自 动 化修 复：在安全关键条件下为 AI 赋能的网络物理系统提供自动化修复

Deyun Lyu，宋嘉扬、张振亚、王志杰、张天一、马磊和查建军 o

arXiv:2304.05617v1 [cs.SE] 12 Apr 2023

摘要—**信息物理系统（CPS）已广泛部署于交通运输、电力和能源等安全关键领域。近年来，为了在复杂的工业安全关键环境中实现更智能的控制和决策，人们对在CPS中采用深度神经网络（DNN）的需求日益增长，由此催生了DNN控制器这一类控制器。然而，由于DNN固有的不确定性和不透明性，人们对DNN赋能的CPS安全性的担忧也日益加剧。在本工作中，我们提出了一种名为AUTOREPAIR的自动化框架。该框架给定一项安全要求后，能够识别DNN控制器中的不安全控制行为，并通过基于优化的方法对其进行修复。在获得系统执行的不安全信号后，AUTOREPAIR迭代探索控制决策空间，并为DNN控制器寻找最优修正方案，以满足安全要求。我们在来自不同安全关键领域的6个工业级DNN赋能的CPS实例上，对AUTOREPAIR进行了全面评估。评估结果表明，AUTOREPAIR成功修复了DNN控制器中的关键安全问题，并显著提升了CPS的可靠性。**

索引词—**信息物理系统，神经网络控制器，系统修复。**

一、引言

信息物理系统（信息物理系统）采用计算机技术来监测和控制物理组件，以执行各种现实世界任务。信息物理系统的示例包括自适应巡航控制系统、风力涡轮机系统、智能电网系统等。通常，信息物理系统由物理装置和数字控制器组成——控制器根据装置状态和外部环境作出控制决策，而装置执行控制决策以触发状态转换。

传统控制器，例如基于比例-积分-微分控制（*PID*）[1] 和模型预测控制（*MPC*） [2],的控制器，通常采用人工设计的启发式方法或线性模型来做出控制决策。开发这类控制器成本高昂，需要大量投入和反复迭代，才能确保其具备足够的鲁棒性和可靠性，以应对现实世界中复杂的情况。近期研究 [3]–[5]表明，在鲁棒性和运行时性能方面，DNN控制器可优于传统控制器

吕德云、张振亚和赵建军任职于日本福冈九州大学信息科学与电气工程学院及研究生院（电子邮箱：zhang@ait.kyushu-u.ac.jp）。宋佳阳、王志杰和马磊任职于加拿大埃德蒙顿阿尔伯塔大学电气与计算机工程系。张天一任职于美国西拉法叶普渡大学计算机科学系。

在多个信息物理系统领域，这些研究展示了使用DNN控制器替代传统控制器的潜力。

尽管在CPS中采用DNN控制器的趋势日益增长，但由于DNN固有的不确定性和不透明性，人们也对此产生了越来越多的担忧。由于CPS通常部署于安全关键领域，不当的控制决策可能导致灾难性的系统故障，进而造成难以承受的损失和严重的社会影响。现有研究大多致力于CPS测试（例如，基于优化的证伪 [6]–[13]），旨在检测此类危险情形。然而，为了提升DNN赋能的CPS的可靠性，并降低不当控制决策带来的风险，在发现不安全的系统行为后，有必要对其进行诊断和修复，而这一问题尚未受到足够关注。

在本文中，我们沿着这一方向进行了初步尝试，并提出了一个框架 AUTOREPAIR，用于在安全关键条件下对DNN赋能的CPS进行自动化修复。给定一组安全要求，AUTOREPAIR首先通过检查系统输出的安全满足情况来识别不安全的系统行为。然后，AUTOREPAIR通过基于优化的方法修复控制决策。给定一次不安全的系统执行，AUTOREPAIR迭代探索控制决策空间，并搜索最优的控制修正，使系统能够满足安全要求。最后，AUTOREPAIR将修复后的控制输入和输出纳入训练数据集，并对DNN控制器进行鲁棒再训练。 შედეგად，获得了一个增强的DNN控制器，该控制器能够纠正不安全的系统行为，并提高系统在安全关键条件下的可靠性。

总而言之，我们作出以下贡献：

- 首先，我们提出了一种在安全关键条件下对DNN赋能的CPS进行自动化修复的新技术。我们将该问题构建为一个多目标优化问题，并采用优化求解器搜索控制信号的最优修正，以确保修复后的系统执行满足安全要求；
- 随后，我们将所提出的技术实现为一个通用框架AUTOREPAIR，该框架能够处理*Simulink*中的工业网络物理系统；Simulink是业界用于信息物理系统建模环境的事实上的标准；
- 我们对6个行业级DNN赋能的CPS实例进行了全面的实验评估。评估结果证明了我们方法的有效性。据我们所知，这是首个旨在

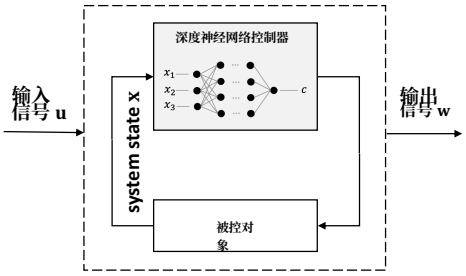


图1. 典型DNN赋能的CPS架构

针对系统级安全要求修复深度神经网络控制器。随着工业网络物理系统中采用 DNN 的需求日益增长，这项工作为进一步研究以及 DNN 在复杂工业网络物理系统中的安全实际应用铺平了道路。

II. 预备知识

本节将概述DNN赋能的CPS（第II-A节）、DNN控制器（第II-B节），以及以信号时序逻辑（*STL*）（第II-C节）表达的系统需求，作为本文所提技术的预备知识。

*A. DNN*赋能的*CPS*

图1展示了DNN赋能的CPS典型的整体系统架构。具体而言，该系统由一个物理对象和一个DNN控制器组成。物理对象是具有连续状态转移的物理组件，其行为高度非线性且复杂。DNN控制器以系统状态$\mathbf{x}$和外部输入信号$\mathbf{u}$作为输入。随后，它在每个时间戳发出控制信号$\mathbf{c}$，以控制物理对象的运行。该过程反过来会产生一个可从外部观测到的输出信号$\mathbf{w}$。为了对信号和系统状态（即$\mathbf{u}$、$\mathbf{w}$、$\mathbf{x}$、$\mathbf{c}$）的动态进行建模，我们将其定义为时间序列，将时间瞬间 $t \in [0,T]$ $(T \in \mathbb{R}_{\geq 0})$映射到向量$\mathbf{s}(t) \in \mathbb{R}^d$；其中，$d$是信号或状态的维度，$[0,T]$是信号或状态的时间范围。

具体来说，$\mathbf{c}$ 是一维的，其取值范围由一个实数闭区间 $\mathcal{B}_{\mathbf{c}}$ 限定，即 $\forall t \in [0,T]\,.\,\mathbf{c}(t) \in \mathcal{B}_{\mathbf{c}}$。

*B. DNN*控制器及其训练集

传统信息物理系统控制器，例如基于 比例-积分-微分控制 [1]和模型预测控制 [2]，的控制器，采用人工设计的启发式方法或线性模型来做出控制决策。近年来，DNN作为传统控制算法的替代方案，在工业领域中受到越来越多的研究，用于处理现实世界中的复杂条件。目前，通常采用两种方法来训练集DNN控制器，即 深度强化学习（*DRL*）和 监督学习：

- 深度强化学习允许训练者在动作空间中进行探索；如果训练者选择了“好”的动作，就会获得奖励；如果选择了“坏”的动作，就会受到惩罚。最终，训练者通过反复“试错”，以深度神经网络的形式获得最优策略。
- 在监督学习中，学习器从收集于信息物理系统中的历史或经验数据中学习，并将其作为训练数据集 $\{\langle \mathbf{x},\ \mathbf{c}\rangle\}$，其中每个元素都是一个系统状态$\mathbf{x}$ 及其对应控制决策$\mathbf{c}$组成的二元组。训练阶段旨在优化 DNN 参数，从而在给定$\mathbf{x}$时，尽可能准确地预测$\mathbf{c}$ 。

在本文中，我们考虑白盒设置下CPS 开发和维护环境中的深度神经网络控制器修复，以帮助工程师在早期阶段识别并修复潜在的安全问题。换言之，工程师可以访问深度神经网络控制器的内部结构，因此，借助通过我们的方法或从实际 CPS 物理环境中收集的更多修复后的信号数据，可以修复并增强 CPS 的安全行为，例如通过重新训练或微调深度神经网络控制器，以实现持续交付。此外，我们假设 $\mathbf{u}$,$\mathbf{w}$,$\mathbf{x}$,$\mathbf{c}$ （即系统组件之间以及系统与外部环境之间的信号）都是可观测的。此外，我们假设深度神经网络控制器所作出的控制决策会导致 CPS 的不安全行为。

