arXiv:2301.02191v1 [physics.comp-ph] 5 Jan 2023

用于导电边界包围的带电粒子的物理信息神经网络

预印本

Fatemeh Hafezianzade基础科学高等研究院物理系，伊朗赞詹，45195-1159
fahafe98@gmail.com

Morad Biagooi IntelligentData Aim Ltd (IDA Ltd)，基础科学高等研究院科技园，伊朗赞詹，45137-65697

SeyedEhsanNedaaee Oskoee*基础科学高等研究院物理系，伊朗赞詹，45195-1159
nedaaee@iasbs.ac.ir

2023年1月6日

摘要

本文开发了一种基于物理信息神经网络的新模型，用于预测导电壁包围的点电荷粒子的势。对于相应的仿真示例，所提出的物理信息神经网络模型的均方误差低于 7%， R^2 分数高于 90%。我们将结果与典型神经网络以及作为标准机器学习算法的随机森林进行了比较。随机森林模型的R^2 分数为 70%，而标准神经网络则无法得到良好训练。此外，与有限元求解器相比，计算时间显著缩短。

关键词 泊松方程· 拉普拉斯方程 · 物理信息神经网络 · 带电粒子 · 导电边界 · 超级电容器

1 引言

计算电磁仿真（CES）在许多科学和工程领域中发挥着重要作用，例如软物质、电气工程、生物医学工程和化学。此外，它在工业中也有许多应用。例如，它是研究和设计超级电容器工作过程的主要工具之一。超级电容器是一种多孔储能器件，在工业中有许多应用，尤其适用于需要高功率消耗或传输的场景Miller and Simon [2008]。在此，研究超级电容器电荷存储所涉及的物理机制，对于进一步的技术发展至关重要Salanne et al. [2016],Simon and Gogotsi [2008]。

*通讯作者

求解麦克斯韦方程，尤其是本研究中的泊松方程，是计算电磁算法的重要组成部分Jackson [1962]。求解泊松方程有助于科学家计算任意系统中电荷源的势。然而，由于电相互作用具有长程特性，实际求解时会遇到许多困难。尤其是，估算点电荷组件在导电壁环境中的势颇具挑战性，因为边界上存在感应电荷。

通常，求解泊松方程有两种方法：解析解Jackson [1962]和数值方法。可用于解析求解的技术有限，例如适用于规则几何形状的镜像电荷法；然而，这些方法无法保证获得实用结果。例如，若将一个粒子置于立方体导电容器中，镜像电荷法会产生无穷级数。另一方面，数值方法通过对空间和/或时间域进行离散化，得到近似解。有限元法 (FEM)Jin[2014] 是典型的数值方法之一，它对连续的偏微分方程 (PDEs) 进行离散化，并构成线性代数方程组S. 等人 [1991]。然而，即使是有限元法，也无法计算带电粒子所在位置的势，因为电势在电荷所在位置是奇异的。为解决这一问题，人们开发了多种方法和算法，包括适用于二维的感应电荷 MMM2D (ICMMM2D)Tyagi 等人 [2007] 、适用于二维 + h 的 ELCICTyagi 等人 [2008]、适用于三维周期性的感应电荷计算 ($ICC*$)Tyagi 等人 [2010], Kesselheim 等人 [2010], Arnold 等人 [2013] ，以及 Reed 等人提出的方法Reed 等人 [2007]。此外，近期还出现了一种名为 PLT 的算法。Rostami 等人 [2016], 首次展示了该算法在两个金属板之间受限的部分周期性系统中的应用，随后又将其应用于 CAVIAR Biagooi 等人 [2020], ，这是一款用于模拟被非平凡导电边界包围的带电粒子的分子动力学模拟软件包。使用 CAVIAR 软件包对这些问题进行数值求解十分准确；此外，与$ICC*$ Biagooi 等人 [2020] 相比，所用时间更短，但仍然耗时且占用大量内存。

近年来，另一种基于深度机器学习、用于求解偏微分方程的数据驱动方法也备受关注。例如，Shan 等人提出了一种卷积神经网络，可预测二维和三维模型中具有不同激励和介电常数分布时的电势。与有限元法相比，该方法快速且高效。Jin [2014]然而，有几个问题使其无法在分子动力学模拟过程中用作泊松方程求解器：首先，它无法处理离散密度函数，例如点电荷的离散密度函数；其次，这是一种不含物理信息的方法，因此难以考虑边界条件。要解决第一个问题，可以使用 PLT算法。此外，Raissi 等人提出了物理信息神经网络（PINN）。由于其损失函数的定义中使用了控制方程、边界条件和初始条件，因此 Raissi 等人 [2019] 提出的损失函数是传统深度学习方法的一种出色替代方案。

本文应用了一种基于物理信息神经网络（PINN）的新模型，用于预测被导电壁包围的点电荷粒子的势。随后，我们将结果与作为标准机器学习算法的典型神经网络和随机森林进行了比较。例如，我们尝试将这些模型应用于球形容器中的带电粒子。之所以采用这个简单示例，是因为可以通过解析方法——镜像电荷法——求得该问题的精确解。作为起点，我们使用 PLT算法，将泊松方程转换为具有修正边界条件的拉普拉斯方程。然后，我们训练模型求解具有新边界条件的拉普拉斯方程。输入数据包括我们希望计算势的位置以及修正后的边界条件；输出数据则是该位置对应的电势。

2 方法

本文旨在构建一个机器学习模型（ML-Model），以预测被导电壁包围的点电荷粒子的势。带电粒子的势通过求解泊松方程计算，该方程可写为 Jackson [1962]:

$$\nabla^2\phi = -\rho/\epsilon_0 = -\sum_{i=1}^{N} q_i\delta(x - x_{q_i})/\epsilon_0, \tag{1}$$

其中， ϕ 是势， ρ 是电荷分布。首先想到的一个直接明了的机器学习模型，是将 x_q 和 x 作为输入，并将 ϕ (x_q， x)作为输出的模型。这里， x_q 是点电荷粒子的位置， x 是我们要计算其势的位置， ϕ(x_q， x)是对应的势。因此，输入特征的数量取决于带电粒子的数量；例如，在三维空间中，如果有N个带电粒子，输入特征必须为 $3 + 3 \times N$。因此，

这种模型只能预测固定数量的带电粒子的势。在该方法的许多应用中，例如分子动力学模拟，带电粒子的数量并非固定，在仿真过程中甚至可能增加或减少。我们使用PLT算法将泊松方程转换为带有新边界条件的拉普拉斯方程，以解决这一问题。该算法将在2.1节中进一步详细讨论。因此，我们可以训练一个模型，将 x 和经过修改的边界条件作为输入特征，并将 $\phi(\phi_b,\ x)$作为输出。我们仅在边界上的 N_b 个固定点处定义边界条件$\{\phi_1,\ \phi_2,\ \ldots\ldots,\ \phi_{N_b}\}$。在这种情况下，借助PLT算法，我们可以构建一个具有固定数量输入特征的模型，用来预测任意带电粒子的势。

