面向时变参数化问题的非侵入式数据驱动降阶建模

Junming Duan[a,*], Jan S. Hesthaven[a]

[a]计算数学与仿真科学教席，洛桑联邦理工学院，洛桑，*1015*，瑞士

arXiv:2303.02986v1 [math.NA] 6 Mar 2023

摘要

降阶模型对于多查询或实时问题不可或缺。然而，为时变参数化问题构建高效的降阶模型仍面临诸多挑战。对于时变非线性问题，使用线性降阶空间效率低下，尤其是对于输运主导问题。非线性通常需要通过超降阶技术（如 DEIM）处理，但这类技术具有侵入性，并且依赖于参数仿射依赖的假设。本文提出并研究了一种适用于时变参数化问题的非侵入式降阶建模方法。该方法完全由数据驱动，并自然地分为离线和在线阶段。在离线阶段，训练一个由编码器和解码器组成的卷积自编码器，以实现降维。编码器将全阶解快照压缩到非线性流形或低维降阶／潜在空间中。解码器则能够从潜空间恢复全阶解。为处理时变问题，本文采用高阶动态模态分解（HODMD），对每个参数对应的潜空间轨迹进行建模。在在线阶段，首先利用 HODMD 模型获取新时刻的潜变量，然后采用插值技术恢复新参数值对应的潜变量，最后通过解码器恢复全阶解。本文通过若干数值测试表明，该方法能够快速、准确地预测新时刻和新参数值下未见过的全阶解，其中包括输运主导问题。

关键词：降阶建模、卷积自编码器、动态模态分解、参数化问题、时变问题、非线性问题

*通讯作者电子邮箱：junming.duan@epfl.ch (Junming Duan), jan.hesthaven@epfl.ch (Jan S.Hesthaven)预印本已提交至计算物理学杂志 *3*月*7*日，*2*

1. 引言

在科学和工程领域的诸多现实应用中，例如优化、控制或不确定性量化，许多时变问题都可描述为参数化偏微分方程（PDE）。参数可能源于物理性质、几何构型、初始条件或边界条件等。此类参数化偏微分方程可通过标准数值方法求解，例如有限差分法、有限体积法、谱方法、有限元法等。然而，要获得足够精确的解，即全阶模型（FOM）的高保真解，通常需要数千个自由度，因而对计算资源提出了很高要求。此外，在多查询或实时任务中，需要针对大量不同的参数值求解这些偏微分方程。在这种情况下，开发高效的低维模型，使其能够在可控精度损失的前提下，快速计算新时间点和新参数值下的关注输出，具有重要意义。

过去几十年来，降阶基模型（RBM） [17, 31] 已被开发出来以解决这一问题。RBM 的核心思想是：在采样时刻和参数值下收集的一组全阶解中寻找低维结构，并用代理模型取代全阶模型（FOM）。这些全阶解称为快照，用于描述解流形的时空动力学。最受欢迎的方法之一是基于投影的降阶基模型 [4], ，它通过基的线性组合张成解流形的低维近似（降维子空间），然后将全阶模型投影到该低维降维子空间中，以获得 RBM。为了得到最优的线性低维近似，采用本征正交分解（POD），基于奇异值分解（SVD）对快照矩阵进行分解，从而获得占主导地位的正交归一模态。基于投影的方法分为离线和在线阶段，大部分计算成本在离线阶段完成，包括快照的收集和 RBM 的构建。由于计算成本随 RBM 的维数变化，该模型可在在线阶段用于快速响应请求。

对于具有认证误差控制的椭圆型、稳态和线性问题，降阶基模型的发展已得到充分研究。读者可参阅专著 [17, 31], 近期的综述文章 [16], 及其中的参考文献。额外的挑战来自于

全阶模型中的非线性，这导致降阶基模型中非线性项的计算成本随全阶模型的高维度而增长。在这种情况下，采用超降阶策略 [32] 。这些方法大多依赖于通过对非线性算子进行插值的稀疏采样，例如缺失点估计 [2], 、经验插值法（EIM） [3], 、离散经验插值法（DEIM） [8], 、近似张量高斯–牛顿法（GNAT） [6], 等。然而，这些方法通常具有侵入性，需要访问原始全阶求解器，并且高效实现可能并非易事。这促使人们设计非侵入式方法。在 [18],中，使用神经网络来近似从参数空间到降阶系数的映射，并证明其效率较高。高斯过程回归也被用于在 [14]中构建非侵入式降阶基模型。

另一个困难是，当 Kolmogorov n-宽度衰减非常缓慢时，POD 生成的线性降阶空间的维数可能会很高，例如，对流方程 [11]的移动前沿解。在这种情况下，RBM 的高维性会限制其效率。在 [7]中，采用 h自适应来扩充降阶基。二次算子推断[29] 被提出用于构建非侵入式的基于投影的降阶基模型。神经网络也被用于构建低维降阶空间，例如， [24, 25], 中使用的一种自编码器，其性能被证明优于经典的基于POD的线性子空间。

本文关注时变问题，因此 RBM 应当捕捉其内在动力学。 [27, 28], 中提出了自适应降阶基和基于低秩更新的采样，可将其视为 r自适应。循环神经网络（RNN），更具体地说，长短期记忆（LSTM）网络，也已被用于 [25], 中对动力学进行建模，但可用的仅是在某些离散时刻的状态，而非完整轨迹。另一种方法是将时间视为另一个参数，并使用神经网络建立时间和参数到降阶系数的映射 [12, 13]。

在本研究中，我们将基于动态模态分解（DMD）的高阶扩展，即高阶动态模态分解，构建潜在动力学的代理模型。此类方法已用于时变问题的模型降阶，并可通过Koopman谱理论加以解释 [5]。其核心思想是选择合适的测量值/坐标，并将原始非线性动力系统转换为作用于这些新坐标上的线性系统。在DMD中，全阶解直接用作坐标；而在高阶动态模态分解中，则引入时间延迟

嵌入以丰富坐标。传统动态模态分解算法的第一步是基于本征正交分解进行降维，而我们将使用非线性降维得到的潜变量作为输入，从而获得更高效的低维表示。

本文提出并研究了一种面向时变参数化问题的新型非侵入式数据驱动降阶建模方法，该方法分为离线和在线阶段。在离线阶段，深度卷积自编码器生成降阶／潜在空间；研究表明，相较于线性子空间，这种方法能够得到效率更高的近似。训练好的编码器为快照生成潜变量，并采用高阶动态模态分解为每个训练参数值下的潜在动力学构建代理模型。在在线阶段，对于给定的新时间和参数值，首先利用 HODMD 模型获取该时刻的潜变量，然后通过插值获得新参数值对应的潜变量，最后由解码器重构全阶解。我们的方法完全由数据驱动，不依赖于对底层物理模型的先验知识。我们通过三个测试验证该方法，分别是一维 Burgers 方程、二维瑞利-贝纳德对流和二维 Kelvin-Helmholtz 不稳定性。结果表明，对于具有低维潜空间的输运主导问题，该方法表现良好。它能够快速、准确地预测新时间和新参数值下未见过的全阶解。该方法采用松耦合设计，因此可以调整各个组件并优化相应参数，从而便于控制各部分的误差。

本文的结构安排如下。第2 节介绍全阶模型以及快照的采集。第3节将详细介绍我们的数据驱动非侵入式方法。第4 节给出一些数值结果，以验证该方法的有效性和性能；第5节总结全文。

2. 全阶模型公式

T本文研究一组一般的参数化偏微分方程（PDE s)

$$\begin{cases} \dfrac{\partial}{\partial t}\boldsymbol{u}(t,\boldsymbol{x};\boldsymbol{\omega}) + \boldsymbol{f}(\boldsymbol{u},\nabla\boldsymbol{u},\nabla^2\boldsymbol{u},\cdots;\boldsymbol{\omega}) = 0, \\ \boldsymbol{u}(t=0,\boldsymbol{x};\boldsymbol{\omega}) = \boldsymbol{u}_0(\boldsymbol{x};\boldsymbol{\omega}), \end{cases} \tag{2.1}$$

其中 $\boldsymbol{u}(t,\boldsymbol{x};\boldsymbol{\omega}) \in \mathbb{R}^m$ 是解向量，包含 m 个分量，时间为 $t \in [0,+\infty)$，空间坐标为 $\boldsymbol{x} \in \mathbb{R}^d$，取决于参数向量 $\boldsymbol{\omega} \in \mathbb{R}^{d_p}$，且 $\boldsymbol{f}$ 是一个涉及 $\boldsymbol{u}$空间导数的非线性函数。对于给定参数值 $\boldsymbol{\omega}$，可采用有限差分、有限体积、有限元或谱方法求解模型（2.1），得到以下半离散全阶模型（FOM）

$$\begin{cases} \dfrac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{u}_h(t;\boldsymbol{\omega}) + \boldsymbol{f}_h(\boldsymbol{u}_h;\boldsymbol{\omega}) = 0, \\ \boldsymbol{u}_h(t=0;\boldsymbol{\omega}) = \boldsymbol{u}_{h,0}(\boldsymbol{\omega}), \end{cases} \tag{2.2}$$

其中 $\boldsymbol{f}_h$ 是离散空间算子。通过对（2.2）进行时间积分，可得到全阶解 $\boldsymbol{u}_h(t;\ \boldsymbol{\omega}) \in \mathbb{R}^{n_h}$ 在时间 t。在本文中，$n_h = m \times n_y \times n_x$，其中 n_x，n_y分别是 x方向和 y方向上的空间自由度数量（$n_y = 1$ 对于一维问题），尽管我们的方法可以毫无困难地扩展到三维。

3. 非侵入式数据驱动方法

本节将介绍一些现有结果，然后针对离散全阶模型（2.2）给出我们的降阶建模方法。

3.1. 线性模型降阶：*POD-Galerkin* 方法

经典的基于投影的降阶模型广泛应用于许多领域 [4, 17, 31]。在这类方法中，解流形由一个降阶的 $n_{\mathtt{rb}}$ 维线性空间 $\mathcal{S}_{n_{\mathtt{rb}}}$ 近似，该空间由矩阵 $\mathcal{V}_{\mathtt{rb}} \in \mathbb{R}^{n_h \times n_{\mathtt{rb}}}$ 的 $n_{\mathtt{rb}}$ 列张成

$$\boldsymbol{u}_h(t;\boldsymbol{\omega}) = \mathcal{V}_{\mathtt{rb}}\boldsymbol{u}_{\mathtt{rb}}(t;\boldsymbol{\omega}) + \overline{\boldsymbol{u}_h}, \tag{3.1}$$

其中，$\boldsymbol{u}_{\mathtt{rb}}$ 是降阶基函数的系数，$\boldsymbol{u}_h$ 是参考解。本征正交分解是获取正交归一模态（或 $\mathcal{V}_{\mathtt{rb}}$）的最常用方法之一，这些模态基于奇异值分解（SVD）张成最佳线性子空间。如果在采样时刻 t_1, $t_2, \cdots, t_{n_t}$和参数值 $\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \cdots, \boldsymbol{\omega}_{n_p}$处收集快照，并构造快照矩阵

$$S = [\boldsymbol{u}_h(t_1;\boldsymbol{\omega}_1) - \overline{\boldsymbol{u}_h}, \boldsymbol{u}_h(t_{n_t};\boldsymbol{\omega}_1) - \overline{\boldsymbol{u}_h}, \ldots, \boldsymbol{u}_h(t_{n_t};\boldsymbol{\omega}_{n_p}) - \overline{\boldsymbol{u}_h}] \in \mathbb{R}^{n_h \times n_s}$$

快照矩阵的奇异值分解为

$$S = \mathcal{V}\Sigma\widetilde{\mathcal{V}}^{\mathrm{T}},$$

其中 $\Sigma = \mathrm{diag}\,\{\sigma_1, \sigma_2, \cdots, \sigma_r\} \in \mathbb{R}^{n_h \times n_s}$ 包含奇异值 $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_r \geqslant 0$ 和 $r = \min\{n_h, n_s\}$。这里，$\mathcal{V} \in \mathbb{R}^{n_h \times n_h}$ 和 $\mathcal{V} \in \mathbb{R}^{n_s \times n_s}$ 的列是正交归一左、右奇异向量。随后，矩阵 $\mathcal{V}_{\mathtt{rb}}$ 选取为 $\mathcal{V}$的前 $n_{\mathtt{rb}}$ 列，而维数 $n_{\mathtt{rb}}$ 可由相对能量阈值确定

$$\sum_{l=1}^{n_{\mathtt{rb}}} \sigma_l^2 \Big/ \sum_{l=1}^{r} \sigma_l^2 \geqslant 1 - \epsilon, \tag{3.2}$$

且 ϵ 是一个接近零的小数。根据施密特-埃卡特-扬定理，[10, 34], 矩阵$\mathcal{V}_{\mathtt{rb}}$ 使投影误差最小

$$\sum_{i=1}^{n_t}\sum_{j=1}^{n_p} \left\| \boldsymbol{u}_h(t_i;\boldsymbol{\omega}_j) - \mathcal{V}_{\mathtt{rb}}\mathcal{V}_{\mathtt{rb}}^{\mathrm{T}}\boldsymbol{u}_h(t_i;\boldsymbol{\omega}_j) \right\|_{\mathbb{R}^{n_h}}^2 = \min_{\mathcal{W}\in\mathcal{S}_{n_{\mathtt{rb}}}} \sum_{i=1}^{n_t}\sum_{j=1}^{n_p} \left\| \boldsymbol{u}_h(t_i;\boldsymbol{\omega}_j) - \mathcal{W}\mathcal{W}^{\mathrm{T}}\boldsymbol{u}_h(t_i;\boldsymbol{\omega}_j) \right\|_{\mathbb{R}^{n_h}}^2$$

