arXiv:2302.10184v2 [cs.LG] 5 Feb 2025

AttNS：注意力启发数值求解 有限数据场景下的数值求解

Zhongzhan Huang[† 1 2] **Mingfu Liang** [3] **Shanshan Zhong** [1] **Liang Lin** [1 2]

摘要

我们提出了注意力启发数值求解器（AttNS），这是一种简洁的方法，有助于解决人工智能混合数值求解器因数据有限而在求解微分方程时面临的泛化和鲁棒性问题。AttNS 的设计灵感来源于残差神经网络（ResNet）中的注意力模块在增强模型泛化能力和鲁棒性方面的有效性。基于 ResNet 的动力系统视角，我们将注意力机制无缝融入针对微分方程求解特性而设计的数值方法中。我们在从高维问题到混沌系统的一系列基准上的结果表明，AttNS 无需复杂的模型设计即可持续提升各种数值求解器的性能。最后，我们通过实验和理论分析了 AttNS，证明了其在确保求解器收敛性的同时实现强泛化能力和鲁棒性的能力。这包括与其他先进方法相比，仅需更少的数据即可达到相当的泛化误差，并能更好地防止求解微分方程时出现数值爆炸问题。https://github.com/dedekinds/NeurVec。

1. 引言

人工智能混合数值求解器（AHS）是用于求解微分方程的新兴方法（Mishra, 2018; San & Maulik, 2018; Bar-Sinai 等人, 2019; Kochkov 等人, 2021; Subel 等人, 2021; Bruno 等人, 2022; List 等人, 2022; Dresdner et al., 2022; Frezat 等人, 2022; Huang et al., 2023b; Xu et al., 2023），旨在将经典数值技术与深度学习技术相结合。这些方法利用高分辨率数据为低分辨率数值解学习修正项（Um 等人，2020）。其目标是结合两种技术的优势——经典数值方法的可靠性和神经网络的表达能力——以缓解求解微分方程时固有的速度-精度权衡。例如，对于一个常微分方程（ODE）

$$\mathrm{d}\mathbf{u}(\mathbf{t})/\mathrm{d}t = \mathbf{f}[\mathbf{u}(t)], \mathbf{u}(0) = \mathbf{c}_0, \tag{1}$$

其中，$\mathbf{u}(\mathbf{t})$ 是一个随时间变化的 d维状态，$\mathbf{c}_0 \in \mathbb{R}^d$ 为初始条件。方程（1）的前向 AHS 通常（Dresdner 等人，2022）可写为

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \underbrace{S(\mathbf{f}, \mathbf{u}_n, \Delta t_c)\Delta t_c}_{\text{Low-resolution term}} + \underbrace{\text{Net}(\mathbf{u}_n|\phi, \mathbb{D}_f)}_{\text{Correction term}}, \tag{2}$$

其中，Net(·) 是一个具有可学习参数的神经网络 ϕ，$S(\cdot)$ 是一种数值积分方案，$\mathbb{D}_f$ 表示高分辨率数据，即通过细步长 Δt_f 生成的高精度数据。粗步长 Δt_c 可以加速计算，而由此导致的精度下降则通过使用 $\mathbb{D}_f$ 训练的修正项 Net($\mathbf{u}_n|\phi$, $\mathbb{D}_f$) 予以补偿，从而在求解微分方程时实现速度与精度之间的良好平衡。

然而，由于修正项采用了数据驱动的深度学习方法，当前 AHS 的性能高度依赖于高质量训练数据的数量，而**获取足够的训练数据成本非常高**。例如，如果我们使用 $\Delta t_f = 0.001$的数据来训练 $\Delta t_c = 0.1$的修正项，生成一条数据轨迹将需要超过 $100\times$ 额外的计算，而由于方程（2）的迭代性质，这些计算很难通过并行化加速。此外，复杂方程对数据的高需求（Huang 等，2023年b）进一步加剧了数据获取成本。鉴于这些挑战，我们提出一个关键问题：

我们能否改进 *AHS*，使其即使在有限数据场景下也能实现高效计算？

为回答这一问题，本文以式（1）所示的常微分方程为例，进行全面分析。首先，我们在第 2.3 节介绍有限数据场景下 AHS 面临的具体挑战。除了数据有限导致神经网络固有的泛化问题外，求解微分方程时与混沌相关的鲁棒性问题（Liang 等人，2022）也需要审慎考虑。

[1]中国中山大学 [2]中国鹏城实验室 [3]美国西北大学。† 曾在鹏城实验室进行为期两周的非正式短期访问。通讯作者：Liang Lin <linliang@ieee.org>。

发表于 41^{st} 国际机器学习会议，奥地利维也纳。PMLR 235，2024。版权所有 © 2024，作者所有。

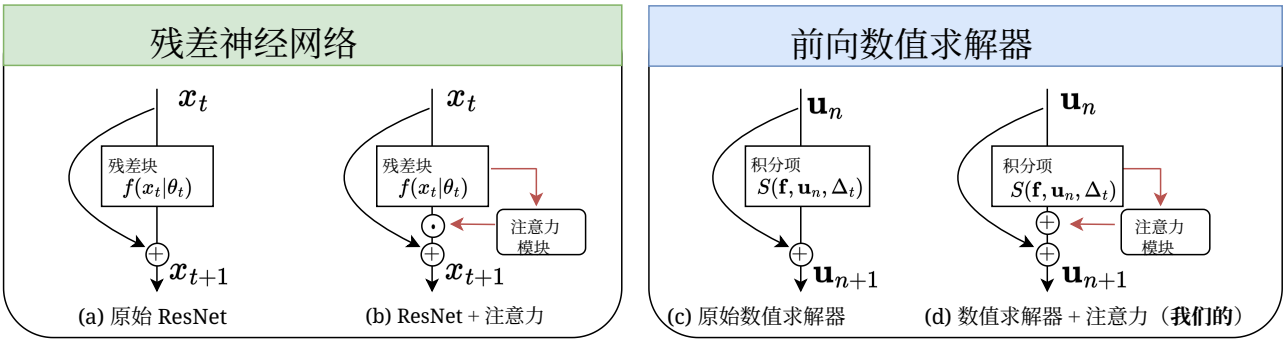


图 1：(a) ResNet 与 (c) 前向数值求解器之间的对应关系。(d) 是一种受 (b) 带注意力机制的 ResNet 结构启发、采用注意力机制的数值求解器（我们的方法）。 $\odot$ 表示逐元素乘法， $\oplus$ 表示加法运算符。使用 $\oplus$ 是**专为**求解微分方程而设计，详情请参见第3 节。

针对这些挑战，我们注意到，残差神经网络（ ResNet）中的注意力机制模块已在传统深度学习任务中得到广泛验证，能够有效提升模型的泛化能力和鲁棒性（Hu 等，2018年；Huang 等，2020；Woo 等，2018年；Wang 等人，2020；Liang 等人，2020；Zhong 等，2023年；参见第2.4节）。此外，鉴于从 ResNet 的动力系统视角出发所建立的 ResNet 与数值求解器之间的对应关系（参见第2.2 节和图1），我们遵循微分方程求解的特点，将注意力机制模块无缝融入数值求解器，并在第3节提出一种简单而有效的注意力启发式数值求解器（AttNS）。

通过在若干高维或混沌系统的标准基准测试上开展实验，包括弹簧-质量块系统、弹性摆以及 K连杆摆，第4节的结果表明，在有限数据场景下，AttNS 始终优于现有最先进的 AHS，且无需精心设计模型。接下来，我们在第5节对 AttNS 进行实验和理论分析，证明了它具有很强的泛化能力和鲁棒性，这与注意力机制在传统深度学习任务中的表现类似，同时还能确保求解器收敛。这包括与其他先进方法相比，使用更少的数据实现同等的泛化误差，并更有效地避免求解微分方程时出现数值爆炸问题。最后，我们讨论 AttNS 的局限性。我们的贡献总结如下：

- 为了在有限数据场景中有效应用 AHS，我们将注意力机制融入 AHS，并提出了一种简单而有效的方法 AttNS。标准基准测试表明，AttNS 的表现始终优于其他先进的 AHS 方法。
- 我们对 AttNS 进行了实验和理论分析，证明其具备出色的泛化能力和鲁棒性，同时确保求解器的收敛性。最后，我们讨论了 AttNS 的局限性。

2. 预备知识与相关工作

2.1. 微分方程的求解

对于给定的方程，如式（1）所示，经典的显式数值方法，例如欧拉方法（Shampine，2018年）、Runge-Kutta方法（Butcher，2016年）等，使用迭代公式对其进行求解

$$\mathbf{u}_{n+1} = \mathbf{u}_n + S(\mathbf{f}, \mathbf{u}_n, \Delta t)\Delta t, \quad \mathbf{u}_0 = \mathbf{c}_0. \tag{3}$$

不同的方法具有不同的 S。例如，对于欧拉方法（ Shampine，2018年），有 $S(\mathbf{f}, \mathbf{u}_n, \Delta t)\Delta t = \mathbf{f}(\mathbf{u}_n)\Delta t$，其中 Δt 为给定的步长，$\mathbf{u}_n \in \mathbb{R}^d$ 为时间 $\sum_{i=0}^{n} \Delta t$处的近似解。本文研究的 AI-Hybrid 数值求解器（AHS），如式（2）所示，引入了一个修正项，即使步长较大，也能高精度地求解微分方程，从而加快求解速度。对于其他利用深度学习技术进行求解的方法，例如使用神经网络直接拟合（Geneva 与 Zabaras，2022；Liang 等人，2021）、侧重于替换微分算子的基于算子的方法（Li 等人，2021b；Lu 等人，2021）、基于神经常微分方程的方法（Chen 等人，2018；Kidger 等人，2021；Dupont 等人，2019）、基于哈密顿量的方法（ Greydanus 等人，2019；Chen 等人，2019；Liang 等人，2022），以及物理信息神经网络（Choudhary 等人，2020；Raissi 等人，2019；Ji 等人，2021；Cai 等人，2021），由于这些方法与 AHS 的范式存在根本差异，难以进行公平比较。因此，本文主要关注 AHS，而非其他基于深度学习的求解方法。

2.2. ResNet 的动力系统视角

如图1（a）所示，ResNet 中的简化残差块（He 等人，2016年）可以写为

$$x_{t+1} = x_t + f(x_t|\theta_t), \tag{4}$$

其中，$x_t \in \mathbb{R}^d$ 是神经网络 $f(\cdot|\theta_t)$的输入，该网络在第 t 个块中具有可学习参数 θ_t。近期的一些研究（Weinan，2017；Queiruga 等人，2020；Zhu 等人，2022；Meunier 等人，2022）揭示了残差块与动力系统之间有价值的联系。也就是说，残差块可以解释为式（3）和图1(c)中显式数值方法的一步。式（3）中的初始条件 $\mathrm{u}_0 = \mathrm{c}_0$ 对应于网络的初始输入 x_0，而 u_t 对应于第 t 个块中的输入特征 x_t。第 t个块中的神经网络 $f(\cdot|\theta_t)$的输出可以视为一个积分 $S(\mathrm{u}_t, \mathrm{f}\ \Delta t)$，其步长为 Δt，数值积分方案为 S。基于上述联系，一些动力系统理论（Chang 等人，2017；Chen 等人，2018；Huang 等人，2022b；Lu 等人，2018）可以应用于 ResNet 的分析。

2.3. 有限数据场景下 AHS 面临的挑战

我们首先简要定义有限数据场景（LDS）。通常，AHS 的性能与训练数据量正相关（Huang 等，2023年b），并在数据量达到 N_b时趋于饱和，即进一步增加数据规模几乎不会提升性能。本文将数据规模为 $\lfloor pN_b \rfloor$的情况定义为 LDS，其中 $1 \le p \le 0.5$。事实上，LDS 很容易出现，因为在求解和生成数据之前，并不知道给定方程所需的 N_b。一种稳妥的做法是尽可能确保求解器即使在 LDS 下也能保持足够好的求解性能，这正是本文的目标。实现这一目标的主要挑战如下：

(1) **泛化问题**。当训练数据量不足时，AHS 中的深度学习模块会受到固有泛化问题的影响（Zhang 等，2021年；Neyshabur 等，2017），即无法对训练数据分布之外的输入做出良好预测。这导致训练得到的相关项无法准确补偿低分辨率项造成的精度损失。

(2) **鲁棒性问题**。许多微分方程具有混沌性（Greydanus 等人，2019年），即对输入中的微小扰动敏感，这使得 AHS 仅凭有限数据难以从根本上对普遍存在且难以捉摸的混沌随机性进行建模，从而导致较差的泛化能力（Abu-Mostafa 等人，2012）。对于缺乏鲁棒性的 AHS，在式（2）所示的迭代求解过程中，误差很容易累积并传播，最终导致数值爆炸。