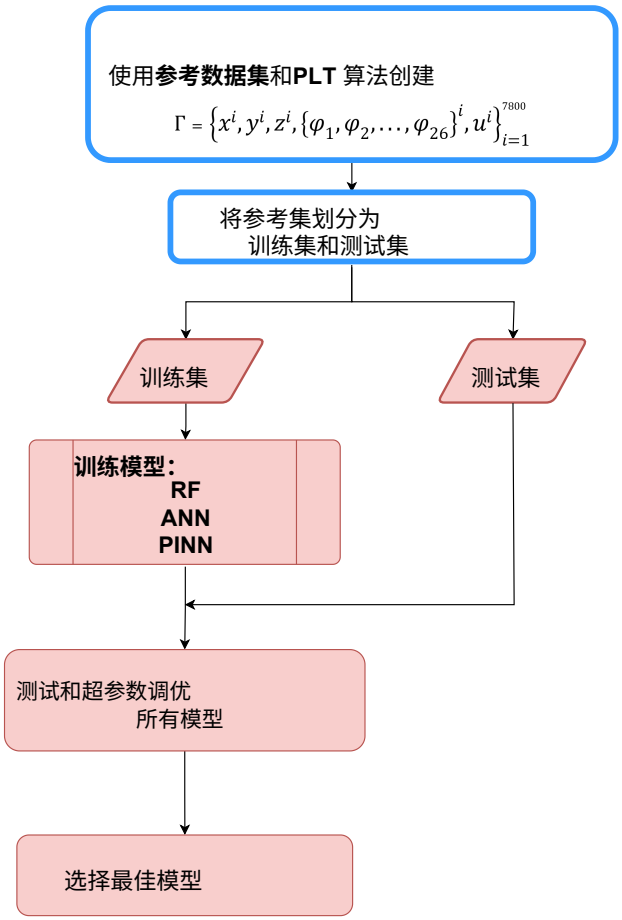


图 1：方法流程图。蓝色部分：准备数据，基于 PLT算法创建参考数据集。红色部分：模型训练过程。首先，将参考集划分为训练集和测试集；然后，在训练集上应用 RF、人工神经网络和物理信息神经网络模型；调整超参数后，选出最佳模型。

2.1 泊松方程到拉普拉斯方程变换（PLT）

根据 PLT算法，电势分为两部分：奇异势（ϕ_{si}）和光滑势（ϕ_{sm}）；$\phi\left(\vec{x}\right) = \phi_{si}\left(\vec{x}\right) + \phi_{sm}\left(\vec{x}\right)$。需要注意的是，这里的光滑部分是具有修正边界条件的拉普拉斯方程的解，

$$\nabla^2 \phi_{sm}(\vec{x}) = 0, \tag{2}$$

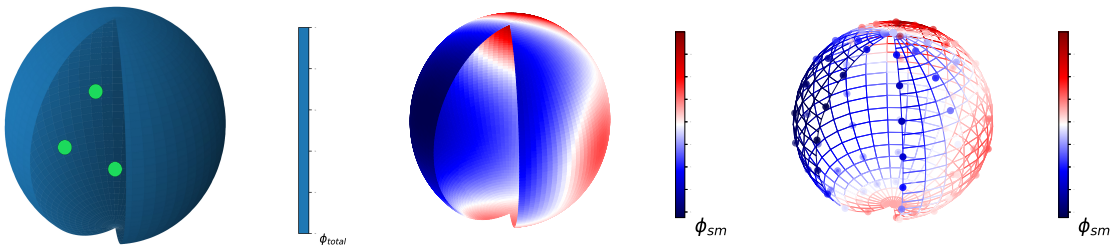
而 ϕ_{si} 遵循著名的库仑定律

$$\phi_{si}(\vec{x}) = \sum_{i=1}^{N} \frac{q_i}{4\pi\epsilon_0 \|\vec{x} - \vec{x}_i\|}. \tag{3}$$

可以看出， ϕ_{sm} 的修正边界条件表示为

$$\phi_{sm}\|_{\vec{x}_{bc}} = \phi\|_{\vec{x}_{bc}} - \phi_{si}\|_{\vec{x}_{bc}}, \tag{4}$$

其中， $\phi\|^{\vec{x}_{bc}}$ 对应于边界上的初始电势。最后，借助 PLT算法，我们可以通过新的修正边界条件，将泊松方程转换为拉普拉斯方程；随后，以这些修正边界条件作为输入参数、以光滑势作为输出，训练一个机器学习模型。之后，将奇异势与预测的光滑势相加，即可得到总势。采用 PLT算法的优点是输入数据的数量固定，因为输入数据的数量与点电荷粒子的数量无关。



(a) (b) (c)

图 2：PLT方法示意图：a）包含点电荷的主系统；b）不含任何点电荷且边界经过修正的新系统；c） N_b 边界上的点，用作模型的输入。

2.2 数据工程

要训练出高精度模型，优质的训练集至关重要。在本研究中，参考集为$\Gamma = \left\{x^i, y^i, z^i, \vec{\varphi_{bc}^i}, \phi^i\right\}_{i=1}^{N}$，其中输入由$\{x, y, z, \vec{\varphi}_{bc} = \{\phi_1, \phi_2, ..., \phi_{N_b}\}\}$组成， ϕ 为目标。x, y, z 是容器中某个点的坐标，我们希望计算该点的势； ϕ 是该点势的数值，而$\{\phi_1, \phi_2, ..., \phi_{N_b}\}$是边界上 N_b 个点的边界条件。首先，在容器中选取 N_p 个位置，以预测这些位置处的势。实际上，对于每个边界条件$\{\phi_1, \phi_2, ..., \phi_{N_b}\}$，我们都希望计算 N_p 个点所在位置的势。然后，可针对 N_q 种不同的边界条件构建参考集。因此，参考集由 $N = N_q \times N_p$ 个样本组成，可划分为训练集和测试集。在本例中，容器为球体，我们还设定了 $N_p = 78$, $N_b = 26$和 $N_q = 100$。因此，参考集包含100种不同的边界条件；对于每个边界条件$\{\phi_1, \phi_2, ..., \phi_{26}\}$，球体内有78个我们希望计算其势的点。我们使用镜像电荷法的解（式5）计算参考集的目标：

$$\phi(\vec{x}) = \frac{1}{4\pi\epsilon_0}\{\frac{q}{\|\vec{x} - \vec{x}_q\|} + \frac{q'}{\|\vec{x} - \vec{x_{q'}}\|}\}, \quad q' = -\frac{rq}{a}, \quad \vec{x_{q'}} = \frac{a^2}{r}\frac{\vec{x}_q}{\|\vec{x}_q\|}, \tag{5}$$

其中， a 是导电球壳的半径， r 是点电荷 q 到其中心的距离。势的数值很小（$\sim 10^{-9}$），在计算过程中会导致明显的舍入误差；因此，使用电子在距离其 $1m$ 处的势 $1.44 \times 10^{-9}[V]$作为单位

以使方程 5 无量纲化。我们从参考集中随机选取了 5000 个样本和 1000 个样本，分别构建训练集和测试集。训练集和测试集不包含任何相同的样本。此外，最佳模型能够充分预测测试样本的势，以及训练集和测试集中具有不同边界条件的样本的势。因此，为了更好地评估模型，我们准备了一个外推集，其中包含具有 55 种不同边界条件的 1000 个样本。

2.3 机器学习算法

本研究采用了三种不同的监督学习方法，并根据 2.4 中介绍的指标评估了它们的回归精度。我们主要采用物理信息神经网络（PINN，Raissi 等人 [2019]），但为了将结果与其他机器学习算法进行比较，也使用了随机森林（RF，Breiman [2001]）和人工神经网络（ANN）。本文简要介绍了所有模型，对超参数进行了微调，并报告了模型的性能。Scikit-learn Pedregosa 等人 [2011],、TensorFlow Abadi [2016], 、Keras Chollet [2015], 和 NumPy Walt 等人 [2011] 是本项目使用的所有 Python 库。