在集合 $\mathcal{S}_{n_{\mathtt{rb}}} = \{\mathcal{W} \in \mathbb{R}^{n_h \times n_{\mathtt{rb}}} :\ \mathcal{W}^{\mathrm{T}}\mathcal{W} = I_{n_{\mathtt{rb}}}\}$ 上，即所有秩为 $n_{\mathtt{rb}}$ 的正交归一基。

将 代入 ansatz（3.1）到半离散全阶模型（2.2）中，可得到投影的降阶基模型

$$\begin{cases} \dfrac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{u}_{\mathtt{rb}}(t;\boldsymbol{\omega}) + \mathcal{V}_{\mathtt{rb}}^{\mathrm{T}}\boldsymbol{f}_h(\mathcal{V}_{\mathtt{rb}}\boldsymbol{u}_{\mathtt{rb}} + \overline{\boldsymbol{u}_h}) = 0, \\ \boldsymbol{u}_{\mathtt{rb}}(t=0;\boldsymbol{\omega}) = \mathcal{V}_{\mathtt{rb}}^{\mathrm{T}}\left(\boldsymbol{u}_{h,0}(\boldsymbol{\omega}) - \overline{\boldsymbol{u}_h}\right). \end{cases} \tag{3.3}$$

尽管 POD-伽辽金方法已得到广泛应用，但对于时变非线性问题，要准确近似快照矩阵，线性降阶空间的维数可能需要很大。Kolmogorov n-宽度 [22] 为量化最优 $n_{\mathtt{rb}}$ 维线性试探子空间 $\mathcal{S}_{n_{\mathtt{rb}}}$ 对离散解流形 $\mathcal{M}_h$的近似提供了一种方法，其定义如下：

$$d_{n_{\mathtt{rb}}}(\mathcal{M}_h) = \inf_{\mathcal{S}_{n_{\mathtt{rb}}}} \sup_{f\in\mathcal{M}_h} \inf_{g\in\mathcal{M}_h} \|f - g\|.$$

在某些情况下，例如参数空间中正则性较高的椭圆型问题 [17], ，Kolmogorov n-宽度会呈指数衰减，$d_{n_{\mathtt{rb}}}(\mathcal{M}_h) \leqslant Ce^{-cn_{\mathtt{rb}}}$。然而，对于输运主导问题，例如线性平流方程的移动前沿解，线性子空间的效率并不高，$d_{n_{\mathtt{rb}}}(\mathcal{M}_h) \leqslant Cn_{\mathtt{rb}}^{-1/2}$ [11]。

将 RBM（3.3）应用于非线性问题时还会遇到另一个难点：这需要使用 $\mathcal{V}_{\mathtt{rb}}\boldsymbol{u}_{\mathtt{rb}}$，而计算成本会随全阶模型的维数增长；全阶模型的维数通常很大。必须采用超降阶技术来处理非线性项，例如（离散）经验插值方法（EIM/DEIM） [3, 8], ，这些方法旨在恢复参数的仿射依赖。然而，在一般应用中，参数仿射依赖这一假设可能并不成立，而且近似成本可能很高。

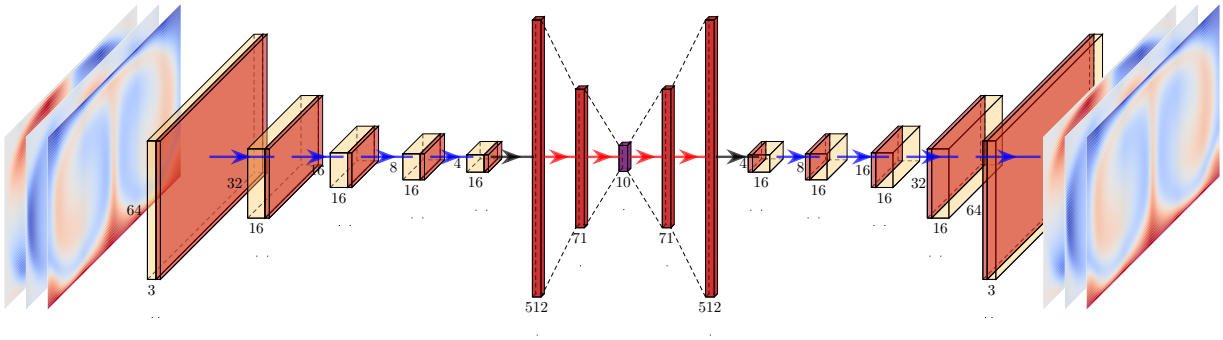


图3.1：A　　用于第3节二维瑞利-贝纳德对流的卷积自编码器架构　　4.2.

3.2. 基于自编码器的非线性潜在流形

上一节提到的难点促使我们在降阶空间中采用非线性试探空间，并对非线性项进行非线性近似。神经网络已被证明是进行函数逼近的强大工具。在本研究中，我们采用自编码器进行降维。

自编码器由编码器和解码器两部分组成，分别表示为 $\mathfrak{E}(\ \cdot\ ;\ \Theta_{\mathtt{enc}})$ 和$\mathfrak{D}(\ \cdot\ ;\ \Theta_{\mathtt{dec}})$，其中 $\Theta_{\mathtt{enc}}$ 和 $\Theta_{\mathtt{dec}}$ 是神经网络中的参数（权重和偏置），通过训练获得。两者均由多个层组成：前者将全阶解压缩为低维潜变量，后者则根据潜变量重构全阶解。若仅使用全连接层，当输入数据规模较大时，自编码器往往包含极大量的参数，尤其是在处理二维或三维全阶模型的解时。这种情况下，训练需要大量数据，成本也可能很高。由于结构化网格上的全阶解可视为图像，我们采用卷积神经网络（CNN），这类网络已在许多图像相关任务的压缩方面展现出优异性能。卷积神经网络的效率很高，因为参数共享减少了参数总数。本文采用卷积自编码器（CAE），该方法也用于 [24, 25], 以生成低维潜在流形。图3.1展示了卷积自编码器的一种典型架构。给定输入 $\boldsymbol{u}_h \in \mathbb{R}^{m\times n_y\times n_x}$，编码器部分由多个堆叠层组成，如下所示：

$\mathfrak{E}(\ \cdot;\ \Theta_{\mathtt{enc}}) = \boldsymbol{h}_{n_L}(\ \cdot;\Theta_{\mathtt{enc},\ n_L}) \circ \sigma_{n_L-1}(\boldsymbol{h}_{n_L-1}(\ \cdot;\Theta_{\mathtt{enc},\ n_L-1})) \circ \cdots \sigma_1(\boldsymbol{h}_1(\ \cdot;\Theta_{\mathtt{enc},\ 1}))$，其中第 i 层 $\boldsymbol{h}_i,\ i=1,\ \cdots,\ n_L$ 具有权重和偏置 $\Theta_{\mathtt{enc},\ i}$ 可以是卷积层

或全连接层，而 $\sigma_i(\cdot)$ 是逐分量应用的相应非线性激活函数，例如修正线性单元（ReLU）、双曲正切（Tanh）、Sigmoid 线性单元（SiLU）等。我们将使用定义如下的 SiLU 函数

$$\texttt{SiLU}(x) = x\ \texttt{sigmoid}(x) = \frac{x}{1+\exp(-x)}.$$

卷积层会缩减全阶解的空间维度并改变通道数。经过多个卷积层后，输出会重塑为一个长向量，再经过若干全连接层转换为低维潜向量。对于解码器部分，其结构与编码器部分对称，

$$\mathfrak{D}(\cdot\,;\Theta_{\texttt{dec}}) = \bar{\boldsymbol{h}}_1(\cdot\,;\Theta_{\texttt{dec},1}) \circ \sigma_2\left(\bar{\boldsymbol{h}}_2(\cdot\,;\Theta_{\texttt{dec},2})\right) \circ \cdots \sigma_{n_L}\left(\bar{\boldsymbol{h}}_{n_L}(\cdot\,;\Theta_{\texttt{dec},n_L})\right),$$

其中 $\bar{\boldsymbol{h}}_i$ 是反卷积层或全连接层，其输入和输出维度与 $\boldsymbol{h}_i$ 编码器中的对应层相反。全连接层、卷积层和反卷积层的具体定义可参见 PyTorch 文档 [26]。$n_{\texttt{latent}}$ 维中心潜变量编码了主要特征和物理结构，并张成一个低维空间，该空间通常是非线性的。如果编码器和解码器中都只使用一个不含激活函数的全连接层，那么自编码器与 POD 方法类似，并且 $n_{\texttt{latent}}$ 就是截断奇异值分解中所选主导模态的数量。

自编码器的损失函数是重构误差，其定义为

$$\mathcal{L} = \frac{1}{n_t n_p}\sum_{i=1}^{n_t}\sum_{j=1}^{n_p} \|\boldsymbol{u}_h(t_i,\boldsymbol{\omega}_j) - \mathfrak{D}(\mathfrak{E}(\boldsymbol{u}_h(t_i,\boldsymbol{\omega}_j);\Theta_{\texttt{enc}});\Theta_{\texttt{dec}})\|_2^2, \tag{3.4}$$

其中，$\boldsymbol{u}_h(t_i,\boldsymbol{\omega}_j)$ 是时间 t_i 和参数值 $\boldsymbol{\omega}_j$下的全阶解，求和是在训练集上进行的。在训练过程中，采用基于梯度的方法来寻找最优参数 $\Theta_{\texttt{enc}},\Theta_{\texttt{dec}} = \arg\min \mathcal{L}$。自编码器训练完成后，我们使用$\Theta_{\texttt{enc}},\Theta_{\texttt{dec}}$编码器来获得潜变量 $\boldsymbol{q} = \mathfrak{E}(\boldsymbol{u}_h;\Theta_{\texttt{enc}})$，这些潜变量张成潜在流形。

注 3.1。为避免过拟合，在损失函数（3.4）中添加正则化项是有益的，例如 $\lambda(\|\Theta_{\texttt{enc}}\|_1 + \|\Theta_{\texttt{dec}}\|_1)$ 或 $\lambda\left(\|\Theta_{\texttt{enc}}\|_2^2 + \|\Theta_{\texttt{dec}}\|_2^2\right)$，其中 λ为正则化参数。该 L^2 正则化等价于标准随机梯度下降（SGD）优化器中的权重衰减技术，在该技术中，正则化项直接出现在

参数的更新中。我们使用 PyTorch 中 ADAM 优化器实现的权重衰减技术，因为事实证明它具有较好的稳健性。

注释 3.2. 在数值测试中，我们也采用早停技术；换言之，如果验证集上的误差在给定数量的训练轮次内未能低于历史最佳误差，就会终止训练。

注释 3.3. 在 [12, 13], 本文中，使用卷积自编码器生成低维潜空间，但由于全阶解并非在结构化网格上获得，因此首先将其重塑为图像。重塑算子可能会破坏空间相关性，因此本文仅考虑结构化网格上的全阶解。

3.3. 用于潜轨迹时间建模的高阶动态模态分解

为实现在新时刻进行快速预测，我们构建了一个适用于潜变量完整轨迹的代理模型。对时间动力学建模的一种方法是使用神经网络，例如循环神经网络（RNN） [25]。然而，RNN 只能给出离散时刻的状态，而无法提供完整轨迹。此外，众所周知，RNN 中的误差会迅速累积。本文采用 DMD [5, 33], ，这是一种数据驱动方法，已被用于时变问题的模型降阶。经典 DMD 利用本征正交分解在空间中构建降阶基，即 DMD 使用线性子空间，正如第 3.1 节所述，这可能需要大量基函数。我们提出先使用自编码器获取潜变量，以降低空间自由度，然后基于 DMD 构建潜在动力学。

给定参数 $\boldsymbol{\omega}$ 在 $t = t_1, \cdots, t_{n_t}$ 下的潜变量 $\boldsymbol{q} = \mathfrak{E}(\boldsymbol{u}_h; \Theta_{\text{enc}})$，我们将其收集为快照矩阵 $\mathcal{Q} = [\boldsymbol{q}(t_1, \boldsymbol{\omega}) \cdots \boldsymbol{q}(t_{n_t}, \boldsymbol{\omega})]$。然后，可以利用线性动力系统来近似潜变量的非线性动力学

$$\mathcal{Q}_2 = \mathcal{A}\mathcal{Q}_1, \quad \mathcal{Q}_1 = [\boldsymbol{q}(t_1), \cdots \boldsymbol{q}(t_{n_t-1})], \quad \mathcal{Q}_2 = [\boldsymbol{q}(t_2), \cdots \boldsymbol{q}(t_{n_t})],$$

其中 ω 被省略。算法1 用于计算 DMD 特征值、模态和幅值。

算法1： DMD

输入： $\mathcal{Q}_1$, $\mathcal{Q}_2$, ϵ

输出： Λ, Θ

1 计算截断奇异值分解：$\mathcal{Q}_1 = \mathcal{X}\Sigma \qquad \widetilde{\mathcal{X}}^{\mathrm{T}}$，其中 ϵ 相对能量阈值（3.2）为选择主导奇异向量的数量 L；

2 求得约化动态模态分解算子 $\widetilde{\mathcal{A}} = \mathcal{X}^{\mathrm{T}}\mathcal{Q}_2\widetilde{\mathcal{X}}\Sigma^{-1}$;

3 计算特征分解 $\widetilde{\mathcal{A}}\mathcal{Y} = \mathcal{Y}\Lambda$ 以获得约化动态模态分解模态 $\mathcal{Y}$ 和 DMD 特征值 $\Lambda = \mathrm{diag}\ \{\lambda_1, \cdots, \lambda_L\}$;

4 动态模态分解算子 $\mathcal{A}$ 的 DMD 模态定义为 $\Xi = [\boldsymbol{\xi}_1, \cdots, \boldsymbol{\xi}_L] = \mathcal{Q}_2 \qquad \widetilde{\mathcal{X}}\Sigma^{-1}\mathcal{Y}$;

