

# 基于深度学习的地下流动降阶建模 模拟

赵阳拉里·伊敏·刘<sup>a</sup>·路易·J·杜洛夫斯基<sup>a</sup>

<sup>a</sup>斯坦福大学能源资源工程系, 加利福尼亚州斯坦福, 94305

## 摘要

提出了一种新的基于深度学习的降阶建模(ROM)框架, 用于地下流动模拟。降阶模型是基于现有的嵌入控制(E2C)框架, 包括一个自动编码器, 将系统投影到一个低维子空间, 以及一个线性过渡模型, 它近似于低维系统状态的演化。除了在原始 E2C 框架中考虑的数据失配的损失函数外, 我们还引入了一个基于物理的损失函数, 该函数惩罚与控制流方程不一致的预测。损失函数也作了修改, 以强调关键井量的准确性(例如, 流体产量)。事实表明, E2CROM 非常类似于现有的 ROM, POD-TPWL, 它已被广泛开发用于地下流动模拟。新的 ROM 应用于非均质油藏的油水流动, 由 9 口井在时变控制规范下运行驱动。在离线阶段总共进行了 300 次高保真训练模拟, 网络训练需要在特斯拉 V100GPU 节点上 10-12 分钟。在线(运行时)计算相对于全阶模拟实现了  $O(1000)$  的加速。给出了广泛的测试用例结果, 在大范围内有很好的控制。准确的 ROM 预测是实现全球饱和和压力场在特定的时间, 以及注入和生产良好的反应作为时间的函数。当使用 100 或 200 (而不是 300) 训练运行来构造 E2CROM 时, 误差会增加。

关键词: 储层模拟, 降阶模型, 深度学习, 物理信息神经网络, 自动编码器, 嵌入控制, E2C

## 1. 引言

储层模拟被广泛应用于地下水流的建模和管理。然而, 由于控制方程的非线性性质和地质描述的多尺度特性, 计算成本可能很高, 特别是当高度时

---

\*相应的作者

电子邮件地址: zjin@stanford.edu (昭阳拉里金), yiminliu@stanford.edu (刘益民), lou@stanford.edu(路易斯 J.杜洛夫斯基)

预印提交给计算物理学杂志  
日.

2019 年 6 月 11

使用解析模型。当应用模拟工具进行优化、不确定性量化和数据同化时，计算需求可能变得令人望而却步，在这种情况下，可能需要数千次模拟运行。

降阶模型(ROM)已被开发并应用于在各种设置中加速流量预测。我们在这项工作中的目标是开发一个新的基于深度学习的降阶建模过程。遵循嵌入到控制框架，这里介绍的方法包括线性过渡模型和自动编码器(AE, 也称为编解码器)。采用编解码器结构，通过构造低维表示的映射来实现降维。声发射分量是多个卷积神经网络(CNN)层和密集前馈层的叠加。线性过渡模型表示具有多个线性前馈层的系统状态的逐步演化。构造 E2C 过程来预测油水储层模拟问题中的关键井量，如时变生产和注入速率和/或井底压力(BHPs)，以及全球压力和饱和场。

ROM 方法近年来受到了广泛的关注。这些过程通常涉及离线（训练时间）组件，其中执行训练运行并处理和保存相关的解决方案信息，以及在线（测试时间）组件，其中执行新的（测试）运行。一种流行的方法是基于正交解分(POD)ROM，其中 POD 被应用于在线计算中解未知数的低维表示。这些方法还要求将方程组投影到低维（这种投影也称为约束约简）。Galerk in 投影和最小二乘 Petrov-Galerk in 投影是这一步通常使用的两种方法。

还需要对解非线性进行处理，并且在基于 POD 的 ROM 中已经有了一些处理。一种有效的方法是 Gauss-Newton，具有近似张量或 GNAT，它还使用 POD 进行状态约简和最小二乘 Petrov-Galerkin 投影。GNAT 是由 Carlberg 等人开发的。[1][2][3][4][5][6 7 8 9 10 11] 后来被用于结构和固体力学、机电学和计算流体力学。GNAT 是离散经验插值方法(DEIM)的推广，这两种方法(GNAT 和 POD-DEIM)已被应用于许多涉及地下流动模拟的研究中。径向基函数(RBF)多维插值方法也被用来处理 POD 表示的低维空间中的非线性，得到的过程被称为 POD-RBF 方法。轨迹分段线性化，最初由 Rewienski 和 White 引入，涉及围绕“附近”训练解的线性化。POD-TPWL 已广泛应用于地下流动模拟，包括油水、油气组成、CO<sub>2</sub> 储存和耦合流动地质力学系统。特雷汉和杜洛夫斯基扩展了 POD-TPWL，包括一个二次项，它给出了一个轨迹分段二次(POD-TPWQ)过程。[12 13][14][15 16 17 18 19 20][21]

深度学习在图像处理中最近的成功激发了利用深层神经网络进行地下建模的算法的快速发展。这些方法已应用于地质参数化、不确定性量化和代理/降阶建模。对于地质参数化和不确定性量化，Canchumuni 等人。[22] 利用变分自动编码器(VAE)从随机低维潜变量中生成新的地质实现。一个 VAE 需要一个类似于 AE 的卷积编解码器神经网络结构，其中编码器组件将高维分布投影到一个低维随机向量中，每个元素遵循独立的高斯分布。解码器充当编码器的逆，并将采样的高斯分布随机变量投影回高维。Laloy 等人。[23] 使用生成对抗性网络(GAN)实现了类似的目标，其中高维投影是通过训练两个对抗性神经网络（称为生成器和鉴别器）来确定的。Liu 等人[24][25][26][27] 刘和杜洛夫斯基将基于主成分分析(PCA)的表示扩展到 CNN-PCA 过程中。该方法应用“快速神经风格转移”算法来表示以多点空间统计为特征的复杂地质模型，并证明了能够更有效地进行数据同化。朱和扎巴拉斯将代理建模作为图像对图像的回归，并构建了用于地质不确定性量化的贝叶斯深卷积神经网络。随后，Mo 等人。[28] 扩展了该模型以处理多相流问题，并通过引入额外的物理约束进一步提高了性能。

[29][30] 最近在 ROM 中使用深度学习技术的发展表明这种方法有很大的潜力。李和卡尔伯格介绍了一种改进的 GNAT 程序，用 AE 代替 POD。该方法应用于一维动态汉堡方程和二维准静态化学反应流动问题，测试中的边界条件与训练运行中的边界条件不同。卡尼和埃尔谢克开发了一种深

度残差递归神经网络(D R-RN N)过程, 该过程采用 RNN 逼近 POD-DEIM 过程中控制方程的低维残差函数。然后将得到的 ROM 应用于一个一维油水问题, 该问题的孔隙度分布在测试运行中受到训练运行的扰动。张等人[31]使用完全连接的网络来替换 POD-DEIM 过程中的牛顿迭代.. 该方法用于预测二维油水问题中的井响应, 其中试验运行的井控和渗透率场的组合不同于训练模拟。虽然上述所有方法相对于“标准”实现都实现了精度的提高, 但所有这些开发都在现有的 ROM 设置中; 也就是说, 没有一个采用端到端深度学习框架。

[32][33][34][35] 其他研究人员开发了 ROM 方法, 这更多地代表了偏离现有方法。Wang 等人例如, 使用长短期记忆(LST M)RNN 来近似 POD 构造的低维子空间中的流动动力学。随后, Gonzalez 和 Balajewicz 用 VAE 代替 POD 步骤进行低维表示.. 然而, 这两种方法都适用于相对简单的问题, 其中在线和离线模拟运行之间的唯一差异是系统的初始条件(边界条件相同)。在地下流动方程中, 井以局部源/汇项的形式出现, 基本上充当“内部”边界条件。在离线和在线计算之间改变井设置的能力是用于石油生产优化和相关领域的 ROM 的基本功能。因此, 上述实现可能不直接适用于这些问题。另一个潜在的限制是, 这些程序纯粹是数据驱动的, 不考虑基本的控制方程。这可能导致解决方案的视觉吸引力, 但身体上不现实。

[36] 许多方法被应用于将物理约束纳入深层神经网络。这些程序有不同的名称, 但往往有相同的关键思想。Raissi 等人。介绍了一种物理知情的深度学习框架(后来称为物理知情神经网络或 PINN), 该框架使用密集连接的前馈神经网络。在 PINN 中, 将与控制偏微分方程(PDEs)相关的残差函数引入到神经网络的损失函数中。朱等。[37][38]将 PDE 约束概念扩展到基于深度流的生成模型(GLOW), 并利用控制方程的残差而不是模拟输出构建了不确定性量化的代理模型。Watter 等人。[39] 在机器人规划系统的背景下, 提出了一个嵌入控制(E2C)框架, 以直接感觉数据(图像)和时变控制作为输入来预测系统状态的演化。该 E2C 框架结合了一个 VAE, 它既用作将系统状态投影到低维子空间的推理模型, 也用作生成模型, 以全阶重构预测结果, 并具有线性过渡模型。后者基于时变控制输入近似于低维态的演化..

[39][36 37] 在本文中, 我们开发了一个基于 E2C 模型和上述物理知情处理的地下流动系统降阶建模的深度学习框架。介绍了现有 E2C 模型的两个关键修改。具体来说, 我们将 VAE 简化为 AE, 以获得更好的确定性测试用例的准确性, 并且我们包含了一个全面的损失函数, 该函数引入了基于 PDE 的物理约束, 并在井数据中强制一致性。后一种处理方法对于提高感兴趣的井量的精度具有重要意义, 这是石油生产优化程序中必不可少的。由于我们正在考虑一个有标记数据(输入和输出对)的监督学习问题, 我们引入物理约束的方式将我们的模型与 Raissi 等人的模型区分开来。[36] 朱等人。[37]其中, 需要访问 PDE 残差来计算训练过程中的损失函数。我们的治疗可能更适合在许多实际环境中, 其中潜在的 PDE 的剩余值可能无法访问。例如, 如果使用商业模拟器, 就可能是这样。有趣的是, 我们的 E2C 过程与现有的 POD-TPWL 方法学相当相似, 我们详细讨论了这两种方法之间的关系。

2 3 4 本文进行如下。在本节中, 我们给出了地下油水流动的控制方程, 然后简要描述了 POD-TPWLROM.. 在本节中, 给出了 E2C 公式, 并强调了 E2C 与 POD-TPWL 之间的对应关系。我们给出了一个二维油水问题的结果。测试用例涉及不同时变井设置的规范, 就像在优化问题中遇到的那样。我们还提出了几个关键数量的详细误差评估。最后, 我们对今后的工作提出了总结和建

议。提供了 E2C 模型中使用的编码器和解码器的详细架构 5  
附录 A。给出了基于深度学习的 ROM 的附加仿真结果。附录 B

## 2. 控制方程和 POD-TPWLROM

在这一部分中，我们给出了油水流动方程.. 然后，我们提供了一个 POD-TPW LROM 的概述，这将允许我们绘制与 E2CROM 的类比。

### 2.1. 管理方程式

不混溶油水流动的控制方程来源于各组分的质量守恒，并结合达西对各相的定律。由此得到的方程，忽略了毛细管压力效应

$$\frac{d}{dt} \sum_w (P_j) (\sum_j P_j k V_p) + E P_j q_w = 0, \quad (1)$$

其中下标  $j(j=0, w)$  表示油和水)表示流体相。地质表征用  $E_q$  表示。1 通过孔隙度  $\phi$  和渗透率张量  $k$ ，岩石和流体之间的相互作用由相迁移率  $A_j$  指定，其中  $X_j = K_{RJ}/\mu_j$ ， $K_{RJ}$  相  $j$  的相对渗透率和“顶相  $j$  的粘度。其他变量是压力  $p$  和相位饱和  $S_j$ （这些是主要的解变量）、时间  $t$  和相位密度  $P_j$ 。问题  $w$  项表示井  $w$  的相源/汇项。该油水模型是通过强制饱和约束  $S_o + S_w = 1$  来完成的。由于本工作中考虑的系统是水平的(在  $x$ - $y$  平面上)，所以忽略了重力效应。

采用标准有限体积公式对油水流动方程进行离散化，并对每个网格块计算其解。在本工作中，我们使用斯坦福的基于自动微分的通用研究模拟器，AD-GPRS，用于所有的流模拟。设  $n$  表示模型中网格块的个数.. 通过使用  $p$  和  $S$  两个主要变量来充分定义流系统[40].. 在每个网格块中，系统中的变量总数为  $2n$ 。我们定义  $x_t = [p \text{ 和 } S]^T \in \mathbb{R}^{2n}$  作为流动变量在特定时间步长  $t$  的状态向量，其中  $p_t$  和  $S_t$  表示时间步长  $t$  的每个网格块中的压力和饱和度。

表示离散化全隐式系统的非线性代数方程组的集合可以表示为：

$$g(X_{t+1}, X_t, U_{t+1}) = 0, \quad (2)$$

在那里  $2n$  是我们寻求驱动到零的残差向量（非线性代数方程组），下标  $t$  表示当前时间电平， $t$  表示下一个时间电平+1， $U_t$  指定井控变量，可以是井底压力(BHPs)或井速率的任何组合。给你  $w$  表示系统中的井数。在本工作中，我们根据必和必拓的规格操作生产井，并在下面操作注射井

费率规格。我们的治疗在这方面是一般的，其他控制设置也可以应用。

2. 牛顿法通常用于求解方程定义的全阶离散非线性系统.. 这就需要构造维数  $x_{2nb}$  的稀疏雅各比矩阵，然后在每个时间步长的每次迭代中求解维数为  $2n^{\wedge}$  的线性系统。线性系统的求解往往是仿真中最耗时的部分.. 正如稍后将解释的那样，POD-TPWL 和基于深度学习的 E2CROM 都避免了这个高维系统的测试时间构建和解决方案。

## 2.2. POD-TPWL 配方

许多基于深度学习的模型涉及的治疗方法 with 现有 ROM 中使用的 not 直接相似。相反，它们需要从图像分类、语言识别或其他基于 PDE 的应用程序中派生出来的机器学习方法。我们的 E2CROM 在这个意义上有点不同，因为它的三个主要组件直接类似于现有 ROM, POD-TPWL 中使用的组件。由于 POD-TPWL 已被广泛开发用于地下流动应用，我们认为值得讨论 POD-TPWL 和 E2CROM 之间的对应关系。为了使这一讨论成为可能，我们首先提供了用于油藏模拟的 POD-TPWL 的高级概述。有关最近 POD-TPWL 实现的详细信息，请参阅，。[17 18 19 20]

如前所述，POD-TPWL 和其他基于 POD 的 ROM 涉及离线（训练时间）阶段和在线（测试时间）阶段。在离线阶段，使用全阶模拟器（本工作中的 AD-GPR S）执行许多训练模拟运行。这里的目标是预测测试时间结果与不同的井控序列。因此，在训练运行期间，我们应用不同的井控序列  $U=[U_1, \dots, U_{N_{ctrl}}]$   $x_{N_{ctrl}}$