2.3.1 随机森林（RF）

RF 是回归问题中最受欢迎的机器学习算法之一，原因有很多；但在本项目中，选择该模型是因为

a) 学习速度快。
b) 不易过拟合。
当训练样本的表现完美而测试样本的表现不佳时，就会出现过拟合。随机森林是一种集成模型，它通过对许多互不相关的树取平均值，来确定目标数据集的预测势。虽然每棵树都是弱学习器，但将许多树组合起来后，就能构成强学习器。随机森林通过为每棵树选择训练数据和特征的一个子集来随机化树。这里我们使用 scikit-learn Pedregosa 等人 [2011] 随机森林实现。

2.3.2 人工神经网络

典型的神经网络架构由输入层、多个隐藏层以及输出层组成，每层包含若干个神经元。总体而言：

- 输入层：输入层中的神经元就是输入特征。
- 隐藏层：隐藏层中每个神经元的值都是前一层神经元的线性组合，随后应用激活函数（式 6）；在大多数情况下，激活函数是非线性的。

$$a_n = \sigma_l \left(a_{n-1} \mathbf{w}_n + \mathbf{b}_n \right). \tag{6}$$

$\mathbf{n}$ 表示层数，$\mathbf{w}$ 和$\mathbf{b}$ 分别表示模型参数中的权重和偏置，σ_l 是基于 Goodfellow 等人 [2016]提出的激活函数。

- 输出层：输出层中的神经元即为模型目标，其值通过式 6 并使用线性激活函数计算。
- 损失函数：所有神经网络中都有一个函数，在训练阶段必须通过反向传播对模型参数进行最小化；通常，损失函数是实际值与预测值之间的均方误差。

$$\mathrm{Loss}(w) = MSE_d + \lambda \sum_w w^2, \tag{7}$$

$$MSE_d = \frac{1}{N} \sum_{\mathrm{i}=1}^{\mathrm{N}} \left[\mathrm{U} \left(\mathbf{X}_\mathrm{i}, \mathbf{w} \right) - \mathrm{T}_\mathrm{i} \right]^2. \tag{8}$$

U 和 T 分别表示预测输出值和真实目标值，X 表示输入数据，w 表示神经网络的参数、权重和偏置。公式 7 中的第一项是均方误差，而

第二项用于防止过拟合，即 $L2$ 正则化A. K. Connect 等人 [1992], 用于减小较大权重的影响。

2.3.3 物理信息神经网络

PINN Raissi 等人 [2019] 将拉普拉斯方程这一电磁系统的物理定律作为约束施加于神经网络。本文提出一种基于 PINN 的方法，用于求解边界条件可变的拉普拉斯方程。图 3 展示了该方法中神经网络布局的示意图。基于 PINN 的模型是经过修改的神经网络，其 $loss$函数为:

$$\text{Loss} = \lambda_1 MSE_d + \lambda_2 MSE_f + \lambda_3 MSE_b + \lambda_4 \sum_w w^2, \tag{9}$$

公式 9 的第一项和最后一项与公式 7 中典型神经网络的对应项相同。第二项对应控制物理方程，即拉普拉斯方程；第三项对应边界条件；

$$MSE_d = \frac{1}{N_d} \sum_{i=1}^{N_d} \|u(\mathbf{x}_d^i, \vec{\varphi_d^i}; w) - \phi_d^i\|^2, \tag{10}$$

$$MSE_f = \frac{1}{N_f} \sum_{i=1}^{N_f} \|f(\mathbf{x}_f^i, u_f^i; w)\|^2, \tag{11}$$

and

$$MSE_b = \frac{1}{N_b} \sum_{i=1}^{N_b} \|\mathcal{B}(\mathbf{x}_b^i, \vec{\varphi_b^i}, u_b^i; w)\|^2. \tag{12}$$

这里定义 $f(\mathbf{x},\ \mathrm{u}; w)$

$$\begin{aligned} f\left(\mathbf{x}, \mathrm{u}; w\right) &= 0, \qquad \mathrm{x} \in \Gamma_f \\ &= \nabla^2 u \\ &= \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \\ &= \frac{\partial w}{\partial x}\frac{\partial}{\partial w}\left(\frac{\partial w}{\partial x}\frac{\partial u}{\partial w}\right) + \frac{\partial w}{\partial y}\frac{\partial}{\partial w}\left(\frac{\partial w}{\partial y}\frac{\partial u}{\partial w}\right) + \frac{\partial w}{\partial z}\frac{\partial}{\partial w}\left(\frac{\partial w}{\partial z}\frac{\partial u}{\partial w}\right), \end{aligned} \tag{13}$$

采用狄利克雷边界条件

$$\begin{aligned} \mathcal{B}\left(\mathbf{x}, \vec{\varphi}, u; w\right) &= 0, \qquad \mathrm{x} \in \Gamma_b \\ &= \sum_{j=1}^{26} (u(\mathbf{x}^j, \vec{\varphi}; w) - \varphi_b^j). \end{aligned} \tag{14}$$

λ_1，λ_2，λ_3 在公式 9 中分别对应数据贡献项、拉普拉斯方程和边界损失项的权重系数。我们参考 Kag 等人的研究来确定权重系数。Kag 等人 [2022]。最后一句是 $L2$ 正则化项，A. K. Connect 等人 [1992]。请注意，采用 $\lambda_2 = \lambda_3 = 0.0$ 的模型正是上一小节所述的典型神经网络。