C. 信息物理系统的安全需求

在这项工作中，我们采用信号时序逻辑（*STL*） [14]来规定 DNN赋能的CPS 的安全要求。STL 是一种广受认可的形式化语言 [15], [16] ，用于表达用户期望的性质。它将经典的线性时序逻辑（*LTL*） [17]从离散时间设置扩展到连续时间和行驶距离域，并配备了一种称为鲁棒性的定量语义，它不仅能指出某项性质是否得到满足，还能衡量其满足程度。STL 的定量语义已成为许多其他安全保障技术的技术基础，例如运行时监控 [18]–[20]和证伪 [6]–[11]。下面我们介绍 STL 的语法和鲁棒性。

STL 语法 在 STL 中，原子命题 α 定义为 $\alpha \equiv (f(\mathbf{w}) > 0)$，其中 f 是一个将输出信号映射为实数的函数。STL 公式 φ 的定义递归如下：

$$\varphi :\equiv \alpha \mid \bot \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \Box_I \varphi \mid \Diamond_I \varphi \mid \varphi_1 \,\mathcal{U}_I\, \varphi_2$$

其中，I 表示一个时间区间。这里，$\Box_I\varphi$、$\Diamond_I\varphi$ 和 $\varphi_1 \,\mathcal{U}_I\, \varphi_2$ 是时间命题，其中 $\Box_I\varphi$ 要求某事始终发生，$\Diamond_I\varphi$ 要求某事最终发生，而 $\varphi_1 \,\mathcal{U}_I\, \varphi_2$ 要求 φ_1 持续发生，直到 φ_2。其他常见运算符（如 $\vee$）可以由现有运算符导出：$\varphi_1 \vee \varphi_2 \equiv \neg(\neg\varphi_1 \wedge \neg\varphi_2)$。**STL鲁棒性**给定系统输出$\mathbf{w}$ 和一个 STL 公式 φ，鲁棒性 $\mathtt{rob}(\mathbf{w}\ \varphi)$返回一个实数，表示$\boldsymbol{w}$ 满足 φ的**鲁棒程度**。$\mathtt{rob}(\mathbf{w}\ \varphi)$的定义如下，按 STL 公式的结构进行归纳：

$$\mathtt{rob}(\mathbf{w},\alpha) := f(\mathbf{w}) \qquad \mathtt{rob}(\mathbf{w},\bot) := -\infty$$
$$\mathtt{rob}(\mathbf{w},\neg\varphi) := -\mathtt{rob}(\mathbf{w},\varphi)$$
$$\mathtt{rob}(\mathbf{w},\varphi_1 \wedge \varphi_2) := \min\left(\mathtt{rob}(\mathbf{w},\varphi_1),\mathtt{rob}(\mathbf{w},\varphi_2)\right)$$
$$\mathtt{rob}(\mathbf{w},\Box_I\varphi) := \inf_{t\in I}\left(\mathtt{rob}\left(\mathbf{w}^t,\varphi\right)\right)$$
$$\mathtt{rob}(\mathbf{w},\varphi_1 \,\mathcal{U}_I\, \varphi_2) := \sup_{t\in I} \min\begin{pmatrix} \mathtt{rob}(\mathbf{w},\varphi_2), \\ \inf_{t'\in[0,t)} \mathtt{rob}\left(\mathbf{w}^{t'},\varphi_1\right) \end{pmatrix}$$

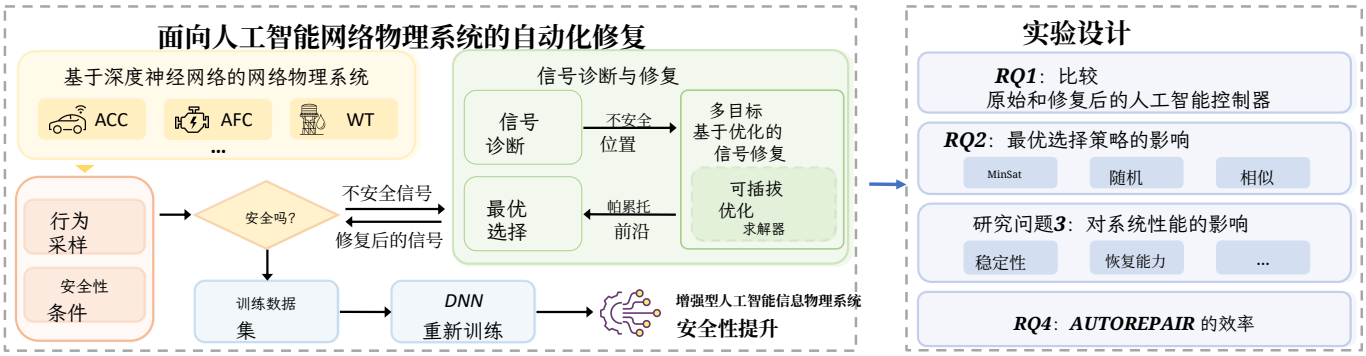


图 2. 我们提出的用于DNN赋能的CPS自动化修复的框架 AUTOREPAIR 的工作流程和研究设计。

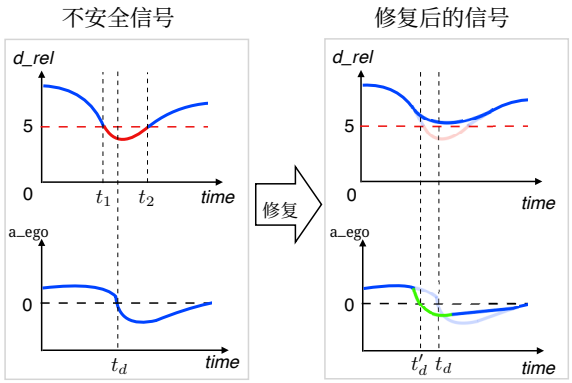


图3. 自适应巡航控制（ACC）系统信号修复的说明性示例。

这里，$\mathbf{w}^t$ 表示将$\mathbf{w}$ 的起始点平移 t 个时间单位后得到的信号。直观而言，rob ($\mathbf{w}$ φ)越大，$\mathbf{w}$ 满足 φ的鲁棒性就越强；一旦 rob ($\mathbf{w}$ φ)为负，则表示$\mathbf{w}$ 违反了 φ。

III. AUTOREPAIR 框架

本文提出了一种名为 AUTORE-PAIR的框架，以提升 DNN 控制器的系统安全性。具体而言，AUTOREPAIR (i) 识别系统执行中的不安全输出信号；(ii) 通过搜索控制信号补丁进行修复，确保系统在给定执行期间满足安全要求；(iii) 利用安全执行轨迹数据修复 DNN 控制器。我们首先在示例III.1中说明如何修复不安全的系统执行，然后介绍 AUTOREPAIR 的工作流程。

示例 III.1. 考虑自适应巡航控制（*ACC*）场景：本车试图与前车保持安全距离。我们将本车视为所研究的 DNN 赋能系统，将前车视为外部环境。本车的控制器根据传感器数据（包括本车和前车的状态）决定其加速度或减速度a_{ego}。在此场景中，系统要求 φ可表示如下：

$$\varphi \equiv \square_{[0,T]}(d_{rel} > 5)$$

其中要求两辆汽车之间的相对距离d_{rel} 始终大于 5 的安全距离。

图3 左侧展示了违反安全要求的系统执行示例。在 $[t_1, t_2]$，d_{rel} 小于 5。这是因为，t_d （即本车开始减速的时刻）太晚，无法避免相对距离过近。图 3 右侧展示了信号修复示例，其中一个控制信号补丁（绿色线段）替换了原始控制信号。该信号补丁将减速时刻 t_d 提前至 t'_d。因此，本车及时减速，避免违反安全要求。现在，借助控制信号补丁，我们可以重新运行系统，获得一条避免不安全行为的新执行轨迹。我们将使用这条新的执行轨迹修复深度神经网络控制器，从而提高系统可靠性。

本示例阐释了 AUTOREPAIR 的主要任务：对于不安全的系统执行，需要合成控制信号补丁，使系统执行满足安全要求。在介绍技术细节之前，我们先概述 AUTOREPAIR 的高层工作流程。

工作流程。给定一个配备深度神经网络控制器的信息物理系统，AUTOREPAIR首先运行系统并执行行为采样，以获取执行轨迹。每条执行轨迹都包含系统输入信号$\mathbf{u}$、系统输出信号$\mathbf{w}$、控制决策信号$\mathbf{c}$以及系统状态信号$\mathbf{x}$。随后，给定安全要求 φ，AUTOREPAIR 会检查执行轨迹中的每个系统输出信号$\mathbf{w}$ 是否满足安全要求。如果发现不安全信号，就会触发信号诊断与修复流程，以修复不安全信号并生成新的安全保障执行轨迹。该流程将在第IV节详细介绍。