在那里你和  $6$   $k=1, \dots, N_{ctr}$  包含控制步骤  $k$  处所有井的设置，而  $N_{ctrl}$  表示训练运行中控制步骤的总数。在典型的模拟中，控制步骤比时间步骤少得多（在我们的例子中，我们有 20 个控制步骤和大约 100 个时间步骤）。所有网格块中的状态变量（称为快照）和导数矩阵在训练运行的每个时间步骤保存。在测试时，用与训练运行不同的控制序列进行模拟。从训练运行中保存的信息被用来（非常有效地）近似测试解决方案。

POD-TPWL 需要 (1) 从高维空间投影到低维子空间，(2) 低维子空间动力学的线性近似，(3) 向高维空间投影。投影矩阵  $\Phi \in \mathbb{R}^{2nb \times n^{\wedge}}$  是基于解快照矩阵的奇异值分解(SVD)构造的（这些快照矩阵在所有训练运行的所有时间步骤中都包含全阶解）。给定  $\Phi$ ，高维状态  $x \in \mathbb{R}^{2nb}$  可以用  $\Phi \in \mathbb{R}^{2nb \times n^{\wedge}}$  低维变量表示。占使用

$$x \approx \Phi \xi, \tag{3}$$

缩小空间的维数在哪里，具有  $n^{\wedge}$ 。注意，在实践中， $6$

对于压力和饱和变量，SVD 和随后的投影分别执行。因为鱼是正交的，所以我们也  $\mathbf{f} = \mathbf{x}$ 。

在讨论低维空间中的 POD-TPWL 近似之前，我们首先给出了高维的线性化。下面，TPWL 公式(用 POD 表示状态， $\mathbf{x} = \mathbf{f}$ ，应用于右侧)可以表示为[18]

$$\mathbf{f}_{t+1}^x = \mathbf{f}_t + \mathbf{A}(\mathbf{f}_t - \mathbf{f}_t^b) + \mathbf{B}(\mathbf{u}_{t+1} - \mathbf{u}_t), \quad (4)$$

在哪里

$$\frac{d\mathbf{g}_{t+1}}{d\mathbf{x}_{t+1}} \mathbf{e} R^{2nb \times 2nb} \mathbf{f}_{t+1} = \mathbf{e} R^{2nb \times 2nb} \frac{d\mathbf{g}_{t+1}}{d\mathbf{x}^z}, \quad \mathbf{b}_{t+1} = \frac{d\mathbf{g}_{t+1}}{d\mathbf{u}_{t+1}} \mathbf{e} R^{2nb \times n_w}. \quad (5)$$

在这里，下标  $t$  和  $t+1$  表示测试运行中的时间步骤，而下标  $i$  和  $i+1$  表示训练模拟中的时间步骤。注意那个等式。4[18] 与表达式略有不同，因为时间步长名称现在是订阅的，以符合嵌入到控制方程稍后显示。变量  $\mathbf{f}_t$  是 Eq. at 时间步长  $t$  的真（高阶）解的投影。变量  $\mathbf{x}_{t+1} \mathbf{e} R^{2nb}$  不同于  $\mathbf{X}_{t+1}$ ，因为它表示时间步长  $t+1$  通过线性化而不是通过全阶系统(E q)的解来近似的全阶变量。2)。从这里开始，我们将使用没有“帽子”的变量来表示真正的高阶解(例如， $\mathbf{x}$ )或用矩阵鱼投影的真解(例如， $\mathbf{f} = \mathbf{x}$ )。此外，我们将使用带有“HATS”(x 和  $\mathbf{f}$ )的变量来指定由 ROM 近似的解（重建或预测，稍后将详细解释）。变量  $\mathbf{u}_{t+1}$ ， $\mathbf{u}_{t+1} \mathbf{e}$  是时间步骤  $t+1$  和  $i+1$  的设置——这些是由用户指定的或由优化算法指定的。

在方程的左侧应用 POD 表示和两侧的约束约简（投影）。4 在经过一定的重排后，给出了低维空间中的解近似值。

$$\mathbf{f}_{t+1}^x = \mathbf{f}_t + \mathbf{A}(\mathbf{f}_t - \mathbf{f}_t^b) + \mathbf{B}(\mathbf{u}_{t+1} - \mathbf{u}_t), \quad (6)$$

用简化的导数矩阵定义为

$$\mathbf{J}_{t+1} = (\mathbf{J}_{t+1})^T \mathbf{J}_{t+1} \mathbf{A}_{t+1} = (\mathbf{J}_{t+1})^T \mathbf{J}_{t+1} \mathbf{B}_{t+1} = (\mathbf{J}_{t+1})^T \mathbf{B}_{t+1}. \quad (7)$$

这里  $\mathbf{J}_{t+1} \mathbf{e}$  鸭， $\mathbf{A}_{t+1} \mathbf{e}$  鸭和  $\mathbf{B}_{t+1} \mathbf{e}^{n_w}$ 。矩阵  $\mathbf{A}_{t+1}$  表示时间步骤  $i+1$  时的约束约简矩阵。变量  $\mathbf{f}_{t+1} \mathbf{e} R^{2nb}$  表示在时间步长  $t+1$  时通过线性化近似的约化变量。

在线阶段（测试时间），我们不知道  $\mathbf{f}$ (E q 的投影真解。2 在时间步骤  $t$ )。相反，我们有  $\mathbf{f}_t$ ，减少的变量近似通过线性化在时间步骤  $t$ (从方程计算。6 在上一个时间步骤)。因此，在测试时，Eq. 6 变成

$$\mathbf{C} \mathbf{T} + \mathbf{x} = \mathbf{f}_t + \mathbf{A}(\mathbf{f}_t - \mathbf{f}_t^b) + \mathbf{B}(\mathbf{u}_{t+1} - \mathbf{u}_t). \quad (8)$$

请注意，CT 现在出现在右侧，而不是在测试时，执行线性化的训练‘点’(这一点定义了  $i$  和  $i+1$ )，是使用‘点选择’过程确定的。这一点的选择依赖于 CT（详见），因此约简导数矩阵  $\mathbf{J}_{t+1}$ ， $\mathbf{A}_{t+1}$  和  $\mathbf{B}_{t+1}$  都可以被认为是 CT 的函数。在 POD-TPWL 的最后一步中，近似解通过  $\mathbf{x} = \mathbf{f}$  的应用被投影回全阶空间，其中 CE 是子空间中的近似测试状态，变量  $\mathbf{x} \mathbf{e} R^{16 \times 19 \times 2nb}$  是全维空间中相应的近似状态。

在 POD-TPWL 中，上述每个步骤都可以看作是一个优化（在某些情况下是启发式的），正如

我们现在所考虑的那样。利用 POD 过程构造投影矩阵  $\mathcal{P}$ 。这具有生成的基矩阵最小化投影误差  $e_{\text{proj}}$  的属性，定义为

$$e_{\text{proj}} = \|\mathbf{x} - \mathcal{P}\mathbf{x}\|_2 \quad (9)$$

其中  $\mathbf{x} \in \mathbb{R}^{2nb}$  是全阶状态变量。

[18]此外，正如 He 和 Durlofsky 所讨论的，约束约简误差可以定义为

$$e_{\text{c}} = \|\mathbf{x} - \mathbf{h}^*\|_2 \quad (10)$$

其中  $\mathbf{x}$  对应于方程中的解  $\mathbf{x}_{t+i}$ 。4（在应用约束约简之前）；该变量表示为  $\mathbf{x}_{2in}$ 。变量  $\mathbf{C}$  对应于方程中的解  $\mathbf{C}_{t+i}$ 。6（在应用约束约简后），表示为  $\mathbf{C}_{3in}$ 。符号  $\|\cdot\|_{\text{II}}$  是一个定义为  $\|\mathbf{e}\|_{\text{II}} = \sqrt{\mathbf{e}^T \mathbf{S} \mathbf{e}}$  的规范，具有  $\mathbf{e} \in \mathbb{R}^{2nb}$  和  $\mathbf{S} \in \mathbb{R}^{2nb \times 2nb}$  其中  $\mathbf{S}$  是对称正定矩阵。最优约束约简矩阵  $\mathbf{S}$  可以通过最小约束约简误差来确定，即，

$$\mathbf{S} = \mathbf{A} \mathbf{M} \mathbf{A}^T \quad (11)$$

如果矩阵  $\mathbf{S}$  定义为  $\mathbf{J}$  [18] 然后，跟随 EQ. 21 到 27 英寸，我们到达最小二乘彼得罗夫-伽勒金投影，即，

$$\mathbf{S} = \mathbf{J} \mathbf{J}^T \quad (12)$$

这种治疗，正如我们所看到的，在一个特定的规范中是最佳的，现在通常在 POD-TPWL 中使用。

因此，很明显，在 POD-TPWL 过程中，低维态表示和约束约简过程是基于两个不同的优化，目标函数  $e_{\text{proj}}$  和 POD-TPWL 的其余方面是点选择。不同的点选择策略已经被用于 dif

不同的应用程序，这些通常包括一个启发式组件。然而，这些过程确实需要最小化一个“距离”度量，它量化当前测试点和一组大型训练运行点之间的距离（在特定于应用程序的意义上）。因此，这一步骤也需要优化。这些 POD-TPWL 组件优化直接对应于将应用于嵌入到控制框架中的损失函数最小化。然而，一个关键的区别是，在 E2C 框架中，所有步骤都是一起优化的，而不是像 POD-TPWL 那样单独优化。

### 3. 嵌入控制配方

在本节中，我们开发了一个包含物理约束的嵌入到控制 ROM。对各种 E2C 组件建立了 POD-TPWL 的类比。这里提出的 E2C 模型通常遵循 Watter 等人开发的模型。虽然介绍了几个重要的修改，如下所示。[39]

#### 3.1. E2C 概述

嵌入到控制框架需要三个处理步骤：编码器或推理模型，将系统变量从高维空间投影到低维子空间（这里称为潜在空间），一个近似于低维系统动力学的线性过渡模型，以及一个解码器或生成

模型，将解决方案投影回高维(全阶)空间。最初由 Watter 等人提出的 E2C 框架。使用 VAE 架构的编码器和解码器程序，这允许他们解释不确定性的预测。在这里的公式中，VAE 体系结构被简化为自动编码器(AE)体系结构，因为我们正在考虑确定性系统。我们注意到，自动编码器(AE)体系结构通常用于语义分割，其中图像的每个像素与类标签相关联，并且用于深度预测，其中场景的三维几何是从二维图像中推断出来的。在地下流动模拟的背景下，AE 结构被用来构造代理模拟模型作为图像到图像的回归，其中输入图像是储层性质（例如渗透率场），输出是状态变量。  
[39][41][42][27 28]

图 1 显示了我们的嵌入到控制模型的总体工作流。压场  $P_{ie}$  是此插图中显示的唯一状态变量(下标  $i$ ，与之不同表示训练运行中的时间步骤)，尽管我们的实际问题还包括饱和场  $S=e$ 。附加的状态变量将出现在更一般的设置中（例如，如果考虑耦合流动-地质力学模型，则位移）。

图中的方框 1。1 显示压力快照  $p^*e$ “气我=1 完全有序的  
空间，其中  $N_s$  是快照的总数。符号  $Q^{nc}$  在漏斗 2 中，编码器表示将整个空间投影到一个潜在空间， $\odot$  表示编码器中的所有“可学习”参数。方框 3 中的变量  $Z_{pe}$  是压力的潜在变量，有  $L$  潜在空间的维度。

在方框 3 中，测试模拟结果用线性过渡模型近似于潜在空间。变量  $z_{0e}$  表示测试运行的初始潜在状态， $u_{te}$ ， $t=1:N_{ctri}$  指定测试运行的控制序列， $n_w$  那个



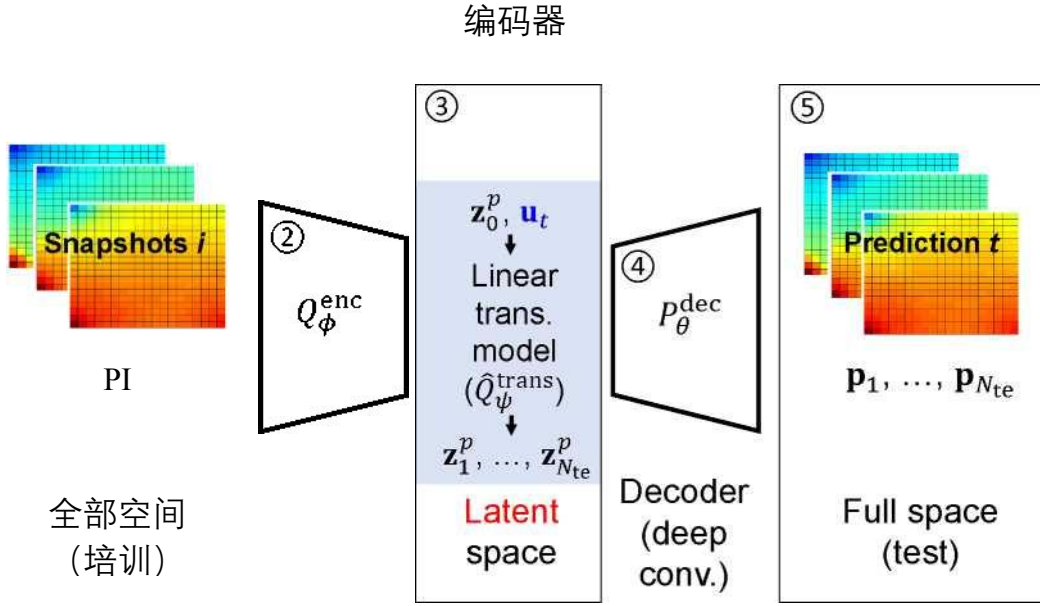


图 1：嵌入到控制(E2C)概述.

井数（如前面所述），下标  $t$  表示测试运行中的时间步长， $N_{ctri}$  是测试运行中的控制步骤数。线性过渡模型  $Q_0$  表示可学习参数)以  $z_{ppe}$  和  $u_e$  为输入，输出  $z_{pe} \neq 1, \dots, N_{特}$  依次，其中  $N_{特}$  是测试运行中的时间步骤总数。

解码器  $P_{\theta}^e$ （由漏斗 4 表示，其中  $\theta$  表示解码器中的所有可学习参数）然后投影变量  $z^p$  回到全阶状态  $p_{te}$ ，如框 5 所示..