2.4. ResNet 的注意力机制

对于式（4）中提到的残差块，注意力机制[1] 可表述为

$$x_{t+1} = x_t + f(x_t|\theta_t) \odot \alpha_t, \tag{5}$$

其中 $\alpha_t = Q[f(x_t|\theta_t)]$ 是 t-个 注意力模块（Hu 等，2018年；Woo 等，2018年），而 $\odot$ 是逐元素乘法，如图所示1（b）。许多近期研究（Hu 等，2018年；Huang 等，2020；Woo 等，2018年；Wang 等人，2020；Liang 等人，2020；Zhong 等，2023年）证实了这些注意力模块在提升模型于**传统深度学习任务**中的泛化能力和鲁棒性方面的有效性，例如图像分类、检测、分割、风格迁移等。具体而言，

(1)**提升泛化能力**。注意力机制能够自适应地学习不同任务和输入数据中的特征权重（Qin 等人，2021年；Huang 等，2020；Wang 等人，2020）。这种灵活性使模型能够关注与任务相关的信息、适应新数据，并有效捕捉复杂的关系，最终提升各种传统深度学习任务的泛化性能。

(2)**增强鲁棒性**。理论上已证明，注意力机制能够缓解输入中的微小扰动，从而提升鲁棒性（Zhong 等，2023年）。具体而言，设 ϵ表示噪声引起的扰动，且满足 $\|x_0^\epsilon - x_0\| = \epsilon$。由 x_0^ϵ 作为输入所导致的第 L 层误差 $\|x_t^\epsilon - x_t\|$ 的上界为 $\epsilon \prod_{t=1}^{L-1}(1 + \alpha_t \|\theta_t\|_2)$，根据公式（5），其中注意力权重 α_t发挥自适应调控作用。在风格转换和噪声攻击等任务中的实验结果也支持这一观点（Liang 等人，2020；Zhong 等，2023年）。

3. 方法

如第 2.4 节所述，注意力机制有助于解决传统深度学习任务中的泛化和鲁棒性问题。正如第 2.3 节所指出的，这些问题恰好是有限数据场景下 AHS 面临的主要挑战。因此，我们自然会考虑利用前向数值求解器与 ResNet 之间的对应关系（如第 2.2 节所述），将注意力机制无缝迁移到数值求解器中。一些先前的研究也考虑使用注意力来辅助求解（Geneva 与 Zabaras，2022；Takamoto 等，2023年；Rodriguez-Torrado 等，2022；Hemmasian 与 Barati Farimani，2023年），**但这些研究并非针对 AHS，也未采用公式（5）**中的注意力机制。

（1）一种直接的基线方法（AttNS-m）。根据公式（5），采用注意力机制的直接数值求解可表示为公式（6），其中使用乘法算子。

$$\text{AttNS-m:}\quad \hat{\mathbf{u}}_{n+1} = \hat{\mathbf{u}}_n + \hat{S}\Delta t_c \odot Q[\hat{S}|\phi], \tag{6}$$

其中 $\hat{S} = S(\mathbf{f}, \hat{\mathbf{u}}_n,\ \Delta t_c)$ 以及 ϕ 是注意力模块 $Q[\cdot|\phi]$ 的可学习参数。训练损失定义为 ℓ_2估计轨迹 $\hat{\mathbf{u}} = [\hat{\mathbf{u}}_1, \ldots, \hat{\mathbf{u}}_N]$ 与数据集中的真实轨迹 $\mathbb{D}(\text{traj}(\mathbf{u}))$ 之间的平方距离，遵循先前研究（Dresdner 等人，2022）：

$$R_e = \lambda \cdot \|\hat{\mathbf{u}} - \mathbf{u}\|_2^2 / N, \tag{7}$$

[1]不同于基于 Transformer 的注意力机制（Vaswani 等，2017），本文所考虑的注意力机制用于 ResNet，例如 SENet（Hu 等，2018年）。

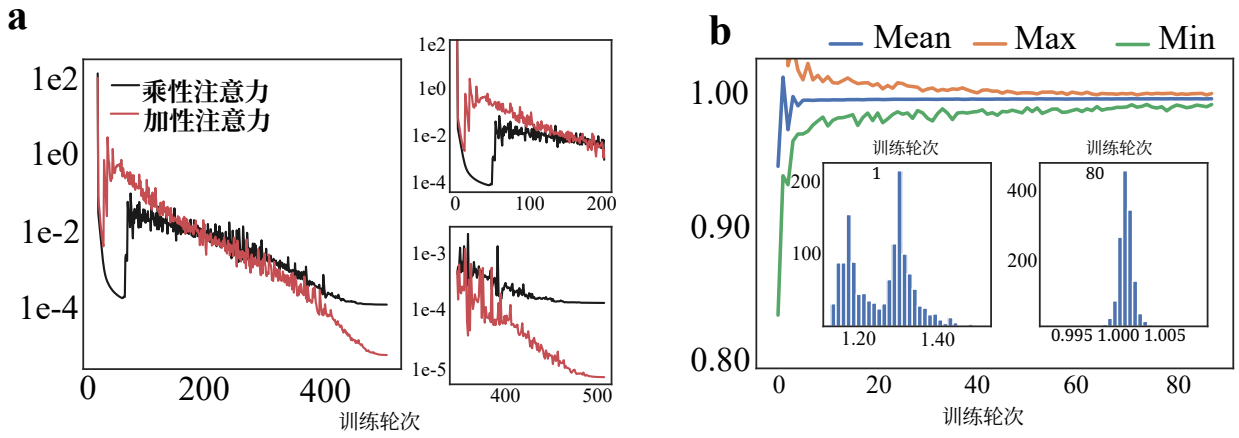


图 2： **a**。两种 AttNS-m 和 AttNS 的损失曲线。乘性注意力的损失最小化在开始时很快，随后在最后几个训练轮次中变慢，并在损失较大的情况下达到局部最小值； **b**。使用公式$Q[\hat{S}|\phi]$（蓝色）时注意力值的均值$S|\phi]$，其中使用公式（6），该均值在训练过程中迅速收敛到常数 1 附近。

其中 λ 是惩罚系数。 λ 可以缓解 R_e 的值因$\mathbf{u}$ 在某些特定微分方程中的量级过小而导致其过小的问题。这是因为在这种情况下，梯度会足够小，从而影响可学习参数 ϕ的优化。对于 $\hat{S} \in \mathbb{R}^d$，方程（Xue 等人，2023年）中的注意力模块架构为6）是

$$Q[\hat{S}|\phi] = \mathbf{W_h} \circ \mathbf{a} \circ \cdots \circ \mathbf{W_2} \circ \mathbf{a} \circ \mathbf{W_1}[\hat{S}], \quad (8)$$

其中 $\mathbf{a}$ 是有理激活函数（Boull´e 等，2020）；$\mathbf{W_i}$、$i = 2$、$\cdots$、$h-1$ 是 $d_1 \times d_1$ 矩阵，$\mathbf{W_h} \in \mathbb{R}^{d \times d_1}$和 $\mathbf{W_1} \in \mathbb{R}^{d_1 \times d}$。我们默认设置 $d_1 = 1024$ 和 $h = 2$。在第5节中，我们将进一步讨论公式（8）中的简洁设计如何针对 AHS 的高效求解进行定制：一方面，较少的参数数量能够实现更快的推理速度；另一方面，可以证明该架构具有理想的利普希茨连续性，为所提出求解器的收敛性分析提供了有力保障。此外，我们还将该注意力模块架构与其他变体进行比较。

（2）一种专为 AHS 设计的基线方法（AttNS）。事实上，在数学上，式（6）中 AttNS-m 的注意力采用乘法算子来校准低分辨率项带来的误差，这可能会影响精度。原因如下：（i）ˆ用于 Δt_c 中 S 离散化的步长通常不等于 AttNS-m 中的新步长 $\Delta t_c'$：$= \Delta t_c \odot Q[\hat{S}|\phi]$，因此不满足式（ˆ3）中的数学正确性；（ii）在 $Q[S|\phi]$的学习过程中，这种数学正确性还会迫使其花费一些优化工作，主要用于拟合一个所有元素均为 1 的常向量$\mathbf{I}$，从而对精度产生负面影响（Wang 等人，2022）。具体而言，在式（2）中，修正项旨在补偿低分辨率项的误差，因此其幅值

算法 1 AttNS 和 AttNS-m 的处理过程。

输入： 粗步长 Δt_c；满足评估时间 $T = N\Delta t_c$ 的步数 N；给定方程 $\mathbf{du}/\mathrm{d}t = \mathbf{f(u)}$，$\mathbf{u}(0) = \mathbf{c}_0$；给定数值积分方案 S；高质量数据集 $\mathbb{D}$(traj(u))。注意力模块 $Q[\cdot|\phi]$；学习率 η。

输出： 。学习到的注意力模块 $Q[\cdot]$

对于从 0 到 Total epoch 的每个训练轮次，执行 ▷ 估计轨迹 样本 $\mathbf{u} = [\mathrm{u}_0, \mathrm{u}1, ..., \mathrm{u}_N] \sim \mathbb{D}$(traj(u))； ˆ$\mathrm{u}_0 \leftarrow \mathrm{u}_0$；对于 t 从 0 到 $N-1$，执行 ˆ 计算积分项 $S \leftarrow S$(f, $\mathrm{u}_t, \Delta t_c$)； ˆ 计算注意力项 $Q[S|\phi]$；根据 Eq.(6)（AttNS-m）或 Eq.(10)（AttNS）得到 ˆu_{t+1}； 结束循环▷ 更新注意力模块 ˆ$\mathbf{u} \leftarrow$ [ˆ$\mathrm{u}1, ..., \hat{}\mathrm{u}_N$]；根据 Eq.(7) 计算损失 R_e；通过 $\phi \leftarrow \phi - \eta\nabla_\phi R_e$更新参数 ϕ；结束循环 返回注意力模块 $Q[\cdot|\phi]$

补偿项 ϵ_c 会很小。因此，如果使用乘法ˆ ˆˆ注意力，则有 $S\Delta t_c + \epsilon_c = S\Delta t_c \odot Q[S|\phi]$，即 $(Q[\hat{S}|\phi] - \mathbf{I}) \hat{} S\Delta t_c = \epsilon_c$。由于 $\|\hat{S}\Delta t_c\| \gg \|\epsilon_c\|$，我们有 $Q[\hat{S}|\phi] \approx \mathbf{I}$。图 2b 也支持这一分析。

为缓解这一问题，一种简单而有效的策略是 ˜ ˆ将注意力归一化为 $\mathbf{I} + Q[S|\phi]$ 并仅学习残差部分 $\widetilde{Q}[\hat{S}|\phi]$，我们可以将公式（6）改写为

$$\begin{aligned}\hat{\mathbf{u}}_{n+1} &= \hat{\mathbf{u}}_n + \hat{S}\Delta t_c \odot \left(\mathbf{I} + \widetilde{Q}[\hat{S}|\phi]\right) \\ &= \hat{\mathbf{u}}_n + \hat{S}\Delta t_c \underbrace{+\hat{S}\Delta t_c \odot \widetilde{Q}[\hat{S}|\phi]}_{\text{Additive attention}},\end{aligned} \quad (9)$$

其中，ˆ$S\Delta t_c \odot \widetilde{Q}[\hat{S}|\phi]$ 这一项可视为加性

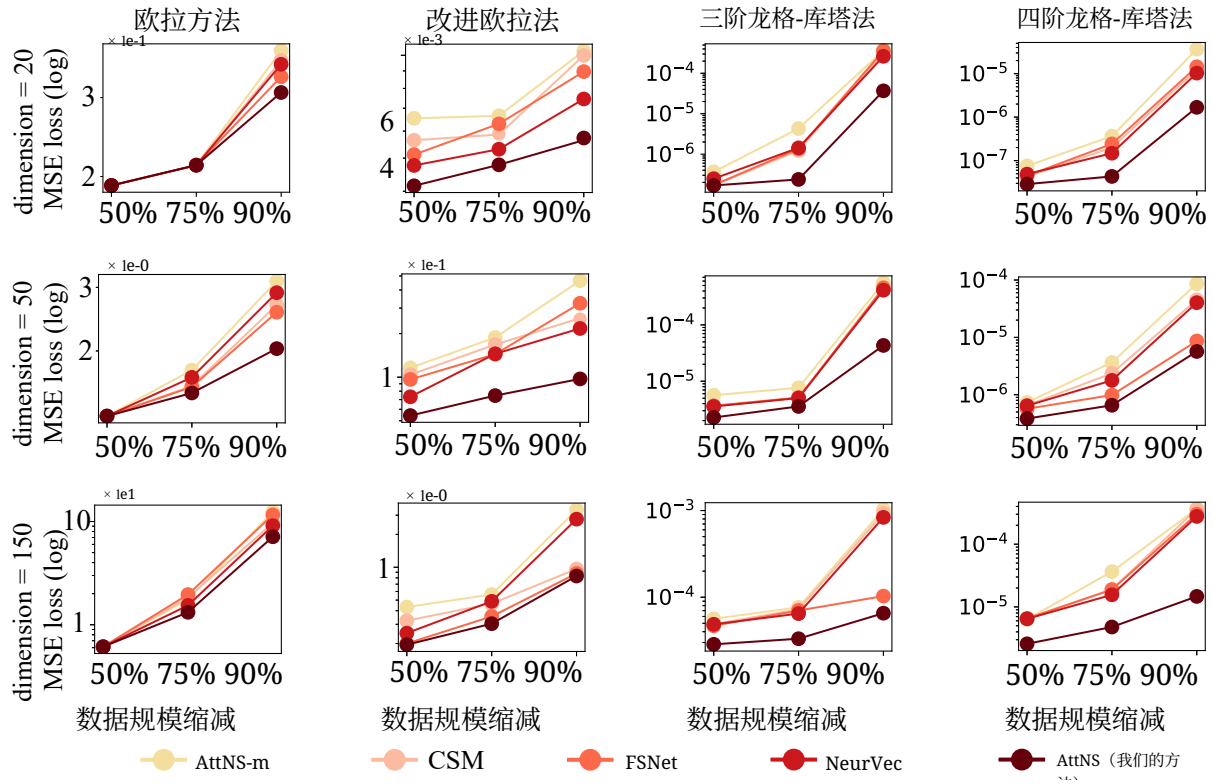


图 3：不同维度的弹簧-质量系统上，四种前向数值求解器对应的各种 AHS 的结果。“50%”表示训练数据量减少了 50%。**损失越小，性能越好**。

注意力（Wu 等人，2021年；Cheuk 等，2021年；Huang 等，2022a；Li 等人，2021a；Gao 等人，2021年）修正项。简而言之，我们将 AttNS 的迭代公式设为

$$\text{AttNS:}\quad \hat{\mathbf{u}}_{n+1} = \hat{\mathbf{u}}_n + \hat{S}\Delta t_c + Q[\hat{S}|\phi]. \tag{10}$$

