利用物理信息神经网络学习基于热力学的非线性本构材料模型的解

Shahed Rezaei[1, *], Ahmad Moeineddin[2], Ali Harandi[3]

[1]德国亚琛 *D-52072，Intzestr. 5，ACCESS e.V.*

[2]德国德累斯顿 *D-01187，Georg-Schumann-Str. 7*，德累斯顿工业大学结构分析研究所

[1]德国亚琛 *D-52074，Mies-van-der-Rohe-Str. 1*，亚琛工业大学应用力学研究所

* 通讯作者：*s.rezaei@access-technology.de*

arXiv:2304.06044v2 [cs.CE] 6 Sep 2023

摘要

我们应用物理信息神经网络求解非线性、路径相关材料行为的本构关系。因此，训练后的网络不仅满足所有热力学约束条件，还能在无需初始数据的情况下，针对任何给定的加载场景即时提供当前材料状态的信息（即自由能、应力以及内部变量的演化）。这项工作的一个优势是，它绕过了求解复杂材料模型中非线性方程所需的重复牛顿迭代。此外，我们还提供了降低获取切线算子所需导数阶数的策略。训练后的模型可作为用户定义材料模型，直接用于任何有限元软件包（或其他数值方法）中。然而，在适当定义配点，以及整合多个同时处于激活或非激活状态的非等式约束方面，仍然存在挑战。我们使用具有非线性硬化定律的经典von Mises塑性模型等率无关过程，以及用于具有非线性软化定律的界面开裂行为的局部损伤模型，对该方法进行了测试。为了展示该方法处理三维（三维）场景中复杂路径依赖性的适用性，我们利用控制三维界面模型损伤模型的方程对所提方法进行了测试。这类模型常用于晶界处的晶间断裂。我们观察到，通过所提方法获得的结果与采用经典方法获得的结果完全一致。此外，与传统方法相比，所提方法在实现工作量和计算时间方面都显著降低。最后，我们讨论了这一新方法未来发展的潜力和仍待解决的挑战。

关键词：物理信息神经网络、本构关系、非线性材料行为、路径相关材料模型、有限元分析。

1. 引言

实现和评估非线性材料模型是一项艰巨但至关重要的任务，对于在许多应用中获得可靠的预测结果不可或缺。例如，可能需要采用先进模型来描述金属、土壤和聚合物复杂的塑性行为，或考虑材料随着损伤和微裂纹演化而发生的软化 [1, 2, 3]。通常，建模人员

使用热力学框架推导这些模型，从而帮助他们为材料的内部状态建立所有必要的方程和演化定律。完成这一步后，必须使用特定的编程语言编写复杂的代码，这一过程可能既困难又耗时。求解材料本构方程时，通常采用迭代牛顿法，该方法往往非常有效。然而，由于在有限元（FE）框架中需要针对每个积分点和每次非线性迭代重复执行算法，这种方法也可能带来高昂的计算成本。总之，非线性材料模型的开发与实现需要丰富的专业知识和大量投入，还需要获得适当的计算资源。此外，材料行为还受到较低尺度的显著影响。多尺度材料建模需要求解微观尺度上的边值问题。获得微观尺度解后，均匀化结果会传递至宏观尺度的结构问题 [4]。然而，尺度间的信息转换是多尺度分析中的瓶颈，因为其中涉及计算密集型任务，可能耗费大量时间。

计算材料力学领域的一个最新趋势是使用深度学习（DL）方法（如神经网络（NN））来绕过昂贵的计算。这一理念是利用足够数量的数据训练神经网络，以预测材料属性 [5, 6, 7]，或将训练好的神经网络用作特定材料系统的替代模型 [8, 9]。例如，Ali 等人开展的研究 [10]，将神经网络模型与率相关晶体塑性有限元方法相结合，并展示了所提模型对非比例加载路径的预测能力。这种方法的成功高度依赖于可用数据的质量和数量，以及所使用的机器学习优化器。Huang 等人 [11]开发了一种基于机器学习的材料建模框架，其中提出将累积绝对应变作为塑性模型的历史变量，以考虑加载历程。Frankel 等人 [12] 使用卷积神经网络，根据初始微观结构和外部加载预测应力场的演变。更多信息请参阅 [13, 14, 15], 以及 [16, 17, 18, 19] 中的研究论文和综述。

在前述方法中，神经网络的预测结果完全基于现有数据，因此人们担心，当遇到超出训练范围的情况时，神经网络的可靠性会受到影响。然而，近期研究表明，在训练过程中引入热力学约束，可以提高神经网络对未见情形的预测准确性 [20]。Xu 等人 [21] 提出了一种神经网络架构，其输出为切线刚度矩阵的 Cholesky 因子，从而以弱约束的方式保证应变能函数的凸性。Liu 等人 [22] 开发了一种数据驱动的多尺度材料建模方法，将均匀化理论原理和基本物理学原理与机器学习技术相结合。另请参阅 [23]，了解该方法在再现纤维增强材料蠕变行为方面的进一步拓展。Heider 等人 [24] 讨论了基于信息图的神经网络在各向异性弹塑性行为建模中的应用，并提出了生成坐标系不变机器学习本构模型的方法。Vlassis 和 Sun [25] 提出了一个深度学习框架，用于训练具有可解释组件（如屈服面和塑性流动）的平滑弹塑性模型；这些组件依据一组网络预测结果演化。他们还表明，该方法能够准确预测循环应力路径。Flaschel 等人 [26] 开发了一种方法，用于无监督地发现属于未知本构行为类别（如黏弹性和弹塑性）的材料定律。其核心思想是利用位移和反力数据，确定亥姆霍兹自由能和耗散势。Fuhg 等人 [27]讨论了一种模块化弹塑性公式，其中模型的不同组件可根据可用信息量选择采用现象学模型或数据驱动模型。

视可用信息量而定。Weber 等人 [28] 通过引入物理约束改进了超弹性材料的神经网络训练，并表明这些改进能够带来更好的学习效果。Kalina 等人 [29] 提出了一个数据驱动多尺度框架，利用物理约束神经网络解决有限应变超弹性问题。其思路是将亥姆霍兹自由能密度设为输出，并以一组不变量作为神经网络的输入来训练神经网络。因此，若干物理原理能够自动得到满足。另请参阅 [30]关于多凸神经网络本构模型的研究。

Masi 等人 [31] 提出了用于本构建模的基于热力学的神经网络。作者提出将材料的自由能学习为应变和内部状态变量的函数。另请参见 [32] 关于学习弹性胞状超材料有效应变能的研究。Masi 和 Stefanou [33]还扩展了其方法，使其能够根据数据和第一性原理识别复杂材料中内部变量的容许集合。

除前馈神经网络外，循环神经网络（RNN）等其他深度学习方法也是处理路径相关材料行为的合适选择 [14, 34, 35]。He 和 Chen [36] 提出了一种数据驱动本构建模方法，并考虑了路径相关材料的热力学原理。Bonatti 和 Mohr [37] 研究了一个弹塑性实例，并提出了一种遵循自洽性的循环神经网络架构，该架构应用于由大量小增量组成的长序列时表现最佳。Koeppe 等人 [38]提出了一种物理解释方法，用于解释循环神经网络。作者针对弹塑性和黏弹性本构模型研究了相关案例。Danoun 等人 [39] 提出了一种基于物理与人工智能的混合模型，通过循环神经网络预测非线性力学行为，并在训练阶段考虑特定的热力学约束。另请参见 [40], [41] 和 [42]的相关发展。在所评述的文章中，训练神经网络需要足够的数据。

遵循物理信息神经网络（物理信息神经网络） [43], 的理念，可以将物理定律用作损失函数，以提升网络结果 [44, 45]。一旦问题的底层物理规律被完全掌握，物理信息神经网络就可以在没有任何初始数据的情况下，仅基于给定的方程组进行训练 [46, 47, 48]。Wei等人 [49] 建立了一种一致的神经网络方法，通过将正能量耗散等物理条件作为额外的训练约束整合到损失函数中，来建模界面的本构行为。Tipireddy等人 [50] 研究了使用物理信息神经网络学习动力系统的未知动力学或本构关系。根据作者的研究结果，与学习整个动力学相比，训练后的模型在学习本构关系时表现出更高的准确性。Haghighat等人 [51]提出了一种用于本构建模的物理信息神经网络公式，其中将弹塑性理论的不等式约束嵌入物理信息神经网络的损失函数中。在 [52], 中，作者以冯·米塞斯弹塑性为例，阐述了物理信息神经网络在固体力学中的扩展。另请参见 [53] 关于向有限应变弹塑性扩展的研究，以及 [54] 关于经典弹塑性的深度能量法的研究。Eghbalian等人 [55] 提出了一种融入弹塑性物理规律的神经网络架构，并表明嵌入这一方面增强了模型对训练数据之外加载工况的外推能力。

综述的这些研究对两个主要类别作出了重要贡献。第一类研究利用从数值模拟或实验测量中获得的充足数据训练神经网络（例如见 [56, 57]）。为提高预测精度，这些方法纳入了物理约束 [58, 39, 59, 55]。第二类研究则应用物理信息神经网络技术，针对特定边值问题研究非线性材料行为 [52, 53]，或用于表征本构模型 [51]。

本研究的主要创新贡献在于应用 PINN，并具备满足热力学约束以及针对任意加载实时预测材料状

态的能力，且无需初始数据。需要注意的是，本研究采用了简单的前馈神经网络（FFNN），可轻松集成到任何有限元计算中。此外，采用自动微分来计算切线算子，进一步简化了实现过程。

为帮助读者更清晰地理解，图 1 总结了在不同加载条件下预测复杂且具有路径相关性的材料行为的现有方法。根据实验或模拟所得的可用数据，可以选择数据驱动（DD）方法或基于物理的方法。在基于物理的建模中，需要通过数值方法求解模型的一致方程，而我们所提的方法可与标准返回映射算法相媲美。值得注意的是，我们的方法不依赖任何数据，这使其区别于现有的 DD 方法。

在 DD 方法中，可以采用无模型方法 [60]，也可以训练深度学习模型来拟合可用数据并构建代理模型 [34]。在此背景下，所提出的 COMM-PINN 方法可以作为额外的损失项整合到神经网络中，这与其他作者提出的类似方法一致，正如前文所述（例如，[49, 55, 39, 53, 52]）。

接下来，我们将总结推导一致材料模型的步骤，并介绍一些案例研究，包括采用非线性各向同性硬化定律的弹塑性行为，以及用于模拟界面开裂的非线性软化定律局部损伤模型。最后，我们将讨论研究结果和未来可能的发展方向。

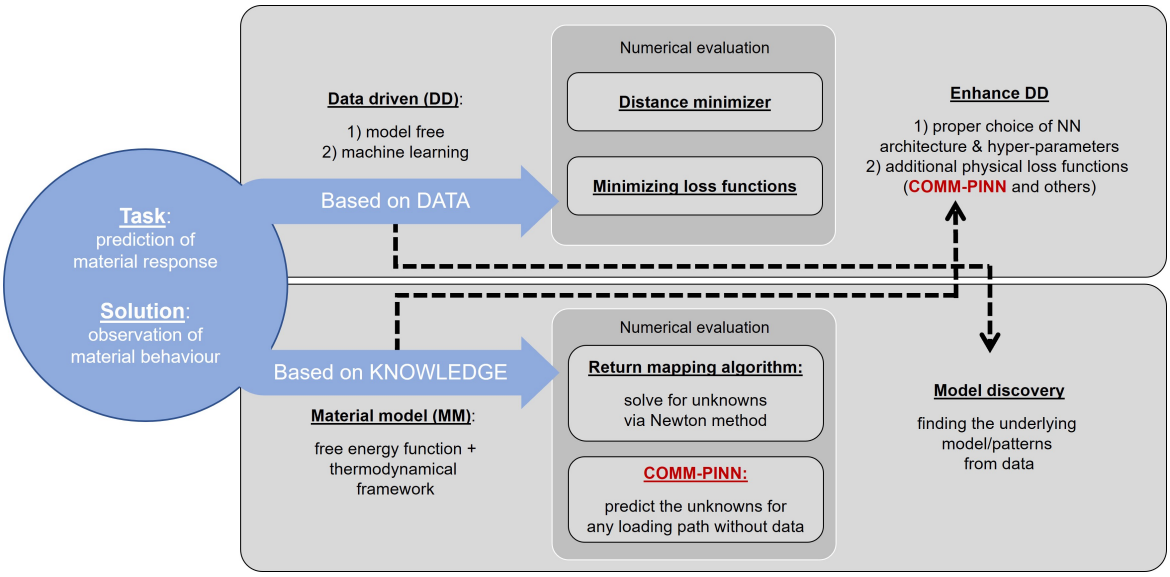


图 1：预测材料行为的不同方法概述。本文提出的方法（COMM-PINN）的核心思想是利用神经网络替代返回映射算法，并改进其他数据驱动方法。各类方法的参考文献见引言部分。

2. 获取一致的本构关系并使用物理信息神经网络求解

小变形、纯力学情形以及等温过程下，耗散不等式的局部形式为 [3]

$$-\dot{\psi}(\boldsymbol{\varepsilon}, \boldsymbol{\xi}_k) + \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}} \overset{!}{\geq} 0. \tag{1}$$

其中， $\psi = \psi(\varepsilon,\ \boldsymbol{\xi}_k)$是材料的自由能， $\boldsymbol{\sigma}$ 是应力张量， $\boldsymbol{\xi}_k$ 表示一组内部（状态）变量， ε 是应变张量。上述不等式可简化为

$$(\underbrace{\boldsymbol{\sigma} - \partial_{\boldsymbol{\varepsilon}}\psi}_{=\,\mathbf{0}})\ \dot{\boldsymbol{\varepsilon}} - \Sigma_k \underbrace{\partial_{\boldsymbol{\xi}_k}\psi}_{=\,\boldsymbol{q}_k}\ \dot{\boldsymbol{\xi}}_k \overset{!}{\geq} 0. \tag{2}$$

接下来，热力学共轭力自动满足

$$\boldsymbol{\sigma} = \partial_{\boldsymbol{\varepsilon}}\psi, \quad \boldsymbol{q}_k = \partial_{\boldsymbol{\xi}_k}\psi. \tag{3}$$

. ！耗散不等式的剩余部分（即 $\mathcal{D} = -\Sigma_k \boldsymbol{q}_k\ \dot{\boldsymbol{\xi}}_k \geq 0$）仍需对任意过程成立。为确保耗散率为正并对非线性材料响应进行建模，通常需要引入适当的屈服准则 $YLD(\boldsymbol{\xi}_k,\ .....)$，以及状态变量的演化方程 $EVL(\boldsymbol{\xi}_k,\ .....)$。在接下来的章节中，我们将通过塑性和损伤模型这两个实际例子，更详细地探讨这一部分。

至此，我们已具备建立神经网络以预测非线性材料行为所需的所有要素。图 2 提出了三种架构，用于通过物理信息神经网络（COMM-PINN）求解任意本构材料模型中的方程。

接下来，我们将考察每种建议网络的优缺点。在所有情况下，用户都要定义自由能函数和所需的状态变量。此外，网络的输入是当前应变和材料历史。

对于情况 A，只有状态变量是输出。所有热力学共轭力（$\boldsymbol{\sigma}^{i+1}$ 和 $\boldsymbol{q}_k^{i+1}$ ）以及一致切线算子 $\mathbb{C}^{i+1} = \mathrm{d}\boldsymbol{\sigma}^{i+1}/\mathrm{d}\boldsymbol{\varepsilon}^{i+1}$ 均通过自动微分（AD）计算。一种可能的缺点是需要进行二阶微分，因此必须使用适当的激活函数 [31]。此情况下的损失函数基于屈服函数和演化定律。

对于情况 B，热力学共轭力被定义为额外的输出，其定义则作为额外的损失函数使用（关于如何在物理信息神经网络中降低导数阶数，参见 [61] 及其中的参考文献）。第二种架构的一个优点是，只需一阶导数即可得到切线算子。不过，这种情况下的网络可能需要更密集，以处理多个输出。在上述两种情况下，用户唯一需要做的就是定义自由能函数；所有热力学力以及更新后的状态变量，都是网络自然产生的结果。最后，值得注意的是，由于所涉及方程之间相互纠缠，且需要考虑众多损失项，在这种情况下训练神经网络可能尤其困难。

对于案例 C，我们利用自由能函数的解析微分，并将热力学共轭力（例如应力张量）的解析表达式输入网络，从而简化训练。对于复杂的能量函数，推导此类表达式可能是一个缺点，但通常求得此类偏导数并不困难。其优点是损失项更少、方程之间的纠缠更少，并且只需一阶导数即可得到切线算子。另一种方法是利用解析微分求得切线算子，从而避免使用 AD。虽然这种方法可能不适用于更复杂的材料模型，但它是一个值得考虑的有趣选择。

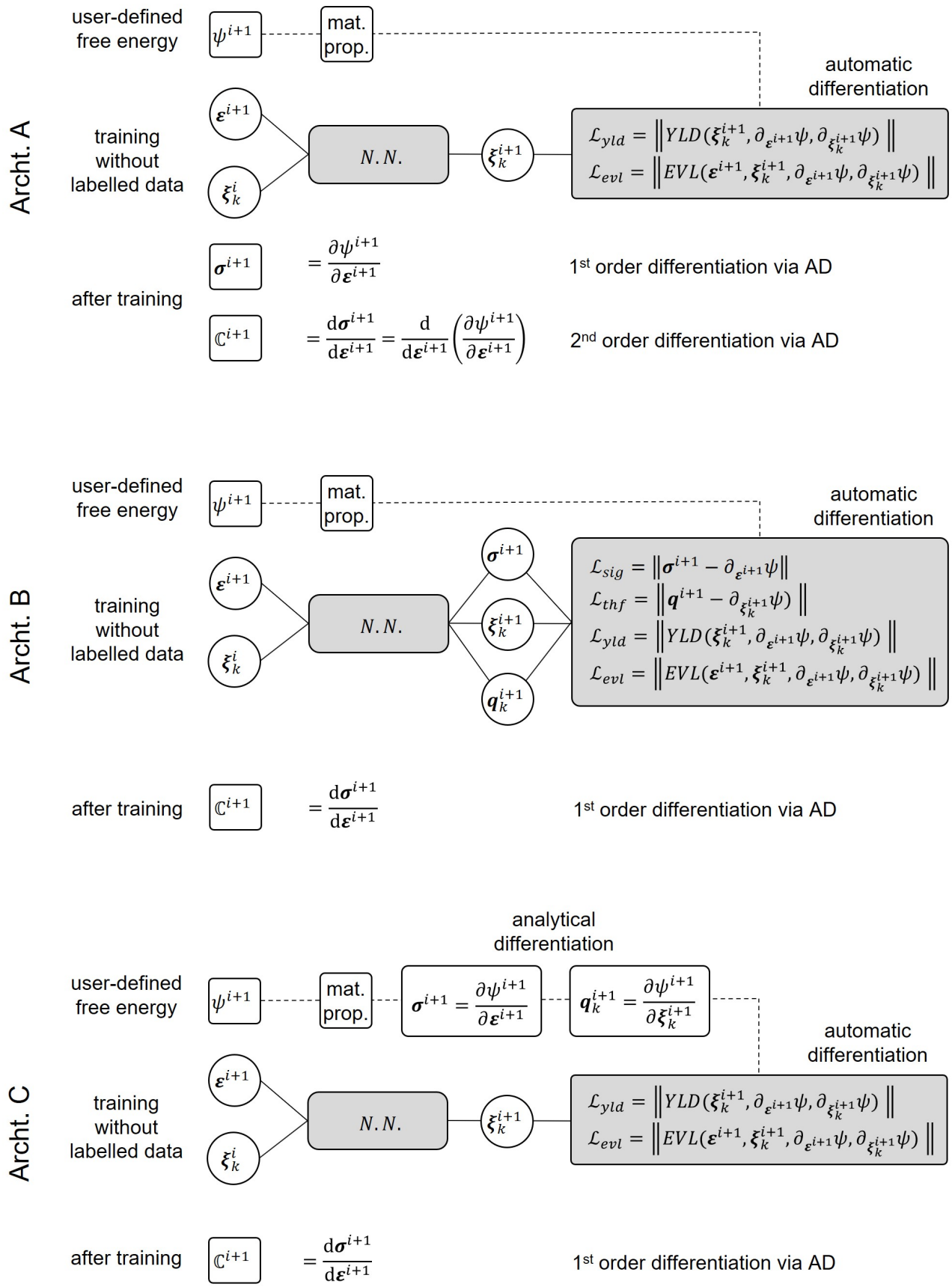


图 2：通过物理信息神经网络（COMM-PINN）求解本构材料方程的三种神经网络架构和损失函数建议。

3. von Mises塑性模型案例研究

3.1. 模型推导

我们仅考虑小变形条件下的一维情形。尽管如此，当前的研究也可以扩展到更高维度，以及预期具有更强非线性行为的案例研究。应变变量 ε 定义为位移 u 对坐标 x 的导数（即 $\varepsilon = \mathrm{d}u/\mathrm{d}x$）。在弹塑性行为中，应变变量可加性分解为弹性部分和塑性部分，如下所示

$$\varepsilon = \varepsilon_e + \varepsilon_p. \tag{4}$$

亥姆霍兹自由能 ψ 由弹性部分（ψ_e）和塑性部分（ψ_p）构成。因此，我们写出以下表达式

$$\psi(\varepsilon, \varepsilon_p, \xi_p) = \psi_e(\varepsilon, \varepsilon_p) + \psi_p(\xi_p), \tag{5}$$

$$\psi_e\left(\varepsilon, \varepsilon_p\right) = \frac{1}{2} E(\varepsilon - \varepsilon_p)^2, \tag{6}$$

$$\psi_p\left(\xi_p\right) = h_1 \left(\xi_p + \frac{e^{-h_2 \xi_p} - 1}{h_2}\right). \tag{7}$$