有了安全保障执行轨迹，我们就可以修复原先发出不当控制信号并导致系统行为不安全的深度神经网络控制器。为了增强通用性，作为深度神经网络控制器修复的早期步骤，AUTOREPAIR 中的深度神经网络修复模块被设计为可扩展的，以支持多种修复方法。基本模式是利用新的安全保障执行轨迹重新训练原始深度神经网络控制器，以提升其性能，同时保持其余训练配置不变。尽管这种模式成本最低，但由于模型架构可能并非最优，有时会产生鲁棒性较差的 DNN控制器。除基本模式外，其他选项还包括神经架构搜索（*NAS*） [21], ，它可以搜索不同的 DNN 配置——

算法 1 基于优化的信号修复

要求： 具有深度神经网络控制器 $\mathcal{C}$的原始系统、安全要求 φ，以及包含输入信号 $\mathbf{u}$、输出信号 $\mathbf{w}$ 的执行轨迹，控制信号 $\mathbf{c}$ 和状态信号$\mathbf{x}$

1: **过程** DIAGANDREPAIR($\mathbf{u}, \mathbf{w}, \mathbf{c}, \mathbf{x}$)
2: **如果** $\mathtt{rob}(\mathbf{w}\ \varphi) < 0$ **则** ▷$\mathbf{w}$ 不安全
3: $\begin{bmatrix} \tau_{\mathrm{S}} \leftarrow$违规事件开始时 $\\ \tau_{\mathrm{E}} \leftarrow$违规事件结束时 $\end{bmatrix}$ ▷ 诊断
4: $\mathbf{c}' \leftarrow$ 参数 $\begin{bmatrix} \max\limits_{\mathbf{c}'} & \mathtt{rob}\left(\mathbf{w}^*_{[0,\tau_{\mathrm{E}}]},\ \varphi\right) \\ \min\limits_{\mathbf{c}'} & \mathrm{dist}\left(\mathbf{c}', \mathbf{c}^{[t^{\mathrm{L}}_{\mathbf{c}'},\ t^{\mathrm{U}}_{\mathbf{c}'}]}\right) \\ \text{s.t.} & t^{\mathrm{L}}_{\mathbf{c}'} \in [0, \tau_{\mathrm{S}}] \\ & t^{\mathrm{U}}_{\mathbf{c}'} \in [t^{\mathrm{L}}_{\mathbf{c}'}, \tau_{\mathrm{E}}] \\ & \forall t \in \left[t^{\mathrm{L}}_{\mathbf{c}'}, t^{\mathrm{U}}_{\mathbf{c}'}\right] .\mathbf{c}'(t) \in \mathcal{B}_{\mathbf{c}} \end{bmatrix}$ ▷ repair
5: $\mathbf{w},\mathbf{c},\mathbf{x} \leftarrow \mathrm{SYSEXE}(\mathbf{u},\mathbf{c}^*)$ 与 $\mathbf{c}^* = \langle \mathcal{C},\mathbf{c}' \rangle$ ▷系统重新运行
6: 诊断并修复$(\mathbf{u}, \mathbf{w}, \mathbf{c}, \mathbf{x})$ ▷递归调用
7: 将修复后的 $\langle \mathbf{x}, \mathbf{c} \rangle$ 添加到再训练数据集

并利用新的执行轨迹找出最佳模型。在本研究中，我们通过实验评估了重新训练所得修复后深度神经网络控制器的性能，并将对其他选项的研究留作未来工作。

由于信号诊断与修复是 AUTOREPAIR 框架的主要技术贡献，我们将在下一节详细阐述。

IV. 信号诊断与修复

本节将详细介绍 AUTOREPAIR 的关键组成部分，即信号诊断与修复，其目标是在系统执行过程中修复导致不安全系统输出的控制信号。

A. 算法概述

算法 1描述了信号诊断与修复算法。它以一个包含深度神经网络控制器 $\mathcal{C}$ 的信息物理系统、一个 STL 安全要求 φ，以及一条包含输入信号 u、输出信号 w、控制信号c和状态信号 x的执行轨迹作为输入。需要注意的是，所有这些信号都被定义为时间序列，以在时间范围 $[0, T]$ 内对其动态进行建模，具体定义见第II-A节。该过程首先检查输出信号w 是否满足安全要求 φ （第 2 行）。如果w是安全的，则无需对其进行修复。否则，算法首先定位不安全输出信号的起始和结束时刻（第3 行）。然后，它搜索一个最优控制信号补丁 $\mathbf{c}'$，用于修正系统输出信号，使其满足安全要求（第 4行）。形式上，$\mathbf{c}'$ 是在时间区间 $[t^{\mathrm{L}}_{\mathbf{c}'}, t^{\mathrm{U}}_{\mathbf{c}'}]$内定义的部分信号，其中，$t^{\mathrm{L}}_{\mathbf{c}'}$ 是控制必须开始进行修正以避免不安全系统行为的下限时刻，而 $t^{\mathrm{U}}_{\mathbf{c}'}$ 是控制修正可以结束的上限时刻。关于确定 $t^{\mathrm{L}}_{\mathbf{c}'}$ 和 $t^{\mathrm{U}}_{\mathbf{c}'}$ 的细节将在第 IV-B 节中给出。

使用控制信号补丁 $\mathbf{c}'$，我们在原控制器 $\mathcal{C}$ 和补丁$\mathbf{c}'$ 之间交替控制，以重新执行系统，具体如下：

- 在 $[0, t^{\mathrm{L}}{}_{\mathbf{c}'}]$ 中，控制由原始控制器提供；
- 在 $[t^{\mathrm{L}}{}_{\mathbf{c}'},\ t^{\mathrm{U}}{}_{\mathbf{c}'}]$ 中，控制切换到 $\mathbf{c}'$；
- 在 $[t^{\mathrm{U}}{}_{\mathbf{c}'},\ T]$，控制会切换回原始控制器，直到系统执行结束。

使用此控制开关，我们可以通过重新执行系统（第 $\mathbf{w}$、$\mathbf{c}$ 和 $\mathbf{x}$ 行）获得新的 5。此过程会使用新的 $\mathbf{w}$、$\mathbf{c}$ 和 $\mathbf{x}$ 递归调用，以诊断和修复新的不安全行为，直到整个系统执行都是安全的。最后，将此次新的系统执行返回到 AUTOREPAIR 框架，并作为输入馈送至深度神经网络修复模块，以增强深度神经网络控制器（第 7行）。

B. 信号诊断

一种简单的选择时间范围 $[t^{\mathrm{L}}_{\mathbf{c}'}, t^{\mathrm{U}}_{\mathbf{c}'}]$ 的方法是取整个时间范围 $[0, T]$，并在其中搜索用于修复不安全系统执行的控制信号补丁 $\mathbf{c}'$ 。然而，在这种情况下，由于搜索空间过大，搜索成本会过高。本节将利用 STL 信号诊断 [22], [23] 的最新进展来缩小搜索空间。

信号诊断是一种能够定位信号中违规事件的技术。直观而言，违规事件是信号中的一段，可将其视为导致整个信号违反给定安全属性的原因。一般来说，信号诊断通过递归地收集安全属性的原子命题的评估导致给定 STL 公式违规的时间点来实现，具体依据不同 STL 运算符的语义。我们通过示例IV.1说明这一过程。更多详情请参阅 [22]。

示例 IV.1。在示例III.1中，$[t_1, t_2]$ 中的片段可视为违规原因，因为d_{rel} 在此片段中始终小于 5，这违反了原子命题$d_{rel} > 5$ ，其属于 φ。因此，该原子命题的违规导致 φ 被违反，故这段信号会被诊断为一个违规事件。

在第3 行的算法1中，τ_{S} 设为违规事件开始的时间戳，即系统输出开始违反安全要求的时间戳；而 τ_{E} 设为违规事件结束的时间戳，即系统恢复到安全状态的时间戳。使用信号诊断推导控制器修复的时间约束，其正确性基于以下事实：不当控制决策导致的不安全状态只会沿时间域向后传播，因此，只需考虑输出信号安全违规事件之前或与之重合的时间区间进行修复。相关说明请参见引理IV.2 。

引理 IV.2. 设$\mathbf{c}'$ 为定义在区间 $[t^{\mathrm{L}}_{\mathbf{c}'}, t^{\mathrm{U}}_{\mathbf{c}'}]$内的控制信号补丁，$\tau_{\mathrm{S}}$ 和 τ_{E} 分别为违规事件的起点和终点。那么，$\mathbf{c}'$ 的下时间界 $t^{\mathrm{L}}_{\mathbf{c}'}$ 不应晚于 τ_{S}；$\mathbf{c}'$ 的上时间界 $t^{\mathrm{U}}_{\mathbf{c}'}$ 不应晚于 τ_{E}。

引理 IV.2 的证明基于 STL 信号诊断的定义，以及安全性违规仅会在时间范围内向后传播这一事实。

如果$t^{\mathrm{L}}_{\mathbf{c}'} > \tau_{\mathrm{S}}$，则 $[\tau_{\mathrm{S}}, t^{\mathrm{L}}_{\mathbf{c}'}]$ 中将存在一段不安全部分，该部分未被控制信号补丁 $\mathbf{c}'$修复。因此，$\mathbf{c}'$ 的时间下界 $t^{\mathrm{L}}_{\mathbf{c}'}$ 不应晚于 τ_{S}。