[34]嵌入到控制 ROM 将控制变量  $u_{te}$  自然地包含在框架中.. 这是相对于 Gonzalez 和 Balajewicz 开发的基于 VAE-LSTM 的 ROM 的一个重要区别，其中系统控制不包括在模型中。在下面的小节中，将详细讨论嵌入到控制框架的三个主要组件，编码器、线性过渡模型和解码器。还将介绍具有物理约束的损失函数以及 E2C 实现细节。

### 3.2. 编码器组件

[39]编码器提供全阶状态变量的低维表示.. 与 Watter 等人最初的嵌入到控制实现相比。在这里，我们采用 AE 而不是 VAE 结构。在这种处理下，只估计潜在变量的平均值，而不是方差。此外，

我们不需要在潜在空间中进行采样过程。因此，在列车时刻，编码器可以简单地表示为.

$$\mathbf{z}_T = \mathbf{Q}(\mathbf{t}),$$

在哪里  $Q^c$  表示编码器，如前面所解释的。变量  $x_{gr}^{2nb}$  是时间步长  $t$  和  $z$  的全阶状态变量。式中， $G$  为对应的潜变量，与  $l_z$  潜在空间的维度。在后面给出的例子中，我们考虑了一个二维  $60 \times 60$  油水模型（这意味着全阶系统为 7200 维），我们设置了  $l_z = 50$ 。请注意，Eq. 类似于 Eq. 13.3 在 POD-TPWL 过程中，除了 POD 中的线性投影外，还用非线性投影  $Q^c$  代替  $c$  在编码器里。按照前面描述的惯例，我们使用没有“HAT”的变量来表示 Eq. 的（投影）真解，这些真解可以从训练运行中获得。带帽子的变量指定由测试时间 ROM 提供的近似解。2

2 编码器在 E2C 模型中的详细布局如图所示。在训练过程中，通过编码器网络输入压力和饱和快照序列，以及潜在状态变量  $z$  序列，生成  $G$ 。这里使用的编码器网络由四个编码块的堆栈、三个剩余卷积(Res Conv)块的堆栈和一个密集层组成。与使用的编码器相比，图中的编码器更复杂(即，它包含 Res Conv 块，并且具有更多的卷积层)。这里可能需要一个更复杂的结构，因为与在（例如，手推车平衡和三链路机器人手臂规划）中处理的原型规划任务相比，基于 PDE 的压力和饱和场的

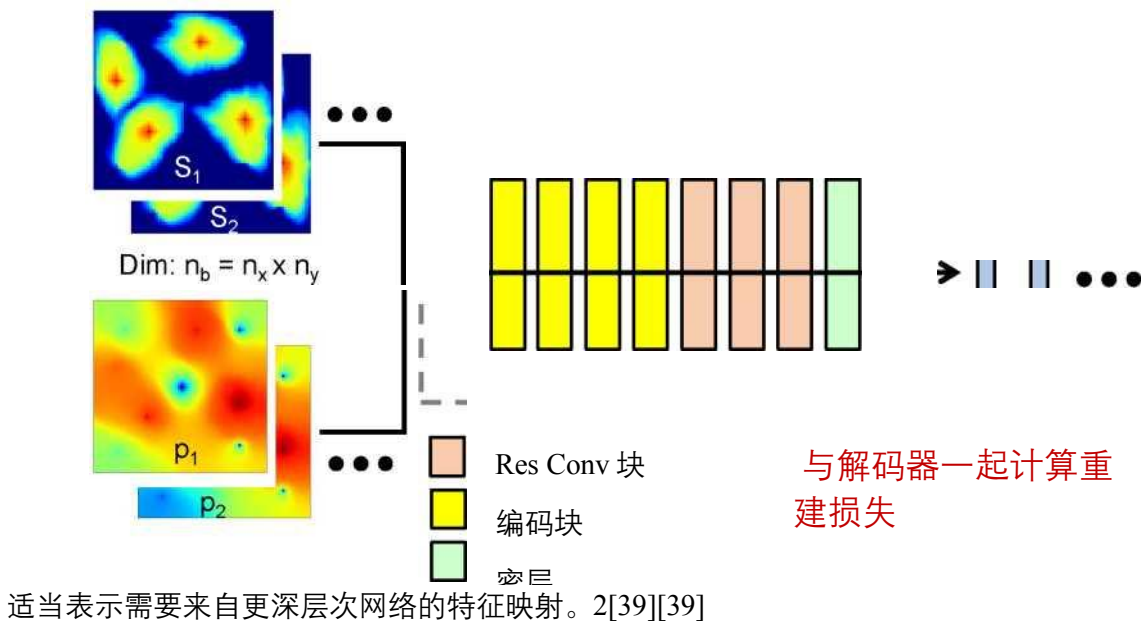


图 2：编码器布局

[24][29-34] 类似于 Liu 等人提出的 CNN-PCA.. 它使用 CNN 中的滤波器操作来捕获表征地质特征的空间相关性，嵌入控制框架使用大量的卷积滤波器来表示由底层控制方程确定的压力和饱和场的空间分布。早期的实现基于 AE/VAE 的 ROM，已经证明了卷积滤波器在流体力学问题中捕捉这类字段的潜力。因此，我们的编码器网络主要由这些卷积滤波器(以二维卷积层的形式，即卷积 2D 层)组成。编码器网络的更多细节在 Table of 中提供。[43] a. 2 附录 A

编码块的输入首先通过卷积操作输入，卷积操作也可以看作是线性滤波器。下面表达式中，  
编码器网络

性滤波的数学公式是.[24]

$$f_{ij}(x) = \sum_{p=-n}^n \sum_{q=-n}^n w_{p,q} x_{i+p, j+q} + b \quad (14)$$

[44] 其中  $x$  是输入状态图，下标  $i$  和  $j$  表示  $x$  和  $y$  坐标方向指数， $w$  表示大小  $(2n+1) \times (2n+1)$  的线性滤波器（模板）的权重， $f_{ij}(X)$  在空间位置  $(i, j)$  指定  $x$  的滤波器响应图（即特征图）， $b$  是一个标量参数，称为偏差。请注意，通常有许多与 conv2D 层相关联的过滤器，因此收集所有这些操作的过滤器响应映射是三阶张量。然后，输出滤波器响应映射通过批处理归一化(批处理 Norm)层，该层对每个训练数据子集应用归一化操作（将平均值移动到零，并按标准差重新标度）。批处理 Norm 操作是深度神经网络有效训练的关键步骤，因为它使学习过程对参数初始化不那么敏感，这意味着可以使用更大的初始学习速率。将非线性激活函数 ReLU(校正线性单元， $\max(0, x)$ ) 应用于归一化滤波器响应映射，给出编码器块的最终响应（输出）。这种非线性响应被称为编码块的“激活”。卷积 2D 批处理 Norm-ReLU 体系结构（排序有变化）是 CNN 中的一个标准处理步骤。编码块结构的说明可以在图中找到。[45] a. 18(a) 附录 A

为了正确地结合能够表示空间压力和饱和分布的特征映射，如基础控制方程所确定的，需要一个具有许多卷积层的深度神经网络。然而，深层神经网络很难训练，主要是由于梯度消失问题。这意味着损失函数相对于模型参数（滤波器的权重）的梯度变得非常小，这对训练产生了负面影响。他等人。[46][47] 通过创建一个额外的标识映射来解决这个问题，称为 RESNET，它绕过非线性层。遵循 ResNet 的思想，我们在编码器网络中添加了一堆 Res Conv 块，以加深网络，同时减轻消失梯度问题。在 res Conv 块中的非线性层通常仍然遵循 conv2D 批处理 Norm-ReLU 体系结构.. 见图 1。a. 18 附录 A(c) 输入 Res Conv 块的视觉表示。

与编码块相似，Res Conv 块的输出是一堆低维特征映射。这个特征映射堆栈被“扁平”到一个向量（由于大量的特征映射，它仍然是一个相对高维的向量），然后输入到一个密集层。密集（完全连接）层只是一个线性投影，它将高维向量映射到低维向量。

[27][48][27][27][41] 这里使用的编码器网络的总体结构与朱和扎巴拉斯在三个关键方面构造的不同。首先，RESNET 在我们的编码器中使用，而它们使用密集网来缓解消失梯度问题。另一个关键区别是编码器（和解码器）在末端不包括密集层，这意味着编码器在末端输出一堆特征映射。大量的特征映射（即高但相对较薄的三阶张量）对于随后由线性过渡模型执行的顺序线性操作来说是太高的维度。最后，Zhu 和 Zabarar 采用了 U-Net 体系结构，当编解码器（例如压力场）的输出与输入(例如渗透率图)不同时，这是合理的，就像它们的设置一样。然而，U-Net 体系结构是不适当的，当输入和输出类型相同（即压力和饱和场）时，可能会导致过度拟合，就像这里的情况一样。

嵌入到控制 ROM 中的编码器（和解码器）类似于 POD-TPWL 中使用的 POD 表示。如前所述，\$通过快照矩阵的 SVD 构造的基矩阵具有其最小化 Eproj 的特点。9。在编码器的上下文中，计算了类似于 POD 的 eproj 的重建损失 LR。概念上，“最好的” $Q^{nc}$  是通过最小化 LR 来发现的。然而，正如前面提到的，用于嵌入到控制模型的优化涉及到一起考虑的所有三个处理步骤，因此 LR 不是单独最小化的。

### 3.3. 线性过渡模型

3 线性过渡模型将潜变量从一个时间步骤演化到下一个时间步骤，给出了控制。图显示了线性过渡模型是如何在离线阶段（列车时间）构建和评估的。线性过渡模型的输入包括当前状态  $z$  的潜在变量  $z_t$ ，当前步骤控制  $UT_t$  和时间步长。该模型输出下一时间步骤  $z_t$  的预测潜在状态  $z_{t+1}$ 。例如：我们重申这一点， $z_{t+1}$  [39] 表示线性过渡模型的输出。线性过渡模型的结构通常遵循 In，由三个变换(Trans)块和两个致密层组成。跨块遵循密集批 Norm-ReLU 体系结构（密集表示密集层），这被认为是完全连接网络的标准处理步骤。反式块结构如图所示。a. 18 附录 A (c) 加入。变量  $z_t$  和  $At$  首先被输入反式块。反式块的最终激活向量  $h^{ns}$  然后用于构造 E 处的线性化矩阵  $_{XLZ}$  和  $BTE_{\text{opt}}$  通过两个独立的致密层。然后将  $At$  和  $Bt$  矩阵与当前状态  $z_t$  和当前阶跃控制  $ut$  的潜在变量相结合，在下一个时间步骤  $ZT$  预测潜在变量  $_{x}$ 。

用于确定线性过渡模型参数的优化再次类似于 POD-TPWL 中的关键步骤。在 POD-TPWL 中，目标是最小化预测的还原状态  $\hat{f}$  与预测的真实状态  $f$  之间的差异  $\| \hat{f} - f \|$ 。这部分是通过确定最优约束约简矩阵史来实现的，如 EQ 中所描述的。10 11 以及。给定这个最优史矩阵，出现在 POD-TPWL 中的矩阵。



图 3：线性过渡模型布局

6)演化方程(E q., 即  $J; +I, A^*$ 和  $B^*$ )都是完全定义的。如前所述, 点选择表示 PODTPWL 中出现的另一个 (启发式) 优化.. 同样, 在嵌入到控制公式中, 通过瓦+]与  $z_{t+i}$  相比较, 计算过渡损耗  $LT$ , 其中&+]是线性过渡模型的输出,  $z_{t+i}$  是编码器在时间步长  $t+1$  处投影的状态。过渡损失有助于总损失函数, 这在离线阶段被最小化。

列车时刻的线性过渡模型也可以表示为.

$$\text{泽特+我} = \mathbf{Q} \mathbf{R} (z_t, u_{t+1}, a_t) \quad (15)$$

其中  $AT$  是时间步长, 函数  $Q^*$  是先前定义的线性过渡模型 ( $\wedge$ 表示模型中的所有可学习参数),  $Z_{t+IE}$  表示线性过渡模型预测的  $t+1$  处的潜在变量。更具体地说, 情商。15 可以表示为

$$+我 = \text{在}(Z_t, a_t) Z_t + B_t(Z_t, a_t) u_{t+i}, \quad (16)$$

在那里  $_{xz}$  和  $BTE$   $_{xrw}$  是矩阵。与里面的表达一致, [39] 这些矩阵是由

$$\text{VEC}[AT] = W_a h^* (z_t, a_t) + b_A, \quad (17)$$

$$\text{VEC}[B T] = W_B h^* (z_t, a_t) + b_b, \quad (18)$$

其中  $VEC$  表示矢量化, 因此  $\text{VEC}[At]eR^{(l_2)^{+-}}$ 和  $\text{VEC}[Bt]eR^{(l_{znw})^{+-}}$ 。变量碍严  $e^*$ ”析冲表示三个转换块 (总共称为转换网络) 之后的最终激活输出。情商中的寸。17 18 是方程中明的子集。15 因为后者还包括转换网络之外的参数。在这里  $我是 Z_{xntrans}$ , 世界银行  $(l_{znw})_{xntrans} b$  一个  $eR^{(l_{Zxi})}$  和  $b_b e R^{(l_{znw})^{+-}}$ ,

地点，地点  $\mathbb{A}NS$  表示变换网络的维数。我们准备好了  $\mathbb{A}NS=200$  在这里测试的模型中。

在在线阶段（测试时间），线性过渡模型略有不同，因为潜在变量输入模型( $z_t$  从最后一个时间步骤预测  $E$ )。因此，在测试时，Eq. 16 变成

$$Z_{t+1} = (Z_t, \Delta t)Z_t + B_t(Z_t, \Delta t)u_t \text{ 和 } I_o. \quad (19)$$

注意，唯一的区别是  $Z_t$  在等式的右边.. 16 用 Eq 中的  $Z_t$  代替。 19.

