基于深度学习的地下径流的降阶模型模拟

# 摘要

本论文提出了一种新的基于深度学习的降阶模型（ROM）框架，该框架可用于地下径流模拟。降阶模型基于现有的嵌入控制（E2C）框架，而且包括一个能将系统投影到低维子空间的自动编码器，和一个低维状态下逼近系统状态演化的线性转换模型。除了原始E2C框架中考虑的数据失配的损失函数外，我们还引入了一种基于物理的损失函数，该函数对与控制流方程式不一致的预测进行了惩罚。该损失函数也作了修改，来强调感兴趣的关键井数量的准确性（例如，流体的生产率产量的井）。E2C ROM框架与现有的ROM、POD-TPWL框架非常相似。ROM和POD-TPWL框架已被广泛开发用于地下径流模拟。新的ROM 应用于非均质油藏的油水流动，由9口井在时变控制规格下运行驱动。该框架在离线阶段总共进行300次高保真训练模拟，每一次网络训练需要在Tesla V100 GPU节点上工作10-12分钟。在线（运行时）计算相对于满阶模拟实现了0（1000）的加速。本文给出了大量的随着井控大范围变化的测试用例结果。对于特定时间的整体饱和度和压力场，以及随时间变化的注入和生产井响应， ROM可以实现准确的预测。当使用100或200（而不是300）次训练构建E2C ROM时，误差会增加。

**关键词：油藏模拟，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**

# 引言

油藏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，油藏模拟的计算成本可能很高，特别是当使用高度解析模型时，计算成本可能会更高。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模型程序。该降阶模型程序遵循嵌入控制框架，由线性过渡模型和自动编码器(AE，也称为编解码器)组成，采用编解码器结构，通过构造低维表示的映射来实现降维。AE组件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。E2C程序旨在预测油水油藏模拟问题中的关键井数量，如随时间变化的产量、注入速率或井底压力(BHPs)，以及全局压力场和饱和度场。

ROM方法近年来受到了广泛的关注。这些过程通常包括离线（训练时间)组件，这个组件实现训练的运行并处理和保存相关的解决方案信息，也包括在线(测试时间)组件，这个组件执行新的(测试）运行。一种流行的方法类是基于正交分解(POD)的ROM，其中POD用于实现在线计算中解未知数的低维表示。这些方法要求将方程组投影到低维（这种投影也称为约束约简），这一步中通常使用Galerkin投影和最小二乘Petrov- Galerkin投影这两种方法。

还需要对解的非线性进行处理，并且在基于POD的ROM文章中有很多的处理方法。一种有效的方法是具有近似张量或GNAT的高斯-牛顿法，这种方法也使用了进行状态约简的POD和最小二乘Petrov-Galerkin投影。 GNAT 由Carlberg 等人开发[1]，后来被用于结构和固体力学[2]、机电学[3]和计算流体力学[4]等领域。GNAT 是离散经验插值方法(DEIM)的推广[5]，GNA和POD-DEIM这两种方法已应用于涉及地下水流模拟的许多研究中[6,7,8,9,10,11]。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，得到结果的过程被称为 POD-RBF方法[12,13]。最初由 Rewienski和White引入的轨迹分段线性化方法，通过围绕“附近”训练解进行线性化[14]。POD-TPWL已广泛应用于包括油水、油气组成、CO2储存和耦合流动地质力学系统的地下径流模拟[15,16,17,18,19,20]。Trehan和Durlofsky[21]扩展了POD-TPWL方法，使它包括一个二次项，这个二次项给出了一个轨迹分段二次(POD-TPWQ)过程。

深度学习在图像处理中最近的成功使得利用深层神经网络进行地下径流模型的算法快速发展。这些方法已应用于地质参数化、不确定性量化和代理/降阶模型。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]通过使用生成对抗性网络(GAN)实现了类似的目标，其中通过训练两个对抗性神经网络（称为生成器和鉴别器）来确定高维投影。 Liu 等人[24]刘和Durlofsky[25]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“快速神经风格迁移”[26]算法来表示以多点空间统计为特征的复杂地质模型，并证明了这种方法能够更有效地进行数据同化。Zhu和Zabaras [27]将代理模型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]扩展了该模型用来处理多相流问题，并通过引入其他物理约束进一步改善了性能。

涉及在ROM中使用深度学习技术的最新发展表明，此类方法具有巨大的潜力。Lee和Carlberg [29]通过用AE代替POD引入了一种改进的GNAT程序。将所得方法应用于一维动态Burgers方程和二维准静态化学反应流动问题，测试运行与训练运行中的边界条件不同。Kani和Elsheikh [30]开发了一种深度残差递归神经网络（DR-RNN）程序，该程序使用RNN来近似POD-DEIM过程中控制方程的低维残差函数。然后，将所得的ROM应用于一维油水问题，其测试过程中的孔隙率分布与训练过程的孔隙率分布有关。张等[31]使用一个全连接网络替换POD-DEIM过程中的牛顿迭代，将该方法用于预测二维油水问题中的井响应，其中测试运行中井的井控和渗透率场的组合与训练模拟的不同。尽管相对于“标准”实现，上述所有方法的准确性都得到了提高，但是所有这些开发都在现有的ROM设置之内，即没有人采用端到端的深度学习框架。

其他研究人员已经开发了与现有方法不同的ROM方法。例如，Wang等 [32]使用长短期记忆（LSTM）RNN [33]来近似由POD构造的低维子空间中的流体动力学。随后，Gonzalez和Balajewicz [34]用VAE [35]替换POD步骤来进行低维表示。但是，这两种方法都适用于相对简单的问题，在线和离线模拟运行之间的唯一区别是系统的初始条件（边界条件相同）。在地下流动方程中，井表现为局部的源/汇项，本质上是“内部”边界条件。对于在石油生产优化及相关领域中使用的ROM，在离线计算和在线计算之间改变油井设置的能力是一项基本功能。因此，以上实现方式可能不适用于这些问题。另一个潜在的局限性是，这些过程纯粹是数据驱动的，没有考虑基本的控制方程式。这可能会导致解决方案在视觉上吸引人但在物理上不现实。

许多方法在深度神经网络中加入了物理约束。这些过程的名称不同，但通常具有相同的关键思想。Raissi等[36]引入了一种使用物理知识的深度学习框架（后称物理信息神经网络或PINN），该框架使用了紧密连接的前馈神经网络。在PINN中，与控制性偏微分方程（PDEs）相关的残差函数被引入到神经网络的损失函数中。朱等[37]将此PDE约束概念扩展到基于流的生成模型（GLOW [38]），并使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替代模型。沃特等[39]在机器人计划系统的背景下提出了一个嵌入到控制（E2C）的框架，该框架使用直接的传感数据（图像）和时变控制作为输入来预测系统状态的演变。E2C框架结合了VAE和线性转换模型，VAE既可以用作将系统状态投影到低维子空间的推理模型，又可以用于完整地重构预测结果的生成模型。后者根据时变控制输入来近似低维状态的演化。

在本文中，我们基于E2C模型[39]和上述基于物理知识的处理方法，开发了一个地下径流系统的降阶模型的深度学习框架[36 37]，并介绍了现有E2C模型的两个关键修改。第一个修改是为了提高确定性测试用例的准确性，我们将VAE简化为AE。第二个修改是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，并增强井数据的一致性。后一个修改对于提高感兴趣的井数量的准确性具有重要意义，这在采油优化程序中至关重要。由于我们正在考虑的是一个带有标记数据（输入对和输出对）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[36]和朱等人[37]的模型区分开来，后者需要访问PDE残差来计算训练过程中的损失函数。在许多实际环境中，基础偏微分方程的残差值可能无法访问，我们的方法会更适合。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方法很类似，我们将详细讨论这两种方法之间的关系。

本文的安排如下。在第2节，我们给出了地下油水流动的控制方程，然后简单介绍了POD-TPWL 的降阶模型方法。在第3节，我们给出了E2C方法，并给出了E2C与POD-TPWL之间的对应关系。在第4节，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。我们还给出了几个关键数量的详细误差评估。在第5节，我们给出了本论文的总结以及在今后工作中的展望。附录A中提供了E2C模型中使用的编码器和解码器的详细体系结构。附录B中提供了基于深度学习的ROM的其他仿真结果。

# 控制方程和POD-TPWL ROM

在本节中，我们介绍了油水流动方程。然后，我们给出针对此问题POD-TPWL ROM的概述，再与E2C ROM方法进行类比。

## 2.1控制方程

不相混溶油水流动的控制方程由各组分的质量守恒和各相的达西定律推导导得出。在忽略毛细管压力影响的情况下，得到的方程为

 （1）

其中下标j（j=o、w分别表示油和水）表示流体相。地质特征通过孔隙度和渗透率张量用等式1表示。岩石和流体之间的相互作用由相迁移率指定，其中，其中表示相对渗透率，表示粘度。是压力，是饱和度（这些是主要的解变量），时间t和密度。表示井w的源项。通过限制饱和约束来完成此石油模型，本系统忽略了重力的作用，系统是水平（在x-y平面上）的。

采用标准有限体积方法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，我们利用斯坦福的基于自动微分的通用研究模拟器AD-GPRS[40]用于所有的流模拟。在每个网格块中，使用和两个主要变量来充分定义流系统，设n表示模型中网格块的个数，则系统中的变量总数为。我们定义特定时间步长t时的流量变量的状态向量为，其中和分别表示时间步长t时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：

.