在能量的弹性部分（ψ_e）中，材料的杨氏模量为 E。此外，在能量的塑性部分中，我们引入参数 ξ_p，它表示累积塑性应变。这里只考虑各向同性硬化，常数 h_1 和 h_2 为相应的塑性硬化参数。利用求导的链式法则，并考虑应变率的弹塑性加性分解，等温局部形式的克劳修斯-迪昂不等式可写为

$$\left(\sigma - \partial_\varepsilon \psi\right) \dot{\varepsilon} - \partial_{\epsilon_p} \psi \; \dot{\varepsilon}_p - \partial_{\xi_p} \psi \; \dot{\xi}_p \overset{!}{\geq} 0. \tag{8}$$

上述表达式必须对任意热力学过程成立。通常的做法是令方括号中的表达式为零。模型的状态关系以及热力学共轭力可自动表示为

$$\sigma = \partial_\varepsilon \psi = \partial_\varepsilon \psi_e = E(\varepsilon - \varepsilon_p), \tag{9}$$

$$q_p = \partial_{\xi_p} \psi = \partial_{\xi_p} \psi_p = h_1 \; (1 - e^{-h_2 \xi_p}). \tag{10}$$

此时，耗散不等式的剩余部分可写为

$$\sigma \; \dot{\varepsilon}_p - q_p \; \dot{\xi}_p \overset{!}{\geq} 0. \tag{11}$$

塑性屈服准则（基于 von Mises 模型）表示为 s

$$\phi_p = |\sigma| - \left(\sigma_{y0} + q_p\right). \tag{12}$$

此外，采用以下塑性内部变量的演化定律:

$$\dot{\varepsilon}_p = \dot{\lambda}_p \; \frac{\partial \phi_p}{\partial \sigma} = \dot{\lambda}_p \; \mathrm{sgn}(\sigma), \tag{13}$$

$$\dot{\xi}_p = -\dot{\lambda}_p \; \frac{\partial \phi_p}{\partial q_p} = \dot{\lambda}_p. \tag{14}$$

最后， 模型的塑性加载/卸载条件被纳入考虑 t:

$$\dot{\lambda}_p \geq 0, \quad \phi_p \leq 0, \quad \dot{\lambda}_p \; \phi_p = 0. \tag{15}$$

通过考虑关系式 13、14 以及 15 中的约束条件，不等式 11 得以满足。

3.2. 塑性模型的实现与算法方面

算法 1 概述了通常用于求解塑性方程的返回映射算法。为了求解未知量，必须对塑性方程（或任何其他非线性材料程序）进行线性化，这通过在有限元方法的每个积分点处使用局部牛顿求解器来实现。此外，用户还必须推导用于有限元程序的切线算子方程。在结构层级的每个全局迭代步骤中，以及计算域内的每个积分点处，都必须重复这些任务，因此计算成本很高。

Algorithm 1 Solving the elastoplastic governing equations at pseudo time t^{i+1}

Input: material parameters, current strain: ε^{i+1}, history variables: ε_p^i, ξ_p^i
Output: current stress: σ^{i+1}, current history variables: ε_p^{i+1}, ξ_p^{i+1}, tangent: $C_p^{i+1} = \mathrm{d}\sigma^{i+1}/\mathrm{d}\varepsilon^{i+1}$

$k = 1$
Set trial values: $\varepsilon_p^{i+1,k} = \varepsilon_p^i$, $\xi_p^{i+1,k} = \xi_p^i$
loop *plasticity solver* **do**
    if $\phi_p^{tr} = |E(\varepsilon^{i+1} - \varepsilon_p^i)| - \left(\sigma_{y0} + q_p(\xi_p^i)\right) \leq 0$
        $\varepsilon_p^{i+1} = \varepsilon_p^i$ and $\xi_p^{i+1} = \xi_p^i$ and $C^{i+1} = E$
    else solve for $\varepsilon_p^{i+1,k+1}$ and $\xi_p^{i+1,k+1} = \xi_p^i + \Delta\lambda_p^{k+1}$ (see Eqs 12,13)
        $r_p^{(1)} = \varepsilon_p^{i+1,k+1} - \varepsilon_p^i - (\xi_p^{i+1,k+1} - \xi_p^i)\dfrac{\partial \phi_p}{\partial \sigma} \stackrel{!}{=} 0$
        $r_p^{(2)} = \phi_p^{i+1} = |\sigma^{i+1,k+1}| - \left(\sigma_{y0} + q_p\left(\xi_p^{i+1}\right)\right) \stackrel{!}{=} 0$
        $[\varepsilon_p^{i+1,k+1} \quad \xi_p^{i+1,k+1}]^T = [\varepsilon_p^{i+1,k} \quad \xi_p^{i+1,k}]^T - \boldsymbol{K}_p^{-1}\boldsymbol{r}_p$
    end if
    if $\left|\varepsilon_p^{i+1,k+1} - \varepsilon_p^{i+1,k}\right| > \mathrm{tol}$ **OR** $\left|\xi_p^{i+1,k+1} - \xi_p^{i+1,k}\right| > \mathrm{tol}$
        $k++$
        CYCLE loop *plasticity*
    else
        $g_p^{i+1} = g_p^{i+1,k+1}$, $\xi_p^{i+1} = \xi_p^{i+1,k+1}$
        EXIT loop *plasticity solver*
    end if
end loop
Compute stress $\sigma^{i+1} = E(\varepsilon^{i+1} - \varepsilon_p^{i+1})$ see Eq. 53
Compute tangent $C^{i+1} = E(1 - \frac{E}{E + h_1 h_2 \, \exp(-h_2 \xi_p^{i+1})})$

算法 1 中的上标 k 表示内部循环中塑性求解器的第 k–次迭代。此外，我们使用符号 $\Delta\lambda_p = \Delta t \ \lambda_p$，其中 Δt 表示伪时间步。一旦塑性被激活（$\phi_p > 0$），就会根据牛顿-拉夫森方法对塑性残差（$r_p^{(1)}$和 $r_p^{(2)}$）进行线性化，以求解未知量（$\varepsilon^{i+1,\,k+1}_{,p}$和 $\xi^{i+1,\,k+1}_{,p}$）。矩阵 $\boldsymbol{K}_p = \partial \boldsymbol{r}_p / \partial \boldsymbol{U}_p$ 包含塑性残差向量 $\boldsymbol{r}_p$ 对未知量（$\boldsymbol{U}_p = \{\varepsilon_p^{i+1}, \xi_p^{i+1}\}$）的导数。如果未知内变量解的变化小于某个容差（$\mathrm{tol} = 10^{-10}$），则将所得结果视为收敛值，并用于计算应力值和切线算子。最后，我们能够在有限元层面建立全局残差 [3, 62]。

3.3. 使用物理信息神经网络求解具有非线性各向同性硬化规律的弹塑性行为

基于 Raissi 等人提出的方法 [43], ，网络的输入为配点的位置。在材料建模中，配点包含当前加载状态（施加的应变/间隙）信息，以及历史变量的可接受数值（即塑性应变、损伤等）。这里我们提议采用图 2 所示的 C 架构。因此，输出层包含当前（更新后的）历史变量（即新的塑性应变、损伤等）。应力、能量和切线算子等其他量，则根据已知的材料自由能方程计算。换言之，我们无需额外的神经网络来计算这些量。

每个神经网络的结构都采用标准形式，即可分为单个输入层、若干个可能的隐藏层和输出层。每一层都与下一层相连接，以传递信息 [63]。在每一层内部，其神经元之间不存在连接。因此，我们通过向量 $\boldsymbol{z}^l$表示从 $l-1$ 层到 l 的信息旁路。向量 $\boldsymbol{z}^l$ 的每个分量均通过

$$z_m^l = a(\sum_{n=1}^{N_l} w_{mn}^l z_n^{l-1} + b_m^l), \quad l = 1, \ldots, L. \tag{16}$$

在公式(16) 中， z_n^{l-1}是 $l-1$ 层中的第 n个神经元。分量 w_{mn} 显示了 $l-1$ 层的第 n个神经元与l 层的第 m 个神经元之间的连接权重。 l个隐藏层中的每个神经元都拥有一个偏置变量 b_m^l。数值 N_I对应于 l 个隐藏层中的神经元数量。隐藏层的总数为 L。字母 a 代表每个神经元中的激活函数。激活函数 $a(\cdot)$ 通常是一个非线性函数。激活函数的适当选择取决于问题，应基于超参数研究 [46], [64] 获得。

对于塑性问题，输入层包含 ε^{i+1}、 ε_p^i 和 ξ_p^i。我们用向量 $\boldsymbol{X} = \{\varepsilon^{i+1}, \varepsilon_p^i, \varepsilon_p^i\}$ 表示输入层。输出层写作 $\boldsymbol{Y} = \{\varepsilon_p^{i+1}, \xi_p^{i+1}\}$。我们针对每个输出变量分别使用一个全连接前馈神经网络（另见图 3）。网络的可训练参数集表示为 $\boldsymbol{\theta} = \{\boldsymbol{W}, \boldsymbol{b}\}$，分别对应神经网络的权重和偏置。将每个神经网络结构视为 $\mathcal{N}$，通过神经网络进行本构材料建模的结果为

$$\varepsilon_p^{i+1} = \mathcal{N}_{\varepsilon_p}(\boldsymbol{X}; \boldsymbol{\theta}), \qquad \xi_p^{i+1} = \mathcal{N}_{\xi_p}(\boldsymbol{X}; \boldsymbol{\theta}). \tag{17}$$

神经网络的输出是可训练参数的函数，训练过程则通过最小化物理损失函数来完成。接下来，我们根据已定义的输入层和输出层，为所引入的塑性模型构建残差和条件。为此，需要将屈服函数以及内变量的演化方程纳入神经网络的损失函数中。将塑性问题的总损失项之和记为 $\mathcal{L}_{pt}$，并根据算法 1，定义 $\mathcal{L}_{pt}$ 为

$$\mathcal{L}_{pt} = \underbrace{w_{uep}\mathcal{L}_{uep} + w_{uxp}\mathcal{L}_{uxp}}_{\text{elastic response}} + \underbrace{w_{evp}\mathcal{L}_{evp} + w_{ylp}\mathcal{L}_{ylp}}_{\text{plastic evolution}} + \underbrace{w_{kep}\mathcal{L}_{kep} + w_{kyp}\mathcal{L}_{kyp}}_{\text{KKT conditions}}. \tag{18}$$

在公式（18）中，不同的损失项涵盖了所有可能的加载、卸载和再加载场景。前两项（$\mathcal{L}_{uep}$ and $\mathcal{L}_{uxp}$）确保当试屈服函数（ϕ_p^{tr}）为负时，塑性应变和累积塑性应变不会发生演化。一旦 $\phi_p^{tr} > 0$,

演化定律的主要塑性残差以及当前屈服函数便会激活，分别记为 $\mathcal{L}_{evp}$ and $\mathcal{L}_{ylp}$。最后，为确保始终满足 Karush–Kuhn–Tucker 条件（KKT 条件），我们加入了最后两个损失项（$\mathcal{L}_{kep}$ and$\mathcal{L}_{kyp}$）。下面总结了所有这些相关损失项

$$\mathcal{L}_{uep} = \text{均方误差}\left((\varepsilon_p^{i+1} - \varepsilon_p^i)\text{ReLU}(-\phi_p^{tr})\right), \tag{19}$$

$$\mathcal{L}_{uxp} = \text{均方误差}\left((\xi_p^{i+1} - \xi_p^i)\text{ReLU}(-\phi_p^{tr})\right), \tag{20}$$

$$\mathcal{L}_{evp} = \text{MSE}\left((\varepsilon_p^{i+1} - \varepsilon_p^i - (\xi_p^{i+1} - \xi_p^i)\text{sgn}(E(\epsilon^{i+1} - \varepsilon_p^{i+1})))\text{Relu}(\phi_p^{tr})\right), \tag{21}$$

$$\mathcal{L}_{ylp} = \text{MSE}\left(\left(\text{abs}(E(\varepsilon_p^{i+1} - \varepsilon_p^{i+1})) - (\sigma_{y0} + h_1(1 - \exp(-h_2\xi_p^{i+1})))\right)\text{Relu}(\phi_p^{tr})\right), \tag{22}$$

$$\mathcal{L}_{kep} = \text{均方误差}\left(\text{ReLU}(\phi_p^{i+1})\right), \tag{23}$$

$$\mathcal{L}_{kyp} = \text{均方误差}\left(\text{ReLU}(-\xi_p^{i+1} + \xi_p^i)\right). \tag{24}$$

在上述关系中，ϕ_p^{tr}是所谓的试屈服函数，它通过输入层中的（已知）量进行评估（另见算法 1）：

$$\phi_p^{tr} = \text{abs}(E(\varepsilon^{i+1} - \varepsilon_p^i)) - \left(\sigma_{y0} + h_1(1 - \exp(-h_2\xi_p^i))\right). \tag{25}$$

为完整起见，表达式“abs”和“sgn”分别表示绝对值和符号函数。均方误差定义为

$$\text{MSE}(\bullet)_{type} = \frac{1}{n}\sum_{i=1}^{n}(\bullet)^2, \tag{26}$$

其中，n为观测点的数量。最终损失项在每一个配点处最小化。该数学优化问题可写为

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathcal{L}_{pt}(\boldsymbol{X};\boldsymbol{\theta}), \tag{27}$$

其中 $\boldsymbol{\theta}^*$ 是网络的最优可训练参数（权重和偏置） k.

备注 1 在上述损失项中，“ReLU”函数充当开关（if 条件）。换句话说，我们通过 ϕ_p^{tr} 的符号来区分弹性加载/卸载和塑性变量的演化。另请参见经典迭代求解器中的相同过程（即算法 1）。为此，也可以使用其他选项，例如“sgn”函数。根据我们的研究，我们发现使用“ReLU”函数更有益。

在图 3中，我们总结了所有损失项以及通过物理信息神经网络进行本构材料建模背后的理念。为清晰起见，所有输入变量均以蓝色表示，而输出（未知）变量则以红色表示。请注意，材料属性目前保持不变，并以绿色表示。

训练完成后，网络会预测未知的内部变量（即对于弹塑性问题，ε_p^{i+1}和 ξ_p^{i+1}）。有了更新后的内部变量以及当前应变（ε^{i+1}），便可构建自由能函数（ψ^{i+1}）和应力值（σ^{i+1}）。请注意，根据热力学框架，应力的功能可通过对能量函数关于应变进行解析微分来获得 [31, 29]，另见图 2 中的架构 C。有限元计算中的最后一个重要参数是切线算子，它定义为当前

应力值关于当前应变值的全导数。该参数通过对训练后的网络进行自动微分获得。与其他研究（尤其是 [31] 中的研究）相比，本研究的主要区别在于，我们没有向前馈神经网络提供任何数据。

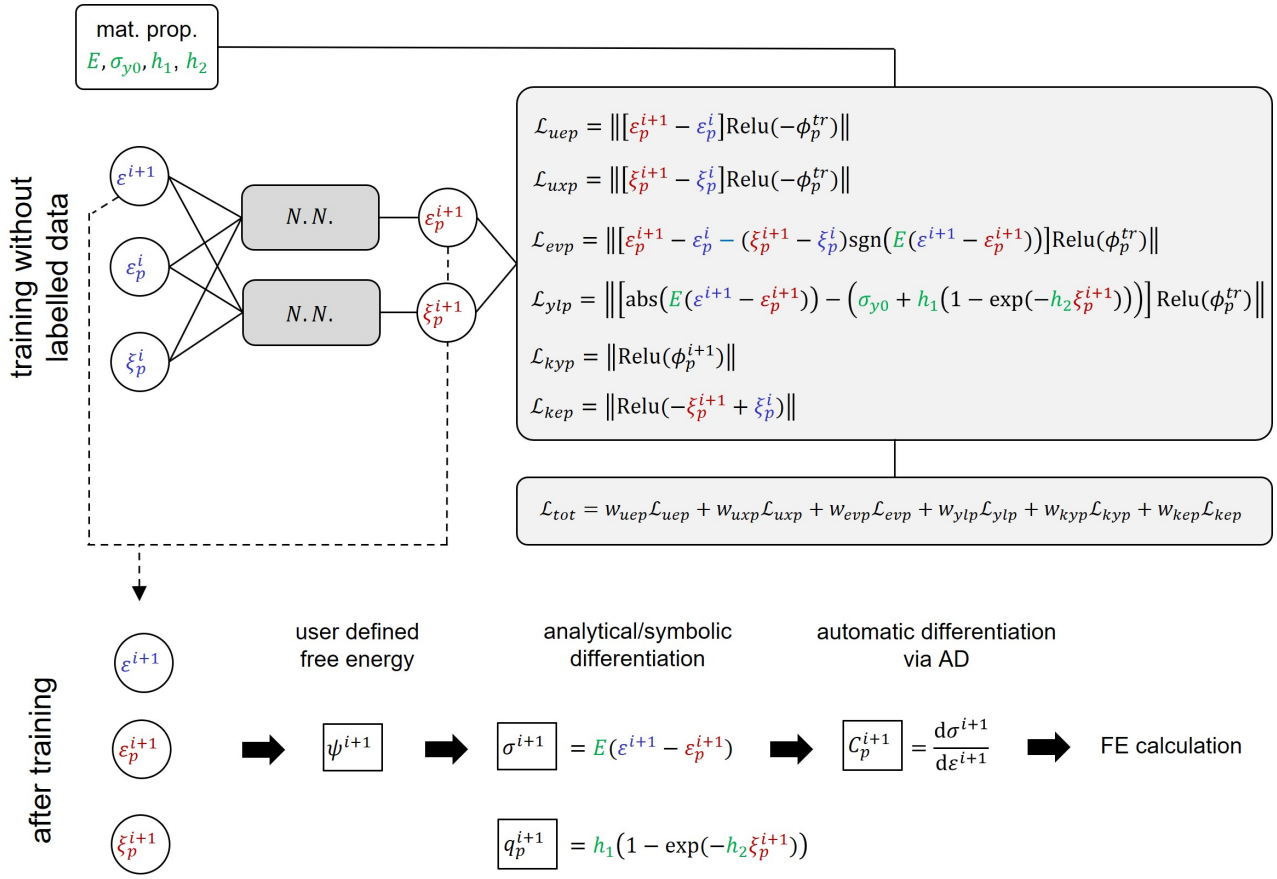


图3：用于采用各向同性 Voce 型硬化定律的弹塑性材料模型的 COMM-PINNs 网络架构与损失函数。

注释2 尽管本研究中材料属性保持不变，但也可以将其作为额外的输入参数。这样，神经网络便能学习给定的广泛材料属性范围内弹塑性问题的解。参见 [46, 65]关于利用迁移学习纳入材料属性影响的研究。

注释3 在上述损失函数中，可以按如下形式加入数据项的贡献： $\mathcal{L}_{data}$ = 均方误差 $\left(\varepsilon_p^{i+1} - \varepsilon_{alg}^{i+1}\right)$ + 均方误差 $\left(\xi_p^{i+1} - \xi_{alg}^{i+1}\right)$，其中下标 alg 表示来自返回映射算法的数据。我们将在结果部分考察并比较此类网络设计。

注释4 可以通过解析导数推导切线算子的各项。其优点在于，可以代入当前的应变值和其他状态变量来求得这些量，而不会遇到梯度消失的问题。其缺点是获取解析导数可能存在困难。在微分较为复杂的情况下，也可以利用参数微分方面的最新进展来避免繁琐的推导。

4. 界面开裂损伤模型案例研究

4.1. 模型推导

间隙值 g 用于构建界面能和界面力学。间隙变量与位移跳跃 $\langle \boldsymbol{u} \rangle$ 或界面两相对侧之间的距离相关。在I型张开情况下，我们将法向间隙变量写为

$$g = \langle \boldsymbol{u} \rangle \cdot \boldsymbol{n}, \tag{28}$$

其中， $\boldsymbol{n}$ 是界面的法向量。界面处的损伤由标量参数 d 表示，该参数是一个内变量，用于描述黏结力的损失。（法向）牵引参数 T 与间隙值 g 及其他内变量（如损伤 d）相关。与体材料的应力-应变行为类似，尖锐界面的力学行为遵循牵引-分离关系。界面处的亥姆霍兹自由能 ψ 由弹性部分（ψ_e）和损伤硬化（ψ_d）组成，因此总能量可表示为：

$$\psi(g, d, \xi_d) = \psi_e(g, d) + \psi_d(\xi_d), \tag{29}$$

$$\psi_e\left(\varepsilon, \varepsilon_p\right) = \frac{1}{2} f_{\mathrm{d}}(d)\ K\ g^2, \tag{30}$$

$$\psi_d\left(\xi_d\right) = h_1 \left(\xi_d + \frac{e^{-h_2 \xi_d} - 1}{h_2} \right), \tag{31}$$

