

轴类感应淬火机器学习项目报告

Flos

2026 年 1 月 6 日

摘要

本项目旨在利用机器学习预测轴类感应淬火过程中的温度场与相变场演化。

构建了包含 64 组不同频率与电流参数的 COMSOL 有限元仿真数据集，并提出了一种名为 Parallel U-FNO 的双流并行神经网络架构。该模型创新性地结合了傅里叶神经算子 (FNO) 在捕捉全局热传导特征上的优势与 U-Net 在处理局部相变边界上的特长。实验结果表明，Parallel U-FNO 在测试集上的整体相对 L2 误差仅为 1.24%，能够精确预测温度场演变及马氏体硬化层的分布，且计算速度较传统 FEM 提升了数个数量级。本研究验证了神经算子在复杂多物理场耦合问题中的应用潜力，为智能热处理工艺设计提供了新的技术路径。

目录

1 引言	3
2 数据来源与描述	3
2.1 仿真模型构建	3
2.2 网格离散化策略	3
2.3 物理场配置与耦合	4
2.4 数据生成策略	5
2.5 数据采样与特征描述	5
3 数据预处理与数据集构建	6
3.1 原始数据解析与清洗	7
3.2 张量重构与几何掩码	7
3.3 特征工程与归一化	7
3.4 智能边界保护划分	8

4 模型架构	9
4.1 总体架构设计	9
4.2 子模块详解	9
4.2.1 FNO 分支	9
4.2.2 U-Net 分支	10
4.3 特征融合与双头机制	10
4.4 损失函数设计	12
4.4.1 基础数值损失	13
4.4.2 边界增强损失	13
4.4.3 物理约束的实现	13
4.4.4 几何掩码机制	13
5 模型训练	13
5.1 实现细节	13
5.2 超参数设置	14
5.3 训练策略	14
6 实验结果与分析	14
6.1 模型收敛性分析	15
6.2 定量评估指标	15
6.3 物理场预测可视化	15
6.4 硬化层预测精度	16
6.5 不足	17
7 结论	18

1 引言

本项目是机器学习课程的项目，通过机器学习预测轴类感应淬火的温度场和相变场。相关代码及数据集已开源至 GitHub¹。

2 数据来源与描述

2.1 仿真模型构建

本项目的数据集通过 COMSOL Multiphysics 6.4 有限元仿真软件生成，基于官方给出的案例的“圆柱销感应淬火”模型（Induction Hardening of a Cylindrical Pin）进行了二次开发与优化。

几何模型：模型采用二维轴对称（2D Axisymmetry）结构以提高计算效率。工件为一个长度 30 mm、直径 10 mm 的钢制圆柱销，两端设有 1 mm 的倒角。

材料属性：工件材料设定为亚共析铁素体-珠光体钢。初始微观组织设定为 50% 铁素体与 50% 珠光体的混合物。材料的热学、电学及机械属性均随温度变化，数据来源于 JMatPro® 数据库。

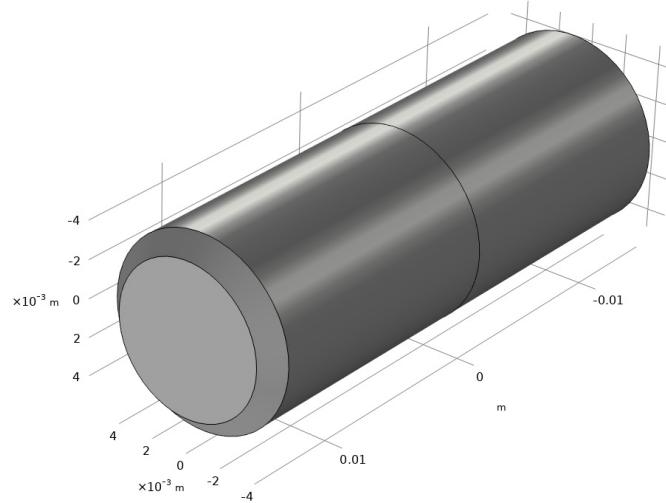


图 1：几何模型示意图

2.2 网格离散化策略

考虑到感应加热过程涉及高频电磁场（最高 120 kHz）引起的显著趋肤效应（Skin Effect），本研究在空间离散化上沿用了原模型的高精度网格设置。

¹<https://github.com/Flower101010/Induction-Hardening-ML>

- **网格划分:** 对工件 (Pin) 区域采用了“极细化”的预定义网格尺寸，以保证有限元计算的收敛性与精度。
- **边界层处理:** 为了精确解析工件表层的电磁场与温度梯度，在工件表面施加了边界层网格，总厚度设定为 1 mm。这种处理确保了感应电流集中的表层区域具有足够高的网格密度，从而生成高保真的物理场数据。

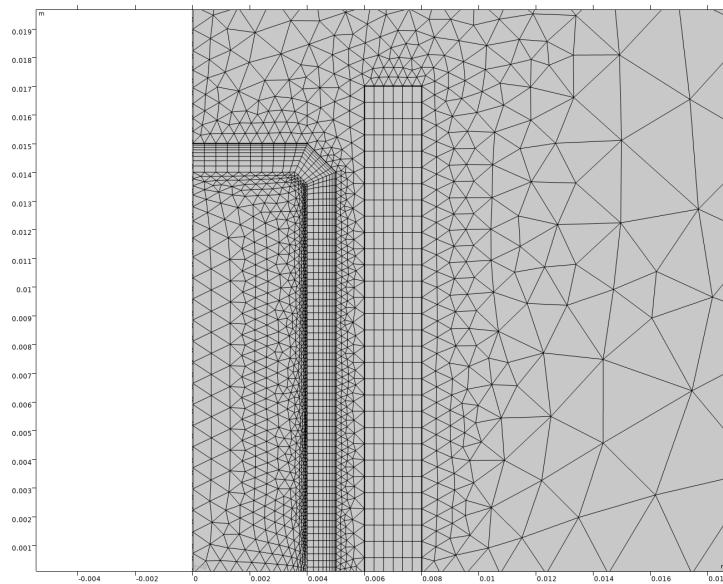


图 2: 网格局部放大图 (显示边界层网格)