其中表示我们要想要使其为零的残差矢量（非线性代数方程组），下标t表示当前时间水平，t +1表示下一个时间水平，指定油井控制变量，该变量可以是任何组合井底压力（BHPs）或井速（well rates），表示系统中的井的数量。在这项工作中，我们按照BHP规范操作生产井，并按照速率规范操作注入井。在这方面，我们的处理是通用的，也可以应用于其他控制设置。

方程2定义的全阶离散非线性系统通常使用牛顿法求解，这就需要构造维数为的稀疏雅各比矩阵，然后在每个时间步长的每次迭代中求解维数为 的线性系统。线性系统的求解往往是仿真中最耗时的部分。但POD-TPWL和基于深度学习的E2C ROM都避免了这个高维系统的测试时间构建和解决方案，之后将会解释。

## 2.2 POD-TPWL方法

许多基于深度学习的模型涉及的处理方式与现有ROM使用的不直接相似。相反，它们需要使用从图像分类、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。这个意义上，我们的E2C ROM有点不同，因为它的三个主要组件与现有的ROM，POD-TPWL中使用的组件很类似。由于POD-TPWL已针对地下径流应用进行了广泛开发，因此我们认为POD-TPWL 和E2C ROM 之间的对应关系值得讨论。为了便于讨论，我们首先提供了用于油藏模拟POD-TPWL方法的总概述。有关最近POD-TPWL实现的详细信息，请参阅文献[17，18，19，20]。

如前所述，POD-TPWL和其他基于POD的ROM包含涉及离线（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(本工作中使用的是AD-GPRS)执行许多训练模拟运行，目标是预测不同的井控序列的测试时间结果。因此，在训练运行中，我们使用不同的井控序列，其中包含控制步骤k中所有油井的设置，而表示训练运行中控制步骤的总数。在典型的仿真中，控制步骤数要比时间步骤数少得多（在我们的示例中，我们有20个控制步骤和大约100个时间步骤）。在训练运行的每个时间步骤中，所有网格块中的状态变量（称为快照）和导数矩阵都会被保存。在测试时，将执行与训练运行不同的控制序列的模拟。从训练运行中保存的信息被用于（非常有效地）近似测试的解。

POD-TPWL要求从高维空间投影到低维子空间，对低维子空间中的动力学进行线性近似，然后投影回高维空间。基于解快照矩阵的奇异值分解（SVD）构造投影矩x（这些快照矩阵在所有训练运行的所有时间步中都包含全阶解）。 给定，高维状态可以使用低维状态表示



其中是降阶空间的维数，且。需要注意的是，在实践中，SVD和随后的投影都是针对压力和饱和度变量进行的。因为是正交的，所以我们有。

在讨论低维空间中的POD-TPWL逼近之前，我们首先展示高维空间的线性化。按照[18]，TPWL方法（状态的POD表示，x = $£，应用于右侧）可以表示为

# 3 嵌入控制方法

在本节中， 我们开发了一个包含物理约束的嵌入控制ROM框架，建立了与POD-TPWL类似的各种E2C组件。我们在Watter等人开发的E2C模型上做了几个重要的修改，如下所述。

## 3.1 E2C简介

嵌入到控制框架需要三个处理步骤：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)的编码器或推理模型，一个近似于低维系统动力学的线性过渡模型，以及一个将解决方案投影回高维(全阶）空间的解码器或生成模型，最初由Watter等人[39]提出的E2C框架使用的是VAE架构的编码器和解码器程序，这使得他们能够考虑预测中的不确定性。在这里的嵌入控制方法中，由于我们正在考虑的是确定性系统，所以VAE体系结构被简化为自动编码器(AE)体系结构。我们注意到，自动编码器(AE)体系结构通常用于语义分割[41]，语义分割图像的每个像素与类标签相关联，然后用于深度预测[42]，其中场景的三维几何从二维图像中推断出来。在地下径流模拟的背景下，AE结构被用来构造代理模拟模型作为图像到图像的回归，其中输入图像是油藏性质（例如渗透率场），输出是状态变量[27 28]。

图1显示了我们的嵌入式控制模型的总体工作流程。压力场是图中显示的唯一状态变量(下标i表示训练运行中的时间步长)。我们的实际问题也包括饱和场，更多的状态变量将出现在更一般的环境中。(例如，如果考虑流-地质力学耦合模型，将出现位移这个状态变量)

接下来解释每个块1 3 5 的含义。

嵌入到控制ROM将控制变量包含在框架中，这是与Gonzalez和Balajewicz开发的基于VAE-LSTM的ROM的一个重要区别。在下面的小节中，将详细讨论嵌入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的实现细节。

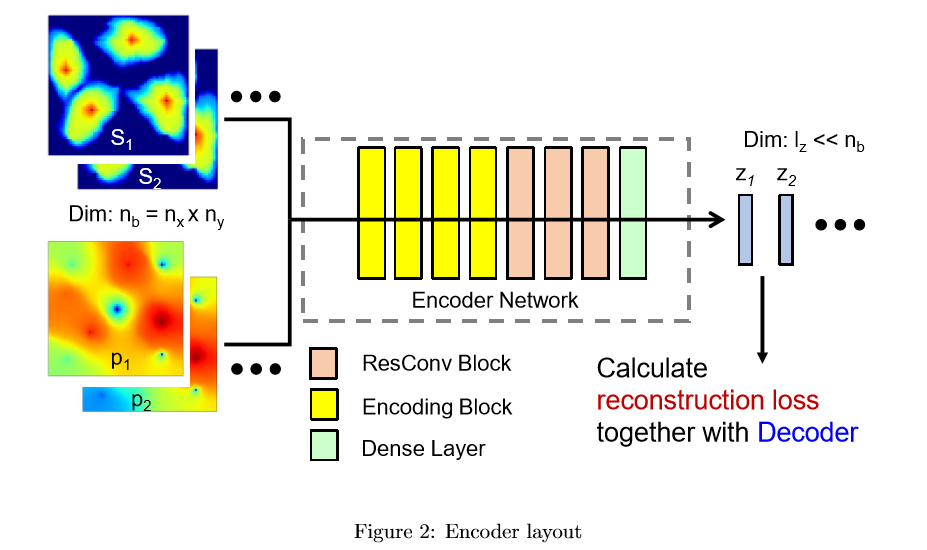
## 3.2 编码器组件

编码器提供了满阶状态变量的一个低维表示。这与Watter等人[39]开发的最初的嵌入控件实现相反，这里我们采用的是AE架构而非VAE架构。用这种处理仅仅估计潜在变量的平均值，而不估计方差。另外，我们不需要在潜在空间中进行采样。因此，在训练时间，编码器可以简单地表示为



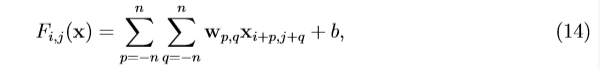
其中，就像之前解释的表示的是编码器。变量是在时间步长为t时的满阶状态变量，是对应的潜在变量，是潜在空间的维数。在后面的例子中，我们考虑一个二维60x60的油水模型（这意味着满阶系统为 7200维），我们设置。在POD-TPWL进程中，方程式13与方程式3是类似的，区别是在编码器中，POD的线性投影被一个非线性投影所替代。按照前面描述的惯例，我们使用没有“帽子”的变量来表示方程式2的（投影）真解，这些变量（真解）可以从训练运行中获得。带有帽子的变量指定近似解，这些近似解被测试时间ROM所提供。

在E2C 模型中的编码器的详细布局如图2所示。在训练过程中，通过编码器网络输入压力和饱和度快照序列，然后生成潜在状态变量序列。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(resConv)块的堆栈和一个密集层组成。与[39]使用的编码器相比，图中的编码器更复杂(它包含resConv块，并且具有更多的卷积层)。因为与在[39]中处理的原型规划任务（例如手推车平衡和三链路机器人的手臂规划）相比，基于PDE的压力和饱和场的适当表示需要来自更深层次网络的特征图，所以这里可能需要一个更复杂的结构。



类似于Liu等人提出的CNN-PCA，它使用CNN中的过滤器操作来捕获表征地质特征的空间相关性，嵌入控制框架使用卷积过滤器的堆栈来表示由基本控制方程确定的压力和饱和度场的空间分布。在更早时候，基于AE/VAE的ROMs的框架的实现已经表明卷积过滤器在流体动力问题中捕获此类场的潜力。因此，我们的编码器网络主要是这些卷积滤波器组成（以二维卷积层的形式，如conv2D 层[43]），附录A的表A.2提供了有关编码器网络的更多详细信息。

编码块的输入首先通过卷积运算进行输入，卷积运算也可以看作是一个线性滤波器。根据[24]中的表达式，线性滤波的数学公式为



其中x 是输入状态图，下标 i和j表示x和y的坐标方向指数，w 表示大小为的线性滤波器（模板）的权重，表示在空间位置的x的滤波器响应图（即特征图），b 是一个称为偏差的标量参数。通常，与一个conv2D 层相关联的过滤器和过滤器响应图有许多，收集所有这些操作的过滤器响应图是三阶张量。将输出过滤器响应图传递给批处理归一化（batchNorm）层[44]，该层对训练数据的每个子集应用归一化操作（将均值偏移为零并按标准偏差重新缩放）。批处理范数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。在归一化滤波器响应图上应用非线性激活函数ReLU(整流线性单元，max(0,x))[45]，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”。Con2D-batchNorm-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附录A.18(a)中找到。