这里， ξ_d 是损伤硬化变量。此外， h_1 和 h_2 是界面损伤硬化参数，用于控制软化行为 [62]。参数 K 是界面的初始刚度。此外， f_{d}（d）是一个正标量值，对于未损伤界面，其值为 1；对于完全损伤界面，其值接近临界值 0。该函数的一种常见形式如下

$$f_{\mathrm{d}}(d) = (1 - d)^2. \tag{32}$$

通过以克劳修斯-迪昂不等式的形式应用热力学第二定律，推导出模型的一致状态关系。共轭力如下。

$$T = \partial_g \psi = \partial_g \psi_e = f_{\mathrm{d}}(d) K\ g, \tag{33}$$

$$Y = -\partial_d \psi = -\partial_d \psi_e = -\frac{\mathrm{d}f_d}{\mathrm{d}d} \frac{1}{2} K g^2, \tag{34}$$

$$q_d = \partial_{\xi_d} \psi = \partial_{\xi_d} \psi_d = h_1 (1 - e^{-h_2 \xi_d}). \tag{35}$$

与塑性类似，损伤屈服准则定义为

$$\phi_d(Y, q_d) = Y - \left(Y_0 + q_d\right). \tag{36}$$

在该模型中，损伤起始点由界面参数 Y_0 控制，而q_d 则描述非线性损伤硬化。损伤内部变量的演化定律为：

$$\dot{d} = \dot{\lambda}_d\ \frac{\partial \phi_d}{\partial Y} = \dot{\lambda}_d, \tag{37}$$

$$\dot{\xi}_d = -\dot{\lambda}_d\ \frac{\partial \phi_d}{\partial q_d} = \dot{\lambda}_d. \tag{38}$$

最后，考虑损伤模型的加载/卸载条件：

$$\dot{\lambda}_d \geq 0, \quad \phi_d \leq 0, \quad \dot{\lambda}_d\ \phi_d = 0. \tag{39}$$

4.2. 损伤模型的实现与算法方面

与塑性情况类似，算法 2 中的上标 k 表示损伤求解器内部循环中的第 k 次迭代。在该算法中，$\Delta\lambda_d = \Delta t\ \lambda_d$，其中 Δt 为伪时间步。当损伤处于激活状态时（$\phi_d > 0$），使用牛顿-拉夫森方法对损伤残差（$r_d^{(1)}$ 和 $r_d^{(2)}$）进行线性化，以求解未知量（d^{k+1} 和 ξ_d^{k+1}）。矩阵 $\boldsymbol{K}_d = \partial \boldsymbol{r}_d / \partial \boldsymbol{U}_d$ 包含损伤残差向量 $\boldsymbol{r}_d$ 相对于未知量（$\boldsymbol{U}_d = d^{i+1},\ \xi_d^{i+1}$）的导数。如果未知内部变量的解的变化小于某个容差（$\text{tol} = 10^{-10}$），则认为结果已收敛，并用于计算界面处的牵引力值和切线算子。这些量用于在有限元层面 [62]建立全局残差向量。

算法 2 求解伪时间 t^{i+} 下的损伤控制方程

Input: 界面的 arameters 当前 a: $^{i+1}$g历史变量: d^i, ξ_d^i

Output: current traction: T^{i+1}, current history variables d^{i+1}, ξ_d^{i+1}, ta切线: $C_d = \mathrm{d}T^{i+1}/\mathrm{d}g^{i+1}$

$k = 1$
Set trial values: $g_p^k = g_p^i$, $d^k = \xi_d^i = d^i$
loop *damage solver* **do**
    if $\phi_d^{tr} = K(g^{i+1})^2 - (Y_0 + q_d(\xi^i)) \leq 0$
        $d^{i+1} = d^i$, $\xi_d^{i+1} = \xi_d^i$ and $C_d = (1 - d^{i+1})^2 K$
    else solve for d^{k+1} and $\xi_d^{k+1} = \xi_d^i + \Delta\lambda_d^{k+1}$ (see Eqs 36,37)
        $r_d^{(1)} = d^{k+1} - d^i - (\xi_d^{k+1} - \xi_d^i)\dfrac{\partial \phi_d}{\partial Y} \stackrel{!}{=} 0$
        $r_d^{(2)} = \phi_d = Y - \left(Y_0 + q_d\left(\xi_d^{i+1}\right)\right) \stackrel{!}{=} 0$
        $[d^{i+1,k+1} \quad \xi_d^{i+1,k+1}]^T = [d^{i+1,k} \quad \xi_d^{i+1,k}]^T - \boldsymbol{K}_d^{-1}\boldsymbol{r}_d$
    end if
    if $\left|d^{i+1,k+1} - d^{i+1,k}\right| > \text{tol}$ **OR** $\left|\xi_d^{i+1,k+1} - \xi_d^{i+1,k}\right| > \text{tol}$
        $k++$
        CYCLE loop *damage solver*
    else
        $d^{i+1} = d^{i+1,k+1}$, $\xi_d^{i+1} = \xi_d^{i+1,k+1}$
        EXIT loop *damage solver*
    end if
end loop
Compute traction $T = (1 - d^{i+1})^2 K g^{i+1}$
Compute tangent C_d

4.3. 用物理信息神经网络求解　　用于局部损伤模型或界面处的非线性软化行为

对于局部损伤演化（界面断裂）的情况，输入层包括间隙值 g^{i+1}、前一时刻的损伤值 d^i 以及损伤硬化变量 ξ_d^i。我们用向量 $\boldsymbol{X} = \{g^{i+1}, d^i, \xi_d^i\}$ 表示输入层。输出层包括 $\boldsymbol{Y} = \{d^{i+1}, \xi_d^{i+1}\}$。与前一种情况类似，我们打算为每个输出变量分别使用全连接前馈神经网络（另见图 4）。通过神经网络进行本构材料建模得到的结果表示为

$$d^{i+1} = \mathcal{N}_d(\boldsymbol{X}; \boldsymbol{\theta}), \qquad \xi_d^{i+1} = \mathcal{N}_{\xi_d}(\boldsymbol{X}; \boldsymbol{\theta}). \tag{40}$$

接下来，我们为所引入的损伤模型构建残差。这里需要构建损伤屈服函数以及内部变量的演化方程。这些表达式将用于构造神经网络的损失函数。将损伤的总损失项之和记为 $\mathcal{L}_{dt}$，根据算法 2，其定义为

$$\mathcal{L}_{dt} = \underbrace{w_{ued}\mathcal{L}_{ued} + w_{uxd}\mathcal{L}_{uxd}}_{\text{elastic response}} + \underbrace{w_{evd}\mathcal{L}_{evd} + w_{yld}\mathcal{L}_{yld}}_{\text{damage evolution}} + \underbrace{w_{ked}\mathcal{L}_{ked} + w_{kyd}\mathcal{L}_{kyd}}_{\text{KKT conditions}} \tag{41}$$

在方程（41）中，不同的损失项涵盖了所有可能的加载、卸载和再加载场景。前两项（$\mathcal{L}_{ued}$ 和 $\mathcal{L}_{uxd}$）确保当试屈服函数（ϕ_d^{tr}）为负时，损伤不会演化。一旦 $\phi_d^{tr} > 0$，演化定律的主要损伤残差以及当前屈服函数便会激活，分别记为 $\mathcal{L}_{evd}$ 和 $\mathcal{L}_{yld}$。最后，为确保始终满足 KKT 条件，我们还设置了最后两个损失项（$\mathcal{L}_{ked}$ 和 $\mathcal{L}_{kyd}$）。以下汇总了所有相关损失项

$$\mathcal{L}_{ued} = \text{MSE}\left((d^{i+1} - d^i)\text{Relu}(-\phi_d^{tr})\right), \tag{42}$$

$$\mathcal{L}_{uxd} = \text{均方误差}\left((\xi_d^{i+1} - \xi_d^i)\text{ReLU}(-\phi_d^{tr})\right), \tag{43}$$

$$\mathcal{L}_{evd} = \text{MSE}\left((d^{i+1} - d^i - (\xi_d^{i+1} - \xi_d^i))\text{Relu}(\phi_d^{tr})\right), \tag{44}$$

$$\mathcal{L}_{yld} = \text{MSE}\left(\left((1 - d_{i+1}^{i+1})K(g^{i+1})^2 - (Y_0 + h_1(1 - \exp(-h_2\xi_d^{i+1}))\right)\text{Relu}(\phi_d^{tr})\right), \tag{45}$$

$$\mathcal{L}_{ked} = \text{均方误差}\left(\text{ReLU}(\phi_d^{i+1})\right), \tag{46}$$

$$\mathcal{L}_{kyd} = \text{均方误差}\left(\text{ReLU}(-\xi_d^{i+1} + \xi_d^i)\right). \tag{47}$$

所谓的试屈服函数 ϕ_d^{tr} 通过输入层的量进行计算（另见算法 2）：

$$\phi_d^{tr} = (1 - d^i)Kg^{i+1} - \left(Y_0 + h_1(1 - \exp(-h_2\xi_d^i))\right). \tag{48}$$

图 4 总结了具有非线性软化行为的界面断裂模型的主要损失项。为清晰起见，所有输入变量均以蓝色表示，而输出（未知）变量则以红色表示。请注意，材料（界面）属性保持不变，并以绿色表示。

训练完成后，网络会预测未知的内部变量（即 d^{i+1} 和 ξ_d^{i+1}）。有了更新后的内部变量以及当前间隙值（g^{i+1}），便可构造自由能函数和牵引力（T^{i+1}）。请注意，牵引力函数是通过对能量函数关于间隙进行解析微分得到的。最后，切线算子定义为当前牵引力关于当前间隙值的全导数，并通过自动微分获得。

最终损失项在每一个配点处均达到最小。该数学优化问题可写为

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathcal{L}_{dt}(\boldsymbol{X}; \boldsymbol{\theta}), \tag{49}$$

其中 $\boldsymbol{\theta}^*$ 是网络的最优可训练参数（权重和偏置）

k.

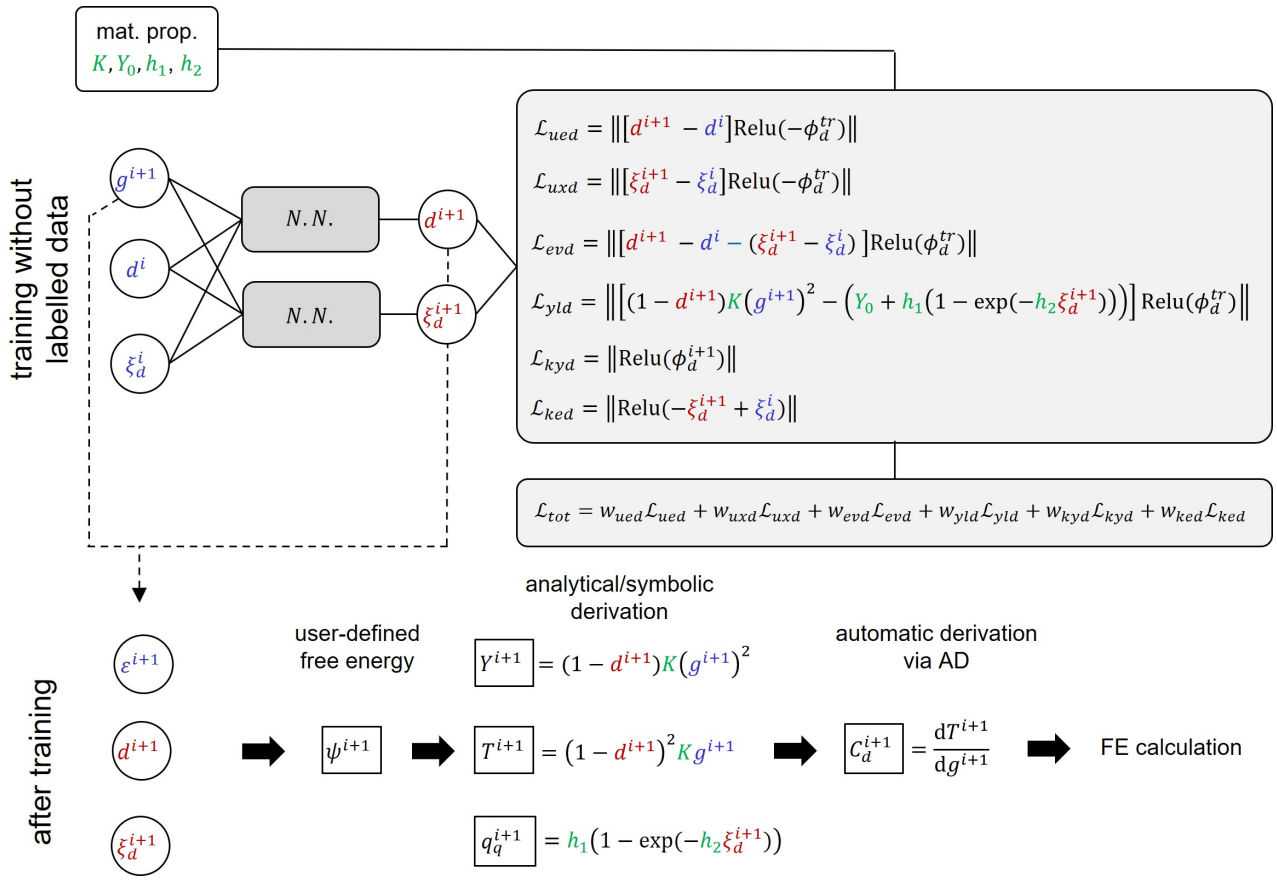


图 4：COMM-PINN 应用于具有非线性软化定律的界面失效问题时的网络架构和损失函数。

5. 配点的生成

配点的位置对训练 [19, 46]至关重要。配点是需要满足（收集）控制方程的初始输入。配点的位置可以不同于数据点，对于这些数据点，我们预先知道其解。神经网络的输入包括给定的第 $i+1$ 步应变或间隙（即 ε^{i+1} 或 g^{i+1}），以及前一时间步 i 的状态变量（即 ε_p^i 和 ξ_p^i 或 d^i 和 ξ_d^i）。

在此方法中，我们提出为输入集合指定可接受的数值。不过，生成这些点时应考虑两个方面：1）输入的目标范围；2）指定随机数值时避免违反重要的物理规律。例如，在弹塑性问题中，累积塑性应变始终为正，且大于塑性应变。同样，在界面损伤和局部损伤模型中，损伤变量和损伤硬化变量始终为正、小于 1，且二者相等。这些限制通常广为人知，也很容易纳入考虑。然而，对于更复杂的材料模型，未来的开发工作将需要采用更简单的策略。必须确保生成的数值符合物理规律且处于目标范围内，以避免训练过程中出现问题，并提高模型的准确性。

另一种思路是生成随机加载路径，从材料模型中收集数据作为初始值，并用训练好的神经网络替代这些模型。在当前工作中，我们尝试了这两种方案，并得出结论：采用这两种方法中的任一种，都能够

求解本构材料方程。为简单起见，我们按照以下算法所述，报告生成配点的更直接方法。

对于弹塑性行为，请读者参阅算法 6.3.5。这里，我们使用以下区间来生成配点： $bg_e = 0$, $st_e = 0.01$, $en_e = +1$, $bg_{\varepsilon_p} = 0$,$st_{\varepsilon_p} = 0.01$, $en_{\varepsilon_p} = +1$, $bg_{\xi_p} = 0$, $st_{\xi_p} = 0.01$ 和 $en_{\xi_p} = +1$。算法 6.3.5 的主要输出是存储在矩阵**输入**$_p$ 中的配点位置，该矩阵有三列，分别对应神经网络的三个主要输入。行数取决于用户对区间和步长的选择。

算法 3 用于求解弹塑性本构关系的配点生成输入：步长： st**，区间起点** bg **和终点** en
输出：用于训练的可容许输入集 IN$_p$**1:** $n = 1$**2: 初始化：** $\boldsymbol{\varepsilon} = [bg_{\varepsilon} : st_{\varepsilon} : en_{\varepsilon}]$, $\boldsymbol{\varepsilon}_p = [bg_{\varepsilon_p} : st_{\varepsilon_p} : en_{\varepsilon_p}]$, $\boldsymbol{\xi}_p = [bg_{\xi_p} : st_{\xi_p} : en_{\xi_p}]$**3: 对** $\boldsymbol{\varepsilon}$ **中的 l 执行4: 对** $\boldsymbol{\varepsilon}_p$ **中的 j 执行 5: 对** $\boldsymbol{\xi}_p$ **中的 k 执行6:** $\mathbf{IN}_p[n, 1] = \boldsymbol{\varepsilon}[l]$**7:** $\mathbf{IN}_p[n, 2] = \boldsymbol{\varepsilon}_p[j]$**8: 如果** $\boldsymbol{\xi}_p[k] \geq \boldsymbol{\varepsilon}_p[j]$**9: IN** $_p[n, 3] = \boldsymbol{\xi}_p[k]$**10:** $n++$**11: 结束如果**

关于界面损伤模型，请读者参阅算法 4。我们使用以下区间生成配点： $bg_g = 0$, $st_g = 0.02$, $en_g = +1$, $bg_d = 0$,$st_d = 0.02$, $en_d = +1$, $bg_{\xi_d} = 0$, $st_{\xi_p} = 0.02$ 和 $en_{\xi_d} = +1$。算法 4 的主要结果是用于训练损伤模型的配点位置，这些位置存储在矩阵 $\mathbf{IN}_d$ 中。

算法 4 用于求解界面损伤模型的配点生成算法 输入：步长： st**，区间的起点** bg **和终点** en **输出：用于训练的可容许输入集 IN**$_d$**1：** $n = 1$**2：初始化：** $\boldsymbol{g} = [bg_g : st_g : en_g]$, $\boldsymbol{d} = [bg_d : st_d : en_d]$, $\boldsymbol{\xi}_d = [bg_{\xi_d} : st_{\xi_d} : en_{\xi_d}]$**3：对于** $\boldsymbol{g}$ **中的 l，执行 4：对于** $\boldsymbol{d}$ **中的 j，执行 5：对于** $\boldsymbol{\xi}_d$ **中的 k，执行 6：** $\mathbf{IN}_d[n, 1] = \boldsymbol{g}[l]$**7：** $\mathbf{IN}_d[n, 2] = \boldsymbol{d}[j]$**8：如果** $\boldsymbol{\xi}_d[k] = \boldsymbol{d}[j]$**9：** $\mathbf{IN}_d[n, 3] = \boldsymbol{\xi}_d[k]$**10：** $n++$**11：结束如果**

6. 结果

当前工作中开发的算法在 SciANN 软件包 [66] 中实现，该方法也可以轻松移植到其他编程平台。所有报告的结果均采用 Adam优化器。我们先介绍损伤模型的结果，因为与塑性情况相比，该模型更为简单。由于这两个模型有一些相似之处，为避免不必要的重复，大多数超参数研究仅针对其中一个模型进行报告。

6.1. 界面损伤模型案例研究

用于学习界面处损伤模型的相关网络参数汇总于表1中。其中一些参数是在广泛的（超）参数研究后获得的。为使其影响对读者更加清晰，我们将改变这些数值和函数，以研究其对所得结果的影响。

表1：损伤模型的 COMM-PINN 网络参数汇总。

参数	数值
输入，输出	$\{g^{i+1}, d^i, \xi_d^i\}$, $\{d^{i+1}, \xi_d^{i+1}\}$
激活函数	ReLU，双曲正切，Sigmoid
层数及每层的神经元数（L, N_l）	（5, 10），(5, 50)，(5, 100)
批量大小	500
学习率 α，训练轮数	(10^{-4}，10_3)

表2中报告的材料参数经过选择，以使应力值低于数值1；此时，它们不一定代表任何真实材料。不过，可以基于其他一些测量值（见 [62]）校准模型，并将模型变量归一化，使其保持在 -1 到 1 之间。正如我们所验证的，对于其他选定的材料属性，所提出方法的性能并未发生变化。

表2：第4.3节所述界面损伤模型的材料参数。

	Unit	数值
法向界面刚度 K	[兆帕/毫米]	5.0
损伤起始准则 Y_0	[兆帕 毫米]	0.1
损伤硬化参数 h_1	[兆帕 毫米]	2.0
损伤硬化参数 h_2	[1/毫米]	1.0×10^2

图5展示了各损失项随训练周期变化的情况。训练所用的配点基于上一节的内容。起初，我们对所有损失项赋予相同的权重，结果表明这样做得到的神经网络模型预测结果可以接受（即我们有 $w_{ued} = w_{uxd} = w_{evd} = w_{yld} = w_{ked} = w_{kyd} = 1.0$）。此外，对于图5所示的结果，我们采用 ReLU 作为激活函数，并使用 5 个隐藏层，每层包含 100 个神经元。其余参数参照表1设置。所有损失函数都同时衰减，且在 1000 个训练周期后，我们没有观察到任何显著改善。在不同的超参数研究中，我们也观察到了所有损失项的相同响应。