相似地，我们可以推导出，上限时间 $t^{\mathrm{U}}_{\mathbf{c}'}$ 的$\mathbf{c}'$不应晚于 τ_{E}。□

示例 IV.3. 在示例 III.1中， τ_{S} 和 τ_{E} 分别计算为 $\tau_{\mathrm{S}} = t_1$ 和 $\tau_{\mathrm{E}} = t_2$。因此，当我们搜索控制信号补丁$\mathbf{c}'$时，其下界时间$t^{\mathrm{L}}_{\mathbf{c}'}$ 应不晚于 t_1，其上界时间 $t^{\mathrm{U}}_{\mathbf{c}'}$ 应不晚于 t_2。

直观地说，这是正确的，因为如果 $t^{\mathrm{L}}_{\mathbf{c}'}$ 晚于 t_1，那么违反安全性的区间 $[\tau_{\mathrm{S}}, t^{\mathrm{L}}_{\mathbf{c}'}]$ 无法通过 $\mathbf{c}'$ 修复；如果$t^{\mathrm{U}}_{\mathbf{c}'}$ 晚于 τ_{E}，那么在 $[\tau_{\mathrm{E}}, t^{\mathrm{U}}_{\mathbf{c}'}]$ 中不存在安全性违规。

C. 基于多目标优化的修复

在给定时间约束 $[t^{\mathrm{L}}_{\mathbf{c}'}, t^{\mathrm{U}}_{\mathbf{c}'}]$的情况下，如何构建控制信号补丁 $\mathbf{c}'$ 仍然是一个问题。$\mathbf{c}'$ 所需具备的属性涉及以下两个方面：

- 首先，将原始控制信号替换为$\mathbf{c}'$，即可解决 $[\tau_{\mathrm{S}}, \tau_{\mathrm{E}}]$ 中的安全性违规；
- 其次，原始控制信号$\mathbf{c}$ 与控制信号补丁$\mathbf{c}'$ 之间的变化应尽可能小，以尽量减少安全性以外其他方面的性能损失，同时避免产生严重偏离常态的控制信号，从而损害训练数据的质量。

本文使用多目标优化来搜索此类控制信号补丁$\mathbf{c}'$。一般而言，多目标优化包含多个目标函数；这些函数可能彼此冲突，即优化一个函数可能会妨碍另一个函数。因此，多目标优化的结果是一组最优解，称为帕累托前沿，而非单一最优解。直观而言，帕累托前沿包含所有彼此不可比较的最优解——通常不存在一个解在所有目标函数上都优于另一个解。

我们将搜索问题表述为$\mathbf{c}'$ 多目标优化，如算法4第1行：

- 第一个目标函数是最大化修复后的输出信号$\mathbf{w}^*_{[0,\tau_{\mathrm{E}}]}$ 在 $[0, \tau_{\mathrm{E}}]$ 中对安全要求的满足度。最初，$\mathbf{w}^*_{[0,\tau_{\mathrm{E}}]} = \mathbf{w}_{[0,\tau_{\mathrm{E}}]}$，因此 $\mathrm{rob}\left(\mathbf{w}^*_{[0,\tau_{\mathrm{E}}]}, \varphi\right)$ 为负。该目标函数的目标是找到一个$\mathbf{c}'$，使得修复信号 $\mathbf{w}^*_{[0,\tau_{\mathrm{E}}]}$ 的满足度 $\mathrm{rob}\left(\mathbf{w}^*_{[0,\tau_{\mathrm{E}}]}, \varphi\right)$ 为正，从而 $\mathbf{w}^*_{[0,\tau_{\mathrm{E}}]}$ 满足安全要求 φ。
- 第二个目标函数是根据特定距离度量，最小化$\mathbf{c}'$ 与$[t^{\mathrm{L}}_{\mathbf{c}'}, t^{\mathrm{U}}_{\mathbf{c}'}]$中原始控制信号之间的变化。
- 约束包括第IV-B节中的时间约束以及第II-A节中的行驶距离约束。

处理 $\mathbf{c}'$ 的无限搜索空间时，由于其在时间和行驶距离上的连续性，会产生一个问题。遵循信息物理系统社区 [15], [16], [24],的惯例，我们采用 参数化表示 来表示 $\mathbf{c}'$，因此连续信号 $\mathbf{c}'$ 可以通过有限数量的变量来确定。具体而言，我们选择 分段线性 信号，即给定超参数 k后，$\mathbf{c}'$ 在每个区间 $\left[t^{\mathrm{L}}_{\mathbf{c}'} + {}^{i}_{k}(t^{\mathrm{U}}_{\mathbf{c}'} - t^{\mathrm{L}}_{\mathbf{c}'}), t^{\mathrm{L}}_{\mathbf{c}'} + {}^{(i+1)}_{k}(t^{\mathrm{U}}_{\mathbf{c}'} - t^{\mathrm{L}}_{\mathbf{c}'})\right)$ $(i \in \{1, \ldots, k-1\})$内是线性的。在这种情况下，$\mathbf{c}'$由 $t^{\mathrm{L}}_{\mathbf{c}'}, t^{\mathrm{U}}_{\mathbf{c}'}$确定，而 k 变量则刻画了 $\mathbf{c}'$ 的形状。

需要高效的优化求解器来求解第4行中表述的优化问题。在我们的场景中，对目标函数$\mathtt{rob}\left(\mathbf{w}^*_{[0,\tau_{\mathrm{E}}]}, \varphi\right)$ 的评估需要执行信息物理系统，而其动态特性通常难以建模。为此，我们选择*NSGA-II* [25],这一基于遗传算法（GA）的多目标优化算法作为求解器，因为 NSGA-II 无需了解目标函数的内部动态特性。与其他遗传算法一样，它通过反复执行变异和交叉来计算帕累托前沿。

优化求解器返回的帕累托前沿包含一组最优解；有了这样的帕累托前沿后，我们还需要进一步执行最优选择，以选出用于信号修复的解。在自动修复中，我们将此选择过程设为可配置，并在此提供以下三种可能的策略。在第VI节的 RQ2 中，我们通过实验比较了它们的性能。

- MinSat 选择使安全要求的满足度最低的最优解，以兼顾另一个目标函数；
- 随机指一种朴素的选择策略：只要满足安全要求，就从帕累托前沿中随机选择一个解；¨
- 相似会选择相对于原始控制决策变化最小的解决方案，前提是满足安全要求。

修复触发错误的信号后，下一步是利用此类信号数据进行深度神经网络控制器修复，如III节所述。一般而言，AUTOREPAIR可以轻松集成到启用DNN的信息物理系统的开发与维护中，用于DNN修复和增强，从而提高其安全性。

V. 实验设计与设置

本节介绍用于评估 AutoRepair的实验设计与设置，以及其有效性和性能。所有源代码和基准测试均已公开提供：https://github.com/lyudeyun/AutoRepair。

A. 研究问题

我们设计实验旨在探究以下研究问题：

- **RQ1：AUTOREPAIR 能否有效提升 DNN赋能的CPS 的安全性?**

在本研究问题中，我们旨在评估由AUTOREPAIR 修复的、具有深度神经网络控制器的目标信息物理系统是否确实能够提高系统的安全性。为回答这一问题，我们首先通过对这些系统的执行轨迹进行采样来创建再训练数据集，并应用AUTOREPAIR，以获得配备经过再训练的 DNN控制器的修复系统。为进行测试，我们还为每个系统创建了一个测试集。随后，我们将来自再训练数据集和测试集的输入信号输入原始系统和修复系统，从而执行这些系统。我们比较这些系统的安全率，该指标被定义为再训练数据集和测试集中安全执行的数量与所有执行数量之比。

表 I 用于评估 AUTOREPAIR 的代表性信息物理系统。**组件数** 列报告系统中的组件块数量，以衡量模型的复杂度。

主题 信息物理系统	描述	输入信号	控制信号	块数
自适应巡航控制	与前车保持安全距离	前车加速度	本车加速度	297
抽象燃油控制	保持参考空燃比	踏板角度与发动机转速	燃料指令	281
水箱	将水位保持在参考值。	参考水位	水流量速率	919

- **RQ2：最优选择策略如何影响 AUTOREPAIR 的有效性?**

  自动修复中的一个关键阶段是第 IV节中介绍的信号诊断与修复过程。给定一条不安全的执行轨迹，我们搜索能够确保系统安全性的最优控制信号补丁。然而，由于优化求解器给出的输出是一个帕累托前沿，因此我们需要选择一个特定的最优解，并将其应用为控制信号补丁以修复系统。在第 IV-C节中，我们介绍了三种最优解选择策略，即`MinSat`、`Random`、`Similar`。在该研究问题中，我们比较不同最优解选择策略下系统的安全率，以研究这些策略的影响。