为了正确地合并表示空间压力和饱和分布的特征图，如基础控制方程所确定的，我们需要一个许多卷积层的深度神经网络。然而，由于梯度消失问题深层神经网络很难训练。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]通过创建一个绕过非线性层的额外的标识图（resNet）解决了这个问题。遵循resNet的思想，我们在编码器网络中添加了一堆ResConv块，以加深网络，同时减轻消失梯度问题。在resConv块中的非线性层通常仍然遵循conv2D批处理Norm-ReLu 体系结构。有关resConv块的可视表示，请参见附录A中的图A.18(c)。

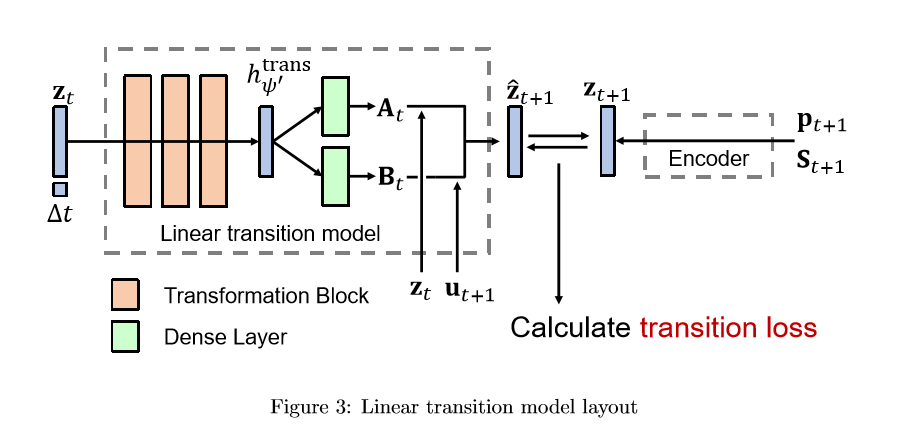
与编码块相似，ResConv块的输出是一堆低维特征图。这堆特征图被“扁平”到一个向量（由于大量的特征图，它仍然是一个相对高维的向量），然后输入到一个密集的层。密集（完全连接）层只是将高维向量映射到低维向量的一个线性投影。

这里使用的编码器网络的总体架构与Zhu和Zabaras构建的网络在三个关键方面有所不同。第一个方面，我们的编码器中使用的是resNet来缓解消失的梯度问题，而他们使用的是denseNet。第二个方面是[27]中的编码器(和解码器)不包括末端的密集层，这意味着编码器在末端会输出一堆特征图。对于随后由线性过渡模型执行的连续线性运算，大量的特征图（即高但相对较薄的三阶张量）的维数太大。最后，Zhu和Zabaras[27]采用了U-Net[41]架构，当编码器/解码器的输出（例如，压力场）与输入（例如，渗透率图）不同时，在他们的环境设置中，这是合理的。然而，当输入和输出类型相同（即压力和饱和场）时，U-Net架构是不适当的，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 构造的基矩阵具有使方程式9中的最小的特征。在编码器的上下文中，重建损失LR和POD的是类似的。概念上，“最好的”是通过最小化LR来发现的。然而，正如前面提到的，用于嵌入控制模型的优化涉及的是所有的三个处理步骤，因此LR不是单独最小化的。

## 3.3 线性传输模型

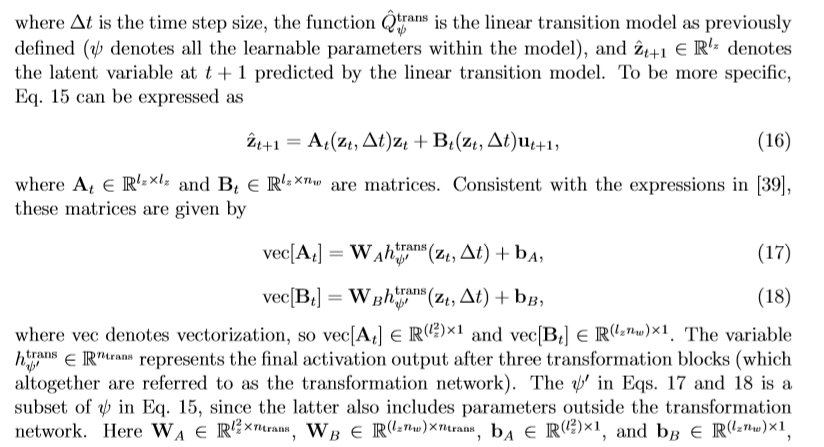
线性过渡模型在给定控制的情况下将潜在变量从一个时间步长发展到下一时间步长。图3显示了线性过渡模型是如何在离线阶段（训练时间）构建和评估的。线性过渡模型的输入包括当前状态的潜在变量，当前步骤控制和时间间隔。该模型输出下一时间步骤潜在状态的预测值。我们重申这一点表示线性过渡模型的输出。线性过渡模型的结构通常遵循[39]，由三个变换(Trans) 块和两个密集层组成。转换快遵循dense-batchNorm-ReLU 体系结构（密集表示密集层），这被认为是全连接网络的标准处理步骤。 附录A的图A.18（c）中显示了转换块的体系结构。变量和首先被输入转换块。转换块的最终激活向量通过两个独立的密集层构造线性矩阵和。然后将和矩阵与当前状态和当前阶跃控制的潜在变量相结合来预测下一个时间步骤的潜在变量。

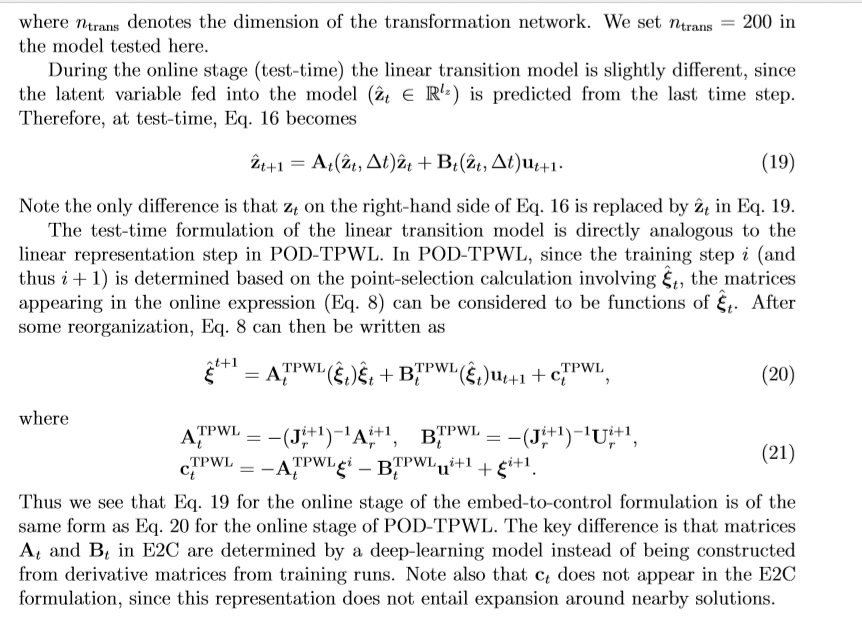


确定线性过渡模型参数的优化与POD-TPWL中的关键步骤是类似的。在POD-TPWL中，目标是最大程度地减少预测的缩减状态和预测的真实状态之间的差异，这部分通过确定最佳约束约减矩阵来实现，如方程式10和11中所述。给定了最佳的矩阵，出现在POD-TPWL中的矩阵，和都是完全定义的。如前所述，点选择代表了POD- TPWL中出现的另一种(启发式)优化。同样,在嵌入控制方法中,过渡损耗LT通过比较和得出，其中是线性转换模型的输出, 是在时间t + 1步被编码器投影的状态。转换的损失会被计入总损耗函数，该函数将会在脱机阶段被最小化。

