arXiv:2603.03238v1 [cs.LG] 3 Mar 2026

关于具有潜在神经常微分方程动力学的自编码器降阶模型的几何正则化

Mikhail Osipov 独立研究者，
意大利 *

2026年3月3日

摘要

我们研究编码器–解码器降阶模型中学习得到的潜在表示的几何正则化策略。在针对平流–扩散–反应（ADR）方程的固定实验设置中，我们使用神经常微分方程对潜在动力学进行建模，并评估自编码器预训练期间采用的四种正则化方法：(a) 解码器雅可比矩阵的近等距正则化，(b) 基于随机方向增益的随机解码器增益惩罚，(c) 二阶方向曲率惩罚，以及 (d) 对第一解码器层进行斯蒂费尔投影。通过多个随机种子进行实验后，我们发现，(a)–(c) 通常会产生使后续使用冻结的自编码器进行潜在动力学训练变得更加困难的潜在表示，尤其是在长时域滚动预测中；即便这些方法改善了解码器的局部平滑性或相关的敏感性代理指标，情况依然如此。相比之下，(d) 始终能改善学习得到的潜在动力学的条件性相关诊断指标，并且往往带来更好的滚动预测性能。我们探讨了这样一种假设：在这一设置下，潜在几何失配对下游任务的影响超过了解码器平滑性提升所带来的益处。

1 引言

自编码器（AE）架构在机器学习（ML）和科学机器学习（SciML）中被广泛使用。通过编码器–解码器构造进行降维，使得能够在低维潜在空间中对高维动力学进行建模 [1, 2, 3, 4, 5]。

神经常微分方程（NODE） [6, 7] 通常用于学习潜在空间中的连续时间动力学。更广泛地说，算子学习方法也已成功应用于偏微分方程（PDE） [8]。

*osipov.ma@phystech.edu

近期研究 [9, 10]为通过自编码器进行降阶建模（ROM）提供了数学框架，探讨了其理论局限性，并展示了其在代表性基准测试中的计算效率。

基于自编码器的降阶模型（ROM）面临的一个根本问题是，潜在空间$\mathbb{R}^d$的维数通常低于环境空间 $\mathbb{R}^n$。当 $d < n$ 时，编码器 $E : \mathbb{R}^n \to \mathbb{R}^d$ 不可能是全局单射，因此会在数据流形之外造成信息损失。因此，解码器 D:$\mathbb{R}^d \to \mathbb{R}^n$ 可能沿流形的某些方向表现出局部扩张行为。这可能导致解码后潜空间误差被放大，而这一效应在长时域滚动预测中尤为明显。

解码器对潜在误差的放大程度可用其利普希茨常数 [11]来量化。令 $u(t)$ 表示环境空间中的真实（离散化）轨迹，令 $z(t) = E(u(t))$ 表示其潜在表示。令 $\hat{}z(t)$ 表示潜在空间中的数值近似，并定义 $\hat{}u(t) = D(\hat{}z(t))$。那么

$$\|u-\hat{u}\|_2 = \|u-D(\hat{z})\|_2 \le \|u-D(E(u))\|_2+\|D(E(u))-D(\hat{z})\|_2 \le \varepsilon_{\text{recon}}+L_D\|z-\hat{z}\|_2,$$

其中，$\varepsilon_{\text{recon}}$ 表示重构误差，L_D 是解码器在潜在空间相关区域上的局部利普希茨常数。

控制表示学习中敏感性的一种经典方法是引入基于雅可比的正则化 [12, 13, 14, 15, 16]。一些作者也讨论了自编码器的曲率正则化 [17]。另一种常用的正则化方法与 Stiefel 投影有关 [18, 19, 20, 21, 22]。

在这项工作中，我们研究了若干旨在控制解码器局部敏感性和几何性质的启发式方法，以及投影到 Stiefel流形上的结构化投影：

(a) 对解码器雅可比矩阵施加局部等距惩罚，以满足 $\|J_D^\top J_D - \mathbf{I}\|_{\text{Frobenius}} \approx 0$；(b) 基于方向范数 $\|J_D v\|$ 对随机单位向量 v施加随机解码器增益惩罚；(c) 基于雅可比矩阵与向量乘积的变化施加二阶方向曲率惩罚；(d) 对第一解码器层应用 Stiefel（列正交）投影。

我们通过数值实验展示：在将平流–扩散–反应（ADR）系统与潜在神经常微分方程相结合的设定下，使用正则化项 (a)–(c) 对自动编码器进行预训练，往往会产生这样的潜空间几何：学习稳定的潜在动力学变得更加困难，长时域滚动预测的性能也会下降。即使内在几何诊断显示局部光滑性有所提高，或解码器敏感性代理指标有所降低，这种情况仍会发生。相比之下，斯蒂费尔投影 (d) 能够持续改善学习得到的潜在动力学的条件性相关诊断指标，并带来略好的（或至少不差的）滚动预测性能。

2 几何正则项

本节介绍我们在实验中使用的正则化方法，并简要说明每种方法的动机。完整的形式化定义见附录 A。

2.1 (a)近等距雅可比惩罚

一种经典的自编码器正则化方法基于雅可比矩阵控制 [12]。作为我们的正则化选项之一，我们使用解码器雅可比矩阵近等距惩罚，并在自编码器预训练期间将其加入损失函数：

$$\mathcal{R}_a \;=\; \mathbb{E}\Big[\big\|J_D^\top J_D - \mathbf{I}\big\|_F^2\Big]\,.$$

在实践中，该项既可精确评估（通过基向量雅可比－向量积），也可使用 Hutchinson 式随机估计器 [23] 进行近似评估。在我们的实验中，我们并未观察到该惩罚的精确评估与随机评估之间存在显著差异。

2.2 (b)方向增益惩罚

为了在不强制完全近等距的情况下控制解码器敏感度，我们还考虑一种较弱的方向增益惩罚：

$$\mathcal{R}_b \;=\; \mathbb{E}_v\Big[\big(\|J_D v\| - 1\big)^2\Big]\,,$$

其中， v 是潜在空间中的随机单位向量。该惩罚鼓励采样方向上的增益近似为单位增益，但不同于 $\mathcal{R}_a$，它不会直接约束 J_D的完整谱。所需的雅可比-向量积通过有限差分进行数值估计。

2.3 (c) 平坦性惩罚

为了探测解码器诱导几何的局部平坦性，同时不显式强制施加严格的等距约束或增益约束，我们引入一种曲率相关惩罚：

$$\mathcal{R}_c \;\approx\; \mathbb{E}\Big[\big\|\nabla^2 D(z)\big\|^2\Big] \;\approx\; \mathbb{E}\left[\frac{\|J_D(z+\varepsilon v)\,v - J_D(z)\,v\|^2}{\varepsilon^2}\right].$$

该项近似表示解码器的方向二阶变化。为计算该项，我们使用自动微分雅可比－向量积，并通过对偏移基点处的这些雅可比－向量积进行有限差分来估计方向曲率。

2.4 (d)Stiefel 投影

将结果投影到 Stiefel流形上，是矩阵流形优化中的标准方法 [18, 19, 20, 21, 22]。在我们的设置中，每次梯度步之后，都会将选定解码器层的权重矩阵投影回Stiefel流形，从而确保该层的列正交。