线性过渡模型的试验时间公式直接类似于 POD-TPWL 中的线性表示步骤.. 在 POD-TPWL 中，由于训练步骤  $I$  (因此  $I+1$ ) 是根据涉及  $\ell$  的点选择计算确定的，因此在线表达式(E q)中出现的矩阵。8) 可以被认为是七的功能。经过一些重组，Eq. 8 然后可以写成

$$Z_{t+1} = (M_t)MT + BT (M_t)u_{t+1} + cT \quad (20)$$

在哪

$$A T = (J ; \cdot)^T A^{+1}, \quad BT = -(j ; \cdot)^T u ; ^{+1}, \quad (21)$$

19 因此，我们看到 Eq. 对于嵌入到控制公式的在线阶段，Eq. 与 Eq 相同。20 用于 POD-TPWL 的在线阶段。关键的区别是，E2C 中的矩阵  $A_t$  和  $B_t$  是由深度学习模型确定的，而不是由训练运行中的导数矩阵构造的。还请注意，CT 不出现在 E2C 公式中，因为这种表示不需要围绕附近的解决方案展开。

### 3.4. 解码器组件

译码器与编码器相似，可以表示为.

$$z = PT(Z T), \quad (22)$$

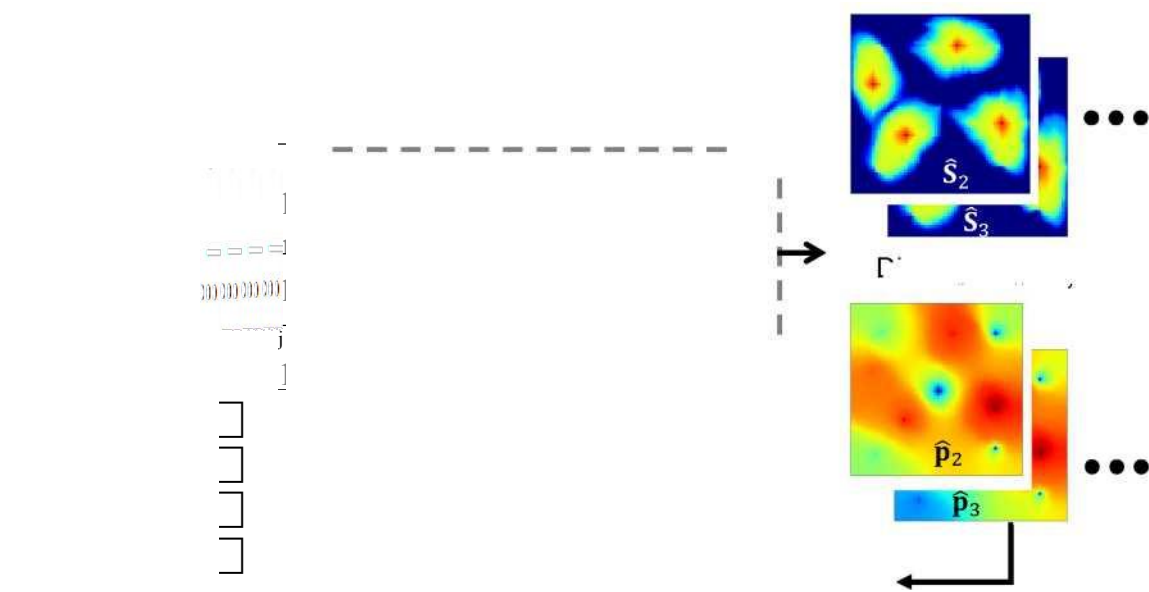
在那里  $PG^*$  是以前定义的解码器。变量  $z_t \in \mathbb{R}^{2nb}$  表示时间步长  $t$  处的重构状态变量(与高保真状态变量  $x_t \in \mathbb{R}^{2nb}$  不同)从训练快照)，虽然输入到解码器  $ZTE$  是由  $XT$  编码确定的潜在变量。如果输入为  $2t+1E$ ，则该变量是线性过渡模型方程在时间步长  $t+1$  时预测的潜在变量。22 变成

$$X_{t+1} = PT(z_{t+1}), \quad (23)$$

其中  $X_{t+1}$  是时间步长  $t+1$  的预测状态变量。注意那个等式。22 只出现在列车时间过程中（计算重建状态），而方程.. 23 在列车时间和测试时间都有相同的形式..

4 解码器的详细结构如图所示。线性过渡模型预测的潜在变量(在时间步长  $t+1$ )被输入到解码器网络作为输入，15

并输出预测的高维状态.. 解码器的结构类似于编码器，除了它是相反的顺序（这并不奇怪，因为解码器正在进行逆操作）。这里的解码器由一个密集层、三个 RES Conv 块的堆栈、四个解码块的堆栈和一个 Conv2D 层组成。密层将一个低维潜向量转换为一堆特征映射（重塑后）.. 这些特征映射是在经过一堆 res Conv 块和解码块时展开的。压力和饱和场的空间分布是从特征映射中连续“提取”的，因为我们在解码器中进行下游。末端的 conv2D 层将扩展的特征映射转换为压力和饱和场作为最终输出.. 编码器的更多细节在 Tablein 中提供。解码块的布局如图所示。 a. 3 附录 a a. 18(b) 附录



A

图 4：解码器布局

为了确定解码器中的可学习参数 B，在离线过程中最小化了预测损失（以及其他损失）。关于这一优化的更多细节将在稍后介绍。

### 3.5. 具有物理约束的损失函数

我们已经描述了嵌入到控制框架的每个组件。我们现在解释模型参数是如何在离线阶段确定的。嵌入到控制框架的参数分别为编码器、线性过渡模型和解码器的饱和加和 B。要最小化的目标函数是量化模型预测输出状态变量的总体性能的总损失函数。

我们简要介绍了重建损失(Lr)、线性过渡损失(Lt)和预测损失(Lpd)，它们是总损失函数的主要组成部分。更具体地说，训练数据点 I 的重建损失可以表示为

$$L_r = \sum_{i=1}^N \left( \left\| \mathbf{x}_t - \hat{\mathbf{x}}_t \right\|^2 + \left\| \mathbf{u}_t - \hat{\mathbf{u}}_t \right\|^2 \right) \quad (24)$$



在那里  $i=1, \dots, N_t$ ，与  $N_t$  表示在训练运行中生成的数据点的总数。注意， $N_t = n_s - n_u$  艾恩，哪里  $N_s$  是训练运行和  $n$  中快照的总数， $A_{IN}$  是所进行的训练模拟的数量。在这里  $i$  和  $N_s$  不同的原因是，对于包含  $N$  的训练模拟  $n$  快照，只有  $N_{tr}$  可以收集—1 个数据点（因为需要对状态，在顺序时间步骤）。变量  $x_t$  是来自训练模拟的时间步长  $t$  和  $X$  的状态变量  $z_t = PG^e(z_t) = PG^e(Q^{\wedge}(x_t))$  表示编码器和解码器重建的状态。

训练点  $I$  的线性过渡损失被类似地定义为

$$(L)_{T \text{ 我}} = \{ \|z_{t+1} - Q^{\wedge}_{nc}(XT+I)\|^2 \}_{i=1}^{N_t} \quad (25)$$

在那里  $z_{t+1} = Q^{\wedge}_{nc}(XT+I)$  是从  $t+1$  处的全阶状态变量编码的潜在变量，变量  $z_t + I = Q^{\wedge}_{nc}(z_t, u_t + i, a_t)$  表示线性过渡模型预测的潜在变量。最后，将训练点  $i$  的预测损失定义为

$$(PD)_{\text{我}} = \{ \|XT+I - X_{t+1}\|^2 \}_{i=1}^{N_t} \quad (26)$$

其中  $XT+I$  从训练模拟中指定时间步骤  $t+1$  的状态变量，而  $XT+I = P^{\wedge}_{nc}(z_t + i)$  表示 ROM 预测的全阶状态变量。数据失配损失是所有训练数据点平均损失的总和，

$$l_d = \frac{1}{N_t} \sum_{i=1}^{N_t} (L)_{\text{我}} + \lambda (PD)_{\text{我}} \quad (27)$$

其中  $A$  是一个权重项。

直到这一点所描述的 ROM 是一个纯粹的数据驱动模型，即模型的目标是最小化 E2C 输出与高保真解之间的像素级差 (HFS 作为“真”参考解)。物理行为在某种程度上是由 E2C 从输入压力和饱和快照推断出来的，但它没有明确执行。如果 ROM 是使用 Eq 中给出的损失函数  $L_d$  训练的。然而，可以观察到非物理效应。这如图所示。在这里，我们展示了在特定时间压力场的预测 (问题设置将在章节中详细描述。高保真解如图所示.. 4) 5(A)，仅基于  $L_d$  的 E2C 压力场出现在图中。5(b)。虽然这两个结果在视觉上是相似的，但图中的差异图。5(c) 表明 E2C 结果不够平滑，在某些空间位置出现较大的误差。这可能会对井率预测产生重大影响，这是一个基本的 ROM 输出。

为了解决这一问题，我们将数据失配的损失与基于流物理学的损失函数结合起来。具体来说，我们寻求最小化每对相邻网格块之间的通量不一致。额外的重量也放在关键井的数量上。我们考虑重建(在时间步骤  $t$ )和预测(在时间步骤  $t+1$ )。因此，我们定义了每个数据点的基于物理的损失， $(LP)^*$ ，作为。

$$(l) p)我=\{||k[{}^{t}vpt-vpt)侦察+(vpt+1-vpt+1)pred||2\}i +7\{他^{-發)侦察} +^{(a)} 不+1^{-QW_{+1}}||2\}i 前。 \quad (28)$$

这里是  $p_t, p_{t+i6}$  是训练数据中时间步骤  $t$  和  $t+1$  的压力场，它们是状态变量  $x_t$  和  $x_{t+i}$  的分量， $p_t, p_{t+i6}$  表示 ROM 压力重构(在时间步长  $t$ ，在 Eq 之后定义)。24) 和预测(在时间步长  $t+1$ ，定义在等式之后。26)。变量  $qW, qW+i6$  是训练数据中的良好量， $qW, qW+i6$  是用 ROM 重建(在时间步长  $t$ )和预测(在时间步长  $t+1$ )的良好量。回想一下  $w$  是井的总数。变量  $y$  是一个参数，它定义了损失函数  $L_p$  中井数据损失的权重。

28 方程右边的术语，对应于方程中的通量和源项。1。在本文的例子中，我们指定了注入井的速率和生产井的 BHPs。使用此规范，注入速率损失为零。生产井跟踪的关键数量是每口井的井块压力。这是因为生产速率与井筒压力(BHP 在这种情况下，这是指定的)和井块压力之间的差异成正比。比例系数是相迁移率  $A_j$  和所谓的井指数的乘积，它取决于渗透率、块体尺寸和井筒半径。由于在这种情况下，整体井率很大程度上受到井块压力的影响，我们将第二项设置在 Eq 的右侧。[49] 28 到  $Yllpf-p; II2$ ，其中  $p^w_6$  和  $p^w_6(j=t, t+1)$  表示真实和 ROM 井块压力，而 NP 是生产井的数量。这里  $Y$  是一个修改的权重，它解释了井指数。

基于物理的损失函数是通过在所有数据点上的平均  $(L_p)^*$  来计算的，

$$\mathcal{L}_p = \frac{1}{N_t} \sum_{i=1}^{N_t} (\mathcal{L}_p)_i. \quad \text{即,} \quad (29)$$

将数据失配损失与这种基于物理的损失结合起来，总损失函数就变成了

$$我=我+了一笔贷款, \quad (30)$$

其中  $a$  是一个权重项。通过有限的数值实验，我们发现  $a=0.033$  和  $Y=20$  是这些参数的合适值。使用总损耗函数  $L$  的压力场在特定时间的 E2CROM 预测如图所示。5(e)。图 15(D)再次是高保真解(与图中的相同)。5(A)，差异图如图所示。5(f)。我们看到，当  $L_p$  包含在损失函数中时，ROM 预测得到了明显的改善。具体而言，减小了最大压力误差。

从 97psi 到 16psi，得到的场更平滑（因此更物理）。这证明了将基于物理的损失纳入 E2CROM 的好处。

### 3.6. E2C 实施和培训细节

为了训练 E2C 模型，我们使用数据集  $D = \{(x_t, x_{t+\tau}, u_t, x_{t+\tau})\}$ ， $t = 1, \dots, N$ ，其中  $N$  是训练运行数据点的总数。在本文的例子中，我们模拟了总共 300 次训练运行。这比 POD-TPWL（我们通常模拟三到五次训练运行）使用的要多得多，但我们期望 E2C 具有更高的鲁棒性。通过这一点，我们的意思是，ROM 有望在大范围的控制规范中提供准确的结果，而不是在 POD-TPWL 中的有限范围内。

而不是训练所有快照，这里我们设置  $N_{ctrl} = N_{tr} = N_{test} = 20$ 。这加速了训练，并将 ROM 预测集中在控制正在变化的时间步骤中感兴趣的数字上。这导致了  $N$  个数据点的总数  $n = 300 \times 20 = 6000$ 。

总损失函数相对于模型参数  $(\theta, \phi, 0)$  的梯度是通过嵌入到控制框架的反向传播来计算的。自适应矩估计(ADAM)算法用于这种优化，因为它已被证明是有效的优化深层神经网络。模型参数在每次迭代时更新的速率由学习速率  $\eta$  控制[50]。我们开始了  $\eta = 10^{-4}$ 。

归一化是一个重要的数据预处理步骤，其适当的应用可以提高学习过程和输出质量。对于饱和，我们有  $SE[0, 1]$ ，所以不需要归一化。压力和井数据，包括控制变量，进行归一化处理。归一化率  $q^0$  和压力（网格块压力和必和必拓） $p^0$ ，都是由

$$q^0 = \frac{q_{\text{如}}}{q_{\text{最大值}}}, \quad p^0 = \frac{p - p_{\text{民}}}{p_{\text{最大值}} - p_{\text{民}}} \quad (31)$$

这里  $q$  表示模拟器速率输出，单位为  $m^3$  每天， $Q_{\text{max}}$  和  $Q_{\text{min}}$  是注入速率的上下界， $p$  要么是网格块压力，要么是生产井必和必拓(psi 单位)， $P_{\text{min}}$  是必和必拓上的下界， $P_{\text{max}}$  是场压力的 1.1 倍（1.1 的因子基本上确保所有数据都在范围内）。

[51][52] 每个全阶训练模拟需要大约 60 秒的时间在双 IntelXeonES-2670CPU (24 核) 上运行。我们的 E2CROM 是使用带有 TensorFlow 后端的 Keras 实现的。离线训练过程（不包括训练模拟运行时）在特斯拉 V100GPU 节点上大约需要 10-12 分钟(确切的时间取决于分配的内存，内存可以从 8-12GB 变化)。该模型应用于 100 次测试运行，将在下一节中详细讨论。几乎所有的测试结果都是基于使用 300 次训练运行，尽管我们还使用 100 和 200 次训练运行提供了汇总错误统计。对于这些情况的离线训练需要大约与 300 次训练运行相同的时间，除了在全阶训练模拟中直接节省外。

#### 4. 结果使用嵌入到控制 ROM

在这一部分中，我们描述了油水模拟的模型设置，并给出了基于深度学习的 ROM 的仿真结果。本节详细考虑了其中一个测试用例；还提供了两个额外测试用例的结果。在本节中，我们还介绍了所有 100 个测试用例的汇总错误结果。附录 B

