∂CBD：可微因果框图

Thomas Beckers*范德堡大学，美国纳什维尔 tho mas.beckers@vanderbilt.edu

Ján Drgoňa*约翰斯·霍普金斯大学，美国马里兰州 jdrgona1@jh.edu

Truong X. Nghiem*中佛罗里达大学，美国奥兰多 truong.nghiem@ucf.edu

arXiv:2602.07581v1 [eess.SY] 7 Feb 2026

摘要

现代信息物理系统（CPS）融合了物理、计算与学习，因此需要兼具可组合、可学习和可验证性的建模框架。然而，现有方法将这些目标彼此割裂：因果块图（CBD）支持模块化系统互连，却缺乏可用于学习的可微性；可微编程（DP）支持端到端的基于梯度的优化，却无法充分保证正确性；而基于契约的验证框架与数据驱动的模型细化仍基本脱节。为解决这些局限，我们提出可微因果框图（∂CBD），这是一种统一形式体系，整合了上述三种视角。我们的方法（i）保留 CBD 的组合结构和执行语义；（ii）引入假设–保证（A–G）契约，以支持模块化正确性推理；（iii）提出基于残差的契约，作为可微的轨迹级证书，与自动微分（AD）兼容，从而支持基于梯度的优化与学习。这些要素共同构成一种可扩展、可验证且可训练的建模流程，在保持因果性和模块化的同时，支持面向信息物理系统、由数据、物理规律和约束共同指导的优化。

关键词

因果块图、假设—保证、可微编程、物理信息机器学习

ACM 参考文献格式：

Thomas Beckers、Ján Drgoňa 和 Truong X. Nghiem。2018年。∂CBD：可微因果模块图。载于《*ACM/IEEE* 国际信息物理系统会议论文集》（*ICCPS 2026*）。ACM，纽约州纽约市，美国，12 页。https://doi.org/XXXXXXX.XXXXXXX

1 引言

信息物理系统日益融合基于模型的设计与数据驱动组件，以在多种运行条件下实现高性能 [42, 27, 4, 10]。随着这些系统的复杂性和自主性不断提升，工程师面临三方面的权衡：通过组合性管理结构复杂性、通过可学习性利用数据调整模型和控制器，以及通过可验证性确保安全性以及正确性 [48]。如今的方法通常只能满足其中一两项要求，却很少能在单一且易于处理的工作流程中兼顾全部三项。因果块图（CBD）仍是信息物理系统设计的通用语言，参见 [28], ，这得益于其清晰的执行语义和对模块化组合的支持；然而，它们并非原生可微，因此与现代优化和学习流程的联系较弱。可微编程（DP）框架提供端到端梯度和灵敏度分析，但其运行所依托的计算图，其语义侧重于数值执行，而非正确性和组合推理 [31]。现代 DP 系统通常依赖自动微分（AD）作为其计算核心，从而实现高效的梯度传播，但在形式化契约的指定或验证方面支持有限。基于契约的验证，尤其是通过假设-保证规约实现的验证，可提供可扩展认证 [12], ，但通常与学习相互脱节，因此难以在数据驱动优化过程中落实安全性和性能约束。这种脱节迫使从业者在结构化、可认证但难以优化的模型，与缺乏可靠保证的学习工具之间作出选择。

本文介绍了可微因果框图（∂CBD），这是一种建模与优化框架，将 CBD 组合、假设-保证合约语义和可微编程整合到单一流程中。在 ∂CBD 中，每个块都配备了明确的接口和时间语义、可连续、离散或混合的执行映射，以及一组用于形式化假设和保证的可微合约残差。这些残差可以量化属性满足程度，例如无源性、增益界和李雅普诺夫证书；并且能够在互连间组合，同时保留执行语义。关键的是，通过速率提升和循环展开，∂CBD 组合可在有限时域上生成可执行自动微分的计算图，从而使梯度能够以规范的方式传播通过基于物理的模型（例如 ODE/DAE 求解器、不动点层和优化层）以及数据驱动组件（例如神经网络）。因此，契约成为可微证书，可在训练过程中加以约束或正则化，使学习过程能够由物理、安全性和性能约束引导，而不再仅仅依赖经验损失。

相关工作。自动微分（AD）已成为现代机器学习和科学计算的计算支柱，使我们能够通过任意复杂的程序高效地计算机器精度梯度。在机器学习中，反向模式自动微分是 TensorFlow、Py- Torch 和 JAX 等深度学习框架的基础，支持对大规模神经网络架构进行优化 [1, 37, 16]。在科学计算和最优控制中，自动微分还可应用于微分方程，通过数值求解器提供灵敏度信息 [6, 29, 30, 18]。这些进展确立了自动微分作为可微编程这一范式的关键推动力，

*作者按字母顺序排列；所有作者对本研究的贡献相同。

允许为个人或课堂使用而免费制作本作品的全部或部分数字副本或纸质副本，前提是不得为营利或商业利益制作或分发副本，且副本首页须附有本声明及完整引文。对于本作品中归其他作者所有的组件，必须尊重其版权。注明出处后可以摘录。若要以其他方式复制、再版、发布到服务器或分发至列表，则须事先获得具体许可和/或支付费用。请向 permissions@acm.org 申请许可。*ICCPS 2026*，法国圣马洛© 2018年版权归所有者/作者所有。出版权已授权给 ACM。ACM ISBN 978-1-4503-XXXX-X/2018/06 https://doi.org/XXXXXXX.XXXXXXX

该范式在单一计算图中统一了仿真、学习和优化 [31, 9, 47, 15]。在此基础上，所提出的 ∂CBD 框架将自动微分语义直接嵌入 CBD 的组合结构，从而在信息物理系统中实现可扩展、可验证且具有物理依据的学习。

贡献。 首先，我们提出一种因果块图的可微扩展，在保留组合结构和执行语义的同时，实现端到端的基于梯度的优化。其次，我们将假设—保证契约作为可微残差嵌入建模流程，将模块化规范转化为面向优化的证书，以指导并约束学习。第三，我们构建了从 ∂CBD 到可由自动微分执行的图的规约，并确保在异构组件之间进行定义良好的灵敏度传播，为信息物理系统的可扩展训练与验证奠定基础。总之，这些要素为设计兼具可组合性、可学习性和可验证性的系统，建立了实用且有原则的基础。

2 方法

我们引入可微因果模块图 （∂CBD），这一形式体系统一了：(i) CBD 的组合式建模，(ii) 用于模块化正确性验证的假设—保证（A–G）契约，以及 (iii) 用于端到端学习的可微编程（DP）。通过结合 CBD 的结构清晰性、A–G 契约的组合式验证能力，以及自动微分（AD）框架的可扩展性，我们的方法能够对复杂信息物理系统进行可扩展、可验证且可组合的 建模、学习和优化。

在所提出的 ∂CBD 框架中，每个块都配备了一个可微契约，用于编码其物理、安全或运行属性。CBD 组合（串行、并行或反馈）通过契约组合保留这些属性，并转换为可由自动微分执行的有向无环图（DAG）。由此得到的系统既可作为因果图（用于仿真和模块化推理），也可作为可微程序（用于学习和优化）。假设—保证契约作为平滑的惩罚函数或证书函数嵌入计算图中，使形式化正确性推理能够与基于梯度的训练并存。形式上， ∂CBD 框架由五个抽象层组成：

(1) **结构层：** 将带契约的块定义为因果块图中的基本单元（第 2.1 节）；(2) **语义层：** 定义块的组合规则和因果执行语义（第 2.2 节）；(3) **契约层：** 为块指定假设-保证规约，以确保组合正确性（第 2.3 节）；(4) **可微层：** 通过自动微分揭示因果块图的导数（第 2.4 节）。(5) **优化层：** 支持针对因果块图进行可验证的基于梯度的学习和优化（第 2.5 节）。

这些层共同构成了统一的建模与学习流水线，在保留因果性和组合结构的同时，通过自动微分兼容原语提供局部梯度，并借助可微契约实现大规模系统级验证。这种集成构成了 ∂CBD 的方法论核心。

2.1 带契约的因果模块图

因果模块图（CBD）是一种广泛采用的建模形式体系，适用于信息物理系统 [28] 的实际基于模型的设计。最具代表性的 CBD 实现或许是 Simulink [20], ，它近期扩展了基于契约的语义和精化功能 [43, 52]。类似地， Ptolemy II 将 CBD 作为其核心建模范式，在统一的基于块的架构中提供了一个灵活的框架，用于组合异构计算模型 [38]。我们的框架采用了这些实现和相关文献中常用的核心 CBD 形式体系，具体如下。CBD 的假设—保证（A–G）契约扩展将在第 2.3 节介绍。

符号表示。 带下标的小写字母（例如 v_i）表示变量，大写字母（例如 V）表示变量集合 v_i，粗体大写字母（例如 $\boldsymbol{V}$）表示 V中变量的向量拼接，花体字母（例如 V）表示 $\boldsymbol{V}$的值空间。

信号。 我们首先定义信号，它是从时间到值的函数，用于表示系统中随时间变化的变量。时间域由非负实数表示，记作$\mathbb{R}^+$。信号可以是连续时间或离散时间。

定义 *2.1*（信号）。 连续时间信号 s 是从 $\mathbb{R}^+$ 到 S的函数，其中 S 是信号的非空值集合。离散时间信号 s 是分段常数的连续时间信号，其值仅在离散时刻 $k\tau$发生变化，其中自然数 $k \in \mathbb{N}$是时间步，正实数常量 τ是周期，即 $\forall t \in [k\tau, (k+1)\tau)\ s(t) = s(k\tau)$。

在上述定义中，值集 S可以是任意非空集合，包括布尔集等由离散值构成的有限集合。然而在实践中， S通常是实数集 $\mathbb{R}$。为简单起见，在本文中我们假设 $S \equiv \mathbb{R}$。

模块。 在 CBD 中，块 表示一个组件，通常会将某些输入信号转换为某些输出信号，并具有定义明确的行为。在我们的框架中，每个块都可以关联一个假设-保证契约，用于规定给定条件下该块的行为。契约是我们框架中验证和学习的关键要素，它规定了安全需求和物理先验等系统属性，并可将其整合到学习中。块可以是连续时间、离散时间或混合型（同时具有连续时间和离散时间行为），并且可以关联多个采样周期。我们采用以下形式化定义来描述带契约的块，如图 1a 所示。

定义 *2.2*（带契约的块）。块是一个元组

$$\mathcal{B} = (U, X, Y, T, F, g, \mathrm{Init}, C),$$

- $U = \{u_1, \ldots, u_n\}$ 是该块的输入变量集合，每个变量都表示该块使用的信号。 $\boldsymbol{U}$是所有输入变量的向量拼接，其值空间为 U。
- $X = \{x_1, \ldots, x_p\}$ 是内部状态变量集合，每个变量都表示该块内部的信号。 $\boldsymbol{X}$是所有状态变量的向量拼接，其值空间为 X。
- $Y = \{y_1, \ldots, y_m\}$ 是输出变量集合，每个变量都表示该块生成的信号。 $\boldsymbol{Y}$是所有输出变量的向量拼接，其值空间为 Y。