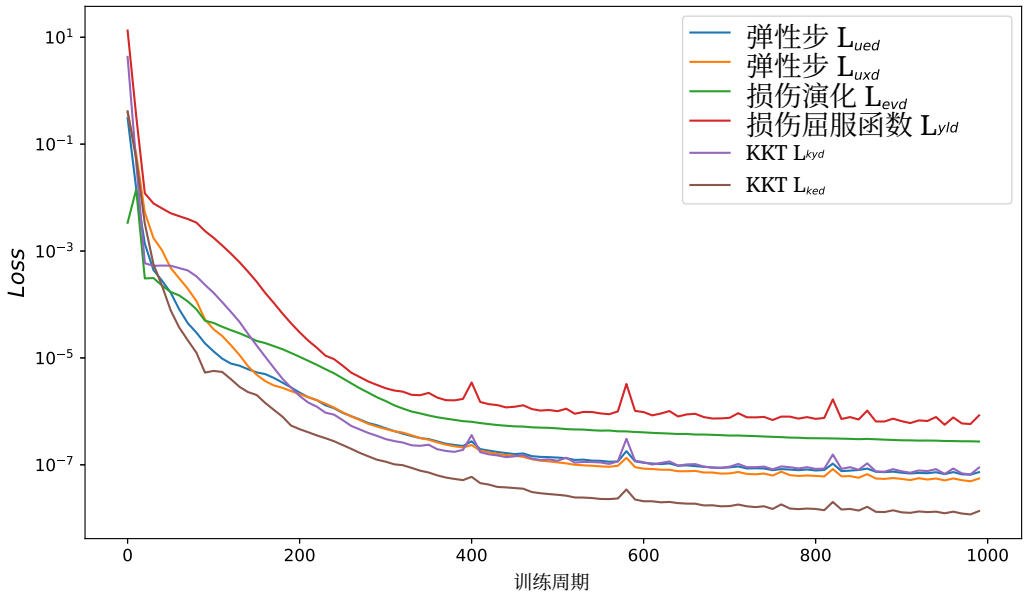


图 5：采用所提出的 COMM-PINN 算法处理具有非线性软化定律的界面损伤模型时，显著的损失项。

接下来，我们将报告训练后的神经网络与基于经典材料模型流程（即返回映射算法）得到的响应之间的性能对比，后者是我们的参考解。由于神经网络并非基于任何给定数据（即已知且可用的解）进行训练，因此，我们为应变值定义的任意加载路径都可以作为测试案例来评估其性能。不过，在介绍结果的过程中，我们会考察神经网络在配点范围之外的表现。图6展示了若干不同加载路径下的一些初步预测结果。第一列表示随时间变化的输入间隙值（即 $g(t)$），第二列表示随时间变化的牵引力（即$T(t)$）。第一行和第二行分别施加了 $g(t) = t$ 和 $g(t) = t^3$。第三行和第四行分别为 $g(t) = 0.5|t\sin(5\pi t)| + 0.5|\sin(2\pi t)|$ 和 $g(t) = 1.0|t\sin(3\pi t)|$。

为了评估神经网络的性能，我们还绘制了预测牵引力数值的相对误差。误差 err 通过逐点比较神经网络T_N给出的牵引力与标准返回映射算法 T_M计算得到的牵引力来计算。因此，我们有 $err = \frac{|T_N - T_M|}{T_M} \times 100$。不同加载场景下的平均误差约为 1%，在某些极端情况下，最大误差会达到 2.5%。本节将讨论改进神经网络预测结果的可能建议。

为避免重复，本文其余部分的所有研究均采用一条较为复杂的加载路径，其中包含多个加载、卸载和再加载场景。因此，我们可以确保所有可能的场景都能被激活。对于以下研究，我们通过下式强制间隙向量随时间变化 $g(t) = 2.0|t\sin(3\pi t)|$。在此加载情况下，间隙将超过 1.5 毫米，而我们训练神经网络时使用的配点间隙值最高仅为 1.0。因此，在加载路径接近末端时，我们检验了训练后神经网络的外推能力。

请注意，即使在没有任何初始数据的物理信息神经网络中，配点的位置也会对最终性能产生重大影响。换句话说，超出配点范围后，无法保证神经网络能够给出令人满意的预测结果。

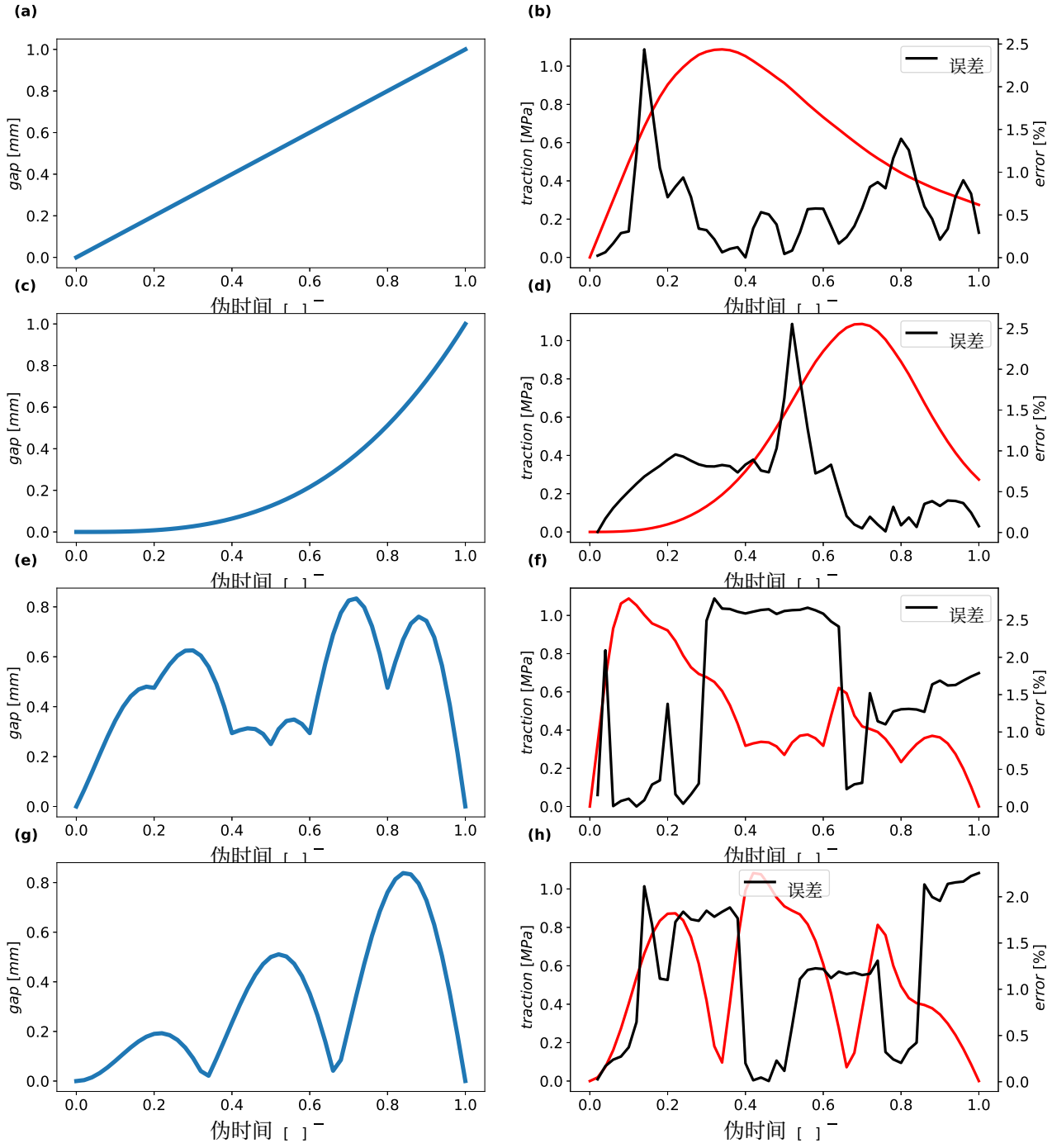


图 6：预测结果的　　采用非线性软化定律的界面损伤模型的训练后 COMM-PINN　　.

6.1.1. 神经网络架构中神经元数量的影响

在图 7 中，我们研究了不同神经网络架构的影响。此处，配点以 0.02 的步长生成，并采用 ReLU 激活函数。通过比较不同的神经网络，我们得出结论：要捕捉所有复杂的非线性行为，需要足够数量的神经元。达到一定程度后，增加神经元或层不再有益，甚至可能导致模型过拟合。层数方面也观察到了类似的性能表现：层数过少时，性能非常差。有趣的是，对于采用了适当架构的红色曲线，

神经网络能够准确捕捉所有加载、卸载和再加载测试案例中的非线性响应。在加载接近结束时，超出训练所提供配点范围的部分，预测响应出现了一些偏差。换言之，在评估超出配点范围的神经网络响应时，即使热力学约束也可能被违反。值得肯定的是，与纯数据驱动方法不同，纯数据驱动方法需要获取新范围内的新数据，而配点的范围和密度始终可以轻松增加。需要注意的是，使用更多配点会增加训练时间。纯物理驱动与纯数据驱动的神经网络训练在损失函数、初始数据集和计算成本的性质上各不相同，因此要进行公平比较并非易事。尽管如此，本文仍将进行一些初步比较，以观察纯数据驱动方法在外推时产生的偏差。

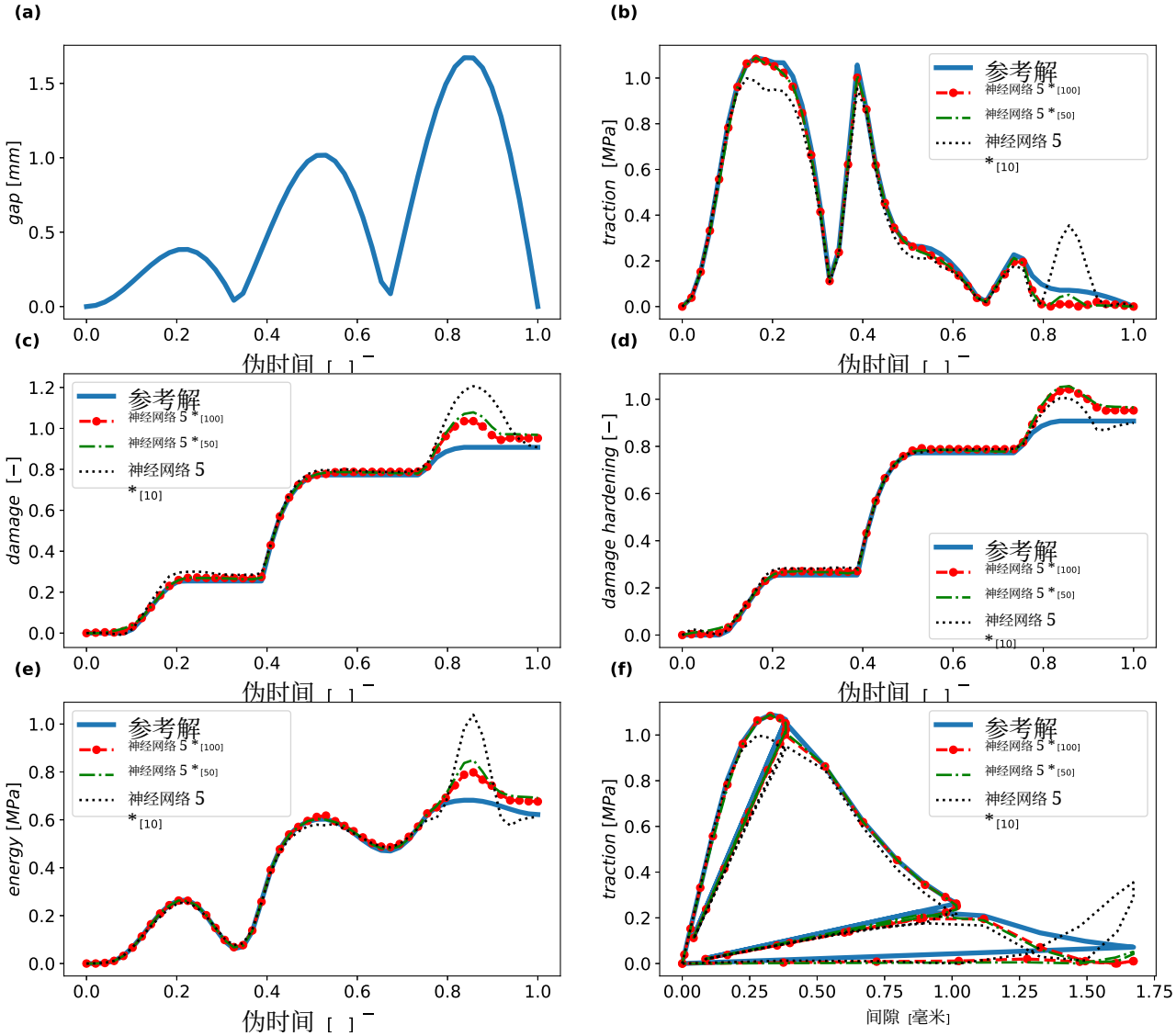


图7：神经网络层对网络预测的影响。(a) 加载（随时间变化的间隙）基于 $g(t) = 2.0|t\sin(3\pi t)|$。(b) 通过训练神经网络预测的随时间变化的牵引力 $T(t)$。(c) 通过训练神经网络预测的随时间变化的损伤变量 $d(t)$。(d) 通过训练神经网络预测的随时间变化的损伤硬化变量 $\xi_d(t)$。(e) 随时间变化的预测自由能函数 $\psi_d(t)$。(f) 预测牵引力与间隙值的关系。

6.1.2. 激活函数的影响

在图 8 中，我们考察了不同激活函数选择对预测结果的影响。初始训练集的说明与前文类似。这里所有案例研究均采用 5 层结构，每层含 100 个神经元。我们得出结论，与其他函数相比，ReLU 函数表现最佳。尽管未在文中报告，采用其他激活函数（如 Swish 和 Softplus）也会得出相同结论。ReLU 函数表现更佳，或许是因为其具有陡峭的过渡特性，能够捕捉卸载、再加载以及屈服条件下行为的突然变化。

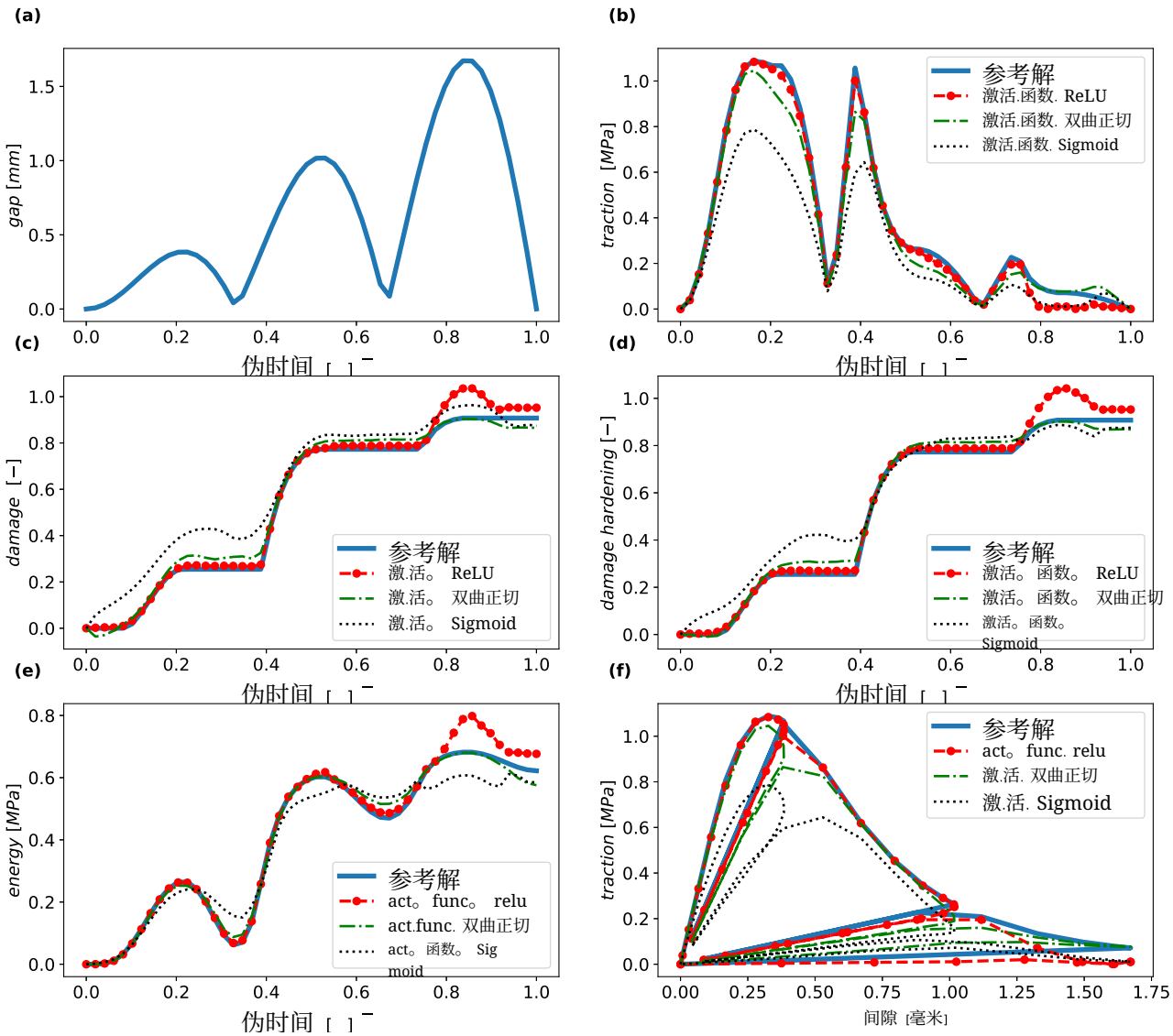


图 8：激活函数选择对网络预测的影响。(a) 加载（间隙随时间的变化）基于 $g(t) = 2.0|t\ \sin(3\pi t)|$。(b) 通过训练神经网络预测的牵引力 $T(t)$ 随时间的变化。(c) 通过训练神经网络预测的损伤变量 $d(t)$ 随时间的变化。(d) 通过训练神经网络预测的损伤硬化变量 $\xi_d(t)$ 随时间的变化。(e) 预测的自由能函数 $\psi_d(t)$ 随时间的变化。(f) 预测的牵引-分离关系。

6.1.3. 配点密度的影响

在图 9 中，我们探讨了配点密度的不同选择对预测结果的影响。本研究选用的激活函数为 ReLU 函数，且

网络有 5 层，每层包含 100 个神经元。配点密度从 0.1 变化到 0.02。我们得出结论，即使配点数量很少，神经网络也能捕捉此类非线性加载路径的整体行为。我们还发现，在训练过程中向神经网络提供更多配点，可以提升神经网络模型的性能。此外，增加配点数量还能提升神经网络的外推能力（见时间 0.8 之后得到的响应）。另一方面，更多配点会显著增加训练的计算成本。步长为 0.02 时的训练耗时是步长为 0.1 时的两倍。可能需要根据配点数量适当调整神经网络的架构。

需要注意的是，在所有情况下，测试加载路径均由 50 个时间点生成（即时间步长为 0.02）。正如我们将在下一节中进一步讨论的那样，可以选择不同的步长并调用同一个神经网络，同时仍能获得令人满意的性能。