2.3 物理场配置与耦合

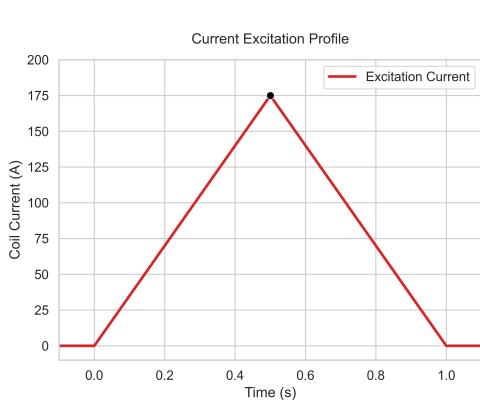
为了平衡计算精度与数据生成的效率，我们在原模型基础上进行了特定调整，移除了固体力学 (Solid Mechanics) 模块，专注于电磁场、传热场与微观组织相变的强耦合模拟：

- **电磁加热:** 使用“磁场” (Magnetic Fields) 接口模拟感应加热过程。励磁线圈被建模为 20 匝的均匀多匝线圈，通过高频交流电在工件表面诱导涡流产生焦耳热。
- **相变动力学:** 使用“奥氏体分解” (Austenite Decomposition) 接口模拟相变。
 - 加热阶段：采用线性相变模型模拟铁素体/珠光体向奥氏体的转变（奥氏体化），相变起始与终止温度分别为 723°C 与 900°C 。
 - 冷却阶段：模拟奥氏体分解为马氏体、贝氏体、铁素体或珠光体的过程，重点关注淬火后硬化层中马氏体的生成。
- **边界条件:** 模型考虑了向环境空气的对流换热与热辐射。淬火过程通过喷水冷却模拟，设定了随温度变化的强对流换热系数，仅在感应加热结束后激活。

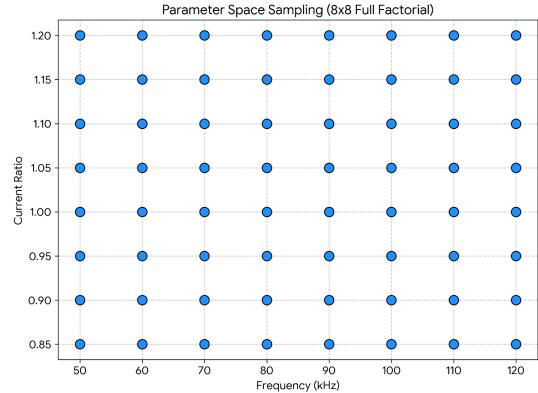
2.4 数据生成策略

为了构建适用于机器学习训练的高质量数据集，我们针对感应淬火工艺中的两个关键工艺参数——频率与电流强度——进行了全因子参数化扫描。

- **原始基准：**原模型基准频率为 120 kHz，基准峰值电流为 175 A（线性增加后减小，总励磁时间为 1 秒）。
- **参数空间：**
 - 频率 (f)：范围设定为 50 kHz 至 120 kHz，步长为 10 kHz，共 8 个水平。该范围涵盖了不同趋肤深度 (Skin Depth) 的加热工况。
 - 电流倍率 (I_{ratio})：引入电流倍率参数对基准电流曲线进行缩放，范围设定为 0.85 至 1.2，步长为 0.05，共 8 个水平。
- **样本总量：**通过组合上述参数，共进行了 $8 \times 8 = 64$ 组独立的仿真实验。



(a) 随时间变化的电流波形图



(b) 频率与电流倍率的参数空间分布

图 3: 激励信号与参数空间图

2.5 数据采样与特征描述

针对 64 组参数化扫描实验，我们导出了完整的瞬态演化过程数据，涵盖了从加热起始到冷却结束的全生命周期。

- **时间维度：**仿真总时长为 10 s。数据包含了 0 – 1 s 的感应加热阶段（电流线性上升后下降）以及后续的喷水冷却阶段。捕获完整的时间序列数据使得机器学习模型不仅能预测最终状态，还能学习相变动力学的演化规律。
- **输出变量：**对于每一个时间步，导出了全域节点的以下关键物理量作为训练样本：

- 温度场 (T): 驱动相变的核心热力学势。
- 马氏体体积分数 (ξ_M): 预测目标, 反映淬火硬化效果。
- 奥氏体体积分数 (ξ_A): 中间相变量, 连接加热与冷却过程。