2.4 评估指标

不同算法在势能估计方面的性能评估取决于不同的指标,

$$\Delta\phi = \phi_{True} - \phi_{Pred}, \tag{15}$$

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (\Delta\phi)^2}, \tag{16}$$

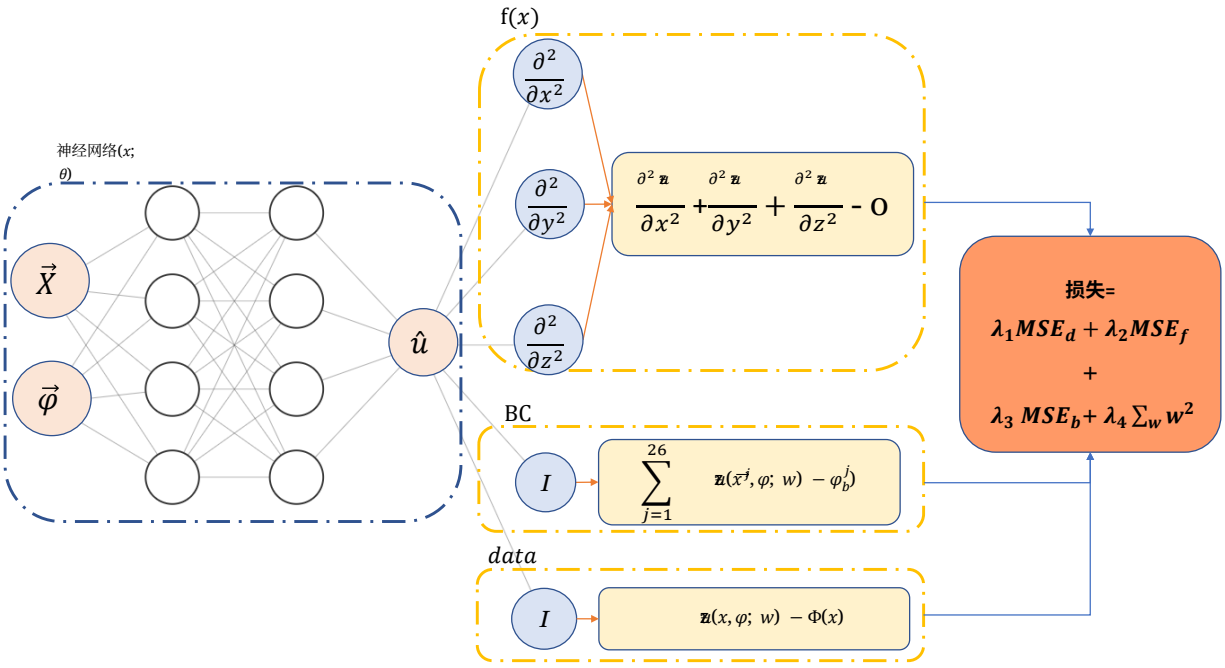


图 3：用于求解具有可变边界的拉普拉斯方程的物理信息神经网络方案

$$\mathrm{R}^2 = 1 - \frac{\sum_{i=1}^{n} \left(\phi_{\mathrm{True}} - \phi_{\mathrm{Pred}}\right)^2}{\sum_{i=1}^{n} \left(\phi_{\mathrm{True}} - \bar{\phi}_{\mathrm{True}}\right)^2}, \tag{17}$$

$$\mathrm{MSE} = < (\Delta\phi)^2 > . \tag{18}$$

其中， ϕ_{True}为真实势能， ϕ_{Pred}为预测势， ϕ_{True}为给定测试样本的真实势能均值。本研究采用散布 σ、 R^2 分数和 MSE 作为评估指标。

3 结果

本文预测球形导电容器中点电荷粒子的光滑势。首先，我们设置包含 5000 个样本的训练集和包含 1000 个样本的测试集（2.2），然后训练模型以预测光滑势。通过将光滑势和奇异势相加，我们可以计算总势（详见 2.1）。然而，为了将本文结果与 CAVIARBiagooi 等人 [2020], 的结果进行比较，我们研究的是光滑势。

3.1 随机森林

3.1.1 随机森林的超参数

我们针对唯一的超参数进行优化，即森林中的树的数量，该参数会影响随机森林模型的拟合效果。在图4中，我们绘制了测试集上 MSE （左图）和 R^2 得分（右图）随树的数量变化的曲线，以确定最优超参数。我们发现，树的数量设为 100 棵时最优，因为超过 100 棵后性能提升微乎其微。之后，我们使用 100 棵树训练随机森林模型。

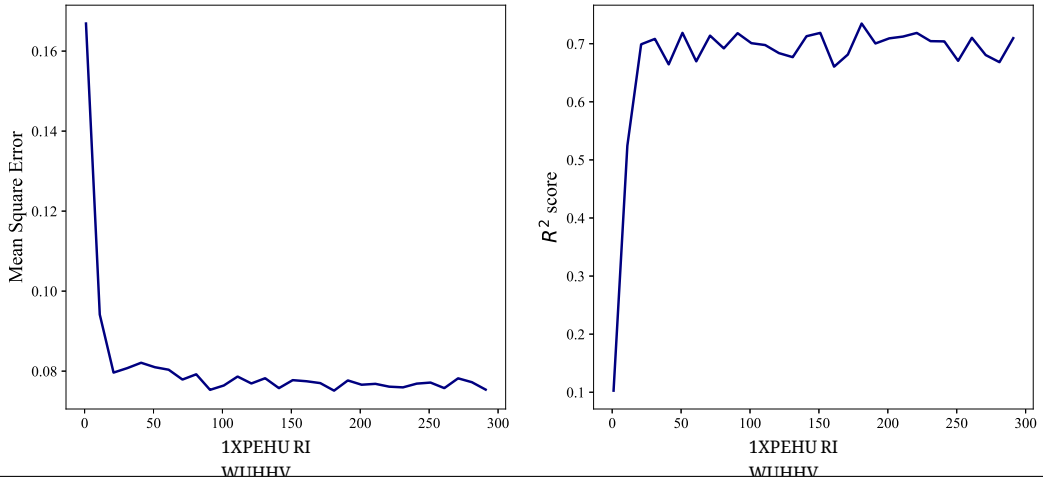
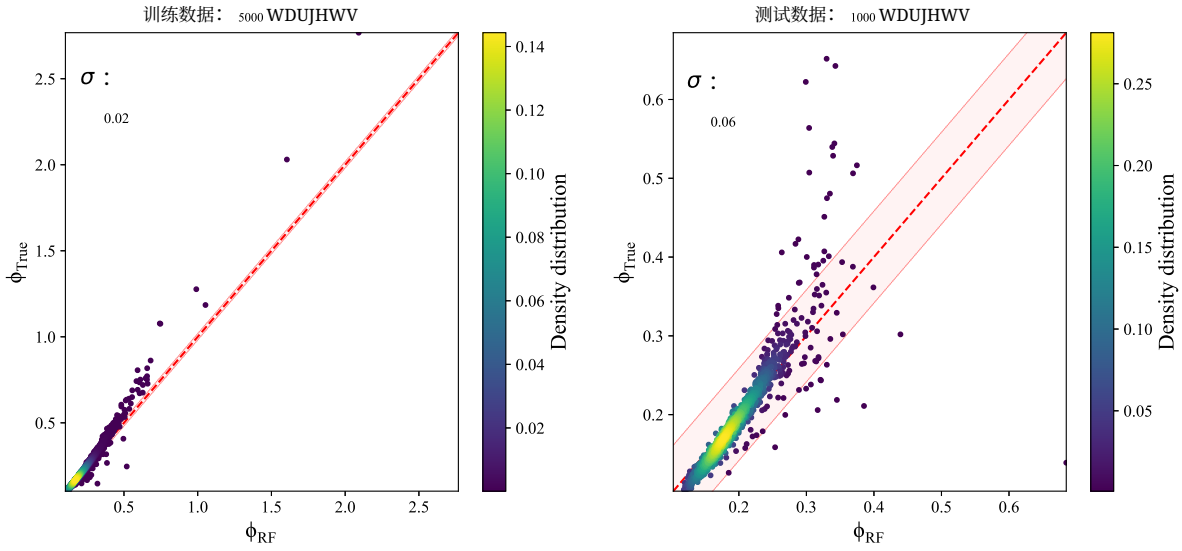


图4：1000个不同测试集样本的 MSE （左）和 R^2 分数（右）

3.1.2 随机森林预测

图5展示了包含 100 棵树的随机森林模型。结果表明，当势的数值小于 0。3（$\phi_{True} < 0.3$）时，预测效果可以接受；但在 $\phi_{True} >= 0.3$ 的情况下，预测精度不足。图5中的曲线比较了训练数据集（左图）和测试数据集（右图）上的真实势能 ϕ_{True} 与随机森林模型预测势 ϕ_{RF}。随机森林方法的速度相对较快；然而，它仅适用于预测势平滑且数值相对较小的情况，不适用于点电荷粒子的情况（在电荷所在位置，势的梯度和数值都很高）。此外，由于势的梯度较大，该方法无法准确预测边界附近的势。



(a) (b)