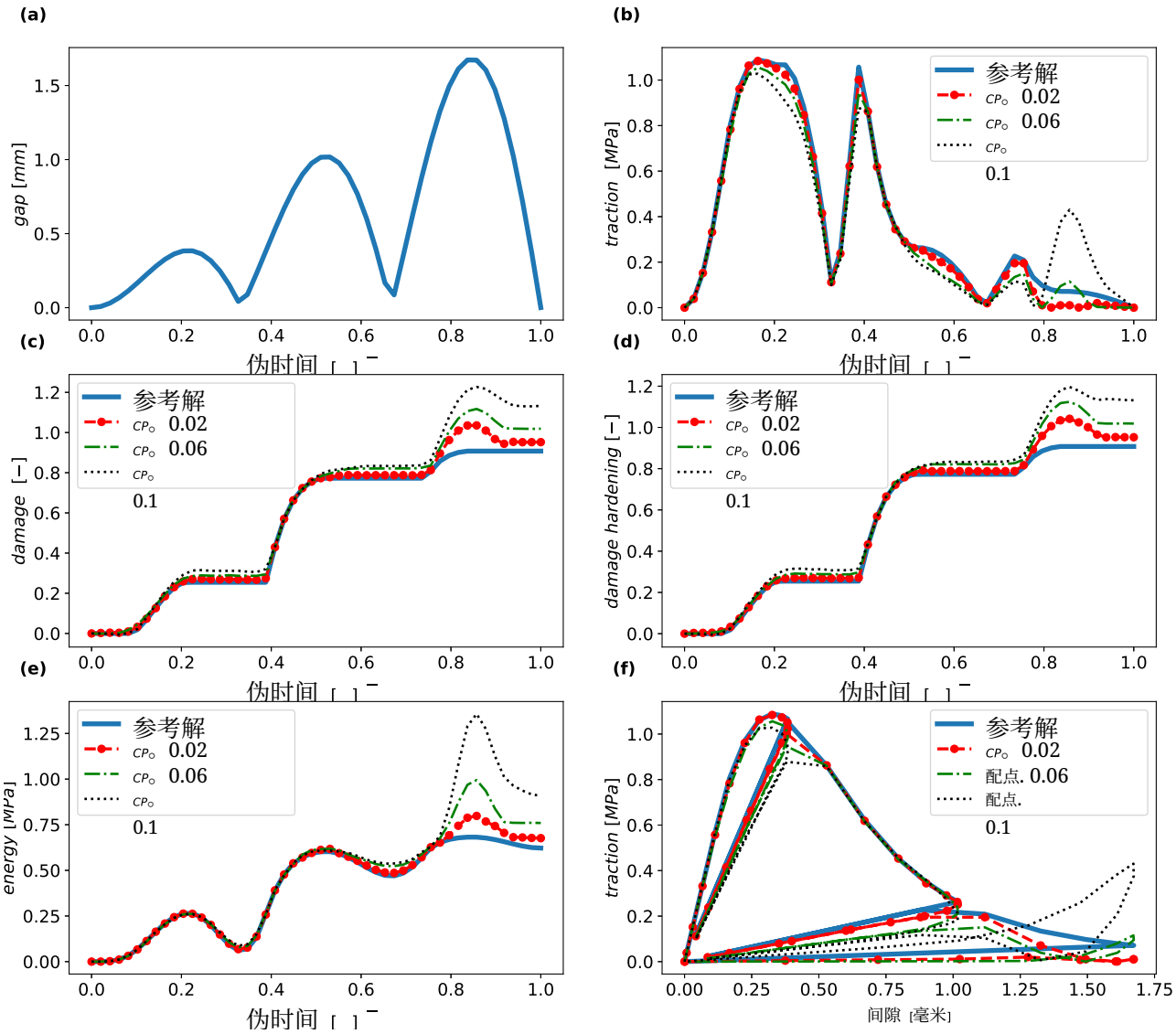


图 9：生成配点所用步长对预测结果的影响。(a) 加载（间隙随时间变化）基于 $g(t) = 2.0|t\ \sin(3\pi t)|$。(b) 通过训练神经网络随时间预测的牵引力 $T(t)$。(c) 通过训练神经网络随时间预测的损伤变量 $d(t)$。(d) 通过训练神经网络随时间预测的损伤硬化变量 $\xi_d(t)$。(e) 随时间预测的自由能函数 $\psi_d(t)$。(f) 预测的牵引力与间隙值的关系。

6.2. 扩展到三维情形：基于各向异性势的内聚区模型

基于 [62] 及其中引用的文献，我们总结三维模型如下。图 10 展示了多晶材料在微观尺度上常见的一种晶间损伤模式。为捕捉这种断裂行为，所有晶粒之间的界面（通常称为晶界）均采用经典内聚区单元进行增强。两个任意表面之间的位移跳跃或间隙向量 $\boldsymbol{g}$ 根据这些表面上内聚区单元节点的位置计算得出。法向 n 和两个剪切方向s_1 、 s_2 根据中平面的位置定义。在三维构型中，间隙向量 $\boldsymbol{g}$ 和牵引向量 $\boldsymbol{t}$ 各有三个分量。因此，我们有：

$$\boldsymbol{g}^T = [g_{s1},\ g_{s1}\ g_n]\,. \tag{50}$$

$$\boldsymbol{t}^T = [t_{s1},\ t_{s1}\ t_n]\,. \tag{51}$$

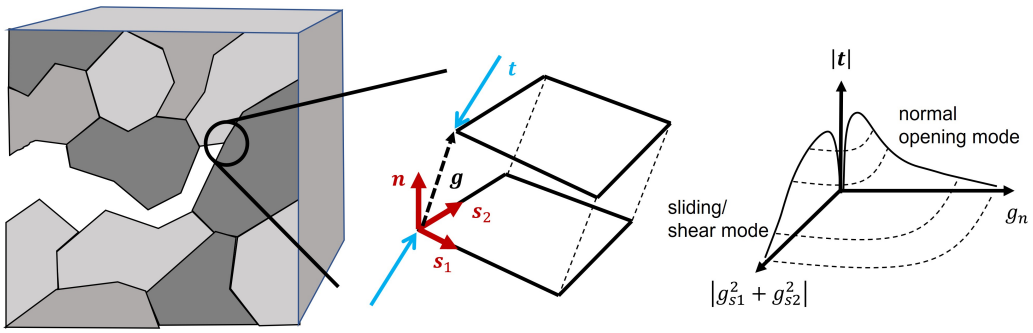


图 10：晶间　　采用三维内聚区单元建模的多晶材料中的 r 型断裂　　.

界面处的损伤被假定为各向同性，因此，引入标量参数 d 作为内变量，以表示特定界面处结合力的损失。界面处的亥姆霍兹自由能 ψ 由两个部分组成：弹性部分（ψ_e）和损伤硬化部分（ψ_d）。这可以表示如下：

$$\psi = \psi_e\left(\boldsymbol{g}, d\right) + \psi_d\left(\xi_d\right) = (1-d)^2\frac{1}{2}(\boldsymbol{g})^T\boldsymbol{K}\boldsymbol{g} + h_1\left(\xi_d + \frac{e^{-h_2\xi_d}-1}{h_2}\right). \tag{52}$$

在上述表达式中， ξ_d 表示损伤（各向同性）硬化变量， h_1 和h_2 是界面损伤硬化参数。参数 $\boldsymbol{K} = K_{s_1}\boldsymbol{s_1}\otimes\boldsymbol{s_1} + K_{s_2}\boldsymbol{s_2}\otimes\boldsymbol{s_2} + K_n\ \boldsymbol{n}\otimes\boldsymbol{n}$ 是表示界面各向异性初始刚度的二阶张量。与界面相关的热力学共轭力如下：

$$\boldsymbol{t} = \partial_{\boldsymbol{g}}\psi = (1-d)^2\boldsymbol{K}\boldsymbol{g}, \tag{53}$$

$$Y = -\partial_d\psi = (1-d)(\boldsymbol{g})^T\boldsymbol{K}\boldsymbol{g} = (1-d)\left(K_{s_1}g_{s_1}^2 + K_{s_2}g_{s_2}^2 + K_n g_n^2\right). \tag{54}$$

屈服函数（方程 36）和演化方程（方程 37）的关系保持不变。此外，损失项的表达式由方程 42 至 47 给出。需要注意的是，对于其他模型，例如三维塑性模型，需要为塑性应变的每个分量定义额外的损失项。不过，这项任务相对直接，因为只需将方程表示为损失项的形式，这与一维示例所采用的方法非常相似。

用于训练三维内聚区模型的网络参数总结于表 3 中。这些参数是通过先前的（超）参数研究确定的。值得注意的是，许多神经网络参数可以从简化的一维分析中直接迁移，从而实现一致且高效的学习过程。有关本研究所采用网络参数的详细概览，请参阅表 3。

表 3：三维内聚区模型的 COMM-PINN 网络参数汇总 .

参数	数值
输入、输出	$\{g_{s1}^{i+1}, g_{s1}^{i+1}, g_{n}^{i+1}, d^{i}, \xi_{d}^{i}\}$, $\{d^{i+1}, \xi_{d}^{i+1}\}$
激活函数	Relu
层数及每层神经元数（L， N_l）	（5，100）
批量大小	50
学习率 α，训练轮数	(10^{-4}，10₃)

表4中列出的材料参数经过精心选择，以捕捉与各种断裂模式相关的各向异性行为。在研究过程中，我们发现，无论选择何种材料属性，所提出方法的表现始终一致，这表明该方法具有稳健性。所选材料参数及其在捕捉系统各向异性行为方面的作用详见表4。

表4：第4.3节所述界面损伤模型的材料参数。

	Unit	数值
法向刚度（I型张开） K_n	[兆帕/毫米]	5.0
剪切刚度 K_{s1} （II型滑移）	[兆帕/毫米]	0.5
剪切刚度 K_{s2} (III型滑移)	[兆帕/毫米]	2.0
损伤起始准则 Y_0	[兆帕 毫米]	0.1
损伤硬化参数 h_1	[兆帕 毫米]	2.0
损伤硬化参数 h_2	[1/毫米]	1.0×10^2

用于训练的配点基于上一节所述的方法。唯一的修改是为两种剪切模式加入额外的配点，以捕捉完整的三维行为。值得注意的是，与屈服准则相关的损失函数需要相应更新。然而，损伤演化方程与上一节相同。另请参见表3。

在图11至图14中，我们展示了训练后的 COMM-PINN 模型与返回映射算法响应的对比结果。在所有结果中，第一列显示间隙向量的分量，表示界面在三维空间中的加载方向。每种加载路径对应的方程均列于图注中，感兴趣的读者可在线查阅相关代码和数据。第二列展示了随时间变化的牵引力分量。此外，还报告并比较了两个模型的损伤值和损伤硬化参数。

在所有案例中，从加载过程开始到结束，神经网络都不会接触任何数据。仅提供初始条件，即材料状态。在每个时间步，神经网络都能够预测解。随后，我们将预测得到的解作为输入提供给神经网络，以预测后续时间步。因此，第一步预测结果中的任何不一致或错误都可能

传播至最后一步。尽管如此，我们仍观察到神经网络在整个加载历程中始终表现出色。

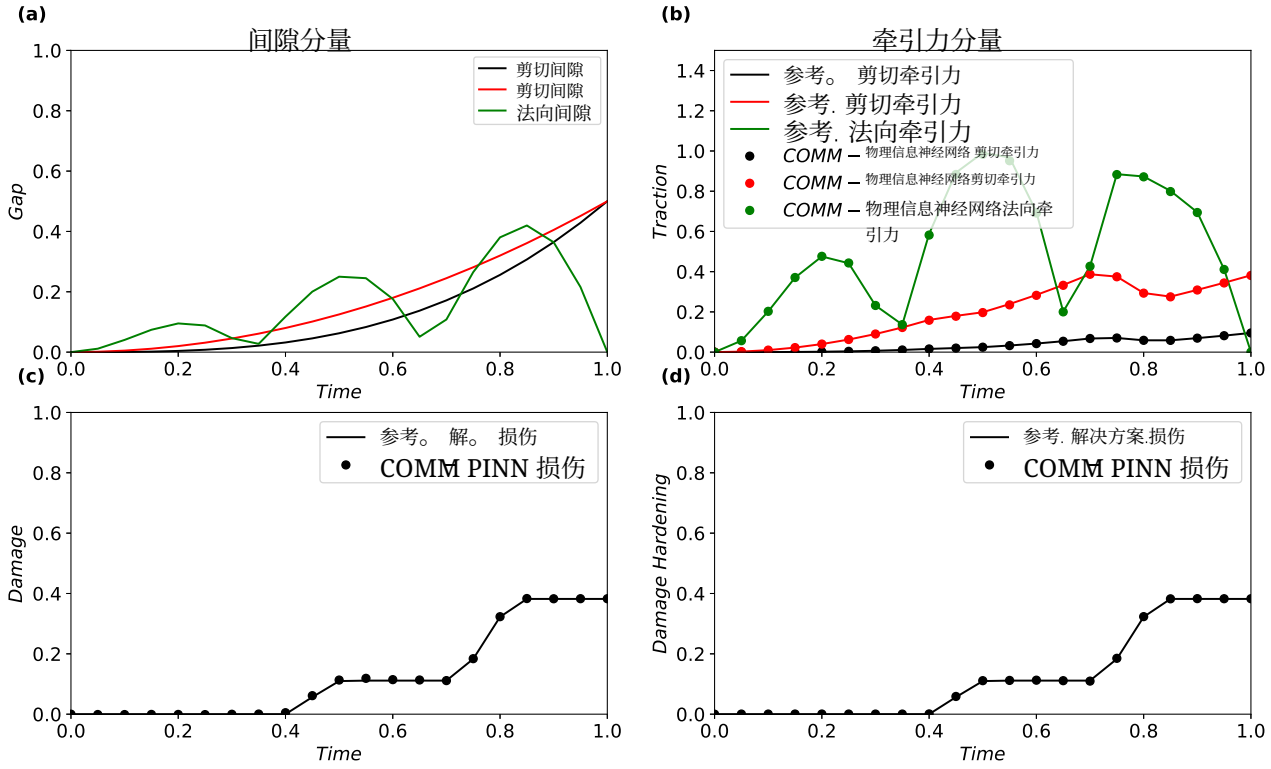


图11：所引入的三维内聚区模型的给定间隙向量和预测牵引向量。间隙向量的分量为 $g_{s1}(t) = 0.5t^3$ 和 $g_{s2}(t) = 0.5t^2$，$g_n(t) = 0.5|t\ \sin(3\pi t)|$。

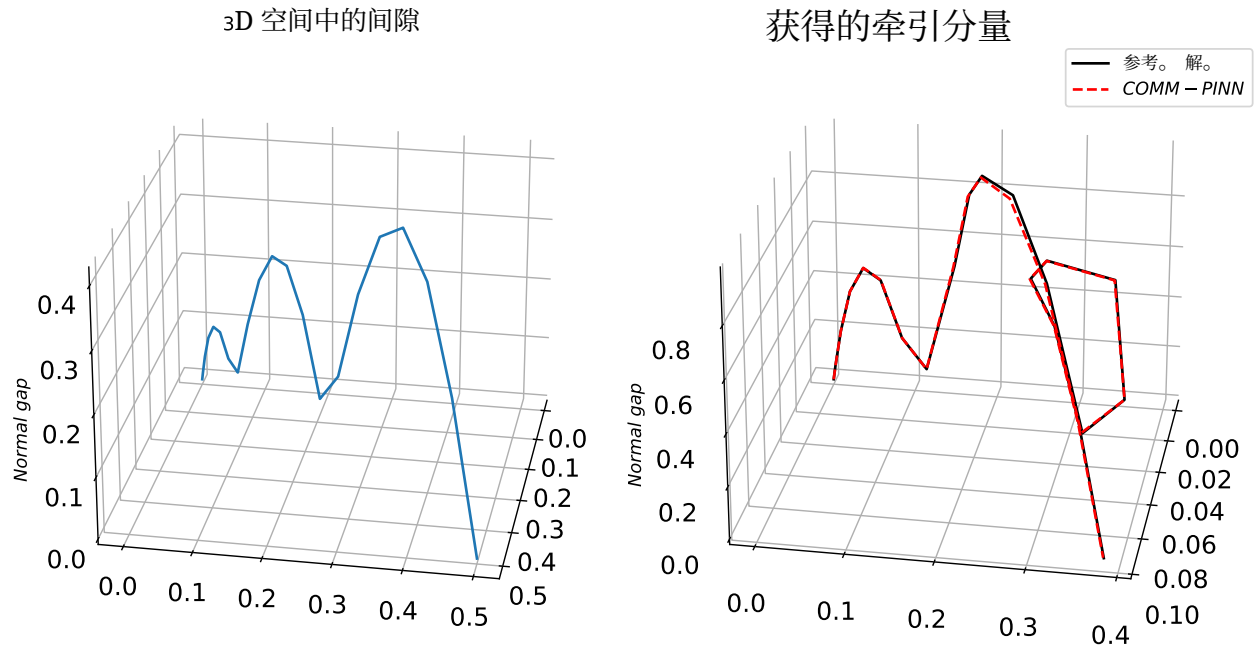


图12：所引入的三维内聚区模型的给定间隙向量和预测牵引向量。间隙向量的分量为 $g_{s1}(t) = 0.5t^3$ 和 $g_{s2}(t) = 0.5t^2$，$g_n(t) = 0.5|t\ \sin(3\pi t)|$。

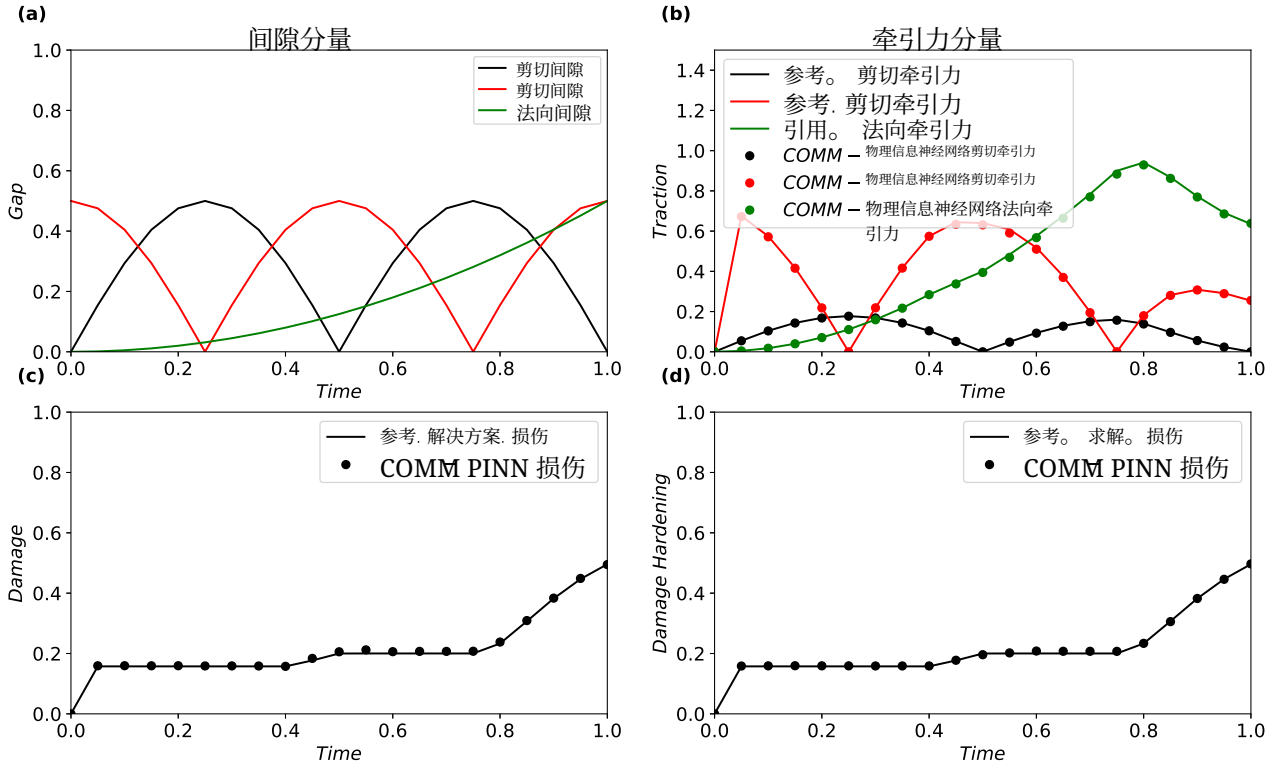


图 13：给定间隙向量以及所提出的三维内聚区模型预测的牵引向量。间隙向量的分量为 $g_{s1} = 0.5|\sin(2\pi t)|$, $g_{s2} = 0.5|\cos(2\pi t)|$, $g_n = 0.5t^2$

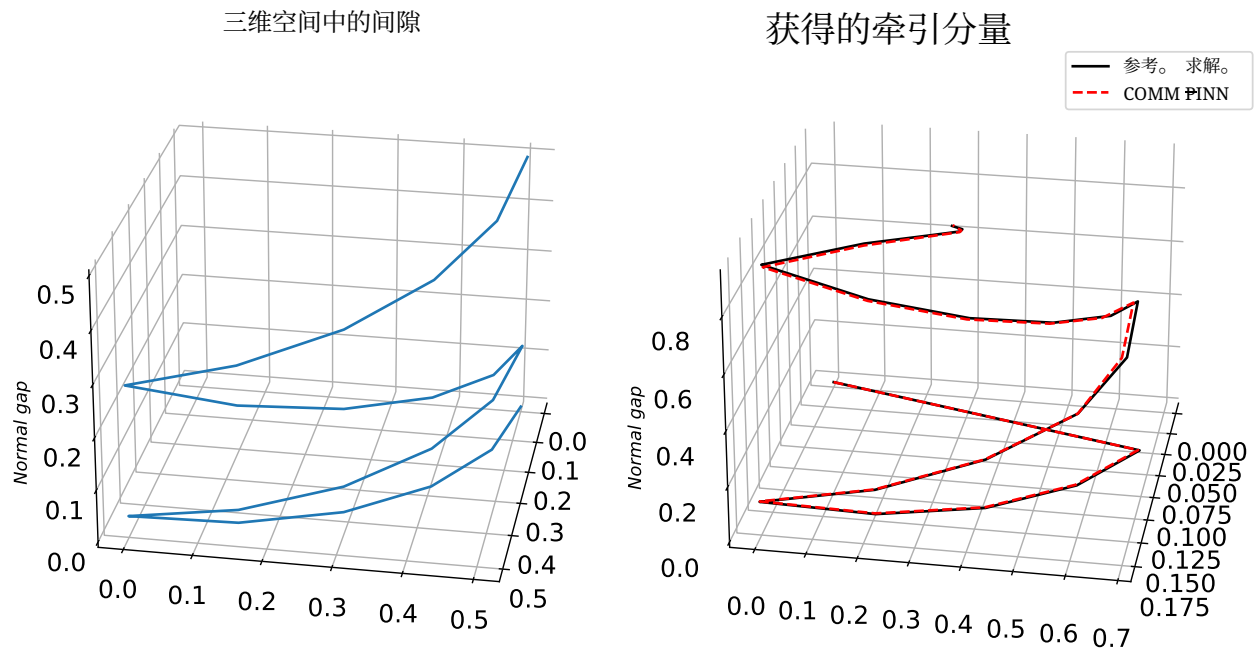


图 14：给定间隙向量以及所提出的三维内聚区模型预测的牵引向量。间隙向量的分量为 $g_{s1} = 0.5|\sin(2\pi t)|$, $g_{s2} = 0.5|\cos(2\pi t)|$, $g_n = 0.5t^2$。