• T是该块的时间语义，即由唯一采样周期组成的集合 $T = \{\tau_1, \ldots, \tau_q\}$，$\tau_i \in \mathbb{R}^+$，其中约定零采样周期表示连续时间。$T$必须至少包含一个元素。

• $F = \{f(\tau)\}_{\tau \in T}$是与采样周期 τ对应的状态转移函数 $f(\tau)$的集合。每个 $f(\tau)$ 仅更新状态变量的一个非空子集 $X(\tau) \subseteq X$，且 $X(\tau)$ 的集合是 X 的一个划分，即 $\cup_{\tau \in T} X(\tau) = X$，并且对于任意 $\tau, \tau' \in T$，$X(\tau) \cap X(\tau') = \emptyset$。$f(\tau)$ 的语义取决于其时间语义：

– 如果 $\tau = 0$：$X^{(0)}$ 中的所有状态信号都是连续时间且取连续值，并且 $f(0)$ 定义了一个微分方程：$\dot{X}(0)(t) = f(0)(X(t), U(t))$，该方程定义良好，且对于任意初始条件都有唯一解 $X(0)(t)$。– 如果 $\tau > 0$：$X^{(\tau)}$ 中的所有状态信号都是离散时间的，其取值可以是连续的，也可以是离散的。它们的值在每个离散时刻 $k\tau$ 根据差分方程更新：$X(\tau)((k+1)\tau) = f(\tau)(X(k\tau), U(k\tau))$。• g 是输出函数，用于根据当前状态和输入计算输出 Y：$Y(t) = g(X(t), U(t))$。它可以表示从 X 和 U 到 Y 的显式方程，也可以表示形式为 $Y(t) = h(Y(t), X(t), U(t))$ 的隐式方程，并由不动点求解器求解。• Init 是状态 X 的初始条件，即 $X(0) = \text{Init}$。• C 是该块所满足的契约集合（见第 2.3 节）。

假设*1*（块）。块必须满足以下条件 ing:

• 它至少有一个输入或输出，即集合 U、X 和 Y 可以为空，但 U 和 Y 不能同时为空。

• 其由状态转移函数 f（τ）和输出函数 g_τ 指定的动力学是因果的。• 其动力学由时间驱动，而非由事件驱动。

备注*1*。我们作如下说明。

• 如果 U 为空，则该块是生成信号的源块。如果 Y 为空，则该块是仅消耗信号的汇块。如果 X 为空，则该块是无状态或静态的；否则，该块是有状态或动态的。

• 块可以是混合型的，同时具有连续时间语义（$\tau = 0$）和离散时间语义（$\tau > 0$）。

• 本文其余部分将聚焦于连续值状态 X。离散状态变量的事件驱动动力学将在未来研究中探讨。

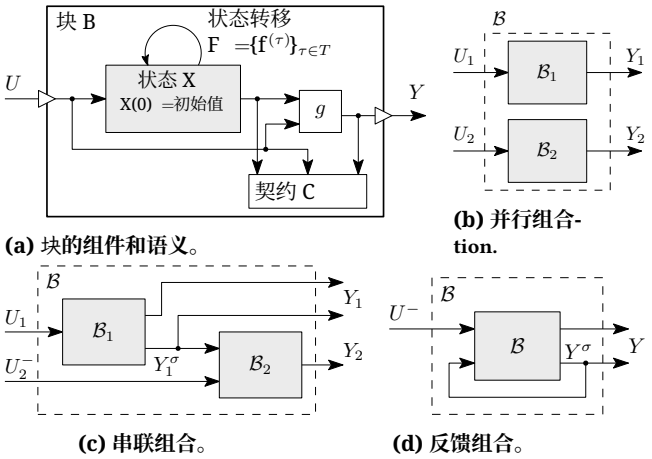
2.2 CBD 的执行与组合

给定块 B 及其输入信号 $U(t) = \{u_1(t) \ldots, u_n(t)\}$，即可执行该块以生成状态信号

$X(t) = \{x_1(t) \ldots,$

$x_p\ t$ $Y\ t = y_1\ t$ $y_m\ t$ $0 \le t \le t_f()\}$ 以及输出信号 $()\{()\ldots,()\}$，其中 t_f为终止时间，通过求解由 F 得出的因果的 微分方程和差分方程。下面简要介绍模块执行算法。

(1) 根据 T 中的离散采样周期，确定一系列时刻 $0 = t_0,\ t_1,\ \ldots\ldots,\ t_N = t_f$，其中，对于每个满足 $0 < i < N$ 的 t_i，都存在 $\tau \in T$，使得对于某个 $k \in \mathbb{N}$，$t_i = k\tau$。(2) 设置初始状态和输出：$X(0)$ $g\ X\ 0\ U\ 0$ Init 和 Y (0) (( ) ( ))。



F图 1：块及其组合的示意图

(3) 对于每个区间 $(t_i,\ t_{i+1}]$，从初始条件 $X(t_i)$ 出发，求解相应的微分方程和差分方程，以获得该区间内的状态值。然后，根据 g 计算输出信号 Y。

如果信号 $X(t)$ 和 $Y(t)$ 是由 B 根据输入信号 $U(t)$ 的执行语义生成的，我们记作 $(X(t)\ Y(t)) \in \text{Exec}(B\ U(t))$。因此，每个块都定义了一个从输入信号 $U(t)$ 到输出信号 $Y(t)$ 的输入–输出映射，使用相同的块符号表示记作 $Y = B(U)$。

模块组合。CBD 由相互连接的模块组成。通过在有向无环图（DAG）中组合各个模块及其连接，可以将 CBD“扁平化”为一个等价的块。本节其余部分将形式化两个块之间的单向连接以及模块的基本组合规则。

定义*2.3*（单向连接）。给定两个块 $B_i = (U_i, X_i, Y_i, T_i, F_i, g_i, Init_i, C_i)$、$i = 1$、$2$，从 B_1 到 B_2 的单向连接 σ_1、2 是满足以下条件的关系 $\sigma_{1,2} = \{(y_{1,i}, u_{2,j}) \mid y_{1,i} \in Y_1, u_{2,j} \in U_2\}$：

(1) 每一对 $(y_{1,\ i},\ u_{2,\ j})$ 中的输出和输入在类型和维度方面是兼容的；(2) 一个输入只能连接一个输出；(3) 对所有 $t \in \mathbb{R}^+$，$u_{2,\ j}(t) = y_{1,\ i}(t)$。

连接 $\sigma_{1,\ 2}$ 中的输出集合和输入集合分别记为 $\sigma_1^{\text{out}}{}_{,\ 2} \subseteq Y_1$ 和 $\sigma_1^{\text{in}}{}_{,\ 2} \subseteq U_2$。

因此，CBD 是由块及其连接构成的组合。

定义*2.4*（因果模块图（*CBD*））。CBD 是由若干块 $\mathcal{B}$ Σ 组成的元组 $D = (\mathcal{B}, \Sigma)$，其中块集合为 $\mathcal{B} = \{B_i\}_i$，并包含这些块之间的一组单向连接 $\Sigma = \{\sigma_{i,\ j} \mid B_i \in \mathcal{B},\ B_j \in \mathcal{B}\}$。这些连接必须满足：每个输入最多只能连接一个输出，但一个输出可以连接多个输入。

现在，我们定义三种基本组合算子的语义，这些算子可用于组合出更复杂的 CBD。以下符号表示是必需的：如果 $f\colon X \mapsto Y$ 是一个函数，其输出变量向量为 Y，且 $Y' \subseteq Y$，那么 $f|_{Y'}$表示

将函数 f 限制到输出的子集 Y' 上，即 $f|_{Y'} := \pi_{Y'} \circ f$，其中 $\pi_{Y'}$ 是从 Y 到 Y' 的投影算子。

两个块的并行组合（如图 1b 所示）只是将这两个块合并为一个块。

定义 *2.5*（并行组合）。给定两个块 $\mathrm{B}_i = (U_i, X_i \quad T_i\ F_i\ g_i \quad _i \quad _i \quad i = 1\ 2 \quad _1 \parallel \quad _2, Y_i, ,, \mathrm{Init}, \mathrm{C})$，$,_\circ$ 一个并行组合 B B 是一个块 B $= U\ X \quad T\ F\ g \quad , \quad U = U_1 \cup U_2\ X = (,, Y, ,, , \mathrm{Init}\ \mathrm{C})$，其中 $, , X_1 \cup X_2\ Y = Y_1 \cup Y_2\ T = T_1 \cup T_2\ F = F_1 \cup F_2\ g\ \boldsymbol{X}\ \boldsymbol{U} =$ $, , , (,)\ (g_1(X_1, U_1)\ g_2(X_2, U_2))$，$\mathrm{Init} = (\mathrm{Init}_1, \mathrm{Init}_2)$，而组合后的契约将在第 2.3 节中介绍。

两个块的串联组合将一个块的部分输出连接到另一个块的部分输入，如图 1c 所示。

定义 *2.6*（串行组合）。给定两个块 $\mathrm{B}_i = (U_i, X_i \quad T_i\ F_i\ g_i \quad _i \quad _i \quad i = 1\ 2 \quad \sigma_{1\,2} \quad _1 \quad , Y_i, ,, , \mathrm{Init}, \mathrm{C}), ,,$ 以及一个从 B 到 B_2 的连接 $,_\circ$ 令 $U_2^- = U_2 \setminus \sigma_{1,2}^{\mathrm{in}}$ 表示 B_2 中未通过连接 $\sigma_{1,2}$ 连接的输入，并令 $Y_1^\sigma = \sigma_{1\,2}^{\mathrm{out}},$ 表示与 B_2 相连的 B_1 的输出。串联组合 $\mathrm{B}_1; \mathrm{B}_2$ 是一个块 $\mathrm{B} = (U, X, Y\ T\ F \quad , , , g, \mathrm{Init}, \mathrm{C})$，其

- $U = U_1 \cup U_2^-$，$X = X_1 \cup X_2$，$Y = Y_1 \cup Y_2$，$T = T_1 \cup T_2$，
- $F = F_1 \cup F_2^\sigma$，其中 F_2^σ 由 F_2 中相同的状态转移函数组成，但仅限于由 $\sigma_{1,\ 2}$ 确定的连接输入。具体而言，F_2 中的每个 $f_{(\tau)\,2}(X_2, U_2)$ 都会变为 $f_{(\tau)\,2}(X_2, (U_2^-, g_1|_{Y_1^\sigma}(X_1, U_1))) = f_{(\tau)\,2}(X, U)$，属于 F_2^σ。
- $g(X, U) = (g_1(X_1, U_1)\ g_2(X_2, (U_2^-, g_1|_{Y_1^\sigma}(X_1, U_1))))$ 反映了从 B_1 到 B_2 的连接。它也可以表示为两步：$Y_1 = g_1(X_1, U_1)$ 和 $Y_2 = g_2(X_2, (U_2^-, \pi_{Y_1^\sigma}(Y_1)))$。
- $\mathrm{Init} = (\mathrm{Init}_1, \mathrm{Init}_2)$
- C 是组合契约，将在第 2.3 节介绍。