与上述惩罚项不同，该方法仅对单个层施加结构约束（在我们的案例中，是解码器的第一个卷积层）。因此，它可以改善该层的条件数，而不会直接对整个解码器雅可比矩阵施加近等距、低曲率或受控增益约束。因此，应将其视为一种局部架构正则化，而非对整个解码器几何结构的显式约束。

3 数值实验

3.1 数据集生成

我们使用 dolfinx 有限元求解器，在单位正方形上为参数化对流–扩散–反应（ADR）问题生成全阶轨迹。空间离散化采用均匀三角形网格上的一阶拉格朗日单元，网格划分为 31×31 个子区间，因此每个快照包含 $32\times32=1024$ 个标量自由度。

控制动力学的参数向量为 $\mu=(\mu_1,\mu_2,\mu_3)\in\mathbb{R}^3$，其中 μ_1 控制扩散，而 (μ_2,μ_3) 确定高斯源项的中心。反应系数固定，对流场则随时间变化。

全阶轨迹在区间 $[0,T]$ 上积分，其中 $T=10\pi$，采用后向欧拉法，时间步数为 1000。由此，每条轨迹生成 1001 个细网格快照。对于降阶模型训练，我们还通过每隔五个全阶时间步保留一个时间步来构建粗时间网格，因此每条轨迹包含 201 个快照。

参数域划分为三个子集：内层参数框中的结构化训练集/验证集、由同一参数框内的中点参数构成的插值测试集，以及在训练框之外但位于更大外盒之内采样得到的外推测试集。

我们还使用时刻 $T_1=4\pi$ 和 $T_2=5\pi$ 来区分每条轨迹中的训练区间、时间插值区间和时间外推区间。

我们的方案总体遵循 [10] 的方法；数据集生成方案的详细说明见附录B。

3.2 训练

我们首先在训练集中的各个快照上预训练一组自编码器，并应用第 2.1–2.4 节所述的正则化方法之一。作为基线，我们还训练了一对未正则化编码器–解码器。

所有自编码器均采用相同的架构，区别仅在于预训练期间所采用的几何正则化策略。为降低下游结果由某个自编码器特别有利的实现所导致的可能性，我们针对每种方法进行多次独立的自编码器运行（在主要设置中，每种方法采用两个自编码器随机种子）。每次预训练运行的计算预算相同，均为 200 个训练轮次。

在检查点选择方面，我们的主要协议使用一个所有方法都能达到的共享验证集均方误差目标；对于每个自编码器，我们选择首次达到该目标的训练轮次。在另一组实验中，我们改用特定于方法的检查点选择，即为每种方法选择验证损失最低的训练轮次。

完成自编码器预训练后，我们冻结编码器和解码器，并在潜在空间中训练神经常微分方程。对于每个选定的自编码器，我们训练多个彼此独立的神经常微分方程运行实例（在我们的主要设置中，每种方法使用 16 个 NODE 随机种子）。关键在于，对于给定的 NODE 随机种子，所有方法都从相同的潜在动力学初始化开始，而每种方法使用各自冻结的自编码器。这样，神经常微分方程的比较便在随机种子层面实现配对，并使我们能够更直接地将差异归因于相应自编码器所诱导的潜在几何，而不是潜在动力学模型的有利初始化。

这一两阶段方法隔离了自编码器所诱导的潜在几何对潜在动力学可学习性和长时域滚动预测质量的影响。

3.3 评估

在评估中，我们从所选数据集划分中采样一组固定的滚动预测窗口，并对所有方法使用完全相同的窗口，从而使比较在窗口层面成对进行。在我们的主要设置中，评估在精细时间网格上针对若干滚动预测时域进行。

对于每个采样窗口，我们对初始状态进行编码，在相应的绝对时间网格上积分潜在神经常微分方程，解码预测的潜在轨迹，并将其与全阶轨迹进行比较。我们报告随时间聚合的绝对与相对滚动误差，包括平均误差和最大误差。

此外，在一部分评估窗口上，我们计算学习得到的潜在动力学和解码器的内在诊断，包括潜在动力学雅可比矩阵的奇异值和条件数、解码器增益代理指标以及潜在跟踪误差。

由于实验中的主要变异来自神经常微分方程训练，而非评估窗口的选择，因此我们的主要比较是在匹配的 NODE 随机种子层面进行的。也就是说，对于固定的 NODE 随机种子，所有方法都使用相同的滚动预测窗口和相同的潜在动力学初始化进行比较，因此所得差异可解释为方法之间按随机种子配对的差值。因此，在每次运行中，我们通过窗口级配对来降低评估噪声，而主要

统计比较则基于汇总并比较匹配的 NODE 随机种子所对应的结果。

所有汇总统计量均按方法报告，并使用这些匹配的逐种子评估结果，与未正则化基线进行配对比较。

详细的正式评估协议见附录 C。

4 结果

4.1 训练动态

正如预期，在自编码器预训练期间加入基于几何的目标函数会影响优化和收敛。为确保正则化方法与未正则化基线之间的比较公平，我们在自编码器预训练期间采用统一的检查点选择标准：对于每个自编码器随机种子（版本 $= 1, 2$），我们定义一个所有方法都能达到的共同验证重建目标（验证 MSE），并为每种方法选择首次达到该目标时的检查点。因此，除细微差异外，所有方法进入 NODE 阶段时的自编码器重建质量均相当。

表 1：不同正则化方法的潜在 NODE 训练元数据

方法	$\mathcal{L}_{\mathrm{val}}$	$\mathrm{CI}_{95}(\mathcal{L}_{\mathrm{val}})$	ep_{target}	$\mathrm{CI}_{95}(ep_{\mathrm{target}})$
Stiefel 投影	$8.6 \cdot 10^{-4}$	$1.0 \cdot 10^{-4}$	13.5	1.6
原始基线	$8.8 \cdot 10^{-4}$	$1.2 \cdot 10^{-4}$	14.1	1.4
曲率惩罚	$1.1 \cdot 10^{-3}$	$1.4 \cdot 10^{-4}$	14.5	1.2
增益惩罚	$1.4 \cdot 10^{-3}$	$1.8 \cdot 10^{-4}$	18.1	2.5
等距惩罚	$1.4 \cdot 10^{-3}$	$1.8 \cdot 10^{-4}$	19.3	3.4

对于每种方法，表中报告了最佳验证损失的均值 $\mathcal{L}_{val}$、相应的 *95%* 置信区间半宽、首次达到共享验证目标时的平均训练轮次 ep_{target}，以及相应的 *95%* 置信区间半宽。数值为匹配的 *AE/NODE* 实例的平均值。

尽管自编码器层面的表现相匹配，但后续神经常微分方程的训练动态在不同方法之间存在显著差异。对于潜在神经常微分方程，我们监测：(i) 最佳验证损失（验证集划分上的轨迹滚动均方误差，粗网格上固定预测时域为 $H = 40$），(ii) 达到该最佳验证损失时的训练轮次，以及 (iii) 在事后比较中首次达到共享验证损失目标时的训练轮次（见表 1）。

在固定训练预算为 50 个训练轮次且自编码器被冻结的情况下，基于 Stiefel 正则化自编码器构建的神经常微分方程，其验证损失略优于原始基线。曲率正则化 AE 的验证损失略差，而等距正则化和增益正则化

自编码器的验证损失则显著更差。在共享目标分析中也能看到类似的模式：斯蒂费尔正则化、基线和曲率正则化系统在相近的训练轮次达到共享验证目标（定性排序相同），而等距与增益变体则晚得多才达到相同目标。