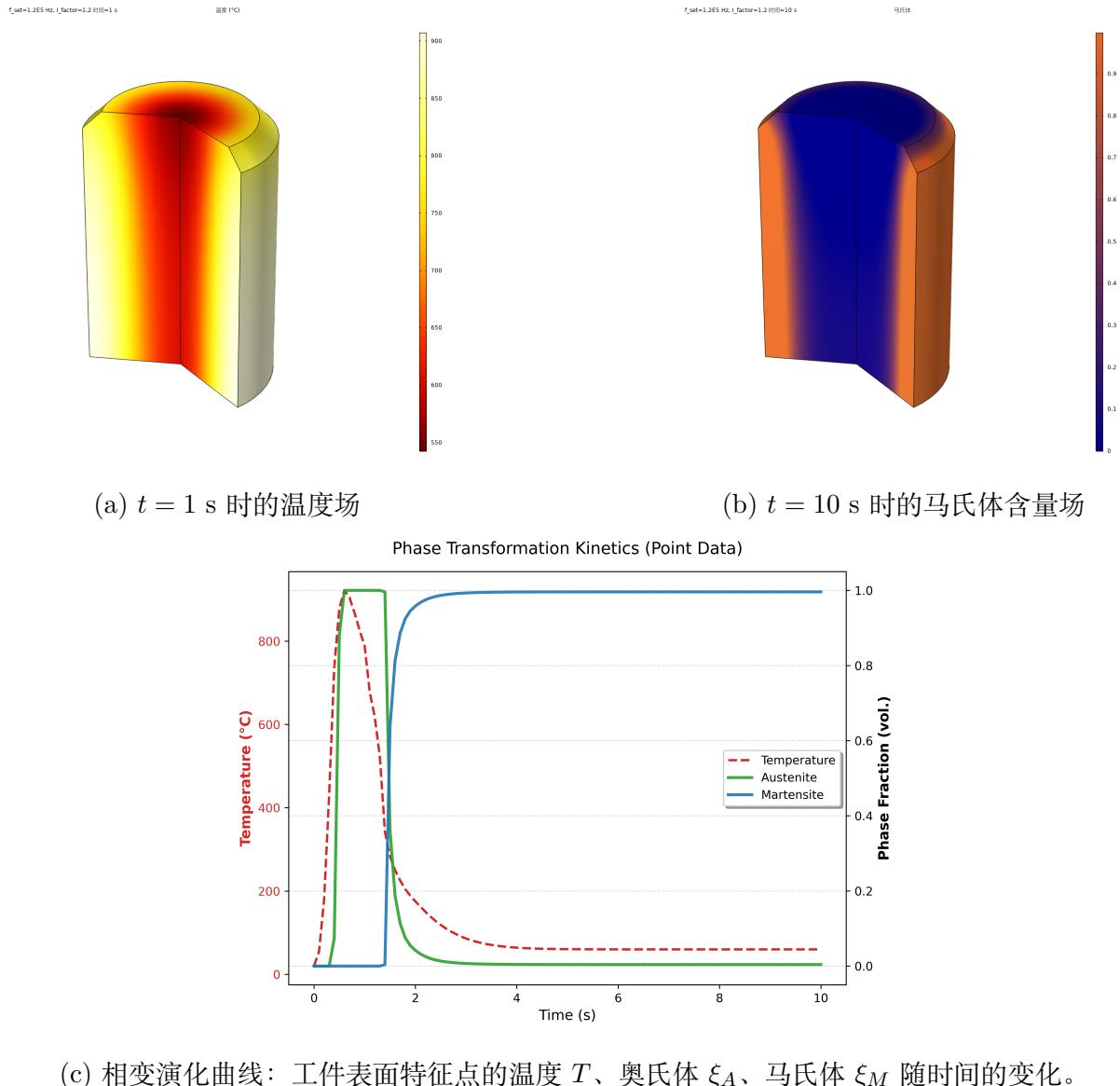


图 4: 典型物理场云图与相变演化曲线

3 数据预处理与数据集构建

为了将 COMSOL 导出的原始有限元仿真数据转换为适用于神经算子 (Neural Operator) 训练的标准张量格式, 本项目设计了一套完整的 ETL (Extract, Transform, Load) 流水线。预处理流程主要包含原始数据解析、张量重构、特征工程与数据集划分四个阶段。

3.1 原始数据解析与清洗

原始数据为非结构化的 CSV 格式，包含不同工艺参数（频率 f 、电流 I ）下的节点仿真结果。

- **表头解析**: 通过自动化脚本识别 COMSOL 导出的动态表头，建立变量名（Temperature, Austenite, Martensite）与列索引的映射关系。
- **坐标对齐**: 原始数据采用 (r, z) 二维轴对称坐标系。由于网格节点在不同时间步保持不变，脚本提取唯一的几何节点坐标，并剔除因网格划分产生的无效数据（NaN）。
- **序列化存储**: 将原始的大型宽表转换为 Parquet 格式的长表（Long Format），包含 $[r, z, \text{time}, \text{freq}, \text{current}, \text{Temperature}, \text{Austenite}, \text{Martensite}]$ 等字段，显著降低了后续处理的内存开销。

3.2 张量重构与几何掩码

为了适配 FNO 模型的卷积操作，我们需要将离散的节点数据重构为规则的网格张量。核心逻辑如下：

1. **排序与重塑**: 数据首先按照 Time $\rightarrow z(\text{Height}) \rightarrow r(\text{Width})$ 的优先级进行严格排序，随后重塑为 (T, H, W, C) 的四维张量。其中 $H = 128, W = 64$ 。
2. **几何掩码 (Geometry Mask)**: 由于轴承钢工件并非填满整个矩形计算域（存在倒角和空气域），利用温度场中的 NaN 值自动生成二值几何掩码 $M \in \{0, 1\}^{H \times W}$ 。
 - **实体区域 ($M = 1$)**: 保留物理场数值。
 - **空气区域 ($M = 0$)**: 将所有物理量填充为 0.0，并在计算 Loss 时通过掩码忽略该区域的误差。
 - 掩码被保存以供训练和推理时复用。