CBD 中一种重要且常用的组合是反馈组合，它通过有向环将块 B 的某些输出反馈到其自身的某些输入中，如图 1d 所示。反馈组合面临的一项挑战是代数环反馈。我们首先介绍直接馈通的概念。如果块 B 的输出函数 g 满足：输入 u_i 直接馈通 g 至输出 y_j，即 $u_i(t)$ 的值直接影响 $y_j(t)$ 的值，那么 B 就存在从 u_i 到 y_j 的直接馈通 。当反馈组合中输出与输入之间的连接形成一个由直接馈通构成的闭环时，就会出现代数环 ，即 u_i 直接馈通至 y_j，且 y_j 与 u_i 相连。换句话说，块的一个输出

在同一时间步内由自身驱动。尽管代数环可以表示为代数方程，并在 CBD 执行中求解，但用这种方式处理它存在固有困难。处理代数环的另一种方法是人为地在环中插入一个有状态模块（例如单位时间延迟或积分器），从而打破该环。虽然这可能会稍微改变原始 CBD 的语义，但实现起来很直接，因此在实践中经常采用，例如在 Simulink 中。为简单起见，我们假设所有反馈组合均不存在代数环。

假设 *2*。 C 中不存在代数环 BD.

未来的研究将探讨如何通过显式处理代数环来消除这一假设。

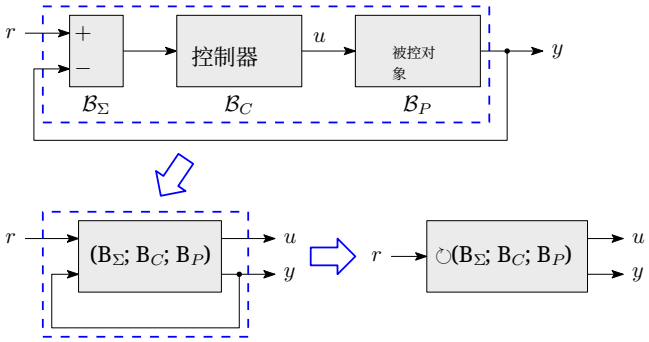


图 2：标准反馈控制回路（上方）可以使用串联组合规则和反馈组合规则展平为一个等效块（右下方）。

定义 *2.7*（反馈组合）。给定一个块 $\mathrm{B} = (U, X \quad T\ F \quad , Y, \quad , g,$ 初始，C），以及一个从 B 到其自身的连接 σ_{fb}。假设该组合不存在代数环。令 $U^- = U \setminus \sigma^{\mathrm{in}}{}_{\mathrm{fb}}$ 表示 B 中未通过连接 σ_{fb} 连接的输入，并令 $Y^\sigma = \sigma^{\mathrm{out}}{}_{\mathrm{fb}}$ 表示通过 σ_{fb} 连接的 B 的输出。不存在代数环的反馈组合 $\circlearrowleft$ B 是一个块 $\mathrm{B}_{\mathrm{fb}} = (U_{\mathrm{fb}}, X_{\mathrm{fb}}, Y_{\mathrm{fb}}, T_{\mathrm{fb}}, F_{\mathrm{fb}}, g_{\mathrm{fb}},$ 初始$_{\mathrm{fb}}$，C_{fb}），其中

- $U_{\mathrm{fb}} = U^-$，$X_{\mathrm{fb}} = X$，$Y_{\mathrm{fb}} = Y$，以及 $T_{\mathrm{fb}} = T$，
- $F_{\mathrm{fb}} = F^\sigma$ 由与 F 中相同的状态转移函数组成，但仅限于由 $\sigma_{i,\ j}$ 确定的相连输入。若不存在代数环，这些函数便是定义良好的。
- $g_{\mathrm{fb}}(X_{\mathrm{fb}}\ U_{\mathrm{fb}}) = g(X_{\mathrm{fb}}, (U_{\mathrm{fb}}\ g|_{Y^\sigma}(X_{\mathrm{fb}}\ U_{\mathrm{fb}})))$ 在没有代数环的情况下是定义良好的。存在代数环时，g_{fb} 会变成一个隐式方程，这在定义 2.2 中是允许的，并且可以在温和条件下通过不动点求解器求解。
- Initfb =Init
- C 是组合契约，将在第 2.3 节介绍。

上述组合具有若干有用的性质。

引理 2.8. 给定块 $\mathcal{B}_1, \mathcal{B}_2,$ 和 $\mathcal{B}_3, ,$ 以及它们之间的可接受连接，以下性质成立。
(1) $(\mathrm{B}_1; \mathrm{B}_2); \mathrm{B}_3 = \mathrm{B}_1; (\mathrm{B}_2; \mathrm{B}_3)$ (2) $\mathrm{B}_1 \parallel \mathrm{B}_2 = \mathrm{B}_2 \parallel \mathrm{B}_1$ (3) $(\mathrm{B}_1 \parallel \mathrm{B}_2) \parallel \mathrm{B}_3 = \mathrm{B}_1 \parallel (\mathrm{B}_2 \parallel \mathrm{B}_3)$.

这些性质直接源自 CBD 执行语义中信号串接和代换的结合律与交换律，并且是 CBD 形式体系中的标准性质 [28, 52]。利用这些基本组合及其性质，任何无代数环 CBD D $= (\mathcal{B}\ \Sigma)$ 都可以转换，或称为扁平化，为一个与原始 CBD 具有相同执行语义的等效块 B。图 2 展示了如何运用基本组合规则，将标准反馈控制 CBD 扁平化为一个等效块。我们首先对前馈路径应用串联组合规则。该路径由求和模块 B_Σ、控制器模块 B_C 和被控对象模块 B_P 组成，由此得到组合块 $(\mathrm{B}_\Sigma; \mathrm{B}_C; \mathrm{B}_P)$。然后，我们对该组合块应用反馈组合规则，将原始 CBD 扁平化为块 $\circlearrowleft(\mathrm{B}_\Sigma; \mathrm{B}_C; \mathrm{B}_P)$。该块以参考信号 r 为输入，以控制信号 u 和被控对象输出信号 y 为输出。

此外，CBD 中任意一组模块及其相关连接（这些模块和连接本身也构成一个 CBD），都可以等效地替换为其扁平化后的块，该块称为子系统。子系统可以包含其他子系统。因此，任何 CBD 都可以拆分为嵌套子系统。这样一来，也可以用近似块（例如基于机器学习的模型）替换由其扁平化块表示的子系统。

2.3 CBD 的契约

CBD 支持模块化系统构建，但要推理互连模块的正确性，就需要明确规定契约保证。本节介绍 CBD 的两种互补契约概念。假设—保证（A–G）契约为组合式正确性推理提供规范性、系统级的规格说明；基于残差的契约则为实际执行提供可微的定量证书，主要用于优化和学习。

CBD 的假设—保证（***A–G***）契约。A–G 契约通过组合式规格说明和验证，为推理系统正确性提供了一个形式化框架 [12], , 并已成功集成到接口自动机 [19] 和动力系统 [41]等框架中。基于这些思想，我们为 CBD 形式体系扩展了 A–G 契约语义，使块结构模型能够进行基于契约的推理。

定义 *2.9*（块的假设—保证契约）。块 B 的假设–保证（A–G）契约是一个二元组

$$C^{\mathrm{AG}} = (\mathcal{A}, \mathcal{G}),$$

其中 $\mathrm{A} \subseteq \mathrm{U}$ 表示可接受的输入行为（假设）集合，$\mathrm{G} \subseteq \mathrm{U} \times \mathrm{Y}$ 表示所保证的输入–输出行为集合。称块 B 满足其契约，如果

$$\forall U \in \mathcal{A}: \quad (U, \mathcal{B}(U)) \in \mathcal{G}.$$

假设—保证契约的复合反映了模块的结构复合，其形式化定义是：在强制假设与保证之间的兼容性条件的同时，对内部信号进行投影与存在量化。

对于集合 $\mathrm{S} \subseteq \mathrm{U} \times \mathrm{Y}$，令 $\pi_U(\mathrm{S})$ 与 $\pi_Y(\mathrm{S})$ 分别表示其到输入与输出上的投影。对于串联连接 $\sigma_{1,2}$，令 $U_2^- = U_2 \setminus \sigma_{1,2}^{\mathrm{in}}$ 与 $Y_1^\sigma = \sigma_{1\,2}^{\mathrm{out}}$,如定义 2.6 所述，并令 $\sigma_{1,2}(Y_1^\sigma)$ 表示从信号 Y_1^σ到连接信号 $\sigma_{1,2}^{\mathrm{in}}$的映射。

定义 *2.10*（串联连接的 *A–G* 兼容性）。设 $\mathrm{C}_1^{\mathrm{AG}} = (\mathrm{A}_1, \mathrm{G}_1)$ 和 $\mathrm{C}_2^{\mathrm{AG}} = (\mathrm{A}_2, \mathrm{G}_2)$ 是分别针对 B_1 和 B_2 的假设 — 保证契约。若每个与 $\mathrm{C}_1^{\mathrm{AG}}$一致的上游行为都会提供一个对 B_2, 可接受的连接信号，则串联连接 $\sigma_{1,\,2}$ 是兼容的，即，

$$\sigma_{1,2}\left(\pi_{Y_1^\sigma}(\{(U_1, Y_1) \in \mathcal{G}_1 \mid U_1 \in \mathcal{A}_1\})\right) \subseteq \pi_{\sigma_{1,2}^{\mathrm{in}}}(\mathcal{A}_2).$$

定义 *2.11*（*A–G* 契约组合）。设 $C_i^{\mathrm{AG}} = (\mathcal{A}_i, \mathcal{G}_i)$ 是针对模块 $\mathcal{B}_i$的假设—保证契约。
并行：并行组合 $\mathrm{B}_1 \parallel \mathrm{B}_2$ 的契约为

$$C_{\parallel}^{\mathrm{AG}} = (\mathcal{A}_1 \times \mathcal{A}_2,\ \mathcal{G}_1 \times \mathcal{G}_2),$$

其中 $(\mathrm{A}_1 \times \mathrm{A}_2) \subseteq (\mathrm{U}_1 \times \mathrm{U}_2)$，且 $(\mathrm{G}_1 \times \mathrm{G}_2) \subseteq (\mathrm{U}_1 \times \mathrm{U}_2) \times (\mathrm{Y}_1 \times \mathrm{Y}_2)$。