我们将这些差异解读为一种早期迹象，表明等距和增益惩罚项所产生的潜在表示不太有利于学习稳定潜在动力学，即使相应的自编码器在重构误差上与基线相匹配。

4.2 滚动预测动力学

我们在精细时间网格和外推划分上，针对若干滚动时域（$H = 80$,160, 240, 320）评估滚动预测质量。在主要实验方案中，自编码器检查点依据上文所述的共享重构目标进行选择，而 NODE 检查点则选取验证损失最优的训练轮次。对于 NODE 权重，我们还对所选训练轮次及其前两个训练轮次对应的三个检查点额外应用随机权重平均（SWA）。

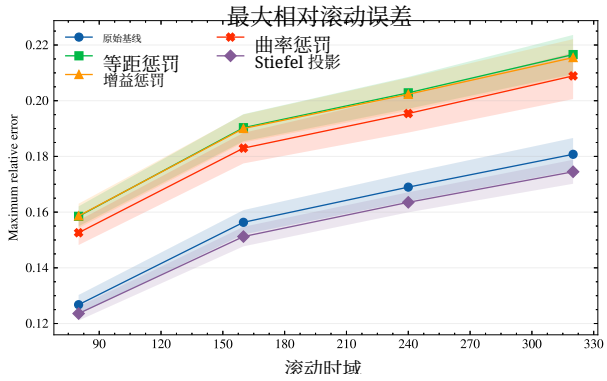


图 1: 不同正则化方法及无正则化的“原始基线”在细网格上、滚动时域取值$\in \{80, 160, 240, 320\}$ 的最大相对误差。不确定性以 *95%* 置信区间报告。

在所有测试时域上，都观察到相同的定性趋势。基于等距、增益和曲率正则项所生成的冻结自编码器训练的神经常微分方程，其平均相对滚动误差和最大相对滚动误差均高于原始基线。相比之下，斯蒂费尔变体是唯一始终具有竞争力、且通常略优于原始基线模型的方法。这一趋势在 $\varepsilon_{\text{mean}}$ and $\varepsilon_{\max}$ 的时域相关图中清晰可见（图 1–2）。

为使比较更加稳健，我们还计算了配对差值

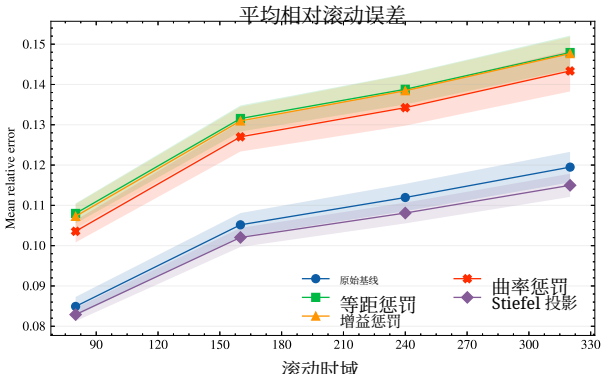


图 2： 不同正则化方法和无正则化“原始基线”在细网格上的滚动时域 $\in \{80, 160, 240, 320\}$内的平均相对误差。误差范围以 *95%* 置信区间表示。

，比较各方法与原始基线。对于每个自编码器随机种子和 NODE 随机种子，我们先对共同评估窗口集合上的滚动误差取平均，然后计算配对差值

$$\Delta = \varepsilon(\texttt{vanilla}) - \varepsilon(\texttt{method}).$$

因此，正值表示正则化方法优于基线。在所有匹配的自编码器/神经常微分方程实现中取平均后，Stiefel 是唯一具有正配对效应的方法，而其他所有方法在平均相对滚动误差和最大相对滚动误差上的配对差值均为负。具体而言，Stiefel 带来的改进在配对单侧检验中仍具有统计一致性，而曲率、增益和等距正则化方法的表现均逊于原始基线（见表 2）。

我们的解释是，即使自编码器的重构质量相当，正则化项 (a)–(c) 仍会偏向于那些使潜在 NODE 训练更困难、并降低长时域滚动预测表现的潜在表示。

在附录中，我们还报告了一项更严格的比较：在该比较中，不仅 AE 检查点，NODE 检查点也根据一个共享的验证滚动目标（在每个自编码器随机种子内定义）进行选择。相同的排序依然成立：即使各方法按验证滚动预测质量进行匹配，它们在外推上的表现仍然分化；这种额外的匹配也无法挽救曲率、增益或等距正则化变体。

4.3 潜在动力学

为了支持某些正则化项会导致有害的潜在几何结构这一假设，我们报告了学习得到的潜在动力学和解码器的内在诊断结果。

表3总结了三个量：(i) 学得的潜在动力学雅可比矩阵（而非解码器雅可比矩阵）的条件数，(ii) 解码器增益代理指标，

表 2：相对于原始基线的配对推演误差差异

方法	（A）配对 $\Delta\varepsilon_{\max}$	（B）配对 $\Delta\varepsilon_{\text{mean}}$
Stiefel 投影	$\mathbf{0.006} \pm 4.2 \cdot 10^{-3}$	$\mathbf{0.005} \pm 3.1 \cdot 10^{-3}$
曲率惩罚	$\mathbf{-0.028} \pm 8.7 \cdot 10^{-3}$	$\mathbf{-0.024} \pm 5.1 \cdot 10^{-3}$
增益惩罚	$\mathbf{-0.035} \pm 7.6 \cdot 10^{-3}$	$\mathbf{-0.028} \pm 4.6 \cdot 10^{-3}$
等距惩罚	$\mathbf{-0.036} \pm 7.9 \cdot 10^{-3}$	$\mathbf{-0.028} \pm 4.5 \cdot 10^{-3}$

配对差值定义为 $\Delta = \varepsilon(\mathit{vanilla}) - \varepsilon(\mathit{method})$，因此正值表示相较于未正则化基线有所改善。列 ***(A)*** 报告最大相对滚动误差的配对差值，列 ***(B)*** 报告平均相对滚动误差的配对差值。数值是对匹配的 *AE/NODE* 实例（2×16 对）取平均所得。结果针对细网格上的滚动时域 $H = 320$ 给出，*NODE* 检查点依据最佳验证误差选取。数值报告为均值 $\pm 95\%$ 置信区间半宽。

以及（iii）潜在跟踪误差。对于条件性统计量，我们采用条件数的运行级中位数（第50百分位数），然后对各次运行的该数值取平均；这样既能避免对极端离群值过度敏感，又能保留不同方法之间的排序。

表3：学习得到的潜在动力学的条件性、解码器增益、潜在误差

方法	动态条件数	解码器增益	潜在误差
Stiefel 投影	$\mathbf{124.2} \pm 11.5$	$\mathbf{4.428} \pm 9.7 \cdot 10^{-2}$	$\mathbf{0.988} \pm 0.022$
原始基线	$\mathbf{145.3} \pm 13.9$	$\mathbf{3.380} \pm 3.0 \cdot 10^{-3}$	$\mathbf{1.156} \pm 0.043$
曲率惩罚	$\mathbf{316.6} \pm 18.9$	$\mathbf{1.570} \pm 1.6 \cdot 10^{-2}$	$\mathbf{3.434} \pm 0.119$
等距惩罚	$\mathbf{407.2} \pm 19.8$	$\mathbf{1.000} \pm 7.0 \cdot 10^{-5}$	$\mathbf{4.066} \pm 0.122$
增益惩罚	$\mathbf{424.3} \pm 21.6$	$\mathbf{1.017} \pm 7.2 \cdot 10^{-3}$	$\mathbf{4.217} \pm 0.133$