在训练时间的线性传输模型可以表示为：



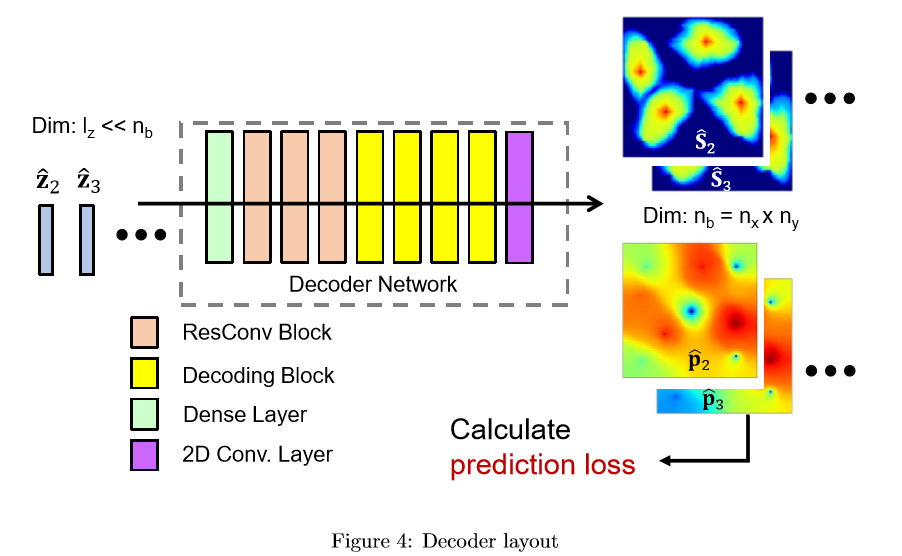
其中



## 3.4 解码器组件

解码器是类似于编码器的，能被表示为：

线性过渡模型预测的潜在变量(在时间步长t+1)被喂进解码器网络作为输入，预测的高维状态变量是输出。解码器的结构和编码器是类似的，只是组件的顺序相反。这里的解码器由一个密集层、一个由三个剩余卷积块块组成的堆栈、一个由四个解码块组成的堆栈和一个conv2D层组成。稠密层将低维潜在向量转换为一堆特征图(经过重新构造)。特征图在遍历resConv块和解码块的堆栈时展开。压力场和饱和场的空间分布在解码器的下游按顺序从特征图中“提取”出来。最后的conv2D层将扩展后的特征图转换为压力和饱和度场作为最终输出。关于编码器的详细信息在附录A的表A.3中给出。解码块的布局在附录A的图A.18(b)中显示。

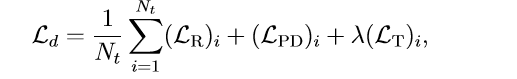


## 3.5 带有物理约束的损失函数

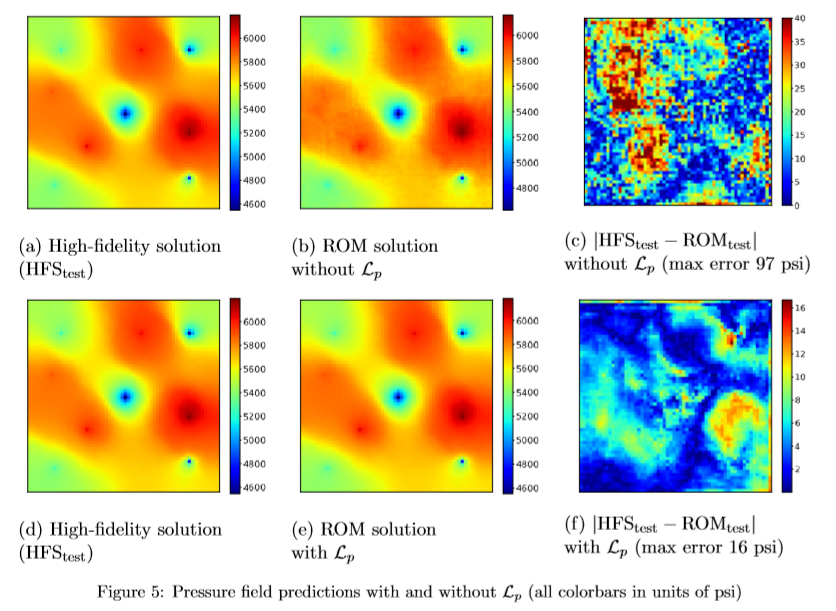
我们现在解释模型参数是如何在离线阶段被确定的。嵌入到控制框架的参数分别为编码器中的、线性过渡模型的和解码器的（都表示在此阶段可学习的参数）。目标函数为总损失函数，该函数量化了模型在预测输出状态变量时的总体性能。

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。

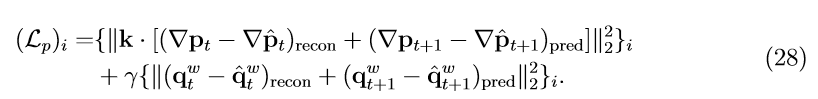
数据失配损失是所有训练数据点平均损失的总和：



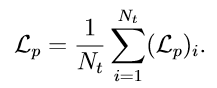
到目前为止所描述的ROM是一个纯粹的数据驱动模型，即该模型的目标是最小化E2C输出与高保真解之间的像素级差异(以HFS作为“真”参考解)。在某种程度上，物理行为是由E2C从输入压力和饱和度快照中推断出来的，但它并没有明确地强制执行。如果ROM使用在Eq.27中给出的损耗函数Ld进行训练，然而，非物理效应可以被观察到。如图5所示，我们在图5中给出了特定时刻的压力场预测(问题设置将在第4节中详细描述)。高保真解如图5(a)所示，仅基于Ld的E2C压力场如图5(b)所示。虽然两种结果在视觉上相似，但从图5(c)的差分图可以看出，E2C结果不够平滑，在某些空间位置出现了较大的误差。



为了解决这个问题，我们将数据不匹配的损失与基于流物理的损失函数结合起来。具体来说，我们试图最小化相邻网格块之间通量的不一致性。额外的权重也放在关键井数量上。我们考虑重构(时间步长t)和预测(时间步长t+1)，因此我们定义每个数据点的基于物理的损失，as



方程式28右侧的项对应式1中的通量项和源项。在本文的例子中，我们规定了注入井的速度和生产井的井底压力。有了这个规范，注入速率的损失为零，生产井跟踪的关键量是每口井的井壁压力。这是因为产量与井筒压力(这个例子中BHP被指定)和井筒阻塞压力之间的差成正比。比例系数是相流Aj和所谓的井眼指数[49]的乘积，这取决于渗透率、区块尺寸和井筒半径。

基于物理的损失函数通过平均的Lp来计算。将数据不匹配的损失和基于物理信息的损失结合起来，总的损失函数表示为：



通过有限的数值试验，我们发现a=0.033，r’=20是比较合适的。利用总损失函数L对某一时刻的压力场进行E2C ROM预测，如图5(e)所示。图5(d)再次为高保真解(与图5(a)相同)，差分图如图5(f)所示。当基于物理信息的损失函数（Lp）加入总损失函数后，ROM的预测有了明显的提高。显然，最大的压力误差从97ψ到16ψ，而且得到的场更光滑（因此更物理）。这证明了将基于物理的损失纳入E2C ROM是有益处的。

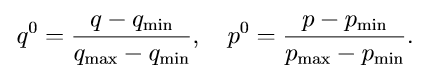
## 3.6嵌入到控制框架的实现和训练的细节

为了训练E2C模型，我们使用一个数据集D = {(xt, xt+i, ut+x)i}， i = 1，…（包含全阶状态变量和井控制）其中，N为训练运行数据点的总数。在本文的示例中，我们总共模拟了300次训练运行。这比使用POD-TPWL(我们通常在其中模拟三到五次训练运行)要多得多，但是我们期望E2C具有更高的鲁棒性。我们的意思是期望ROM在大范围的控制规范上提供准确的结果，而不是像在POD-TPWL中那样在一个有限的范围内有好的结果。

不是训练所有快照，这里我们设置Nctri=Ntr=Nte=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量上。N个数据点的总数 Nt=300x20=6000。

总损失函数相对于模型参数的梯度通过嵌入到控制框架的反向传播来计算。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控制，我们设置。

归一化是一个重要的数据预处理步骤，他的合适的应用可以提高学习过程和输出结果的准确性。因为饱和度的范围为，所以不需要归一化。对压力、井数据和控制变量进行归一化。

 解释变量的含义

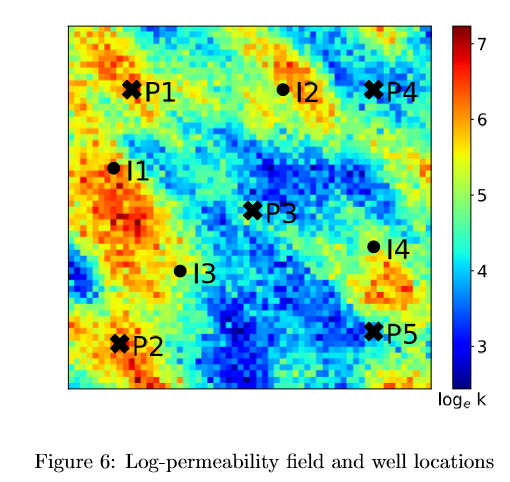
在双IntelXeonES-2670CPU（24核）上运行时，每个全阶训练模拟需要大约60秒的时间。我们的E2C ROM是使用带有TensorFlow后端的Keras实现的。离线训练过程（不包括训练模拟运行时）在特斯拉V100GPU节点上大约需要10-12 分钟(确切的时间取决于分配的内存，内存可以从8-12GB变化)。该模型应用于 100次测试运行，将在下一节中详细讨论。几乎所有的测试结果都是使用300次训练运行产生的，我们还提供了使用100和200次训练运行的汇总误差统计。对于100次和200次的离线训练，除了在全阶训练模拟中直接节省的时间，它们需要的训练运行时间与300次大约相同。