5 通过 $\boldsymbol{q}(\tilde{t})$ =预测 $t = \tilde{t}$ 处的潜变量 $\sum_{l=1}^{L} a_l\boldsymbol{\xi}_l\lambda_l^{(\tilde{t-t_1})/\Delta t}$，其中 $\boldsymbol{a} = (a_1, a_2, \cdots, a_L)^{\mathrm{T}}$ DMD 振幅。

DMD 振幅可由 $\boldsymbol{a} = \Xi^{\dagger}\boldsymbol{q}(t_1)$ 计算。或者，采用 [21]，中提出的最优幅值，其通过求解以下优化问题获得，并使重构结果与所有快照之间的误差最小化

$$\arg\min_{\boldsymbol{a}} \|\mathcal{Q}_1 - \Xi\ \mathrm{diag}\{a_1, a_2, \cdots, a_L\}\ V_{\mathrm{and}}\|_F^2,$$

以及范德蒙德矩阵

$$V_{\mathrm{and}} = \begin{bmatrix} 1 & \lambda_1 & \cdots & \lambda_1^{L-1} \\ 1 & \lambda_2 & \cdots & \lambda_2^{L-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \lambda_L & \cdots & \lambda_L^{L-1} \end{bmatrix},$$

可以考虑。通过求解线性方程组来计算该解

$$\left((\Xi^*\Xi) \circ (\overline{V_{\mathrm{and}}V_{\mathrm{and}}^*})\right)\boldsymbol{a} = \overline{\mathrm{diag}(V_{\mathrm{and}}\mathcal{Q}_1^*\Xi)}. \tag{3.5}$$

对于本文所研究的潜在动力学，DMD 不够准确，无法捕捉完整的动力学。因此，本文提出采用 HODMD [23]，，它可以被视为一种叠加的 DMD，能够在滑动窗口中包含更多信息。HODMD 中使用了所谓的时间延迟嵌入，

$$\widehat{\boldsymbol{q}}_k = [\boldsymbol{q}(t_k)^{\mathrm{T}}, \boldsymbol{q}(t_{k+1})^{\mathrm{T}}, \cdots, \boldsymbol{q}(t_{k+n_{\mathtt{delay}}-1})^{\mathrm{T}}]^{\mathrm{T}},$$

快照矩阵构造为汉克尔矩阵

$$\widehat{\mathcal{Q}}_1 = \left[\widehat{\boldsymbol{q}}_1, \widehat{\boldsymbol{q}}_2, \cdots, \widehat{\boldsymbol{q}}_{n_t - n_{\texttt{delay}}}\right], \quad \widehat{\mathcal{Q}}_2 = \left[\widehat{\boldsymbol{q}}_2, \widehat{\boldsymbol{q}}_3, \cdots, \widehat{\boldsymbol{q}}_{n_t - n_{\texttt{delay}}+1}\right].$$

然后求解以下问题

$$\widehat{\mathcal{Q}}_2 = \mathcal{A}\widehat{\mathcal{Q}}_1$$

使用算法1，但对于第四步中的预测，我们只取前 $n_{\texttt{latent}}$个分量来恢复 $n_{\texttt{latent}}$ 维潜变量。

备注 3.4。参数$n_{\texttt{delay}}$ 不应接近于 1，因为在这种情况下没有足够的时间延迟嵌入，从而导致结果不准确。同样也不能选择过大的$n_{\texttt{delay}}$，因为$\widehat{\quad}$ $\widehat{\quad}$在 $\mathcal{Q}_1$中只有少量经过修改的快照， $\mathcal{Q}_2$，因此末尾时刻的结果将不准确。不同$n_{\texttt{delay}}$ 下的结果将在第4节进行比较。

备注 3.5。潜变量的维度较低，因此在我们的测试中不对奇异值分解进行截断，即将 ϵ 设为零。在这种情况下， $L = n_{\texttt{latent}} n_{\texttt{delay}}$。

3.4. 潜在流形中对新参数进行插值

如果只考虑一个参数（$d_p = 1$），则可以使用拉格朗日插值或线性插值，或者使用流形插值 [35]，以获得新参数值 $\widetilde{\boldsymbol{\omega}}$处的潜变量

$$\boldsymbol{q}(t, \widetilde{\boldsymbol{\omega}}) = \mathcal{I}(t, \widetilde{\boldsymbol{\omega}}) = \mathcal{I}\left(\boldsymbol{q}(t, \boldsymbol{\omega}_1), \cdots, \boldsymbol{q}(t, \boldsymbol{\omega}_{n_p})\right).$$

对于更多参数，可以采用径向基函数（RBF）插值

$$\mathcal{I}(t, \widetilde{\boldsymbol{\omega}}) = \sum_{s=1}^{n_p} \boldsymbol{w}_s \varphi(\|\boldsymbol{\omega} - \boldsymbol{\omega}_s\|),$$

其中 $\varphi(r) = \varphi(\|\boldsymbol{\omega} - \boldsymbol{\omega}_s\|)$ 为核函数，例如薄板核 $\varphi(r) = r^2 \ln(r+1)$，每个参数 $\boldsymbol{\omega}_s$ 对应的权重 $\boldsymbol{w}_s$ 通过求解最小二乘问题获得

$$\sum_{s=1}^{n_p} \boldsymbol{w}_s \varphi(\|\boldsymbol{\omega}_p - \boldsymbol{\omega}_s\|) = \boldsymbol{q}(t, \boldsymbol{\omega}_p),\ p = 1, \cdots, n_p.$$

关于径向基函数插值的研究已十分广泛，感兴趣的读者可参阅 [30] 了解更多详情。

注 3.6. 本节中的参数化DMD方法与 [1], 中的分区方法类似，不同之处在于我们使用自编码器生成潜空间，而非经典POD。其他参数化DMD方法还包括对DMD特征对或算子进行插值 [19], ，采用流形插值 [15]。在一项最新研究中 [9], ，自编码器与SINDy方法相结合，用于周期性问题。

3.5. 非侵入式数据驱动降阶模型

基于前面三节的讨论，我们提出了一种工作流程，称为CAE-PHODMD，用于构建非侵入式数据驱动降阶基模型，见算法2-3，分别描述离线和在线阶段。在离线阶段，我们收集适当的快照，训练自编码器，并为每个参数值构建 HODMD 模型。在在线阶段，为了预测新时间和新参数值下的全阶解，我们首先获取新时间下的潜变量，然后在参数空间中插值以获得新的潜变量，最后使用解码器恢复全阶解。

算法 2： CAE-PHODMD：离线阶段

输入： 在 t_1、$\cdots$、t_n 处的全阶解快照 $_t$ 以及 ω_1、$\cdots$、ω_n $_p$

输出： 自编码器：$\mathfrak{E}$、$\mathfrak{D}$；高阶动态模态分解：Λ、Θ，对于每个参数值 ω_1、$\cdots$、ω_n $_p$

1 训练自编码器以获得最优参数 Θ_{enc}、Θ_{dec}；

2 通过编码器计算潜变量 $\left[\boldsymbol{q}(t_1,\boldsymbol{\omega}_1)\ \ldots\boldsymbol{q}(t_{n_t},\boldsymbol{\omega}_1)\ldots\ldots,\boldsymbol{q}(t_{n_t},\boldsymbol{\omega}_{n_p})\right]=$

$\left[\mathfrak{E}(\boldsymbol{u}_h(t_1,\boldsymbol{\omega}_1);\ \Theta_{\text{enc}})\ \ldots\mathfrak{E}(\boldsymbol{u}_h(t_{n_t},\boldsymbol{\omega}_1);\ \Theta_{\text{enc}})\ldots\ldots,\mathfrak{E}(\boldsymbol{u}_h(t_{n_t},\boldsymbol{\omega}_{n_p});\ \Theta_{\text{enc}})\right]$;

3 根据快照为每个参数值 $\boldsymbol{\omega}_j$ 构建 HODMD模型

潜变量 $[\boldsymbol{q}(t_1,\boldsymbol{\omega}_j)\ \cdots\boldsymbol{q}(t_{n_t},\boldsymbol{\omega}_j)]$。

算法 3： CAE-PHODMD：在线阶段

输入： 一个新的时间 $\tilde{t}$ 和参数值 $\tilde{\boldsymbol{\omega}}$

输出： 全阶解 $\boldsymbol{u}($ $\tilde{t},\tilde{\boldsymbol{\omega}})$

1 使用高阶动态模态分解进行时间预测：$\mathcal{Q}(\boldsymbol{\omega}_1)\ \cdots,\ \mathcal{Q}(\boldsymbol{\omega}_{n_p})\Rightarrow\left[\boldsymbol{q}(\tilde{t},\ \boldsymbol{\omega}_1)\ \cdots,\ \boldsymbol{q}(\tilde{t},\ \boldsymbol{\omega}_{n_p})\right]$;

2 对新的参数值进行插值：$\left[\boldsymbol{q}(\tilde{t},\ \boldsymbol{\omega}_1)\ \cdots,\ \boldsymbol{q}(\tilde{t},\ \boldsymbol{\omega}_{n_p})\right]\Rightarrow\boldsymbol{q}(\tilde{t},\ \tilde{\boldsymbol{\omega}})$;

3 通过解码器重构：$\boldsymbol{u}_h($ $\tilde{t},\tilde{\boldsymbol{\omega}})=\mathfrak{D}(\boldsymbol{q}(\ \tilde{t},\ \tilde{\boldsymbol{\omega}});\Theta_{\text{dec}})$.

4. 数值结果

本节将对典型的参数化偏微分方程进行数值测试。我们将评估给定测试时间和参数值 $(t_{\texttt{test}}, \boldsymbol{\omega}_{\texttt{test}})$ 下的以下相对误差

$$\epsilon_{\texttt{CAE}}(t_{\texttt{test}}, \boldsymbol{\omega}_{\texttt{test}}) = \frac{\|\boldsymbol{u}_h(t_{\texttt{test}}, \boldsymbol{\omega}_{\texttt{test}}) - \mathfrak{D}(\mathfrak{E}(\boldsymbol{u}_h(t_{\texttt{test}}, \boldsymbol{\omega}_{\texttt{test}}); \Theta_{\texttt{enc}}); \Theta_{\texttt{dec}})\|_2}{\|\boldsymbol{u}_h(t_{\texttt{test}}, \boldsymbol{\omega}_{\texttt{test}})\|_2},$$
$$\epsilon_{\texttt{latent}}(t_{\texttt{test}}, \boldsymbol{\omega}_{\texttt{test}}) = \frac{\|\boldsymbol{q}(t_{\texttt{test}}, \boldsymbol{\omega}_{\texttt{test}}) - \mathcal{I}\left(t_{\texttt{test}}, \boldsymbol{\omega}_{\texttt{test}}\right)\|_2}{\|\boldsymbol{q}(t_{\texttt{test}}, \boldsymbol{\omega}_{\texttt{test}})\|_2},$$
$$\epsilon_{\texttt{CAE-PHODMD}}(t_{\texttt{test}}, \boldsymbol{\omega}_{\texttt{test}}) = \frac{\|\boldsymbol{u}_h(t_{\texttt{test}}, \boldsymbol{\omega}_{\texttt{test}}) - \mathfrak{D}(\mathcal{I}\left(t_{\texttt{test}}, \boldsymbol{\omega}_{\texttt{test}}\right); \Theta_{\texttt{dec}})\|_2}{\|\boldsymbol{u}_h(t_{\texttt{test}}, \boldsymbol{\omega}_{\texttt{test}})\|_2},$$

其中，$\boldsymbol{q}(t_{\texttt{test}}, \boldsymbol{\omega}_{\texttt{test}}) = \mathfrak{E}(\boldsymbol{u}_h(t_{\texttt{test}}, \boldsymbol{\omega}_{\texttt{test}}); \Theta_{\texttt{enc}})$ 是由编码器获得的潜变量向量，而 $\mathcal{I}(t_{\texttt{test}}, \boldsymbol{\omega}_{\texttt{test}}) = \mathcal{I}\left(\boldsymbol{q}(t_{\texttt{test}}, \boldsymbol{\omega}_1) \cdots, \boldsymbol{q}(t_{\texttt{test}}, \boldsymbol{\omega}_{n_p})\right)$ 是新参数值处潜变量的插值。整个测试集的误差定义为

$$E_{\texttt{CAE}} = \frac{1}{N_{t_{\texttt{test}}} N_{\boldsymbol{\omega}_{\texttt{test}}}} \sum_{t_{\texttt{test}}} \sum_{\boldsymbol{\omega}_{\texttt{test}}} \epsilon_{\texttt{CAE}}(t_{\texttt{test}}, \boldsymbol{\omega}_{\texttt{test}}),$$
$$E_{\texttt{latent}} = \frac{1}{N_{t_{\texttt{test}}} N_{\boldsymbol{\omega}_{\texttt{test}}}} \sum_{t_{\texttt{test}}} \sum_{\boldsymbol{\omega}_{\texttt{test}}} \epsilon_{\texttt{latent}}(t_{\texttt{test}}, \boldsymbol{\omega}_{\texttt{test}}),$$
$$E_{\texttt{CAE-PHODMD}} = \frac{1}{N_{t_{\texttt{test}}} N_{\boldsymbol{\omega}_{\texttt{test}}}} \sum_{t_{\texttt{test}}} \sum_{\boldsymbol{\omega}_{\texttt{test}}} \epsilon_{\texttt{CAE-PHODMD}}(t_{\texttt{test}}, \boldsymbol{\omega}_{\texttt{test}}),$$

其中 $N_{t_{\texttt{test}}}$ 和 $N_{\boldsymbol{\omega}_{\texttt{test}}}$ 分别为测试时刻数和参数值数。我们的自编码器基于 PyTorch 库 [26]实现。在所有测试中，均采用初始学习率为 0.001 、批量大小为 32 的 mini-batch ADAM 优化器进行训练。并使用步长为 50 、衰减率为 0.95 的 StepLR 调度器，使学习率为 $\lambda = 0.001 \times 0.95^{\lfloor n/50 \rfloor}$，其中 n 是当前训练轮次。在卷积自编码器的训练中，采用早停技术以避免过拟合。具体而言，如果验证集上的最佳重构误差连续 100 个训练轮次未得到改善，则终止训练。所有测试均使用 SciPy 软件包提供的线性插值。采用高阶插值也不会产生差异。