细网格上滚动时域 $H = 320$ 处的内在诊断摘要。第一列报告所学潜在动力学雅可比矩阵条件数的中位数（第*50*百分位数），并在匹配的自编码器/神经常微分方程实现上取平均。第二列和第三列分别报告解码器增益代理指标和潜在跟踪误差。*NODE* 检查点按最佳验证误差选取。数值报告为均值 $\pm 95\%$ 置信区间半宽。

与原始基线相比，斯蒂费尔正则化自编码器的潜在动力学条件数更好，潜在误差更低，尽管其解码器增益略大。相比之下，曲率、等距和增益正则化项显著降低了解码器增益——其中等距和增益惩罚项几乎将其降至零——但这并未带来更好的滚动预测。相反，这些方法会使潜在动力学雅可比矩阵的条件数变差，并显著增大潜在误差。曲率惩罚的效果介于两者之间

在这方面如此，但其滚动预测质量仍明显逊于原始基线。

综合来看，这些诊断结果支持这样一个结论：在此设置下，仅减少解码器侧的扩张并不足够；潜在表示的结构及其对潜在动力学可学习性和条件数的影响更为重要。

5 结论

我们在受控的潜在降阶模型设置中，对基于几何的正则化方法进行了系统研究，该设置结合了 ADR 基准、卷积自编码器和采用解耦式自编码器/NODE 训练的潜在神经常微分方程。

尽管基于雅可比的正则化是控制解码器敏感度的一种自然方法，但我们的实验表明，与未正则化基线相比，此处考察的解码器侧正则项——近等距、方向增益和曲率——始终会削弱下游潜在 NODE 的性能。这种性能退化既体现在 NODE 训练动态中，也体现在更重要的长时域滚动预测质量上。

我们的结果支持这样一种假设：在此设置下，正则化项 (a)–(c) 会产生不太适合学习稳定连续时间潜在动力学的潜在表示。这些正则化项虽能减少解码器侧的误差放大，但无法弥补其伴随造成的潜在动力学条件数恶化和潜在状态跟踪精度下降。

相比之下，Stiefel 投影即使仅应用于单个编码器层，也能持续改善与条件数相关的潜在动力学诊断指标，并在平均意义上带来略好的滚动预测性能。这表明，当目标不仅是提高重构质量，还包括稳定、准确地学习潜在动力学时，较温和的结构约束可能比直接基于雅可比矩阵的惩罚项更有效。

我们希望这些发现能为未来研究提供参考，包括针对 ADR 类型系统的潜在 ROM、广义上的几何感知正则化，以及混合正则化器、条件数感知惩罚项和使用未冻结自编码器进行联合训练等更实用的策略。

我们在 https://github.com/ miosipof/adr_latent_dynamics 提供了完整的实现代码。

参考文献

[1] K. Lee 和 K.T.Carlberg，“利用深度卷积自编码器对非线性流形上的动力系统进行模型降阶”，《计算物理学杂志》，第 404 卷，文章编号 108973，2020年。 [在线]。网址：https://www.sciencedirect.com/science/article/pii/S0021999119306783

[2] L. Fulton、V. Modi、D. Duvenaud、D. Levin 和 A. Jacobson，“用于降阶可变形体仿真的潜空间动力学”，《计算机图形学论坛》，第 38 卷，第 379–391 页，2019年06月。[3] S. Fresca、L. Dede 和 A. Manzoni，“基于深度学习的非线性含时参数化偏微分方程降阶建模综合方法”，《科学计算杂志》，第 87 卷，2021年05月。[4]S. Fresca 和 A. Manzoni，“POD-DL-ROM：通过本征正交分解增强用于非线性参数化偏微分方程的基于深度学习的降阶模型”，《计算力学与工程方法》，第 388 卷，文章编号 114181，2022年。[在线]。网址：
https://www.sciencedirect.com/science/article/pii/S0045782521005120[5] W. D. Fries、X. He 和 Y. Choi，“LASDI：参数化潜空间动力学识别”，《计算力学与工程方法》，第 399 卷，文章编号 115436，2022年9月。 [在线]。网址：http://dx.doi.org/10.1016/j.cma.2022.115436[6]R. T. Q. Chen、Y. Rubanova、J. Bettencourt 和 D. K. Duvenaud，“神经常微分方程”，载于《神经信息处理系统进展》，S. Bengio、H. Wallach、H. Larochelle、K. Grauman、N. Cesa-Bianchi 和 R. Garnett 编，第 31 卷。Curran Associates, Inc.，2018年。 [在线]。网址：
https://proceedings.neurips.cc/paper files/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf[7] P. Kidger，“关于神经微分方程”，2022年。 [在线]。网址：
https://arxiv.org/abs/2202.02435[8]N. Kovachki、Z. Li、B. Liu、K. Azizzadenesheli、K. Bhattacharya、A. Stuart 和 A. Anandkumar，“神经算子：学习函数空间之间的映射及其在偏微分方程中的应用”，《机器学习研究杂志》，第 24 卷，第 89 期，第 1–97 页，2023年。 [在线]。网址：
http://jmlr.org/papers/v24/21-1524.html[9]E. Ballini、M. Gambarini、A. Fumagalli、L. Formaggia、A. Scotti 和 P. Zunino，“通过自编码器对参数化常微分方程进行模型降阶：保结构潜动力学与收敛性分析”，2025年。 [在线]。网址：https://arxiv.org/abs/2509.21280[10] N. Farenga、S. Fresca、S. Brivio 和 A. Manzoni，“非线性降阶建模中的潜动力学学习”，2024年。 [在线]。网址：https://arxiv.org/abs/2408.15183[11]S. Brivio、S. Fresca、N. R. Franco 和 A. Manzoni，“POD-DL-ROM模型的误差估计：一种通过本征正交分解增强的非线性参数化偏微分方程降阶建模深度学习框架”，《计算数学进展》，第 50 卷，第 3 期，文章编号 33，2024年4月。[在线]。网址：
https: //doi.org/10.1007/s10444-024-10110-1