根据公式（9），如图2a 所示，AttNS 在保证数学正确性的同时，也保留了注意力相较于 AttNS-m 的优势。从公式上看，AttNS 与一般 AHS 的主要区别在于输入：前者为 $S(\mathbf{f}, \hat{}\mathbf{u}_n, \Delta t_c)$，后者为$\hat{}\mathbf{u}_n$。事实上，仅凭这一差异，AttNS 就足以取得更好的泛化误差并确保鲁棒性，详见第 5节。此外，注意力输入带来的性能差距在传统深度学习任务中也已有观察（Guo 等，2020；Hu 等，2018；Huang 等，2023a），这进一步验证了 AttNS 背后的原理。我们在算法 1 中总结了 AttNS 和 AttNS-m。

4. 实验

本节从两个角度验证所提出的 AttNS 的有效性：（1）在不同数值求解器上的表现；（2）在不同微分方程基准上的表现。具体而言，由于 AttNS 是一种 AHS，我们采用多种常用前向数值求解器作为骨干，以评估 AttNS 带来的提升，包括欧拉方法、改进欧拉法、三阶和四阶龙格-库塔法（详情见附录）。另一方面，为了证明 AttNS 能够胜任复杂微分方程，我们还在四阶龙格-库塔法上对两个混沌动力系统进行了实验，即 k连杆摆和弹性摆。本节所有实验均遵循（Huang 等，2022c；Chen 等人，2020）中的设置，以公平比较各项初始条件生成方法和指标。本节将 AttNS、AttNS-m、NeurVec（Huang 等，2023b）、CSM（Dresdner 等人，2022）和 FSNet（Xu 等人，2023年）作为 AHS 的基线方法。

(1) **关于不同数值求解器**。本节中，我们考虑一个高维线性系统，即弹簧-质量系统，以及四种前向数值求解器（详情见附录）。在弹簧-质量系统中，有 d个质量块和 $d+1$ 个弹簧依次连接，并水平放置，两端分别连接到两个固定块。该系统对应的常微分方程为

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} q_i \\ p_i \end{pmatrix} = \begin{pmatrix} p_i/m \\ k_i\left(q_{i-1}-q_i\right)+k_{i+1}\left(q_{i+1}-q_i\right) \end{pmatrix}, \tag{11}$$

$i=1,\ 2,\ \cdots,\ d,\ q_0 = q_{d+1} = 0$，其中 m_i和 k_i分别是

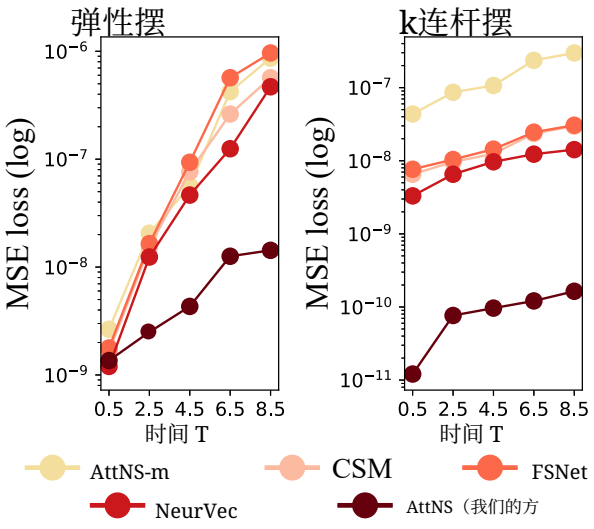


图4：步长为 $1e-1$ 时在不同混沌系统上的模拟。测试集中 (a) 摆和 (b) 弹性摆的均方误差（MSE）损失。

第 t 个质量块的质量和第 t 根弹簧的力系数。第 t 个质量块的动量和位置分别表示为 p_i 和 q_i。我们为数值求解器采用粗步长 $\Delta t = 2e-1$，并使用细步长 $1e-3$ 训练 AHS。

评估时间的实验结果 T=20 如图3所示。对于模拟精度较低的欧拉方法，我们的 AttNS 在不同维度的弹簧-质量系统上取得了与最先进水平（SOTA）的 AHS 相当的稳定性能。对于其他精度更高的数值求解器，AttNS 对求解器的提升效果优于其他方法。即使对于维度不断增加的弹簧-质量系统，尽管模拟难度随之提高，AttNS 仍能保持其性能。此外，我们观察到，我们的 AttNS 使用较少的数据也能达到与最先进水平的 AHS 相近的性能，这表明我们能够利用少量数据训练出高效且稳健的 AHS。在大多数不同设置下，AttNS-m 的表现较差，这与第3节

(2)**关于不同的混沌系统**。我们将训练数据量减少 50%，并使用两个混沌动力系统验证 AttNS 的有效性，即弹性摆和 K-link 摆。弹性摆由一个无体积的球体和一根弹性杆组成。在重力和弹簧力的作用下（Breitenberger 与 Mueller，1981），球体的运动会呈现混沌，其常微分方程为

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix}\theta\\ r\\ \dot\theta\\ \dot r\end{pmatrix} = \begin{pmatrix}\dot\theta\\ \dot r\\ \frac{1}{r}(-g\sin\theta - \dot\theta\dot r)\\ r\dot\theta^2 - \frac{k}{m}(r-l_0) + g\cos\theta\end{pmatrix}, \tag{12}$$

其中 k、m、l_0 和 g 是相关常数。有两个变量 θ 和 r 在公式（12）中。具体而言，r 是弹簧的长度，而 θ 是弹簧与竖直轴之间的角度。对于k连杆摆，它考虑了 K 个首尾相连的球体，并由 K 根杆连接，在重力作用下（Lopes & Tenreiro Machado，2017），其常微分方程为

$$\mathrm{d}(\theta,\dot\theta)/\mathrm{d}t = (\dot\theta, \mathbf{A}^{-1}\mathbf{b}) \tag{13}$$

其中 $\theta = (\theta_1, \theta_2, \cdots, \theta_K)$，$\theta_i$ 表示i^{th}杆与竖直轴之间的夹角。令 $\mathbf{b} = (b_1, b_2, \cdots, b_K)$，并令$b_i = -\sum_{j=1}^{K}\left[c(i,j)\,\dot\theta_j^2 \sin(\theta_i - \theta_j)\right] - (K-i+1)g\sin\theta_i$。$\mathbf{A}$ 是一个 $K\times K$ 矩阵，$\mathbf{A}_{i,j}$ 中的元素为 $[K - \max(i,j) + 1]\cos(\theta_i - \theta_j)$。仿真结果如图 4 所示。从图 4 可以看出，即使对于更难求解的混沌系统，所提出的 AttNS 在有限数据场景下仍能保持更高的解精度，并优于其他 AHS。

5. 讨论

(1) **AttNS 的收敛性分析**。AttNS 是一种数值求解器，因此我们需要估计其数值收敛性。现在以步长为 $k\Delta t$ 的 AttNS 和欧拉方法为例，在定理 5.1 中对所提出的 AttNS 进行收敛性分析。

定理 5.1。考虑常微分方程 $d\mathbf{u}/dt = \mathbf{f}(\mathbf{u}), \mathbf{u}(0) = \mathbf{c}_0$以及欧拉方法 $\mathbf{u}_{n+1} = \mathbf{u}_n + \Delta t\mathbf{f}(\mathbf{u}_n)$。我们作如下假设：*(1)* $\mathbf{f}$ 是 *Lipschitz* 连续的，*Lipschitz* 常数为L；*(2)* 真实解 $\mathbf{u}$ 的二阶导数一致有界于 $M>0$，即 $\|\mathbf{u}''\|_\infty \le M$在 $[0,T]$上。此外，我们假设 *AttNS* 中的注意力模块 $Q[\cdot]$ 是 *Lipschitz* 连续的，*Lipschitz* 常数为 $k\Delta t L_{att}$。对于步长为 $k\Delta t$的 *AttNS* 解 $\hat{}\mathbf{u}$，有

$$|\hat{\mathbf{u}}_N - \mathbf{u}(T)| \le \alpha\Delta t + \beta\sqrt{\delta}, \tag{14}$$

其中 $\alpha = \frac{1}{2L}M\exp(2TL)$、$\beta = \sqrt{T\exp(TL(1+L_{att}))L(1+L_{att})}$，$\delta$是关于训练损失 R_e 的误差项。如果 *AttSlover* 能很好地拟合训练数据，即 $R_e \to 0$，则误差 $\delta \to 0$。

Proof. (See Appendix C). □

事实上，定理5.1 的条件较为宽松。首先，注意力模块 Q 的 Lipschitz 条件由第 8 节中的公式（5 （2）分析给出，我们证明了该条件很容易满足。其次，对于多个物理系统，可以直接验证真实解 $\mathbf{u}$ 的有界性（见附录）。定理 5.1 还可用于分析 AttNS 的收敛性。例如，一些已有研究（Du 等人，2019年；Jacot 等人，2018年）表明，在某些宽松条件下，梯度下降可以使神经网络收敛到全局最优

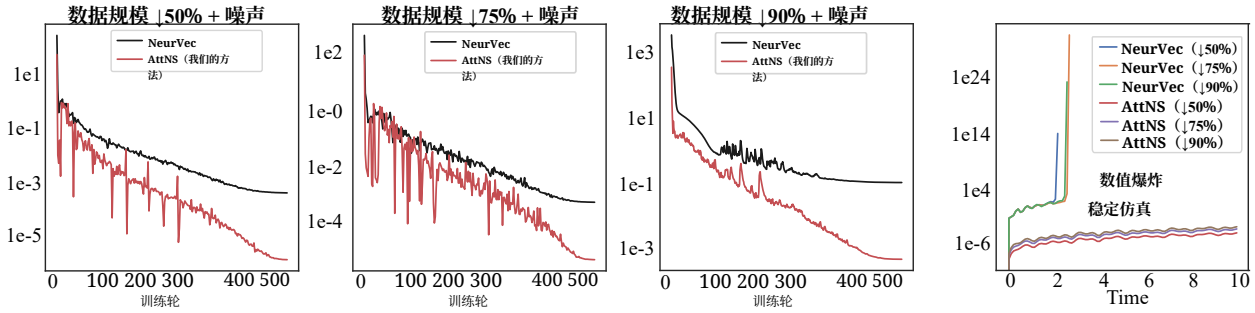


图 5：不同数据规模下弹性摆的噪声攻击实验。与其他 AHS 相比，采用注意力机制的 AttNS 能更好地减轻噪声的不利影响。

表 1：注意力模块的深度 h 和宽度 d_1 对仿真性能的影响。相对推理速度是指基于我们提出的 AttNS 设置（即 $h=2$ 和 $d_1=1024$）的网络相对推理速度。