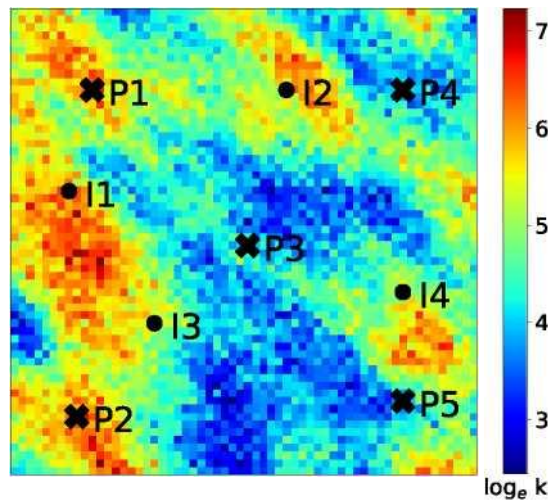
##### 4.1 模型设置

6 地质模型，在对数渗透场方面，如图所示。还显示了四个注射井和五个生产井的位置。储层模型包含  $60 \times 60$ （共 3600 个）网格块，每个块的尺寸为  $50\text{m} \times 50\text{m} \times 10\text{m}$ 。对数渗透场的相关结构采用指数变图模型，最大和最小相关长度为 1000m 和 500m，方位角为  $45^\circ$ 。算术平均渗透率为 158mD，对数渗透率标准差为 0.88.. 渗透率被认为是各向同性的，孔隙度被设置为 0.2 的恒定值。

相对渗透率函数由.

$$k_{rw}(s_w) = \begin{cases} 1 & \text{如果 } s_w = 1 \\ \frac{k_{rw}(s_w) - k_{rw}(s_{wr})}{1 - k_{rw}(s_{wr})} & \text{如果 } s_w < 1 \end{cases} \quad (32)$$

其中  $k=1.0$ ,  $k_w=0.7$ ,  $S_{or}=0.3$ ,  $S_{wr}=0.1$ ,  $a=3.6$ ,  $b=1.5$ 。流体密度设置为  $\rho_o=800\text{kg/m}^3$  和  $\rho_w=1000$



公斤/米<sup>3</sup>粘度指定为 $\mu_o=0.91\text{cp}$ ,  $\mu_w=0.31\text{cp}$ 。毛细压力效应被忽略。

库顶初始压力为 4712psi (325bar)，初始含水饱和度为 0.1%.. 系统中的主要变量总数为 3600x2-7200 模型运行共 2000 天.. 注入井通过指定时变水率来控制，生产井通过指定时变 BHPs 来控制。生产井和注入井的控制每 100 天改变一次，这意味着有 20 个控制周期。因此，在整个仿真时间框架内，我们总共有  $9 \times 20 = 180$  个控制参数。注射率的范围在 1500 至 6500bbl/天之间（238 至

1033 米之间)<sup>3)</sup>。这是一个非常大的范围，良好的操作与通常认为的 ROM，。生产 BHPs 的范围为 3770 至 3988psi (260 至 275 巴之间) 。[19 20]

培训和测试运行的控件指定如下。对于每个注入井，我们随机采样，从 2000 到 6000bbl/day 之间的均匀分布，基线注入速率，然后，在每个控制期间，我们在范围[-500, 500]bbl/day 上均匀采样一个扰动  $q_f$ 。然后规定控制期的速率为  $q_{W^{ASE}+QW}$ 。每个控制步骤的生产者 BHPs 在[3770, 3988]psi 范围内均匀采样。对于生产井，没有基线必和必拓，从控制步骤到控制步骤的设置是不相关的。这种指定注入速率的方法导致了广泛的溶液行为（例如饱和分布），因为井喷在运行中有很大的变化。这种处理也避免了平均效应，如果注入速率不引用基线值  $q$ ，可能会发生 ASE。使用此过程生成的测试用例的井规范如图所示。7.

我们执行 300 次训练模拟来构造 E2CROM，除非另有说明。正如以前的论文（例如，）所讨论的，如图所示的井表类型。[19]7 旨在代表在优化程序期间评估的井控概况，其中的目标是最大限度地提高石油产量或盈利能力，或尽量减少环境影响或某种程度的风险。最后，我们注意到 E2C 潜在空间的维数  $L_z$  设置为 50。

#### 4.2. 测试用例 1 的结果

在本节中，我们将给出特定测试用例的详细结果。这些包括井量(注入 BHPs 和生产速率)和全球量（压力和饱和场）。测试用例 1 的注入速率和 BHP 配置文件显示在图中。7. 在这里，我们显示了四个注入井的水率（图）。7(A)-(D)，以及五口生产井的 BHPs（图 1）。7(e)-(i)。

我们现在评估基于深度学习的 ROM 在这个测试用例中的性能。首先考虑了全局饱和场的时间演化。图 18 显示 200 天的饱和场。在图中。给出了全阶饱和场(也称为高保真解 HFS)，并给出了相应的 E2CROM 饱和结果。8(b)。色标表示水饱和度（因此红色表示水）.. 图之间的密切视觉一致性。8(A)和(B)表明，基于深度学习的 ROM 能够为这一数量提供准确的结果。两种解决方案之间的一致性水平在图中被量化。其中显示 HFS 和 ROM 解决方案之间的差异。请注意，这里的色条刻度与图中的颜色非常不同。8(a)和(b)。ROM 和 HFS 结果之间的误差很小。

为了更好地量化 E2CROM 的预测能力，我们引入了“最近训练运行”的概念。我们使用这个术语来表示执行的 300 次训练运行中最接近（在特定意义上）测试用例的特定训练运行。测试运行与每个训练运行之间的“距离”是根据归一化控制参数向量之间的欧几里德距离来量化的，而“最近的训练运行”(k\*)是最小距离的训练运行。具体而言，

$$k^* = \underset{k}{\operatorname{argmin}} \|U_0 \mathbf{u}_k - \mathbf{u}_t\|_2, \quad (33)$$

其中  $k=1300$ ，表示训练运行的指数， $U_0 \mathbf{u}_k$  代表

$\mathbf{u}_t$  代表

测试运行的控制输入， $U_0 \mathbf{u}_k$  指示训练运行  $k$  的控制输入，上标 0 指定控件中的归一化压力和速率，如 Eq 中的归一化。31.

33 提供了一个非常近似的指标‘最近的训练运行’。这一定义具有简单性的优点，尽管更多的参与（和计算要求）评估将有望提供更密切的培训解决方案。然而，这将需要采用一种类似于 POD-TPWL 中使用的“点选择”过程的方法。这将需要计算每次训练运行的多个时间步骤的距离。由于我们在这里有 300 次培训（而不是 3 或 5 次 POD-TPWL），这可能变得非常耗时。因此，我们应用了 Eq 中定义的简单方法。[19] 33 认识到可以设计更复杂（而且可能更准确）的程序。

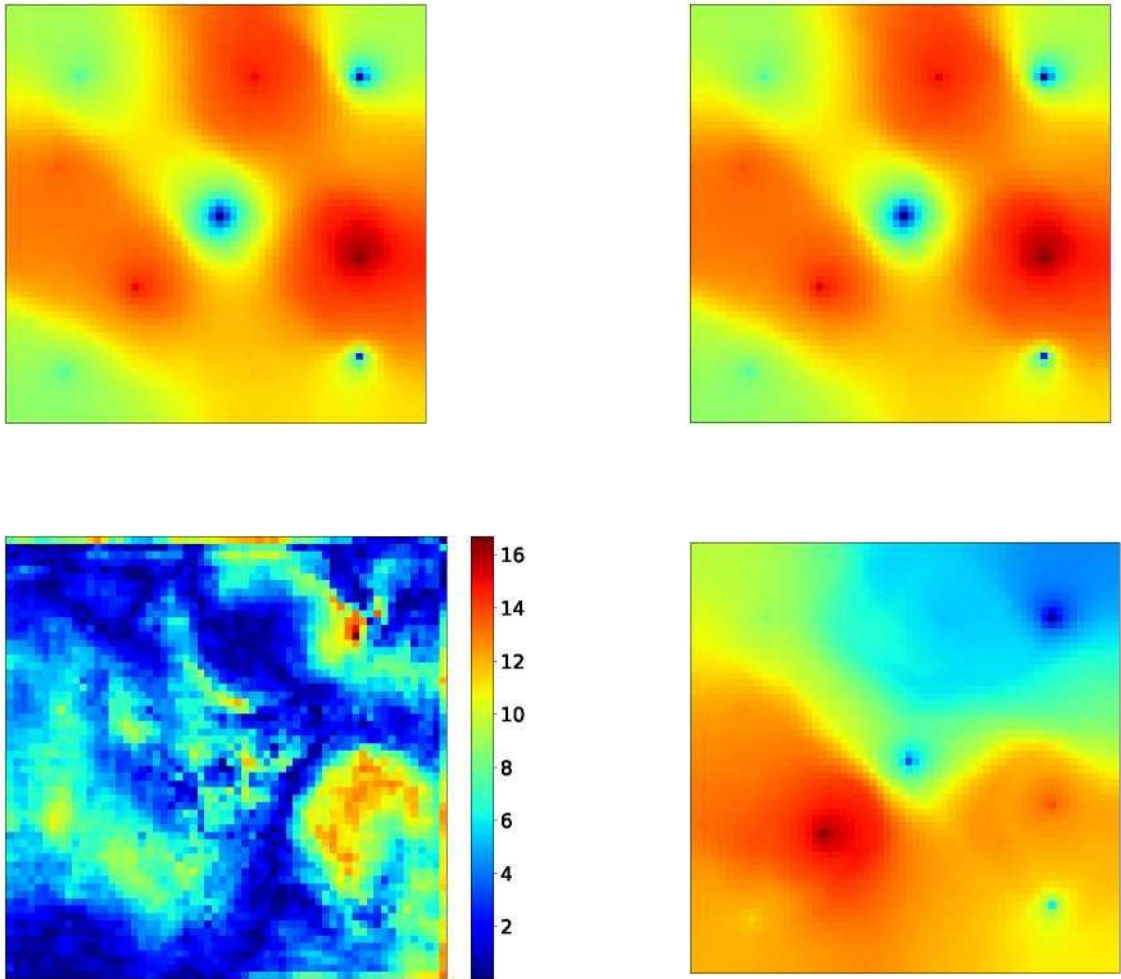
我们现在回到全球饱和的结果。图 1 8(D)显示了“最接近的训练运行”（如我们刚才在高保真度下描述和模拟的那样确定）与测试用例饱和场之间的差异。色条刻度与图中相同。8(c)。应用基于深度学习的 ROM 的优点是明显的，通过比较图。8(c)和(d)。更具体地说，图中的错误。8(C)大约比图中所示的差异小一个数量级。8(d)。

无花果。9 和 10 显示类似的结果，饱和在 1000 天和 1800 天。饱和场随时间的演化是明显的，基于深度学习的 ROM 解决方案（图 1）。9(B)和 10(B)再次被视为与 HFS 密切一致（图 1）。9(a)和 10(a)。误差映射在图中。9(c)和 10(c)进一步量化基于深度学习的 ROM 的准确性。与“最近的训练运行”和 HFS 之间的差异图相比，这些误差是相当小的，如图所示。9(d)和 10(d)，这进一步说明了 ROM 的有效性。

请注意，在 ROM 解中，我们确实观察到饱和羽流中的一些局部（非物理）极值。这是一个小问题，因为不同的地图显示了 ROM 和高保真解之间的小的总体差异。然而，在某些情况下，这可能引起关切。这方面的一个可能的补救办法是在基于物理的损失函数中添加一个项，从而惩罚与控制流方程不一致的饱和局部极值。

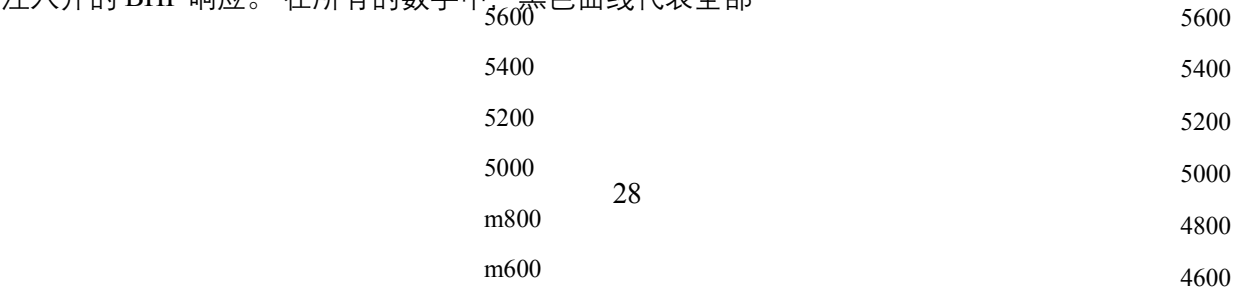
在特定时期的全球压力场也令人感兴趣。在图中。(a)和(b)我们 11

在 1000 天内显示 HFS 和 ROM 压力解决方案。密切的视觉一致表明，基于深度学习的 ROM 能够提供准确（和平滑）的压力预测。图 1 11(c)显示 ROM 解的误差图，我们看到误差确实很小。这



些错误远小于“最近的训练运行”的错误，如图所示。 11(d)。

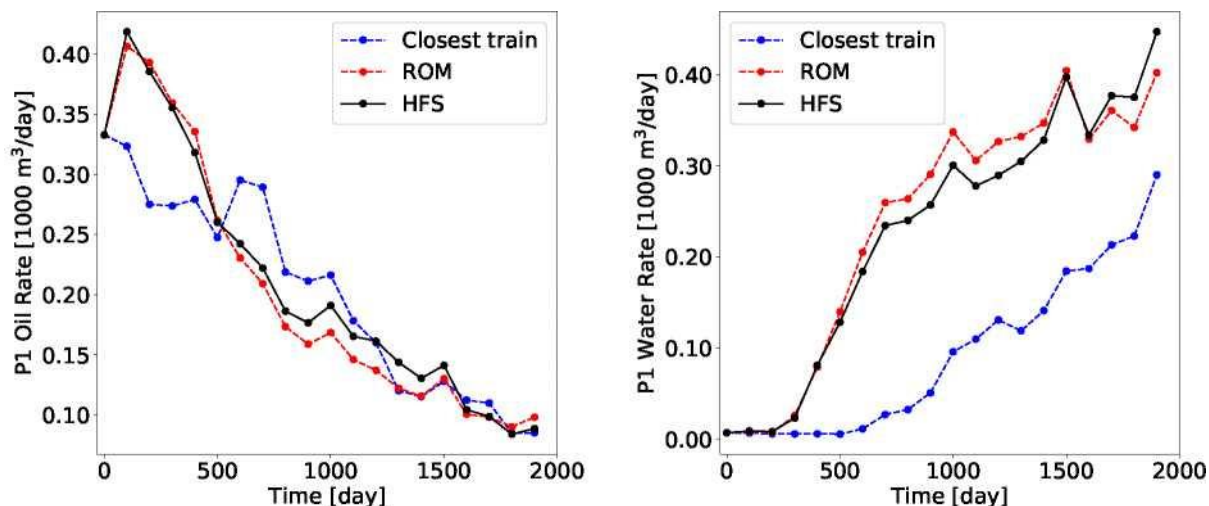
在许多地下水流应用中，井响应是主要的兴趣。对这些数量的 E2CROM 预测现在将被评估。由于在这个问题中，我们指定了注入率和生产井 BHP，利息的数量是注入井 BHP 和油水产量。图中显示了 WellsP1 和 P2 的相流量，它们是对总场生产贡献最大的井。图 1 12 13 14 显示了所有四口注入井的 BHP 响应。在所有的数字中，黑色曲线代表全部