[12]S. Rifai、P. Vincent、X. Muller、X. Glorot与Y. Bengio，“收缩自编码器：特征提取过程中的显式不变性”，载于第*28*届国际机器学习大会（*ICML*）论文集，2011年。[在线]。可获取：https://icml.cc/2011/papers/ 455 icmlpaper.pdf[13]P. Ghosh、M. S. M. Sajjadi、A. Vergari、M. J. Black与B. Sch¨olkopf，“从变分自编码器到确定性自编码器”，$CoRR$，第abs/1903.12436卷，2019年。[在线]。可获取：http://arxiv.org/abs/1903.12436[14]A. Kumar、B. Poole与K. Murphy，“基于松弛单射概率流的正则化自编码器”，2020年。[在线]。可获取：https://proceedings.mlr.press/v108/kumar20a/kumar20a.pdf[15] Y. Lee、S. Yoon、M. Son与F. Park，“用于等距表示学习的正则化自编码器”，2022年，出版商版权：© 2022 ICLR 2022——第10届国际学习表征会议。保留所有权利。；第10届国际学习表征会议，ICLR 2022；会议日期：2022年4月25日至2022年4月29日。[16] P. Nazari、S. Damrich与F. A. Hamprecht，“几何自编码器——所见即所解码”，载于第*40*届国际机器学习大会论文集，第202卷，2023年。[17] J. Z. Kim、N. Perrin-Gilbert、E. Narmanli、P. Klein、C. R. Myers、I. Cohen、J. J. Waterfall与J. P. Sethna，“γ-vae：用于揭示高维数据中涌现的低维几何结构的曲率正则化变分自编码器”，*arXiv*，第abs/2403.01078卷，2024年。[在线]。可获取：https://api.semanticscholar.org/CorpusID:268230957[18] N. J. Higham，“极分解的计算及其应用”，*SIAM*科学与统计计算杂志，1986年。[在线]。可获取：https://epubs.siam.org/doi/10.1137/0907079[19] P.-A. Absil、R. Mahony与R. Sepulchre，矩阵流形上的优化算法。普林斯顿大学出版社，2008年。[20] C. Moustapha、B. Piotr、G. Edouard、D. Yann与U. Nicolas，“帕塞瓦尔网络：提升对对抗样本的鲁棒性”，国际机器学习大会论文集，2017年4月。[21] L. Huang、X. Liu、B. Lang、A. Yu与B. Li，“正交权重归一化：深度神经网络中多个相依斯蒂费尔流形上优化问题的求解方法”，*AAAI*人工智能大会论文集，第32卷，2017年9月。[22] J. Li、F. Li与S. Todorovic，“基于凯莱变换的Stiefel流形高效黎曼优化”，$CoRR$，第abs/2002.01113卷，2020年。[在线]。可获取：https://arxiv.org/abs/2002.01113

[23]M. F. Hutchinson，“拉普拉斯平滑样条影响矩阵迹的随机估计量”，《统计学通信——模拟与计算》，1990年。 [在线]。可获取：https://www.tandfonline.com/doi/abs/10.1080/03610919008812866

A 损失函数与几何正则化项的形式化定义

A.1 正则化项概述

– **近等距惩罚：** 通过随机 Hutchinson 估计，强制使 $J_D(z)^\top J_D(z) \approx \mathbf{I}$，即实现解码器的局部近等距。使用权重 $= 0.1$。

– **方向增益惩罚：** 使 $\|J_D(z)v\|$ 在随机方向上的典型取值与目标值相匹配；此项控制的是平均雅可比尺度，而非最坏情况扩张。使用权重 $= 0.1$。– **方向曲率惩罚：** 惩罚 $J_D(z)v$ 沿同一方向 v的变化，即解码器的方向二阶变化。使用权重 $= 1.0$。– **Stiefel 投影：** 强制解码器第一层权重矩阵满足列正交归一性，作为一种部分结构正则化器，而非对完整解码器雅可比矩阵的直接约束。

A.2 符号与张量形状

Let

– $u \in \mathbb{R}^{1\times 32\times 32}$ 表示归一化标量场快照；– $z \in \mathbb{R}^{1\times 4\times 4}$ 表示潜在场；– $E_\phi : \mathbb{R}^{1\times 32\times 32} \to \mathbb{R}^{1\times 4\times 4}$ 为编码器；– $D_\psi : \mathbb{R}^{1\times 4\times 4} \to \mathbb{R}^{1\times 32\times 32}$ 为解码器。

我们用

$$d := 1 \cdot 4 \cdot 4 = 16, \qquad n := 1 \cdot 32 \cdot 32 = 1024$$

展平后的潜在维度和环境维度。我们用 vec(·) 表示向量化，因此

$$\mathrm{vec}(z) \in \mathbb{R}^d, \qquad \mathrm{vec}(D_\psi(z)) \in \mathbb{R}^n.$$

归一化锚点。我们还使用常数场

$$u_{\min} \coloneqq -\mathbf{1} \in \mathbb{R}^{1\times 32\times 32},$$

这对应于经最小–最大归一化至 $[-1,1]$ 后的物理最小场值。

A.3 自编码器架构

编码器和解码器均为空间连贯的卷积网络，包含残差块和双线性下采样/上采样：

$$E_\phi = \mathrm{Conv}_{\mathrm{out}} \circ \mathrm{Down}^{(L)} \circ \mathrm{Conv}_{\mathrm{in}}, \qquad D_\psi = \mathrm{Conv}_{\mathrm{out}} \circ \mathrm{Up}^{(L)} \circ \mathrm{Conv}_{\mathrm{in}}.$$

每个下采样/上采样阶段都由一个预激活残差块和一个按因子 $1/2$ （下采样）或 2（上采样）进行的双线性插值组成，并重复 L 次。在我们的实验中， $L=3$，对应于分辨率序列

$$32 \;\to\; 16 \;\to\; 8 \;\to\; 4.$$

A.4 自编码器训练目标

给定一个小批量 $\{u^{(b)}\}_{b=1}^{B}$，定义

$$z^{(b)} = E_\phi(u^{(b)}), \qquad \hat{u}^{(b)} = D_\psi(z^{(b)}).$$

重建损失。 批次重建损失为

$$\mathcal{L}_{\mathrm{mse}}(\phi,\psi) = \frac{1}{B}\sum_{b=1}^{B} \frac{1}{n} \left\| \mathrm{vec}\big(\hat{u}^{(b)}\big) - \mathrm{vec}\big(u^{(b)}\big) \right\|_2^2 .$$

锚点损失。 此外，我们还对常数最小场的重建进行惩罚：

$$\mathcal{L}_{\mathrm{anc}}(\phi,\psi) = \frac{1}{n} \left\| \mathrm{vec}\big(D_\psi(E_\phi(u_{\min}))\big) - \mathrm{vec}(u_{\min}) \right\|_2^2 .$$

正则化自编码器目标。 令 $\mathcal{R}(D_\psi; z)$ 表示在潜变量批次 z 上计算的解码器侧正则项（定义见 A.6 节）。自编码器预训练使用

$$\mathcal{L}_{\mathrm{AE}}(\phi,\psi) = \mathcal{L}_{\mathrm{mse}}(\phi,\psi) + \lambda_{\mathrm{reg}}(e)\,\mathcal{R}(D_\psi; z) + w_{\mathrm{anc}}\,\mathcal{L}_{\mathrm{anc}}(\phi,\psi),$$

其中， e 表示训练轮次索引， $\lambda_{\mathrm{reg}}(e)$在预热期后线性递增：

$$\lambda_{\mathrm{reg}}(e) = \begin{cases} 0, & e \le e_{\mathrm{warm}}, \\ \lambda_{\mathrm{reg}} \dfrac{e - e_{\mathrm{warm}}}{E - e_{\mathrm{warm}}}, & e_{\mathrm{warm}} < e \le E. \end{cases}$$