	弹簧-质量系统		k连杆摆		弹性摆	
	均方误差损失	相对推理速度	均方误差损失	相对推理速度	均方误差损失	相对推理速度
$h=2$（我们的方法）	2.58e-6	-	4.26e-9	-	5.41e-7	-
$h=3$	2.49e-6（↑3.61%）	↓88.49%	4.39e-9（↓2.52%）	↓91.60%	5.28e-7（↑2.46%）	↓90.89%
$h=4$	2.55e-6（↑1.18%）	↓93.95%	4.42e-9（↓3.62%）	↓95.64%	5.58e-7（↓3.05%）	↓95.33%
$d_1=512$	3.76e-5（↓83.36%）	↑27.59%	2.56e-8（↓93.14%）	↑15.36%	2.56e-6（↓78.87%）	↑17.25%
$d_1=1024$（我们的方法）	2.58e-6	-	4.26e-9	-	5.41e-7	-
$d_1=2048$	2.41e-6（↑7.05%）	↓26.48%	3.98e-9（↑7.03%）	↓23.12%	4.83e-7（↑12.01%）	↓24.63%

解，此时公式（ R_e 7）中的损失 $\delta \to 0$趋于 0。根据定理 5.1，我们有 $\delta \to 0$,

$$|\hat{\mathbf{u}}_N - \mathbf{u}(T)| \leq \underbrace{|\mathbf{u}_{Nk} - \mathbf{u}(T)|}_{\leq \Delta t \cdot M \cdot \exp(2TL)/2L} + \underbrace{|\hat{\mathbf{u}}_N - \mathbf{u}_{Nk}|}_{\leq \beta\sqrt{\delta} \to 0}. \quad (15)$$

因此，$|\hat{}\mathbf{u}_N - \mathbf{u}(T)| = \mathcal{O}(\Delta t)$。在这种情况下，步长为 $k\Delta t$ 的 AttNS 可以达到与步长为 Δt的欧拉方法相同的精度，其全局截断误差也为 $\mathcal{O}(\Delta t)$ (Butcher, 2016年）。然而，在这种情况下，AttNS 的评估速度约为欧拉方法的 $\mathcal{O}(k)$ 倍。

(2) **网络架构** $Q[\hat{}S|\phi]$。我们设计$\hat{}$注意力模块结构 $Q[S|\phi]$，如式（8）所示，主要有以下原因：一方面，它的参数数量较少，推理速度更快，从而保证了计算效率。例如，在式（2）中，如果数据$\mathbb{D}_f$ 由细步长 Δt_f生成，那么理论上，其评估速度是经典方法式（ $\mathcal{O}(\Delta t_c/[(1+\epsilon)\Delta t_f]$ 倍（Huang 等， 2022c），其中$\epsilon > 0$ 与神经网络的推理速度有关。因此，AHS 中神经网络的推理速度十分重要。

另一方面，定理 5.2 可以证明，该架构具有理想的利普希茨连续性，这为定理 5.1中所述 AttNS 的收敛性分析提供了有力保证。我们将在第6节将这种注意力模块结构与其他变体进行比较。

定理 5.2。 对于 $x \in$ 定义域 A，注意力模块 $Q(x) = \mathbf{W_2} \circ \mathbf{a} \circ \mathbf{W_1} x$ 是 *Lipschitz* 连续的。$\mathbf{a}$ 是有理激活函数，即 $\mathbf{a}(x) = \frac{\sum_{i=1}^{3} a_i x^i}{\sum_{i=1}^{2} b_i x^i}$，其中 $a_i, b_i \in \mathbb{R}$。

证明。 (参见附录 D)。 □

(3) **AttNS 对泛化的有效性** 。现在我们进一步讨论数据规模，定理5.3 在欧拉方法和一些温和假设下给出了一些分析。定理 5.3 基于 Vapnik-Chervonenkis 理论（Vapnik， 1999）。与一般 AHS 相比，AttNS 达到相同的泛化误差所需的数据规模更小，这与图3中的实验结果一致。

定理 5.3。 设 $Net(\hat{}S|\phi, \mathbb{D}_f)$ 和 $Net(\mathbf{u}_n|\phi, \mathbb{D}_f)$分别为一般 *AHS* 和 *AttNS* 的修正项，其中$\hat{}S = S(\mathbf{f}, \mathbf{u}_n, \Delta t_c)$。对于 $\epsilon > 0$，当数据规模大于 N'时，两种方法的经验误差满足$R_e(\phi|AHS) \leq \epsilon$ 和 $R_e(\phi|AttNS) \leq \epsilon$。对于足够小的$\epsilon_0 \ll \epsilon$且采用欧拉方法时，我们有

$$N(AttNS) \lesssim N(AHS), \quad (16)$$

其中， N（$*$）是方法 $*$ 的泛化误差可以达到 $\epsilon(1-\epsilon_0)^{-1}$时的数据规模下界。

表2：（左）跳跃连接以及将 Δt_c 视为输入的一部分所产生的影响。（右）其他 ˆ 类型网络架构在 $Q[S|\phi]$ 上的性能。

基准	使用 跳跃连接	不使用 Δt_c	Ours
弹簧-质量系统	4.57e-4（↓ 99.44%）	3.67e-6（↓ 29.78%）	**2.58e-6**
K-link	6.36e-6（↓ 99.93%）	**4.18e-9**(↑ 1.91%)	4.26e-9
弹性	1.39e-4（↓ 99.61%）	1.65e-6（↓ 67.21%）	**5.41e-7**

模块	MSE（弹性）	MSE（k-link）
LSTM（Hochreiter 和 Schmidhuber，1997）	7.21e-06	1.35e-08
Transformer（Liu 等，2021年）	3.64e-04	2.65e-06
Ours	**5.41e-07**	**4.26e-09**

证明。（见附录E）。 □

尽管我们提出的方法确实减少了最先进水平方法所需的数据规模，但其准确性可能不足以解决自然科学和工程领域中的所有问题。因此，如果我们需要解决一个要求高准确度的问题，最好尽可能增加训练数据量，以进一步提升人工智能增强求解器的性能，而不是仅仅依赖算法设计。这是因为我们观察到，在使用完整训练数据时，AttNS 带来的改进可能不到 10%-15%；这是合理的，因为 AttNS 专为数据不足的情形设计，而我们并未明确地在数据充足时最大限度地提升 AttNS 的性能。未来，我们将探索如何改进 AttNS，使其适用于数据充足的情形。

(4) **AttNS 的鲁棒性效果**。我们提出的 AttNS 方法受 ResNet 中注意力机制的启发。在图 5中，我们通过不同数据规模下的噪声攻击实验，实证探究 AttNS 的鲁棒性。向输入中注入噪声，是添加微小扰动以帮助我们观察模型鲁棒性的一种明确方式。具体而言，在训练过程中，我们可以通过添加常数噪声 $\sigma = 1e-5$ 到 $\hat{\mathbf{u}}_n$，即 $\hat{\mathbf{u}}_n \leftarrow \hat{\mathbf{u}}_n + \sigma$，作用于所有 n，见公式（10）。实验结果表明，与基线方法 AHS（即 NeurVec）相比，我们提出的方法能够显著调节噪声，从而降低训练损失并实现稳定模拟。相反，在所有设置下，NeurVec 都会因噪声的影响而在模拟开始时出现数值爆炸。

上述实验表明，所提出的 AttNS 足够稳健，即使在有限数据场景下也能保持解的稳定。

(5) **局限性** 。根据论文中的实验结果，所提出的 AttNS 使用较少的训练数据也能达到足够好的仿真性能。然而，AttNS 无法完全避免数据集中的固有噪声对解造成干扰。因此，我们旨在缓解噪声问题，而不是彻底解决它。若要进一步缓解这一问题，可能需要更精细的网络结构和训练设置，以提升 AttNS 的有效性。

本文的实验仅考虑常微分方程（ODE），未考虑偏微分方程（PDE）的基准，这是因为常微分方程更有利于本文中的一些理论分析，而复杂的偏微分方程则不易分析。幸运的是，AHS 也可用于偏微分方程求解（Dresdner 等人，2022），因此，未来可以将 AttNS 用于偏微分方程求解。

6. 消融研究

本节中，我们对 AttNS 进行了若干项消融研究。在表1中，我们分析了式（ h 中注意力模块的深度d_1 和宽度的影响。8我们发现，在所有基准中，深度 h对均方误差损失的影响很小，但模型性能与宽度 d_1呈正相关。因此，为了提高推理速度，我们为 AttNS 选择了足够浅的深度 $h = 2$ 和合适的宽度 $d_1 = 1024$。

接下来，我们研究了适用于 AttNS 的常见残差结构。从表2 （左）可以看出，对于高维线性系统和混沌动力系统，跳跃连接都会对模拟产生显著的负面影响。这促使我们为 AttNS 采用式（ 8 ）所示的简单堆叠结构。此外，我们还分析了ˆ注意力模块的输入形式，其中我们将 S 作为输入，而非式（ˆ中的完整积分项 $S\Delta t_c$。8）

ˆ在表2中，我们展示了输入 S 的性能优于ˆ输入 $S\Delta t_c$，k连杆摆除外（性能相近）。此外，表2（右）展示了其他网络架构的性能，ˆ并突显了式（ $Q[S|\phi]$中定义的注意力模块结构的有效性8）。

7. 结论

本文探讨了如何改进 AHS，使其在有限数据场景下也能实现有效计算。基于 ResNet 的动力系统视角，我们将注意力机制引入数值求解器，并提出 AttNS，以应对数据有限时的泛化与鲁棒性问题。实验结果表明，AttNS 能有效改进多种数值求解器；我们还分析了其收敛性、泛化能力和鲁棒性。

致谢

本研究得到国家科技重大专项（No.2021ZD0111601）、国家自然科学基金（No.62325605, No.62272494）、广东省基础与应用基础研究基金（No.2023A1515011374）以及广州市科技计划（No.2024A04J6365）的资助。

影响声明

本文旨在推动科学机器学习领域的发展。我们的工作可能对物理学、化学等领域产生诸多影响。

参考文献

Abu-Mostafa, Y. S., Magdon-Ismail, M., and Lin, H.-T.从数据中学习，第 4 卷。AMLBook，纽约，2012 年。

Bar-Sinai, Y., Hoyer, S., Hickey, J., and Brenner, M. P. 学习偏微分方程的数据驱动离散化。美国国家科学院院刊，116(31):15344–15349，2019年。

Boull´e, N., Nakatsukasa, Y., and Townsend, A. 有理神经网络。神经信息处理系统进展，33:14243–14253，2020。

Breitenberger, E. and Mueller, R. D. 弹性摆：一种非线性范式。数学物理学杂志，22(6):1196–1210，1981。

Bruno, O. P., Hesthaven, J. S., and Leibovici, D. V. 基于 Fc 的激波动力学求解器，采用神经网络局部化人工黏性分配。计算物理学杂志：*X*，15:100110，2022。

Butcher, J. C. 常微分方程的数值方法。约翰·威立父子公司，2016年。

Cai, S., Mao, Z., Wang, Z., Yin, M., and Karniadakis,G. E. 流体力学中的物理信息神经网络（PINN）：综述。力学学报，37(12):1727–1738，2021年。

Chang, B., Meng, L., Haber, E., Tung, F., and Begert, D. 从动力系统视角看多层残差网络。*arXiv*预印本 *arXiv:1710.10348*，2017。

Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. 神经常微分方程。神经信息处理系统进展，31，2018年。

Chen, Z.、Zhang, J.、Arjovsky, M. 和 Bottou, L. 辛几何递归神经网络。载于《国际学习表征会议》，2019年。

Chen, Z.、Zhang, J.、Arjovsky, M. 和 Bottou, L. 辛几何递归神经网络。载于 国际学习表征会议，2020。网址：https://openreview.net/forum?id=BkgYPREtPr。Cheuk, K. W.、Luo, Y.-J.、Benetos, E. 和 Herremans, D. 使用加性注意力重新审视起音与帧模型。*2021*年国际神经网络联合会议（*IJCNN*），第1–8页，2021年。Choudhary, A.、Lindner, J. F.、Holliday, E. G.、Miller, S. T.、Sinha, S. 和 Ditto, W. L. 物理增强神经网络学习有序与混沌。《物理评论*E*》，101（6）：062207，2020年。Dahlquist, G. 常微分方程数值积分中的收敛性与稳定性。《斯堪的纳维亚数学》，第33–53页，1956年。

Dresdner,G., Kochkov, D.,Norgaard, P. C.,Zepeda-Nunez, L., Smith, J., Brenner, M., and Hoyer, S. 学习修正用于模拟湍流的谱方法。《机器学习研究汇刊》，2022。

Du, S. S.、Zhai, X.、Poczos, B. 和 Singh, A. 梯度下降能够对过参数化神经网络进行可证明的优化。载于国际学习表征会议，2019年。网址：https://openreview.net/forum?id=S1eK3i09YQ。

Dupont, E.、Doucet, A. 和 Teh, Y. W. 增广神经常微分方程。神经信息处理系统进展，32，2019年。

Ferziger, J. H.、Peri´c, M. 和 Street, R. L. 流体动力学计算方法，第3卷。Springer，2002。

Frezat, H.、Le Sommer, J.、Fablet, R.、Balarac, G. 和 Lguensat, R. 准地转湍流的后验学习参数化。地球系统建模进展杂志，14(11)：e2022MS003124，2022。