3.3 特征工程与归一化

模型输入与输出的通道设计如下，最终数据格式转换为 PyTorch 推荐的 Channel-First 格式： (N, C, H, W) 。

1. **通道扩展**: 原始数据仅包含奥氏体 (P_{aust}) 和马氏体 (P_{mart}) 两种相。为了满足物理守恒约束（相分数之和为 1），我们显式计算了“初始相/珠光体”通道 (P_{init}):

$$P_{init} = 1.0 - (P_{aust} + P_{mart}) \quad (1)$$

并对其进行截断处理，防止浮点数溢出。最终输出通道为 4 个：Temperature, Austenite, Martensite, Initial。

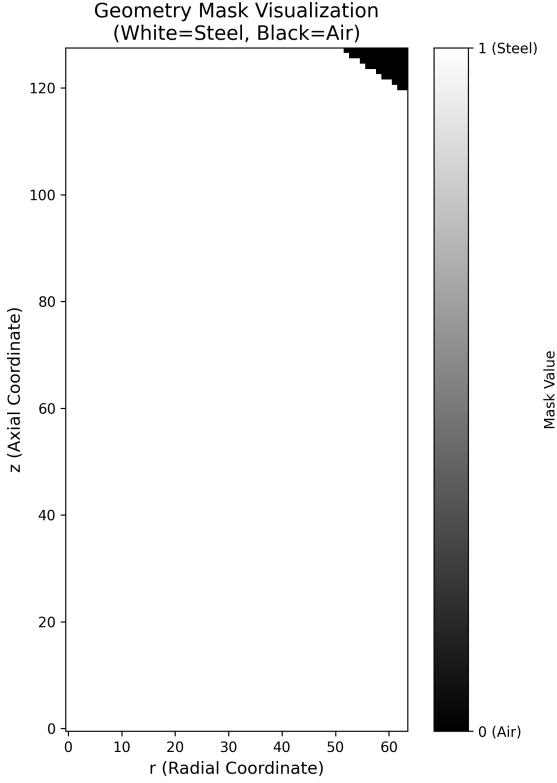


图 5: 几何掩码示意图

2. 混合归一化策略:

- **温度场**: 采用 Min-Max 归一化，将温度映射至 $[0, 1]$ 区间。统计数据来源于全局扫描，确保所有样本使用统一的 T_{min} 和 T_{max} 。

$$T_{norm} = \frac{T - T_{min}}{T_{max} - T_{min}} \quad (2)$$

- **相变场**: 由于相分数本身即为 $[0, 1]$ 之间的无量纲物理量，不进行额外归一化，以保持其物理意义。

3.4 智能边界保护划分

在数据集划分阶段，为了保证模型在参数空间的泛化能力，我们没有采用完全随机的划分方式，而是实施了“边界保护策略”：

1. **参数空间扫描**: 首先提取所有样本的频率 f 和电流 I ，确定参数空间的凸包 (Convex Hull)，即 (f_{min}, f_{max}) 和 (I_{min}, I_{max}) 。
2. **强制训练集**: 凡是落在参数边界上的样本（如 $f = f_{min}$ 或 $I = I_{max}$ ），被强制分配至训练集。这确保了模型能够学习到物理场的边界条件，避免在推理时出现外推 (Extrapolation) 导致的性能下降。

- 随机分配：剩余位于参数空间内部的样本，按照 75%:12.5%:12.5% 的比例随机划分为训练集、验证集和测试集。

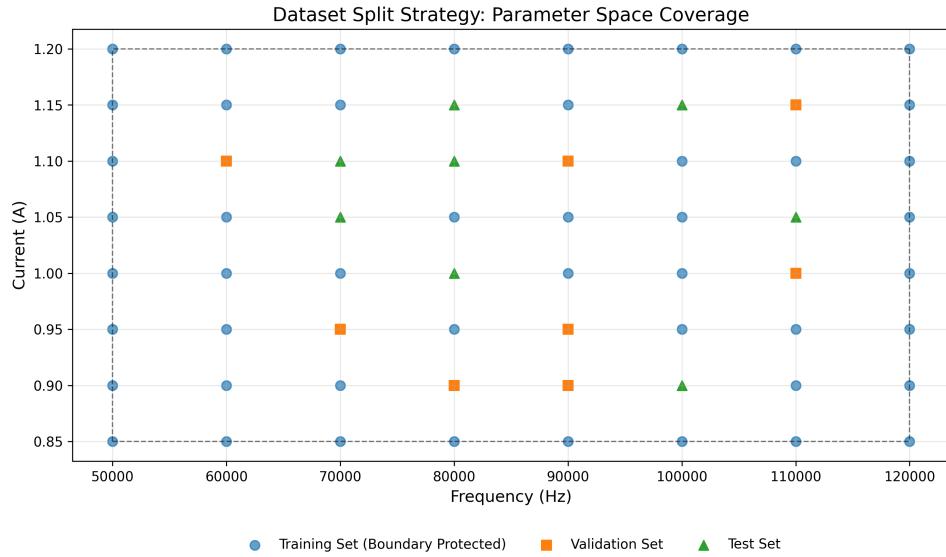


图 6: 参数空间分布与数据集划分示意图

4 模型架构

4.1 总体架构设计

针对感应淬火过程中热传导（全局扩散特性）与相变（局部尖锐边界）并存的物理特性，本项目提出了一种名为 Parallel U-FNO 的双流并行神经网络架构。该架构摒弃了单一模型的局限性，采用“分而治之”的策略，将频域的傅里叶神经算子（FNO）与空域的 U-Net 网络并行结合。

模型主要由三个部分组成：

- 全局流 (Global Stream)**: 基于 FNO，负责捕捉低频、长程的热扩散特征。
- 局部流 (Local Stream)**: 基于 U-Net，负责捕捉高频、局部的相变边界特征。
- 特征融合与多头输出 (Fusion & Dual-Head Output)**: 将两路特征融合后，通过独立的回归头分别预测温度场和相变场。