自编码器训练期间的斯蒂费尔投影。对于斯蒂费尔变体，且仅在预热期之后，首个解码器卷积层的权重会在每次优化器步骤后投影到斯蒂费尔型约束上（见 A.7 节）。

A.5 潜在 NODE 模型和 NODE 训练目标

令 $u(t) \in \mathbb{R}^{1\times 32\times 32}$ 表示一条轨迹，并令 $\mu \in \mathbb{R}^{n_\mu}$ 为参数向量（在我们的实验中，$n_\mu = 3$）。对于在时刻（u_0，……，u_ℓ）采样的滚动窗口（t_0，……，t_ℓ），我们定义

$$z_0 = E_\phi(u_0), \qquad \dot{z}(t) = f_\theta(t, z(t), \mu),$$

其中，潜在常微分方程通过显式 Runge–Kutta 方法积分（在我们的实验中使用二阶 Ralston 方法）。解码预测为

$$\hat{u}_i = D_\psi(z(t_i)), \qquad i = 0, \ldots, \ell.$$

滚动损失。滚动损失为

$$\mathcal{L}_{\mathrm{roll}}(\phi, \psi, \theta) = \frac{1}{B}\sum_{b=1}^{B} \frac{1}{\ell+1}\sum_{i=0}^{\ell} \frac{1}{n} \left\| \mathrm{vec}\big(\hat{u}_i^{(b)}\big) - \mathrm{vec}\big(u_i^{(b)}\big) \right\|_2^2 .$$

窗口重建损失。我们还监测滚动窗口内的重建情况：

$$\mathcal{L}_{\mathrm{rec}}(\phi, \psi) = \frac{1}{B}\sum_{b=1}^{B} \frac{1}{\ell+1}\sum_{i=0}^{\ell} \frac{1}{n} \left\| \mathrm{vec}\big(D_\psi(E_\phi(u_i^{(b)}))\big) - \mathrm{vec}\big(u_i^{(b)}\big) \right\|_2^2 .$$

锚点损失。NODE 训练期间也包含相同的锚点损失 $\mathcal{L}_{\mathrm{anc}}$。

NODE 训练目标。令 $z_\star$ 表示评估解码器侧正则项时所处的潜在状态（例如，窗口中的最后一个潜在状态）。用于潜在动力学训练的损失函数为

$$\mathcal{L}_{\mathrm{NODE}}(\phi, \psi, \theta) = \mathcal{L}_{\mathrm{roll}}(\phi, \psi, \theta) + \lambda_{\mathrm{recon}}\, \mathcal{L}_{\mathrm{rec}}(\phi, \psi) + \lambda_{\mathrm{reg}}\, \mathcal{R}(D_\psi; z_\star) + w_{\mathrm{anc}}\, \mathcal{L}_{\mathrm{anc}}(\phi, \psi).$$

当自编码器被冻结时，仅优化 θ。当自编码器解冻时，ϕ、ψ和 θ 被联合优化（自编码器参数使用较小的学习率）。在我们的主要实验中，我们在 NODE 训练期间保持自编码器冻结，从而实际上

$$\mathcal{L}_{\mathrm{NODE}}(\phi, \psi, \theta) = \mathcal{L}_{\mathrm{roll}}(\phi, \psi, \theta).$$

A.6 解码器侧正则项

所有正则化器均应用于解码器 D_ψ ，并在潜变量批次上求值

$$z \in \mathbb{R}^{B\times 1\times 4\times 4}.$$

在自编码器预训练期间，我们对当前批次使用 $z = E_\phi(u)$。在 NODE 训练期间，我们在当前滚动窗口中选定的潜在状态 $z_\star$ 处评估正则化项。

A.6.1 近等距惩罚

定义展平后的解码器输出相对于展平后的潜状态的雅可比矩阵：

$$J_{D_\psi}(z) \coloneqq \frac{\partial \,\mathrm{vec}(D_\psi(z))}{\partial \,\mathrm{vec}(z)} \in \mathbb{R}^{n\times d}.$$

其拉回格拉姆矩阵为

$$G_\psi(z) \coloneqq J_{D_\psi}(z)^\top J_{D_\psi}(z) \in \mathbb{R}^{d\times d}.$$

精确的近等距惩罚为

$$\mathcal{R}_{\mathrm{a}}(D_\psi; z) = \frac{1}{B}\sum_{b=1}^{B} \left\| G_\psi\big(z^{(b)}\big) - \mathbf{I}_d \right\|_F^2 .$$

这会直接惩罚解码器度量偏离潜在空间中的欧几里得度量。

精确计算。 在精确版本中，通过对标准基向量 $e_i \in \mathbb{R}^d$ 计算 $J_{D_\psi}(z)e_i$ 的前向模式雅可比-向量积（JVP），组装完整的雅可比矩阵，然后构造 $G_\psi(z) = J_{D_\psi}(z)^\top J_{D_\psi}(z)$。

随机 Hutchinson 估计。设

$$A_\psi(z) \coloneqq J_{D_\psi}(z)^\top J_{D_\psi}(z) - \mathbf{I}_d.$$

对于任意满足 $\mathbb{E}[vv^\top] = \mathbf{I}_d$的各向同性探测向量 $v \in \mathbb{R}^d$ ，有

$$\mathbb{E}_v \|A_\psi(z)v\|_2^2 = \|A_\psi(z)\|_F^2.$$

因此，同一量也可以通过随机方式估计为

$$\mathcal{R}_a(D_\psi; z) = \frac{1}{B}\sum_{b=1}^{B} \mathbb{E}_v \left[\left\| \big(J_{D_\psi}(z^{(b)})^\top J_{D_\psi}(z^{(b)}) - \mathbf{I}_d\big)v \right\|_2^2 \right].$$

在实践中，这通常通过有限个探测向量近似实现：使用 JVP–VJP 计算$(J^\top J)v$，而无需显式构造 J。

分离约定。如果将潜在输入 z 分离，正则化项只会更新解码器参数 ψ 。否则，梯度也可能沿着 z 对这些组件的依赖关系，向上游流入编码器和/或潜在动力学模型。

A.6.2 方向增益惩罚

第二个惩罚项用于控制解码器雅可比矩阵的典型方向增益，而不强制实现完整的近等距：

$$\mathcal{R}_{\mathrm{b}}(D_\psi; z) = \frac{1}{B}\sum_{b=1}^{B} \mathbb{E}_v \left[\left(\|J_{D_\psi}(z^{(b)})v\|_2 - \alpha \right)^2 \right],$$

其中， v 是潜在空间中的随机单位向量， α 是目标增益（在我们的主要实验中等于 1）。该正则化项使典型方向增益与目标增益相匹配，并且与雅可比矩阵的弗罗贝尼乌斯尺度有关，但它无法控制 $J_{D_\psi}(z)$ 的完整奇异值谱。

有限差分 JVP 近似。 方向导数通过单边有限差分近似，

$$J_{D_\psi}(z)v \;\approx\; \frac{\mathrm{vec}(D_\psi(z+\varepsilon v)) - \mathrm{vec}(D_\psi(z))}{\varepsilon},$$

其中， $\varepsilon > 0$ 是较小的有限差分步长。

A.6.3 方向曲率惩罚

为探测解码器沿采样方向的非线性，我们采用一种基于方向导数变化的曲率相关惩罚项：

$$\mathcal{R}_{\mathrm{c}}(D_\psi; z) = \frac{1}{B}\sum_{b=1}^{B} \mathbb{E}_v \left[\left\| \frac{J_{D_\psi}(z^{(b)}+\varepsilon v)v - J_{D_\psi}(z^{(b)})v}{\varepsilon} \right\|_2^2 \right].$$