# 4使用嵌入控制的降阶模型的结果

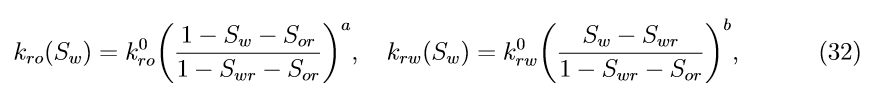
在本节中，我们描述了油水模拟的模型设置，并给出了基于深度学习的ROM的模拟结果。本节详细考虑了其中一个测试用例。附录B中提供了另外两个测试用例的结果。在本节中，我们还提供了所有100个测试用例的汇总误差结果。

## 4.1 模型设置

以对数渗透率场表示的地质模型如图6所示。同时还显示了四个注入井和五个生产井的位置。油藏模型包含60×60(共3600个)网格块，每块尺寸为50米×50米×10米。对数渗透率场的相关结构采用指数变差模型，最大和最小相关长度分别为~1000 m和~500 m，方位角为45°。渗透率的算术平均值为158 mD，对数渗透率的标准差为0.88。渗透率被认为是各向同性的，而孔隙度被设置为0.2的恒定值。



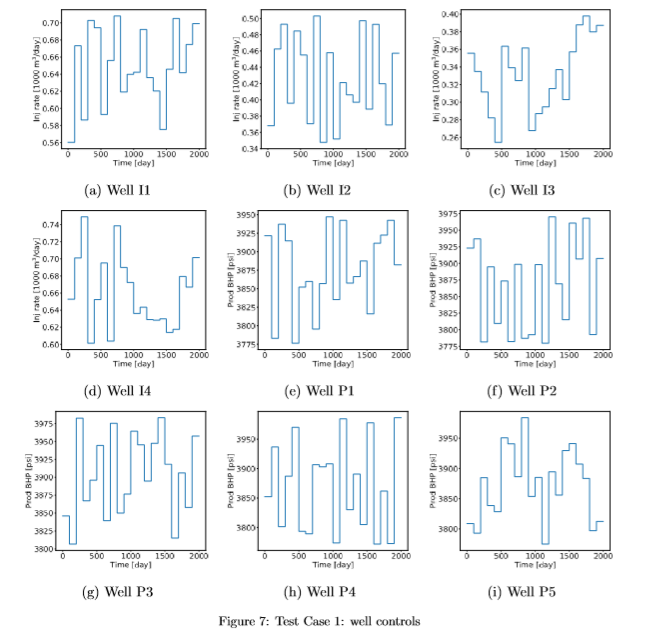
渗透率的函数被给出：



给出每个量的具体值，毛细管压力效应被忽略。

油藏顶部初始压力为4712psi(325bar)，初始含水饱和度为0.1。系统中主变量的总数是3600x2=7200。该模型总共运行2000天。注水井通过指定随时间变化的含水率进行控制，生产井通过指定随时间变化的井底压力进行控制。生产井和注水井的控制措施每100天改变一次，这意味着有20个控制周期。因此，我们在整个仿真时间框架内总共有9x20=18个控制参数。注水量范围为1500~ 6500桶/天(238~1033立方米/天)。与通常认为的ROMs相比，这是一个非常大的井作业范围[19,20]。生产BHPs的范围是3770到3988psi(260到275bar之间)。

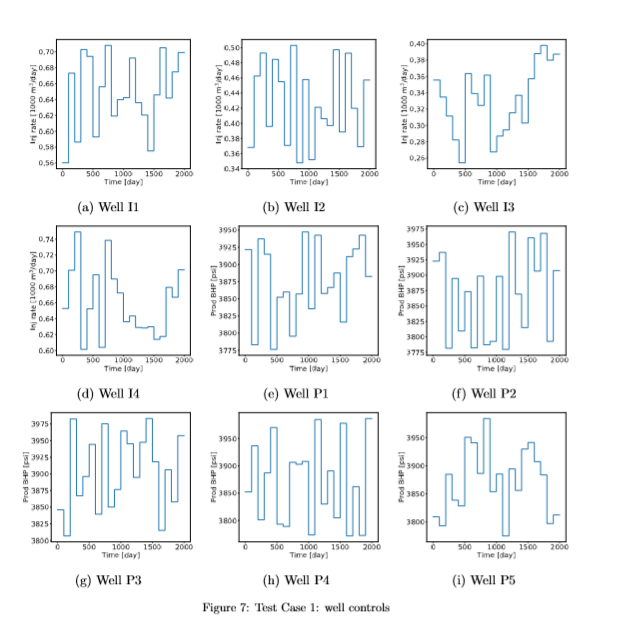
训练和测试运行的控制被指定如下。对于每口注入井，我们随机取样，从2000-6000桶/天的均匀分布中，一个基准注入速度。然后，在每个控制期间，我们在[-500,500]桶/天范围内均匀取样一个扰动。然后规定控制期的速率为。每个控制步骤的生产者BHPs均在[3770,3988]psi范围内采样。对于生产井，没有基准压力BHPs，从控制步骤到控制步骤的设置是不相关的。这种指定注入速率的方法导致了各种各样的溶解行为(例如，饱和度分布)，因为每口井的注入量在不同的井间变化很大。如果注射速率没有参照基线值，这种方式也避免了可能发生的平均效应。使用此过程生成的测试用例的井规范如图7所示。



除非另有说明，否则我们将进行300次训练模拟以构建E2C ROM。如先前的论文所讨论（例如，方程[19]），图7中所示的油井调度表旨在表示在优化程序期间评估的油井控制曲线，其目标是最大程度地提高石油产量或获利能力，或最小化环境影响或某种程度的风险。最后，我们注意到E2C潜在空间的尺寸设置为50。

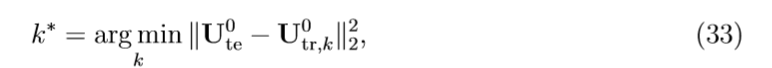
## 4.2 案例1的结果

本节中，我们将给出特定测试用例的详细结果，包括井量(注入BHPs和生产速率)和全局变量（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7中。在这里，我们显示了四个注入井的水的速率（图7(a)-(d)），以及五口生产井的井底压力（图 7(e)-(i)）。

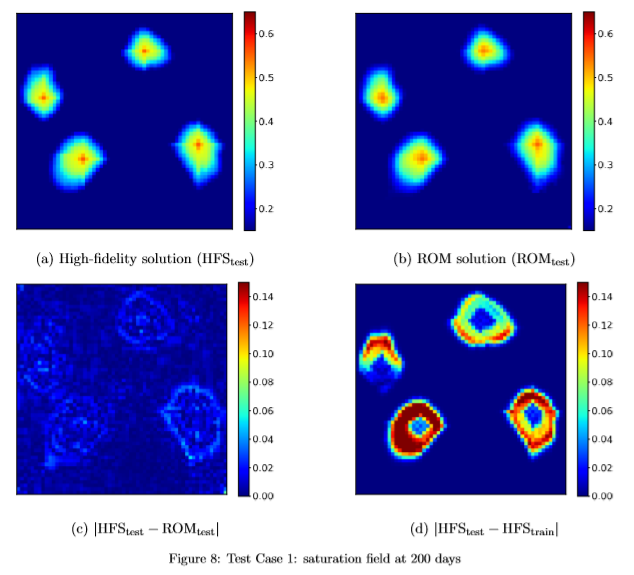


我们现在评估基于深度学习的ROM在这个测试用例中的性能。首先考虑全局饱和场的时间演化。图8显示了200天时的饱和场，图8(a)给出了全阶饱和场(也称为高保真解HFS)，图8(b)给出了相应的E2C ROM饱和结果。色标表示水饱和度（红色表示水）。从图8(a)和(b)中可以看出，基于深度学习的ROM能够提供精确的结果。两种解决方案之间的一致程度在图8（c）中进行了量化，其中显示了HFS和ROM解决方案之间的差异。请注意，这里的颜色条比例与图8(a)和(b)非常不同，ROM和HFS结果之间的误差显然非常小。

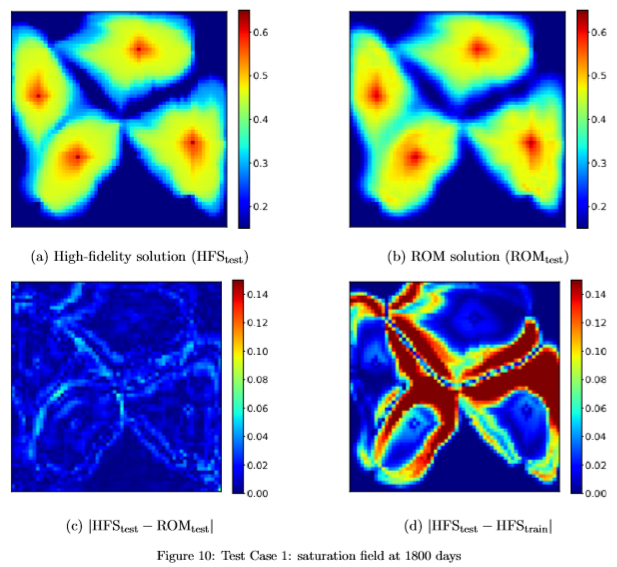
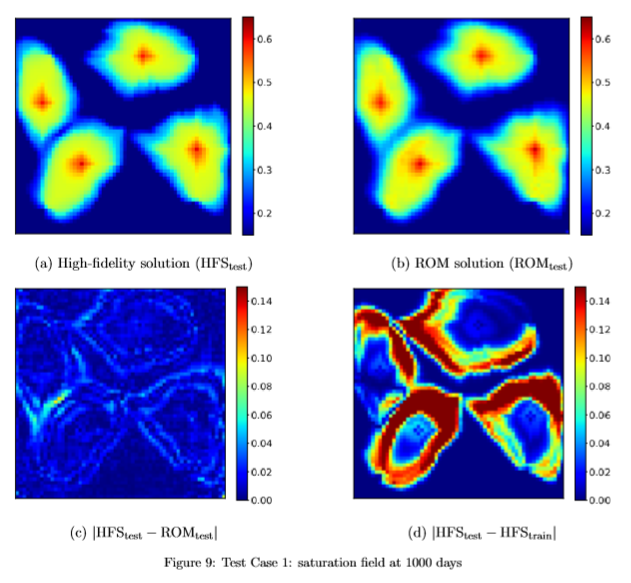
为了更好地量化E2C ROM的预测能力，我们引入了“最近的训练运行”的概念。我们用这个术语来表示所执行的300次训练中，最类似于（在特定意义上）测试用例的特定训练。测试运行和每个训练运行之间的“距离”是根据其标准化控制参数向量之间的欧几里得距离来量化的，“最近的训练运行”（k\*）是具有最小距离的训练运行。特别地，



