬约束	隐藏层修复
微调与重新训练 [7, 8, 9]	No	No	Yes	No	Yes
Arch. 扩展 [5, 6]	Yes	No	Yes	No	—（函数空间）
Goldberg 等人 [10]	Yes	No	Yes	No	No
我们的方法	Yes	No	Yes	Yes	Yes

D 测试较大网络的修复

为了展示我们方法的可扩展性，我们在一个每个隐藏层包含 256 个神经元的网络上进行了修复实验。我们使用 1000 个样本进行修复，并使用 2000 个样本进行测试。我们将该问题构建为混合整数二次规划，并在 Gurobi [11] 求解器上运行程序。10 小时后，我们终止了求解器，并报告找到的最佳可行解。图 1 和表2 分别展示了该实验的控制信号和统计结果。如图所示，我们的方法在 10 小时内修复了一个最多包含 256 个节点的网络，修复效果达到 100%。

表2：针对输入输出约束修复，对每个隐藏层包含 256 个节点的网络进行修复的实验结果，最大踝关节角速度为 2 [rad/s]。表中报告了网络规模、样本数量、修复后的输出与原始输出之间的平均绝对误差（MAE）、得到修复的对抗样本百分比（修复效果）以及运行时间。

	网络规模	样本数量	MAE	修复效果 [%]	运行时间 [小时]
输入输出约束	256	1000	0.62	100	10

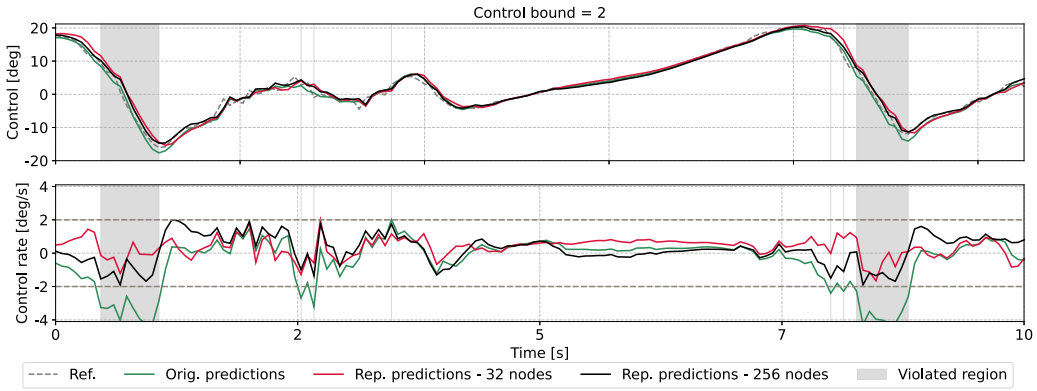


图1：每个隐藏层分别含有32个（红色）和256个（黑色）节点的网络的输入输出约束：踝关节角度和踝关节角速度的边界 $\Delta\alpha_a = 2$。

E 部分修复单层

本节通过示例介绍计算加速和启发式方法，这些方法通过对一个包含 64 个隐藏节点的网络中的单个隐藏层随机选取的节点进行修复 35 次，从而加快神经网络修复。我们让求解器在每次实验中运行 30 分钟（相比之下，完整修复需要 6 小时才能完成求解）。

图 2 展示了平均绝对误差（平均绝对误差）、修复后的权重总数、修复效果以及原始混合整数二次规划代价。为了使修复层的权重改变量具有稀疏性，我们在混合整数二次规划代价函数中加入了 l_1 修复后的权重与其原始值之间的范数有界误差，从而求解原始的完整修复问题。图 2 中的粗条表示对10 个随机选取的稀疏节点进行修复的结果。修复所选稀疏节点后，得到的代价值与原始完整修复问题的代价值非常接近（42.99，而40.29），**耗时仅为 30 分钟，而原方法需要 6 小时**。如图所示，修复某些随机节点也会导致问题不可行（空白条），这表明这些节点无法修复网络。在未来的工作中，我们计划探索神经网络剪枝等技术 [12, 13], ，以便只选择并修复能够满足约束的层和节点，而非修复整个层。本实验结果表明，修复部分节点能够显著缩短我们方法的计算时间。