Gao、C.、Ye、H.、Cao、F.、Wen、C.、Zhang、Q. 和 Zhang、F. 采用加性通道-空间注意力的多尺度融合网络用于图像分割。*Knowl. Based Syst.*，214:106754，2021年。

Geneva、N. 和 Zabaras、N. 用于物理系统建模的 Transformer。*NeuralNetworks*，146:272–289，2022。

Greydanus、S.、Dzamba、M. 和 Yosinski、J. 哈密顿神经网络。*AdvancesinNeural InformationPro-cessing Systems*，32，2019年。

Guo, J., Ma, X., Sansom, A., McGuire, M., Kalaani, A., Chen, Q., Tang, S., Yang, Q., 和 Fu, S. Spanet：用于增强图像识别的空间金字塔注意力网络。发表于*2020 IEEE International Conference on Multimedia and Expo (ICME)*，第 1–6 页。IEEE，2020。

He, K., Zhang, X., Ren, S., 和 Sun, J. 用于图像识别的深度残差学习。发表于计算机视觉与模式识别，2016年。

Hemmasian, A. 和 Barati Farimani, A. 使用 Transformer 对流体流动进行降阶建模。《流体物理》，35(5)，2023年。

Hochreiter, S. 和 Schmidhuber, J. 长短期记忆。《神经计算》，9:1735–1780，1997。

Hu, J.、Shen, L. 和 Sun, G. Squeeze-and-excitation 网络。收录于*IEEE*计算机视觉与模式识别会议论文集，第7132–7141页，2018年。

Huang, W.、Xu, H. 和 Yu, Y. Mrp-net：一种基于修正递归图和加性注意力卷积神经网络的癫痫发作检测方法。*SSRN*电子期刊，2022a。

Huang, Z.、Liang, S.、Liang, M. 和 Yang, H. Dianet：密集与隐式注意力网络。收录于*AAAI*人工智能会议论文集，第34卷，第4206–4214页，2020。

Huang, Z., Liang, S., Liang, M., He, W., and Lin, L. 面向动力系统视角的逐层共享注意力网络。*arXiv* 预印本 *arXiv:2210.16101*，2022b。

Huang, Z., Liang, S., Zhang, H., Yang, H., and Lin, L. 通过 NeurVec 加速大规模动力系统模拟的数值求解器。*arXiv* 预印本 *arXiv:2208.03680*，2022c。

Huang, Z., Liang, M., Qin, J., Zhong, S., and Lin, L. 从动力系统视角理解自注意力机制。载于 *IEEE/CVF* 国际计算机视觉大会（*¡strong¿ ICCV¡/strong¿*）论文集，2023年a。

Huang, Z., Liang, S., Zhang, H., Yang, H., and Lin, L. 使用神经向量增强数值求解器快速模拟动力系统。《科学报告》，13(1)：15254，2023年b。

Jacot, A.、Gabriel, F. 和 Hongler, C. 神经切线核：神经网络中的收敛性与泛化。神经信息处理系统进展，31，2018年。

Ji, W.、Qiu, W.、Shi, Z.、Pan, S. 和 Deng, S. Stiff-PINN：用于刚性化学动力学的物理信息神经网络。《物理化学杂志*A*辑》，125(36)：8098–8106，2021年。

Kidger, P., Chen, R. T., 和 Lyons, T. J. “嘿，这不是常微分方程”：通过半范数加速常微分方程伴随。在国际机器学习会议，第5443–5452页，2021年。

Kochkov, D., Smith, J. A., Alieva, A., Wang, Q., Brenner, M. P., 和 Hoyer, S. 机器学习加速的计算流体力学。美国国家科学院院刊，118(21):e2101784118，2021年。Langley, P. 撰写机器学习论文。载于 Langley, P.（编），第*17*届国际机器学习会议（*ICML 2000*）论文集，第1207–1216页，斯坦福，加利福尼亚州，2000年。Morgan Kaufmann。Lehtonen, J. Lambert W 函数在生态与进化模型中的应用。《生态学与进化方法》，7(9):1110–1118，2016年。

Li, X.、Xu, Q.、Chen, X. 和 Li, C. 面向基于卷积神经网络的分类的加性注意力。*2021*年 *IEEE* 国际机电一体化与自动化会议（*ICMA*），第55–59页，2021a。

Li, Z.、Kovachki, N. B.、Azizzadenesheli, K.、liu, B.、Bhattacharya, K.、Stuart, A. 和 Anandkumar, A. 用于参数化偏微分方程的傅里叶神经算子。载于 国际学习表征会议，2021b。网址：https://openreview.net/forum?id=c8P9NQVtmnO。

Liang, S., Huang, Z., Liang, M. 和 Yang, H. 实例增强批归一化：批噪声的自适应调节器。发表于*AAAI*人工智能会议论文集，第 34 卷，第 4819–4827 页，2020。

Liang, S., Jiang, S. W., Harlim, J. 和 Yang, H. 利用机器学习在未知流形上求解偏微分方程。*arXiv*预印本 *arXiv:2106.06682*，2021。Liang, S., Huang, Z. 和 Zhang, H. 用于学习哈密顿系统的刚性自适应神经网络。发表于国际学习表征会议，2022。网址：https://openreview.net/forum?id=uVXEKeqJbNa。

List, B., Chen, L.-W. 和 Thuerey, N. 利用可微流体求解器进行学习式湍流建模：基于物理的损失函数与优化时域。《流体力学杂志》，949:A25，2022。

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin Transformer：使用移位窗口的分层视觉变换器。发表于*IEEE/CVF*国际计算机视觉会议论文集，第 10012–10022 页，2021年。

Lopes, A. M. and Tenreiro Machado, J. N 连杆摆的动力学：分数阶视角。《*International Journal of Control*》，90(6):1192–1200，2017。

Lu, L., Jin, P., Pang, G., Zhang, Z. 和 Karniadakis, G. E. 基于

算子的通用逼近定理，通过 DeepONet 学习非线性算子。*NatureMachineIntelligence*，3(3):218–229，2021年。

Lu, Y., Zhong, A., Li, Q., 和 Dong, B. 超越有限层神经网络：连接深度架构与数值微分方程。载于 Dy, J. G. 和 Krause, A.（编），第*35*届国际机器学习会议论文集，*ICML 2018*，第80卷，机器学习研究论文集，第3282–3291页。PMLR，2018年。

Meunier, L., Delattre, B. J., Araujo, A., 和 Allauzen, A. Lipschitz 神经网络的动力系统视角。载于国际机器学习会议，第15484–15500页。PMLR，2022。

Mishra, S. 一种用于加速微分方程计算的数据驱动机器学习框架。*arXiv* 预印本 *arXiv:1807.09519*，2018年。

Neyshabur, B., Bhojanapalli, S., McAllester, D., 和 Srebro, N. 探索深度学习中的泛化。神经信息处理系统进展，30，2017。

Qin, Z., Zhang, P., Wu, F., 和 Li, X. FCANet：频率通道注意力网络。载于*IEEE/CVF*国际计算机视觉会议论文集，第 783–792 页，2021年。

Queiruga, A. F., Erichson, N. B., Taylor, D., 和 Mahoney, M. W. 深度连续神经网络。*arXiv* 预印本 *arXiv:2008.02389*，2020。

Raissi, M., Perdikaris, P., 和 Karniadakis, G. E. 物理信息神经网络：一种用于求解涉及非线性偏微分方程的正问题和反问题的深度学习框架。《计算物理学杂志》，378：686–707，2019年。

Rodriguez-Torrado, R., Ruiz, P., Cueto-Felgueroso, L., Green, M. C., Friesen, T., Matringe, S., 和 Togelius, J. 用于双曲型偏微分方程的物理信息注意力神经网络：在 Buckley–Leverett问题中的应用。《科学报告》, 12(1): 7557, 2022。

San, O. 和 Maulik, R. 用于湍流地球物理流动降阶建模的极限学习机。*PhysicalReviewE*, 97(4):042322, 2018年。

Shampine, L. F. 常微分方程的数值解。Routledge, 2018年。

Subel, A., Chattopadhyay, A., Guan, Y. 和 Hassanzadeh, P. 利用深度学习对受迫伯格斯湍流进行数据驱动的子网格尺度建模，并通过迁移学习泛化至更高雷诺数。《流体物理》，33(3)，2021年。

Takamoto, M., Alesiani, F. 和 Niepert, M. 通过参数引导的通道注意力学习神经偏微分方程求解器。*arXiv* 预印本 *arXiv:2304.14118*，2023年。Um, K., Brand, R., Fei, Y. R., Holl, P. 和 Thuerey, N. 求解器在环：从可微物理中学习，以与迭代偏微分方程求解器交互。神经信息处理系统进展，33:6111–6122，2020。Vapnik, V. 统计学习理论的本质。Springer 科学与商业媒体，1999年。

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, N., Jones, L., Gomez, A. N., Kaiser, Ł., 和 Polosukhin, I. 注意力就是你所需要的一切。神经信息处理系统进展, 30, 2017。

Wang, Q., Wu, B., Zhu, P., Li, P., and Hu, Q. Eca-net：用于深度卷积神经网络的高效通道注意力。在计算机视觉与模式识别会议，2020。

Wang, X., Xue, W., Han, Y., and Yang, G. 通过机器学习方法实现高效气候模拟。*arXiv*预印本 *arXiv:2209.08151*，2022年。

Weinan, E. 关于通过动力系统进行机器学习的提议。《数学与统计学通讯》，1(5):1–11，2017。

Woo, S., Park, J., Lee, J.-Y., and Kweon, I. S. Cbam：卷积块注意力模块。在欧洲计算机视觉会议，第3– 19页，2018年。

Wu, C., Wu, F., Qi, T., and Huang, Y. Fastformer：加性注意力就是你所需要的一切。*ArXiv*，abs/2108.09084，2021年。

Xu, Y., Liu, S. 和 Lin, L. Fsnet：一种用于混沌动力系统的快速模拟算法。2023年。网址 https://github.com/small-whirlwind/FSNet。

Zhang, C., Bengio, S.,Hardt, M., Recht, B. 和 Vinyals, O. 理解深度学习（仍然）需要重新思考泛化。《美国计算机学会通讯》，64(3):107– 115，2021年。

Zhong, S., Huang, Z., Wen, W., Qin, J. 和 Lin, L. 注意力类结构重参数化：注意力类结构重参数化。*ArXiv*，abs/2304.06345，2023年。

Zhu, M., Chang, B., 和 Fu, C. 卷积神经网络与龙格–库塔方法相结合。*Neural Computing and Applications*，第 1–15 页，2022。

A. 所提出算法的详细信息。

表 3：本文提及的数据集汇总。

基准	Type	维度	数据规模	步长	生成方法	评估时间 T
弹簧-质量系统	训练	*	5k	1e-3	*	20
弹簧-质量系统	验证集	*	0.1k	1e-5	RK4	20
弹簧-质量系统	Test	*	5k	1e-5	RK4	20
双连杆摆	训练	4	20k	1e-3	RK4	10
双连杆摆	验证集	4	1k	1e-5	RK4	10
双连杆摆	Test	4	10k	1e-5	RK4	10
弹性摆	训练	4	20k	1e-3	RK4	50
弹性摆	验证集	4	1k	1e-5	RK4	50
弹性摆	Test	4	10k	1e-5	RK4	50

数据集详情。我们在表 3 中汇总了本文所有基准的训练、验证集和测试数据信息。我们每隔一个步长获取离散解，直至模型时间 T；本文所有实验结果的均方误差损失评估时间为 T。训练期间，弹簧-质量系统的生成方法取决于正文表 1 中所用的数值求解器。此外，弹簧-质量系统的维度也取决于正文表 1 中的 d，并且如果 $d = 20$，维度为 $2d = 40$。“四阶龙格-库塔法”表示四阶龙格-库塔法。

表 4：不同基准的初始化方式。“均匀随机”表示变量从给定范围的均匀分布中采样。“常数”表示将变量初始化为常数。弹簧-质量系统中$\mathbf{p}$ 和$\mathbf{q}$ 的维度取决于正文表 1 中的 d，如果 $d = 20$，它们的维度为 $d = 20$。

Task	**变量**	**维度**	**Type**	**初始化范围**	**模型输入?**
弹簧-质量系统	$\mathbf{p}$	*	均匀随机	$[-2.5, 2.5]^{20}$	✓
弹簧-质量系统	$\mathbf{q}$	*	均匀随机	$[-2.5, 2.5]^{20}$	✓
弹性摆	θ	1	均匀随机	$[0, \pi/8]$	✓
弹性摆	r	1	常数	10	✓
弹性摆	$\dot{\theta}$	1	常数	0	✓
弹性摆	$\dot{r}$	1	常数	0	✓
弹性摆	l_0	1	常数	10	
弹性摆	g	1	常数	9.8	
弹性摆	k	1	常数	40	
弹性摆	m	1	常数	1	
双连杆摆	θ	2	均匀随机	$[0, \pi/8]^2$	✓
双连杆摆	$\dot{\theta}$	2	常数	0	✓
双连杆摆	m	1	常数	1	
双连杆摆	g	1	常数	9.8	