6.2.1. 关于网络评估计算成本的讨论

在此，我们分析三维界面损伤模型，并比较隐式方法（返回映射算法）、显式方法和 COMM-PINN 方法的计算成本。我们进行了多次测试，对每种方法针对一条加载路径最多运行 7 次。平均结果汇总于表 5，表明与隐式方法相比，单条加载路径的计算成本降低了 50% 以上。

值得注意的是，返回映射算法使用 Python 实现，并尽力优化代码以提高效率。训练完成后，通过一段基于标准前馈神经网络算法的简单 Python 代码调用并评估神经网络的权重和偏置（见公式 16）。在本例中，网络架构由 3 层组成，每层有 40 个神经元；此外，另一个神经网络由 2 层组成，每层有 8 个神经元。有趣的是，神经网络训练完成后，评估成本主要取决于架构中的层数和神经元数。换句话说，对于任意给定的加载路径，我们唯一需要执行的任务就是计算公式 16。另请参阅 [55] 中的论点和讨论。

因此，采用类似的架构时，一维和三维情况下神经网络的评估时间几乎相同。事实上，对于三维情况，可能需要增加每层的神经元数量，以赋予神经网络（NN）更大的灵活性。然而，根据我们的研究，即使神经元和层的数量相近，神经网络也能够准确捕捉一维和三维情形下的界面行为。我们确认，将层数从 5 层减少到 3 层或 2 层，也能以同等精度获得所报告的结果。需要注意的是，这一观察结果未必普遍适用于其他类型的材料模型，仍需针对具体情况开展进一步研究。因此，随着维度的增加，COMM-PINN 的优势日益显著。

请注意，我们没有采用丢弃法等额外的先进技术来提高网络效率。这表明，未来的工作应进一步探索神经网络架构的优化，以降低网络评估的计算成本。

最后需要指出的是，在处理更复杂的耦合方程组时，COMM-PINN 的优势会日益明显，因为回映算法往往会遇到收敛困难。例如，用于描述韧性断裂的高度耦合（各向异性）弹塑性损伤模型就存在很大难度。然而，必须强调的是，探索此类情形属于未来研究的范畴，而当前工作应被视为实现这一目标的初步尝试。

表 5：三维界面损伤模型的网络评估与返回映射算法的计算成本比较。

方法	归一化平均运行时间 评估随机加载路径
显式方法（见附录 A）	1.0
隐式方法 / 返回映射算法（见算法 2 和 [62]）	3.4
COMM-PINN（3 层，每层 40 个神经元）	1.8
COMM-PINN（2 层，每层 10 个神经元）	1.2

6.3. 塑性模型案例研究弹塑性案例的相关网络参数总结于表6中。

表6：塑性损伤模型的 COMM-PINN 网络参数汇总。

参数	数值
输入、输出	$\{\varepsilon^{i+1}, \varepsilon_p^i, \varepsilon_p^i\}$, $\{\varepsilon_p^{i+1}, \varepsilon_p^{i+1}\}$
激活函数	Relu
层数和每层的神经元数量 (L, N_l)	(5, 80)
批量大小	100
学习率 α，训练轮数	(10^{-4}, 10^3)

塑性模型有 4 个输入参数： E、 σ_{y0}、 h_1 和 h_2 。弹塑性模型选取的数值汇总于表 7。当前方法也可用于反向分析，并根据给定的数据 [51] 校准这些材料参数。

表 7：第 3.3 节所述塑性模型的材料参数。

	Unit	数值
弹性刚度 E	[兆帕]	3.0
屈服应力 σ_{y0}	[兆帕]	0.6
塑性硬化参数 h_1	[兆帕]	0.4
塑性硬化参数 h_2	$[-]$	10.0

配点的位置与上一节中的说明一致。在报告结果之前，我们先介绍两项旨在提升神经网络模型性能的改进。第一项改进是平衡不同损失函数的数值。第二项改进则是平滑塑性模型中的硬切换条件。

最初，我们为所有损失项设置了相同的权重。这使模型在训练范围内取得了可接受的性能，但当我们在配点范围之外测试神经网络时，效果并不理想。经过一些数值研究，我们认为，这个问题可能是由训练过程中不同损失项之间的不平衡造成的。我们进行了系统的参数研究，并调整不同损失项的权重，使它们的数值更为接近。具体过程如下：首先，我们将所有权重设为相同，然后提高数值较小的损失函数的权重。通过这种方式，我们观察到所得结果有了显著改善。这一思路也与某些编程软件包中采用的自适应加权策略相符。未来的工作还应进一步研究其他自适应加权技术，以找到更易于使用的策略。通过选择 $w_{uep} = 100$, $w_{uxp} = 100$, $w_{evp} = 1$, $w_{ylp} = 10$,$w_{kep} = 100$, $w_{kyp} = 10$，得到了最终最优结果之一。

在下一项修改中，我们放宽了式19至24中对 Relu 和 sgn 函数的条件。因此，我们用 Swish(x) $= x$ Sigmoid(Rx) 替换 Relu(x)，其中 R 决定了过渡区域的平滑程度。最后，我们用 Sigmoid(Rx) 替换 sgn(x)。在本研究中，我们使用了 $R = 300$。为了让读者清楚了解所引入修改的影响，我们也进行了未采用这些修改的研究。图15展示了各损失项的演变，其中我们使用了5个隐藏层，每层包含80个神经元。此外，对于图15所示的结果，我们采用了上述修改。其余参数按照表6设置。大多数损失函数同时衰减，并且在1000个训练周期后，我们没有发现任何显著改善。

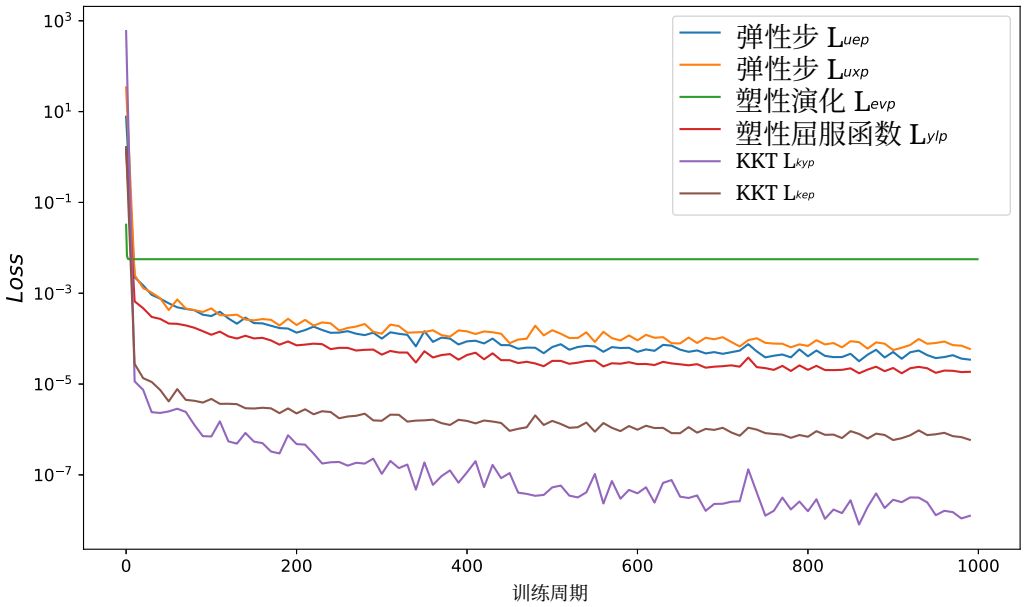


图 15：使用所提出的 COMM-PINN 算法的塑性模型中的主要损失项。

6.3.1. 损失函数中权重的影响

图16所示的研究采用了加载路径 $\varepsilon(t) = 2.0|t\sin(3\pi t)|$。生成该加载路径所用的时间步长为 0.01。材料初始时处于静止状态，且没有任何先前历史。结果表明，神经网络模型准确地捕捉到了屈服点、非线性硬化路径，以及大塑性应变下的应力饱和。需要注意的是，对于累积塑性应变大于 1 的情况，神经网络的响应可视为外推。对于等权重的情况，神经网络在配点范围之外的表现并不可靠。然而，我们观察到，通过调整和平衡损失函数，神经网络能够准确地捕捉配点范围之外的材料行为。

6.3.2. 修改损失函数中的切换选项

在图17中，我们研究了使用 Relu 函数在不同弹性和塑性条件之间进行切换的情况（另见公式19至24）。我们发现，使用该函数更平滑的版本（即 Relu(x) 与 x Sigmoid(Rx)）更有利。研究结果表明，这对于 sgn(x) 中的条件尤为重要，该条件被 Sigmoid(Rx) 取代，其中我们使用 $R = 300$。另见 Haghighat 等人 [51] 的研究。

6.3.3. 时间步长的影响

在此，我们通过使用不同的时间步长求解加载路径来考察输出。选择这些时间步时，确保其与配点不重合。图18第一行中，使用 0.1 的时间步长对神经网络进行评估。除应力-应变曲线外，我们还绘制了仅通过自动微分计算得到的切线算子。对于第二行，我们使用$1/80$ 的步长评估模型，以确保其与配点不重合（配点采用 $1/100$ 的步长生成）。预测结果同样可以接受。然而，当使用更细的时间步（见第三行，步长为 $1/200$）时，我们观察到结果与正确解存在一定偏差，且切线算子出现轻微振荡。作为补救措施，可以在生成配点时始终采用更细的步长，以提高神经网络预测的准确性。

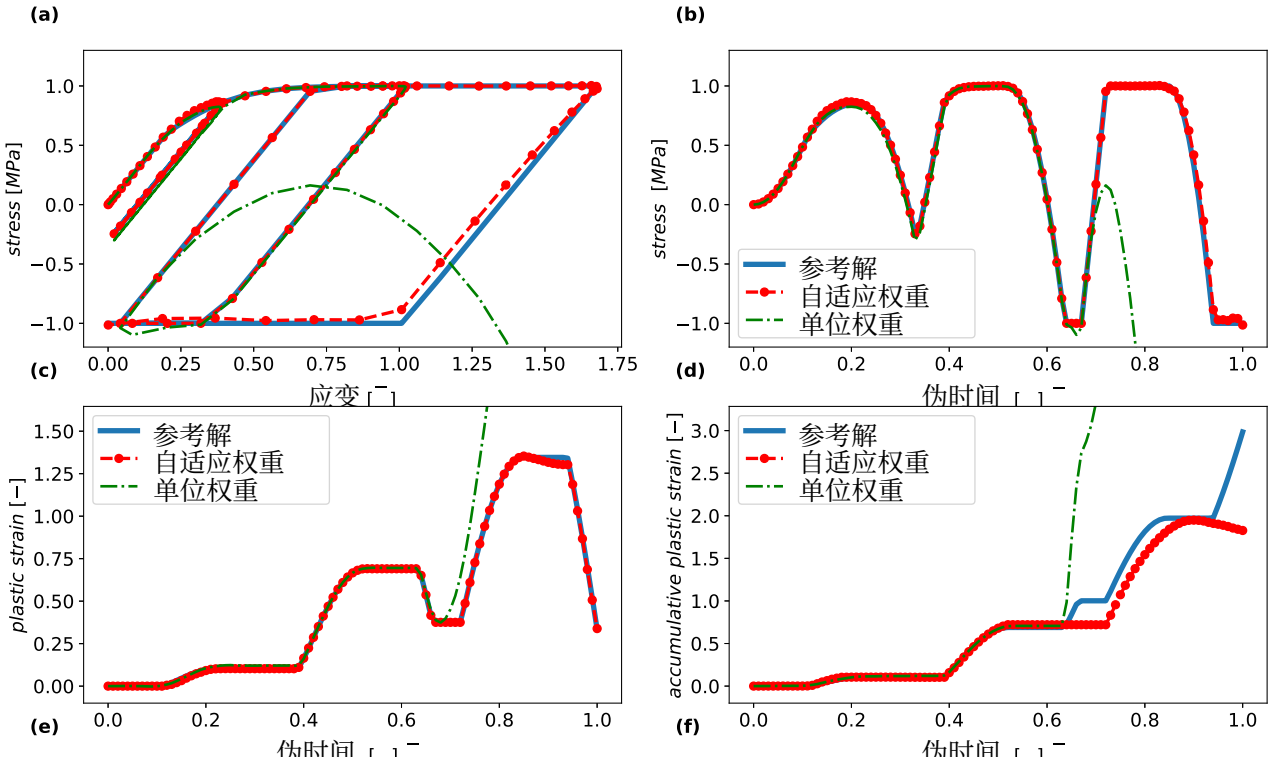


图16：不同损失函数的调整后权重的影响。(a) 应力-应变曲线。(b) 预测应力 $\sigma(t)$。(c) 预测塑性应变 $\epsilon_p(t)$。(d) 预测累积塑性应变 $\xi_p(t)$。

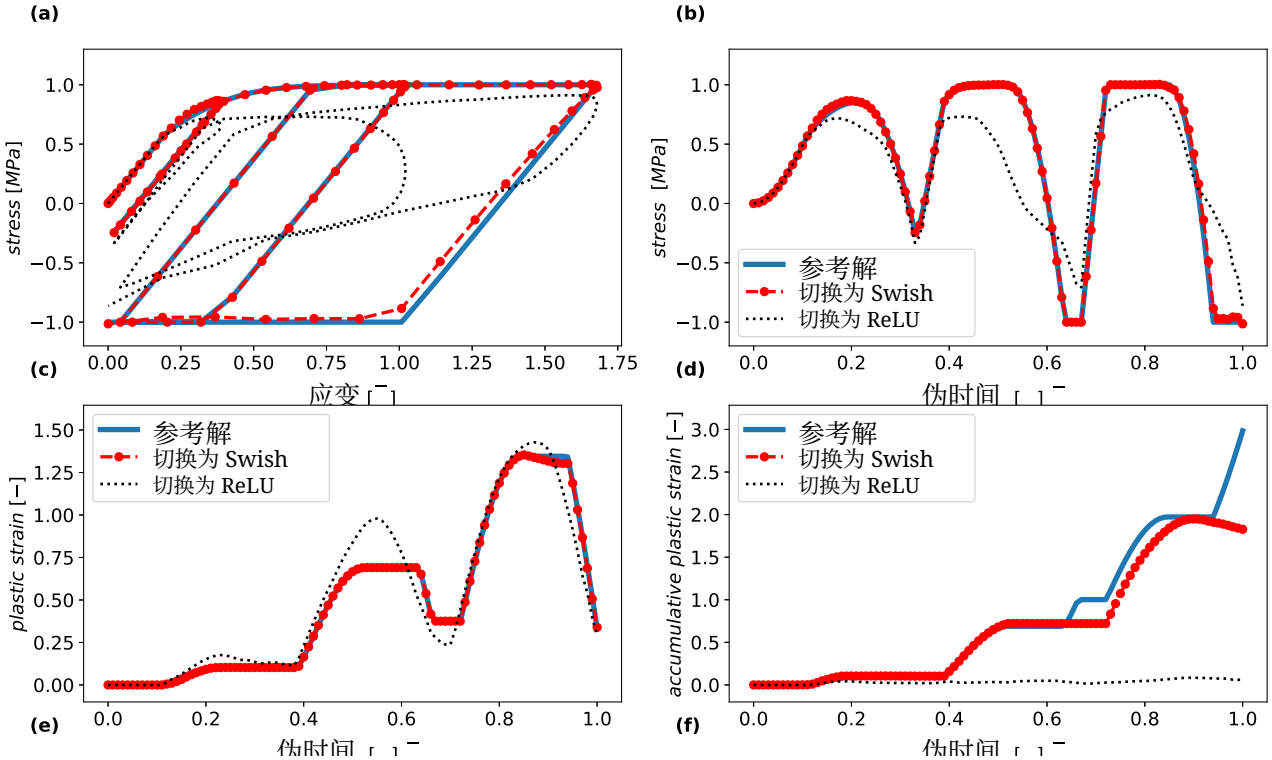


图17：修改损失函数中的切换选项的影响。(a) 应力-应变曲线。(b) 预测应力 $\sigma(t)$。(c) 预测塑性应变 $\epsilon_p(t)$。(d) 预测累积塑性应变 $\xi_p(t)$。

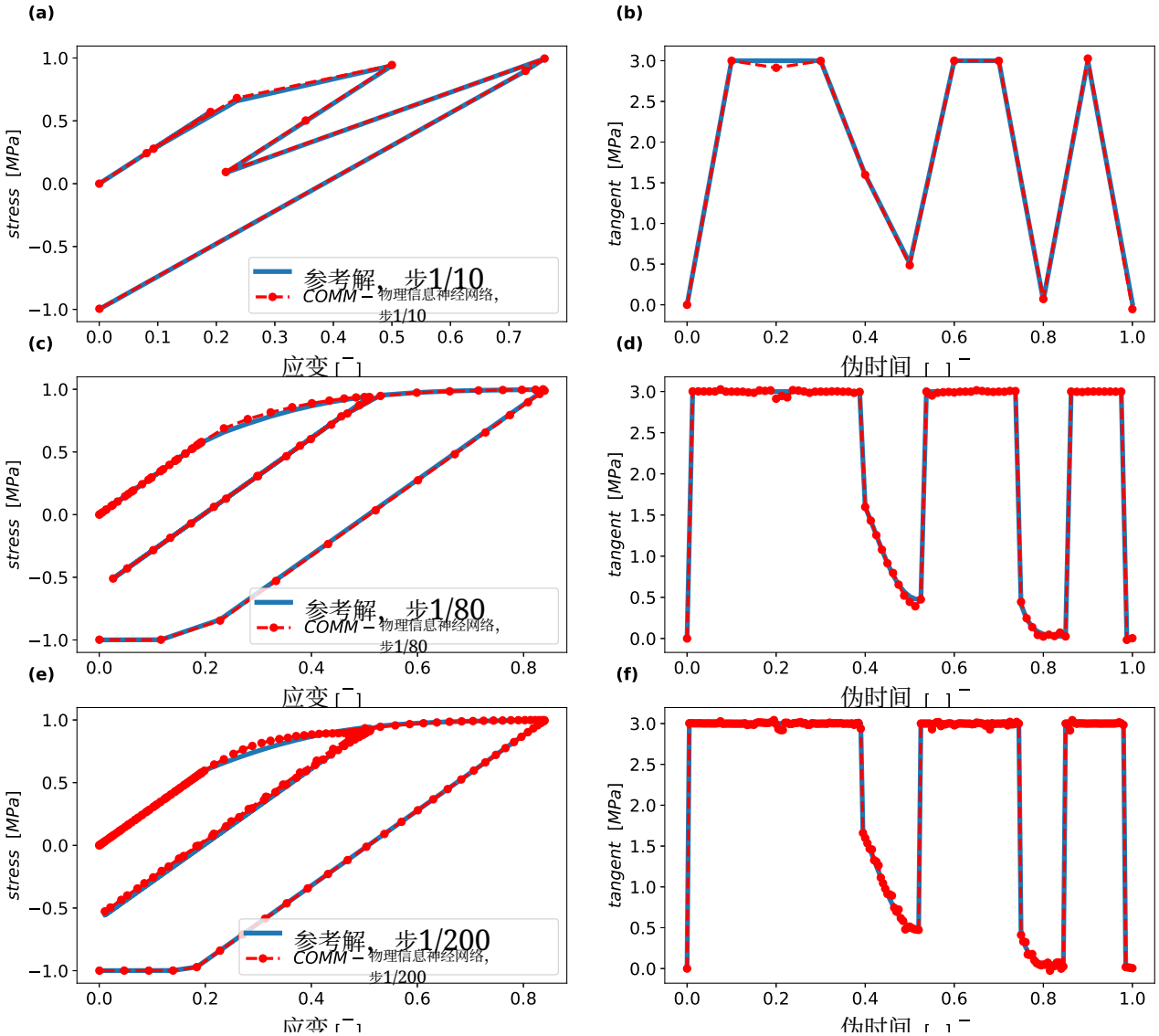


图18：通过训练神经网络和经典材料模型程序获得的应力-应变曲线及切线算子。用于创建加载路径的时间步长经过细化，以评估神经网络对于与配点不完全重合的应变路径的性能。