图5：随机森林模型的势能估计：a）展示包含5000个样本的训练数据集及其$\sigma = 0.02$散布；b）展示包含1000个样本的测试数据集及其 $\sigma = 0.07$散布。红色虚线表示预测势等于真实势能的位置。粉色阴影区域标示势能误差的 1σ 散布。

3.2 基于物理信息神经网络的模型和神经网络

将基于物理信息神经网络的模型中的 λ_2 和 λ_3 都设为零，即可得到完全相同的神经网络模型。因此，我们将这两种模型一并研究，并在以下章节中同时报告它们的结果。

3.2.1 物理信息神经网络和神经网络的超参数

与随机森林模型不同，我们定义了几个超参数：神经元数量、层数、 λ_2和λ_3。为调整所有超参数，我们使用 L-BFGS-B 优化器（Liu 和 Nocedal [1989], ）训练模型，最多进行 100000 个训练轮次，直到模型的容差达到机器精度。除最后一层外，所有层均使用双曲正切激活函数。表 1 报告了在不同超参数取值下，预测势与相同势之间的 MSE； $\lambda_2 = [0, 0.1, 0.2, 0.3]$、 $\lambda_3 = [0, 0.1, 0.2, 0.3, 0.4]$、隐藏层数量$= [1, 3, 5, 7]$ 以及每个隐藏层的神经元数量$= [10, 30, 50]$，针对的是测试集中的 1000 个样本。如表 1 所示，只有一个隐藏层的模型无法很好地预测势。此外，每层有十个神经元的模型也无法取得良好效果。因此，表 2 和表 3 仅报告了具有 $[3, 5, 7]$ 层以及每层含有 $[30, 50]$ 个神经元的模型结果。我们选择 $\lambda_4 = 0.0001$ 以防止过拟合。

3.2.2 物理信息神经网络与神经网络预测

与表 1 相比，表 2 不仅报告了 MSE，还报告了测试集的 $R^2 score$。如表 2 所示，当 λ_2 和 λ_3 分别为 0.3,0.3 或 $0.2, 0.4$ 时，采用七层且每层 50 个神经元的模型效果更好。当 λ_2和 λ_3均为零时，该模型为标准神经网络，其表现不佳；这一点可从表2和图4中看出。

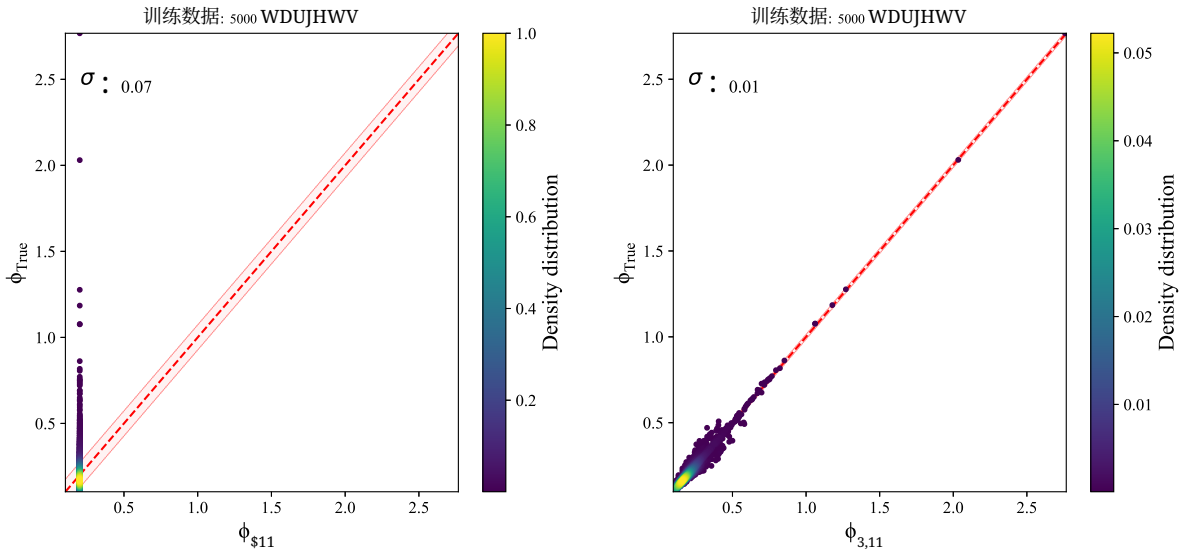


图6：在训练集的 5000 个样本上，经过最佳调优的（a）神经网络模型的势能估计及其 $\sigma = 0.07$ 散布，以及（b）物理信息神经网络模型的势能估计及其 $\sigma = 0.01$散布。红色虚线表示预测势等于真实势能的位置。粉色阴影区域表示势能误差的 1σ 散布。

图6中的两幅图比较了在训练集上，经过最佳调参的神经网络模型和物理信息神经网络模型的真实势与预测势；这两个模型均有7层，每层50个神经元， $\lambda_2 = 0.3$ 和 $\lambda_3 = 0.3$。可以看出，神经网络模型训练得并不好，而基于物理信息神经网络的模型能够准确预测势，散布为0.01。尽管基于物理信息神经网络的模型对训练集的预测效果良好，为了确认没有出现过拟合，我们还在测试集上评估了该模型，结果见图7。

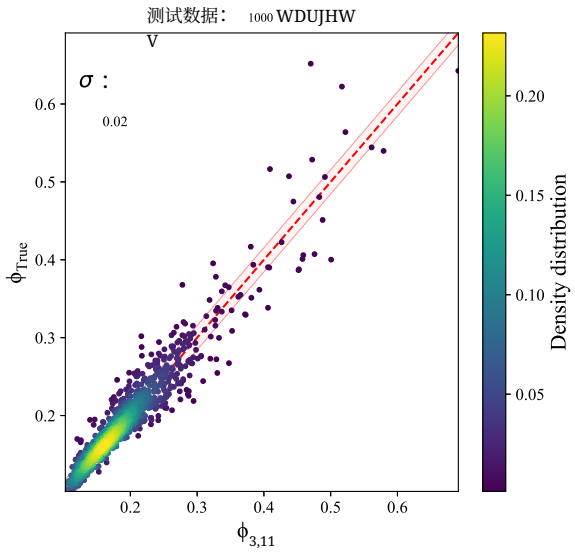


图 7：在测试集的 1000 个样本上，对调优效果最佳的物理信息神经网络模型进行势能估计，并散布显示 $\sigma = 0.02$、$MSE = 0.069$ 和 $R^2 score = 0.851$。红色虚线表示预测势等于真实势能的位置。粉色阴影区域标示了势能误差的 1σ 散布。