数值求解器。在本文中，我们考虑四种前向数值求解器，以验证所提出的 AttNS 的有效性，包括欧拉方法、改进欧拉法、三阶和四阶龙格-库塔法。本节将介绍这些求解器。正如正文中的公式（2）所述，这些求解器具有不同的积分项 $S(f、\ u_n、\ \Delta t)$。

对于三阶龙格-库塔法，系数为 $\lambda_1 = \lambda_3 = \frac{1}{6}$和 $\lambda_2 = \frac{2}{3}$。此外，$K_1 = \mathbf{f}(\mathbf{u}_n)$ $K_2 = \mathbf{f}(\mathbf{u}_n + \frac{\Delta t}{2} K_1)$ 和 $K_3 = \mathbf{f}(\mathbf{u}_n - \Delta t K_1 + 2\Delta t K_2)$。对于四阶龙格-库塔法，$\beta_1 = \beta_4 = \frac{1}{6}$和 $\beta_2 = \beta_3 = \frac{1}{3}$。此外，$J_1 = \mathbf{f}(\mathbf{u}_n)$ $J_2 = \mathbf{f}(\mathbf{u}_n + \frac{\Delta t}{2} J_1)$ $J_3 = \mathbf{f}(\mathbf{u}_n + \frac{\Delta t}{2} J_2)$ 和 $J_4 = \mathbf{f}(\mathbf{u}_n + \Delta t J_3)$。一般而言，四阶龙格-库塔法的全局截断误差最小 $\mathcal{O}(\Delta t^4)$，也就是说，四阶龙格-库塔法的精度最高。

表5：步长为 Δt时不同类型数值求解器的积分项。

数值求解器	积分项 $S(f, u_n, \Delta t)$	全局截断误差
欧拉法	$\Delta t\mathbf{f}\ (\mathbf{u}_n)$	$\mathcal{O}(\Delta t)$
改进欧拉法	$\frac{\Delta t}{2}[\mathbf{f}(\mathbf{u}_n) + \mathbf{f}(\mathbf{u}_n + \Delta t\mathbf{f}(\mathbf{u}_n))]$	$\mathcal{O}(\Delta t^2)$
三阶龙格-库塔法	$\Delta t(\lambda_1 K_1 + \lambda_2 K_2 + \lambda_3 K_3)$	$\mathcal{O}(\Delta t^3)$
四阶龙格-库塔法	$\Delta t(\beta_1 J_1 + \beta_2 J_2 + \beta_3 J_3 + \beta_4 J_4)$	$\mathcal{O}(\Delta t^4)$

然而，由于其积分项最为复杂 $\Delta t(\beta_1 J_1 + \beta_2 J_2 + \beta_3 J_3 + \beta_4 J_4)$，其模拟速度在表5列出的四种数值求解器中最慢。

B. 我们定理证明背后的动机与挑战

B.1. 预备知识

本文聚焦于微分方程的数值求解器。稳定性（或鲁棒性）和收敛性是衡量数值求解器的两个重要指标。具体而言，我们对常微分方程求解器的稳定性和收敛性定义如下（Ferziger 等人，2002；Dahlquist，1956）：

稳定性指求解器在计算过程中对初始条件、参数和舍入误差的敏感程度。即使输入条件出现轻微变化并带有“噪声”，稳定的求解器也能给出可靠的结果。如果数值方法不稳定，则可能产生不合理的结果，甚至导致计算失败。

收敛性指随着步长（Δt）减小，数值计算趋近于真实解的能力。收敛的求解器会给出越来越精确的结果，并且当步长趋近于零时，误差也会趋近于零。如果求解器不收敛，即使步长很小，也无法给出准确的结果。

上述关于稳定性和收敛性的预备知识表明，对于微分方程数值求解器而言，这两个指标彼此不同。因此，我们分别通过定理 2.1 和 Theorem5.1 讨论稳定性和收敛性。具体而言，

对于稳定性，定理 2.1 考察注意力机制调节噪声、增强模型鲁棒性的能力，这启发我们在数值求解器中引入注意力机制。

对于收敛性，定理 5.1 通过分析近似误差考察所提出求解器的收敛性，并给出了在神经网络训练良好时求解器的收敛速率。

兼具良好稳定性和收敛性的数值求解器能够在不同的微分方程场景中获得良好的解。此外，由于人工智能增强型数值求解器本质上仍是数值求解器，只是通过神经网络加以增强，当训练数据量较小时，此类求解器的稳定性和收敛性可能无法得到充分保证。因此，我们必须提出一种新型人工智能增强型数值求解器，使其在稳定性和收敛性方面具有更好的理论保证，从而在训练数据量较小时，比其他现有求解器更有效地抵消对稳定性和收敛性的不利影响。定理 2.1 和定理 5.1 分别从这两个角度揭示了所提出的 AttNS 有望成为足够优良的人工智能增强型数值求解器；我们还通过实验证明，当训练数据量较小时，AttNS 的表现远优于其他求解器。未来的工作可以进一步开展理论分析，以改进现有的 AttNS。

B.2. 证明定理时的技术创新与难点

我们将证明定理 5.1 的难点总结如下：

1. 定理 5.1 用于分析集成注意力机制的数值求解器的收敛性。据我们所知，我们是首个将注意力机制引入数值求解器的，因此现有文献中可用于分析集成注意力模块的数值求解器收敛性的相关证明技巧并不多。

2. 在证明定理 5. 1，我们的目标是使证明方法具有通用性和可推广性，即我们希望设计 a

一种统一的方法，可用于证明许多满足特定条件的注意力模块 Q 中的定理 5.1。这一要求自然增加了构建证明的难度，但也将为我们今后改进注意力模块 Q 提供更普遍的理论依据。

3. 为弥合理论与实践之间的差距，如何设定合理且宽松的条件，使定理 5.1 适用于分析具有特定经验设计的求解器（例如我们的 AttNS）的收敛性，是一项挑战。

4. 由于我们考虑的求解器是 AI 增强型数值求解器，要证明其收敛性，就必须以非平凡的方式将数值求解器与神经网络这两个概念联系起来。然而，在各自的理论分析中，数值求解器和神经网络这两个概念分别涉及许多不同的因素和视角；将二者结合起来，自然会引入更多因素，极大地干扰我们证明定理 5.1 的目标。因此，我们需要设计一种证明方法，剥离并消除与证明目标无关的干扰因素，从而给出简洁明了的证明，直接论证定理 5.1 的证明目标。

我们还将定理 5.1 证明中的创新点总结如下：

1. **解耦**。 正如“难点”部分所述，为了消除数值求解器和神经网络理论分析中的复杂干扰因素，我们巧妙地利用证明中的三角不等式，将 AttNS $|\hat{\mathbf{u}}_{Nk}-\mathbf{u}(T)|$ 的收敛性解耦为原始数值方法 $|\mathbf{u}_{Nk}-\mathbf{u}(T)|$ 的收敛性和注意力模块 $|\hat{\mathbf{u}}_N-\mathbf{u}_{Nk}|$ 的性能，即

$$|\hat{\mathbf{u}}_{Nk}-\mathbf{u}(T)|\leq \underbrace{|\mathbf{u}_{Nk}-\mathbf{u}(T)|}_{\leq \Delta t\cdot M\cdot \exp(2TL)/2L}+\underbrace{|\hat{\mathbf{u}}_N-\mathbf{u}_{Nk}|}_{\leq \beta\sqrt{\delta}\to 0}, \tag{17}$$

由于 $|\mathbf{u}_{Nk}-\mathbf{u}(T)|$ 与模块训练彼此独立，因此可以使用传统数学方法分析 $|\mathbf{u}_{Nk}-\mathbf{u}(T)|$，而 $|\hat{\mathbf{u}}_N-\mathbf{u}_{Nk}|$ 则可以使用神经网络收敛性的相关方法进行证明。这种解耦成功连接了 AttNS 的收敛性与神经网络的训练性能，也可能为未来将神经网络引入微分方程求解器的理论证明提供启发。

2. **通用性**。正如“难点”部分所述，我们的证明应适用于许多不同的注意力模块。具体而言，我们通过考虑注意力模块的利普希茨连续性，使证明具有通用性。这是一种非常普遍的性质，可用于正式评估未来不同的、新型的注意力模块。我们还在引理 D.3 中证明，式（9）中的 AttNS 类 MLP 注意力模块及其有理激活函数满足利普希茨连续性条件。具有 Lipschitz 条件的注意力模块的收敛性可以利用定理 5.1 进行统一估计。

3. **启发性**。 Lipschitz 条件可以指导未来数值求解器中新注意力模块的设计，并且可能是求解器有效性的必要条件。具体而言，满足 Lipschitz 条件的注意力模块（例如公式 (9) 所示、带有理激活函数的“简单”类 MLP 结构）能够在所提出的 AttNS 框架下有效求解微分方程。然而，一些“复杂”的注意力模块结构（如 Transformer 和 LSTM）可能预测效果较差，这是因为它们通常缺乏 Lipschitz 连续性条件，这与正文中的实验观察结果一致。

C. 定理 5.1 的证明

定理 5.1。我们考虑常微分方程 $\mathrm{d}\mathbf{u}/\mathrm{d}t = \mathbf{f}(\mathbf{u}), \mathbf{u}(0) = \mathbf{c}_0$ 以及欧拉方法 $\mathbf{u}_{n+1} = \mathbf{u}_n + \Delta t\mathbf{f}(\mathbf{u}_n)$。我们假设：(1) $\mathbf{f}$ 满足 Lipschitz 连续性，Lipschitz 常数为 L；(2) 真实解 $\mathbf{u}$ 的二阶导数一致有界于 $M > 0$，即在 $[0,T]$ 上有 $\|\mathbf{u}''\|_\infty \le M$。此外，我们假设 AttNS 中的注意力模块 $Q[\cdot]$ 满足 Lipschitz 连续性，Lipschitz 常数为 $k\Delta t L_{\text{att}}$。对于步长为 $k\Delta t$ 的 AttNS 解 $\hat{}\mathbf{u}$，我们有

$$|\hat{\mathbf{u}}_N - \mathbf{u}(T)| \le \alpha\Delta t + \beta\sqrt{\delta}, \tag{18}$$

其中 $\alpha = \frac{1}{2L}M\exp(2TL)$，$\beta = \frac{\sqrt{T}\exp(TL(1+L_{\text{att}}))}{\sqrt{L(1+L_{\text{att}})}}$ δ是关于 AttSlover 训练的误差项。如果 AttSlover 能很好地拟合训练数据，则误差 $\delta \to 0$。

引理 C.1. 若定理 *5.1* 中的假设 *(1)* 和 *(2)* 成立，对于欧拉方法 $\mathbf{u}_{n+1} = \mathbf{u}_n + \Delta t\mathbf{f}(\mathbf{u}_n)$，有

$$|\mathbf{u}_{Nk} - \mathbf{u}(T)| \le \frac{M\exp(2TL)}{2L}\Delta t, \tag{19}$$

其中 $0 = t_0 < t_1 < \cdots < t_{Nk} = T$是 $[0,T]$和 $\Delta t = \frac{T}{Nk}$上的均匀点。 —

引理 C.2. 对于任意 $n \in \mathbb{N}_+$ 和 $x > 0$，有

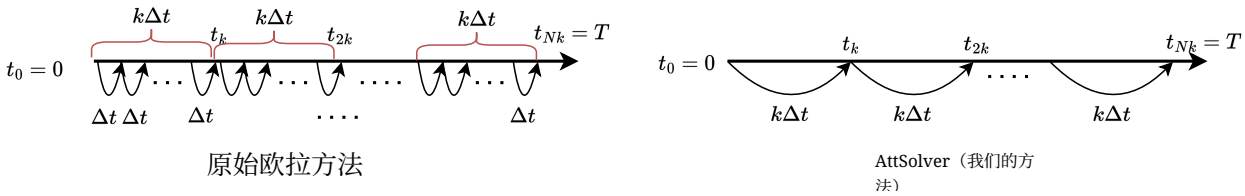
$$(1+x)^n \le \exp(nx), \tag{20}$$

证明。 (引理 C.2)。根据泰勒展开，有

$$\exp(nx) = \sum_{i=0}^{\infty}\frac{(nx)^i}{i!} \ge \sum_{i=0}^{\infty} n^i\frac{x^i}{i!} \ge \sum_{i=0}^{\infty}\prod_{j=0}^{i-1}(n-j)\frac{x^i}{i!} = \sum_{i=0}^{\infty} C_n^i x^i = (1+x)^n \tag{21}$$

□

证明。 (定理 5.1 的证明。)考虑时间区间 $[0,T]$ 的离散化，如图6所示，以及 E_n ：$= \hat{}\mathbf{u}_n - \mathbf{u}_{nk}$