串行：假设$\sigma_{1,\,2}$ 在定义 2.10 的意义下是兼容的。串联组合 B_1; B_2 的契约为

$$C_{;}^{\mathrm{AG}} = (\mathcal{A}_{;}, \mathcal{G}_{;}),$$

其中，$\mathrm{A}_{;} \subseteq \mathrm{U}_1 \times \mathrm{U}_2^-$由互连可接受的外部输入组成，而 $\mathrm{G}_{;} \subseteq (\mathrm{U}_1 \times \mathrm{U}_2^-) \times (\mathrm{Y}_1 \times \mathrm{Y}_2)$ 是通过对内部连接信号进行存在量化而得到的外部输入/输出行为集合。具体而言，$(U_1,\ U_2^-,\ Y_1,\ Y_2) \in \mathrm{G}_{;}$，当且仅当存在一个连接信号 $v \in \pi_{\sigma_{1,\,2}^{\mathrm{in}}}(\mathrm{A}_2)$，使得 $(U_1,\ Y_1) \in \mathrm{G}_1$ 且 $((U_2^-,\ v)\ Y_2) \in \mathrm{G}_2$,，其中 v 与 $\pi_{Y_1^\sigma}(Y_1)$ 相对应。

反馈：假设反馈互连 σ_{fb} 是适定且兼容的（类似于串行兼容性）。那么，按照 CBD 的反馈执行语义，将反馈组合展开为等价的串行组合，即可处理反馈契约；因此，契约组合可归结为反复应用串联组合规则。此构造等价于对内部反馈信号进行存在量化，并投影到其余外部变量上。

定义 *2.12*（系统级 *A–G* 满足性）。CBD $\mathrm{D} = (\mathcal{B}\ \Sigma)$ 满足其假设—保证契约，当且仅当每个块 B_i 在全称语义下满足契约 $\mathrm{C}_i^{\mathrm{AG}} = (\mathrm{A}_i,\ \mathrm{G}_i)$，且 Σ 中的所有互连均兼容（即在适当的投影下，保证蕴含下游假设）。

CBD 的基于残差的契约。为实现可验证的优化与学习，我们引入基于残差的契约，作为沿实际执行评估的定量证书。

定义 *2.13*（模块的基于残差的契约）。块 B 的基于残差的契约定义为一个映射

$$C^{\mathrm{res}}: \mathcal{X} \times \mathcal{U} \times \mathcal{Y} \times \mathbb{R}^{n_\theta} \to \mathbb{R}^m,$$

其中，$\theta \in \mathbb{R}^{n_\theta}$用于参数化该契约。若实际执行 (X, U, Y) 满足基于残差的契约，则

$$C^{\mathrm{res}}(X, U, Y; \theta) \le 0$$

逐元素成立。残差满足性构成一项充分证书，表明对于给定执行，实际输入–输出行为 (U, Y) 满足相应的保证。

基于残差的契约按照 CBD 执行语义在结构层面进行组合：聚合局部残差，并将相连的输出代入输入。

定义 *2.14*（残差契约组合）。考虑以下基于残差的契约 C_i^{res} $(X_i, U_i, Y_i;\ \theta_i) \le 0$，对应于模块 $\mathcal{B}_i$。
并行：并行组合 $\mathrm{B}_1 \parallel \mathrm{B}_2$ 的残差契约由串接给出

$$C_{\parallel}^{\mathrm{res}}(X, U, Y; \theta) = \begin{bmatrix} C_1^{\mathrm{res}}(X_1, U_1, Y_1; \theta_1) \\ C_2^{\mathrm{res}}(X_2, U_2, Y_2; \theta_2) \end{bmatrix} \le 0,$$

其中 $X = (X_1, X_2)$、$U = (U_1, U_2)$、$Y = (Y_1, Y_2)$ 和 $\theta = (\theta_1, \theta_2)$。串行：对于串行互连 $\sigma_{1,2}$，定义 $U_2^- = U_2 \setminus \sigma_{1,2}^{\mathrm{in}}$ 和 $Y_1^\sigma = \sigma_{1\,2}^{\mathrm{out}}$,。串联组合 B_1; B_2 具有残差契约

$$C_{;}^{\mathrm{res}}(X, U, Y; \theta) = \begin{bmatrix} C_1^{\mathrm{res}}(X_1, U_1, Y_1; \theta_1) \\ C_2^{\mathrm{res}}(X_2,\ (U_2^-,\ Y_1^\sigma),\ Y_2; \theta_2) \end{bmatrix} \le 0.$$

反馈：对于反馈互连 σ_{fb}，令 $U^- = U \setminus \sigma^{\text{in}}{}_{\text{fb}}$ 和 $Y^\sigma = \sigma^{\text{out}}{}_{\text{fb}}$。反馈组合 $\circlearrowleft$B 具有残差契约

$$C^{\text{res}}_{\circlearrowleft}(X, U^-, Y; \theta) = C^{\text{res}}\big(X,\ (U^-,\ Y^\sigma),\ Y; \theta\big) \le 0,$$

其中，Y^σ 表示代入相应反馈输入的反馈输出。

定义 *2.15*（系统级残差满足）。CBD D $=(\mathcal{B}\ \Sigma)$ 在某一实际执行中满足其基于残差的契约，当且仅当对于每个块 B_i，均有 $\text{C}_i^{\text{res}}(X_i, U_i, Y_i;\ \theta_i)\ \le 0$，且所有互连均遵守允许输入域。

注释 *2*（作为精确假设—保证契约的残差契约）。在特殊情况下，如果基于残差的契约编码了保证集 G 的必要且充分条件，并且通过构造对所有允许执行强制实施，那么它可以被精确地解释为假设—保证契约。在此情形下，残差仅取决于 CBD 的参数（或结构属性），而不取决于具体轨迹，因此产生一个全称量化的保证。否则，基于残差的契约依赖于轨迹，只能为实际执行提供充分的证书。我们在第 3 节的示例1和示例3中展示了具有代表性的结构属性案例。更广泛地刻画并利用这种等价性是未来的研究方向。

2.4 可微因果框图

自动微分（AD）依赖于雅可比—向量积（JVP）和向量—雅可比积（VJP）。对于具有雅可比矩阵 $J_F = \frac{\partial F}{\partial x}$ 的可微映射 $F\colon \mathbb{R}^n \to \mathbb{R}^m$，它们定义如下

$$\text{JVP}(v) = J_F v, \qquad \text{VJP}(\bar{v}) = J_F^\top \bar{v}. \tag{1}$$

这里，v 表示输入空间中的切向（扰动）方向，$\bar{v}$ 表示输出空间中的伴随量（余向量）。JVP 在前向模式（灵敏度传播）下计算方向导数，而 VJP 在反向模式（梯度反向传播）下计算伴随灵敏度。我们始终假设所有映射都定义在相应欧几里得空间的开集上；“几乎处处”表示几乎处处可微性。

可微块。现在，我们定义带契约的块（定义 2.2）可微所需满足的条件。

定义 *2.16*（可微块）。考虑一个块 B $=(U,$ X T F $, Y, , , g,$ 初始化$, \text{C})$，其中 C 是一组（可能为空的）契约注释。若定义 2.2 成立，并且满足以下条件，则称 B 在 U $\times$X 上是（分段）可微的：

- 连续值状态变量。X 是连续值的。
- 状态转移函数 F。每个 $f_{(\tau)}\colon$ X $\times$U $\to$ X$_{(\tau)}$ 都是局部利普希茨函数，因此根据拉德马赫定理，几乎处处可微，且在紧子集上具有有界雅可比矩阵。
- 输出映射 g。映射 $g\colon$ X $\times$U $\to$Y 是局部利普希茨函数，因此几乎处处可微；对于孤立的非光滑点（例如，次梯度或平滑化），则采用已声明的策略。
- 契约C。假设—保证契约施加语义约束，不参与微分。基于残差的契约由局部利普希茨残差 $\text{C}_{\text{res}J}(X, U, Y;\ \theta)$给出，这些残差几乎处处可微，并通过明确的自动微分策略（例如，次梯度或平滑化）在不可微点提供广义梯度。

在这些条件下，每个可微块都定义了一个兼容自动微分的基本运算，支持正向模式和反向模式的灵敏度传播。许多工程和机器学习模块都是分段 C^1。根据状态转移函数和输出映射的类型，可以考虑若干种可微组件。下面列举最重要的模式。

i) 离散时间动力学。对于带有 $f^{(\tau)}$ 和$\tau > 0$的差分方程，通过在有限时域内展开离散转移来计算雅可比作用。在这里，VJP 等价于时间反向传播 （BPTT）算法 [49, 39]。

ii) 连续时间动力学。对于带有 $f^{(\tau)}$和 $\tau = 0$的微分方程，雅可比作用通过所选的灵敏度方法计算：要么展开数值积分并应用时间反向传播，要么使用连续时间伴随灵敏度方法 [22, 17]。如果 $f_{(\tau)}$ 是神经网络，则这等价于神经常微分方程（NODEs）方法 [18]。

iii) 机器学习模型。对于静态数据驱动组件，块输出 $Y = g_\Theta($ $X, U)$取决于可学习参数 Θ。当模型由可微原语（例如线性层、激活层和归一化层）构成时，映射 g_Θ在(X, U, Θ)中几乎处处可微，并且自动微分框架 [9, 15, 47]原生提供 JVP/VJP运算。

iv) 不动点求解器。如果输出块在内部求解形如 $Y^\star = \Phi($ $Y^\star,\ U)$的 隐式方程，则假设 $(I - \partial_Y \Phi)$ 非奇异（例如，满足压缩映射条件）。根据隐函数定理，平衡映射 $Y^\star(U)$ 的局部灵敏度满足 $(I - \partial_Y \Phi)\ \delta Y^\star = \partial_U \Phi\, \delta U$，其解定义了方向导数 $\delta Y^\star = J_{Y^\star,\ U}\, \delta U$，其中 $J_{Y^\star,\ U} = (I - \partial_Y \Phi)^{-1} \partial_U \Phi$. 该表达式直接对应于平衡映射的雅可比-向量积，可通过运行时使用的同一线性求解来计算。这使得各种不动点或平衡求解器都能作为可微块纳入 ∂CBD 形式体系 [40, 32, 8]。

v) 优化层。当块输出 $Y^\star(U)$ 被隐式定义为某个参数化优化问题的解时

$$Y^\star(U) = \arg\min_Y\ \ell(Y, U) \quad \text{s.t.} \quad h(Y, U) = 0,$$

解由 KKT 条件 $K(Y^\star,\ \lambda^\star,\ U)$ $=$ 0 $Y^\star\ \lambda^\star\ U$ $h\ Y^\star\ U$ L ，刻画，其中 $K = [\nabla_Y L(,\ );\ (,\ )]$ 表示拉格朗日函数。在满足 LICQ 和强正则性的条件下，隐函数定理给出灵敏度关系 $\partial_{(Y,\lambda)} K \begin{bmatrix} \delta Y^\star \\ \delta \lambda^\star \end{bmatrix} + \partial_U K\, \delta U = 0$，该关系通过与运行时相同的线性求解来定义雅可比作用，从而在 ∂CBD 形式体系中实现可微优化层 [5, 3, 45, 34, 14, 13, 2]。

vi) 整数与逻辑端口。对于离散或逻辑信号，必须声明可微性策略，例如停止梯度、直通估计器（STE）或光滑松弛方案，以确保几乎处处定义良好的雅可比-向量积/向量-雅可比积行为 [11, 7, 46]。