4.1. 一维 *Burgers* 方程

4.1.1. 设置

此示例用于验证我们的方法在一维 Burgers 方程上的有效性

$$u_t + \left(\frac{1}{2}u^2\right)_x = \frac{1}{Re}u_{xx}, \ x \in [0, 2], \ u(0, t) = u(2, t) = 0,$$

其精确解为 [25]

$$u(x,t;Re)=\frac{\frac{x}{t+1}}{1+\sqrt{\frac{t+1}{t_0}}\exp(Re\frac{x^2}{4t+4})},\quad t_0=\exp(Re/8),$$

其中， Re为雷诺数。在该测试中，具有128个均匀空间自由度的精确解作为全阶解。数据集由三部分组成：训练集包含 10 Re 个均匀分布于 $[100,800]$ 中的样本，以及101个均匀分布于 $[0,2]$, 的时间点；验证集包含4个随机 Re，以及在相同参数和时间域内随机选取的20个时间点；测试集包含2个随机 Re ，以及在相同参数和时间域内随机选取的10个时间点。所有卷积层和反卷积层的卷积核大小均为5，步幅和填充均为2。

4.1.2. 结果

通过网格搜索确定最佳卷积自编码器架构，搜索不同的权重衰减 10^{-8}、 10^{-9}、 10^{-10}、 10^{-11}、卷积层数量 4、5、6，以及潜空间的维度 $n_{\texttt{latent}}=2$、4、6、8、10。在验证集上获得最佳重构误差的架构被选为最终卷积自编码器模型，如表 4.1 所示。具有其他 $n_{\texttt{latent}}$ 的卷积自编码器模型也用于比较。

图 4.1展示了测试集中的两个特定重构全阶解，分别对应于 $Re=290.8$、646.0 处的 $t=1.85$, 相应的误差为 $\epsilon_{\texttt{CAE}}=1.002\times10^{-3}$、 7.679×10^{-4}。可以看出， $n_{\texttt{latent}}=2$即可在无振荡的情况下获得精确的重构结果，这表明了卷积自编码器的高效率。

卷积自编码器-PHODMD 的性能取决于潜空间的维数 $n_{\texttt{latent}}$，以及 HODMD 中所使用的时间延迟嵌入数量 $n_{\texttt{delay}}$ 。图 4.2 给出了 $E_{\texttt{CAE-PHODMD}}$ 在不同 $n_{\texttt{latent}}$、$n_{\texttt{delay}}$下的误差，以及 $E_{\texttt{CAE}}$ 在不同 $n_{\texttt{latent}}$下的误差。对于$n_{\texttt{latent}}=2$，4，卷积自编码器-PHODMD 的误差随着 $n_{\texttt{delay}}$ 的增加而衰减；而对于 $n_{\texttt{latent}}=6$、8和 10，当 $n_{\texttt{delay}}\geqslant4$时误差保持不变，因为已有足够的时间延迟嵌入来获得准确的结果。可以看出， $E_{\texttt{CAE-PHODMD}}$ 通常大于 $E_{\texttt{CAE}}$，这是因为 HODMD 和插值会引入额外的误差。

对于 $Re=290.8$，646.0 处的 $t=1.85$ 通过CAE-PHODMD在 $n_{\texttt{latent}}=2$、6 和 $n_{\texttt{delay}}=2$、8 获得的预测全阶解绘制于图4.3中。左图表明，增加$n_{\texttt{delay}}$ 可提高 $n_{\texttt{latent}}=2$的精度；从右图可以观察到 $n_{\texttt{delay}}=2$

编码器		
层	输入形状	输出形状
一维卷积和 SiLU	1×128	32×64
一维卷积和 SiLU	32×64	32×32
一维卷积 + SiLU	32×32	32×16
一维卷积 + SiLU	32×16	32×8
一维卷积 + SiLU	32×8	32×4
一维卷积 + SiLU	32×4	32×2
展平	32×2	64
带 SiLU 的线性层	64	11
带 SiLU 的线性层	11	2

解码器		
层	输入形状	输出形状
带 SiLU 的线性层	2	11
带 SiLU 的线性层	11	64
展平还原	64	32×2
带 SiLU 的一维转置卷积	32×2	32×4
带 SiLU 的一维转置卷积	32×4	32×8
带 SiLU 的一维转置卷积	32×8	32×16
带 SiLU 的一维转置卷积	32×16	32×32
带 SiLU 的一维转置卷积	32×32	32×64
带 SiLU 的一维转置卷积	32×64	1×128

表 4.1：一维 Burgers 方程：网格搜索期间在验证集上取得最佳重构误差的卷积自编码器架构。训练中的权重衰减为 10^{-11}。

足以使结果接近通过卷积自编码器获得的重构全阶解。在这种情况下，$n_{\texttt{latent}} = 6$ 在 $n_{\texttt{delay}} = 2$ 的条件下即可确保测试集中的预测精度。

为考察潜空间中的误差，图4.4 展示了 $E_{\texttt{latent}}$ 通过CAE-PHODMD得到的误差。与图4.2类似，对于 $n_{\texttt{latent}} = 2$，4，误差会随着$n_{\texttt{delay}}$ 增大而减小；当 $n_{\texttt{latent}} = 6$，8，10 时，$n_{\texttt{delay}} \geqslant 4$增大则不会改善误差。因此，此时CAE-PHODMD的预测误差

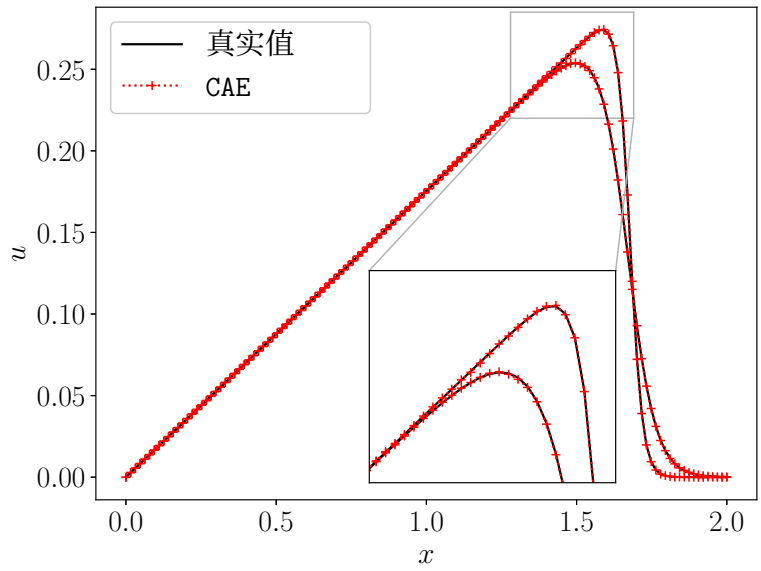


图 4.1：一维 Burgers 方程。CAE 在测试集上针对 $Re = 290.8$，646.0 在$t = 1.85$ 时采用 $n_{\texttt{latent}} = 2$ 得到的重构全阶解。

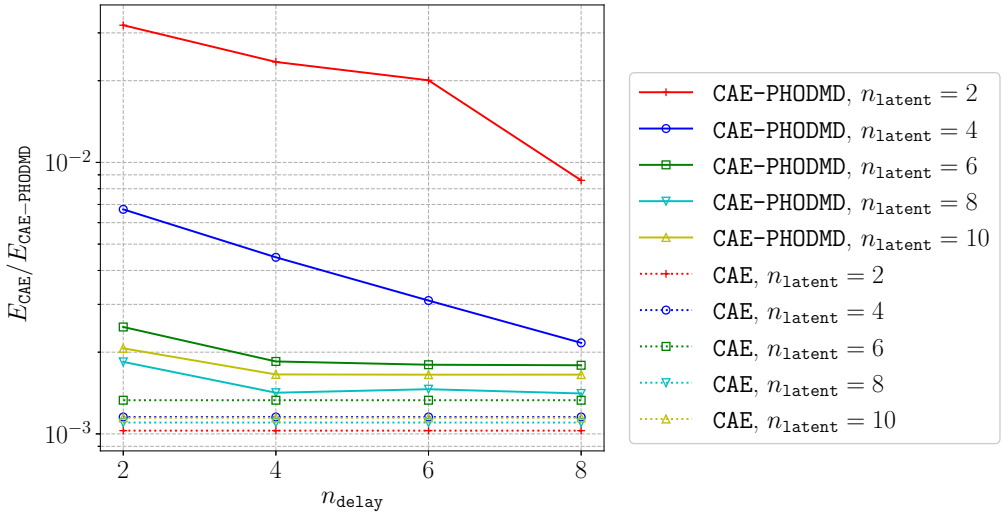
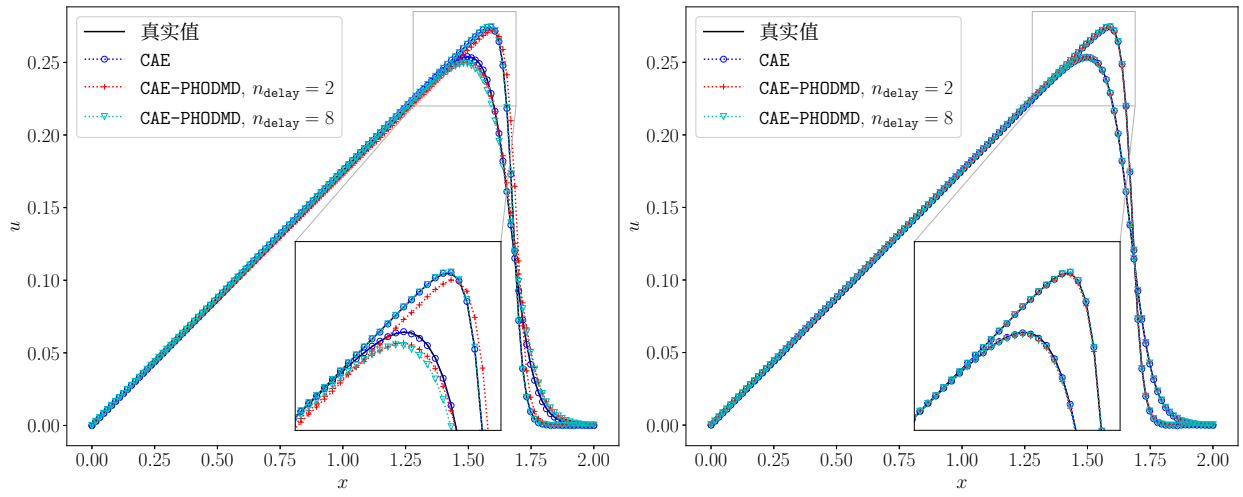


图 4.2：一维 Burgers 方程。CAE 在测试集中使用不同 $n_{\texttt{latent}}$ 得到的重构全阶解误差 $E_{\texttt{CAE}}$，以及 CAE-PHODMD 的预测误差 $E_{\texttt{CAE-PHODMD}}$，与时间延迟嵌入数量 $n_{\texttt{delay}}$ 的关系。

主要由潜空间中的误差主导。图 $n_{\texttt{latent}} = 2$ 针对 $Re = 646.0$ 通过CAE-PHODMD 得到的潜变量时间演化见图4.5。图中还表明，随着时间延迟嵌入的数量$n_{\texttt{delay}}$增加，高阶动态模态分解会更加准确。



(a) $n_{\text{latent}} = 2$ (b) $n_{\text{latent}} = 6$

图 4.3：一维 Burgers 方程。CAE 和 CAE-PHODMD 在 $t = 1.85$ 处针对 $Re = 290.8$、646.0 得到的重构全阶解。

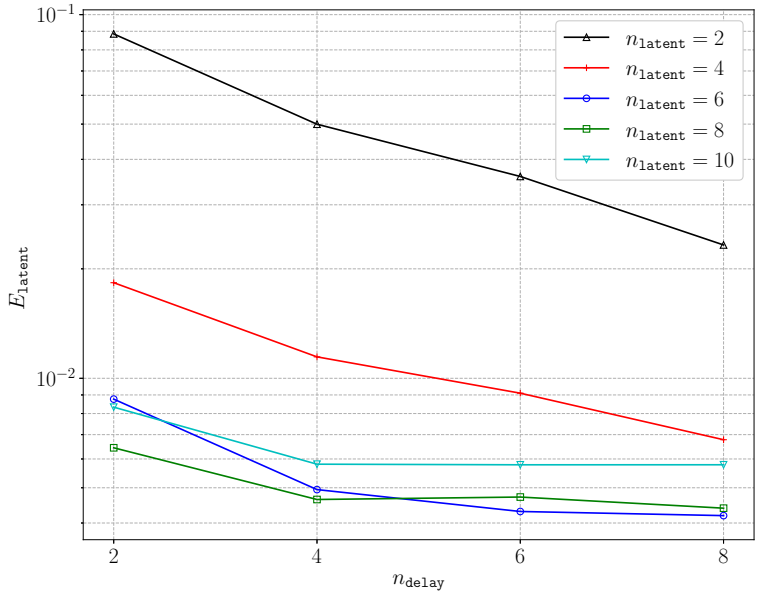
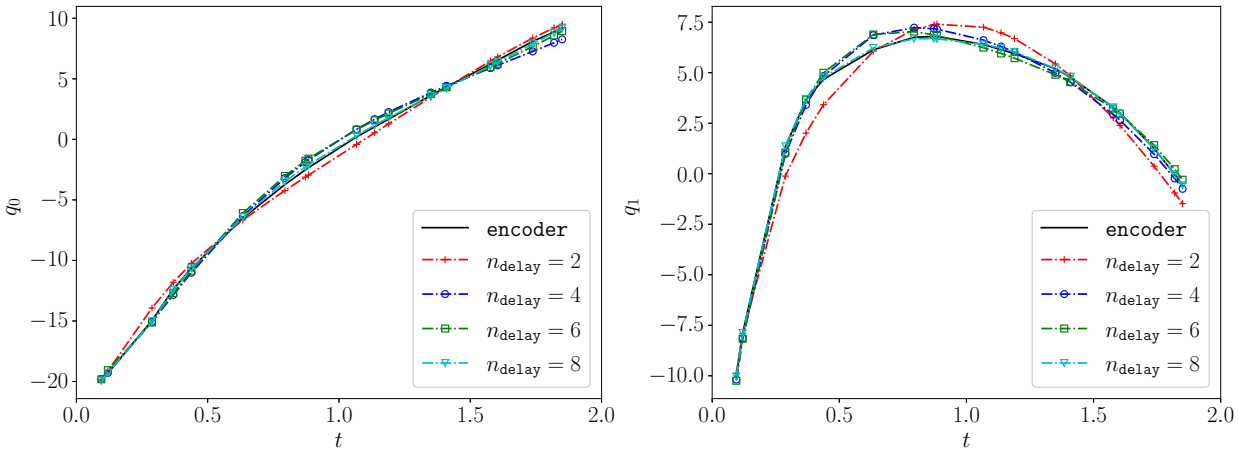