		MSE_{test}			MSE_{test}		
λ_2	λ_3	神经元=**10**	神经元=**30**	神经元=**50**	神经元=**10**	神经元=**30**	神经元=**50**
		隐藏层数量：1			隐藏层数量：5		
0.0	0.0	0.246	0.246	0.246	0.246	0.246	0.246
	0.1	0.162	0.145	0.124	**0.099**	0.145	**0.078**
	0.2	0.191	0.212	0.132	0.136	**0.062**	**0.072**
	0.3	0.175	0.159	0.171	0.14	**0.069**	0.124
	0.4	0.24	0.171	0.18	0.191	**0.064**	**0.076**
0.1	0.0	0.246	0.247	0.247	0.246	0.246	0.246
	0.1	0.222	0.143	0.117	0.107	0.245	0.192
	0.2	0.201	0.165	0.129	0.243	**0.079**	**0.062**
	0.3	0.22	0.159	0.16	0.112	**0.094**	**0.071**
	0.4	0.249	0.177	0.243	0.205	**0.088**	**0.07**
0.2	0.0	0.246	0.246	0.246	0.246	0.246	0.246
	0.1	0.158	0.144	0.154	0.235	0.161	**0.083**
	0.2	0.225	0.174	0.174	0.198	**0.097**	**0.071**
	0.3	0.223	0.143	0.158	0.192	**0.081**	**0.075**
	0.4	0.197	0.185	0.189	0.216	**0.065**	0.118
0.3	0.0	0.246	0.246	0.246	0.246	0.246	0.246
	0.1	0.21	0.132	0.129	0.245	**0.088**	**0.098**
	0.2	0.199	0.139	0.157	0.209	0.117	**0.065**
	0.3	0.216	0.241	0.202	0.182	0.168	**0.08**
	0.4	0.225	0.182	0.196	0.239	**0.083**	**0.097**
		隐藏层数量：3			隐藏层数量：7		
0.0	0.0	0.246	0.246	0.246	0.246	0.246	0.246
	0.1	0.157	0.104	**0.086**	0.244	0.237	0.247
	0.2	0.151	**0.088**	**0.086**	0.238	0.087	0.105
	0.3	0.13	**0.078**	**0.089**	0.244	**0.07**	**0.067**
	0.4	0.202	0.173	0.102	0.188	**0.068**	**0.063**
0.1	0.0	0.246	0.246	0.246	0.246	0.246	0.246
	0.1	**0.099**	**0.089**	0.132	0.244	0.244	0.244
	0.2	0.198	**0.093**	**0.077**	0.23	0.221	**0.069**
	0.3	0.188	**0.088**	**0.083**	0.224	0.223	**0.082**
	0.4	0.262	0.104	**0.079**	0.213	**0.072**	**0.074**
0.2	0.0	0.246	0.246	0.246	0.246	0.246	0.246
	0.1	0.119	0.146	**0.082**	0.244	0.244	0.245
	0.2	0.172	0.09	**0.086**	0.244	0.244	**0.071**
	0.3	0.179	0.084	**0.085**	0.225	0.251	**0.08**
	0.4	0.197	0.185	**0.086**	0.256	0.166	**0.067**
0.3	0.0	0.246	0.246	0.246	0.246	0.246	0.246
	0.1	0.17	0.125	**0.082**	0.244	0.244	0.244
	0.2	0.199	0.117	**0.096**	0.244	**0.083**	**0.08**
	0.3	0.224	0.107	**0.088**	0.202	0.235	**0.069**
	0.4	0.199	0.222	0.152	0.236	**0.097**	**0.075**

表1： MSE 针对不同的 λ_2值、 λ_3和不同数量的隐藏层以及每个隐藏层中的神经元，物理信息神经网络对测试集中的1000个不同样本所预测的势与精确势之间的 $\phi(x)$，以及 $\lambda_4 = 0.0001$ 固定时的 $\lambda_1 = 1 - (\lambda_2 + \lambda_3 + \lambda_4)$。在此表中，粗体数字表示$MSE < 0.1$。

		Num_{layer}=3		Num_{layer}=5		Num_{layer}=7	
λ_2	λ_3	MSE	R^2	MSE	R^2	MSE	R^2
				Num_{neuron}=30			
	0.0	0.246	0	0.246	0	0.246	0
	0.1	0.104	0.774	0.145	0.365	0.237	-14.7
0.0	0.2	0.088	0.85	0.062	0.889	0.087	0.807
	0.3	0.078	0.826	0.069	0.884	**0.07**	**0.904**
	0.4	0.173	0.551	0.064	0.863	0.068	0.868
	0.0	0.246	0	0.246	0	0.246	0
	0.1	0.089	0.796	0.245	-1478	0.244	0
0.1	0.2	0.093	0.825	0.079	0.852	0.221	-2.40
	0.3	0.088	0.793	0.094	0.714	0.223	0.094
	0.4	0.104	0.719	0.088	0.793	0.072	0.887
	0.0	0.246	0	0.246	0	0.246	0
	0.1	0.146	0.64	0.161	0.566	0.244	0
0.2	0.2	0.09	0.769	0.097	0.75	0.244	-3167
	0.3	0.084	0.819	0.081	0.857	0.251	-2.32
	0.4	0.185	0.521	0.065	0.874	0.166	0.07
	0.0	0.246	0	0.246	0	0.246	0
	0.1	0.125	0.626	0.088	0.747	0.244	0
0.3	0.2	0.117	0.666	0.117	0.689	0.083	0.863
	0.3	0.107	0.67	0.168	0.59	0.235	0
	0.4	0.222	0.25	0.083	0.837	0.097	0.804
				Num_{neuron}=50			
	0.0	0.246	0	0.246	0	0.246	0
	0.1	0.086	0.832	0.078	0.865	0.247	-254
0.0	0.2	0.086	0.838	0.072	0.873	0.105	0.711
	0.3	0.089	0.767	0.124	0.66	0.067	0.875
	0.4	0.102	0.734	0.076	0.846	0.063	0.849
	0.0	0.246	0	0.246	0	0.246	0
	0.1	0.132	0.64	0.192	-0.48	0.244	0
0.1	0.2	0.077	0.85	0.062	0.888	0.069	0.897
	0.3	0.083	0.837	0.071	0.895	0.082	0.774
	0.4	0.079	0.847	0.07	0.887	0.074	0.863
	0.0	0.246	0	0.246	0	0.246	0
	0.1	0.082	0.756	0.083	0.859	0.245	-1774
0.2	0.2	0.086	0.796	0.071	0.88	**0.071**	**0.908**
	0.3	0.085	0.799	0.075	0.888	0.08	0.87
	0.4	0.086	0.8	0.118	0.727	**0.067**	**0.902**
	0.0	0.246	0	0.246	0	0.246	0
	0.1	0.082	0.75	0.098	0.753	0.244	0
0.3	0.2	0.096	0.777	0.065	0.894	0.08	0.798
	0.3	0.088	0.785	0.08	0.834	**0.069**	**0.902**
	0.4	0.152	0.557	0.097	0.818	0.075	0.867

表 2: MSE，以及 R^2 预测势与精确势之间的分数 $\phi(x)$，对应不同的 λ_2 和 λ_3，以及物理信息神经网络中不同的隐藏层数和每个隐藏层的神经元数，基于测试集中的 1000 个不同样本。此处，$\lambda_4 = 0.0001$ 固定为 $\lambda_1 = 1 - (\lambda_2 + \lambda_3 + \lambda_4)$。在此表中，粗体数字表示满足 $MSE < 0.1$ 和 $R^2 score > 0.9$ 的情况。