4.2 子模块详解

4.2.1 FNO 分支

FNO 分支利用快速傅里叶变换（FFT）在频域中进行卷积操作。由于热传导方程本质上是积分算子，FNO 能够高效地学习这种全局积分映射。

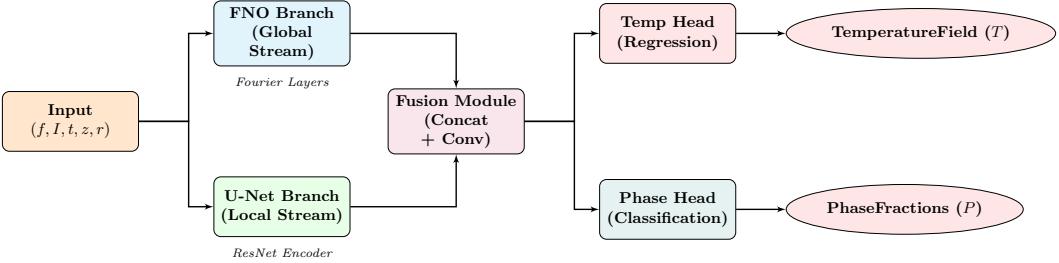


图 7: Parallel U-FNO 整体架构图

- **输入:** 时空坐标与工艺参数张量 $x = (f, I, t, z, r)$ 。模型学习的是从坐标到物理场的直接映射 $G : (f, I, t, z, r) \rightarrow (T, \text{Phases})$ 。
- **核心操作:** 将输入映射到高维空间，进行 N 层 Spectral Convolution (频谱卷积)，最后投影回目标维度。
- **代码实现:** 调用 `neuralop.models.FNO`，配置模式数 `n_modes=(16, 16)`，确保保留主要的低频能量。

4.2.2 U-Net 分支

为了弥补 FNO 在处理高频突变（如硬化层边界）时的吉布斯现象（震荡效应），引入 U-Net 分支。

- **编码器:** 采用 ResNet-18 作为主干网络，提取多尺度空间特征。
- **解码器:** 通过跳跃连接融合深层语义与浅层细节，精确恢复空间几何信息。
- **代码实现:** 基于 `segmentation_models_pytorch` 库构建，与 FNO 并行处理同一输入。

4.3 特征融合与双头机制

这是本模型与普通集成模型最大的不同之处。

1. **拼接:** 模型首先将 FNO 分支输出 O_{fno} 和 U-Net 分支输出 O_{unet} 在通道维度进行拼接：

$$F_{cat} = \text{Concat}(O_{fno}, O_{unet}) \quad (3)$$

2. **非线性融合:** 拼接后的特征通过一个共享融合层 (Shared Fusion Layer)，该层由 1×1 卷积、GELU 激活函数和 BatchNorm 组成。这一步不仅降低了维度，还引入了非线性变换以混合频域和空域特征：

$$F_{shared} = \text{BatchNorm}(\text{GELU}(\text{Conv}_{1 \times 1}(F_{cat}))) \quad (4)$$

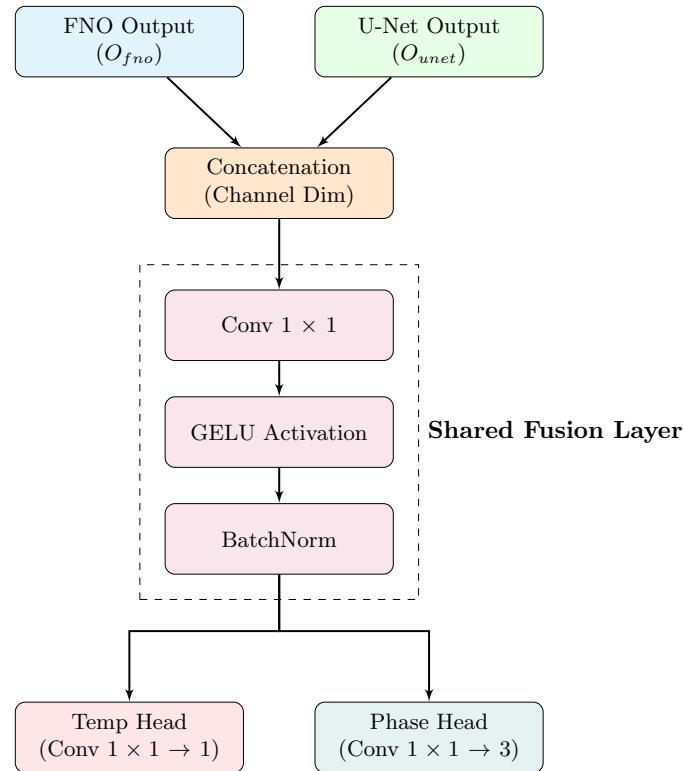


图 8: 融合模块细节图：展示了特征拼接、非线性融合以及双头分叉的详细流程。

3. **独立回归头**: 考虑到温度场（连续平滑）和相变场（存在阶跃）的物理属性差异，模型在最后阶段设计了两个独立的 1×1 卷积头：

- **温度头**: 专门回归温度值 T 。
- **相变头**: 专门回归相变组织分布 P 。

最终输出为两者在通道维度的拼接： $Output = [T, P]$ 。

代码实现关键片段：

```

1 class ParallelUFNO(nn.Module):
2     def __init__(self, ...):
3         # ... (init code)
4         # Fusion Layer: Mixes frequency (FNO) and spatial (U-Net) features
5         self.fusion_shared = nn.Sequential(
6             nn.Conv2d(out_channels * 2, hidden_channels, kernel_size=1),
7             nn.GELU(),
8             nn.BatchNorm2d(hidden_channels),
9         )
10
11     def forward(self, x):
12         # Parallel Execution
13         out_fno = self.fno(x)      # Global features

```

```

14     out_unet = self.unet(x) # Local features
15
16     # Fusion
17     out_cat = torch.cat([out_fno, out_unet], dim=1)
18     features = self.fusion_shared(out_cat)
19
20     # Dual Heads
21     out_temp = self.head_temp(features)
22     out_phase = self.head_phase(features) # Outputs Logits
23
24     return torch.cat([out_temp, out_phase], dim=1)