图 4.4：一维 Burgers 方程。CAE-PHODMD 在测试集中潜变量的误差，E_{latent} 相对于时间延迟嵌入的数量 n_{delay}。

4.2. 二维瑞利-贝纳德对流

4.2.1. 设置

本测试将验证我们的方法在二维瑞利-贝纳德对流问题上的有效性和性能。全阶解通过求解二维不可压缩



（a）第一个潜变量。　　（b）第二个潜变量。

图4.5：一维 Burgers 方程。通过 CAE-PHODMD 获得的 $Re = 646.0$ 的潜变量时间演化。

Navier-Stokes方程

$$\begin{cases} \nabla \cdot \boldsymbol{v} = 0, \\ \boldsymbol{v}_t + \boldsymbol{v} \cdot \nabla \boldsymbol{v} = -\nabla p + \sqrt{\dfrac{Pr}{Ra}} \Delta \boldsymbol{v} + T \boldsymbol{e}_y, \\ T_t + \boldsymbol{v} \cdot \nabla T = \dfrac{1}{\sqrt{PrRa}} \Delta T, \end{cases}$$

其中，$\boldsymbol{v}, T, p$ 分别表示无量纲速度、温度和压力。瑞利数 Ra 和普朗特数 Pr 是控制流动的两个无量纲参数，在本测试中Pr 固定为 0.71。计算域为 $[0,2] \times [0,1]$，所有边界上的速度均采用壁面边界条件。下边界和上边界分别设定高温和低温，$T(y=0) = 1, T(y=1) = 0$，温度在左、右边界采用零诺伊曼边界条件。解向量 $\boldsymbol{u} = (\boldsymbol{v}, T)$由速度和温度组成，并采用均匀四边形网格 128×64 上的半隐式格式和 Taylor-Hood 元计算全阶解。所有模拟均将 $Ra = 10^4$ 的稳态作为初始数据。所有卷积层和反卷积层使用的卷积核大小均为 5，步幅和填充均为 2。

本次测试所用的数据集由三部分组成。训练集包含位于 $Ra = 10^6, 2 \times 10^6, 4 \times 10^6, 5 \times 10^6, 7 \times 10^6, 9 \times 10^6$ 的快照，以及在 $[0, 40]$ 中均匀分布的 200 个时刻。

验证集由 $[0, 40]$ 中 $Ra = 3 \times 10^6$、 8×10^6 处的快照以及 20 个随机时间点组成。测试集由 $[0, 48]$ 中 $Ra = 6 \times 10^6$ 处的快照以及 10 个随机时间点组成。

编码器		
层	输入形状	输出形状
Conv2d 与 SiLU	$3 \times 64 \times 128$	$16 \times 32 \times 64$
Conv2d 和 SiLU	$16 \times 32 \times 64$	$16 \times 16 \times 32$
Conv2d 和 SiLU	$16 \times 16 \times 32$	$16 \times 8 \times 16$
Conv2d 和 SiLU	$16 \times 8 \times 16$	$16 \times 4 \times 8$
展平	$16 \times 4 \times 8$	512
带 SiLU 的线性层	512	71
带 SiLU 的线性层	71	10
解码器		
层	输入形状	输出形状
带 SiLU 的线性层	10	71
带 SiLU 的线性层	71	512
展平还原	512	$16 \times 4 \times 8$
带 SiLU 的二维转置卷积	$16 \times 4 \times 8$	$16 \times 8 \times 16$
带 SiLU 的二维转置卷积	$16 \times 8 \times 16$	$16 \times 16 \times 32$
带 SiLU 的二维转置卷积	$16 \times 16 \times 32$	$16 \times 32 \times 64$
带 SiLU 的二维转置卷积	$16 \times 32 \times 64$	$3 \times 64 \times 128$

表 4.2：二维瑞利-贝纳德对流：在网格搜索期间，验证集上具有最佳重构误差的卷积自编码器架构。训练中的权重衰减为 10^{-11} 。

4.2.2. 结果

为确定较优的架构，我们针对不同的权重衰减 10^{-8}、10^{-9}、 10^{-10}、 10^{-11}、卷积层数量 4、5、6以及潜空间维度$n_{\texttt{latent}} = 4$、6、8、10进行网格搜索。选择在验证集上重构误差最小的架构作为最终卷积自编码器模型，如表 4.2所示。

图4.6展示了误差 $E_{\texttt{CAE}}$ 以及 $E_{\texttt{CAE-PHODMD}}$ 在不同 $n_{\texttt{latent}}$ 和$n_{\texttt{delay}}$下的情况。左侧

和右侧图分别显示了所有测试时刻以及时间插值的误差，即 $t \in [0.48]$和 $t \in [0.40]$。可以观察到，左图中的误差通常大于右图中的误差，且误差 $E_{\texttt{CAE-PHODMD}}$ 大于 $E_{\texttt{CAE}}$。可以看出，使用更大的 $n_{\texttt{latent}}$时，几乎所有误差都会减小，并且 $E_{\texttt{CAE-PHODMD}}$ 随着$n_{\texttt{delay}}$ 增大而减小，随后在 $n_{\texttt{delay}} \geqslant 20$ 时，当 $n_{\texttt{latent}} = 8$，10 后保持不变。在左图中， $E_{\texttt{CAE-PHODMD}}$ 的误差随$n_{\texttt{latent}} = 8$ 在 $n_{\texttt{delay}} = 24$ 处增大，这是由外推误差导致的。右图中的误差在排除 $t \in [40, 48]$ 中的样本后保持不变。

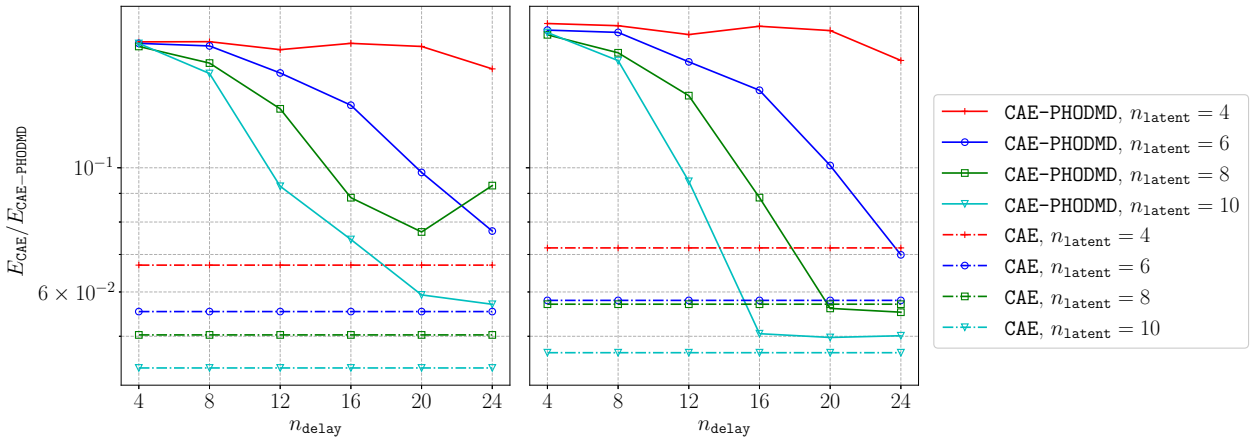
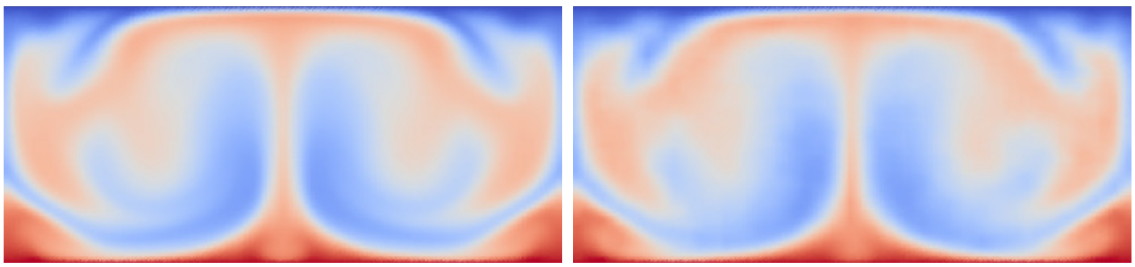
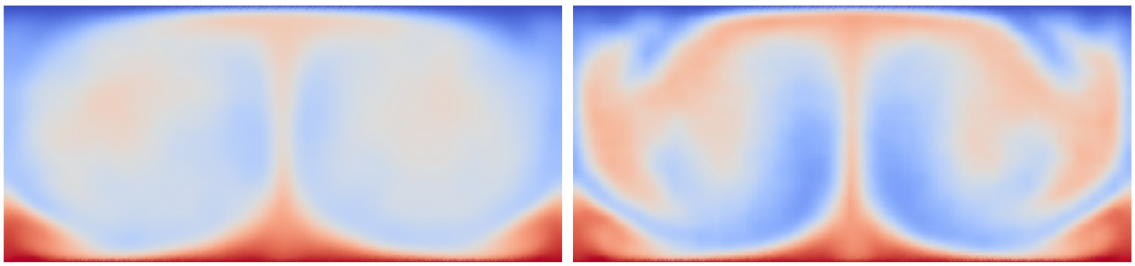


图4.6：二维瑞利-贝纳德对流。不同潜空间维度 $n_{\texttt{latent}}$下，重构误差 $E_{\texttt{CAE}}$ 和预测误差 $E_{\texttt{CAE-PHODMD}}$ 随时间延迟嵌入数量 $n_{\texttt{delay}}$ 的变化。左图对应所有测试时刻 $t \in [0, 48]$，右图仅对应时间插值 $t \in [0, 40]$。

图4.7展示了测试参数值 $Ra = 6 \times 10^6$ 和 $t = 15.809$ 、 $n_{\texttt{latent}} = 10$ 和 $n_{\texttt{delay}} = 4$为 16 时的真实值，以及由卷积自编码器和 CAE-PHODMD 得到的相应解，同时列出了误差。卷积自编码器的重构结果准确，并捕捉到了小尺度特征。对于 $n_{\texttt{latent}} = 10$， $n_{\texttt{delay}} = 4$ 不足以确保预测准确，因为重要模式丢失了。 $n_{\texttt{delay}} = 16$ 对应的结果几乎与卷积自编码器得到的结果一样准确。 $t = 37.139$ 时刻的解如图4.8所示。可以观察到，对于 $n_{\texttt{latent}} = 10$，增加时间延迟嵌入的数量能够降低误差，而 $n_{\texttt{delay}} = 16$ 足以恢复准确结果。在此测试中，我们还通过图4.9检验了卷积自编码器和 CAE-PHODMD 在 $t = 47.656$ 、 $n_{\texttt{latent}} = 10$条件下的外推能力。卷积自编码器的重构稳定，结果准确。 $n_{\texttt{latent}} = 10$ 和$n_{\texttt{delay}} = 8$ 对应的结果准确地捕捉了主要特征，与卷积自编码器得到的结果相当，表明该方法具有良好的预测能力。

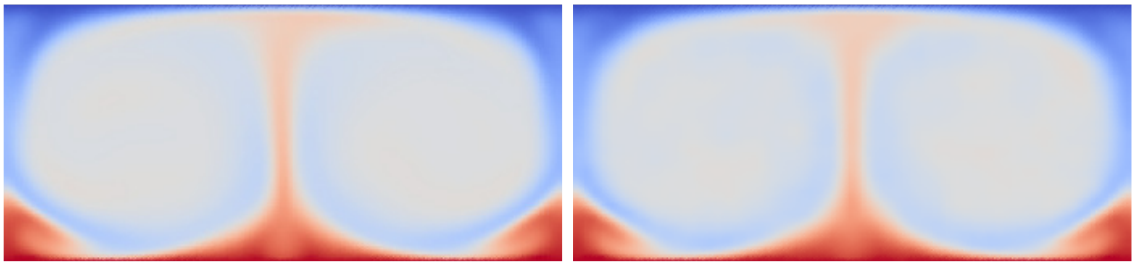


(a) 真实值　　(b) $\epsilon_{\mathtt{CAE}} = 3.779 \times 10^{-2}$

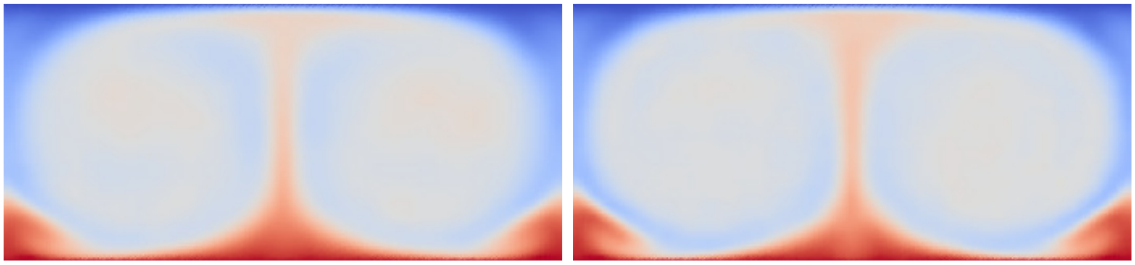


(c) $\epsilon_{\mathtt{CAE-PHODMD}} = 1.930 \times 10^{-1}$, $n_{\mathtt{delay}} = 4$　　(d) $\epsilon_{\mathtt{CAE-PHODMD}} = 4.098 \times 10^{-2}$, $n_{\mathtt{delay}} = 16$