范数中的增量近似于二阶方向导数，

$$\frac{J_{D_\psi}(z+\varepsilon v)v - J_{D_\psi}(z)v}{\varepsilon} \approx \left(\nabla^2 \mathrm{vec}(D_\psi)(z)\right)[v, v],$$

因此，该正则化项会惩罚较大的方向曲率。

实现细节。与方向增益惩罚不同，该项中的雅可比-向量积在 z 和 $z+\varepsilon v$ 处均为精确计算（精度受自动微分精度限制）。

A.7 第一层解码器的斯蒂费尔投影

设首个解码器卷积层的权重张量为

$$W \in \mathbb{R}^{c_{\text{out}} \times c_{\text{in}} \times k_1 \times k_2}.$$

将输入和空间维度展平后，可得到一个矩阵

$$A \in \mathbb{R}^{c_{\text{out}} \times p}, \qquad p \coloneqq c_{\text{in}} k_1 k_2, \qquad A = \text{reshape}(W).$$

当 $c_{\text{out}} \geq p$ 时，我们在每次优化器步后将 A 投影到具有正交列的矩阵集合上：

$$A_{\text{proj}} = A\,(A^\top A)^{-1/2},$$

因此

$$A_{\text{proj}}^\top A_{\text{proj}} = \mathbf{I}_p.$$

逆平方根通过特征分解计算。若

$$A^\top A = Q \Lambda Q^\top, \qquad \Lambda = \text{diag}(\lambda_i),$$

then

$$(A^\top A)^{-1/2} = Q\,\text{diag}(\lambda_i^{-1/2})\,Q^\top,$$

为保证数值稳定性，将特征值下限截断为一个较小的正值。

该投影仅在所选层中强制正交列。因此，它改善了该层的条件数，但并不意味着完整非线性解码器具备近等距或曲率控制。

回退情形。 若 $c_{\text{out}} < p$ （未用于我们的主要设置），实现将回退为逐行归一化，而非真正的 Stiefel 投影。

B 全阶数据集生成

我们考虑单位正方形上的参数化对流–扩散–反应（ADR）问题

$$\Omega = [0,1]^2,$$

其中状态变量为 $u = u(x,t)$，参数向量

$$\mu = (\mu_1, \mu_2, \mu_3) \in \mathbb{R}^3.$$

在实现中，μ_1 为扩散系数，而 (μ_2, μ_3) 指定高斯源项的中心。反应系数固定为

$$c = 1,$$

且时变对流场为

$$b(t) = (\cos t, \sin t).$$

源项选为

$$f(x;\mu_2,\mu_3) = 10\exp\left(-\frac{(x_1-\mu_2)^2+(x_2-\mu_3)^2}{0.072}\right).$$

初始条件为

$$u(x,0) = 0.$$

空间离散化。 我们使用 dolfinx，在均匀三角形网格上采用一阶拉格朗日有限元求解全阶模型，该网格包含 31×31 个细分单元。由此得到 $32 \times 32 = 1024$ 个标量自由度，因此每个快照都可以重塑为一个张量，形状为

$$\mathbb{R}^{1\times 32\times 32}.$$

时间离散化。 全阶轨迹在时间区间

$$[0,T], \qquad T = 10\pi,$$

上采用后向欧拉时间离散化计算，共有

$$N_t^{\mathrm{FOM}} = 1000$$

个时间步。因此，细网格时间步为

$$\Delta t_{\mathrm{FOM}} = \frac{T}{1000},$$

且每条轨迹包含

$$N_t^{\mathrm{FOM}} + 1 = 1001$$

存储的快照，包括 $t = 0$ 和 $t = T$。

对于降阶模型训练，我们通过每隔五个细网格步长进行一次下采样，构建粗时间网格。因此，

$$\Delta t_{\mathrm{train}} = 5\,\Delta t_{\mathrm{FOM}},$$

每条轨迹包含

$$N_t^{\mathrm{train}} + 1 = 201$$

粗网格快照。

时间划分。为区分训练和长时域评估模式，我们定义

$$T_1 = 4\pi, \qquad T_2 = 5\pi.$$

由此得到如下时间划分：

– $[0, T_1]$：主要训练区间；– $(T_1,\ T_2]$：时间插值区间；– $(T_2,\ T]$：时间外推区间。

在降阶模型训练期间，轨迹在粗网格上采样，而长时域滚动评估也在细网格上报告。

参数域与数据集划分。完整参数域为

$$\mathcal{P} = [10^{-2},\, 6 \cdot 10^{-2}] \times [0.3,\, 0.7]^2.$$

训练参数从内盒中采样

$$\mathcal{P}_{\text{train}} = [2 \cdot 10^{-2},\, 5 \cdot 10^{-2}] \times [0.4,\, 0.6]^2.$$

更确切地说，我们构建每个参数维度包含10个点的张量网格，从而得到

$$10^3 = 1000$$

参数三元组。这些参数三元组被随机划分为80%的训练参数集和20%的验证参数集。

为评估参数空间插值，我们使用训练网格各坐标方向中点构成的张量网格，从而得到

$$9^3 = 729$$

同一内盒内的插值测试参数。

为评估外推能力，我们从

$$200$$

$\mathcal{P} \setminus \mathcal{P}_{\text{train}}$中均匀采样参数，即从外盒中排除训练框后的区域进行采样。

存储轨迹。 对于每个采样参数，我们存储细网格轨迹

$$u_{\text{fine}} \in \mathbb{R}^{1001 \times 32 \times 32}$$

以及粗网格轨迹

$$u_{\text{coarse}} \in \mathbb{R}^{201 \times 32 \times 32}.$$

所得数据集用于逐快照自编码器预训练，以及后续的潜在动力学训练和滚动评估。

C 评估协议

我们在一组共同的采样轨迹窗口上评估所有方法，并在各方法间使用完全相同的窗口，以便进行配对比较。

评估划分与时域。令 split 表示数据集划分之一（训练集、插值集或外推集），令 field 表示用于评估的时间网格（u 细网格或 u 粗网格）。在主要实验中，我们在细网格（$\bar{\text{u}}$ 细网格）上进行评估，滚动时域为

$$H \in \{80, 160, 240, 320\},$$

其中，时间跨度以所选评估网格的索引数来衡量。

对于选定的数据划分和场变量，我们首先从 HDF5 文件中确定可用的时间段。随后仅从该时间段中抽取滚动预测窗口，以确保每个起始索引 k_0 都满足

$$k_0 + H \le k_{\max},$$

其中，$k_{\max}$ 是该时间段内最后一个可用的时间索引。

滚动预测窗口的抽样。我们抽取有限个评估窗口

$$(\mu_j, k_{0,j}), \qquad j = 1, \ldots, N_{\text{win}},$$

其中，μ_j 表示一个参数实例的索引，$k_{0,j}$ 是滚动预测窗口的起始时间索引。在默认配置下，我们抽取

$$N_\mu = 100$$

个参数索引（有放回抽样），并为每个抽取的参数采样

$$N_{\text{starts}} = 10$$

从允许的区间内均匀选取起始时间。这会得到

$$N_{\text{win}} = N_\mu \cdot N_{\text{starts}} = 1000$$

个采样窗口。

为确保即使出现重复的 $(\mu,\ k_0)$ 对也能精确配对，每个采样窗口都会分配一个唯一标识符

$$\texttt{window_id} \in \{0, \ldots, N_{\text{win}} - 1\}.$$

配对比较和多次运行聚合使用的是此标识符，而非仅使用元组 $(\mu,\ k_0)$。

归一化。所有滚动预测均使用相同的归一化常数（$u_{\min}$，$u_{\max}$），这些常数仅根据训练集划分中粗时间网格上的训练参数和训练时间索引计算一次。所有方法共用这些归一化常数。