方程33提供了一个非常近似的“最近训练运行”的概念。这个定义的优点是简单，虽然更多的复杂评估(和计算要求)将有望提供更精确的训练解决方案。然而，这将需要采用一种类似于POD-TPWL[19]中使用的“点选择”过程的方法，这将需要为每次训练运行计算多个时间步长的距离度量。由于我们在这里有300次培训(而不是使用POD-TPWL的3次或5次)，这可能会非常耗时。因此，我们应用方程33中定义的简单方法，并认识到可以设计更复杂(可能更精确)的过程。

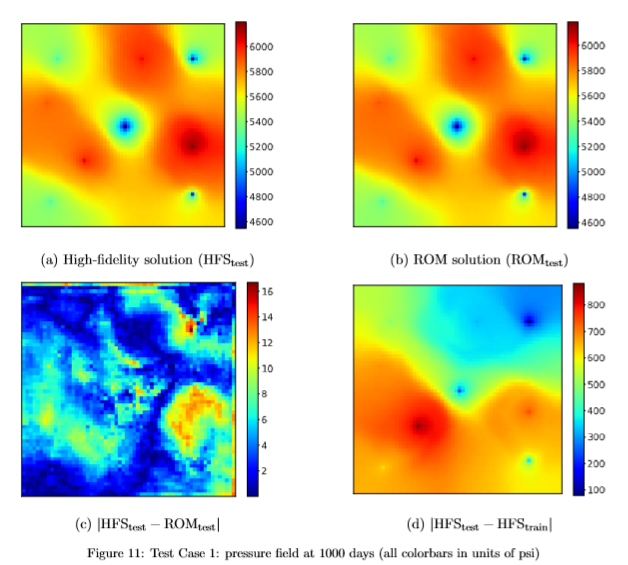


现在我们回到全局饱和结果。图8(d)显示了“最近的训练运行”(我们刚刚定义并在高保真度下模拟)和测试用例饱和场之间的差异。颜色条比例尺与图8(c)相同。通过对比图8(c)和图8(d)，可以明显看出应用基于深度学习的ROM的优势。更具体地说，图8(c)的误差大约比图8(d)的误差小一个数量级。图9和图10显示了1000和1800天饱和度的类似结果。饱和场随时间的演化是明显的，基于深度学习的ROM解(图9(b)和图10(b))再次与HFS有密切的视觉一致性(图9(a)和图10(a))。图9(c)和图10(c)中的误差图进一步量化了基于深度学习的ROM的精度。与图9(d)和图10(d)中显示的“最近训练运行”和HFS之间的差异图相比，这些误差非常小，这进一步说明了ROM的有效性。



请注意，在ROM溶液中，我们确实观察到了饱和羽流中的一些局部(非物理)极值。这是一个小问题，因为差异图显示ROM和高保真度解决方案之间的小的总体差异。然而，在某些情况下，这可能会引起关注。一种可能的补救办法是在基于物理的损失函数中增加一项，使不符合控制流动方程的局部饱和极值受到惩罚。

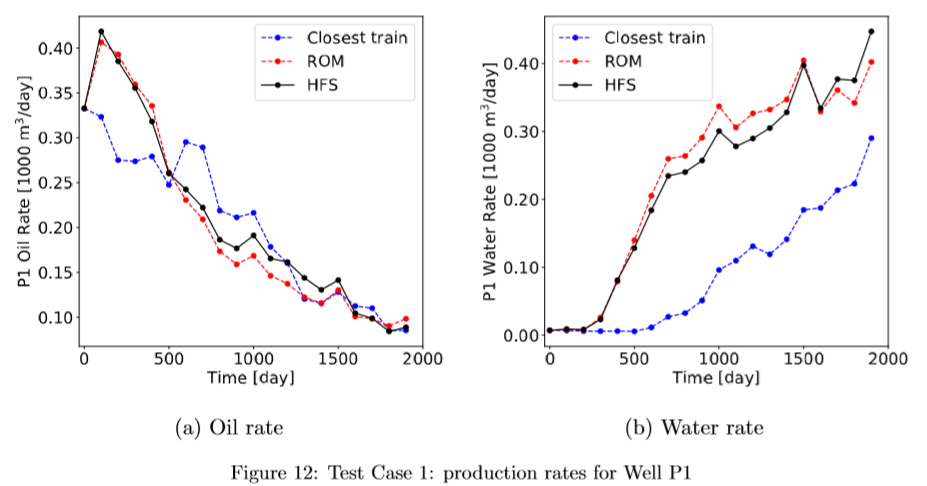
在特定时刻的全局压力场也是令人感兴趣的。图11的(a)和(b)显示在1000天是的HFS和ROM压力解决方案。紧密的视觉一致性表明，基于深度学习的ROM能够提供准确(和平滑)的压力预测。图11(c)显示了ROM解决方案的误差图，我们可以看到误差确实非常小。这些误差比“最近训练运行”的误差要小得多，如图11(d)所示。

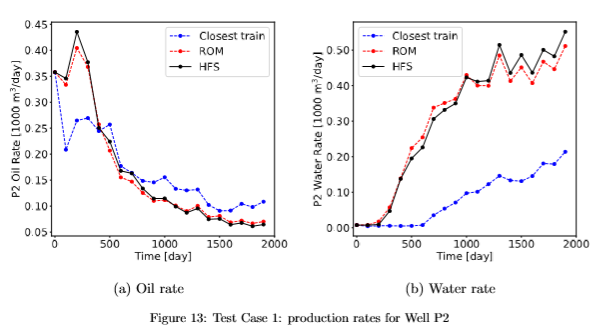


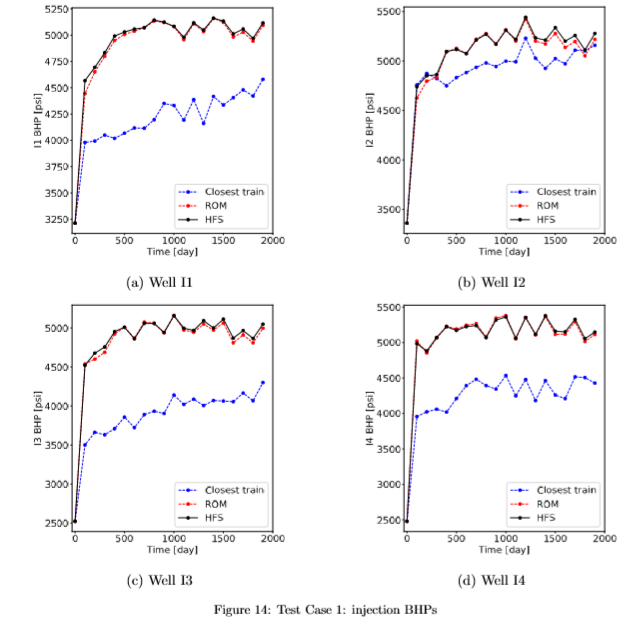
在许多地下径流应用中，井的响应是最重要的。现在将对E2C ROM对井的响应预测进行评估。因为在这个问题中，我们指定了注入速度和生产井的井底压力，所以我们现在感兴趣的量是注入井的井底压力和油水产量。图12和图13显示了P1和P2井的相流速率，这两个井对油田总产量的贡献最大。图14显示了四个注入井的压力响应。在所有的图形中，黑色曲线代表全阶(参考)HFS，红色的曲线是基于深度学习的ROM结果，蓝色的曲线是“最近训练运行”的结果。ROM和HFS结果之间的高度准确性得到了一致的观察。通过在损失函数Lp中增加井块数量的权重，可以增强这些基本量的一致性(参见公式28)。

我们在附录b中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度和压力分布以及感兴趣的量。

最后，我们讨论了高保真度和ROM运行的时间。高保真度测试用例使用AD-GPRS在具有双Intel Xeon cpu(24核)的节点上进行模拟，每个用例耗时60秒。E2C ROM可以在大约2秒内评估全部100个测试用例，在一个分配了大约1GB内存的Tesla V100 GPU节点上使用。直接(虽然过于简单)的比较表明加速因子为3000。