可微因果框图（***∂CBD***）。现在，我们已经具备定义可微 CBD 所需的所有组件。

定义*2.17*（可微因果框图）。CBD $\mathcal{D} = (\mathcal{B}\ \Sigma)$，参见定义 2.4，在 U上可微，当且仅当：

(1) 每个块 $\text{B}_i \in \mathcal{B}$ 在其定义域上均可微，符合定义 2.16。

(2) 互连 Σ 类型一致且允许反馈。

(3) 通过执行 CBD组合（定义2.4）得到的 CBD输入–输出映射 D : U → Y 几乎处处可微，并支持通过局部JVP/VJP组合进行自动微分。

归约为可微有向无环图（*DAG*）：自动微分引擎在 DAG 上运行。对于给定的执行时域，任何可微 CBD 都可以通过以下步骤归约为这样的 DAG:

(1) 速率提升：将所有时间基准 T_i 提升到一个公共超周期，使求值同步进行。(2) 并行组合（分支扩展）：将 $\mathrm{B}_1 \parallel \mathrm{B}_2$ 拆分为相互独立的子图，并串联端口。(3) 反馈组合：将每个反馈回路展开有限个步。(4) 事件扁平化（混合）：使用声明的可微性策略（例如松弛门控）对模式切换进行编码。(5) 拓扑排序：对所得无环图进行排序，以获得与正向和反向自动微分兼容的执行调度 S（求值轨迹）。

这样便得到一个经过速率提升、分支扩展和反馈展开的 DAG，其 CBD 复合映射 D 是可微的，并已准备好用于自动微分。令 $J_{\mathcal{D}} = \partial F/\partial U$ 表示 D 的总雅可比矩阵。自动微分不会计算完整的雅可比矩阵 $J_{\mathcal{D}}$；相反，它会沿着 S 应用局部雅可比作用。实际流程如下:

(1) 构建可微 CBD，并执行速率提升、分支扩展、循环展开和事件扁平化；(2) 执行排序后的 DAG，记录求值轨迹 S；(3) 沿 S 执行自动微分扫描（使用雅可比-向量积进行敏感性分析，使用向量-雅可比积进行学习）。

因此，这一形式体系允许将基于物理的模块（如常微分方程积分器、优化层）与数据驱动模块（如神经网络）集成到统一、可微且可认证 的计算图中，适用于在现代自动微分框架上实现。

2.5 基于可微 CBD 的可验证优化与学习

基于所提出的 ∂CBD 框架的可组合性、可微性和可验证性，我们现在提出一个通用优化问题，在信息物理系统中统一学习、建模、优化和控制。该问题利用 ∂CBD 的可微执行语义实现基于梯度的学习与优化，同时将基于残差的契约纳入可微约束，以确保正确性、安全性和性能要求。

一般 ∂***CBD*** 优化问题。 我们考虑有限时域 $\mathrm{T} = [0, t_f]$（连续时间）或 $\mathrm{T} = \{0, \dots, N\}$（离散时间）上的参数化图 $\mathrm{D}(\theta) = (\mathcal{B}(\theta), \Sigma)$。令 (X, U, Y) 表示根据第 2.1 节执行 $\mathrm{D}(\theta)$ 生成的轨迹。CBD 约束优化问题可表述为:

$$\min_{\theta} \quad \mathcal{J}\,(Y(t), \boldsymbol{U}(t), \boldsymbol{X}(t); \theta) \;+\; w_{\mathrm{reg}}\,\mathcal{R}(\theta) \tag{2a}$$

$$\text{s.t.} \quad (\boldsymbol{X}(t)\ Y(t)) \;\in \text{执行}\ (\mathrm{D}(\theta)\ \boldsymbol{U}(t))\,, \quad \text{其中} \tag{2b}$$

$$(\text{dynamics}) \quad \begin{cases} \dot{X}^{(0)}(t) = f^{(0)}(\boldsymbol{X}(t), \boldsymbol{U}(t); \theta)\,, & t \in \mathcal{T}, \\ X_{k+1}^{(\tau)} = f^{(\tau)}(X_k, U_k; \theta)\,, & \tau > 0,\ k \in \mathcal{T}, \end{cases} \tag{2c}$$

$$\text{（代数输出）} \quad \boldsymbol{Y}(t) \;= g(\boldsymbol{X}(t)\ \boldsymbol{U}(t); \theta)\,, \tag{2d}$$

$$\text{（契约）}\ \mathrm{C}^{\mathrm{res}}(\boldsymbol{X}(t)\ \boldsymbol{U}(t)\ \boldsymbol{Y}(t); \theta) \leq\ 0, \qquad \forall t \in \mathrm{T}。 \tag{2e}$$

这里，J 表示任务性能（例如跟踪、成本、能量、鲁棒性），R 是正则项，而 Exec(·) 表示 CBD 在串行/并行/反馈组合下的前向执行语义。第 2.3 节中的残差契约实例化为可微约束 $\mathrm{C}^{\mathrm{res}}(\theta) \leq 0$，其梯度可融入训练或认证目标。因此，可以在同一自动微分流水线中联合优化性能与安全性（例如无源性、李雅普诺夫稳定性、增益界）。组合模块时，契约残差会沿 DAG 求和（或取最大值），从而在保留梯度访问能力的同时实现模块化验证。

范围与通用性。 优化问题 (2) 定义了一个通用且可组合的框架，涵盖了广泛的建模、学习、优化和控制任务。根据模块、契约和目标函数 J 的选择，它可以表示:

- 约束优化，涵盖广泛的优化问题，包括以 CBD 块组合形式表达的凸、非线性、确定性、随机或双层形式问题。
- 约束系统辨识，其中 θ 用于参数化未知动力学或代理模型，而契约则用于编码物理先验或稳定性约束；
- 最优控制与策略优化，其中 θ 用于参数化反馈策略或最优控制动作序列。
- 控制协同设计，在同一个 CBD 中联合优化系统模型和控制策略参数。
- 约束机器学习与科学机器学习，其中 CBD 编码通用或物理信息机器学习模型，并提供安全性或可行性保证。

因此，该表述允许对异构子系统进行联合优化，为复杂信息物理系统的端到端、可验证设计提供统一基础。

*CBD*约束优化问题的可微性。在满足定义2.17的可微性条件时，式（2）中的所有约束都公开局部雅可比-向量积/向量-雅可比积运算，从而支持基于梯度的优化。每个块和契约都会贡献局部雅可比作用，使整个图示可直接进行自动微分，并支持跨异构子系统的联合端到端优化。目标函数和契约残差的梯度通过对 CBD 的速率提升并展开循环后的 DAG 表示进行反向模式自动微分获得（第2.4节）。局部雅可比-向量积/向量-雅可比积运算能够通过串行、并行和反馈连接无缝组合，从而支持跨多个领域、时间尺度和决策层的可扩展、内存高效的反向传播与验证。

基于梯度的优化与学习： ***∂CBD***。为求解可验证优化问题（2），我们采用一种重构方法，使自动微分流水线能够支持端到端训练与验证。契约约束和不等式约束被放宽为光滑惩罚项或障碍项，从而得到无约束目标函数

$$\min_{\theta} \mathcal{L}(\theta) = \mathcal{J}(Y,U,X;\theta) + w_{\text{reg}}\,\mathcal{R}(\theta) + w_C\,\Phi_\beta\big(C(X,U,Y;\theta)\big), \quad (3)$$

其中， Φ_β 表示惩罚函数或障碍函数，而 $w_C > 0$ 则用于权衡任务性能与合约满足度。对（3）的梯度通过在 ∂CBD 的计算图上执行反向模式自动微分获得，从而能够利用基于梯度的求解器进行可扩展优化。

注释 *3*（严格可行性）。可行性恢复层（例如可微优化层[2, 3, 22, 46]和不动点求解器）能够以严格保证的方式确保合约满足度。得益于 ∂CBD 的组合结构，这些机制既可在块级别局部应用，也可在系统级别全局应用。

注释 *4*（替代优化后端）。虽然我们使用惩罚—障碍形式（3）进行兼容自动微分的训练，但 ∂CBD 框架并不依赖特定的优化后端。增广拉格朗日法、算子分裂或经典非线性规划求解器（例如 IPOPT）均可集成，并利用自动微分提供的梯度，例如在 CasADi [6] 或 JAX [14]中。

备注 *5*（可扩展性）。 ∂CBD 的可扩展性取决于三个基本相互独立的因素：CBD 计算图的执行与求导、所选形式化方法下的 A–G验证，以及对 (3) 进行数值优化。这三者均受益于 CBD 的组合结构，从而支持模块化分解和针对特定后端的可扩展性。系统研究可扩展性，涵盖理论复杂度界以及不同验证和优化后端的实证表现，是未来一项重要的研究方向。

3 示例

我们展示三个案例研究，说明 ∂CBD 如何将带有契约的组合式 CBD 模型编译为自动微分就绪图，以用于基于梯度的设计[1]。这些示例涵盖：(i) 稳定性构造下的增益调优，(ii) 针对非线性系统的轨迹级李雅普诺夫证书学习，以及 (iii) 构造稳定性的深度Koopman辨识。所有示例均遵循相同的结构：(i) 通过组合模块指定 CBD，(ii) 指定 A–G契约或残差契约，(iii) 通过对编译后的 ∂CBD 图进行自动微分，求解受契约约束的优化问题，以及 (iv) 报告任务性能和合约满足度。我们使用 PyTorch [37] 和 Neuromancer [24]实现所有示例。

3.1 示例 1：结合输入到状态稳定性证书的标量反馈增益调节

这个介绍性示例旨在演示如何利用我们的 ∂CBD 框架，对标量反馈增益进行稳定性认证调节。这里我们考虑一个简单的闭环系统，如图 2 所示，该系统使用了三个模块。

[1]匿名代码可在 https://zenodo.org/records/18226276 获取。

∂CBD 表示。 我们假设被控对象模型带有有界加性扰动 $|w_k| \le w_{\max}$，其表示为有状态模块 B_P，输入为 $U_P = \{u,\ w\}$，状态为 $X_P = \{x\}$，输出为 $Y_P = \{y\}$，采样周期为 $T_P = \{\tau\}$，其状态转移映射和输出映射为

$$x_{k+1} = f_P^{(\tau)}(x_k, u_k, w_k; a, b) = a\,x_k + b\,u_k + w_k, \quad (4a)$$
$$y_k = g_P(x_k, u_k, w_k) = x_k, \quad (4b)$$

由 $a \in \mathbb{R}, b > 0$ 参数化。静态比例控制器表示为无状态控制器模块 B_C，其输入为 $U_C = \{e\}$，输出为 $Y_C = \{u\}$，代数输出映射为 $u_k = g_C(e_k;\ \kappa) = \kappa\, e_k$，由可调增益 $\kappa \in \mathbb{R}$ 参数化。跟踪误差表示为无状态求和模块 B_Σ，其输入为 $U_\Sigma = \{r,\ y\}$，输出为 $Y_\Sigma = \{e\}$，输出映射为 $e_k = g_\Sigma(r_k,\ y_k) = r_k - y_k$.