原始欧拉方法　　AttSolver（我们的方法）

图 6：原始欧拉方法（左）和我们的 AttNS（右）的离散化。

$$\begin{aligned} |\hat{\mathbf{u}}_{Nk} - \mathbf{u}(T)| &\le |\mathbf{u}_{Nk} - \mathbf{u}(T)| + |\hat{\mathbf{u}}_N - \mathbf{u}_{Nk}| && \text{Since } |a-b| \le |a| + |b| \\ &= \frac{M\exp(2TL)}{2L}\Delta t + E_N && \text{Since Lemma D.1} \end{aligned}$$

接下来，我们估计 E_N的上界。对于任意 $n \ge 0$，有

$$\begin{aligned} \hat{\mathbf{u}}_{n+1} - \mathbf{u}_{k(n+1)} &= \hat{\mathbf{u}}_n + \mathbf{f}(\hat{\mathbf{u}}_n)(k\Delta t) + Q(\mathbf{f}(\hat{\mathbf{u}}_n)|\phi) - \mathbf{u}_{k(n+1)} \\ &= \hat{\mathbf{u}}_n - \mathbf{u}_{kn} + \big(\mathbf{f}(\hat{\mathbf{u}}_n) - \mathbf{f}(\mathbf{u}_{kn})\big)(k\Delta t) + Q(\mathbf{f}(\hat{\mathbf{u}}_n)|\phi) - Q(\mathbf{f}(\mathbf{u}_{kn})|\phi) - (k\Delta t)V_n. \end{aligned} \tag{22}$$

其中 V_n 是关于训练的误差项，即：

$$V_n = \frac{1}{k\Delta t}(\mathbf{u}_{k(n+1)} - \mathbf{u}_{kn} - \mathbf{f}(\mathbf{u}_{kn})k\Delta t - Q(\mathbf{f}(\mathbf{u}_{nk})|\phi)). \tag{23}$$

接下来，根据 Lipschitz条件和三角不等式，我们有

$$\begin{aligned}|\hat{\mathbf{u}}_{n+1}-\mathbf{u}_{k(n+1)}| &\le |\hat{\mathbf{u}}_n-\mathbf{u}_{kn}|+L|\hat{\mathbf{u}}_n-\mathbf{u}_{kn}|(k\Delta t)+k\Delta tL_{\text{att}}|\mathbf{f}(\hat{\mathbf{u}}_n)-\mathbf{f}(\mathbf{u}_{kn})|+(k\Delta t)|V_n| \\ &\le |\hat{\mathbf{u}}_n-\mathbf{u}_{kn}|+L|\hat{\mathbf{u}}_n-\mathbf{u}_{kn}|(k\Delta t)+k\Delta tL_{\text{att}}\cdot L|\hat{\mathbf{u}}_n-\mathbf{u}_{kn}|+(k\Delta t)|V_n| \\ &= (1+k\Delta tL+k\Delta tL_{\text{att}}L)|\hat{\mathbf{u}}_n-\mathbf{u}_{kn}|+(k\Delta t)|V_n|.\end{aligned} \tag{24}$$

令 $w=(1+k\Delta tL+k\Delta tL_{NV}L)$。式(24)可改写为

$$\begin{aligned}|E_{n+1}| &\le w|E_n|+(k\Delta t)|V_n| \\ &\le w(w|E_{n-1}|+(k\Delta t)|V_{n-1}|)+(k\Delta t)|V_n| \\ &= w^2|E_{n-1}|+w(k\Delta t)|V_{n-1}|+(k\Delta t)|V_n| \\ &\le w^{n+1}|E_0|+(k\Delta t)\sum_{i=0}^{n}w^i|V_{n-i}| = (k\Delta t)\sum_{i=0}^{n}w^i|V_{n-i}|,\end{aligned} \tag{25}$$

其中 $E_0=0$ 当 $E_0=\hat{\mathbf{u}}_0-\mathbf{u}_0=\mathbf{c}_0-\mathbf{c}_0=0$。令

$$\delta=\frac{1}{N}(\|V_0\|_2^2+\|V_1\|_2^2+\ldots+\|V_{N-1}\|_2^2) \tag{26}$$

根据柯西不等式和式(25),

$$\begin{aligned}|E_N| &= (k\Delta t)\sum_{i=0}^{N-1}w^i|V_{N-1-i}| \\ &\le (k\Delta t)(\sum_{i=0}^{N-1}w^{2i})^{\frac{1}{2}}(\sum_{i=0}^{N-1}|V_{N-1-i}|^2)^{\frac{1}{2}} && \text{Since Cauchy inequality} \\ &= (k\Delta t)\sqrt{(w^{2N}-1)/(w^2-1)}\sqrt{N\delta}. && \text{Since Eq.(26)}\end{aligned}$$

接下来，我们简化项 $\sqrt{(w^{2N}-1)/(w^2-1)}$。注意到 $w=(1+k\Delta tL+k\Delta tL_{\text{att}}L)\ge 1$，因此

$$w^2-1\ge w-1. \tag{27}$$

因此,

$$\begin{aligned}\sqrt{(w^{2N}-1)/(w^2-1)} &\le \sqrt{[(1+k\Delta tL+k\Delta tL_{\text{att}}L)^{2N}]/[k\Delta tL+k\Delta tL_{\text{att}}L]} && \text{Since Eq.(27)} \\ &\le \sqrt{\frac{\exp(2Nk\Delta tL(1+L_{\text{att}}))}{k\Delta tL(1+L_{\text{att}})}} && \text{Since Lemma C.2}\end{aligned}$$

因此,

$$|E_N|\le (k\Delta t)\frac{\exp(TL(1+L_{NV}))}{\sqrt{k\Delta tL(1+L_{\text{att}})}}\sqrt{N\delta}=\frac{\sqrt{T}\exp(TL(1+L_{\text{att}}))}{\sqrt{L(1+L_{\text{att}})}}\sqrt{\delta}.$$

因此，我们有

$$|\hat{\mathbf{u}}_N-\mathbf{u}(T)|\le\alpha\Delta t+\beta\sqrt{\delta}, \tag{28}$$

其中 $\alpha=\frac{1}{2L}M\exp(2TL)$，$\beta=\frac{\sqrt{T}\exp(TL(1+L_{\text{att}}))}{\sqrt{L(1+L_{\text{att}})}}$。由式(23)，如果 AttSlover 能很好地拟合训练数据，$\|V_n\|_2\to 0$，并且有 $\delta\to 0$。 □

D. 我们的 $Q[\cdot]$是 Lipschitz 连续的

引理 D.1. 对于 $x \in [a,b]$，多项式函数 $P(x) = \sum_{k=0}^{n} c_k x^k$ 是 *Lipschitz* 连续的。

证明。 对于所有 $x,\ y \in [a,b]$，我们有 $|x^k - y^k| = (x-y)\sum_{i=0}^{k-1} x^i y^{k-1-i} \le |x-y| \cdot k \cdot$最大值$(a,\ b)^{k-1}$. 因此，我们有

$$|P(x) - P(y)| \le \sum_{k=0}^{n} |c_k||x^k - y^k| \le \underbrace{\{\sum_{k=0}^{n} k \cdot |c_k| \cdot \max(a,b)^{k-1}\}}_{\text{Lipschitz constant}} \cdot |x-y|. \tag{29}$$

□

引理 D.2。 对于 $x \in$ 定义域 A，有理激活函数 $\mathbf{a}(x) = \frac{\sum_{i=1}^{3} a_i x^i}{\sum_{i=1}^{2} b_i x^i}$ 是 *Lipschitz* 连续的。

证明。 对于所有 $x_1,\ x_2 \in A$，令 $p(x) = \sum_{i=1}^{3} a_i x^i$ 和 $q(x) = \sum_{i=1}^{2} b_i x^i$。对于所有 $x_1,\ x_2 \in A$，我们有

$$|\mathbf{a}(x_1) - \mathbf{a}(x_2)| = |\ \frac{p(x_1)}{q(x_1)} - \frac{p(x_2)}{q(x_2)}| \tag{30}$$

$$= |\frac{p(x_1)q(x_2) - q(x_1)p(x_2)}{q(x_1)q(x_2)}| \tag{31}$$

$$= |\frac{p(x_1)q(x_2) - p(x_1)q(x_1) + p(x_1)q(x_1) - q(x_1)p(x_2)}{q(x_1)q(x_2)}| \tag{32}$$

$$\le \frac{|p(x_1)|}{|q(x_1)q(x_2)|}|q(x_2) - q(x_1)| + \frac{|q(x_1)|}{|q(x_1)q(x_2)|}|p(x_2) - p(x_1)|. \tag{33}$$

注意，由于多项式函数 $p(x)$和 $q(x)$在 A 上连续，因此它们在 A 上有界。令最大值$(|p(x)|, |q(x)|) \le M$，最小值$(|p(x)|, |q(x)|) \ge N$。此外，根据引理 D.1，对于所有 x_1、$x_2 \in A$，我们假设 $|q(x_2) - q(x_1)| \le L_q|x_2 - x_1|$ 且 $|p(x_2) - p(x_1)| \le L_p|x_2 - x_1|$。

因此，我们有

$$|\mathbf{a}(x_1) - \mathbf{a}(x_2)| \le \frac{|p(x_1)|}{|q(x_1)q(x_2)|}|q(x_2) - q(x_1)| + \frac{|q(x_1)|}{|q(x_1)q(x_2)|}|p(x_2) - p(x_1)| \tag{34}$$

$$\le \frac{M}{N^2}|q(x_2) - q(x_1)| + \frac{M}{N^2}|p(x_2) - p(x_1)| \tag{35}$$

$$\le \underbrace{\frac{M}{N^2} \cdot (L_p + L_q)}_{\text{Lipschitz 常数}} \cdot |x_1 - x_2|. \tag{36}$$

□

引理 D.3. 对于 $x \in$ 定义域 A，正文中式 *(9)* 所示的注意力模块，即 $Q(x) = \mathbf{W_2} \circ \mathbf{a} \circ \mathbf{W_1} x$，是 *Lipschitz* 连续的。$\mathbf{a}$ 是有理激活函数，即 $\mathbf{a}(x) = \frac{\sum_{i=1}^{3} a_i x^i}{\sum_{i=1}^{2} b_i x^i}$，其中 $a_i, b_i \in \mathbb{R}$。 ————

证明。 对于所有 $x, y \in A$，根据引理 D.2，存在一个常数 L_1 使得 $||\mathbf{a}(x) - \mathbf{a}(y)|| \le L_1||x-y||$. 因此，

$$||\mathbf{a} \circ \mathbf{W_1}(x) - \mathbf{a} \circ \mathbf{W_1}(y)|| \le L_1||\mathbf{W_1}(x) - \mathbf{W_1}(y)|| \tag{37}$$

$$\le L_1||\mathbf{W_1}|| \cdot ||x-y||. \tag{38}$$

此外，我们考虑注意力模块 $Q[\cdot]$，并且有

$$||Q(x) - Q(y)|| = ||\mathbf{W_2} \circ \mathbf{a} \circ \mathbf{W_1}(x) - \mathbf{W_2} \circ \mathbf{a} \circ \mathbf{W_1}(y)|| \tag{39}$$

$$\le ||\mathbf{W_1}|| \cdot ||\mathbf{a} \circ \mathbf{W_1}(x) - \mathbf{a} \circ \mathbf{W_1}(y)|| \tag{40}$$

$$\le \underbrace{L_1 \cdot ||\mathbf{W_2}|| \cdot ||\mathbf{W_1}||}_{\text{Lipschitz constant}} \cdot ||x-y||. \tag{41}$$

因此，我们在正文公式 (9) 中考虑的注意力模块是 Lipschitz 连续的 .