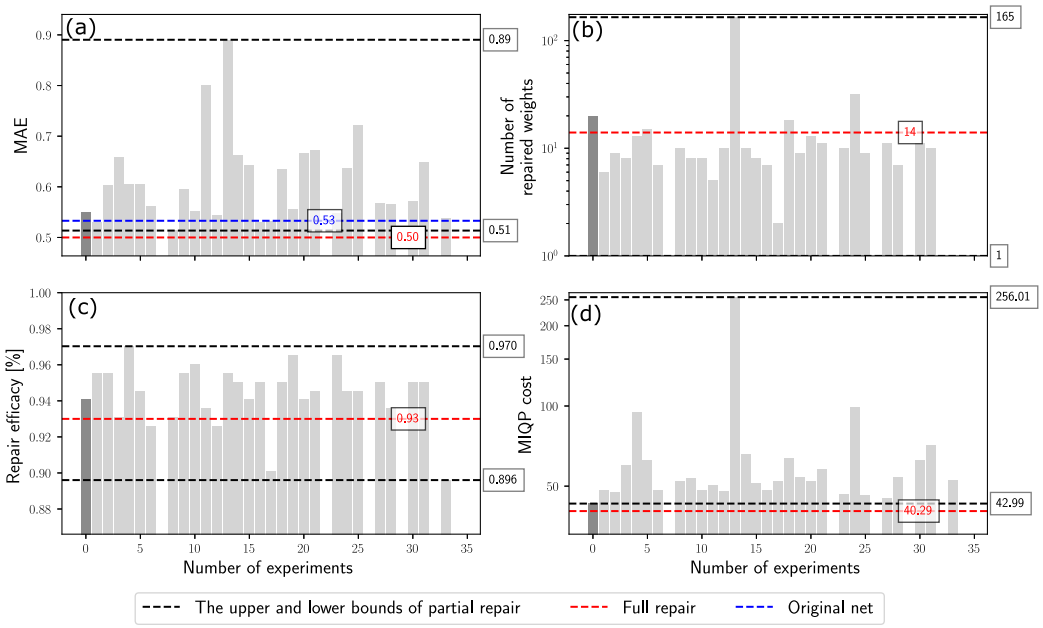


图2：部分修复与完全修复：我们在每个隐藏层有64个节点的网络中，随机选取10个节点进行30分钟的修复。(a) 平均绝对误差（平均绝对误差），(b) 修复后的权重总数，(c) 修复效果，以及 (d) 混合整数二次规划成本。我们进行了35次随机选择。

参考文献

[1] R. E. Moore、R. B. Kearfott 和 M. J. Cloud。《区间分析导论》/Ramon E. Moore、R. Baker Kearfott、Michael J. Cloud。费城，2009 年。[2] V. Tjeng、K. Y. Xiao 和 R. Tedrake。使用混合整数规划评估神经网络的鲁棒性。载于《第七届学习表征国际会议》（ICLR），2019 年。[3] X. Huang、M. Kwiatkowska、S. Wang 和 M. Wu。深度神经网络的安全性验证。载于《计算机辅助验证国际会议》，第 3–29 页。施普林格，2017 年。[4] G. Katz、C. Barrett、D. L. Dill、K. Julian 和 M. J. Kochenderfer。Reluplex：一种用于验证深度神经网络的高效 SMT 求解器。载于《计算机辅助验证国际会议》，第 97–117 页。施普林格，2017 年。[5] F. Fu 和 W. Li。具备最小性和局部性保证的可靠且完备的神经网络修复。arXiv 预印本 arXiv:2110.07682，2021 年。[6] M. Sotoudeh 和 A. V. Thakur。深度神经网络的可证明修复。载于《第 42 届 ACM SIGPLAN 编程语言设计与实现国际会议》，第 588–603 页，2021 年。[7] A. Sinitsin、V. Plokhotnyuk、D. Pyrkin、S. Popov 和 A. Babenko。可编辑神经网络。载于《国际学习表征会议》，2019 年。[8] X. Ren、B. Yu、H. Qi、F. Juefei-Xu、Z. Li、W. Xue、L. Ma 和 J. Zhao。用于 DNN 修复的少样本引导混合。载于《2020 年 IEEE 软件维护与演化国际会议》（ICSME），第 717–721 页。IEEE，2020 年。[9] G. Dong、J. Sun、X. Wang、X. Wang 和 T. Dai。探索正确修复神经网络的方法。载于《IEEE 第 21 届软件质量、可靠性与安全国际会议》（QRS），第 714–725 页，2021 年。[10] B. Goldberger、G. Katz、Y. Adi 和 J. Keshet。利用验证对深度神经网络进行最小修改。载于《第 23 届程序设计、人工智能与推理逻辑国际会议》，第 73 卷，第 260–278 页，2020 年。[11] Gurobi Optimization, LLC。《Gurobi Optimizer 参考手册》，2021 年。网址：https:// www.gurobi.com。[12] B. Hassibi、D. Stork 和 G. Wolff。最优脑外科医生：扩展与性能比较。《神经信息处理系统进展》，第 6 卷，1993 年。[13] Y. LeCun、J. Denker 和 S. Solla。最优脑损伤。《神经信息处理系统进展》，第 2 卷，1989 年。