图4.7：二维瑞利-贝纳德对流。 $t = 15.809$、 $Ra = 6 \times 10^6$, 其中 $n_{\mathtt{latent}} = 10$。



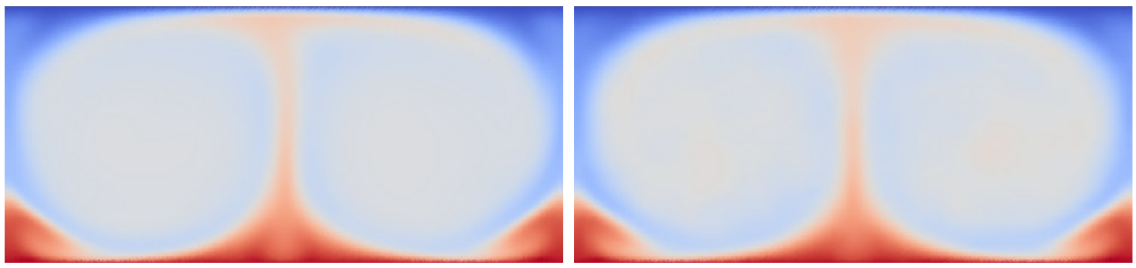
(a) 真实值　　(b) $\epsilon_{\mathtt{CAE}} = 2.764 \times 10^{-2}$



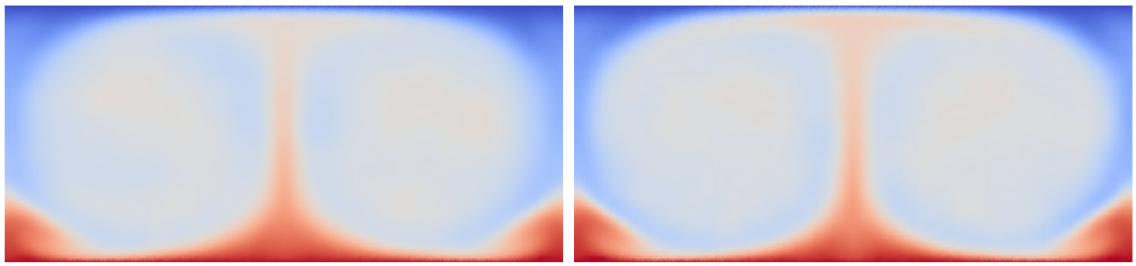
(c) $\epsilon_{\mathtt{CAE-PHODMD}} = 6.319 \times 10^{-2}$, $n_{\mathtt{delay}} = 4$　　(d) $\epsilon_{\mathtt{CAE-PHODMD}} = 2.596 \times 10^{-2}$, $n_{\mathtt{delay}} = 16$

图4.8：二维瑞利-贝纳德对流。 $t = 37.139$、 $Ra = 6 \times 10^6$, 其中 $n_{\mathtt{latent}} = 10$。

图 4.10绘出了误差 $E_{\mathtt{latent}}$ 相对于 $n_{\mathtt{delay}}$ 在不同 $n_{\mathtt{latent}}$下的变化。左图和右图分别对应所有测试时刻和时间插值。预计左图中的误差（包括外推误差）较大，而插值误差会减小



(a) 真实值　　(b) $\epsilon_{\mathtt{CAE}} = 3.304 \times 10^{-2}$



(c) $\epsilon_{\mathtt{CAE-PHODMD}} = 6.534 \times 10^{-2}$, $n_{\mathtt{delay}} = 4$　　(d) $\epsilon_{\mathtt{CAE-PHODMD}} = 3.716 \times 10^{-2}$, $n_{\mathtt{delay}} = 8$

图4.9：二维瑞利-贝纳德对流。 $t = 47.656$， $Ra = 6 \times 10^6$，其中 $n_{\mathtt{latent}} = 10$。

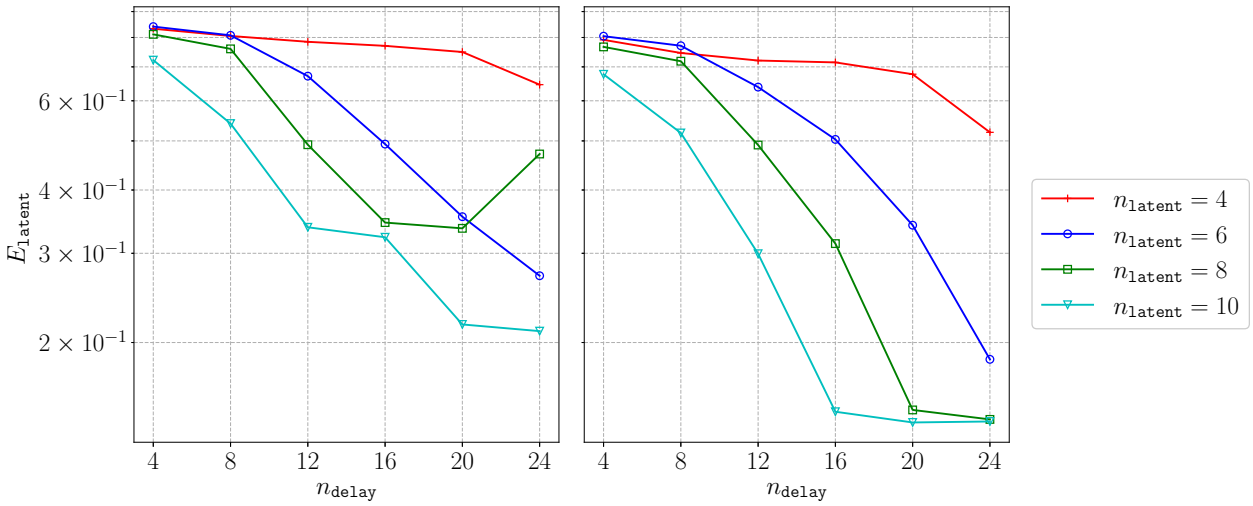


图4.10：二维瑞利-贝纳德对流。对于潜空间的不同维数 $n_{\mathtt{latent}}$，潜空间中的误差 $E_{\mathtt{latent}}$ 与时间延迟嵌入数量 $n_{\mathtt{delay}}$ 的关系。左图对应所有测试时刻，右图对应时间插值。

随着 $n_{\mathtt{delay}}$ 增加。为了进一步了解潜空间中的动力学，图 4.11 展示了 $n_{\mathtt{latent}} = 10$ 在不同 $n_{\mathtt{delay}}$下第一个潜变量的演化。可以看到，在 $t = 40$之前，即插值时，随着 $n_{\mathtt{delay}}$ 增加，参数化高阶动态模态分解的精度提高；而在 $t = 40$之后，误差略大。从总能量的演化中也能观察到这一点

，如图 4.12所示。

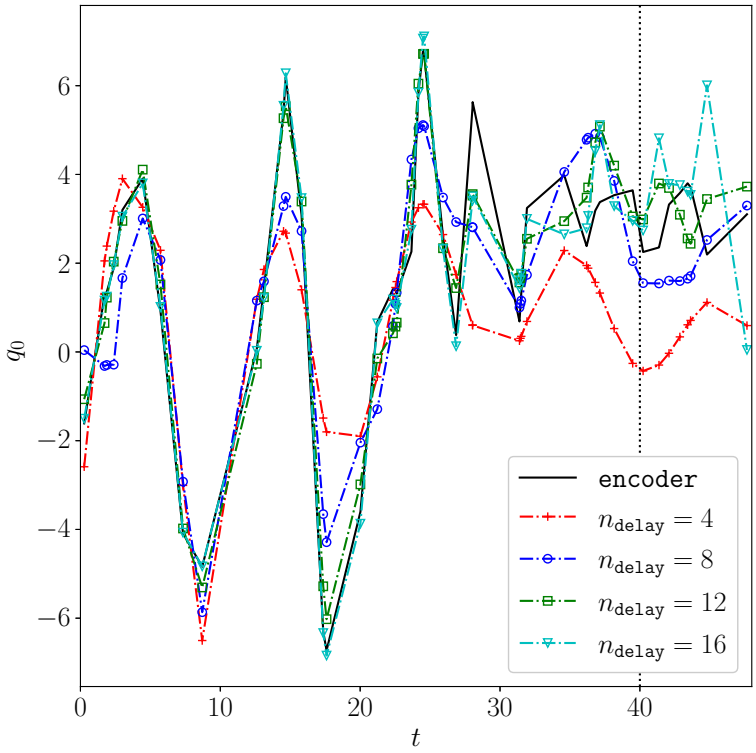


图 4.11：二维瑞利-贝纳德对流。 $Ra = 6 \times 10^6$在 $n_{\text{latent}} = 10$ 和不同 n_{delay}下第一个潜变量的时间演化。

4.3. 二维开尔文-亥姆霍兹不稳定性

4.3.1. 设置

本节考虑二维开尔文-亥姆霍兹不稳定性，这是一种输运主导问题。全阶解通过求解以下可压缩欧拉方程获得

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho v_x\\ \rho v_y\\ E\end{pmatrix}+\frac{\partial}{\partial x}\begin{pmatrix}\rho v_x\\ \rho v_x^2+p\\ \rho v_x v_y\\ (E+p)v_x\end{pmatrix}+\frac{\partial}{\partial y}\begin{pmatrix}\rho v_y\\ \rho v_x v_y\\ \rho v_y^2+p\\ (E+p)v_y\end{pmatrix}=0,$$

其中， ρ， v_x， v_y， E分别表示质量密度、速度和总能量。压力 p可由理想气体状态方程求得

$$p=(E-\frac{1}{2}\rho(v_x^2+v_y^2))/(\Gamma-1),$$

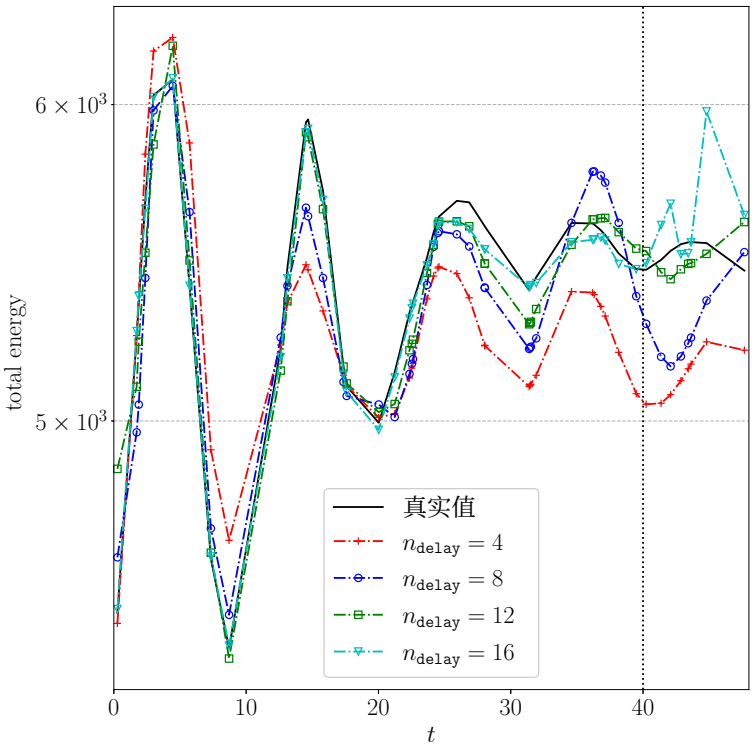


图 4.12：二维瑞利-贝纳德对流。 $Ra = 6 \times 10^6$ with$n_{\texttt{latent}} = 10$ 和不同 $n_{\texttt{delay}}$时总能量的时间演化。

本测试中的绝热指数为 $\Gamma = 1.4$ 。初始数据为

$$
\begin{aligned}
&y_1 = 0.25,\ y_2 = 0.75,\\
&\begin{cases}
\rho = 2,\ v_x = 0.5 + \omega, & \text{if } y_1 < y < y_2,\\
\rho = 1,\ v_x = 0.5 - \omega, & \text{else},
\end{cases}\\
&v_y = 0.1 \sin(4\pi x)(\exp(-0.5(y - y_1)^2)/0.05 + \exp(-0.5(y - y_2)^2)/0.05),\\
&p = 2.5,
\end{aligned}
$$

其中 ω 控制初始间断的剪切速度。采用五阶有限差分加权本质无振荡格式 [20] 在均匀 128×128 网格上求得全阶解。数据集的选取如下。训练集由参数空间内的均匀采样构成，范围从 -0.1 到 0。1，步长为 0。01，但 $\{-0.03, 0.08, -0.04, 0.05\}$除外，其中前两个用于验证，后两个用于测试。训练集包含 100 个在 $[0.01, 1]$ 内均匀分布的时间点，且 $\Delta t = 0.01$；验证集和测试集分别包含 20 个和 10 个随机时间点，位于