		Num_{layer}=3		Num_{layer}=5		Num_{layer}=7	
λ_2	λ_3	MSE	R^2	MSE	R^2	MSE	R^2
				Num_{neuron}=30			
	0.0	0.281	0	0.281	0	0.281	0
	0.1	0.14	0.71	0.148	0.364	0.275	-15.28
0.0	0.2	0.118	0.764	0.077	0.847	0.126	0.646
	0.3	0.104	0.737	0.09	0.782	0.089	0.835
	0.4	0.208	0.587	0.072	0.842	0.083	0.822
	0.0	0.281	0	0.281	0	0.281	0
	0.1	0.105	0.808	0.28	-1801	0.28	0
0.1	0.2	0.112	0.775	0.09	0.845	0.259	-2.137
	0.3	0.105	0.793	0.123	0.697	0.264	0.07
	0.4	0.116	0.735	0.12	0.684	0.092	0.813
	0.0	0.281	0	0.281	0	0.281	0
	0.1	0.183	0.601	0.201	0.484	0.28	0
0.2	0.2	0.118	0.739	0.114	0.68	0.28	-3620
	0.3	0.095	0.829	0.109	0.799	0.275	-1.82
	0.4	0.209	0.586	0.078	0.851	0.205	0.183
	0.0	0.281	0	0.281	0	0.281	0
	0.1	0.154	0.663	0.105	0.743	0.28	0
0.3	0.2	0.13	0.74	0.139	0.688	0.109	0.725
	0.3	0.143	0.565	0.2	0.604	0.271	0.012
	0.4	0.269	0.356	0.158	0.364	0.118	0.726
				Num_{neuron}=50			
	0.0	0.281	0	0.281	0	0.281	0
	0.1	0.116	0.764	0.112	0.769	0.28	-310
0.0	0.2	0.1	0.837	0.092	0.829	0.135	0.598
	0.3	0.101	0.79	0.145	0.719	0.083	0.843
	0.4	0.117	0.706	0.096	0.801	0.106	0.662
	0.0	0.281	0	0.281	0	0.281	0
	0.1	0.149	0.657	0.227	-0.465	0.28	0
0.1	0.2	0.121	0.658	0.077	0.832	0.09	0.824
	0.3	0.108	0.719	0.096	0.817	0.102	0.734
	0.4	0.117	0.716	0.085	0.843	0.093	0.842
	0.0	0.281	0	0.281	0	0.281	0
	0.1	0.094	0.782	0.112	0.777	0.28	-2164
0.2	0.2	0.114	0.758	0.094	0.785	0.091	0.854
	0.3	0.101	0.762	0.093	0.818	0.104	0.771
	0.4	0.124	0.644	0.143	0.703	0.088	0.849
	0.0	0.281	0	0.281	0	0.281	0
	0.1	0.097	0.772	0.115	0.727	0.28	0
0.3	0.2	0.112	0.768	0.096	0.807	0.167	0.363
	0.3	0.11	0.777	0.113	0.682	**0.089**	**0.851**
	0.4	0.177	0.563	0.114	0.76	0.103	0.741

表 3：对于外推集中的 1000 个不同样本，在不同的 λ_2 和 λ_3 取值、不同的隐藏层数量以及物理信息神经网络中每个隐藏层的神经元数量下，预测势 $\phi(x)$ 与精确势之间的 MSE 和 R^2 得分。此处，$\lambda_4 = 0.0001$ 固定为 $\lambda_1 = 1 - (\lambda_2 + \lambda_3 + \lambda_4)$。表中粗体数字表示我们基于物理信息神经网络的模型所采用的最佳超参数。

3.3 比较

我们评估了随机森林、神经网络和物理信息神经网络，用于估计导电壁包围的点电荷粒子的势。根据图6，神经网络的训练效果不佳，而基于随机森林和物理信息神经网络的模型能够准确预测势。然而，随机森林无法很好地估计 $\phi_{True} > 0.3$。除此之外，最佳模型不仅能够估计训练集和测试集中的势，还能估计不属于训练集或测试集的点电荷粒子的势。因此，我们使用外推样本评估调优效果最佳的物理信息神经网络模型和随机森林，结果见表 3。如图8所示，基于物理信息神经网络的模型对新带电粒子的势的预测优于随机森林模型，其中物理信息神经网络对 $\phi_{True} > 0.3$ 的预测远优于随机森林。

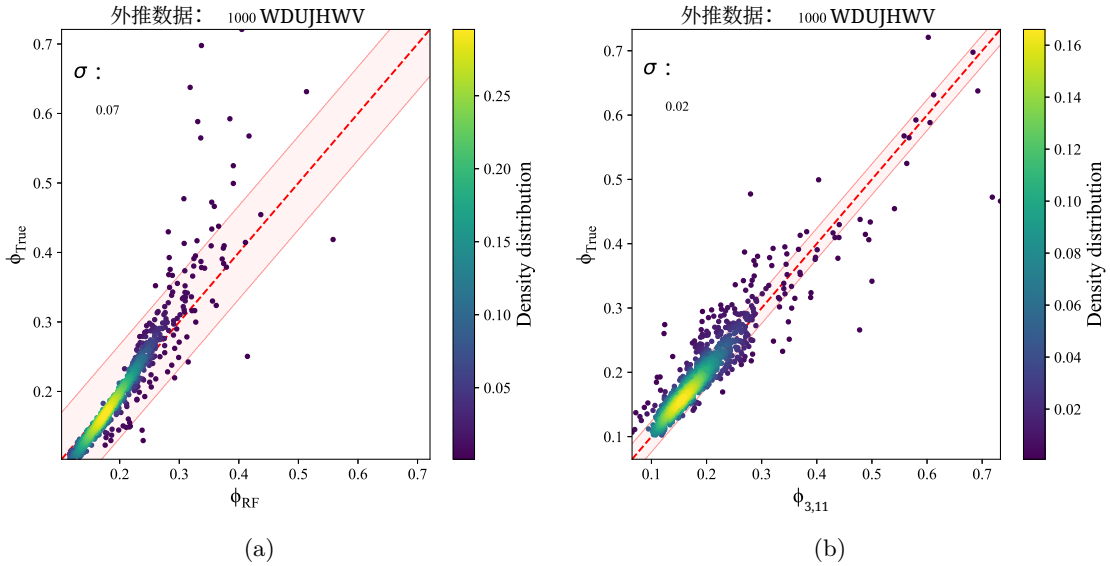


图8：调优效果最佳的 (a) 随机森林对 $\sigma = 0.07$ 的势能估计及其散布，以及 (b) 基于物理信息神经网络的模型对 $\sigma = 0.02$ 的势能估计及其散布，均针对外推集中的 1000 个样本。红色虚线表示预测势等于真实势能的位置。粉色阴影区域标示势能误差散布的 1σ 。

3.4 泛化（多电荷粒子）

为评估泛化能力，我们使用 $\lambda_2 = 0.3$ 和 $\lambda_3 = 0.3$ 对基于物理信息神经网络的模型进行测试，考察多个带电粒子被导电边界包围的情况。由于拉普拉斯方程是线性的，我们分别预测每个带电粒子的势，再将相应的预测势叠加，计算总势。随后，我们报告预测的光滑势与精确光滑解之间的 MSE；该精确解通过镜像电荷法计算得出。图 9 显示了 MSE与带电粒子数量 N之间的关系。正如预期， MSE与带电粒子的数量无关。因此，这意味着我们也可以将该方法用于包含任意数量粒子的问题。