```

4.4 损失函数设计

为了精准捕捉感应淬火过程中的温度场演变与微观组织相变，本项目设计了一种复合损失函数 L_{total} 。该函数主要由数值精度损失和边界梯度损失两部分组成。

总损失函数定义如下：

$$L_{total} = \alpha \cdot L_{MSE} + \beta \cdot L_{Sobel} \quad (5)$$

其中， α 和 β 为超参数权重，分别控制数值精度和边界清晰度的重要性。

损失函数代码实现：

```

1 class CombinedLoss(nn.Module):
2     def forward(self, pred, target, mask=None):
3         # Split predictions
4         pred_temp = pred[:, 0:1, :, :]
5         pred_phases_logits = pred[:, 1:, :, :]
6
7         # Apply Softmax for physical constraint (Sum=1)
8         pred_phases_prob = F.softmax(pred_phases_logits, dim=1)
9
10        # 1. MSE Loss
11        loss_temp = self.mse(pred_temp, target[:, 0:1])
12        loss_phase = self.mse(pred_phases_prob, target[:, 1:])
13
14        # 2. Sobel Gradient Loss (Edge Sharpness)
15        loss_sobel = self.sobel(pred_temp, target[:, 0:1], mask) + \
16                    self.sobel(pred_phases_prob, target[:, 1:], mask)
17
18        return self.alpha * (loss_temp + loss_phase) + self.beta * loss_sobel

```

4.4.1 基础数值损失

这是模型优化的基础，用于确保预测值在数值上逼近真实值。

$$L_{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (6)$$

该项同时应用于温度场和相变场，确保模型能够学习到全局的物理量分布。

4.4.2 边界增强损失

针对感应淬火中硬化层边界模糊的问题，我们引入了基于 Sobel 算子的梯度损失。通过计算预测场与真实场在水平 (G_x) 和垂直 (G_y) 方向上的梯度差异，强制模型关注物理场的边缘锐度：

$$L_{Sobel} = \|\nabla \hat{y} - \nabla y\|^2 = \|S_x(\hat{y}) - S_x(y)\|^2 + \|S_y(\hat{y}) - S_y(y)\|^2 \quad (7)$$

这一项对于精确预测硬化层深度至关重要。

4.4.3 物理约束的实现

值得特别说明的是，关于相变组织演变的物理守恒律（即各相分数之和为 1: $\sum P_i = 1$ ），本项目采用了硬约束而非软约束的方式：

- **架构级保证：**模型输出层输出未归一化的 Logits，在计算损失函数及推理时应用 Softmax 激活函数。
- **优势：**这从数学结构上天然保证了最终输出的相变概率分布归一化 ($\sum P_i = 1$)，无需在损失函数中添加额外的物理惩罚项（如 $\gamma \cdot |1 - \sum P|$ ），从而简化了优化目标，提高了训练稳定性。

4.4.4 几何掩码机制

为了处理非规则几何工件，计算过程中引入了二值掩码 M （工件 =1，空气 =0）。所有损失计算仅在掩码区域内进行：

$$L = \frac{\sum(L_{pixel} \cdot M)}{\sum M} \quad (8)$$

这有效避免了空气域的无效计算对梯度的干扰。

5 模型训练

5.1 实现细节

本项目基于 PyTorch 深度学习框架实现。训练过程在配备 NVIDIA GeForce RTX 4060 Laptop GPU 的计算平台上进行。

- **优化器**: 采用 Adam 优化器，其具有自适应学习率的特性，能够快速收敛。
- **学习率调度**: 使用 StepLR 调度策略。初始学习率设定为 1×10^{-3} ，每经过 50 个 Epoch，学习率衰减为原来的 0.8 倍。这种策略有助于模型在训练初期快速下降，并在后期微调以获得更优的局部极小值。

5.2 超参数设置

主要的训练超参数总结如下表所示：

表 1: 训练超参数设置

参数 (Parameter)	数值 (Value)
Batch Size	64
Epochs	200
Initial Learning Rate	1×10^{-3}
Weight Decay	1×10^{-4}
Loss Weight α (MSE)	5.0
Loss Weight β (Sobel)	2.0
Early Stopping Patience	30

5.3 训练策略

为了提高模型的鲁棒性和泛化能力，我们采取了以下关键策略：

1. **几何掩码机制**: 在计算损失函数时，利用预生成的几何掩码过滤掉空气域的预测误差。这确保了梯度更新仅由工件内部的物理场偏差驱动，避免了无效区域的干扰。
2. **早停机制**: 为了防止过拟合，我们在验证集上监控损失值的变化。如果验证集损失在连续 30 个 Epoch 内没有显著下降（阈值 $\Delta < 1 \times 10^{-5}$ ），则提前终止训练。
3. **模型保存**: 训练过程中自动记录验证集损失最低的模型权重 (Best Model)，确保最终使用的模型具有最佳的泛化性能。

6 实验结果与分析

本节将展示 Parallel U-FNO 模型在测试集上的评估结果。我们将从模型收敛性、定量误差指标、物理场预测可视化以及硬化层深度预测精度四个方面进行详细分析。

6.1 模型收敛性分析

图 9 展示了模型在训练过程中的损失函数下降曲线。可以看出，随着训练轮次 (Epoch) 的增加，训练集损失 (Train Loss) 和验证集损失 (Val Loss) 均呈现快速下降趋势，并在约 100 个 Epoch 后趋于平稳。验证集损失紧随训练集损失，未出现明显的过拟合现象，表明模型具有良好的泛化能力。