窗口滚动预测。对于采样窗口（μ， k_0），令

$$u_0, u_1, \ldots, u_H$$

表示所选评估网格上对应的全阶状态，并令

$$t_0, t_1, \ldots, t_H$$

为其绝对时间。之所以显式使用绝对时间，是因为潜在常微分方程依赖于时间。评估流程如下:

1. 对初始状态 u_0 进行归一化并将其编码，以获得潜在初始条件

$$z_0 = E_\phi(u_0);$$

2. 对潜在常微分方程进行积分

$$\dot{z}(t) = f_\theta(t, z(t), \mu)$$

在绝对时间网格（t_0，.....， t_H）上，使用所选的显式龙格-库塔积分器;

3. 对潜在预测进行解码，以获得

$$\hat{u}_i = D_\psi(z(t_i)), \qquad i = 0, \ldots, H;$$

4. 在物理空间中比较解码轨迹与全阶轨迹。

滚动误差指标。令

$$e_i = \hat{u}_i - u_i, \qquad \|e_i\| = \|\mathrm{vec}(e_i)\|_2, \qquad \|u_i\| = \|\mathrm{vec}(u_i)\|_2.$$

所有汇总指标均在丢弃平凡的初始点 $i = 0$ 后计算，因为 $\hat{}u_0$ 是直接从编码后的初始条件获得的。因此，误差是在以下范围内汇总的

$$i = 1, \ldots, H.$$

对于每个窗口，我们报告:

$$\mathrm{abs_mean} = \frac{1}{H}\sum_{i=1}^{H}\|e_i\|, \qquad \mathrm{abs_max} = \max_{1\le i\le H}\|e_i\|, \qquad \mathrm{abs_fin} = \|e_H\|,$$

and

$$\mathrm{rel_mean} = \frac{1}{H}\sum_{i=1}^{H}\frac{\|e_i\|}{\|u_i\|+\epsilon}, \qquad \mathrm{rel_max} = \max_{1\le i\le H}\frac{\|e_i\|}{\|u_i\|+\epsilon}, \qquad \mathrm{rel_fin} = \frac{\|e_H\|}{\|u_H\|+\epsilon},$$

分母中加上一个较小的数值常数 $\epsilon = 10^{-8}$ 。我们还记录终止时刻均方误差

$$\mathrm{mse_fin} = \frac{1}{n}\,\|\mathrm{vec}(e_H)\|_2^2.$$

内在诊断。 除了滚动误差外，我们还在一部分评估窗口上计算局部内在诊断。默认情况下，我们使用前

$$N_{\text{intr}} = 80$$

采样窗口（由窗口 ID 标识），并在

$$N_{\text{steps}} = 6$$

滚动窗口内近似均匀分布的时间索引处评估诊断指标。

对于每个选定的窗口和内在时间索引，我们记录：

– 潜在动力学向量场相对于潜在状态的雅可比矩阵的奇异值，

$$J_f(t, z, \mu) = \frac{\partial f_\theta(t, z, \mu)}{\partial z},$$

通过自动微分精确计算；

– 相应的局部动力学诊断指标

$$\text{dyn_sigma_max} = \sigma_{\max}(J_f), \qquad \text{dyn_sigma_min} = \sigma_{\min}(J_f),$$

$$\text{dyn_cond} = \frac{\sigma_{\max}(J_f)}{\sigma_{\min}(J_f) + 10^{-12}};$$

– 解码器增益代理指标

$$\text{dec_gain} \approx \|J_D(z)\|_2,$$

通过对 $J_D(z)^\top J_D(z)$进行幂迭代计算；

– 潜在跟踪误差

$$\text{latent_err} = \|z_{\text{pred}} - E_\phi(u_{\text{true}})\|_2,$$

即预测的潜在状态与同一时刻自编码器编码的真实状态之间的距离。

确定性解码器增益估计。 为避免内在诊断中引入额外的蒙特卡洛方差，解码器增益的幂迭代采用固定探针初始化（或等效的确定性设定随机种子的规则）。这样可以确保解码器增益在重复评估中具有可复现性，并可直接跨方法比较。

逐方法汇总。 对于主要报告的统计量，首先在每个匹配的自编码器/神经常微分方程实现内对滚动误差取平均值，然后再跨实现进行汇总。窗口级聚合仅用于单次运行内。

内在诊断分别在选定的内在窗口和内在时间索引上进行汇总。

当多个评估运行拼接在一起时，保存的表格会同时包含方法标签和运行标识符：

(方法、运行、窗口 ID)。

这确保摘要仍按方法正确索引，并且多次运行聚合不会合并不同的运行或重复窗口。

配对比较。 为将某种方法与未正则化基线进行比较，我们使用完全相同的采样窗口进行配对检验。配对依据唯一的窗口标识符进行（汇集多次运行时还依据运行标识符），因此重复的 (μ, k_0) 对仍作为不同观测值处理。对于选定的标量指标（在我们的主要实验中为 rel mean），我们计算相对于基线的逐窗口差值，在每种方法匹配的自编码器/神经常微分方程实现内求平均，并报告：

– 配对差值的均值；– 用于检验该方法是否优于基线的单侧 Wilcoxon 符号秩 p 值。

该协议确保所有方法比较都在相同的滚动预测窗口上进行，不会遗漏重复样本，并能对重复的神经常微分方程运行进行正确记账。

D 补充结果

我们还进行了补充滚动实验，使用事后定义的共享验证滚动目标进行神经常微分方程检查点选择。

这些实验旨在确定，通过匹配训练/验证滚动上的性能，能否减轻方法在外推划分上的滚动预测表现差异。

我们在表 4 中展示，总体模式相同：采用 Stiefel 投影进行正则化的自编码器训练出的神经常微分方程随机种子，其表现略优于原始基线；而其他正则化方法的滚动误差则更高。

表 4：相对于原始基线的配对推演误差差异

方法	$\Delta\varepsilon_{\max}$	Std	95%置信区间	p-值
Stiefel 投影	0.009	0.019	0.010	0.058
等距惩罚	-0.012	0.020	0.010	0.988
增益惩罚	-0.014	0.018	0.010	0.997
曲率惩罚	-0.016	0.020	0.011	0.997

方法	$\Delta\varepsilon_{\text{mean}}$	Std	95%置信区间	p 值
Stiefel 投影	0.005	0.012	0.006	0.080
等距惩罚	-0.015	0.012	0.006	1.000
增益惩罚	-0.016	0.011	0.006	1.000
曲率惩罚	-0.018	0.012	0.007	1.000

配对差值定义为 $\Delta = \varepsilon(\mathit{vanilla}) - \varepsilon(\mathit{method})$，因此正值表示相较于未正则化基线有所改善。数值取匹配的 *AE/NODE* 实例的平均值。结果展示了细网格上滚动时域$H = 320$ 的情况，*NODE* 检查点依据所有方法均达到的共享验证均方误差目标选取。不确定性以 *95%* 置信区间表示。报告的 p值对应于针对备择假设 $\Delta > 0$ 的单侧*Wilcoxon*符号秩检验，即该方法优于基线。