- **研究问题3：AUTOREPAIR 会影响DNN赋能的CPS的安全性以外的其他指标的性能吗？虽然**AUTOREPAIR旨在提升系统安全性，但人们担心它可能会削弱系统在其他方面的表现，例如系统执行效率。在本研究问题中，我们考察 AUTOREPAIR对 DNN赋能的CPS 各项系统性能指标的影响。具体而言，我们将采用默认最优选择策略MinSat的 AUTOREPAIR 修复后的系统与原始系统在这些指标上的表现进行比较。请注意，这些系统性能指标因系统而异，因为不同系统在开发过程中受到不同要求的约束。第 V-B节详细介绍了这些指标。

- **RQ4：AUTOREPAIR 的效率如何?**
  除了 AUTOREPAIR 的有效性之外，其效率也是实际应用中的另一个重要因素。AUTOREPAIR 中最耗时的部分是针对每次不安全系统执行进行信号修复的过程。由于该过程依赖多目标优化算法 NSGA-II，因此耗时取决于 NSGA-II 的配置。此外，不同系统的执行时间不同，这也会影响应用 AUTOREPAIR 的耗时。在本研究问题中，我们考察 AUTOREPAIR 的这一方面，并报告 AUTOREPAIR 修复 DNN赋能的CPS不安全行为的效率。

B. 基准测试

我们从三个工业领域中选取了DNN赋能的CPS作为评估对象系统，分别是自适应巡航控制 （ACC） [26], 抽象燃油控制（AFC） [27], 和水箱 （WT） [28]。这些系统涵盖多个安全关键型工业领域，例如自动驾驶和动力系统控制。表I 概述了这些系统，并简要介绍了其功能。所有这些系统均基于Simulink构建，Simulink是业界事实上的CPS建模标准。本节其余部分将详细介绍每个对象CPS，包括其功能、安全要求及其他性能指标。

自适应巡航控制（ACC）。经过数十年的发展，ACC 作为一项重要的驾驶辅助功能，已在汽车领域得到普及。本研究中提及的 ACC 系统源自 MathWorks [26],，旨在通过调节本车的加速度，使本车与前车保持安全距离。当相对距离大于安全距离时，本车速度则保持为驾驶员设定的速度。为此，整个系统以**前车的加速度**作为输入，并输出前车和本车的速度及行驶距离。ACC 的安全要求表述如下：在仿真过程中，两辆汽车之间的相对距离 d_{rel} 始终应大于安全距离，该距离定义为一个常数d^{safe} 与本车的制动距离之和。同时，本车的速度 v^{ego} 应低于 v_{set}。此处，我们分别将d^{safe} 和 v_{set} 设为 10 和 30。

$$S_{\text{ACC}} \equiv \square_{[0,50]}(d_{rel} \geq d_{safe} \wedge v_{ego} \leq v_{set})$$

ACC 的性能指标描述如下。

- 行驶距离是指本车行驶的纵向距离。行驶距离 越大，驾驶效率越高；
- 速度表示本车处于安全状态行驶时，实际速度与预期速度之间的平均偏差。速度越小，巡航效率越高；
- 稳定性 旨在衡量系统稳定性，其体现为系统满足安全要求的时间段占总时间区间 $[0, 50]$ 的比例；
- 恢复能力 旨在评估系统处于非稳态时，控制器是否能迅速使其恢复到稳态；
- 舒适性 表示本车的舒适程度，通过测量系统执行期间加速度的最大变化率来衡量。

抽象燃油控制 (AFC)。AFC是丰田 [27]发布的一种复杂空燃比控制系统。整个系统从外部环境中获取两个输入信号，即，

踏板角度和发动机转速，并输出 $\mu = \frac{|AF - AF_{ref}|}{AF_{ref}}$，即空燃比 AF 相对于参考值$AF_{ref}$的偏差。通过改变系统输入，燃料控制器应调整进入气缸的进气量，以维持最佳空燃比。该系统的目标是控制偏差 μ，使其在时间区间 $[0, 30]$ 内处于预定义阈值 $\mu_{set} = 0.15$ 之内。

$$S_{\text{AFC}} \equiv \square_{[0,30]} \left(\frac{|AF - AF_{ref}|}{AF_{ref}} \leq \mu_{set} \right)$$

AFC 的性能指标描述如下。

- 误差表示实际偏差率的绝对值 μ。较小的误差表示更好的性能，理想情况下，误差应为 0；
- 稳定性的特征是在总时间区间 $[0, 30]$ 内，系统满足安全性要求的时间段所占的比例。

水箱（WT）。作为控制水流入和流出的容器，水箱已应用于化学工业等领域。该系统由 MathWorks 收集 [28]。水可以流入或流出水箱，直到水箱中的水位达到参考值。经过一段时间后，水位应与参考水位相同。该系统将参考水位h_{ref} 作为输入信号，并实时输出实际水位 h_{out}。绝对偏差 $error$ 介于 h_{ref} 和 h_{out}之间，且应始终小于设定值 $误差_{set}$ 在时间区间 $[4, 5]$, $[9, 10]$ 和 $[14, 15]$内均应满足。这里 $误差_{set}$为 0.86。

$$S_{\text{WT}} \equiv \square_{\text{I}}(|error| \leq error_{set})$$
$$\text{where I} = [4, 5] \cup [9, 10] \cup [14, 15]$$

WT的性能指标表述如下，

- 误差表示参考值与实际值之间的绝对误差值。对于 WT，这由h_{ref} 与h_{out}之间的绝对误差值表示。误差越小，表示控制精度越高；
- 稳定性以系统满足安全性要求的时间段占时间区间 $[4, 5]$, $[9, 10]$ 和 $[14, 15]$的比例来表征。

神经网络控制器。为展示 AutoRepair的有效性，对于每个目标信息物理系统，我们使用不同结构训练了 2 个神经网络控制器，因此共有 6 个实例作为评估系统。表 II详细介绍了部署在这些系统上的原始控制器，包括其类型、结构和训练算法。以 ACC#2 为例，由前馈神经网络 *(FFNN)* 构建的原始深度神经网络控制器包含 3 个隐藏层，每个隐藏层有 30 个神经元。该控制器使用 MATLAB 中深度学习工具箱 内置的 BFG 算法进行训练。

再训练集和测试集。对于每个待研究系统实例，我们随机抽取 10000 条执行轨迹作为再训练数据集，并抽取 1000 条轨迹作为测试集。表 II 列出了这些系统在再训练数据集上使用原始控制器时的安全率（SR），其中使用 $\frac{\#\ safe\ trials}{\#total\ trials}$表示系统的安全程度。如表 II 所示，

表 II 六个主题信息物理系统实例的详细信息

实例	原始深度神经网络控制器	SR
ACC#1	类型：前馈神经网络 结构: [10, 10, 10] 训练算法：LMBP *	9682/10000
ACC#2	类型：前馈神经网络 结构: [30, 30, 30] 训练算法: BFG**	8200/10000
AFC#1	类型：前馈神经网络 结构: [15, 15, 15] 训练算法：LMBP	8287/10000
AFC#2	类型：前馈神经网络 结构： [30, 30, 30] 训练算法：LMBP	8661/10000
WT#1	类型：前馈神经网络 结构: [5, 5, 5] 训练算法：BFG	8404/10000
WT#2	类型：前馈神经网络 结构: [15, 15, 15] 训练算法：BFG	9808/10000

*LMBP： *Levenberg–Marquardt* 反向传播（*LMBP*）算法 [29] **BFG： *BFGS* 拟牛顿（*BFG*）算法 [30]

表 III NSGA-II 算法的参数

NSGA-II 的参数	值
种群规模	40
最大生成量	10
目标数量	3
决策变量数量	6

待修复的 6 个实例的安全率介于 82.0% 至 98.1% 之间。

C. 软件和硬件依赖项

所有研究对象信息物理系统均在 Simulink 中实现，并依赖 MATLAB 工具箱 模型预测控制、控制系统和深度学习。实验在 AWS 云实例上进行，实例类型为 c4.2xlarge （ 2.9 GHz Intel Xeon E5-2666 v3，15G RAM）。对于深度神经网络控制器的重新训练，我们使用 MATLAB 的深度学习工具箱 功能，并在一台配备 3.3GHz Intel i9-10940X CPU、64G RAM 和两块 NVIDIA RTX A6000 GPU 的服务器上运行。评估总共在 20 个 AWS 实例上耗时约 480 小时（20 天 $*$ 24 小时）。

为在第 IV-C 节中进行基于多目标优化的修复，我们选择 NSGA-II 作为求解器。评估中采用的 NSGA-II 主要参数列于表 III。这里，我们将 NSGA-II 的种群规模和最大迭代代数分别设为 40 和 10。目标函数的数量为 3，其中第二个目标函数（关于原始控制信号与控制补丁信号之间的距离）在第 IV-C 节中被实例化为空间约束和时间约束。决策变量的数量等于单个最优解的维度，即 6，包含