$[0.01, 1]$，分别如此。所有卷积层和反卷积层的卷积核大小均为 5，步幅和填充均为 2。

4.3.2. 结果

我们对不同的权重衰减 10^{-8}, 10^{-9}, 10^{-10}, 10^{-11}、卷积层数量 4、5、6，以及潜空间维度 $n_{\texttt{latent}} = 6$、8、10、15 进行网格搜索。最终

CAE架构如表4.3所示。

编码器		
层	输入形状	输出形状
Conv2d 搭配 SiLU	$4 \times 128 \times 128$	$16 \times 64 \times 64$
Conv2d 搭配 SiLU	$16 \times 64 \times 64$	$16 \times 32 \times 32$
Conv2d 搭配 SiLU	$16 \times 32 \times 32$	$16 \times 16 \times 16$
Conv2d 搭配 SiLU	$16 \times 16 \times 16$	$16 \times 8 \times 8$
展平	$16 \times 8 \times 8$	1024
带 SiLU 的线性层	1024	90
带 SiLU 的线性层	90	8
解码器		
层	输入形状	输出形状
带 SiLU 的线性层	8	90
带 SiLU 的线性层	90	1024
展平还原	1024	$16 \times 8 \times 8$
带 SiLU 的二维转置卷积	$16 \times 8 \times 8$	$16 \times 16 \times 16$
带 SiLU 的二维转置卷积	$16 \times 16 \times 16$	$16 \times 32 \times 32$
带 SiLU 的二维转置卷积	$16 \times 32 \times 32$	$16 \times 64 \times 64$
带 SiLU 的二维转置卷积	$16 \times 64 \times 64$	$4 \times 128 \times 128$

表 4.3：二维开尔文-亥姆霍兹不稳定性：网格搜索期间，在验证集上重构误差最小的卷积自编码器架构。训练时的权重衰减为 10^{-8} 。

误差 $E_{\texttt{CAE}}$ 和 $E_{\texttt{CAE-PHODMD}}$ 在不同 $n_{\texttt{latent}}$ 和$n_{\texttt{delay}}$ 下的结果见图4.13。左图考虑了测试集中的所有采样，右图则排除了一个末尾时刻

之外的 $[0, 0.9]$。可以看出，CAE-PHODMD 在最后一个测试时刻之前的结果是准确的。对于给定的 $n_{\texttt{latent}}$，右图中的误差 $E_{\texttt{CAE-PHODMD}}$ 随着 $n_{\texttt{delay}}$ 增大而减小，随后保持不变，且与 CAE 得到的重构误差相当。可以得出结论，使用 $n_{\texttt{latent}} = 8$ 和 $n_{\texttt{delay}} = 10$ 能够恢复准确的结果。

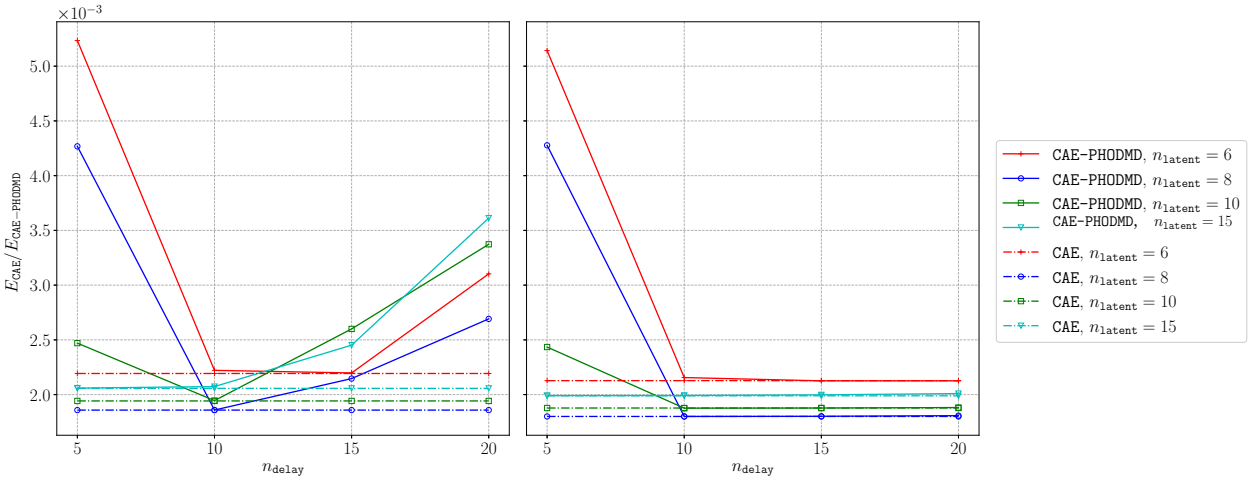
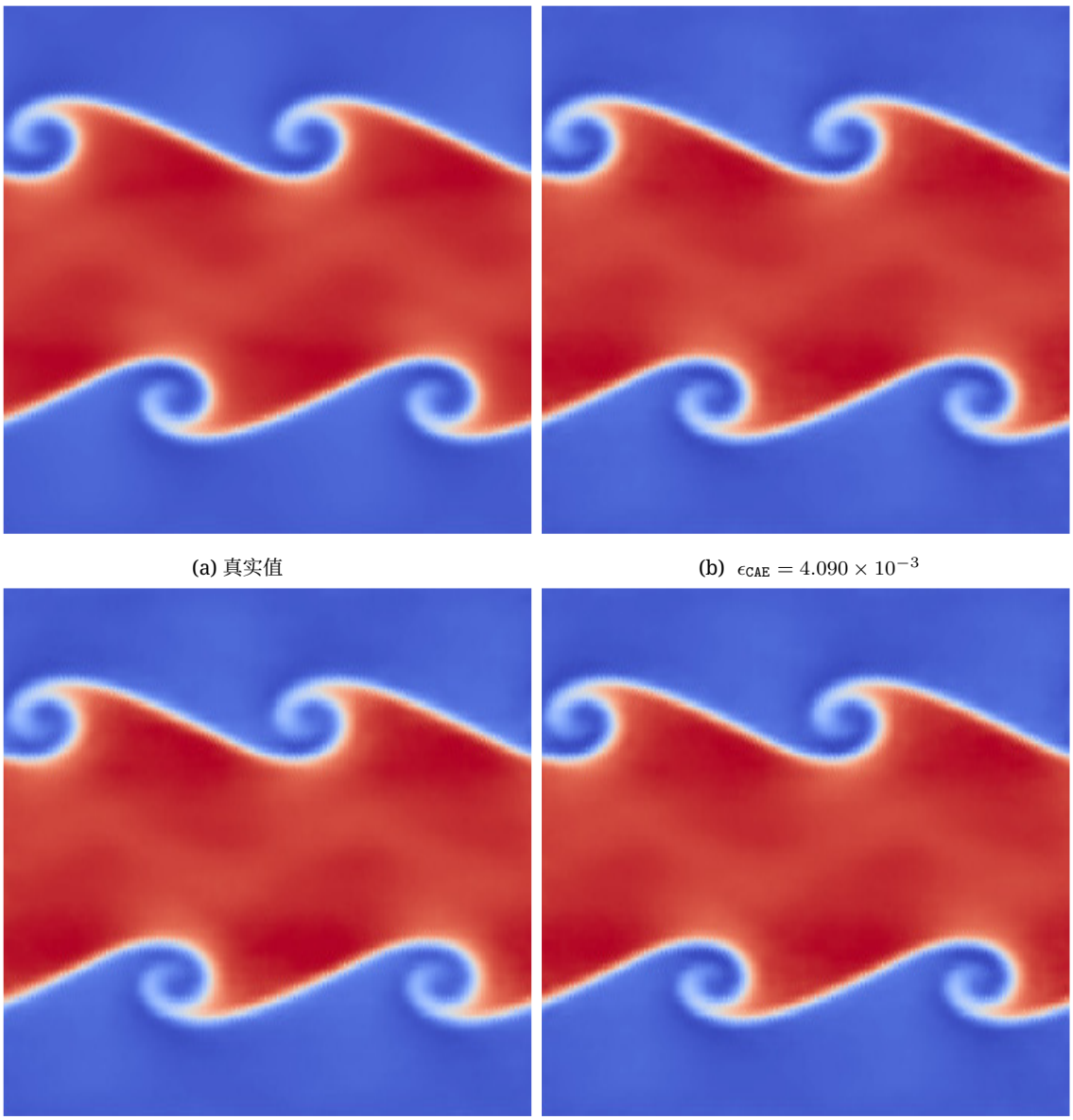


图 4.13：二维开尔文-亥姆霍兹不稳定性。不同潜空间维度 $n_{\texttt{latent}}$ 下，重构误差 $E_{\texttt{CAE}}$ 和预测误差 $E_{\texttt{CAE-PHODMD}}$ 相对于时间延迟嵌入数量 $n_{\texttt{delay}}$ 的变化。左图对应所有测试时刻，右图则排除了最后一个时刻。

真实值以及通过卷积自编码器和 CAE-PHODMD 针对特定测试参数值 $w = 0.05$ 和时刻 $t = 0.568$ （其中 $n_{\texttt{latent}} = 8$ 和 $n_{\texttt{delay}} = 5$，10）如图 4.14 所示，并给出了相应的误差。卷积自编码器的重构结果较为准确，能够在正确位置捕捉到涡卷。对于 $n_{\texttt{latent}} = 8$ 在本例中，使用 $n_{\texttt{delay}} = 5$ 和$n_{\texttt{delay}} = 10$ 均能捕捉主要特征，后者的精度更高；这是预期之中的，因为高阶动态模态分解基于更大的滑动窗口使用了更多信息。在 $t = 0.933$时刻的结果绘制于图 4.15中。此时，图样更加复杂，而 CAE-PHODMD 以 $n_{\texttt{latent}} = 8$ 和 $n_{\texttt{delay}} = 10$恢复出了准确的结果，其效果与卷积自编码器得到的重构结果相当，表明该方法能够捕捉这一输运主导问题的动力学。

为考察潜空间中的动力学，图 4.16 绘制了误差 $E_{\texttt{latent}}$ 相对于 $n_{\texttt{delay}}$ 在不同 $n_{\texttt{latent}}$下的变化。左图对应所有测试时刻的误差，右图则排除了一个末尾时刻。与图 4.13类似，排除一个末尾时刻后的误差在$n_{\texttt{delay}}$ 较大时几乎保持不变。图 4.17 展示了不同 $n_{\texttt{delay}}$ 下，第一个潜变量随 $n_{\texttt{latent}} = 8$的演变。可以观察到，我们能够重构出精确的



(a) 真实值

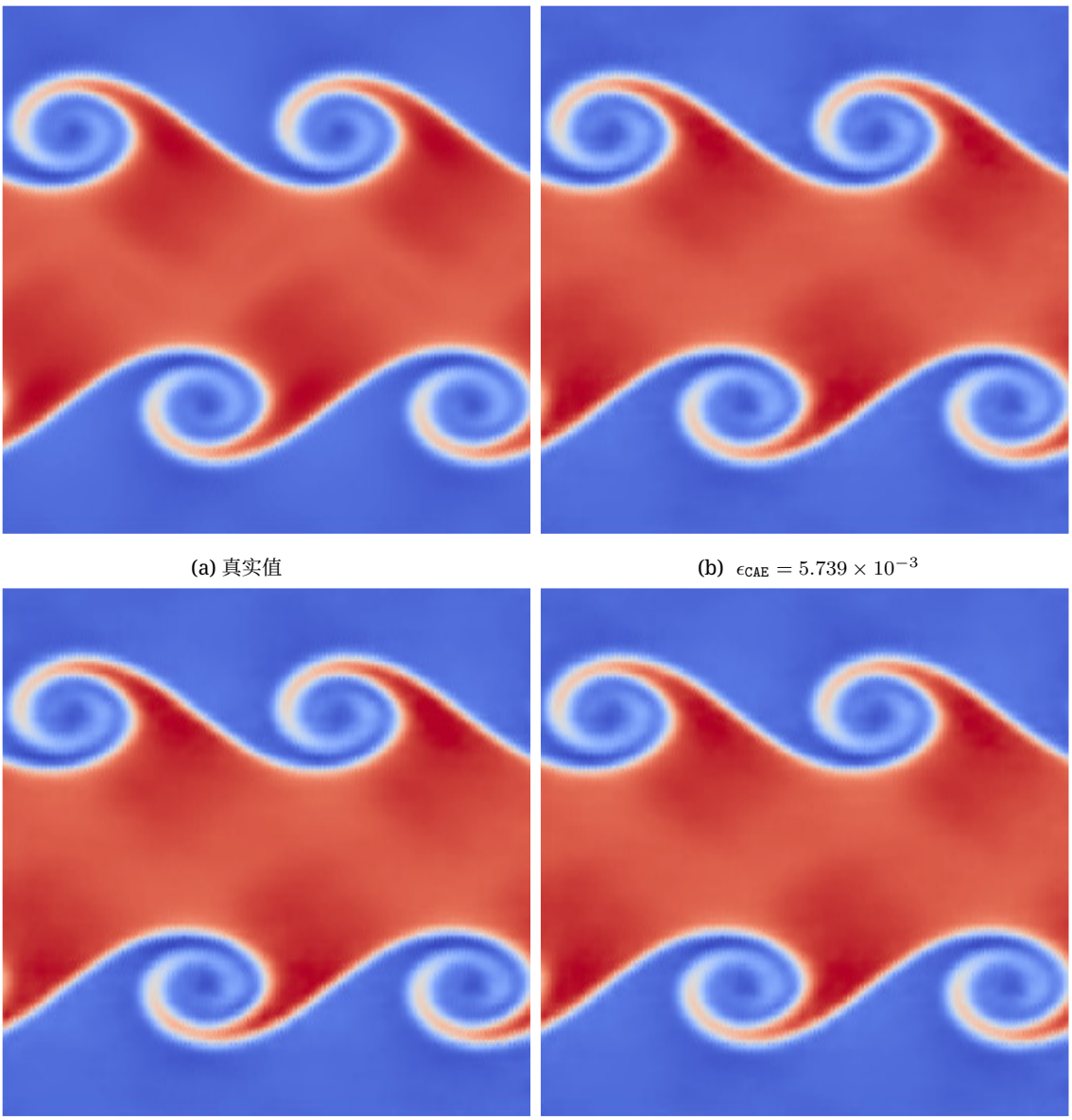
(b) $\epsilon_{\mathtt{CAE}} = 4.090 \times 10^{-3}$