(a)高保真溶液

(b)ROM 溶液(ROMtest)

顺序 (参考) HFS, 红色曲线是基于深度学习的 ROM 结果, 蓝色曲线是“最近的训练运行”的结果。在 ROM 和 HFS 结果之间的高度精度是一致的。通过对损失函数 C 中的井块量的额外加权, 提高



了这些基本数量的一致性水平<sub>p</sub> (见方程式。28)

附录 B 我们给出了另外两个例子（测试用例 2 和 3）的结果。这些结果证实了我们在这里的观察，即基于深度学习的 ROM 能够准确地预测全局饱和和压力分布以及感兴趣的井量。

最后，我们讨论了高保真和 ROM 运行的时间.. 高保真测试用例每个需要 60 秒来模拟在具有双 Intel Xeon CPU (24 核) 的节点上使用 AD-GPRS。在配置约 1GB 内存的特斯拉 V100 GPU 节点上, 大约 2 秒内, 可以使用 E2CROM 评估整批 100 个测试用例。一个直接 (虽然简单) 的比较表明加速因子为 3000。

### 4.3. 所有测试用例的结果和误差测量

在本节中，我们评估了 100 个测试用例的完整集成的 ROM 结果的准确性。我们首先考虑现场累积油水产量，这是由

$$QJ = \int_{t_p}^{\infty} \frac{f(q)}{w} dt \quad (34)$$

这里  $j=0$ ,  $w$  表示相位,  $n_p$  是生产井的总数,  $t$  指定总模拟时间,  $Q_w^j(t)$  表示时间步长  $t$  的相位  $j$  的流体速率。



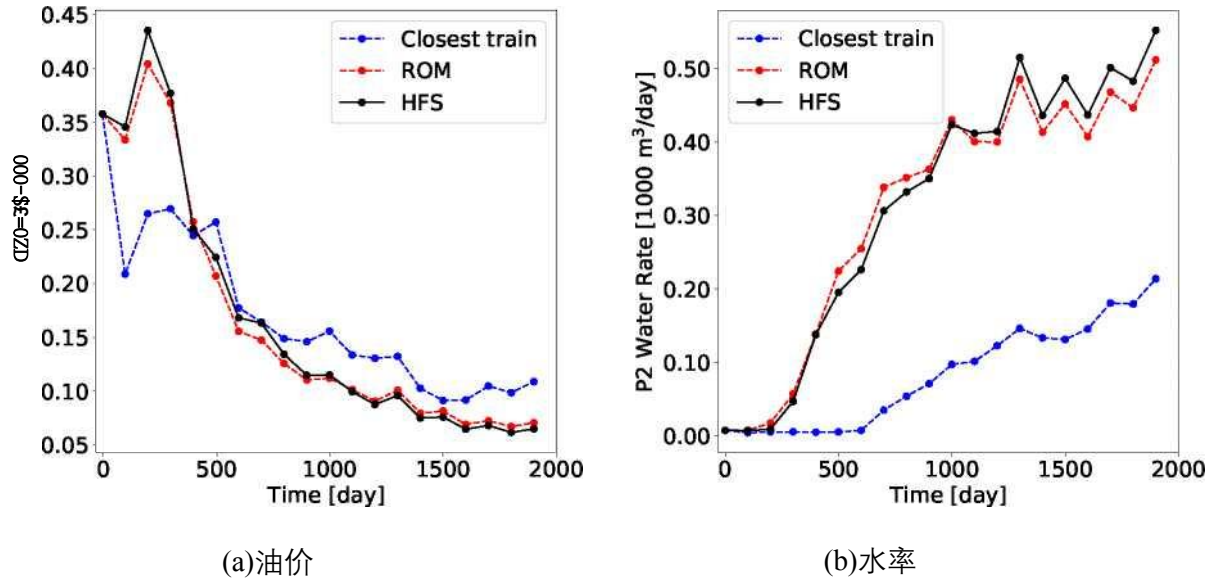


图 13：测试用例 1：P2 井的生产速率

15 在图中, 我们给出了  $Q$  的交叉图。和问题  $w$  对于 100 个测试用例的 HFS 和 ROM 解决方案。每个地块上的三个  $x$  表示测试用例 1、2 和 3 的结果。显然, 这些案件在  $Q_o$  和  $Q$  方面有很大的不同。在这个意义上, 跨越了 100 个测试用例的范围。我们看到, 这两个地块的点落在  $45^\circ$  线附近, 这表明我们的 ROM 解决方案与 HFS 密切一致。结果为  $Q_w$  15 在图中。(b) 表明 ROM 一贯低估累积产水量。然而, 预测不足相对较小, 因为这一情节所涵盖的范围很窄。最后, 请注意, 累积石油生产的轻微过度预测在图中是明显的.. 15(a)。

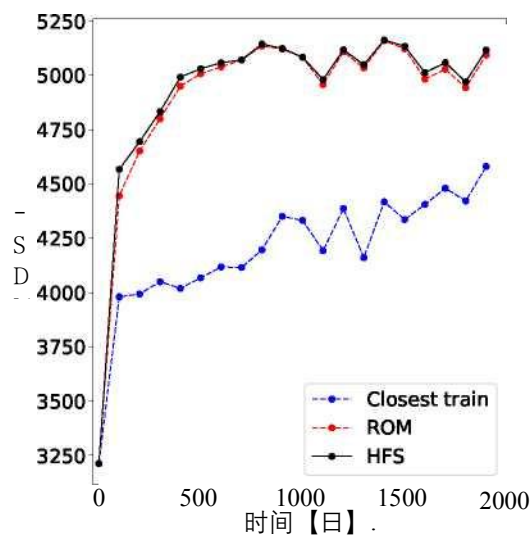
我们现在介绍了一些误差度量, 这些误差度量将用于评估 E2CROM 的一般性能。这些错误度

$$e_P = \frac{\sum_{j=0}^n |Q_{ROM}^{(t)} - F_s^{(t)}|}{\sum_{j=0}^n |Q_{FS}^{(t)}|} DT \quad (35)$$

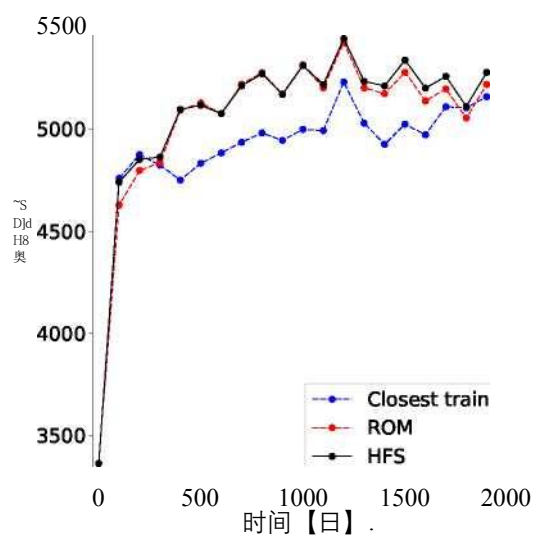
量遵循那些使用。单个生产井  $p$  的产油或产水率的相对误差定义为: [20]

其中  $j=o, w$  是流体相,  $q_p(t)$  是生产井  $p$  的时间  $t$  的产油率或产水率, 下标 HFS 和 ROM 表示高保真和 ROM 结果,  $T$  是我们定义了总体生产速率的误差,  $E_r$  以  $E$  和  $EW$  为所有生产井, 如:

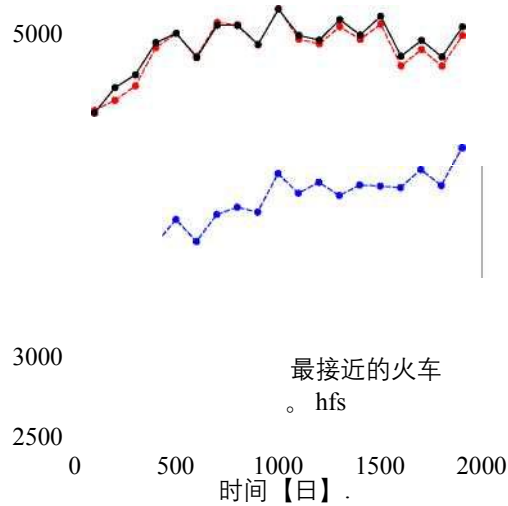
$$e = nZ^{(e)}_o + e_w, \quad (36)$$



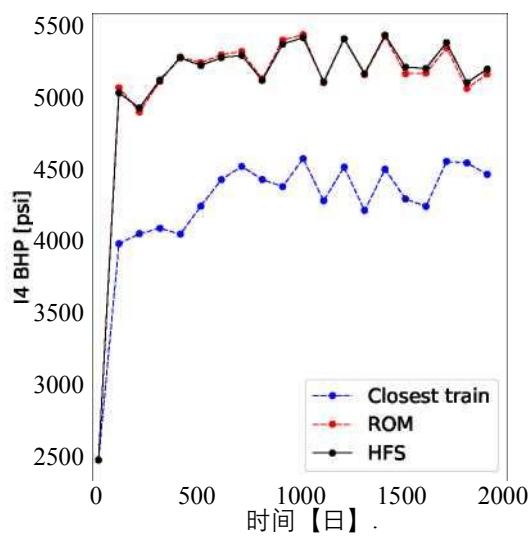
(a)I1



(b)第二类



(c)第三节



(d)第四节

图 14：测试用例 1：注射 BHPs

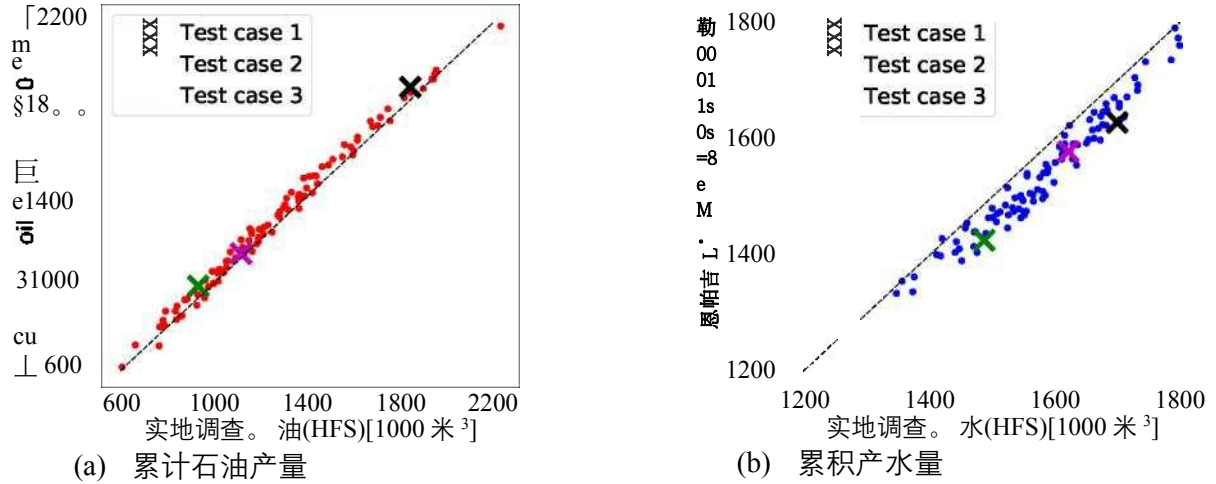


图 15：所有 100 个测试用例的累积油水产量

地点，地点  $p$  是生产井的总数。同样，对于单个注入井  $I$ ，注入 BHP 的相对误差定义为：

$$e_{\text{BHP}}^I = \frac{p_{\text{BHP}}^I(t) - p_{\text{BHP}}^I(t)_{\text{FS}}}{p_{\text{BHP}}^I(t)_{\text{FS}}} \quad (37)$$

在那里  $p_{\text{BHP}}^I(t)$  表示注入井  $I$  时注入 BHP。总体注入井 BHP 误差  $E_{\text{bhp}}$  由：

$$E_{\text{BHP}} = \frac{1}{n} \sum_{i=1}^n e_{\text{BHP}}^i \quad (38)$$

其中  $n$  是注入井的总数。

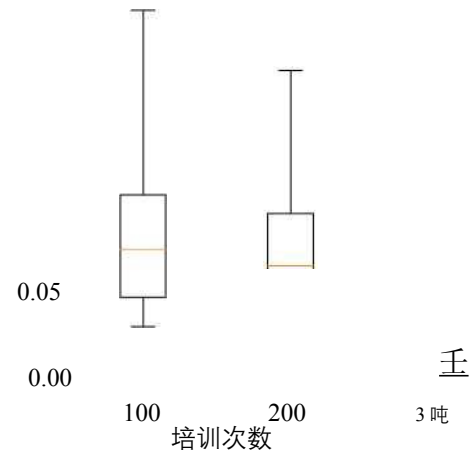
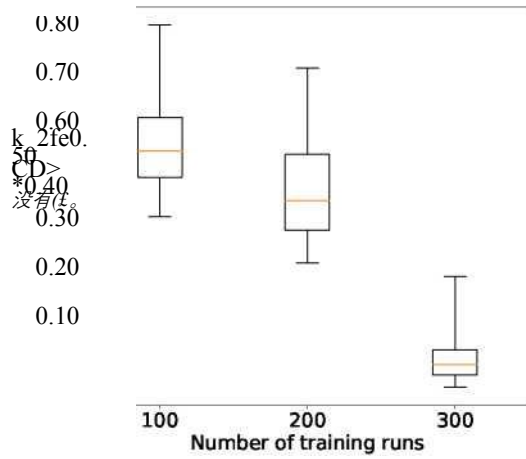
全局数量的错误也是令人感兴趣的。我们定义了全局压力和饱和错误为：

$$e_{\text{global}} = \frac{1}{n} \sum_{k=1}^n \frac{|v_{\text{hfs}}^k - v_{\text{hfs}}^k|^{\text{DT}}}{|v_{\text{hfs}}^k|^{\text{DT}}} \quad (39)$$

在那里  $v_{\text{hfs}}^k$  表示网格块  $k$  (压力  $p$ ) 中感兴趣的全局变量  $k$  或饱和  $S^k$ ), 而  $n$  是模型中网格块的总数。