对 B_Σ、B_C 和 B_P 进行串联组合，然后将 y 反馈至 B_Σ 以闭合回路，可得到闭环模块

$$\mathcal{B}_{\text{cl}} = \circlearrowleft\big(\mathcal{B}_\Sigma\ ;\ \mathcal{B}_C\ ;\ \mathcal{B}_P\big),$$

其由 $y_{k+1} = (a - b\kappa)\ y_k + w_k$. 控制（用于调节 $r_k \equiv 0$）

稳定性契约。 在标量情形下，线性系统的指数稳定性等价于 $|a - b\kappa| < 1$。附加到 B 的残差契约

$$C^{\text{res}}_{\text{stab}}(\kappa, a, b) \ = \ \big|a - b\kappa\big| - (1 - \epsilon), \qquad \epsilon \in (0, 1),$$

对这一性质进行编码。当且仅当 $\mathrm{C}^{\text{res}}{}_{\text{stab}}(\kappa, a, b) \ \le\ 0$，即 $|a - b\kappa| \le 1 - \epsilon$ 时，参数为 (κ, a, b) 的稳定性契约才成立；这为闭环 CBD 强制施加了严格的稳定裕度。在有界扰动 $|w_k| \le w_{\max}$ 下，这保证了输入到状态稳定性（ISS），并使系统收敛至原点的扰动相关邻域。

契约引导的增益调节。 设 $\mathrm{T} = \{0, \ldots, N\}$ 为有限时域。在 T 上执行参数为 κ 的闭环模块 B_{cl}，得到轨迹 $(X, U, Y) = \mathrm{Exec}(\mathrm{B}_{\text{cl}}(\kappa) W)$，对应扰动实现 $W = \{w_k\}_{k \in \mathcal{T}}$，其中 $Y = \{y_k\}_{k \in \mathcal{T}}$ 且 $U = \{u_k\}_{k \in \mathcal{T}}$。我们通过求解 CBD 约束优化来调节标量增益 κ

$$\min_{\kappa \in \mathbb{R}} \quad \mathcal{J}(\kappa) := \sum_{k=0}^{N} \big(\|y_k\|^2 + \lambda_u \|u_k\|^2\big), \quad (5a)$$
$$\text{s.t.} \quad (U, Y) \in \mathrm{Exec}\big(\mathcal{B}_{\text{cl}}(\kappa), W\big), \quad (5b)$$
$$C^{\text{res}}_{\text{stab}}(\kappa, a, b) \ \le\ 0. \quad (5c)$$

这是一般形式的 ∂CBD 可验证优化问题 (2) 的一个标量实例，其中参数为 $\theta = \kappa$，且只有一个稳定性契约。在此示例中，我们不采用惩罚屏障法 (2)，而是通过投影梯度步来强制满足稳定性契约：每次更新后，将 κ 限制在可容许区间 $\{|a - b\kappa| \le 1 - \epsilon\}$ 内。因此，此示例展示了注释 2 中所述的特殊情况：残差 $\mathrm{C}^{\text{res}}{}_{\text{稳定}}(\kappa,\ a,\ b) \le 0$ 编码了闭环标量系统指数稳定的充要条件，并通过投影到可行参数集上，在设计过程中强制满足该条件。因此，在全称量化下，残差满足性等价于满足相应的 A–G 稳定性契约。

仿真结果。 为了说明契约引导调参的行为，我们在有限时域内仿真系统（4），其中被控对象的初始参数不稳定 $a = 1.02$, $b = 1.0$，并受到有界扰动序列 $|w_k| \le 0.1$ 的影响。图3展示了输出轨迹：使用未经调优的增益 κ_0时，输出 y_k 会出现大幅

波动；而使用经过调优的增益 $\kappa^\star$时，轨迹则始终保持在稳定性契约所蕴含的解析ISS界 $\pm w_{\max}$ 以内。图4展示了增益 κ 及相应契约残差 C_{stab}的演变：增益通过梯度步迭代更新，随后投影到可容许区间 $\{|a-b\kappa| \le 1-\epsilon\}$，且残差始终严格为非正值，这证明训练期间所有迭代值均满足稳定性契约。

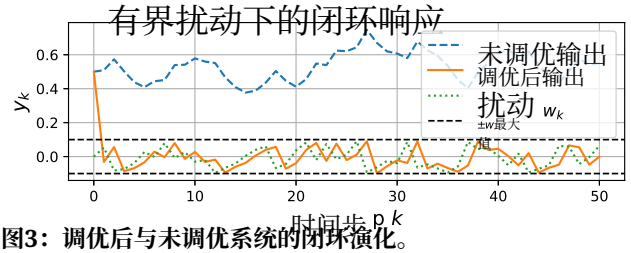


图3：调优后与未调优系统的闭环演化。

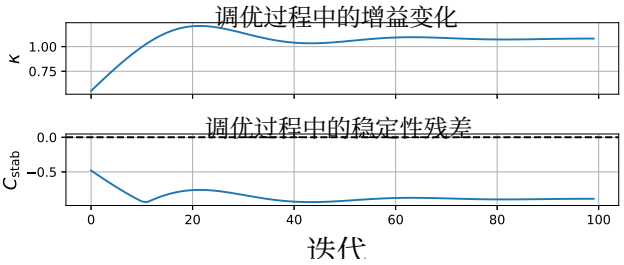


图 4：增益和契约残差的训练演变。

3.2 示例 2：联合学习神经策略与李雅普诺夫证书

第二个示例展示了如何使用 ∂CBD 框架联合学习神经反馈策略和神经李雅普诺夫函数，为非线性闭环系统提供轨迹级稳定性证书。在此示例中，我们通过复现 [25, 23, 36] 所述的可微预测控制（DPC）方法，展示所提出的 ∂CBD 框架的通用性。

*∂CBD 表示。*我们采用标准的受控范德波尔振荡器作为被控对象动力学

$$\dot{x}_1(t) = x_2(t), \quad \dot{x}_2(t) = \mu\big(1 - x_1(t)^2\big)x_2(t) - x_1(t) + U(t), \tag{6}$$

其状态为 $x = [x_1, x_2]^\top \in \mathbb{R}^2$，输入为 $u \in \mathbb{R}$，参数为 $\mu > 0$。在 ∂CBD 形式体系中，被控对象表示为一个有状态块 B_P，其输入为 $U_P = \{u\}$，状态为 $X_P = \{x\}$，输出为 $Y_P = \{y\}$，采样周期为 $T_P = \{\tau\}$，并具有离散时间转移映射和输出映射

$$x_{k+1} = f_P^{(\tau)}(x_k, u_k; \mu), \tag{7a}$$

$$y_k = g_P(x_k, u_k) = x_k, \tag{7b}$$

其中 $f_{(\tau)\,P}$ 是对式（6）在一个采样周期内进行数值积分的结果（此处采用 RK4积分器实现）。

反馈策略表示为无状态控制器模块 B_π，其输入为 $U_\pi = \{x,\ r\}$，输出为 $Y_\pi = \{u\}$，代数映射为

$$u_k = g_\pi(x_k, r_k; \Theta) = \pi_\Theta(x_k, r_k), \tag{8}$$

其中 π_Θ 是一个参数为 Θ 的神经网络，以当前状态 x_k 和参考值 r_k 为输入，并通过最终投影层生成有界控制信号 $u_k \in [u_{\min}, u_{\max}]$。

为验证稳定性，我们引入一个李雅普诺夫模块 B_V，其输入为 $U_V = \{x\}$，输出为 $Y_V = \{V\}$，用于实现标量李雅普诺夫候选函数 V_ϕ：$\mathbb{R}^2 \to \mathbb{R}_+$ 定义代数输出 $g_V(x_k;\ \phi) = V_\phi(x_k)$。我们的实现遵循 [33],，其中 V_ϕ 由一个包裹在正定层中的输入凸神经网络（ICNN）实现，从而确保 $V_\phi(0) = 0$ 且对于 $x \neq 0$，$V_\phi(x) > 0$。

将 B_π、B_P 和 B_V 串联组合，再将测量状态反馈至策略，即可得到闭环CBD

$$\mathcal{B}_{\text{cl}}^{\text{vdp}} = \circlearrowleft\big(\mathcal{B}_\pi\ ;\ \mathcal{B}_P\ ;\ \mathcal{B}_V\big), \tag{9}$$

对于调节任务 $r_k \equiv 0$，它生成具有状态 $X = \{x_k\}$、控制 $U = \{u_k\}$ 和李雅普诺夫值 $V = \{V_\phi(x_k)\}$ 的轨迹。

李雅普诺夫稳定性契约。 对于无扰动闭环系统，可以通过满足离散时间李雅普诺夫不等式的李雅普诺夫函数 V_ϕ 来证明原点的渐近稳定性；这里我们将此条件作为残差证书施加

$$V_\phi(x_{k+1}) - V_\phi(x_k) \ \le\ -\varepsilon, \qquad \varepsilon > 0, \tag{10}$$

于式（7）的有限时域轨迹上。我们将其编码为附加到块 B_V上的契约，并定义李雅普诺夫契约残差

$$C_{\text{lyap}}^{\text{res}}(X;\phi) \ =\ \max_{k\in\mathcal{T}} \Big(V_\phi(x_{k+1}) - V_\phi(x_k) + \varepsilon\Big), \tag{11}$$

对于有限时域 $\mathrm{T} = \{0,\ \ldots\ldots,\ N\}$。当且仅当 $\mathrm{C}^{\text{res}}{}_{\text{李雅普诺夫}}(X;\ \phi) \le 0$，时，该契约才成立，这确保了式（10）在每一步都成立。该残差契约提供的是充分的轨迹级证书，而非全称量化的李雅普诺夫保证。