□

D.1. 关于条件 $\|\mathbf{u}''\|_\infty \le M$ on $[0,T]$。

定理 5.1 中“真实解的二阶导数一致有界”这一条件通常很容易满足，并且我们的算法可以与之协同。具体而言，

(1)“二阶导数” $||\mathbf{u}''||_\infty$ 不会直接影响 AttNS 的训练目标，也就是说，AttNS 可以采用较大的步长 $k\Delta t$ （估计速度快）来求解微分方程，从而达到与采用较小步长 Δt （估计速度慢）的原始数值方法相近的精度。

定理 5.1 的作用是告诉我们，AttNS 能够实现这一预期目标。具体而言，给定评估时间 T和 Lipschitz 常数 L，若神经网络经过良好训练，即 $\beta\sqrt{\delta} \to 0$，那么 AttNS 的精度 $|\hat{\mathbf{u}}_N - \mathbf{u}(T)|$与采用较小时间步长的原始数值算法的精度 Δt , $|\mathbf{u}_{Nk} - \mathbf{u}(T)|$ 相当。这是因为

$$|\hat{\mathbf{u}}_N - \mathbf{u}(T)| \le \underbrace{|\mathbf{u}_{Nk} - \mathbf{u}(T)|}_{\le \Delta t \cdot M \cdot \exp(2TL)/2L} + \underbrace{|\hat{\mathbf{u}}_N - \mathbf{u}_{Nk}|}_{\le \beta\sqrt{\delta} \to 0}. \tag{42}$$

另一方面，上述不等式表明， M $|\mathbf{u}''|_\infty$ 的上界主要影响 AttNS 和原数值方法的精度。当 M 较大时，它们精度的上界可能增大，但这并不影响这样一个目标：使用较大步长 $k\Delta t$ 的 AttNS 可以近似采用较小步长 Δt 的原数值方法。

(2) 大多数“二阶导数” $|\mathbf{u}''|_\infty$ 都相对较小。我们从两个系统（即弹簧链和双连杆摆）的测试集中提取了 1000 条轨迹，并使用二阶差分格式计算了它们的 $|\mathbf{u}''|_\infty$ 。这些轨迹使用步长为 1e-4 的四阶 Runge–Kutta 方法生成，可视为真实解的近似。结果见下表。

表 6： $|\mathbf{u}''|_\infty$的分布。

$\|\mathbf{u}''\|_\infty$ （弹簧链） ¡ $\|\mathbf{u}''\|_\infty$的百分比	30 55.35%	60 80.51%	90 91.84%	120 96.68%	150 98.57%	180 99.41%	210 99.76%	240 99.91%	270 99.97%
$\|\mathbf{u}''\|_\infty$ (双连杆摆) ¡ $\|\mathbf{u}''\|_\infty$的百分比	5 41.10%	10 65.93%	15 80.55%	20 89.48%	25 94.66%	30 97.61%	35 99.03%	40 99.71%	45 99.95%

从这些表格中，我们可以观察到 $|\mathbf{u}''|_\infty$ 呈长尾分布，其中大多数 $|\mathbf{u}''|_\infty$都相对较小。请注意，由于维数和坐标不同， $|\mathbf{u}''|_\infty$ 的大小在不同系统中有所差异。并且只有少数 $|\mathbf{u}''|_\infty$ 较大。这一观察结果与（Liang 等人，2022）关于其他混沌系统（即三体系统和台球系统）的研究发现一致。

较大的 $|\mathbf{u}''|_\infty$ 对应于动力系统中的刚性步，此时解会突然变化，并可能对模型训练产生一定影响。接下来，我们将研究所提出的 Attsovler 能否与这些较大的$|\mathbf{u}''|_\infty$ 协同工作。

（3）我们提出的 AttNS 能够与较大的“二阶导数”协同工作。以弹性摆系统和双连杆摆为例，我们选取测试集中“二阶导数”最大的前 10% 轨迹，并将其重新组合成一个新的测试集。以下给出了在评估时间 T=10 下，使用 RK4（步长 = 1e-1）得到的实验结果。

方法	均方误差损失（弹性）	均方误差损失（双连杆）
RK4	4.33e-0	3.56e-2
NeurVec	6.32e-5	3.54e-6
AttNS（我们的方法）	**1.34e-6**	**8.54e-8**

我们可以发现，尽管性能有所下降，但在这些二阶导数相对较大的轨迹上，我们的方法仍显著优于最先进水平和传统数值方法。

E. 定理 5.3 的证明。

定理 5.3。设 Net($\hat{}S|\phi,\mathbb{D}_f$) 和 Net($\mathbf{u}_n|\phi,\mathbb{D}_f$)，其中 $\hat{}S = S(\mathbf{f},\mathbf{u}_n,\Delta t_c)$ 分别为一般 AHS 和 AttNS 的修正项。对于 $\epsilon > 0$，当数据规模大于 N' 时，两种方法的经验误差满足$R_e(\phi|\text{AHS}) \le \epsilon$且 $R_e(\phi|\text{AttNS}) \le \epsilon$。对于足够小的 $\epsilon_0 \ll \epsilon$ 和欧拉方法，我们有

$$N(\text{AttNS}) \lesssim N(\text{AHS}), \tag{43}$$

其中，$N(*)$ 是方法 $*$ 的泛化误差达到 $\epsilon(1-\epsilon_0)^{-1}$ 时所需数据规模的下界。

引理 E.1. 考虑模型集合 $S_k = \{f(\cdot|w)\ w \in \omega_k\}$。如果 $S_1 \subset S_2 \subset \cdots \subset S_k \subset \cdots$，

$$h_1 \le h_2 \le \cdots \le h_k \le \cdots, \tag{44}$$

其中，h_i 是集合 S_i、$i = 1$、2、$\cdots$ 中模型的 *Vapnik-Chervonenkis* 维数。

引理 E.2. 令 $f(x) = (\ln x + \alpha)/x$，$x \in (0,\ +\infty)$ 且 $\alpha \in \mathbb{R}$。$f(x)$ 在 $x = e^{1+\alpha}$ 处取得最大值；当$x \in (0,\ e^{1+\alpha})$时，$f(x)$ 单调递增；当 $x \in (e^{1+\alpha},\ +\infty)$时，$f(x)$ 单调递减。

引理 E.3。令 $f(x) = x\exp(\frac{-a}{x})$，其中 $a > 0$ 且 $x \in \mathbb{R}$。则 $f(x)$ 单调递增。

引理 E.4。令 $W(x)$ 为 *Lambert* W 函数，若 $|x| < 1/e$，则 $W(x)$ 的泰勒展开为

$$W(x) = \sum_{n=1}^{\infty} \frac{(-n)^{n-1}}{n!} x^n = x - x^2 + \frac{3}{2}x^3 - \frac{8}{3}x^4 + \cdots. \tag{45}$$

假设 E.5。对于 $x \in \mathbb{R}$和一般实函数 $f(x)$，我们考虑映射 f_1：$x \to \nabla_x f(x)^T f(x)$ 以及f_2：$f(x) \to \nabla_x f(x)^T f(x)$。对于足够小的 $\epsilon > 0$，若 S_1和 S_2是 $\forall g_1(\phi) \in S_1$和 $g_2(\phi) \in S_2$所对应的集合，

$$\|g_1(\phi) - f_1\| \le \epsilon, \quad \|g_2(\phi) - f_2\| \le \epsilon, \tag{46}$$

我们假设 $S_1 \subset S_2$。该假设意味着，用于拟合 f_1 的模型复杂度高于用于拟合 f_2的模型 .

证明。（定理 6.1）。

根据用于回归的 Vapnik-Chervonenkis 理论（Vapnik，1999），对于任意数据分布 $P(x,\ y)$、模型 $\mathscr{A}(\phi)$、泛化误差 $R(\phi)$和经验误差 $R_e(\phi)$，不等式

$$R(\phi) \le R_e(\phi)\left(1 - c\sqrt{\delta(h_{\mathscr{A}(\phi)}, N)}\right)_+^{-1}, \tag{47}$$

以概率 $1-\eta$成立。$h_{\mathscr{A}(\phi)}$ 是模型 $\mathscr{A}(\phi)$ 的 Vapnik-Chervonenkis 维数，N 是数据规模，c 通常设为 1，并且

$$\delta(h, N) = (h\left[\ln\frac{N}{h} + 1\right] - \ln\eta)/N. \tag{48}$$

因此，根据式（47），对于泛化误差 $R(\phi|\text{Neur.})$ 和 $R(\phi|\text{Att.})$，我们有

$$R(\phi|\text{Neur.}) \le R_e(\phi|\text{Neur.})\left(1 - c\sqrt{\delta(h_{\text{Neur.}(\phi)}, N)}\right)_+^{-1}, \tag{49}$$

and

$$R(\phi|\text{Att.}) \le R_e(\phi|\text{Att.})\left(1 - c\sqrt{\delta(h_{\text{Att.}(\phi)}, N)}\right)_+^{-1}, \tag{50}$$

接下来，我们考虑项 $(1-\sqrt{\delta(h_{\mathscr{A}(\phi)}N)})^{-1}$ 需要多大的数据规模才能达到精度 $(1-\epsilon_0)^{-1}$。设 $\delta:(\ln N+\alpha)/N$，其中 $N=N/h_{\mathscr{A}(\phi)}$ 和 $\alpha=1-\ln\eta/h_{\mathscr{A}(\phi)}$。

$$\begin{aligned}(\ln\bar{N}+\alpha)/\bar{N}=\epsilon_0 &\iff \ln\bar{N}+\alpha=\bar{N}\epsilon_0\\ &\iff \bar{N}\exp(-\bar{N}\epsilon_0)=\exp(-\alpha)\\ &\iff -\epsilon_0\bar{N}\exp(-\bar{N}\epsilon_0)=-\epsilon_0\exp(-\alpha).\end{aligned}\tag{51}$$

因此 $-\epsilon N=W(-\epsilon_0\exp(-\alpha))$，其中 W 是 Lambert W 函数（Lehtonen，2016年），并且

$$\bar{N}=-\frac{h_{\mathscr{A}(\phi)}}{\epsilon_0}\cdot W\left(-\epsilon_0\exp(\frac{\ln\eta}{h_{\mathscr{A}(\phi)}}-1)\right),\tag{52}$$

根据引理E.2，我们知道，当数据规模为 $N\geq\bar{N}$，$(1-\sqrt{\delta(h_{\mathscr{A}(\phi)}N)})^{-1}$ 时，可以达到精度 $(1-\epsilon_0)^{-1}$。根据式（49）和式（50），当 $N\geq$最大值 $\left\{N',\ -\frac{h_{\text{Neur.}(\phi)}}{\epsilon_0}\cdot W\left(-\epsilon_0\exp(\ln\frac{\eta}{h_{\text{Neur.}(\phi)}}-1)\right)\right\}$ 以及 $N\geq$最大值 $\left\{N',\ -\frac{h_{\text{Att.}(\phi)}}{\epsilon_0}\cdot W\left(-\epsilon_0\exp(\ln\frac{\eta}{h_{\text{Att.}(\phi)}}-1)\right)\right\}$ 时，我们有

$$R(\phi|\text{Att.})\leq\epsilon(1-\epsilon_0)^{-1},\quad R(\phi|\text{Neur.})\leq\epsilon(1-\epsilon_0)^{-1},\tag{53}$$

以概率成立 $1-\eta$。事实上，由于 $\epsilon_0\ll\epsilon$ 以及引理E.2，在式（52）中，$N\gg N'$因此，我们可以设

$$N(\text{AttNS})=-\frac{h_{\text{Att.}(\phi)}}{\epsilon_0}\cdot W\left(-\epsilon_0\exp(\frac{\ln\eta}{h_{\text{Att.}(\phi)}}-1)\right),\quad N(\text{AHS})=-\frac{h_{\text{Neur.}(\phi)}}{\epsilon_0}\cdot W\left(-\epsilon_0\exp(\frac{\ln\eta}{h_{\text{Neur.}(\phi)}}-1)\right).\tag{54}$$

注意，$\exp(\ln\eta/h_{\mathscr{A}(\phi)}-1)<\exp(-1)$，即：

$$\left|-\epsilon_0\exp(\ln\eta/h_{\mathscr{A}(\phi)}-1)\right|<\exp(-1).\tag{55}$$

因此，由引理E.4，对于式（52），有

$$\begin{aligned}-\frac{h_{\mathscr{A}(\phi)}}{\epsilon_0}\cdot W\left(-\epsilon_0\exp(\frac{\ln\eta}{h_{\mathscr{A}(\phi)}}-1)\right)&\approx-\frac{h_{\mathscr{A}(\phi)}}{\epsilon_0}\cdot\left(-\epsilon_0\exp(\frac{\ln\eta}{h_{\mathscr{A}(\phi)}}-1)\right)\\ &=h_{\mathscr{A}(\phi)}\exp(\frac{\ln\eta}{h_{\mathscr{A}(\phi)}}-1).\end{aligned}\tag{56}$$

注意，对于常微分方程 $\mathrm{d}\mathbf{u}/\mathrm{d}t=\mathbf{f}(\mathbf{u})$，$\mathbf{u}(0)=\mathbf{c}_0$，人工智能增强型数值求解器中神经网络的目标标签实际上是不同数值求解器的误差项，它由 $\mathcal{O}[\nabla\mathbf{f}(\mathbf{u})\mathbf{f}(\mathbf{u})]$所支配。根据假设E.5 和引理E.1，有

$$h_{\text{Att.}(\phi)}\leq h_{\text{Neur.}(\phi)}.\tag{57}$$

因此，我们有

$$\begin{aligned}N(\text{AttNS})&=-\frac{h_{\text{Att.}(\phi)}}{\epsilon_0}\cdot W\left(-\epsilon_0\exp(\frac{\ln\eta}{h_{\text{Att.}(\phi)}}-1)\right) && \text{Since Eq.(54)}\\ &\lesssim-\frac{h_{\text{Neur.}(\phi)}}{\epsilon_0}\cdot W\left(-\epsilon_0\exp(\frac{\ln\eta}{h_{\text{Neur.}(\phi)}}-1)\right) && \text{Since Eq.(57) and Lemma E.3}\\ &=N(\text{AHS})\end{aligned}$$

□