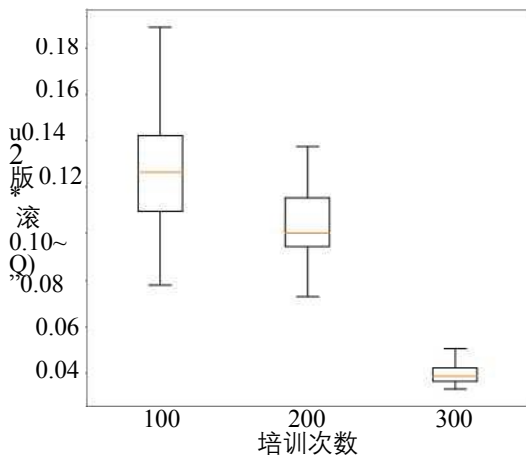
这四个错误量显示为图中的红色点。16. 我们还评估了所有测试用例的“最近的训练运行”的这些错误。在图中，这些点是通过增加“最近的训练运行”（蓝色点）的误差来排序的。测试用例 1、2 和 3 的结果显示在每个图中。我们看到 ROM 误差一直很小，而“最近的训练运行”的误差在许多情况下都很大。有趣的是，ROM 错误似乎不依赖于与“最近的训练运行”相关的错误。这个



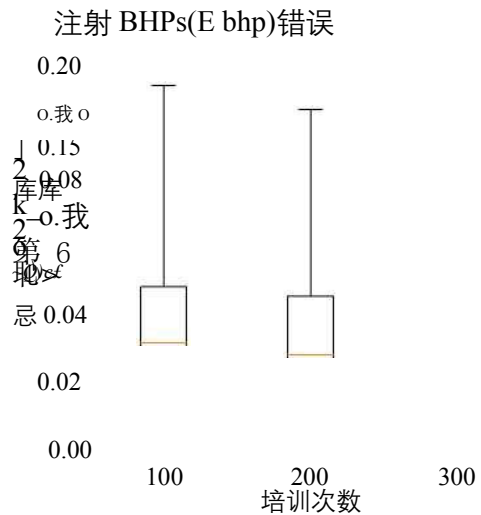


是一个理想的特征，因为它表明在 E2CROM 中具有高度的鲁棒性。

(a) 生产率错误(E<sub>s</sub>)



(c) 饱和场误差(E<sub>S</sub>)



(d) 压力场误差(E<sub>P</sub>)

图 17：不同训练运行次数的 ROM 错误

17 我们现在简要地考虑在 E2CROM 的构建中使用较小数量的训练运行。对于这些情况，我们只给出摘要错误结果。图中显示了上述四个相对错误，在方框图，100，200 和 300 的训练运行。在每个框中，中心橙色线表示中值误差，框的底部和顶部边缘显示第 25 和第 75 百分位数误差。从盒子里延伸出来的“胡须”表示最小和最大的误差。当我们从 200 到 300 次训练运行时，ROM 的准确性有了很大的提高。在今后的工作中，制定确定所需培训次数的方法将是有益的。

由于很难在图中的方框图中显示测试用例 1、2 和 3 的错误，我们将它们呈现在表中。这些结果有 300 次训练。注意 E 的平均值  $17 \text{ l}_{\text{EBHP}}$ ， $E_S$  和  $E_P$  在所有 100 个测试用例中，分别约为 0.14、0.02、0.04 和 0.002。表中显示的三个测试用例的误差值可以看作是在整套测试用例之间的合理分布。有兴趣观察到，这四个

错误似乎没有密切相关的特定测试用例。例如，在测试用例 1 中， $E_p$  在第 98 百分位，而  $E_{bhp}$  在第 15 百分位。

测试用例		$e_r$	$e_{\text{必和必拓}}$	$e_s$	$e_p$
测试用例 1	错误	0.11	0.0054	0.042	0.0033
	是个异教徒	37	15	70	98
测试用例 2	错误	0.19	0.0064	0.050	0.0031
	是个异教徒	96	29	99	95
测试用例 3	错误	0.16	0.012	0.042	0.0017
	是个异教徒	77	84	73	24

表 1：测试用例的平均误差和百分位数

## 5. 结束语

[39]在本工作中，我们介绍了一种基于深度学习的地下流动模拟降阶建模过程。该程序是从现有的嵌入到控制(E2C)程序，虽然我们引入了一些关键的修改相对于配方。从本质上讲，ROM 由自动编码器(AE)和线性过渡模型组成。在我们的 E2C 公式中，一个额外的基于物理的损失函数与数据不匹配损失函数相结合，以提高与控制流方程的一致性。虽然它是基于深度学习的概念和方法，但各种 E2CROM 步骤被证明非常类似于发达的基于物理/数字的 POD-TPWLROM 中使用的步骤。

在我们的大多数评估中，我们在离线步骤中执行了 300 次培训运行。不包括训练模拟的运行时间，离线模型构建需要 10-12 分钟的 ROM 训练使用特斯拉 V100GPU。在考虑的情况下，相对于 AD-GPRS 全阶模拟，在线（运行时）加速为 0(1000)。考虑到离线成本和在线速度，当需要许多（相关）模拟运行时，使用此 ROM 是合适的。生产优化计算、数据同化和不确定性评估都是如此（尽管在本工作中只考虑了一个单一的地质模型）。

基于深度学习的 ROM 在涉及非均质渗透率场的二维油水油藏模拟问题上进行了测试。在测试用例中规定了注入和生产井控制设置中的大变化（相对于训练运行）。共考虑 100 例试验病例。对关键数量感兴趣的 ROM 精度进行了评估，包括注入井 BHP、相生成速率、全球压力和饱和场。在全套测试运行中，E2CROM 被证明是一致准确的。ROM 误差被认为远低于“最近的训练运行”（适当定义）。然而，如果使用 100 或 200 次训练运行而不是 300 次，则错误会增加。

在未来的工作中，E2CROM 应该扩展到更复杂的三维问题，并在现实的案例上进行测试。扩展到三维可以通过用 conv3D 层替换 conv2D 层来接近。该 ROM 可以很容易地与各种优化算法一起用于生产优化，并应评估其在此设置中的性能。重要的是，E2CROM 应该适用于全局和局部优化算法。这与现有的基于 POD 的 ROM 形成了对比，后者只能在更有限的社区中被期望是准确的，因此最适合于本地搜索方法。还有兴趣探讨随着井位变化预测流量响应的潜力。如果这是成功的，ROM 可以应用于井位优化，或结合井位和控制优化问题。为了提高框架在这些更具挑战性的应用中的准确性和鲁棒性，可以评估更复杂的编码器和解码器结构，如密集网。最后，这里使用的自动编码器可以扩展到 VAE 或不确定性自动编码器，以使系统和控制不确定性得到考虑。[53][54]

## 感谢

我们感谢斯坦福大学智能外勤联合会(SFC)为这项工作提供部分资金。我们感谢斯坦福计算地球和环境科学中心(CEES)提供了本研究中使用的计算资源。我们还感谢朱玉科和 AdityaGrover 的有益讨论，感谢 Oleg Volkov 对 AD-GPRS 软件的帮助。

## 附录 A. 嵌入到控制网络体系结构

a. 2 编码器的结构概述见表。编码器有四个编码块的堆栈，分别有 16、32、64 和 128 个滤波器。编码块的详细结构如图所示。a. 18(a)。过滤器的大小是指编码块中的 conv2D 层。批规范层和 ReLU 的尺寸与来自 Conv2D 层的输出的大小一致。编码器还有一个由三个剩余卷积(Res Conv)块组成的堆栈，每个块具有 128 个大小为  $3 \times 3 \times 128$  和步长 1 的滤波器。过滤器大小再次引用 Res Conv 块中 Conv2D 层的过滤器大小。详细结构的 Res Conv 块如图所示。a. 18 (d)。请注意，在输入到密集层之前，来自 Res Conv 块的输出是从维度的三维矩阵(N)中重塑的  $\sqrt{4}$ ,  $n_y$  变成一个长向量。编码器末端的密集层可以作为输入大小为 N 的线性变换处理  $\sqrt{4} \times N_y / 4 \times 128$ ，潜维 1 的输出大小  $\sqrt{4} \times 128$  (第 50=在此审议的案件)。在这里  $\sqrt{4}$  而  $N_y$  是输入图像的高度和宽度，在我们的情况下都是 60。

一层	过滤器编号, 大小和步幅.	输出大小
投入		(N X, Ny, 2)
编码块	3 项中的 16 项x (n)/2, n/2,	3 x2, 跨步 2 16)
块, 编码块, Res Conv	3 项中的 32 项x (Nx/2, Ny/2,	3 x16, 第一 64)
块, Res Conv 块, Res 步	3 个国家中的 64 个 32, 跨步 2 (n)/4,	x 3 x ny/4, 128)
Conv 块, Conv 块,	128 名 3 x3 x 64, 第一步	(N x/4)Ny/4, 128)
Dense.	128 名 3 x3 x 128, 大步 1	(N x/4)Ny/4, 128)
	128 名 3 x3 x 128, 大步 1	(n)/4, ny/4, 128)
	128 名 3 x3 x 128, 大步 1	(N x/4)Ny/4, 128)

a. 3. 表中总结了解码器的结构，解码器的结构类似于编码器，但组件的顺序相反。解码器由一个密集层、三个 RES Conv 块的堆栈、四个解码块的堆栈和一个 Conv2D 层组成。注意，密层的输出是维数 N 的长向量  $\frac{N}{4} \times N_y/4 \times 128$ ，被重塑成一个大小的三维矩阵  $(N/4, N)$ ，在送入 Res Conv 块之前。解码块的详细结构如图所示。(b)。 a. 18

一层	过滤器编号, 大小和步幅.	输出大小
投入		(1z, 1)
很浓		(N x/4x Ny/4x128, 1)
再来一次	128 的 3x3x128, 步幅 1	(N x/4, Ny/4, 128)
再来一次	128 的 3x3x128, 步幅 1	(N x/4, Ny/4, 128)
再来一次	128 的 3x3x128, 步幅 1	(N x/4, Ny/4, 128)
解码块	128 的 3x3x128, 步幅 1	(N x/4, Ny/4, 128)
解码块	64 的 3x3x128, 步幅 2	(N x/2, Ny/2, 64)
解码块	32 的 3x3x64, 大步 1	(N x/2, Ny/2, 32)
解码块	3x3x32 中的 16, 跨步 2	(N X, Ny, 16)
对流 2D	3x3x16 中的 2, 步幅 1	(N X, Ny, 2)

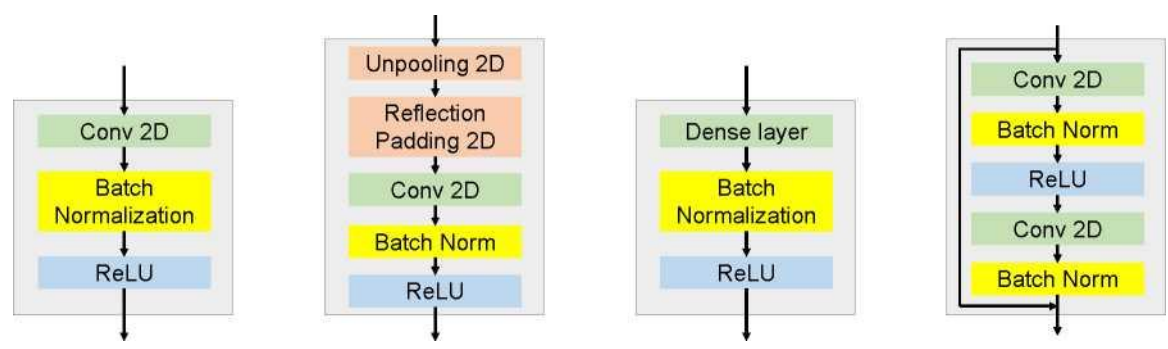
表 A. 解码器的 3：网络体系结构

a. 18 图显示了编码块、解码块、转换块和 Res Conv 块的详细结构。编码块是二维卷积层(Conv2D)、批归一化(BathNorm)层和校正线性单元(ReLU)的顺序组合。通过调整卷积 2D 层的步长（例如，表中编码块的两个步长)来实现空间维数(下采样)的减小。解码块有一个 2D 解冷层，这增加了大小 a. 2)



在输入中，通过重复数据的行和列，二维反射填充层，它还通过在数据的两个维度上填充边界来增加输入的大小，即 Conv2D 层、批处理 Norm 层和 ReLU。空间尺寸的增加（上采样）是通过二维解冷和二维反射填充层实现的。

RES Conv 块的结构与 RESNET 非常相似，其中创建了一个标识映射来绕过非线性层。对于 Res Conv 层，非线性层只需要学习  $H(X) := F(X) - x$  的残差映射，而不是直接映射输入  $x$  到目标函数  $F(X)$ 。在极端（最坏）情况下( $F(X)=X$ )，这将是零，并保证更深的神经网络将达到更高（或至少相等）精度相对于其较浅的对应。非线性层遵循 conv2D 批处理 Norm-ReLU 的标准结构.. 转换块具有密集批处理 Norm-ReLU



(a)编码块                      (b)解码块                      (c)Trans。 封锁                      (d)修院大楼  
的体系结构，其中对于测试的情况，密集层的尺寸设置为 200。