契约引导的策略与证书联合学习。 令 $\theta = (\Theta,\ \phi)$ 汇集策略参数与李雅普诺夫参数。从初始条件 X_0 出发，在时域 T 内执行闭环模块 $\mathrm{B}^{\text{vdp}}{}_{\text{cl}}(\theta)$，得到轨迹 $(X,\ U,\ V) = \mathrm{Exec}(\mathrm{B}^{\text{vdp}}{}_{\text{cl}}(\theta)\ X_0)$。我们通过求解 CBD 约束优化∂来联合训练策略网络与李雅普诺夫网络

$$\min_{\theta} \quad \mathcal{J}(\theta) := \sum_{k\in\mathcal{T}} \|x_k\|^2, \tag{12a}$$

$$\text{s.t.} \quad (X,U,V) \in \mathrm{Exec}\big(\mathcal{B}_{\text{cl}}^{\text{vdp}}(\theta), X_0\big), \tag{12b}$$

$$C_{\text{lyap}}(X;\phi) \ \le\ 0, \tag{12c}$$

$$u_{\min} \le u_k \le u_{\max}, \quad \forall k \in \mathcal{T}, \tag{12d}$$

其中 $(u_{\min}, u_{\max})$ 表示输入边界。我们采用李雅普诺夫契约的惩罚形式求解问题 (12)，同时对输入约束使用投影。通过对整个 CBD （包括数值积分器、控制器和李雅普诺夫模块）进行自动微分，使用基于梯度的更新来最小化由此得到的损失函数。

仿真结果。按照 DPC方法 [25, 36],，我们使用批量随机初始条件 x_0 训练策略–李雅普诺夫联合对，这些初始条件从原点附近的一个紧集采样，并采用固定时域 N。训练完成后，我们在未见过的初始条件下评估学习得到的闭环 CBD。图5展示了由此得到的闭环轨迹：在学习得到的神经控制器作用下，两个状态 x_k 均收敛至原点，控制输入 u_k 保持

在规定的界限内，且李雅普诺夫契约残差$\Delta V_k := V_\phi(x_{k+1}) - V_\phi(x_k)$始终严格为负，表明沿轨迹满足契约。

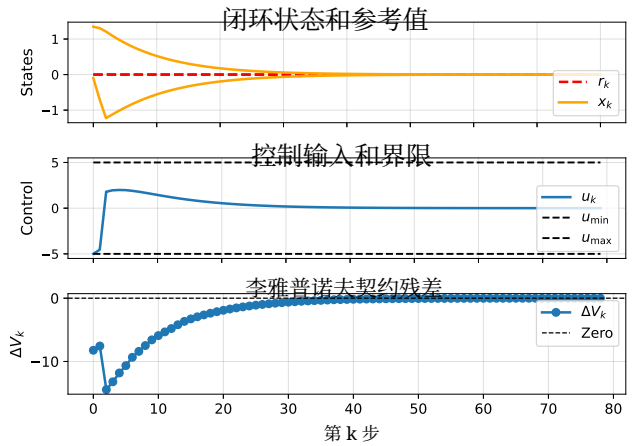


图5：带神经网络控制器和李雅普诺夫契约的范德波尔系统的闭环演化。

3.3 示例 3：带稳定性证书的深度 Koopman 算子学习

本示例展示了 *∂*CBD 形式如何支持契约引导的深度 Koopman 算子（DeepKO）学习，44] [35, 26, 50, 并对潜在动力学的稳定性提供明确保证。

∂CBD 表示。 我们考虑一个由 m 测量轨迹 $\{\hat{y}_k^i\}_{k=0}^N, i = 1,\dots,m$ 构成的数据集，其中 $\hat{y}_k^i \in \mathbb{R}^{n_y}$ 表示时间索引 k 处的观测量。DeepKO 由三个块表示

$$\mathcal{B}_{\text{enc}} : \hat{y}_k \mapsto x_k, \qquad \mathcal{B}_K : x_k \mapsto x_{k+1}, \qquad \mathcal{B}_{\text{dec}} : x_{k+1} \mapsto y_{k+1},$$

将观测量 $\hat{y}_k$ 映射到潜在坐标 $x_k \in \mathbb{R}^{n_x}$，通过 K 使其线性演化，并解码预测结果 y_{k+1}。

完整的 *∂*CBD 是以下串联组合

$$\mathcal{B}_{\text{koop}}(\theta) = (\mathcal{B}_{\text{enc}}(\theta_1)\,;\,\mathcal{B}_K(\theta_2)\,;\,\mathcal{B}_{\text{dec}}(\theta_3)), \qquad \theta = (\theta_1, \theta_2, \theta_3).$$

从初始数据 $\hat{y}_0^i$ 执行该图，生成潜变量轨迹和输出轨迹

$$(X^i, Y^i) = \text{Exec}\Big(\mathcal{B}_{\text{koop}}(\theta),\, \hat{Y}_0^i\Big), \qquad k \in \mathcal{T} = \{0,\dots,N\}.$$

Koopman 模块满足离散时间稳定性契约

$$C_{\text{stab}}(K) = \rho(K) < 1, \tag{13}$$

其中， $\rho(K)$ 表示谱半径。

契约引导的辨识。遵循一般公式（2），DeepKO 学习问题为

$$\min_\theta \quad \mathcal{J}(\theta) := \sum_{i=1}^m \big(\ell_y^i + \ell_{\text{lin}}^i + \ell_{\text{recon}}^i\big), \tag{14a}$$

$$\text{s.t.} \quad (X^i, Y^i) \in \text{Exec}\Big(\mathcal{B}_{\text{koop}}(\theta),\, \hat{Y}_0^i\Big), \qquad i = 1,\dots,m, \tag{14b}$$

$$C_{\text{stab}}(K(\theta_2)) \le 0. \tag{14c}$$

辨识损失为

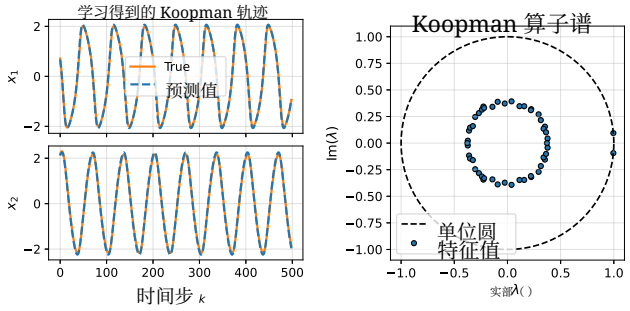
$$\ell_y^i = \sum_{k=0}^{N-1} Q_y \,\|y_{k+1}^i - \hat{y}_{k+1}^i\|_2^2, \tag{15a}$$

$$\ell_{\text{lin}}^i = \sum_{k=0}^{N-1} Q_x \,\big\|\mathcal{B}_{\text{enc}}(\hat{y}_{k+1}^i) - \mathcal{B}_K^k(\mathcal{B}_{\text{enc}}(\hat{y}_k^i))\big\|_2^2, \tag{15b}$$

$$\ell_{\text{recon}}^i = Q_{\text{recon}}\big\|y_0^i - \mathcal{B}_{\text{dec}}(\mathcal{B}_{\text{enc}}(y_0^i))\big\|_2^2. \tag{15c}$$

稳定性契约。为确保 $\rho(K) < 1$，我们通过奇异值分解 (SVD) 对 $K = U\Sigma V^\top$ 进行参数化，通过基于 sigmoid 的缩放将 Σ 中的奇异值限制在指定范围内，并使用 Householder 反射 [51] 确保 U, V 的正交性。通过构造即可满足稳定性契约。与示例 2 不同，此约束是在参数层面施加的，构成了针对潜在线性动力学的精确 A–G 稳定性契约，而非轨迹级证书。

仿真结果。我们通过求解问题 (14) 并使用训练轨迹来训练具有稳定性契约 (13) 的 Deep Koopman CBD，无控制范德波尔振子(6)的 ries。训练后，我们在一个未见过的初始条件上，对模型进行 500 步滚动预测评估。图 6（左）比较了真实状态轨迹与 Koopman 预测，显示二者在整个时域上高度吻合。图 6（右）在复平面中展示了所学 Koopman 算子的特征值以及单位圆；所有特征值均位于单位圆盘内，从而证明所学线性表示是收缩的，并与所施加的稳定性契约一致。



(a) 真实轨迹与预测轨迹的对比。 **(b) Koopman 谱。**

图 6：范德波尔振子的所学 DeepKoopman 模型：时域滚动预测（左）以及稳定化 Koopman 算子的谱（右）。

4 结论

本文介绍了 *∂*CBD，这是一种因果块图的可微扩展，将信息物理系统工程中传统上彼此独立的三种范式统一起来：(i) 通过 CBD 进行组合式建模，(ii) 通过假设–保证和残差契约保证形式化正确性，以及 (iii) 通过可微编程实现学习与优化。通过为模块配备显式接口、时间语义和契约注释，并使其同时支持通用假设–保证语义和可微残差证书， *∂*CBD

将 CBD 从面向仿真的形式体系转变为可训练、可验证且支持组合式建模的框架。由此得到的图兼具因果性、可认证性和梯度兼容性，从而能够端到端地学习嵌入安全性和性能保证的反馈系统。

展望未来， ∂CBD为可验证的智能体AI奠定基础，使感知–规划–控制闭环能够借助可微契约层支持数据驱动适应，同时保持正确性。其组合式契约天然支持对大型、多速率及多智能体系统进行可扩展认证。该框架还支持混合智能架构，在显式契约下结合符号推理、优化层与神经网络组件，以实现可解释且安全的运行。更广泛而言， ∂CBD将现代自动微分库与安全关键验证工具链相衔接，从而在形式化运行时保障下支持实时学习。