(c) $\epsilon_{\mathtt{CAE-PHODMD}} = 9.921 \times 10^{-3}$, $n_{\mathtt{dela\ \ y}} = 5$

(d) $\epsilon_{\mathtt{CAE-PHODMD}} = 4.076 \times 10^{-3}$, $n_{\mathtt{dela\ \ y}} = 10$

图 4.14：二维开尔文-亥姆霍兹不稳定性。 $t = 0.568$, $\omega = 0.05$，其中 $n_{\mathtt{latent}} = 8$。

潜空间中的动力学。然而，在结束时刻，随着$n_{\mathtt{delay}}$ 增加，参数化高阶动态模态分解的准确性往往会降低，这是由高阶动态模态分解的公式所致。



(a) 真实值 (b) $\epsilon_{\mathtt{CAE}} = 5.739 \times 10^{-3}$

(c) $\epsilon_{\mathtt{CAE-PHODMD}} = 8.468 \times 10^{-3}$, $n_{\mathtt{dela\ \ y}} = 5$ (d) $\epsilon_{\mathtt{CAE-PHODMD}} = 5.747 \times 10^{-3}$, $n_{\mathtt{dela\ \ y}} = 10$

图4.15：二维开尔文-亥姆霍兹不稳定性。 $t = 0.933$, $\omega = 0.05$, 其中 $n_{\mathtt{latent}} = 8$。

5. 结论

本文提出并研究了一种针对时变参数化问题的新型非侵入式降阶建模方法。我们的方法完全由数据驱动，并包含离线和在线阶段。大部分计算成本集中在离线阶段，从而能够在在线阶段针对新的时间点和参数值进行快速且准确的预测。在

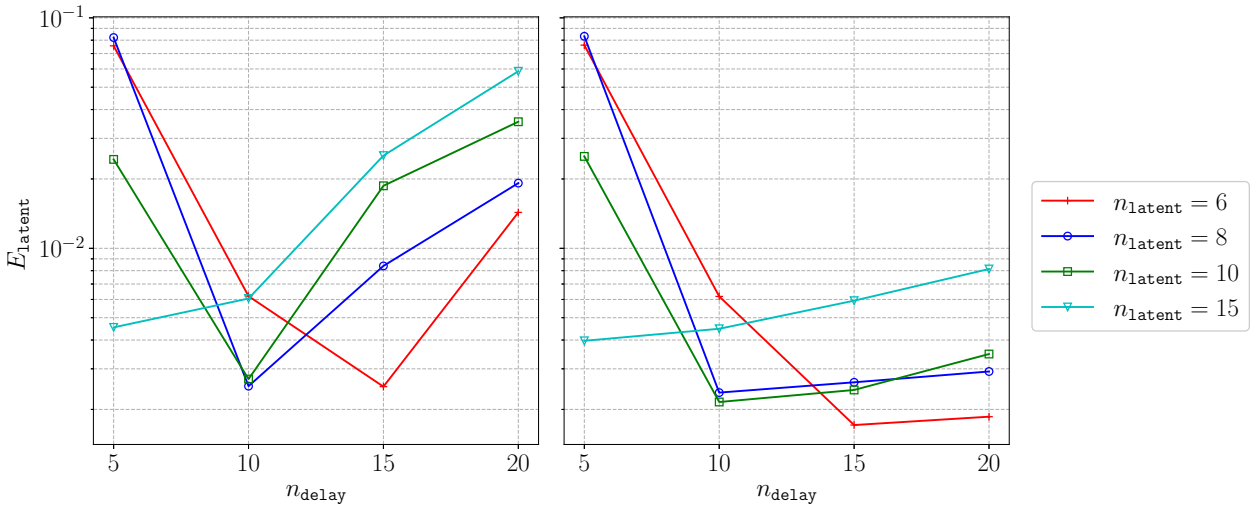


图 4.16：二维开尔文-亥姆霍兹不稳定性。潜空间中的误差 $E_{\texttt{latent}}$ 相对于时间延迟嵌入数量 $n_{\texttt{delay}}$ 的变化，针对不同的潜空间维数 $n_{\texttt{latent}}$。左图显示所有测试时刻的结果，右图则排除了最后一个时刻。

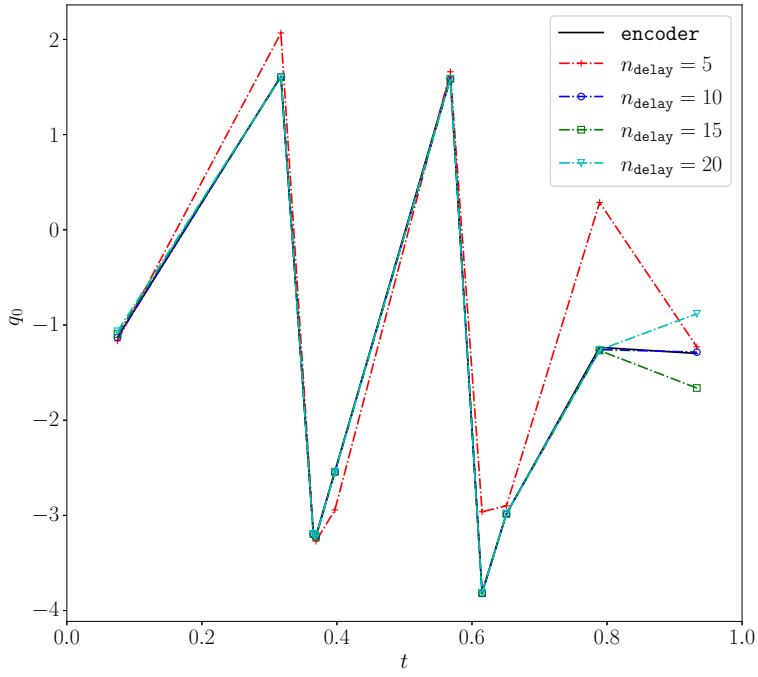


图 4.17：二维开尔文-亥姆霍兹不稳定性。针对 $n_{\texttt{latent}} = 8$，第一个潜变量在不同 $n_{\texttt{delay}}$下的时间演化。

离线阶段，首先训练深度卷积自编码器。编码器将高维全阶解降维为低维潜变量，而解码器部分可以

从潜空间恢复全阶解。随后，针对每个参数值，使用高阶动态模态分解对潜空间中的动力学进行建模。在在线阶段，通过高阶动态模态分解获得新时刻的潜变量，利用插值计算新参数值对应的潜变量，再由解码器从潜变量恢复全阶解。我们针对一维 Burgers 方程、二维瑞利-贝纳德对流和二维开尔文-亥姆霍兹不稳定性进行了数值测试，结果表明，所提出的方法能够准确预测新时刻和新参数值下的全阶解，也适用于输运主导问题。未来，我们将考虑改进用于潜在动力学的高阶动态模态分解方法，并进一步探索多维参数空间以及更高效的训练集采样方法。

参考文献

[1] F. Andreuzzi, N. Demo 和 G. Rozza，《一种用于预测参数化动力系统的动态模态分解扩展》，*arXiv: 2110.09155*，（2021）。[2] P. Astrid, S. Weiland, K. Willcox 和 T. Backx，《采用本征正交分解描述的模型中的缺失点估计》，*IEEE* 自动控制汇刊，53（2008），2237–2251。[3] M. Barrault, Y. Maday, N.C. Nguyen 和 A.T. Patera，《一种"经验插值"方法：在偏微分方程的高效约化基离散中的应用》，《数学报告》，339（2004），667–672。[4] P. Benner, S. Gugercin 和 K. Willcox，《参数化动力系统基于投影的模型降阶方法综述》，*SIAM Rev.*，57（2015），483–531。[5] S.L. Brunton, M. Budišić, E. Kaiser 和 J.N. Kutz，《动力系统的现代 Koopman 理论》，*SIAMRev.*，64（2022），229–340。[6] K. Carlberg, C. Farhat, J. Cortial 和 D. Amsallem，《非线性模型降阶的 GNAT 方法：高效实现及其在计算流体动力学和湍流中的应用》，计算物理学杂志，242（2013），623–647。[7] K.T. Carlberg，《降阶模型的自适应 h-细化》，*Int. J. Numer. Methods Eng.*，102（2015），1192–1210。[8] S. Chaturantabut 和 D.C. Sorensen，《通过离散经验插值实现非线性模型降阶》，*SIAM J.Sci. Comput.*，32（2010），2737–2764。[9] P. Conti, G. Gobat, S. Fresca, A. Manzoni 和 A. Frangi，《通过自编码器和 SINDy方法对参数化系统进行降阶建模：周期解的延拓》，*arXiv: 2211.06786*，（2022）。[10] C. Eckart 和 G. Young，《用另一个低秩矩阵逼近一个矩阵》，《心理计量学》，1（1936），211–218。[11] V. Ehrlacher, D. Lombardi, O. Mula 和 F.X. Vialard，《度量空间上的非线性模型降阶：在 Wasserstein空间中一维守恒偏微分方程的应用

》，*ESAIM*：数学建模与数值分析，54（2020），2159–2197。[12] S. Fresca, L. Dede' 和 A. Manzoni，《一种全面的、基于深度学习的非线性时变参数化偏微分方程降阶建模方法》，*J. Sci. Comput.*，87（2021），1–36。[13] S. Fresca 和 A. Manzoni，《POD-DL-ROM：通过本征正交分解增强基于深度学习的非线性参数化偏微分方程降阶模型》，*Comput. Methods Appl. Mech. Eng.*，388（2022），114181。[14] M. Guo 和 J.S. Hesthaven，《利用高斯过程回归进行非线性结构分析的降阶建模》，*Comput. Methods Appl. Mech. Eng.*，341（2018），807–826。[15] M.W. Hess, A. Quaini 和 G. Rozza，《一种结合动态模态分解与流形插值、用于时变不可压缩纳维-斯托克斯方程的数据驱动代理建模方法》，*arXiv:2201.10872*，（2022）。[16] J.S. Hesthaven, C. Pagliantini 和 G. Rozza，《时变问题的约化基方法》，*Acta Nu-mer.*，（2022），265–345。[17] J.S. Hesthaven, G. Rozza 和 B. Stamm，参数化偏微分方程认证降阶基方法，SpringerBriefs in Mathematics，Springer，Cham，2016。[18] J.S. Hesthaven 和 S. Ubbiali，《利用神经网络对非线性问题进行非侵入式降阶建模》，*J. Comput. Phys.*，363（2018），55–78。[19] Q.A. Huhn, M.E. Tano, J.C. Ragusa 和 Y. Choi，《用于降阶建模的参数化动态模态分解》，*arXiv: 2204.12006*，（2022）。[20] G.S. Jiang 和 C.W. Shu，《加权ENO格式的高效实现》，*J. Comput. Phys.*，126（1996），202–228。[21] M.R. Jovanović, P.J. Schmid 和 J.W. Nichols，《稀疏性促进动态模态分解》，*Phys. Fluids*，26（2014）。[22] A. Kolmogoroff，《关于给定函数类中函数的最佳逼近》，*Ann. of Math.*，37（1936），107–110。[23] S. Le Clainche 和 J.M. Vega，《高阶动态模态分解》，*SIAM J. Appl. Dyn. Syst.*，16（2017），882–925。[24] K. Lee 和 K.T. Carlberg，《利用深度卷积自编码器在非线性流形上对动力系统进行模型降阶》，*J. Comput. Phys.*，404（2020），108973。[25] R. Maulik, B. Lusch 和 P. Balaprakash，《利用循环神经网络和卷积自编码器对平流主导系统进行降阶建模》，*Phys. Fluids*，33（2021）。[26] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai 和 S. Chintala，《PyTorch：一种命令式风格的高性能深度学习库》，载于神经信息处理系统进展 *32*，第8024–8035页，Curran Associates, Inc.，2019。[27] B. Peherstorfer，《通过在线自适应基和自适应采样对输运主导问题进行模型降阶》，*SIAM J. Sci. Comput.*，42（2020），A2803–A2836。

[28] B. Peherstorfer 和 K. Willcox，通过低秩更新对非线性系统进行在线自适应模型降阶，*SIAM J. Sci. Comput.*，37（2015），A2123–A2150。[29] B. Peherstorfer 和 K. Willcox，面向非侵入式基于投影的模型降阶的数据驱动算子推断，*Comput. Methods Appl. Mech. Eng.*，306（2016），196–215。[30] M.J.D. Powell，多变量插值的径向基函数：综述，载于 逼近算法（*Shrivenham，1985*），*Inst. Math. Appl. Conf. Ser. New Ser.* 第10卷，第143–167页，牛津大学出版社，纽约，1987。[31] A. Quarteroni、A. Manzoni 和 F. Negri，偏微分方程的约化基方法，Springer，Cham，2016。[32] D. Ryckelynck，涉及内部变量的力学模型超约化，国际工程数值方法杂志，77（2009），75–89。[33] P.J. Schmid，数值数据和实验数据的动态模态分解，流体力学杂志，656（2010），5–28。[34] E. Schmidt，关于线性和非线性积分方程的理论。第一部分：按照给定系统展开任意函数，数学年刊，63（1907），433–476。[35] R. Zimmermann，流形插值，载于 系统驱动与数据驱动的方法和算法，第1卷，第229–274页，2021。