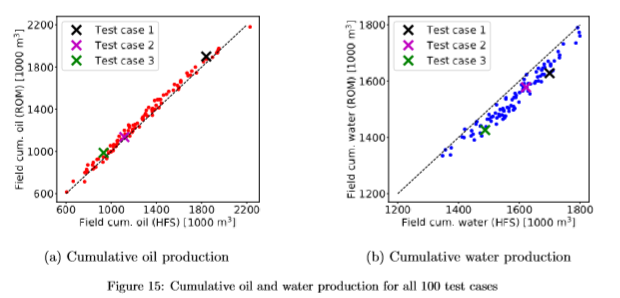




## 4.3 所有测试案例的结果和误差估计

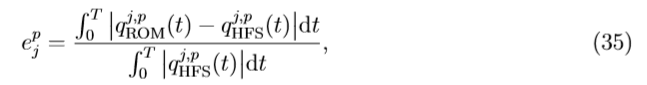
在本节中，我们将评估100个测试用例的完整集成的ROM结果的准确性。我们首先考虑油田的累计产油量和产油量，这是由





在图15中，我们给出了100个测试用例的HFS和ROM产生的的累计油产量累计水产量的交叉图。每个图上的三个x表示测试用例1、2和3的结果。很明显，这些用例的Qo和Qw完全不同，在这种意义上，它涵盖了100个用例的范围。我们看到两个图中的点都位于45°线附近，这表明我们的ROM解决方案与HFS非常一致。图15(b)中的Qw结果表明，ROM始终低估了累积产水量。然而由于该图所涉及的范围较窄，欠预测的范围相对较小。在图15(a)可以看到ROM对累计油产量的轻微过度预测。

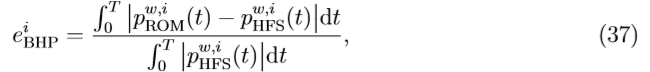
我们现在介绍一些误差度量，这些度量将用于评估E2C ROM的一般性能。这些误差度量遵循[20]中使用的度量。单个生产井p的油水产量相对误差定义为:



我们将所有生产井的总产率的误差用和定义为:



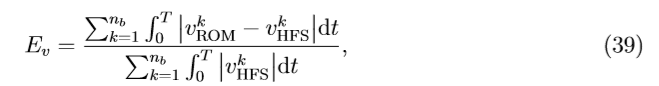
同理，单个注入井i的井底压力相对误差定义为:



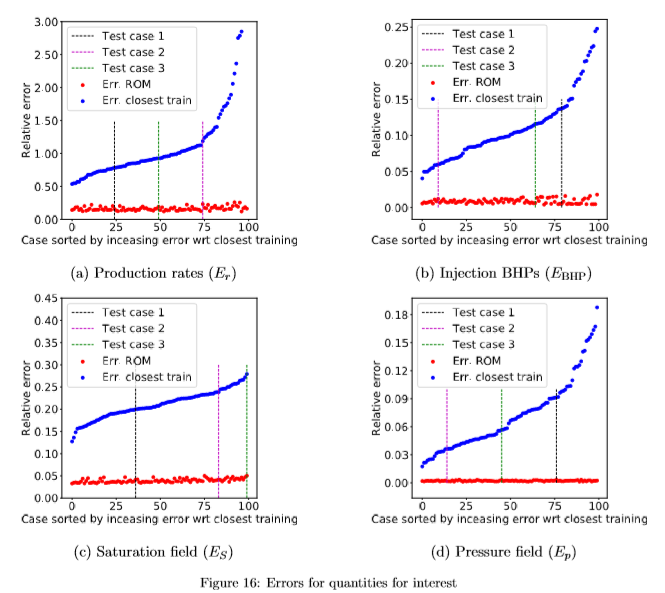
注入井总压力误差为:



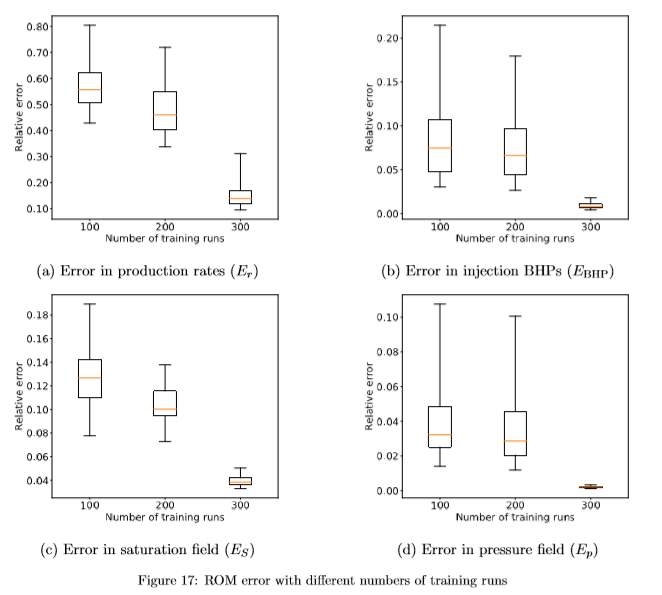
全局变量值的误差也是我们感兴趣的。我们定义全局压力和饱和度的误差为：



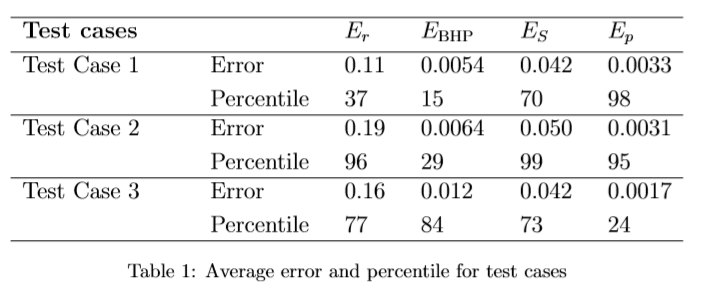
这四个误差量在图16中用红色点表示。我们也用“最近训练运行”为所有测试用例评估这些误差。在图中，这些点是根据“最近训练运行”(蓝点)的递增误差排序的。测试用例1、2和3的结果显示在每个图中。我们看到ROM的误差一直都是非常小的，而“最近训练运行”的误差在许多情况下很大。有趣的是，ROM的误差似乎并不取决于与“最近训练运行”相关的误差。这是一个理想的特性，因为它表明E2C ROM具有高度的健壮性。



现在我们简要地考虑在构建E2C ROM时使用更少的训练运行次数。对于这些情况，我们只给出概要性的误差结果。图17显示了上面考虑的四个相对误差，在箱形图中，针对100、200和300次训练运行。在每个框中，中间的橙色线表示中间误差，框的底部和顶部边缘表示第25百分位和第75百分位误差。从盒子中伸出的“胡须”表示最小和最大误差。当我们从200增加到300次训练次数时，ROM的准确度有了显著的提高。在今后的工作中，提出确定所需训练次数的方法是有用的。



因为很难在图17的方框图中显示测试用例1、2和3的误差，我们在表1中显示它们。这是300次训练的结果。注意，所有100个测试用例中Er、EBHP、ES和Ep的平均值分别约为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被看到代表了整个测试用例集之间的合理分布。观察这四个误差在一个特定的测试用例中并没有紧密的关联是很有趣的。例如，在测试用例1中，Ep位于第98个百分位数，而EBHP位于第15个百分位数。



# 5 结束语

在这项工作中，我们介绍了一个基于深度学习的地下径流模拟降阶建模程序。该程序是从现有的嵌入式控制(E2C)程序改编而来，我们介绍了一些与文献[39]的方法相关的关键修改。从本质上讲，ROM由一个自动编码器(AE)和一个线性转换模型组成。在我们的E2C方法中，给出一个附加的基于物理的损失函数与数据失配损失函数相结合，以增强与控制流方程的一致性。虽然它是基于深入学习的概念和方法，各种E2C ROM步骤被证明是非常类似于那些在已开发的基于物理/数字的POD-TPWL ROM中使用的步骤。

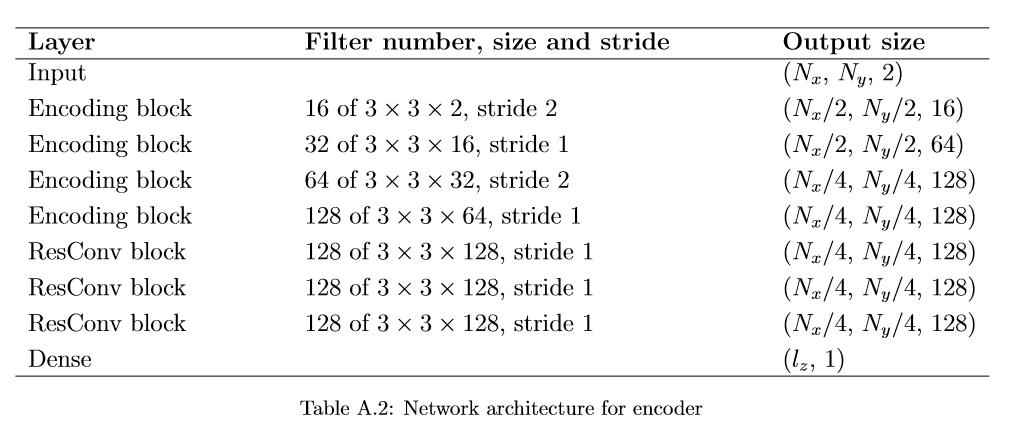
在大多数评估中，我们在训练时间阶段执行了300次训练运行。除去训练模拟的运行时间，使用特斯拉V100 GPU进行ROM训练需要10-12分钟的离线模型构建。在线(运行时)加速0(1000)，相对于AD-GPRS全阶仿真，观察了考虑的情况。考虑到离线成本和在线加速，当需要多次(相关的)模拟运行时，使用这个ROM是合适的。这是生产优化计算、数据同化和不确定性评估的情况(尽管在这项工作中只考虑了一个地质模型)。