6.3.4. 与纯数据驱动方法的比较

注释3表明，同样的分析可以采用纯数据驱动方法。在这种方法中，状态变量增量演化的数据通过使用返回映射算法（见算法1）求解某些随机加载路径上的控制方程获得。获得数据后，我们使用表6中所述的相同神经网络架构。在这种情况下，损失函数基于预测值与塑性应变和累积塑性应变实际值之间的差异，即$\mathcal{L}_{data} =$均方误差$\left(\varepsilon_p^{i+1} - \varepsilon_{alg}^{i+1}\right) +$均方误差$\left(\xi_p^{i+1} - \xi_{alg}^{i+1}\right)$。我们生成了625条加载路径用于数据生成，如图19(a)中的灰色曲线所示。所有加载路径的应变均限制在1，这与配点的范围相似。此外，每条加载路径均以0.01的时间步长离散化，由此得到总计62500个数据点。尽管初始数据点和配点相似，但要让这两个神经网络具有完全相同的初始条件并不容易。

在图19中，我们比较了这两种方法及其预测复杂且未见过的加载路径下材料行为的能力。不过，在图19的(e)和(f)部分，我们更详细地考察了这些方法在第一个卸载循环中的响应。虽然数据驱动方法给出了可接受的预测结果，但它违反了一些基本物理规律，例如弹性卸载期间塑性应变的衰减，这是不可接受的。相反，COMM-PINN方法的结果遵循预期的直线。对于最后一个加载循环，我们评估了基于物理的方法和纯数据驱动方法中神经网络模型的外推能力。当外推至极端条件时，两种方法都可能失效。不过，带有物理约束的基于物理的方法表现略好。未来的研究可以探索这两种方法的结合。通过在训练区域之外增加更多配点，我们可以提升混合神经网络模型的预测能力，尤其是在有限应变区间内数据点有限的情况下。

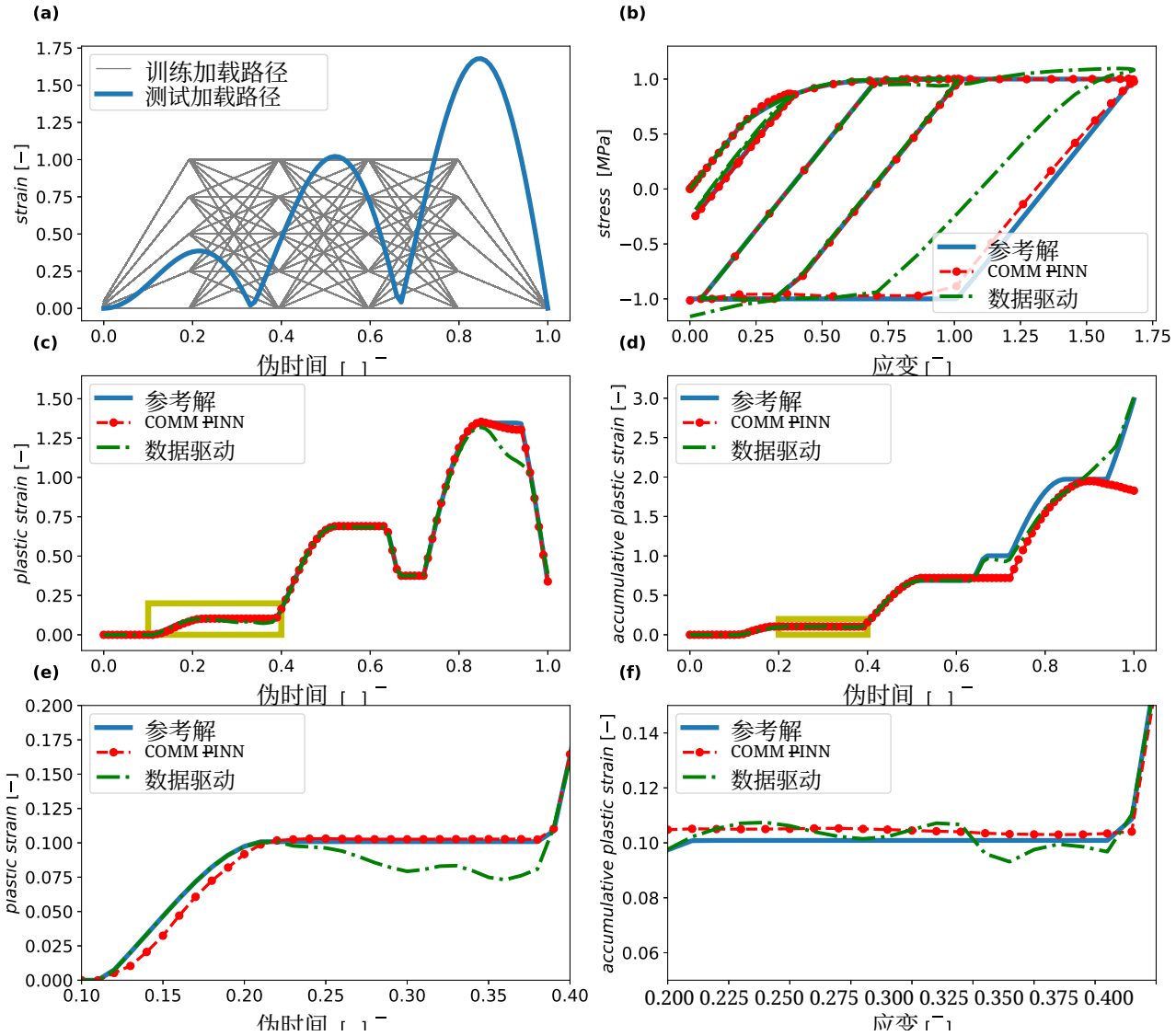


图19：所提方法与纯数据驱动方法的比较。(a) 施加的应变路径以蓝色表示，用于训练数据驱动方法的所有加载路径以灰色表示。(b) 应力-应变曲线。(c) 预测塑性应变 $\epsilon_p(t)$。(d) 预测累积塑性应变 $\xi_p(t)$。

6.3.5. 配点的影响

在上一节中，显然，当超出配点范围时，由于该区域违反了损失函数的约束，结果的准确性可能会降低。然而，需要注意的是，这不应被视为该方法的弱点，因为可以始终根据具体应用扩展范围。为进一步澄清这一问题，我们又进行了两项测试。在图 20 中，我们缩小了加载范围，显然，所获得的结果，尤其是累积塑性应变，得到了显著改善。

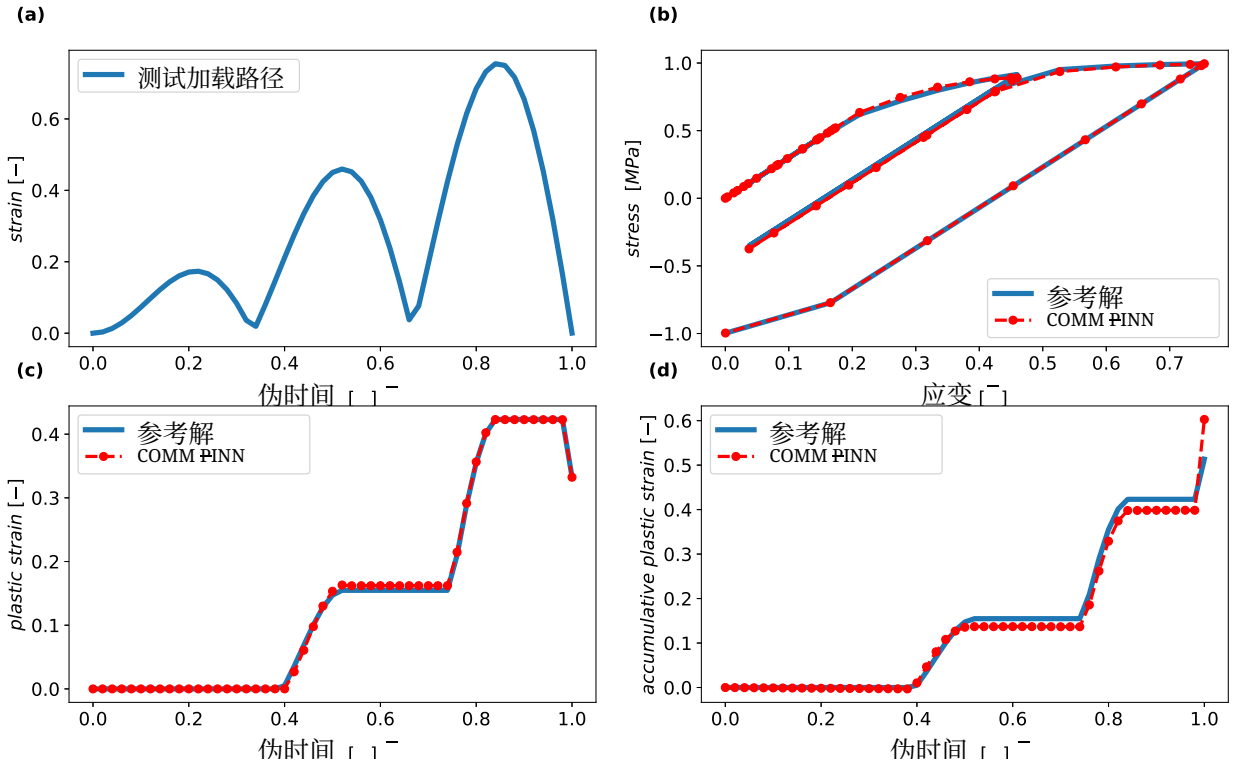


图 20：训练后的 COMM-PINN 在配点范围内准确预测了状态变量的演化。

接下来，对于给定的应变、塑性应变和累积塑性应变（即 $en_{\xi_p} = en_{\varepsilon_p} = en_e = 4.0$），我们尝试将配点范围增加到 4。详情请参阅第 5 节中的算法。其他训练参数与之前保持不变，已在前文各节中说明。图 21 展示了极端加载条件下获得的结果。有趣的是，我们观察到两种方法几乎完全吻合，即使在严苛的循环加载条件下也是如此。需要注意的是，所选的加载路径是一个复杂函数：起初，卸载并未完全回到零，但最终会收敛回原点。之所以这样选择，是为了评估经过训练的神经网络在具有挑战性且复杂的加载条件下的表现。进一步优化网络参数有望改善结果并减小误差范围。

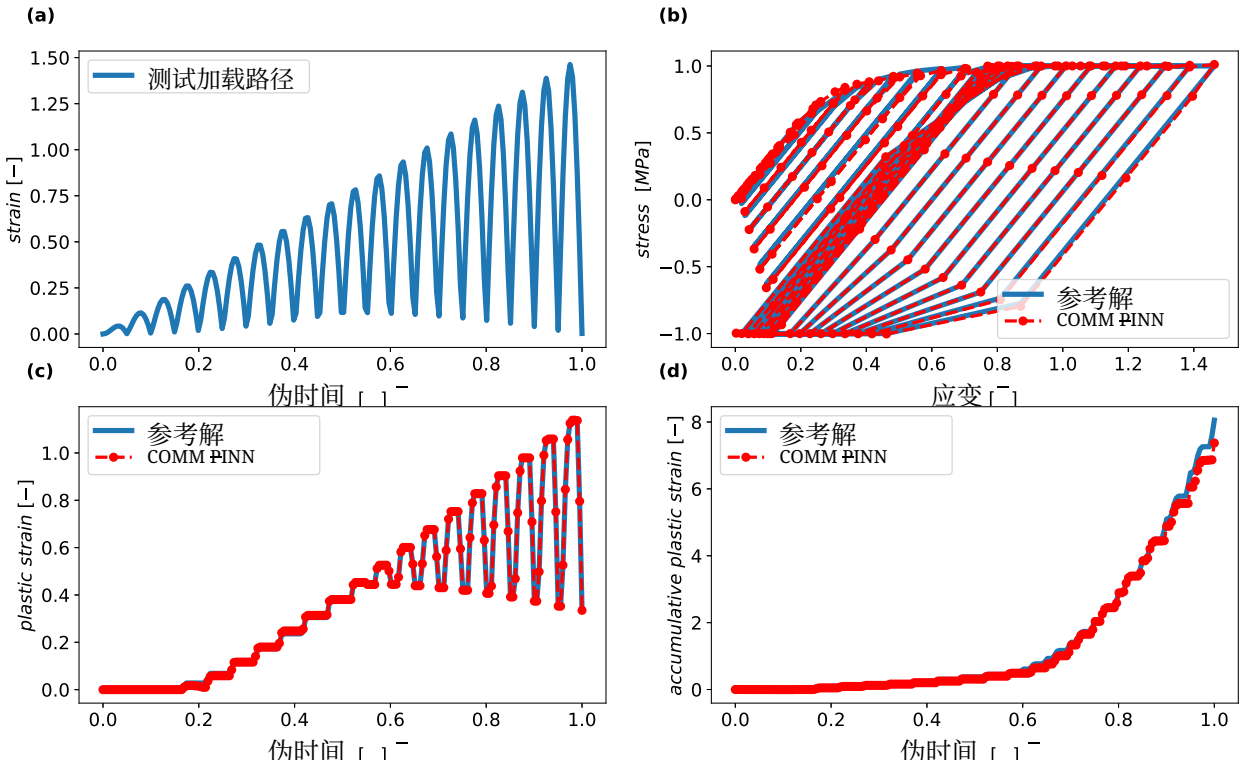


图21：通过扩大配点范围，训练后的 COMM-PINN 能够对更复杂的加载场景做出准确预测。

6.3.6. 将 COMM-PINN 模型与有限元方法集成

我们通过将 COMM-PINN 模型集成到有限元软件包中，测试了其性能。由于该模型基于一维分析进行训练，我们考察了一种桁架结构（或简化的超材料），其中每根杆件都由一个采用标准线性形函数的一维有限元网格表示。为此，我们使用了软件包 *trusspy* [67]。几何形状和施加的边界条件如图22上半部分所示：我们固定了整个结构的左侧，并从右侧竖直边缘施加载荷。加载阶段结束后，整个系统卸载。

这一过程得到的反力如图22下半部分所示。我们首先基于标准返回映射算法进行有限元计算，并引入 Voce型塑性模型，结果以蓝色显示。随后，我们使用训练好的 COMM-PINN 模型代替标准用户定义材料模型，对同一边值问题进行计算。通过比较红色和蓝色曲线，我们得出结论：在不更改现有代码的情况下，这种新方法有望取代有限元计算中的经典方法。最后，在图22右侧，我们比较了加载路径结束时两种方法得到的应力分布，发现结果完全相同。

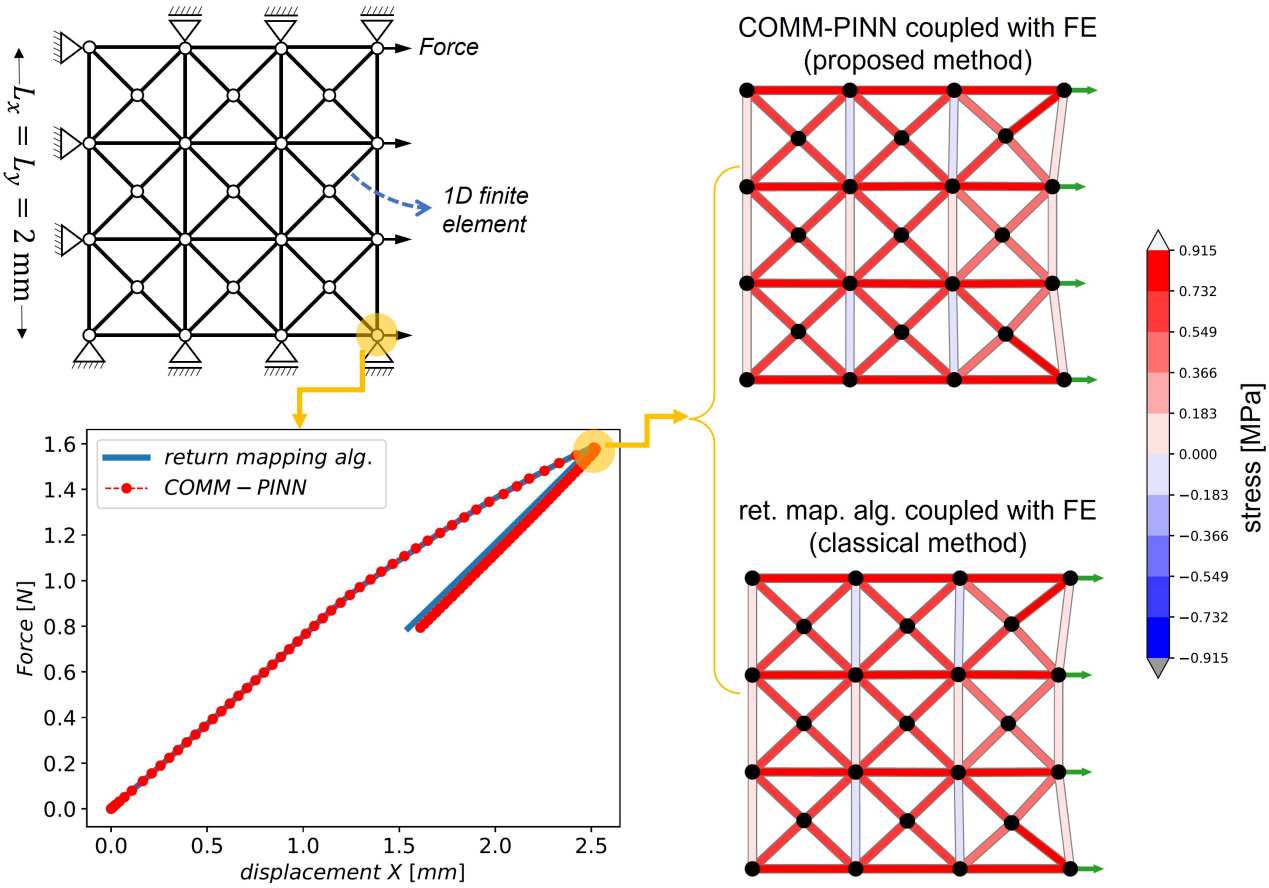


图 22：将所提方法与经典返回映射算法集成到有限元软件包（trusspy [67]）后进行比较。左上：简单桁架结构的几何形状和边界条件。下方：两种方法得到的反力。右侧：两种方法得到的桁架系统应力分布。

7. 结论与展望

本研究展示了物理信息神经网络在求解材料力学本构关系方面的潜力。当模型的自由能函数和内变量事先已知时，该方法便适用。我们提出了构造损失函数的策略，用于训练网络，并利用热力学力、演化定律和屈服函数的表达式。这就免去了训练过程中对标记数据的需求。我们还提出了简化计算流程、减少评估切线算子时所需微分阶数的技术。

与经典返回映射算法和其他纯数据驱动方法相比，所提出的 COMM-PINN 方法具有多项优势。该模型能够立即预测配点范围内任意加载路径下的材料响应。此外，成功训练后，用户可灵活调整时间步长，同时保持可靠的预测结果。这一能力表明，该方法能够稳健地适应不同的时间离散方式，并具有插值能力。通过调整损失函数的权重并开展超参数研究，可以增强模型的外推能力。所提方法无需进行复杂的线性化

，并能从神经网络中自然地得到切线算子。模型训练无需标记数据，整个方法均基于标准热力学原理推导出的控制材料方程。有趣的是，在复杂的测试加载路径下，所提方法的表现优于纯数据驱动方法。最后，训练好的材料程序可以轻松集成到任何有限元软件包中。

所提出的方法为加速复杂材料建模过程带来了良好前景，同时也有进一步改进的空间。我们针对三维界面损伤模型开展的初步研究表明，与标准返回映射算法和隐式数值方案相比，该方法可将计算成本降低约 50%。这一改进能够更高效地模拟代表性体积单元，从而更便捷地获取通过有限元计算得到的全场模拟结果。

采用这一方法后，用户可以将主要精力放在材料模型本身的开发上，因为无论数值评估过程多么复杂，都能得到显著简化。此外，以这种方式实现的模型可以轻松地在不同研究团队之间移植，并集成到各种数值求解器中。这种易于集成的特性，得益于网络评估过程的标准化和低成本；而这又是通过优化神经元和层的数量实现的。

在相同的时间步长下（此时显式算法可获得稳定解，即对于 $\Delta t = 0.0005$），三维界面损伤模型的显式方法速度比隐式方法快 3 到 4 倍，且计算成本与 COMM-PINN 方法几乎相同。不过，需要注意的是，对于这一特定示例和材料模型，隐式方法和 COMM-PINN 方法的时间步长可以增加 100 倍，同时仍能获得一致的结果。这表明，在计算加载路径时，我们可以将步长增加到 $\Delta t = 0.05$。此时，隐式方法的速度约为显式方法的 $100/3.4 \sim 30$ 倍，而 COMM-PINN 方法的速度约为显式方法的 $100/1.2 \sim 80$ 倍。因此，可以得出结论，COMM-PINN 方法兼具各类方法的优势。

在未来的研究中，有必要扩展该框架，并针对其他材料属性对系统进行训练。还需要研究多维情况 [35]以及具有各向异性的更复杂材料子程序 [68] ，以便更好地理解这一方法的优势。作为进一步的示例，可以提及先进的非局部材料模型，以及多物理场环境中的发展 [69]。此外，还需要对配点集进行优化设计，以减少训练时间，同时获得最佳精度[70]。理想情况下，可以针对每种材料模型对神经网络进行一次训练，并将其用作标准材料子程序的替代方案，从而无需使用不同的编程语言重写代码和材料模型。