表 IV 原始系统以及采用三种最优选择策略修复后的系统的安全率。每个系统实例的最佳表现者均已突出显示。（训练集：用于重新训练的数据集。测试：测试集。）

		ACC#1		ACC#2		AFC#1		AFC#2		WT#1		WT#2	
		训练集	test	训练集	test	训练集	test	训练集	test	训练集	test	训练集	test
原始		96.8%	97.4%	82.0%	84.1%	82.9%	85.4%	86.6%	85.5%	84.0%	84.0%	98.1%	97.7%
已修复	MinSat	75.6%	76.0%	95.4%	96.7%	82.3%	83.7%	91.5%	91.9%	86.1%	85.8%	98.2%	97.8%
	Similar	97.8%	97.9%	81.0%	81.9%	96.0%	96.3%	81.1%	81.5%	86.8%	85.9%	98.2%	97.8%
	Random	96.3%	97.1%	71.9%	72.3%	99.4%	99.6%	90.5%	90.3%	93.1%	92.5%	98.5%	98.1%

2 个时间戳 $t^{\mathrm{L}}_{\mathbf{c}'}$ 和 $t^{\mathrm{U}}_{\mathbf{c}'}$，以及 4 个用于生成控制信号补丁$\mathbf{c}'$ 的插值点。

VI. 评估结果

本节介绍用于评估我们提出的 AUTOREPAIR 框架的评估结果。

RQ1：AUTOREPAIR 是否能有效提升 DNN 赋能的 CPS 的安全性?

此 RQ 旨在评估 AUTOREPAIR 是否确实能够提升这些目标信息物理系统的安全性。我们分别使用原始控制器和修复后的控制器，在再训练数据集和测试集上运行这些目标信息物理系统，并将系统执行过程中来自再训练数据集和测试集的输入信号馈入系统。

RQ1 的实验结果如表 IV 所示，其中比较了这些系统采用原始 DNN 控制器和修复后 DNN 控制器时的安全率。我们注意到，在大多数情况下，与原始系统相对较低的安全率相比，在不同配置下，采用修复后控制器的系统都能实现显著更高的安全率。例如，这些系统在再训练集上的安全率至少提高了 0.4%（WT#2），最多提高了 16.5%（AFC#1）；在测试集上的安全率至少提高了 0.4%（WT#2），最多提高了 14.2%（AFC#1）。这些结果清楚地展现了 AUTOREPAIR 在安全性方面的优势：它确实成功修复了原始系统中易受攻击的特性。此外，测试集上的安全性提升也证明了 AUTOREPAIR 的泛化能力。

我们还注意到修复后安全率下降的情况，例如 MinSat 下的 ACC#1。这一问题可能由该方法中的多种因素造成，包括重新训练的质量。在实践中，我们可以通过调整控制器重新训练时的超参数，并尝试使用其他策略运行 AUTOREPAIR 来缓解这一问题。

RQ1 的回答：AUTOREPAIR 能够有效修复不安全的深度神经网络控制器，从而提高整个系统的安全性和可靠性。

RQ2：最优选择策略如何影响 AUTOREPAIR 的有效性?

在 RQ2 中，我们研究信号修复阶段（第 IV-C 节）帕累托前沿的不同最优选择策略如何影响 AUTOREPAIR 的有效性。在这项实验中，我们比较 AutoRepair 在第IV-C节介绍的三种最优选择策略下的表现。

如表 IV 所示，MinSat 在 6 种情况中的 4 种优于原始控制器，但对 ACC#1 无效；随机策略在 6 种情况中的 4 种优于原始控制器，并在 6 种情况中的 3 种优于另外两种策略，但对 ACC#2 无效。这些结果表明，MinSat 和随机策略总体上有效，但在某些情况下也可能无效。正如我们在 RQ1 中分析的那样，在实践中，我们可以通过提高重新训练的质量并尝试不同的策略来提升 AUTOREPAIR 的有效性，从而使修复后的系统达到更高的安全率。

此外，相似在 4/6 种情况下优于原始控制器，而在剩余的 2 种情况下，其表现与原始控制器相似；然而，仅在 1/6 种情况下，其优于另外两种策略。这些结果表明，相似在改变DNN控制器的特性方面相对较弱，因为它总是倾向于选择与原始控制信号偏差最小的控制信号补丁。

RQ2 的回答：在三种最优选择策略中，MinSat 和随机策略在大多数情况下都有效；相似策略在许多情况下也有效，但在改变 DNN控制器的特性方面较弱。

研究问题3：AUTOREPAIR 是否会影响由深度神经网络驱动的信息物理系统在安全性以外其他指标方面的性能?

AutoRepair 对系统各方面性能的影响如图 4所示。具体而言，我们根据不同指标对性能数据进行归一化处理，并展示原始系统与修复后系统之间的性能差异。蓝线和橙线分别表示原始系统和修复后系统的归一化性能。

如图4所示，与原始系统相比，修复后的系统性能几乎没有下降。例如，在 ACC#2 中，虽然 AUTOREPAIR 提高了原始系统的安全性，但几乎不会影响行驶距离性能（即自车行驶的距离）或速度性能（即处于安全状态时速度不应过低）。这是因为，在我们的修复算法中，我们会尽量减小控制信号补丁与原始控制信号之间的偏差。当然，也有一些情况下，安全性的提升伴随着某些方面的性能下降（例如 AFC#2 的误差性能）。这种性能下降的严重程度取决于这些指标的重要性。

表 V AUTOREPAIR 的时间成本（时间（秒））

	sim.	平均成本	违规率	总成本
ACC#1	1.42	393.81	318/10000	1.25×10^5
ACC#2	2.08	558.92	1800/10000	1.01×10^6
AFC#1	2.03	480.69	1713/10000	8.23×10^5
AFC#2	2.54	631.66	1339/10000	8.46×10^5
WT#1	1.07	171.77	1596/10000	2.74×10^5
WT#2	1.36	262.83	192/10000	5.05×10^4

此外，我们还观察到，AUTOREPAIR 在提高安全性的同时，也能改善其他一些方面的性能，例如 WT#1 的稳定性。这是因为在某些情况下，安全性与其他性能指标是一致的。例如，稳定性指标要求系统在一段时间内保持安全状态，因此提高安全性也能改善系统在稳定性方面的性能。

对研究问题3的回答：AUTOREPAIR 可以在几乎不牺牲其他方面性能的情况下，提高系统的安全性。有时，它甚至可以提升与安全性相关方面的性能。

RQ4：AUTOREPAIR 的效率如何?

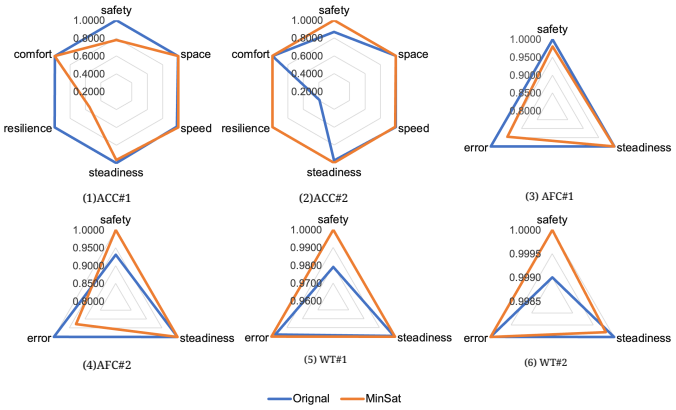
表 V 列出了AUTOREPAIR的时间开销详情，包括一次系统执行的时间开销、修复一次不安全执行的平均开销、再训练集中不安全执行的比例，以及修复再训练集中所有不安全轨迹的总时间开销。

首先，如表 V 所示，修复单次系统执行的成本取决于单次系统执行的时间成本，而且还取决于多目标优化算法的配置。NSGA-II 的参数设置见表 III；用户可以通过调整这些参数，在效率和修复效果之间取得平衡。其次，AUTOREPAIR 的总耗时还取决于收集的数据集中用于重新训练的违规系统执行所占的比例。如果系统相对安全，所需的修复工作就会更少。总而言之，由于我们的方法能够有效提升 DNN赋能的CPS 的安全性，付出这些修复工作是值得的。

对 RQ4 的回答：AUTOREPAIR 的时间成本取决于系统执行时间、优化算法的超参数以及系统的安全状态。对于安全关键系统，为实现更高的安全率，投入此类修复工作是值得的。

VII. 相关工作

DNN控制器。随着人工智能的快速发展，从业者考虑用DNN控制器替代传统控制器 [4], [5], [31]–[34]。深度神经网络控制器可以通过不同方式进行训练，包括监督学习 [35] 和深度强化学习 *(DRL)* [36]。监督学习需要收集原始信息物理系统的系统执行，以训练新的DNN控制器。例如，Hertneck 等人 [35] 使用 DNN 近似鲁棒模型预测控制控制器。相比之下，DRL 遵循“试错”范式，通过与环境交互来学习最佳策略。近期一项工作 [37]将 DRL 用于深度神经网络控制器训练。