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井和生产井控制设置的较大变化(相对于训练井)。总共考虑了100个测试用例。ROM在评估我们感兴趣的关键参数时是准确的，包括注入井的井底压力、生产速率、全局压力和饱和度场。E2C ROM在整个测试运行过程中是准确的。ROM误差被认为比“最近的训练跑”要低得多。但是，如果使用100或200次训练而不是300次，则误差会增加。

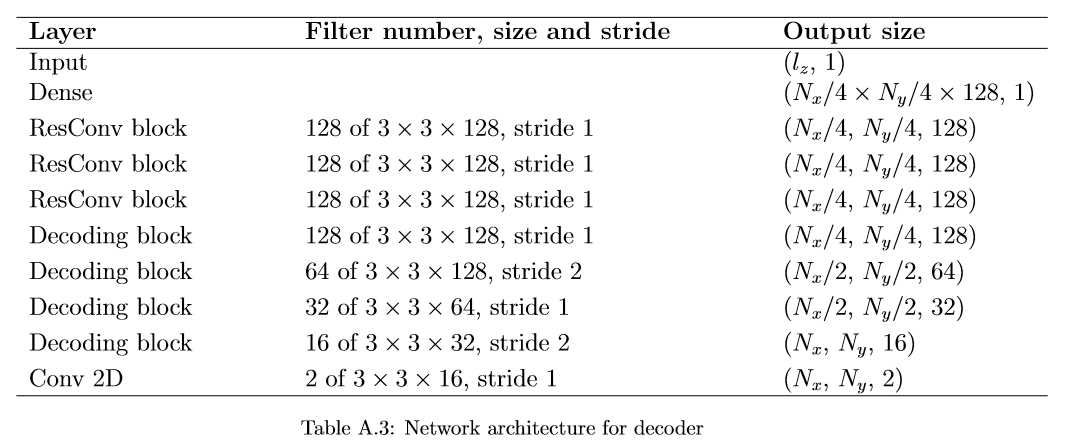
在未来的工作中，E2C ROM应该扩展到更复杂的三维问题，并在现实案例中进行测试。可以通过用conv3D层替换conv2D层来扩展到三维空间。ROM可以很容易地与用于生产优化的各种优化算法一起使用，并且应该评估它在此设置中的性能。重要的是，E2C ROM应该既适用于全局优化算法，也适用于局部优化算法。这与现有的基于特征正交分解的降阶模型形成了对比，后者在有限的邻域中是准确的，因此更适合于局部搜索方法。随着井位的变化，对预测流动响应的潜力的研究也很有意义。如果成功，ROM可以用于井位优化，或可用于将井位和控制优化结合起来问题上。为了提高更多变应用框架的准确性和鲁棒性，可以考虑更复杂的编码器和解码器结构，如denseNet[53]。最后，当考虑的是不确定性的系统和控制时，这里使用的自动编码器可以替换为VAE或不确定性自动编码器[54]。

# 附录A 嵌入到控制的网络架构

表A.2中显示了编码器的结构。该编码器具有四个编码块的堆栈，分别具有16、32、64和128个过滤器。编码块的详细结构如图A.18(a)所示。滤波器的大小是指编码块内的二维卷积层的大小。批处理归一化层和修正线性单元层的尺寸与二维卷积层的输出尺寸一致。编码器还有一个由三个剩余卷积(resConv)块组成的堆栈，每个模块有128个大小为和步长为1的过滤器。过滤器大小再次引用resConv块内的conv2D层的大小。resConv块的详细结构如图A.18(d)所示。注意，在被送入密集层之前，resConv块的输出从一个维度为的3D矩阵被重塑为一个长向量。编码器末端的密集层可视为输入尺寸为的线性变换，输出尺寸为lz(这里考虑的情况下lz = 50)。这里和是输入图像的高度和宽度，在我们的例子中都是60。

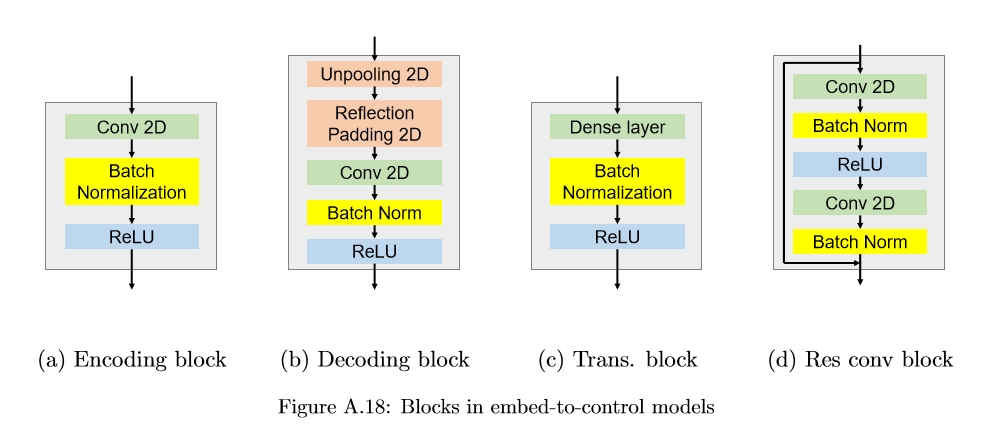


表A.3总结了解码器的体系结构。解码器的结构类似于编码器的结构，但组件的顺序相反。解码器由密集层，三个resConv块的堆栈，四个解码块的堆栈和conv2D层组成。请注意，密集层的输出是尺寸是维度为的长向量，在馈入resConv块之前，将其重新整形为大小为的三维矩阵。 解码块的详细结构如图A.18（b）所示。



图A.18显示了编码块，解码块，传输块和剩余卷积块的详细结构。 编码块是（conv2D），（bathNorm）层和修正线性单元（ReLU）的顺序组合。通过调整二维卷积层的步幅大小（例如，表A.2中的编码块的步幅大小为2），可以实现空间尺寸的减小（下采样）。解码块具有一个二维反池化层、一个二维反射填充层、二维卷积层、批处理归一化层和修正线性单元。二维反池化层通过重复数据的行和列来增加输入的大小，二维反射填充层通过在两个维度的边界上填充边界来增加输入的大小。空间尺寸的增加（上采样）是通过2D解池和2D反射填充层实现的。

resConv块的结构与resNet非常相似，其中创建了恒等变换以绕过非线性层。对于resConv层，非线性层不需要学习输入x到目标函数F（x）的直接映射，而是只需学习H（x）:= F（x）-x的残差映射。在极端（最坏）情况下（F（x）= x），该值将为零，并保证较深的神经网络相对于较浅的神经网络可以实现更高（或至少相等）的精度。非线性层遵循conv2D-batchNorm-ReLU的标准结构。传输块具有dense-batchNorm-ReLU的体系结构，其中对于测试情况，密集层的尺寸设置为200。



# 附录B 其他的测试案例的结果

我们现在给出测试用例2和3的E2C ROM结果。图15和16以及表1显示了这些情况下感兴趣数量的误差。我们在这里的描述非常简短，因为这些结果与第4节中讨论的结果非常相似。

## 案例2的结果

图B.19、B.20、B.21显示了200天、1000天、1800天的饱和场，图B.22显示了1000天的压力场。饱和度的变化表明，与测试用例1相比，这里的扫描略有不同。具体来说，对比图B.21(b)和图10(b)，我们可以看到I2井(右上)附近的水羽更大，而I1井(左上)和I3井(左下)附近的水羽更小。具体井的位置见图6。测试用例2与测试用例1在1000天时的压力图非常不同。图中所示的差异图在B.19(c)、B.20(c)、B.21(c)和B.22(c)表明ROM的预测对于这些全局变量是准确的。

图B.23和B.24显示了P1井和P2井的产油量和产油量，在测试案例2中，这两口井的产油量和产油量对产油量的影响最大。图B.25为四口注水井的注水井的井底压力。ROM的解再次与高保真模型的解紧密一致，并且观察到这些解决方案与“最接近的训练运行”之间的显著差异。

## 案例3的结果

图B.26、B.27、B.28为测试例3的饱和度演变的类似结果，图B.29为1000天压力场。与其他两个测试案例相比，这里的油藏分布略有不同，在I4井(右下)附近的水柱明显较小。差异图再次表明E2C ROM具有很高的准确性。