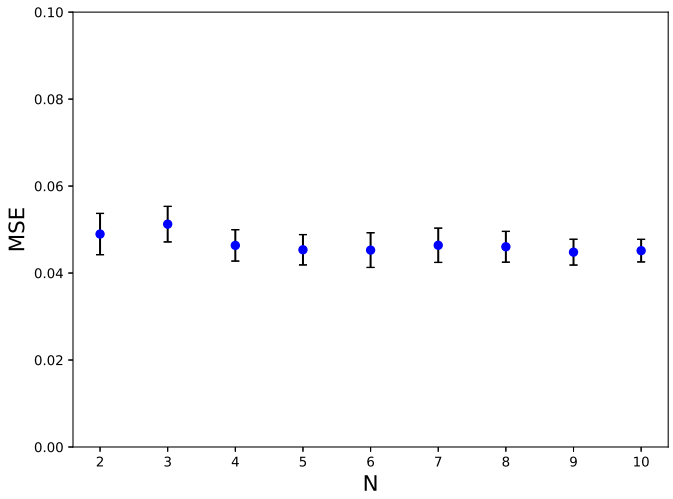


图 9： MSE 在 100 个不同问题中，真实势与预测势之间的差异随带电粒子数量的变化

4 结论

本研究训练了一个机器学习模型，用于预测被导电边界包围的带电组分的光滑势。在这种情况下，借助PLT算法，可将预测的光滑势与奇异势相加，从而轻松计算总势。参考集由解析解构成，即镜像电荷法的解；该参考集被划分为包含5000个样本的训练集和包含1000个样本的测试集。为检验模型的准确性，我们另设了一个名为外推集的数据集，其中包含1000个具有不同边界条件的样本，这些边界条件未出现在训练集甚至测试集中。我们的主要结论总结如下：

- 我们发现，基于物理信息神经网络的模型比随机森林和神经网络模型训练效果更好。随机森林无法预测高势；另一方面，神经网络则完全无法得到良好训练。
- 我们的基于物理信息神经网络的模型能够预测测试集的势，达到 $MSE = 0.069$、$R^2 score = 0.902$，散布为 $\sigma = 0.02$。它也能够预测外推集的势，达到 $MSE = 0.089$、$R^2 score = 0.851$，散布为 $\sigma = 0.02$。

- 由于拉普拉斯方程是线性方程，训练后的模型可以通过将每个粒子的预测势相加，来预测多个带电粒子的势。此外，我们表明，多个粒子的 MSE 与粒子数量无关。

参考文献

John R Miller 和 Patrice Simon。用于能源管理的电化学电容器。科学，321(5889)：651–652，2008年。

Mathieu Salanne、Benjamin Rotenberg、Katsuhiko Naoi、Katsumi Kaneko、P-L Taberna、Clare P Grey、Bruce Dunn 和 Patrice Simon。用于构建更优超级电容器的高效储能机制。自然能源，1(6)：1–10，2016年。

Patrice Simon 和 Yury Gogotsi。电化学电容器材料，2008年。ISSN 14761122。

John David Jackson。Jackson——经典电动力学，1962年。ISSN 00029505。

金建明。《电磁学中的有限元法》第3版。《化学信息与建模杂志》，2014年。

G. W. S.、吉恩·H·戈卢布和查尔斯·F·范·洛恩。矩阵计算。《计算数学》，56，1991年。ISSN 00255718。doi:10.2307/2008552。

Sandeep Tyagi、Axel Arnold 和 Christian Holm。ICMMM2D：一种通过像电荷求和纳入平面介电界面的精确方法。《化学物理学杂志》，127，2007年。ISSN 00219606。doi:10.1063/1.2790428。Sandeep Tyagi、Axel Arnold 和 Christian Holm。采用像电荷的静电层修正：一种用于处理具有介电界面的平板 2D+h 系统的线性标度方法。《化学物理学杂志》，129，2008年。ISSN 00219606。doi:10.1063/1.3021064。Sandeep Tyagi、Mehmet Süzen、Marcello Sega、Marcia Barbosa、Sofia S. Kantorovich 和 Christian Holm。一种用于计算任意介电边界上感应电荷的迭代式快速线性标度方法。《化学物理学杂志》，132，2010年。ISSN 00219606。doi:10.1063/1.3376011。S. Kesselheim、M. Sega 和 C. Holm。icc* 算法：一种将介电边界效应纳入分子动力学模拟的快速方法。*arXiv:1003.1271*，2010年。Axel Arnold、Olaf Lenz、Stefan Kesselheim、Rudolf Weeber、Florian Fahrenberger、Dominic Roehm、Peter Košovan 和 Christian Holm。Espresso 3.1：用于粗粒化模型的分子动力学软件。第89卷，LNCSE，2013年。doi:10.1007/978-3-642-32979-1_1。Stewart K. Reed、Oliver J. Lanning 和 Paul A. Madden。离子液体与模型金属电极之间的电化学界面。《化学物理学杂志》，126，2007年。ISSN 00219606。doi:10.1063/1.2464084。Samare Rostami、S Alireza Ghasemi 和 Ehsan Nedaaee Oskoee。一种用于计算平行金属板约束带电粒子静电相互作用的高精度、高效率算法。《化学物理学杂志》，145(12):124118，2016年。Morad Biagooi、Mohammad Samanipour、S. Alireza Ghasemi 和 Seyedehsan Nedaaee Oskoee。Caviar：用于导电边界环绕环境中带电粒子的仿真软件包。《*AIP Advances*》，10，2020年。ISSN 21583226。doi:10.1063/1.5140052。Tao Shan、Wei Tang、Xunwang Dang、Maokun Li、Fan Yang、Shenheng Xu 和 Ji Wu。基于深度学习技术的泊松方程快速求解器研究。《*IEEE* 天线与传播汇刊》，68(9):6725–6733，2020年。doi:10.1109/TAP.2020.2985172。M. Raissi、P. Perdikaris 和 G. E. Karniadakis。物理信息神经网络：一种用于求解涉及非线性偏微分方程的正问题与逆问题的深度学习框架。《计算物理学杂志》，378，2019年。ISSN 10902716。doi:10.1016/j.jcp.2018.10.045。Leo Breiman。随机森林。《机器学习》，45，2001年。ISSN 08856125。doi:10.1023/A:1010933404324。Fabian Pedregosa、Gael Varoquaux、Alexandre Gramfort、Vincent Michel、Bertrand Thirion、Olivier Grisel、Mathieu Blondel、Peter Prettenhofer、Ron Weiss、Vincent Dubourg、Jake Vanderplas、Alexandre Passos、David Cournapeau、Matthieu Brucher、Matthieu Perrot 和 Édouard Duchesnay。Scikit-learn：Python 机器学习。《机器学习研究杂志》，12，2011年。ISSN 15324435。Martín Abadi。TensorFlow：大规模学习函数。《*ACM SIGPLAN* 通知》，51，2016年。ISSN 0362-1340。doi:10.1145/3022670.2976746。François Chollet。Keras：Python 深度学习库。*Keras.Io*，2015年。Stéfan Van Der Walt、S. Chris Colbert 和 Gaël Varoquaux。NumPy 数组：一种高效数值计算的数据结构。《科学与工程计算》，13，2011年。ISSN 15219615。doi:10.1109/MCSE.2011.37。Ian Goodfellow、Yoshua Bengio 和 Aaron Courville。《深度学习》。麻省理工学院出版社，2016年。 `http://www.deeplearningbook.org`。a. K. Connect、a. Krogh 和 J. a. Hertz。一种简单的权重衰减方法可以改善泛化。《神经信息处理系统进展》，4，1992年。Vijay Kag、Kannabiran Seshasayanan 和 Venkatesh Gopinath。用于二维湍流的物理与数据驱动神经网络。*arXiv* 预印本 *arXiv:2203.02555*，2022年。Dong C Liu 和 Jorge Nocedal。关于用于大规模优化的有限内存 BFGS 方法。《数学规划》，45(1):503–528，1989年。