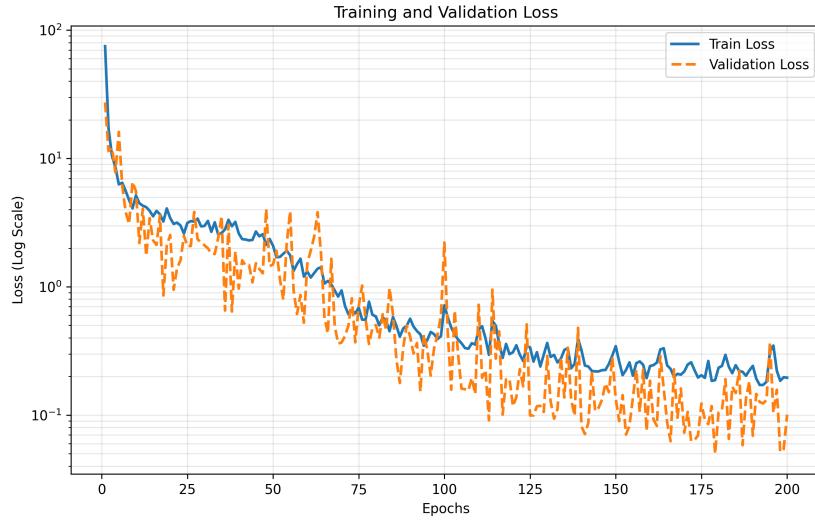


图 9: 训练集与验证集损失函数收敛曲线

6.2 定量评估指标

表 2 列出了模型在测试集上的关键性能指标。我们在温度场 (Temperature) 和相变场 (Phase) 两个维度上分别计算了均方误差 (MSE) 和平均绝对误差 (MAE)。

表 2: 测试集定量评估结果

物理量	MSE	MAE
温度场	2.46×10^{-5}	0.0034
相变场	4.15×10^{-5}	0.0029
整体	3.73×10^{-5}	0.0030

此外，模型的相对 L2 误差仅为 1.24%，这表明模型在全局范围内能够以极高的精度逼近真实物理场。

6.3 物理场预测可视化

为了直观评估模型的预测能力，我们选取了测试集中具有代表性的样本进行可视化对比。

图 10 展示了温度场和马氏体含量的预测值与真实值对比散点图 (Parity Plot)。绝大多数数据点紧密分布在 $y = x$ 对角线附近，表明模型在整个数值范围内（从室温到奥氏体化温度，从无相变到完全相变）都保持了高度的一致性。

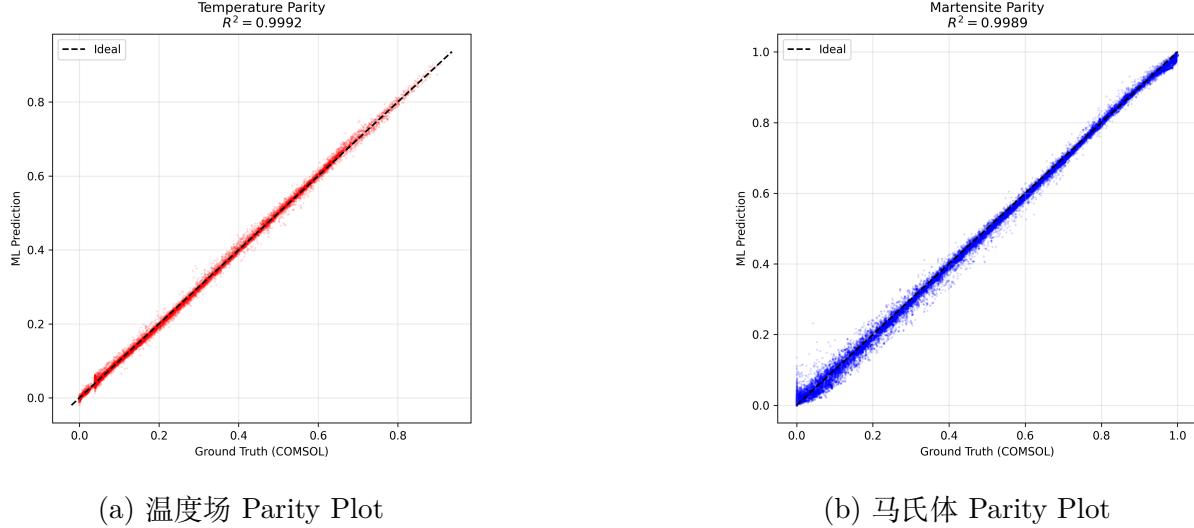


图 10: 预测值与真实值的一致性分析

图 11 展示了某一测试样本在冷却结束时刻的马氏体分布对比。可以看出，Parallel U-FNO 成功捕捉到了感应淬火特有的“月牙形”硬化层轮廓，且在相变梯度的过渡区域表现出了良好的平滑性。