图 A. 18：块嵌入到控制模型中

**附录 B 附加测试用例结果**

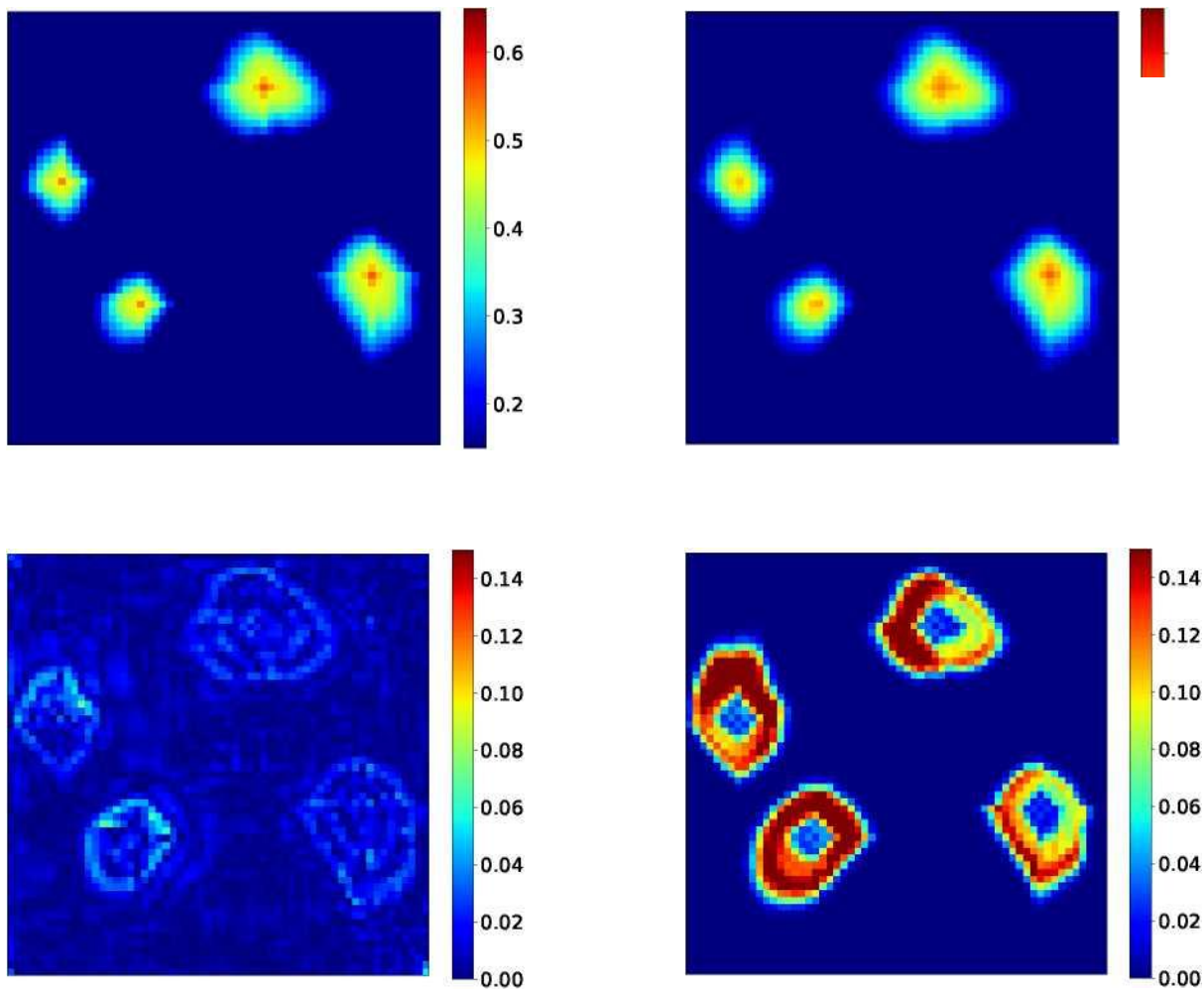
我们现在为测试用例 2 和 3 提供 E2CROM 结果。这些情况下感兴趣的数量的误差如图 16 和表所示。我们在这里的描述很简短，因为这些结果与本节讨论的结果非常相似。 15 1 4

*附录 B 1. 测试用例 2 的结果*

b. 19 无花果，B。 20 和 B. 21 显示饱和场在 200， 1000 和 1800 天， 和图. b. 在这种情况下， 22 显示了 1000 天的压力场。饱和度的演变表明这里的扫描与测试用例 1 的扫描有点不同。具体来说， 比较图. b. 21(b)至图. 10(b)， 我们发现 I2 井（右上）周围的水羽较大， 而 I1（左上）和 I3（左下）周围的水羽较小。 请参阅图. 6 准确的井位。 用于测试的压力图

案例 2 也显示了一个非常不同的模式，与测试案例 1 在 1000 天。差异图如图所示。(c)，B. 20(c)，B. 21(c)和 B. 22(c)表明，ROM 预测对于这些全球量再次准确。b. 19

无花果。b. 23 和 B. 24 显示油井 P1 和 P2 的产油和产水率，这是测试用例 2 中对油田总产量的主要贡献。图 1b. 25 介绍了四口注射井的注射 BHPs。ROM 解决方案再次与高保真解决方案密切



一致，并且观察到这些解决方案与大多数井量的“最近训练运行”之间的显著差异。

图 B 19：测试用例 2：200 天的饱和场

附录 B 2. 测试用例 3 的结果与测试用例 3 饱和度的演化类似的结果如图所示。b. 26，b. 27 和 B. 28，1000 天的压力场如图所示。b. 29. 水库

## 参考资料

- [1] 卡尔伯格 K., Bou-Mosleh, C.Farhat, 通过最小二乘 Petrov Galerkin 投影和压缩张量近似的有效非线性模型约简, 国际工程数值方法杂志 86 (2)(2011) 155-181。
- [2] M.J.Zahr, P.Avery, C.Farhat, 一种基于多级投影的结构和固体力学非线性动态多尺度问题模型阶减框架, 国际工程数值方法杂志 112 (8)(2017) 855-881。
- [3] D.Amsallem, M.J.Zahr, C.Farhat, 基于局部约简阶基的非线性模型阶减, 国际工程数值方法杂志 92 (10)(2012) 891-916。
- [4] 卡尔伯格 K., C.Farhat, J.Cortial, D.Amsallem, 非线性模型约简的 GNAT 方法: 计算流体动力学和湍流的有效实现和应用, 计算物理学杂志 242 (2013) 623-647。
- [5] S.Chaturantabut, D.C.Sorensen, 通过离散经验插值进行非线性模型约简, SIAM 科学计算杂志 32 (5)(2010) 2737-2764。
- [6] 尹 S., Z.M.Alghareeb, J.R.Williams, 地下流动模拟的超减阶模型, SPE 杂志 21 (06)(2016) 2128-2140。
- [7] 杨 Y., Ghasemi, E.Gildin, Y.Efendiev, V.Calo, 用 POD-DEIM 模型还原快速多尺度油藏模拟, SPE 杂志 21 (06)(2016) 2141-2154。
- [8] Y.Efendiev, E.Gildin, Y.Yang, 异构多孔介质流动的在线自适应局部-全局模型约简, 计算 4 (2)(2016) 22。
- [9] 谭 X., E.Gildin, H.Florez, S.Trehan, Y.Y.Yang, N.Hoda, 基于轨迹的 DEIM(TDEIM)模型约简应用于储层模拟, 计算地球科学 23 (1)(2019) 35-53。
- [10] 蒋 R., L.J.Durlofsky, 地下流动模拟 GNAT 降阶模型的实现和详细评估, 计算物理杂志 379 (2019) 192-213。
- [11] H.Florez, E.Gildin, 超低渗透 ULP 油藏耦合流动和地质力学的模型阶减小(SPE 论文 193911), 载于: SPE 油藏模拟会议, 美国得克萨斯州加尔维斯顿, 2019 年。
- [12] 肖 D., F.方, C.Pain, G.Hu, 基于 RBF 插值的 Navier-Stokes 方程的非侵入性降阶模型, 国际流体数值方法杂志 79 (11)(2015) 580595。
- [13] W.Kostorz, A.Muggeridge, M.Jackson, A.Moncorge, 用于重建饱和分布的非侵入性降阶模型(SPE 论文 193831), 载于: SPE 水库模拟会议, 加尔维斯顿, 得克萨斯州, 美国, 2019 年。
- [14] M.Rewienski, J. 怀特, 一种用于非线性电路和微加工器件模型降阶和快速模拟的轨迹分段线性方法, IEEE 集成电路和系统计算机辅助设计交易 22 (2)(2003) 155-170。
- [15] 卡多索 M., L.J.Durlofsky, 用于地下流动模拟的线性化简阶模型, 计算物理学杂志 229 (3)(2010) 681-700。
- [16] j. 他, J.S&trum, L.J.Durlofsky, 地下流动模拟的增强线性化约简阶模型, 计算物理杂志 230 (23)(2011) 8313-8341。
- [17] j. 他, L.J.Durlofsky, 利用轨迹分段线性化进行组合模拟的降阶建模, SPE 杂志 19 (05)(2014) 858-872。
- [18] j. 他, L.J.Durlofsky, 基于 POD-TPWL 的降阶地下流动模型的约束减少程序, 国际工程数值方法杂志 103 (1)(2015) 1-30。
- [19] Z.L.J.Durlofsky, CO<sub>2</sub> 储存操作的降阶建模, 国际温室气体控制杂志 68 (2018) 49-67。
- [20] Z.L.Jin, T.Garipov, O.Volkov, L.J.Durlofsky, 耦合流体力学问题的降阶建模(SPE 论文 193863), 载于: SPE 水库模拟会议, 加尔维斯顿, 得克萨斯州, 美国, 2019 年。
- [21] S.Trehan, L.J.Durlofsky, Trajectory 分段二次降阶地下流动模型, 应用于 PDE 约束优化, 计算物理学杂志 326 (2016) 446-473。
- [22] S.Canchumuni, A. A.Emerick, M.A.C.Pacheco, 历史匹配通道相模型使用集成平滑与深度学习参数化, 在: 第 16 届欧洲石油回收数学会议, 西班牙巴塞罗那, 2018 年。
- [23] E.Laloy, R.Herault, D.Jacques, N.Linde, 利用空间生成对抗性神经网络进行基于训练图像的地质统计反演, 水资源研究 54 (1)(2018) 381-406。
- [24] 刘 Y., 孙伟, L.J.Durlofsky, 一种基于深度学习的历史匹配复杂模型地质参数化, 数学地质学 (2019) <https://doi.org/10.1007/s11004-019-09794-9>。
- [25] 刘 Y., L.J.Durlofsky, 历史匹配复杂储层模型的多层次策略和地质参数化(SPE 论文 193895), 载于: SPE 水库模拟会议, 美国得克萨斯州加尔维斯顿, 2019 年。

- [26] 约翰逊 J., A.Alahi, F.F.Li, 实时风格转移和超分辨率的感知损失, 载于: 欧洲计算机视觉会议, 阿姆斯特丹, 荷兰, 2016 年。
- [27] 朱 Y., N.Zabaras, 贝叶斯深卷积编解码网络的代理建模和不确定性量化, 计算物理杂志 366 (2018) 415-447。
- [28] 莫 S., 朱永强, 扎巴拉斯, 史文生, 吴俊华, 用于异构介质中动态多相流不确定性量化的深卷积编解码网络, 水资源研究 55 (1)(2019) 703-728。
- [29] 李 K., 卡尔伯格, 非线性流形上动力系统的模型简化使用深卷积自编码器, ar Xiv 预印 ar Xiv : 1812.08373。
- [30] J.N.Kani, A.H.Elsheikh, 利用深残余递归神经网络对地下多相流模型进行降阶建模, 多孔介质传输 126 (3)(2019) 713-741。
- [31] 张 J., 张世伟, Y.Efendiev, E.Gildin, E.T.Chung, 水库模拟深度模型约简模型学习(SPE 论文 193912), 载于: SPE 水库模拟会议, 加尔维斯顿, 得克萨斯州, 美国, 2019 年。
- [32] 王 Z., 肖德华, 方福芳, R.Govindan, C.Pain, 郭永国, 利用深度学习对降阶流体动力学系统进行模型识别, 国际流体数值方法杂志 86 (4)(2018) 255-268。
- [33] F.A.Gers, J.Schmidhuber, F.康明斯, “学会忘记: 与 LS TM 的持续预测”, 载于: 第九届人工神经网络国际会议, 爱丁堡, 英国, 1999 年。
- [34] F.J.Gonzalez, M.Balajewicz, 用于学习流体系统低维特征动力学的深卷积递归自编码器, ar Xiv 预印 ar Xiv : 1808.01346。
- [35] D.P.Kingma, M.Welling, 自动编码变分 Bayes, ar Xiv 预印 ar Xiv : 1312.6114。
- [36] M.Raissi, Perdikaris, G.Karniadakis, 物理信息神经网络: 解决非线性偏微分方程正反问题的深度学习框架, 计算物理学杂志 378 (2019) 686-707。
- [37] 朱 Y., N.Zabaras, P.-S.Koutsurelakis, Perdikaris, 物理约束深度学习, 用于高维代理建模和不含标记数据的不确定性量化, ar Xiv 预印 ar Xiv : 1901.06314。
- [38] D.P.Kingma, P.Dhariwal, Glow: 具有可逆 1x1 卷积的生成流, 载于: 神经信息处理系统的进展, 加拿大蒙特利尔, 2018 年。
- [39] M.Watter, J.Springenberg, J.Boedeker, M.Riedmiller, 嵌入控制: 一种从原始图像控制的局部线性潜在动力学模型, 载于: 神经信息处理系统的进展, 加拿大蒙特利尔, 2015 年。
- [40] 周 Y., 并行通用油藏模拟与耦合油藏模型和多层井, 博士。D.论文, 斯坦福大学, 2012 年。
- [41] O.Ronneberger, P.Fischer, T.Brox, U-net: 生物医学图像分割的卷积网络, 载于: 医学图像计算和计算机辅助干预国际会议, 德国慕尼黑, 2015 年。
- [42] D.Eigen, C.Puhrs, R.Fergus, 使用多尺度深度网络从单个图像中预测深度地图, 载于: 神经信息处理系统的进展, 加拿大蒙特利尔, 2366-2374, 2014 年。
- [43] 勒村 Y., L.Bottou, Bengio, P.Haffner, 基于梯度的学习应用于文档识别, IEEE86 (11)(1998) 2278-2324 会议记录。
- [44] S.Ioffe, C.Szegedy, 批量归一化 通过减少内部协变量移位来加速深层网络训练, ar Xiv 预印 ar Xiv :1502.03167。
- [45] X.Glorot, A.Bordes, Y.Bengio, 深度稀疏整流神经网络, 载于: 第十四次人工智能和统计国际会议, Ft.Lauderdale, 佛罗里达, 美国, 2011 年。
- [46] X.Glorot, Y.Bengio, 理解训练深度前馈神经网络的困难, 第十三届人工智能和统计国际会议, 意大利撒丁岛, 2010 年。
- [47] 何 K., 张晓华, 任世荣, J.孙, 图像识别的深度残差学习, 载于: IEEE 计算机视觉和模式识别会议, 拉斯维加斯, 内华达州, 美国, 2016 年。
- [48] 黄 G., 刘泽, 范德马顿, K.Q.Weinberger, 紧密连接的卷积网络, 载于: 计算机视觉和模式识别, 檀香山, 美国, 2017 年。
- [49] D.W.Peace man, 数值油藏模拟中井块压力的解释, SPE 杂志 18 (03)(1978) 183-194。
- [50] D.P.Kingma, J.Ba, Adam: 一种随机优化方法, ar Xiv 预印 ar Xiv : 1412.6980。
- [51] F.Chollet 等人, Keras, 2015 年 <https://github.com/fchollet/keras>, 。

- [52] M.Abadi 等人, TensorFlow : 异构系统上的大规模机器学习, 2015 年 <https://www.tensorflow.org/>, 。
- [53] S.Jegou, M.Drozdzal, D.Vazquez, A.Romero, Y.Bengio, 一百层提拉米苏 : 用于语义分割的完全卷积密度集, 载于 : IEEE 计算机视觉和模式识别研讨会会议, 檀香山, 美国, 2017 年。
- [54] A.Grover, S.Ermon, 不确定性自动编码器 : 通过变分信息最大化学习压缩表示, ar Xiv 预印 ar Xiv :1812.10539。