Fi图4. AUTOREPAIR 对系统性能的影响 。（研究问题3）

深度神经网络修复。 人们采用了不同的修复机制来提升深度神经网络模型的性能和可靠性。现有的深度神经网络修复工作大致可分为两类——对抗训练 和 参数修改。

对抗训练 是指先收集对抗样本，然后使用带有修正后的标签的这些样本重新训练模型的方法。这种方法已被广泛用于修复图像分类器，例如， [38]–[43]。尽管我们的工作可以被视为对抗训练的一种具体实践，但我们在获取对抗样本标签的方式上不同于现有工作，即鉴于DNN赋能的CPS具有很高的复杂性，我们通过优化搜索控制信号的修正方案，从而在系统层面实现安全性。

参数修改 是指通过直接修改深度神经网络的参数来改变对对抗样本 [44]–[46] 的推理结果的方法。例如， [44], [46] 是早期尝试，旨在定位可疑的深度神经网络参数，然后对其进行优化以实现深度神经网络修复。我们的工作与这些研究相互正交，因为我们不会直接操控深度神经网络的权重。

DNN控制器的修复。 已有多项工作尝试修复DNN控制器。Yang等人 [47] 提出，先攻击DNN控制器，再利用识别出的对抗样本进行重新训练。然而，与 [47] 中考虑独立控制器的情况相比，我们的目标是由DNN控制器控制的物理系统；此外，我们在系统层面定义安全要求。另一项相关工作是 [48]，其中Zhou等人使用假定的安全 预言机修复基于学习的控制器。与 [48],相比，我们的方法通过在控制决策空间中搜索修正方案，实现了控制器修复流程的自动化。

安全强化学习。 强化学习（强化学习）是训练控制器的一种典型方法。在强化学习中，有一系列工作 [49]–[52]旨在确保训练后的控制器的安全性；具体而言，安全屏蔽 [49] 提供了一种与我们的框架相似的机制，该框架会修复动作

会触发不安全系统行为。不过，安全强化学习与我们的框架之间存在若干重大差异。首先，强化学习中的许多问题设定局限于马尔可夫决策过程（*MDP*），其中包含有限个状态和转移；相比之下，我们考虑的是对具有稠密时间实值动态特性的非线性连续对象进行控制，而这类对象本质上难以验证。此外，诸如安全屏蔽之类的安全机制可以在运行时实施——检测到不安全动作时，可以激活屏蔽来阻止该动作；而在我们的场景中，由于系统具有稠密时间特性，在运行时禁止单个动作并不能防止不安全情况发生，这也解释了为什么我们要以离线方式分析和修复完整信号。

VIII. 结论与未来工作

尽管DNN赋能的CPS日益受到关注，如何增强DNN控制器仍是一项尚未解决的挑战。在本研究中，我们提出了AUTOREPAIR，这是一种通用的自动化修复框架，可在修复过程中考虑系统级安全要求。AUTOREPAIR采用一种新颖的信号诊断与修复方法：给定系统执行轨迹，该方法能够找出对初始控制信号序列进行最小修改的方案，以避免产生不安全的系统输出。通过将最优修改应用于控制序列，AUTOREPAIR能够获得一条新的安全保障执行轨迹，并可用其修复初始DNN控制器。实验结果表明，AUTOREPAIR能够有效修复不安全的DNN控制器，并提升整个系统的安全性。未来，我们计划扩展该框架，使其支持由更多样化的高级时序逻辑形式体系所规定的安全要求，从而提供更可靠的安全保障与增强能力，并加速DNN在各领域多样化工业CPS中的应用。

致谢

本研究部分经费来自加拿大首席研究卓越基金，作为阿尔伯塔大学未来能源系统研究计划的一部分；加拿大 CIFAR 人工智能讲席计划；Amii RAP 计划；加拿大自然科学与工程研究委员会（NSERC No.RGPIN-2021-02549、No.RGPAS-2021-00034、No.DGECR-2021-00019）；以及日本学术振兴会科研补助金 No.JP19H04086、No.JP20H04168、No.JP21H04877 和 JST-Mirai 计划补助金 No.JPMJMI20B8。D. Lyu 还获得了 JST SPRING 补助金 No. JPMJSP2136 的资助。

参考文献

[1] M. A. Johnson 和 M. H. Moradi，《*PID* 控制》。Springer，2005。[2] E. F. Camacho 和 C. B. Alba，《模型预测控制》。Springer Science & Business Media，2013。[3] S. Alsalehi、N. Mehdipour、E. Bartocci 和 C. Belta，“基于神经网络的多智能体系统时空规范控制”，2021。[4] W. Liu、N. Mehdipour 和 C. Belta，“满足安全约束的信号时序逻辑规范循环神经网络控制器”，*IEEE Control Systems Letters*，2021。[5] S. Yaghoubi、G. Fainekos 和 S. Sankaranarayanan，“存在干扰时使用控制屏障函数训练神经网络控制器”，载于*2020 IEEE* 第 *23* 届智能交通系统国际会议（*ITSC*）。IEEE，2020，第 1–6 页。

[6] Y. Annpureddy、C. Liu、G. Fainekos和 S. Sankaranarayanan，“S-TaLiRo：用于混合系统时序逻辑证伪的工具”，载于第17届国际系统构造与分析工具和算法会议论文集，TACAS 2011，《计算机科学讲义》系列，P. A.Abdulla 和 K. R.M. Leino 编，第6605卷。Springer，2011年，第254–257页。[7] A.Donz´e，“Breach：用于混合系统验证与参数综合的工具箱”，载于第22届国际计算机辅助验证会议论文集，《CAV’10》系列。柏林、海德堡：Springer-Verlag，2010年，第167–170页。[8] Z. Zhang、G. Ernst、S. Sedwards、P.Arcaini 和I. Hasuo，“由蒙特卡洛树搜索引导的混合系统双层证伪”，《IEEE集成电路与系统计算机辅助设计学报》，第37卷，第11期，第2894–2905页，2018年11月。[9] Z.Zhang、I. Hasuo 和 P.Arcaini，“用于混合系统证伪中布尔连接词的多臂老虎机”，载于第31届国际计算机辅助验证会议论文集，CAV 2019，第一部分，《计算机科学讲义》系列，I. Dillig和 S.Tasiran 编，第11561卷。Springer，2019年，第401–420页。[10] Z.Zhang、D. Lyu、P. Arcaini、L.Ma、I.Hasuo 和 J.Zhao，“使用由QB-鲁棒性引导的蒙特卡洛树搜索进行有效的混合系统证伪”，载于第33届国际计算机辅助验证会议论文集，CAV 2021，《计算机科学讲义》系列，A.Silva 和 K. R.M. Leino 编，第12759卷。Springer，2021年，第595–618页。[11] Z. Zhang、P.Arcaini和I. Hasuo，“通过搜索空间变换实现约束（不）等式条件下的混合系统证伪”，《IEEE集成电路与系统计算机辅助设计学报》，第39卷，第11期，第3674–3685页，2020年。[12] Z.Zhang和 P.Arcaini，“基于高斯过程的混合系统证伪置信度估计”，载于第24届国际形式化方法研讨会，FM 2021，《计算机科学讲义》系列，M. Huisman、C. P˘as˘areanu 和N. Zhan 编，第13047卷。Springer，2021年，第330–348页。[13] Z.Zhang、D. Lyu、P. Arcaini、L.Ma、I. Hasuo和 J.Zhao，“Falsifai：由时间感知覆盖准则引导的人工智能赋能混合控制系统证伪”，《IEEE软件工程学报》，2022年。[14] O.Maler和D. Nickovic，“连续信号时序性质的监测”，载于《定时与容错系统的形式化技术、建模与分析》。Springer，2004年，第152–166页。[15] G. Ernst、P.Arcaini、I. Bennani、A. Donz´e、G.Fainekos、G. Frehse、L.Mathesen、C.Menghi、G.Pedrielli、M. Pouzet、S. Yaghoubi、Y.Yama-gata和 Z.Zhang，“ARCH-COMP2020类别报告：证伪”，载于第7届连续与混合系统应用验证国际研讨会（ARCH20）论文集，《计算领域EPiC系列》，G. Frehse 和M. Althoff编，第74卷。EasyChair，2020年，第140–152页。[16] G. Ernst、P. Arcaini、I. Bennani、A. Chandratre、A. Donz´e、G.Fainekos、G. Frehse、K. Gaaloul、J.Inoue、T.Khandait、L.Mathesen、C.Menghi、G. Pedrielli、M. Pouzet、M. Waga、S. Yaghoubi、Y. Yamagata和 Z. Zhang，“ARCH-COMP 2021类别报告：带有结果验证的证伪”，载于第8届连续与混合系统应用验证国际研讨会（ARCH21）论文集，《计算领域EPiC系列》，G. Frehse 和M. Althoff编，第80卷。EasyChair，2021年，第133–152页。[17] A. Pnueli，“程序的时序逻辑”，载于第18届计算机科学基础年度研讨会（sfcs1977）。ieee，1977年，第46–57页。[18] A. Donz´e、T.Ferrere 和 O. Maler，“用于STL的高效鲁棒监测”，载于国际计算机辅助验证会议。Springer，2013年，第264–279页。[19] J. V.Deshmukh、A. Donz´e、S. Ghosh、X.Jin、G.Juniwal和 S. 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》，第51卷，第1期，第5–30页，2017年。[20] S. Jak˘si´c、E.Bartocci、R. Grosu、T. Nguyen和D. Ni˘ckovi´c，“基于编辑距离的STL定量监测”，《系统设计中的形式化方法》，第53卷，第1期，第83–112页，2018年。[21] T. Elsken、J.H.Metzen和 F. Hutter，“神经架构搜索：综述”，《机器学习研究杂志》，第20卷，第1期，第1997–2017页，2019年。[22] Z.Zhang、P. Arcaini和X. Xie，“信号时序逻辑监测的在线重置”，《IEEE集成电路与系统计算机辅助设计学报》，第41卷，第11期，第4421–4432页，2022年。[23] E. Bartocci、T. Ferr`ere、N. Manjunath 和 D.Ni˘ckovi´c，“使用STL定位Simulink/Stateflow模型中的故障”，载于第21届混合系统：计算与控制国际会议论文集（CPS周的一部分），2018年，第197–206页。