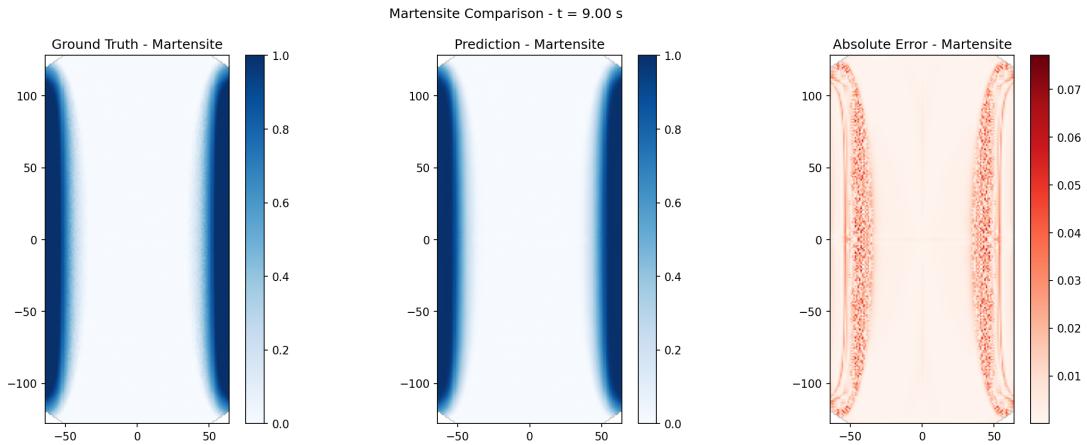


图 11: 马氏体场预测对比：(左) 真实值 (中) 预测值 (右) 绝对误差

6.4 硬化层预测精度

感应淬火的核心工艺指标是有效硬化层深度。图 12 展示了工件表面及特定深度处的马氏体分布曲线与温度分布曲线。模型预测的曲线（虚线）与有限元仿真结果（实线）高度重合，能够精确界定马氏体含量 50% 的临界位置，这对于实际工艺参数的优化具有重要的指导意义。

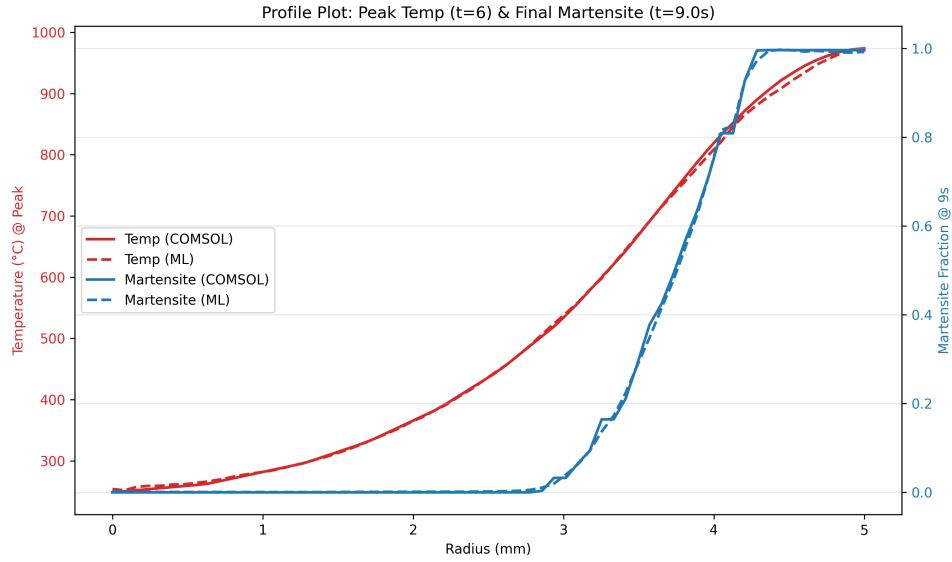


图 12: 不同深度处的马氏体分布与温度曲线的对比

6.5 不足

尽管 Parallel U-FNO 模型在测试集上表现出了优异的预测性能，但本研究仍存在以下局限性，需要在未来的工作中进一步解决：

1. **仿真数据的物理一致性瑕疵**: 在分析相变动力学曲线（如图 12 所示）时发现，部分工况下的马氏体分布曲线在上升阶段呈现出非物理的“小台阶”现象。这主要归因于有限元仿真（COMSOL）在处理相变潜热释放时的数值震荡或时间步长设置不够精细。由于神经网络本质上是在拟合训练数据，这些存在于 Ground Truth 中的噪声限制了模型预测精度的进一步提升，并可能导致模型学习到错误的局部特征。
2. **模型泛化能力的边界**: 当前的训练数据集仅基于单一的圆柱销几何模型和特定的亚共析钢材料属性构建。虽然模型在设定的频率（50 – 120 kHz）和电流倍率（0.85 – 1.2）参数空间内表现良好，但其泛化能力受到严格限制：
 - **几何依赖性**: 模型无法直接迁移应用到齿轮、凸轮轴等复杂几何工件的感应淬火预测中。
 - **材料依赖性**: 对于化学成分不同或相变动力学参数差异较大的钢种，模型需要重新训练。
3. **数据规模的限制**: 本研究仅使用了 64 组仿真实验数据。尽管通过全时空采样生成了大量样本，但对于覆盖高维非线性物理过程而言，参数空间的采样密度仍然较低。这可能导致模型在参数空间的某些稀疏区域预测置信度下降。

7 结论

本项目针对轴类零件感应淬火这一复杂多物理场耦合过程，成功开发并验证了基于 Parallel U-FNO 架构的深度学习代理模型。通过系统性的研究与实验，得出了以下核心结论：

1. **架构设计的有效性：**Parallel U-FNO 架构通过并行融合频域的 FNO 分支和空域的 U-Net 分支，有效地解决了单一模型难以兼顾全局热扩散与局部相变突变特征的难题。实验证明，该架构在温度场和相变场的联合预测任务中表现卓越，整体相对误差控制在 1.24% 以内。
2. **工艺预测的高保真度：**模型不仅在数值上实现了高精度拟合，更重要的是成功捕捉到了感应淬火的关键工艺特征。可视化结果显示，模型能够准确复现“月牙形”硬化层的几何轮廓，并精确界定马氏体含量 50% 的有效硬化层深度，具有极高的工程实用价值。
3. **数据驱动方法的潜力：**本研究展示了数据驱动方法在替代传统数值模拟方面的巨大潜力。相比于耗时数十分钟甚至数小时的有限元仿真，训练好的神经网络模型可以在毫秒级时间内完成全场预测，为未来实现感应淬火工艺的实时闭环控制和大规模参数优化奠定了坚实基础。

综上所述，本项目不仅提供了一个高性能的感应淬火预测模型，也为利用人工智能技术解决复杂制造工艺中的多物理场建模问题提供了有益的探索和范例。