数据可用性:
代码可在 COMM-PINN 获取。

8. 附录 A：与显式方法的比较

在本研究中，我们优先开发了三维界面损伤模型，并成功实现了显式方法。该方法通过利用损伤屈服函数计算损伤参数来实现，而损伤屈服函数依赖于前一个时间步的信息。因此，无需求解额外的方程，从而大大简化了实现过程。不过，需要注意的是，我们发现显式方法存在一定的稳定性问题。为了获得可靠的结果，我们发现必须采用非常小的时间步长，例如 $\Delta t = 0.0005$。相比之下，隐式方法具有更高的稳定性，即使采用较大的时间步长（例如 $\Delta t = 0.05$），也能获得可靠的预测结果，如下图所示。COMM-PINN 方法也同样如此，它可以使用大得多的时间步长（见图18）。

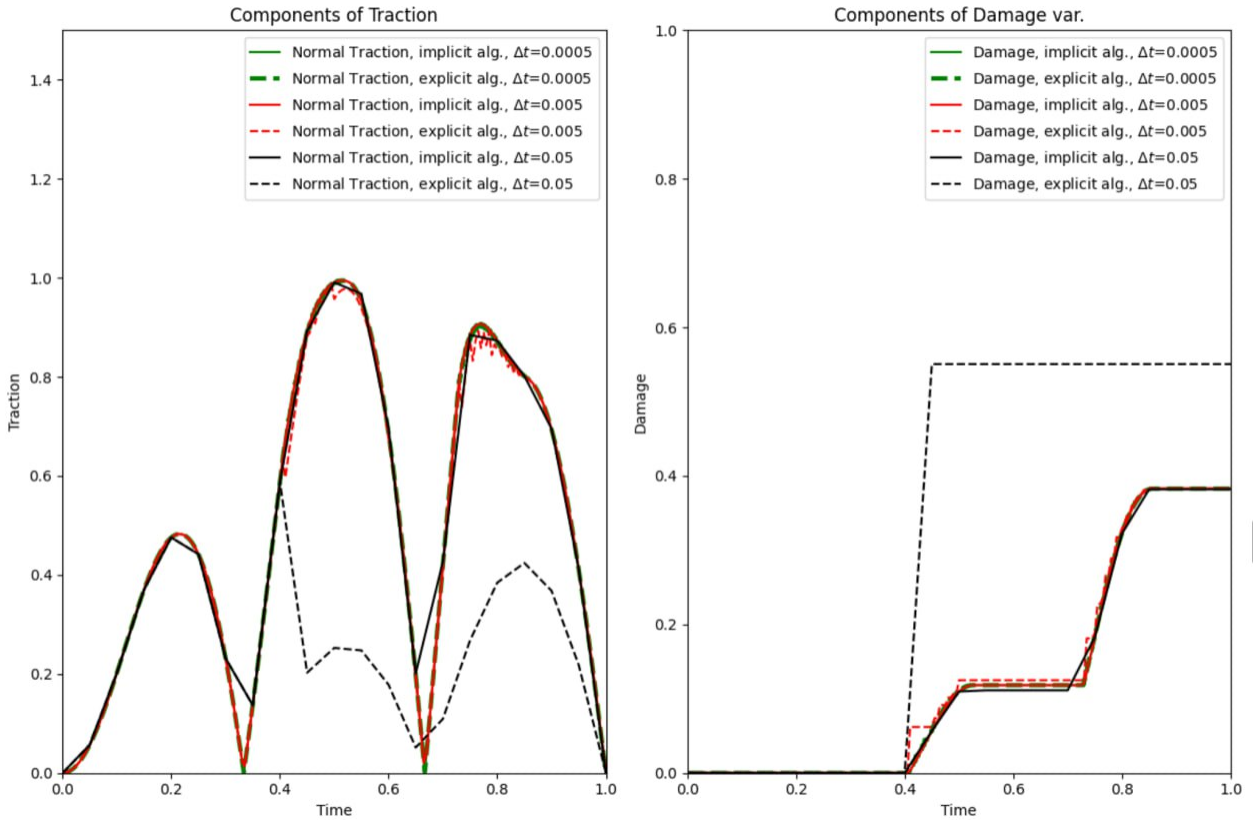


图 23：不同方法和不同时间步长所得结果的比较。

通过采用隐式方法（或 COMM-PINN），我们可以利用更大的时间步。这些方法不仅在计算成本方面表现出色，还具有更高的可靠性。采用隐式方法时，由于直接求解底层方程，我们可以确信热力学一致性得到了满足。相反，采用显式方法时，我们高度依赖前一时间步的结果，却无法保证控制方程在当前时间得到准确满足。这可能导致响应出现振荡，例如 $\Delta t = 0.005$ 的牵引力行为所示。

作者声明:
S.R.：概念化、软件、监督、撰写——审阅与编辑。A. M.：软件、撰写——审阅与编辑。A. H.：软件、撰写——审阅与编辑。

参考文献

[1]F. Roters, M. Diehl, P. Shanthraj, P. Eisenlohr, C. Reuber, S.L. Wong, T. Maiti, A. Ebrahimi, T. Hochrainer, H.-O. Fabritius, S. Nikolov, M. Fri´ak, N. Fujita, N. Grilli, K.G.F. Janssens, N. Jia, P.J.J. Kok, D. Ma, F. Meier, E. Werner, M. Stricker, D. Weygand 和 D. Raabe。《Damask——杜塞尔多夫先进材料模拟工具包：用于模拟从单晶到构件尺度的多物理场晶体塑性、热学和损伤现象》。《计算材料科学》，158：420–478，2019年。[2] Jinhyun Choo 和 WaiChing Sun。《地质材料的相场与塑性耦合建模：从脆性断裂到韧性流动》。*Computer Methods in Applied Mechanicsand Engineering*，330：1–32，2018年。[3] Tim Brepols、Stephan Wulfinghoff 和 Stefanie Reese。《梯度扩展双表面损伤塑性：微形态公式与数值方面》。《国际塑性学杂志》，97：64–106，2017年。[4] Karel Matou˘s、Marc G.D. Geers、Varvara G. Kouznetsova 和 Andrew Gillman。《非均质材料多尺度建模预测性非线性理论综述》。《计算物理学杂志》，330：192–220，2017年。[5]George Weber、Maxwell Pinz 和 Somnath Ghosh。《机器学习赋能的自洽参数化上尺度晶体塑性模型：用于镍基高温合金》。*Com-puter Methods in Applied Mechanics and Engineering*，402：115384，2022年。[6] Rupesh Kumar Tipu、V R Panchal 和 K S Pandya。《使用机器学习算法预测混凝土性能》。《物理学杂志：会议系列》，2273(1)：012016，2022年5月。[7] Alexander E. Mayer、Mikhail V. Lekanov、Natalya A. Grachyova 和 Eugeniy V. Fomin。《基于机器学习的铜弹塑性变形模型及其在冲击波问题中的应用》。《金属》，12，2022年。[8] Frederic E. Bock、Roland C. Aydin、Christian J. Cyron、Norbert Huber、Surya R. Kalidindi 和 Benjamin Klusemann。《机器学习与数据挖掘方法在连续体材料力学中的应用综述》。《材料前沿》，6，2019年。[9] Felix Fritzen、Mauricio Fern´andez 和 Fredrik Larsson。《利用人工神经网络和降阶建模实现非线性双尺度模拟的即时自适应》。《材料前沿》，6，2019年。[10] Usman Ali、Waqas Muhammad 和 Abhijit。《人工神经网络在多晶金属细观力学中的应用》。《国际塑性学杂志》，120：205–219，2019年。[11] Dengpeng Huang、Jan Niklas Fuhg、Christian Weißenfels 和 Peter Wriggers。《一种基于本征正交分解的机器学习塑性模型》。*Computer Methodsin Applied Mechanics and Engineering*，365：113008，2020年。[12] Ari Frankel、Kousuke Tachida 和 Reese Jones。《利用混合神经网络模型预测发生弹塑性变形的多晶体应力场演变》。《机器学习：科学与技术》，1(3)：035005，2020年。

[13] XinLiu、SuTian、Fei Tao和 Wenbin Yu。复合材料本构建模中的人工神经网络综述。*Composites Part B: Engineering*，224:109152，2021年。[14] Johannes Dornheim、Lukas和MorandDirkHelm。用于本构建模的神经网络——从通用函数逼近器到高级模型及物理学的融合。预印本，2023年。[15] Deep Ray、OrazioPinti 和 AssadA.Oberai。深度学习与计算物理学（讲义），2023年。[16] Grace C. Y. Peng、Mark Alber、Adrian Buganza Tepole、William R. Cannon、Suvranu De、Savador Dura-Bernal、Krishna Garikipati、George Karniadakis、William W. Lytton、Paris Perdikaris、Linda Petzold 和Ellen Kuhl。多尺度建模遇上机器学习：我们能学到什么？*Archives of Computational Methods in Engineering*，28:1017–1037，2021年。[17] JaberRezaei Mianroodi、Shahed Rezaei、Nima H Siboni、Bai-XiangXu 和Dierk Raabe。借助人工智能实现无损多尺度本构弹性关系。*npj* 计算材料，8:1–12，2022年。[18] Jan-Hendrik Bastek、Siddhant Kumar、Bastian Telgen、Rapha¨el N. Glaesener和DennisM.Kochmann。通过深度学习反演桁架超材料的结构—性能映射。美国国家科学院院刊，119(1):e2111505119，2022年。[19] Alexander Henkes、JasonK.Eshraghian 和Henning Wessels。用于非线性回归的脉冲神经网络，2022年。[20] Faisal As'ad、Philip Avery和 CharbelFarhat。数据驱动本构建模中的力学信息引导人工神经网络方法。*International Journal for NumericalMethods in Engineering*，123(12):2738–2759，2022年。[21] KailaiXu、Daniel Z.Huang 和 Eric Darve。使用对称正定神经网络学习本构关系。《计算物理学杂志》，428:110072，2021年。[22] Zeliang Liu、C.T.Wu和M.Koishi。用于多尺度拓扑学习和非均质材料加速非线性建模的深度材料网络。*Computer Methodsin Applied Mechanics and Engineering*，345:1138–1168，2019年。[23] Argha Protim Dey、Fabian Welschinger、Matti Schneider、SebastianGajek 和 Thomas B¨ohlke。采用非弹性信息引导策略训练深度材料网络，以再现短纤维增强热塑性塑料的蠕变加载。*Archive of Applied Mechanics*，92：2733–2755，2022年。[24] YousefHeider、Kun Wang和WaiChingSun。各向异性弹塑性材料的信息图引导深度神经网络的 So(3)不变性。*Computer Methods in AppliedMechanics and Engineering*，363:112875，2020年。[25] Nikolaos N.Vlassis 和 WaiChingSun。针对具有水平集硬化的可解释弹塑性模型，对热力学信息引导神经网络进行 Sobolev训练。*ComputerMethods in Applied Mechanics and Engineering*，377:113695，2021年。

[26] Moritz Flaschel、Siddhant Kumar 和 Laura De Lorenzis。利用欧几里得自动发现广义标准材料模型。*Computer Methods in Applied Mechanics andEngineering*，405:115867，2023年。[27] Jan Niklas Fuhg、Craig M. Hamel、Kyle Johnson、Reese Jones 和 Nikolaos Bouklas。模块化机器学习弹塑性：有限数据背景下的泛化。*Computer Methods in Applied Mechanics and Engineering*，407:115930，2023年。[28] Patrick Weber、Jeremy Geiger 和 Werner Wagner。受约束的神经网络训练及其在超弹性材料建模中的应用。计算力学，68:1179–1204，2021年。[29] Karl A. Kalina、Lennart Linden、J¨org Brummund 和 Markus K¨astner。 fe^{ANN}：一种基于物理约束神经网络和自动数据挖掘的高效数据驱动多尺度方法。计算力学，71:827–851，2023年。[30] Dominik K. Klein、Mauricio Fern´andez、Robert J. Martin、Patrizio Neff 和 Oliver Weeger。利用神经网络实现多凸各向异性超弹性。《固体力学与物理学杂志》，159:104703，2022年。[31] Filippo Masi、Ioannis Stefanou、Paolo Vannucci 和 Victor Maffi-Berthier。基于热力学的人工神经网络用于本构建模。《固体力学与物理学杂志》，147:104277，2021年。[32] Tianju Xue、Alex Beatson、Maurizio Chiaramonte、Geoffrey Roeder、Jordan T. Ash、Yigit Menguc、Sigrid Adriaenssens、Ryan P. Adams 和 Sheng Mao。大变形条件下胞状机械超材料非线性力学性质的数据驱动计算方案。*Soft Matter*，16:7524–7534，2020年。[33] Filippo Masi 和 Ioannis Stefanou。Evolution tann 及固体力学中内部变量与演化方程的识别。《固体力学与物理学杂志》，第105245页，2023年。[34] M. Mozaffar、R. Bostanabad、W. Chen、K. Ehmann、J. Cao 和 M. A. Bessa。深度学习预测路径相关塑性。美国国家科学院院刊，116（52）：26414–26420，2019年。[35] Annan Zhang 和 Dirk Mohr。利用神经网络表示具有各向同性硬化的 von Mises 塑性。《国际塑性学杂志》，132:102732，2020年。[36] Xiaolong He 和 Jiun-Shyan Chen。用于路径相关材料数据驱动建模的热力学一致机器学习内部状态变量方法。*Computer Methods in Applied Mechanics and Engineering*，402:115348，2022年。[37] Colin Bonatti 和 Dirk Mohr。自洽性对于表示弹塑性固体的循环神经网络模型的重要性。《固体力学与物理学杂志》，158:104697，2022年。[38] Arnd Koeppe、Franz Bamer、Michael Selzer、Britta Nestler 和 Bernd Markert。面向力学的可解释人工智能：用于本构模型的物理解释型神经网络。《材料前沿》，8，2022年。

[39] Aymen Danoun、Etienne Pruli`ere和 Yves Chemisky。用于预测承受非比例加载路径的耗散材料非线性行为的热力学一致循环神经网络。材料力学，173:104436，2022年。[40] Pin Zhang、Zhen-YuYin、Yin-FuJin和 BrianSheil。用于土壤复杂路径相关行为的物理约束分层数据驱动建模框架。国际数值与解析岩土工程方法学杂志，46(10):1831–1850，2022年。[41] M.A. Maia、I.B.C.M. Rocha、P. Kerfriden和 F.P.van der Meer。路径相关非均质材料的物理循环神经网络：将本构模型嵌入数据驱动替代模型。*Computer Methods in Applied Mechanics and Engineering*，407: 115934，2023年。[42] Max Rosenkranz、Karl A Kalina、J¨org Brummund 和Markus K¨astner。用于建模非弹性的不同神经网络架构的比较研究。*arXiv* 预印本*arXiv:2303.03402*，2023年。[43] M. Raissi、P.Perdikaris和 G.E.Karniadakis。物理信息神经网络：用于求解涉及非线性偏微分方程的正问题与反问题的深度学习框架。《计算物理学杂志》，378:686–707，2019年。[44] Ehsan Haghighat、Maziar Raissi、Adrian Moure、Hector Gomez 和RubenJuanes。用于固体力学反演与代理建模的物理信息深度学习框架。*Computer Methods in Applied Mechanics and Engineering*，379:113741，2021年。[45] Salah A Faroughi、Nikhil Pawar、Celio Fernandes、Maziar Raissi、Subasish Das、Nima K. Kalantari和SeyedKourosh Mahjour。科学计算中的物理引导、物理信息和物理编码神经网络，2023年。[46] Shahed Rezaei、Ali Harandi、Ahmad Moeineddin、Bai-Xiang Xu 和StefanieReese。物理信息神经网络的混合形式：作为非均质域中工程问题的潜在求解器：与有限元方法的比较。*ComputerMethods in Applied Mechanics and Engineering*，401:115616，2022年。[47]E. Samaniego、C. Anitescu、S.Goswami、V.M.Nguyen-Thanh、H. Guo、K. Hamdia、X.Zhuang和T.Rabczuk。通过机器学习求解计算力学中偏微分方程的能量方法：概念、实现与应用。*Computer Methods in Applied Mechanics and Engineering*，362:112790，2020年。[48] Jan N.Fuhg 和NikolaosBouklas。用于解析有限应变超弹性中集中特征的混合深度能量法。《计算物理学杂志》，451: 110839，2022年。[49] Congjie Wei、Jiaxin Zhang、Kenneth M. Liechti和 ChenglinWu。使用热力学一致神经网络对界面牵引–分离关系进行数据驱动建模。*Computer Methods in Applied Mechanics and Engineering*，404:115826，2023年。

[50] Ramakrishna Tipireddy、Paris Perdikaris、Panos Stinis和 AlexandreTartakovsky。使用物理信息神经网络模型学习未知动力学和本构关系的比较研究，2019年。[51] Ehsan Haghighat、Sahar Abouali和 Reza Vaziri。使用物理信息深度学习进行本构模型表征与发现。《style id='3'>Engineering Applications ofArtificialIntelligence，120：105828，2023年。[52] Ehsan Haghighat、Maziar Raissi、Adrian Moure、Hector Gomez 和Ruben Juanes。用于固体力学反演和代理建模的物理信息深度学习框架。《style id='6'>Computer Methods in Applied Mechanics and Engineering，379：113741，2021年。[53] Sijun Niu、Enrui Zhang、YuriBazilevs 和VikasSrivastava。使用物理信息神经网络对有限应变塑性进行建模及网络性能评估。《style id='9'>Journal of the Mechanics and Physics ofSolids，172：105177，2023年。[54] Junyan He、Diab Abueidda、Rashid Abu Al-Rub、SeidKoric 和 Iwona Jasiuk。用于经典弹塑性的基于深度学习能量的方法。《 style id='12'>InternationalJournal ofPlas-ticity，162：103531，2023年。[55] Mahdad Eghbalian、Mehdi Pouragha和Richard Wan。用于经典弹塑性代理建模的物理信息深度神经网络。《style id='15'>Computers and Geotechnics，159：105472，2023年。[56] Dong Phill Jang、PiemaanFazily和 Jeong Whan Yoon。基于机器学习的 J2 塑性本构模型。《style id='18'>International Journal of Plasticity，138：102919，2021年。[57] Mauricio Fern´ández、Shahed Rezaei、Jaber Rezaei Mianroodi、FelixFritzen 和 Stefanie Reese。人工神经网络在界面力学预测中的应用：晶界本构行为研究。《 styleid='22'>《工程科学高级建模与仿真》，7：1–27，2020年。[58] JiaxinZhang、CongjieWei和 ChenglinWu。用于学习材料界面力学的热力学一致神经网络，2020年。[59] Armin Asheri、Mozhdeh Fathidoost、Vedran Glavas、ShahedRezaei和 Bai-Xiang Xu。通过机器学习对固态电池进行数据驱动的多尺度模拟。《style id='26'>Computa-tional MaterialsScience，226：112186，2023年。[60] R. Eggersmann、T. Kirchdoerfer、S. Reese、L. Stainier 和 M. Ortiz。无模型数据驱动的非弹性。《style id='29'>Computer Methods in Applied Mechanics and Engineering，350：81–99，2019年。[61] Shahed Rezaei、Ali Harandi、TimBrepols 和Stefanie Reese。各向异性内聚断裂模型：长度尺度不敏感相场损伤模型的优势与局限。《style id='32'>Engineering Fracture Mechanics，261：108177，2022年。[62] Shahed Rezaei、David Jaworek、Jaber Rezaei Mianroodi、StephanWulfinghoff 和Ste-fanie Reese。原子尺度启发的界面模型：考虑晶界处塑性与损伤耦合。《style id='35'>Journal of the Mechanics and Physics ofSolids，124：325–349，2019年。

[63] J¨urgen Schmidhuber. 神经网络中的深度学习：综述。神经网络，61:85–117，2015年。[64] Ameya D. Jagtap 和 George Em Karniadakis. 激活函数在回归和分类中有多重要? 调查、性能比较与未来方向，2022年。[65] Ali Harandi、Ahmad Moeineddin、Michael Kaliske、Stefanie Reese 和 Shahed Rezaei. 用于热力学耦合系统和非均质域的物理信息神经网络混合形式，2023年。[66] Ehsan Haghighat 和 Ruben Juanes. Sciann：一个基于 keras/tensorflow 的封装工具，用于科学计算以及借助人工神经网络进行物理信息深度学习。计算力学与工程中的计算方法，373:113552，2021年。[67] Andreas Dutzler. trusspy：Python 桁架求解器。网址： https://github.com/adtzlr/trusspy.[68] Olga Ibragimova、Abhijit Brahme、Waqas Muhammad、Julie L´evesque 和 Kaan Inal. 一种新的基于人工神经网络的面心立方材料晶体塑性模型及其在非单调应变路径中的应用。《国际塑性学杂志》，144:103059，2021年。[69] Shahed Rezaei、Armin Asheri 和 Bai-Xiang Xu. 化学-力学内聚断裂的一致性框架及其在固态电池中的应用。《固体力学与物理学杂志》，157:104612，2021年。[70] Ronak Shoghi 和 Alexander Hartmaier. 各向异性塑性中机器学习屈服函数的最优数据生成策略。《材料前沿》，9，2022年。