[24] G. Ernst、P. Arcaini、G. Fainekos、F. Formica、J. Inoue、T. Khandait、M. M. Mahboob、C. Menghi、G. Pedrielli、M. Waga、Y. Yamagata 和 Z. Zhang，“Arch-comp 2022 类别报告：资源无上限的证伪”，载于第九届应用验证连续与混合系统国际研讨会论文集（*ARCH22*），EPiC Series in Computing 丛书，G. Frehse、M. Althoff、E. Schoitsch 和 J. Guiochet 编，第 90 卷。EasyChair，2022 年，第 204–221 页。[25] K. Deb、A. Pratap、S. Agarwal 和 T. Meyarivan，“一种快速且精英的多目标遗传算法：NSGA-II”，*IEEE* 进化计算汇刊，第 6 卷，第 2 期，第 182–197 页，2002 年。[26] MathWorks，“使用模型预测控制的自适应巡航控制系统”，
h
t
t
p
s://
w
w
w.
math
works.com/help/mpc/ug/adaptive-cruise-control-using-model-predictive-controller.html。[27] X. Jin、J. V. Deshmukh、J. Kapinski、K. Ueda 和 K. Butts，“动力传动系统控制验证基准”，载于第 *17* 届混合系统：计算与控制国际会议论文集，2014 年，第 253–262 页。[28]MathWorks，“水箱 Simulink 模型”，
https://www.mathworks.com/help/slcontrol/gs/watertank-simulink-model.html。[29] C. Lv、Y. Xing、J. Zhang、X. Na、Y. Li、T. Liu、D. Cao 和 F.-Y. Wang，“用于安全关键型信息物理系统状态估计的多层神经网络 Levenberg–Marquardt 反向传播训练”，*IEEE* 工业信息学汇刊，第 14 卷，第 8 期，第 3436–3446 页，2018 年。[30] P. E. Gill、W. Murray 和 M. H. Wright，实用优化。SIAM，2019 年。[31] J. V. Deshmukh、J. P. Kapinski、T. Yamaguchi 和 D. Prokhorov，“学习具有安全保证的动力系统深度神经网络控制器”，载于*2019* 年 *IEEE/ACM* 计算机辅助设计国际会议（*ICCAD*）。IEEE，2019 年，第 1–7 页。[32] S. A. Nivison 和 P. P. Khargonekar，“面向飞行应用的鲁棒、稀疏激活深度循环神经网络控制器的开发”，载于*2018* 年 *IEEE* 决策与控制会议（*CDC*）。IEEE，2018 年，第 384–390 页。[33] A. Gilra 和 W. Gerstner，“尖峰神经网络中通过局部学习实现非线性电机控制”，载于国际机器学习会议。PMLR，2018 年，第 1773–1782 页。[34] J. Ferlez、X. Sun 和 Y. Shoukry，“用于非线性系统控制的双层格点神经网络架构”，载于*2020* 年第 *59* 届 *IEEE* 决策与控制会议（*CDC*）。IEEE，2020 年，第 2198–2203 页。[35] M. Hertneck、J. K¨ohler、S. Trimpe 和 F. Allg¨ower，“学习具有保证的近似模型预测控制器”，*IEEE* 控制系统快报，第 2 卷，第 3 期，第 543–548 页，2018 年。[36] K. Arulkumaran、M. P. Deisenroth、M. Brundage 和 A. A. Bharath，“深度强化学习：简要综述”，*IEEE* 信号处理杂志，第 34 卷，第 6 期，第 26–38 页，2017 年。[37] J. Song、D. Lyu、Z. Zhang、Z. Wang、T. Zhang 和 L. Ma，“当信息物理系统遇上人工智能：基准、评估与未来方向”，载于第 *44* 届软件工程国际会议：实践中的软件工程论文集，2022 年，第 343–352 页。[38] I. J. Goodfellow、J. Shlens 和 C. Szegedy，“解释并利用对抗样本”，2014 年。[39] E. Wong 和 Z. Kolter，“通过凸外部对抗多面体对抗样本提供可证明的防御”，载于国际机器学习会议。PMLR，2018 年，第 5286–5295 页。[40] A. Shafahi、M. Najibi、A. Ghiasi、Z. Xu、J. Dickerson、C. Studer、L. S. Davis、G. Taylor 和 T. Goldstein，“免费的对抗训练！”，载于第 *33* 届神经信息处理系统国际会议论文集，2019 年，第 3358–3369 页。[41] S. Ma、Y. Liu、W.-C. Lee、X. Zhang 和 A. Grama，“Mode：通过状态差异分析和输入选择自动调试神经网络模型”，载于*2018* 年第 *26* 届 *ACM* 欧洲软件工程会议与软件工程基础研讨会联合会议论文集，2018 年，第 175–186 页。[42] B. Yu、H. Qi、Q. Guo、F. Juefei-Xu、X. Xie、L. Ma 和 J. Zhao，“DeepRepair：面向真实世界运行环境的深度神经网络风格引导修复”，*IEEE* 可靠性汇刊，第 1–16 页，2021 年。[43] X. Gao、R. K. Saha、M. R. Prasad 和 A. Roychoudhury，“基于模糊测试的数据增强以提升深度神经网络的鲁棒性”，载于*ACM/IEEE* 第 *42* 届软件工程国际会议论文集，ICSE ’20 丛书。美国纽约州纽约市：计算机协会，2020 年，第 1147–1158 页。

[44] J.Sohn、S. Kang和S.Yoo，“深度神经网络的基于搜索的修复”，2019。[45] H. Wang、B.Ustun 和 F. Calmon，“无需重新训练的修复：利用反事实分布避免差异影响”，载于机器学习国际会议。PMLR，2019，第 6618–6627 页。[46] H. Zhang和W. Chan，“Apricot：一种修复深度学习模型的权重适应方法”，载于*2019* 年第 *34* 届 *IEEE/ACM*自动化软件工程国际会议（*ASE*），2019，第 376–387页。[47] X. Yang、T.Yamaguchi、H.-D.Tran、B. Hoxha、T.T.Johnson和 D. Prokhorov，“利用可达性分析修复神经网络”，载于定时系统的形式化建模与分析：第 *20*届国际会议，*FORMATS 2022*，波兰华沙，*2022* 年 *9* 月 *13–15*日，会议论文集。Springer，2022，第221–236 页。[48] W. Zhou、R. Gao、B. Kim、E. Kang 和 W. Li，“运行时安全性引导的策略修复”，载于运行时验证国际会议。Springer，2020，第 131–150 页。[49] M. Alshiekh、R. Bloem、R. Ehlers、B. K¨onighofer、S. Niekum 和 U. Topcu，“通过安全屏蔽实现安全强化学习”，载于 *AAAI*人工智能会议论文集，第 32 卷，2018 年。[50] G.Anderson、A. Verma、I. Dillig 和 S.Chaudhuri，“具有形式化验证探索的神经符号强化学习”，神经信息处理系统进展，第33 卷，第6172–6183页，2020 年。[51] H. Zhu、Z. Xiong、S. Magill和 S. Jagannathan，“可验证强化学习的归纳综合框架”，载于 第 *40* 届 *ACM SIGPLAN* 编程语言设计与实现会议论文集，2019，第 686–701 页。[52] F. Berkenkamp、M. Turchetta、A. Schoellig 和 A. Krause，“具有稳定性保证的安全模型型强化学习”，神经信息处理系统进展，第 30 卷，2017 年。