致谢

本工作部分得到美国国家科学基金会第2514584号奖项的资助。Ján Drgoňa得到约翰斯·霍普金斯大学Ralph O’Connor可持续能源研究所的支持。

参考值

[1] Martín Abadi 等。2016年。TensorFlow：一种用于大规模机器学习的系统。载于第*12*届 *USENIX* 操作系统设计与实现会议论文集（OSDI’16）。USENIX 协会，美国佐治亚州萨凡纳，265–283。[2] A. Agrawal、B. Amos、S. Barratt、S. Boyd、S. Diamond 和 Z. Kolter。2019年。可微凸优化层。载于神经信息处理系统进展。[3] A. Agrawal、S. Barratt、S. Boyd、E. Busseti 和 W. Moursi。2019年。通过锥规划进行微分。应用与数值优化期刊，1，2，107 115– 。[4] Rania Salih Ahmed、Elmustafa Sayed Ali Ahmed 和 Rashid A Saeed。2021年。工业4.0信息物理系统中的机器学习。载于智能信息物理系统的人工智能范式。IGI Global科学出版公司，20–41。[5] Brandon Amos 和 J. Zico Kolter。2017年。Optnet：作为神经网络一层的可微优化。载于第*34*届国际机器学习会议论文集——第*70*卷（ICML’17）。JMLR，悉尼，136–145。[6] Joel A E Andersson、Joris Gillis、Greg Horn、James B Rawlings 和 Moritz Diehl。2019年。CasADi——用于非线性优化和最优控制的软件框架。《数学规划计算》，11，1，1 1 007 36 10 12532 018 0139 4 – 。doi：。/s - - - 。[7] Gaurav Arya、Moritz Schauer、Frank Schäfer 和 Christopher Rackauckas。2022年。具有离散随机性的程序的自动微分。载于神经信息处理系统进展。第35卷。Curran Associates, Inc.，10435–10447。[8] Shaojie Bai、J. Zico Kolter 和 Vladlen Koltun。2019年。深度平衡模型。载于神经信息处理系统进展。第32卷。CurranAssociates, Inc.[9] Atilim Gunes Baydin、Barak A Pearlmutter、Alexey Andreyevich Radul 和 Jeffrey Mark Siskind。2018年。机器学习中的自动微分：综述。《机器学习研究杂志》，18，153，1–43。[10] Thomas Beckers、Leonardo J Colombo、Sandra Hirche 和 George J Pappas。2021年。针对具有不确定动力学的欠驱动车辆，基于在线学习的轨迹跟踪。*IEEE*控制系统快报，6，2090–2095。[11] Yoshua Bengio、Nicholas Léonard 和 Aaron Courville。2013年。针对条件计算，估计或传播随机神经元的梯度。*arXiv*预印本 *arXiv:1308.3432*。[12] Albert Benveniste 等。2018年。系统设计契约。基础与趋势® 电子设计自动化，12，2-3，124–400。[13] Mathieu Besancon、Joaquim Dias Garcia、Benoit Legat 和 Akshay Sharma。2023年。通过模型变换实现灵活的可微优化。*INFORMS*计算杂志，36，2，456–478。doi:10.1287/ijoc.2022.0283。[14] Mathieu Blondel、Quentin Berthet、Marco Cuturi、Roy Frostig、Stephan Hoyer、Felipe Llinares-López、Fabian Pedregosa 和 Jean-Philippe Vert。2021年。高效且模块化的隐式微分。*arXiv*预印本 *arXiv:2105.15183*。

[15] Jérôme Bolte 和 Edouard Pauwels。2020年。机器学习中自动微分的数学模型。《神经信息处理系统进展》（NeurIPS），33，14036–14047。[16] [SW] James Bradbury 等，JAX：Python+NumPy 程序的可组合变换，版本 0.3.13，2018年。网址：http://github.com/jax-ml/jax。[17] Yang Cao、Shengtai Li、Linda Petzold 和 Radu Serban。2003年。微分代数方程的伴随灵敏度分析：伴随微分代数方程系统及其数值解。《SIAM科学计算杂志》，24，3，1076–1089。doi:10.1137/S1064827501380630。[18] Ricky T. Q. Chen、Yulia Rubanova、Jesse Bettencourt 和 David Duvenaud。2018年。神经常微分方程。《神经信息处理系统进展》。[19] Luca De Alfaro 和 Thomas A Henzinger。2001年。接口自动机。《ACM SIGSOFT软件工程札记》，26，5，109–120。[20] Ben Denckla 和 Pieter J Mosterman。2005年。将因果块图形式化以建模一类混合动态系统。载于：第44届IEEE决策与控制会议论文集。IEEE，4193–4198。[21] Priya Donti、David Rolnick 和 J Zico Kolter。2021年。DC3：一种针对硬约束优化的学习方法。载于：国际学习表征会议。[22] Stuart E. Dreyfus。1962年。变分问题的数值解。《数学分析与应用杂志》，5，1，30–45。[23] Ján Drgoňa、Karol Kiš、Aaron Tuor、Draguna Vrabie 和 Martin Klaučo。2022年。可微预测控制：针对未知非线性系统的显式模型预测控制的深度学习替代方案。《过程控制杂志》，116，80–92。doi:https://doi.org/10.1016/j.jprocont.2022.06.001。[24] Jan Drgona、Aaron Tuor、James Koch、Madelyn Shapiro、Bruno Jacob 和 Dra- guna Vrabie。2023年。NeuroMANCER：具有自适应非线性约束和高效正则化的神经模块。https://github.com/pnnl/neuromanc er。[25] Ján Drgoňa、Aaron Tuor 和 Draguna Vrabie。2024年。学习具有保证的受约束参数化可微预测控制策略。《IEEE系统、人类与控制论汇刊：系统》，54，6，3596 1 0 3607 1109 2024 3368026 – 。doi: . /TSMC. . .[26] Carl Folkestad、Daniel Pastor、Igor Mezic、Ryan Mohr、Maria Fonoberova 和 Joel Burdick。2020年。利用学习得到的Koopman特征函数进行扩展动态模态分解，以实现预测与控制。载于：2020年美国控制会议（ACC），3906–3913。doi:10.23919/ACC45564.2020.9147729。[27] Jonas Friederich 和 Sanja Lazarova-Molnar。2021年。迈向信息物理生产系统的数据驱动可靠性建模。《计算机科学辑刊》，184，589–596。[28] Cláudio Gomes、Joachim Denil 和 Hans Vangheluwe。2020年。因果框图：用于信息物理系统因果建模的一系列语言。载于：《信息物理系统多范式建模基础》。施普林格国际出版公司，97–125。doi:10.1007/978-3-030-43946-0。[29] Alan C. Hindmarsh、Peter N. Brown、Keith E. Grant、Steven L. Lee、Radu Serban、Dan E. Shumaker 和 Carol S. Woodward。2005年。SUNDIALS：非线性及微分/代数方程求解器套件。《ACM数学软件汇刊》，31，3，（2005年9月），363–396。doi:10.1145/1089014.1089020。[30] Yuanming Hu、Luke Anderson、Tzu-Mao Li、Qi Sun、Nathan Carr、Jonathan Ragan-Kelley 和 Frédo Durand。2020年。DiffTaichi：用于物理仿真的可微编程。ICLR。[31] Mike Innes、Alan Edelman、Keno Fischer、Chris Rackauckas、Elliot Saba、Viral B Shah 和 Will Tebbutt。2019年。一种连接机器学习与科学计算的可微编程系统。arXiv预印本 arXiv:1907.07587。[32] Ethan King、James Kotary、Ferdinando Fioretto 和 Ján Drgoňa。2024年。利用度量学习加速算子分裂方法的收敛。载于：2024年IEEE第63届决策与控制会议（CDC），– 。。doi: . /CDC . . .[33] J. Zico Kolter 和 Gaurav Manek。2019年。学习稳定的深度动力学模型。载于：《神经信息处理系统进展》。第32卷。Curran Associates, Inc.[34] James Kotary、Ferdinando Fioretto、Pascal Van Hentenryck 和 Bryan Wilder。[n. d.] 端到端受约束优化学习：综述。国际人工智能联合会议。doi:10.24963/ijcai.2021/610。[35] Bethany Lusch、J. Nathan Kutz 和 Steven L. Brunton。2018年。利用深度学习对非线性动力学进行通用线性嵌入。《自然·通讯》，9，1，（2018年11月），4950。[36] Sayak Mukherjee、Ján Drgoňa、Aaron Tuor、Mahantesh Halappanavar 和 Draguna Vrabie。2022年。神经Lyapunov可微预测控制。载于：2022年 110 9 61 2097 2104 10 51059 2022 9992386 IEEE st 决策与控制会议（CDC），– 。doi: . /CDC . . .[37] Adam Paszke 等。2019年。PyTorch：一种命令式风格、高性能的深度学习库。载于：第33届国际神经信息处理系统会议论文集，文章721。Curran Associates Inc.[38] Claudius Ptolemaeus，（编）2014年。《使用Ptolemy II进行系统设计、建模与仿真》。Ptolemy.org，美国加利福尼亚州伯克利。https://ptolemy.org/books/Systems/。

[39] GV Puskorius 和 LA Feldkamp。1994年。用于神经控制的截断时间反向传播与卡尔曼滤波器训练。载于：*1994*年*IEEE*国际神经网络会议（*ICNN'94*）论文集。第4卷。IEEE，2488–2493。 [40] Rajiv Sambharya、Georgina Hall、Brandon Amos 和 Bartolomeo Stellato。2024年。学习热启动不动点优化算法。《机器学习研究杂志》，25，166，1–46。 [41] Adnane Saoud、Antoine Girard 和 Laurent Fribourg。2021年。连续时间系统的假设-保证契约。《自动化》，134，109910。 [42] Mischa Schmidt 和 Christer Åhlund。2018年。作为信息物理系统的智能建筑：面向能源效率的数据驱动预测控制策略。可再生与可持续能源评论，90，742–756。 [43] Quan Sun、Wei Zhang、Chao Wang 和 Zhiming Liu。2023年。混合Simulink框图的基于契约的语义与精化。《系统架构杂志》，143，102963。doi:10.1016/j.sysarc.2023.102963。 [44] Naoya Takeishi、Yoshinobu Kawahara 和 Takehisa Yairi。2017年。学习用于动态模态分解的Koopman不变子空间。载于：神经信息处理系统进展。第30卷。Curran Associates, Inc. [45] Bo Tang 和 Elias B Khalil。2024年。Pyepo：基于PyTorch的端到端“先预测再优化”库，适用于线性规划和整数规划。数学规划计算，（2024年7月）。doi:10.1007/s12532-024-00255-x。

[46] Bo Tang、Elias B. Khalil 和 Ján Drgoňa。2025年。带可行性保证的混合整数非线性规划优化学习。（2025年）。arXiv：2410.11061 [计算机科学—机器学习]。[47] Bart van Merriënboer、Dan Moldovan 和 Alexander Wiltschko。2018年。机器学习中的自动微分：我们目前所处的位置以及未来的发展方向。神经信息处理系统进展（*NeurIPS*），31。[48] Kush R Varshney 和 Homa Alemzadeh。2017年。论机器学习的安全性：信息物理系统、决策科学与数据产品。《大数据》，5，3，246 255–。[49] Paul J. Werbos。1990年。时间反向传播：其作用及实现方法。《*IEEE*会刊》，78，10，1550–1560。[50] Enoch Yeung、Soumya Kundu 和 Nathan Hodas。2019年。为非线性动力系统的 Koopman 算子学习深度神经网络表示。在*2019*年美国控制会议（*ACC*），4832–4839。doi:10.23919 / ACC.2019.8815339。[51] Jiong Zhang、Qi Lei 和Inderjit S. Dhillon。2018年。通过高效奇异值分解参数化稳定深度神经网络的梯度。在国际机器学习会议。[52] Wei Zhang、Quan Sun、Chao Wang 和 Zhiming Liu。2022年。通过精化证明 Simulink 块图的正确性。《无线通信与移动计算》，2022年，1，8015896。doi:10.1155/2022/8015896。