

基于多种深度学习模型的刀具磨损状态预测

作者：马张诚

二、数据采集与预处理

2.1 实验背景与数据采集

本研究所用数据来源于切削加工实验平台，使用Kistler声发射传感器测量切削过程中产生的高频应力波。该传感器可同时获取三类物理量：三个方向的切削力（x、y、z方向）；三个方向的振动信号（x、y、z方向）；声发射信号（AE）。

上述7个通道信号由NI数据采集卡统一采样，形成时间序列数据。每次走刀后，使用LEICMZ12显微镜测量每个刀片后刀面平均磨损带宽（VB），作为刀具磨损的真实标签。



图1 两种磨损度刀具的对比



图2 切削过程记录

2.2 数据集结构

数据集共包含三个子集（c1、c4、c6），分别对应不同工况下的切削实验，每个子集中包含315个文件（编号从c_1_001.csv至c_1_315.csv），每个文件记录了一次走刀过程的7通道信号，采样频率为15385 Hz，每条信号长度约为10万点。

整体数据规模为：120000×7×315×3。其中，120000为单次走刀信号长度，7为通道数，315为样本数量，3为实验组数（c1, c4, c6）。总样本量多、数据维度高、时序性强，具有典型的工业大数据特征。

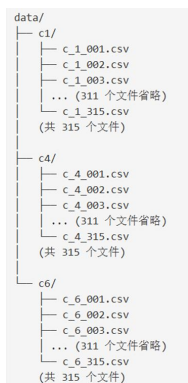


图3 文件树

127385	0.889	1.449	0.867	-0.041	-0.041	-0.036	-0.004
127386	0.811	1.395	0.885	-0.024	-0.017	-0.02	-0.004
127387	0.755	1.396	0.908	0.175	0.183	0.182	-0.004
127388	0.636	1.283	0.878	-0.03	-0.021	-0.03	-0.004
127389	0.534	1.25	0.873	-0.023	-0.008	-0.017	-0.004
127390	0.466	1.164	0.832	0.214	0.222	0.223	-0.004
127391	0.428	1.054	0.802	0.035	0.03	0.039	-0.004
127392	0.383	0.918	0.75	-0.013	-0.017	-0.01	-0.004
127393	0.315	0.796	0.666	0.193	0.197	0.201	-0.004
127394	0.274	0.724	0.598	0.085	0.092	0.099	-0.005
127395	0.248	0.692	0.531	-0.085	-0.082	-0.071	-0.004
127396	0.245	0.689	0.483	0.184	0.186	0.194	-0.004
127397	0.22	0.63	0.415	0.067	0.066	0.073	-0.004
127398	0.218	0.537	0.346	-0.099	-0.098	-0.087	-0.004
127399	0.207	0.483	0.292	0.111	0.114	0.125	-0.004

图4 表格数据举例

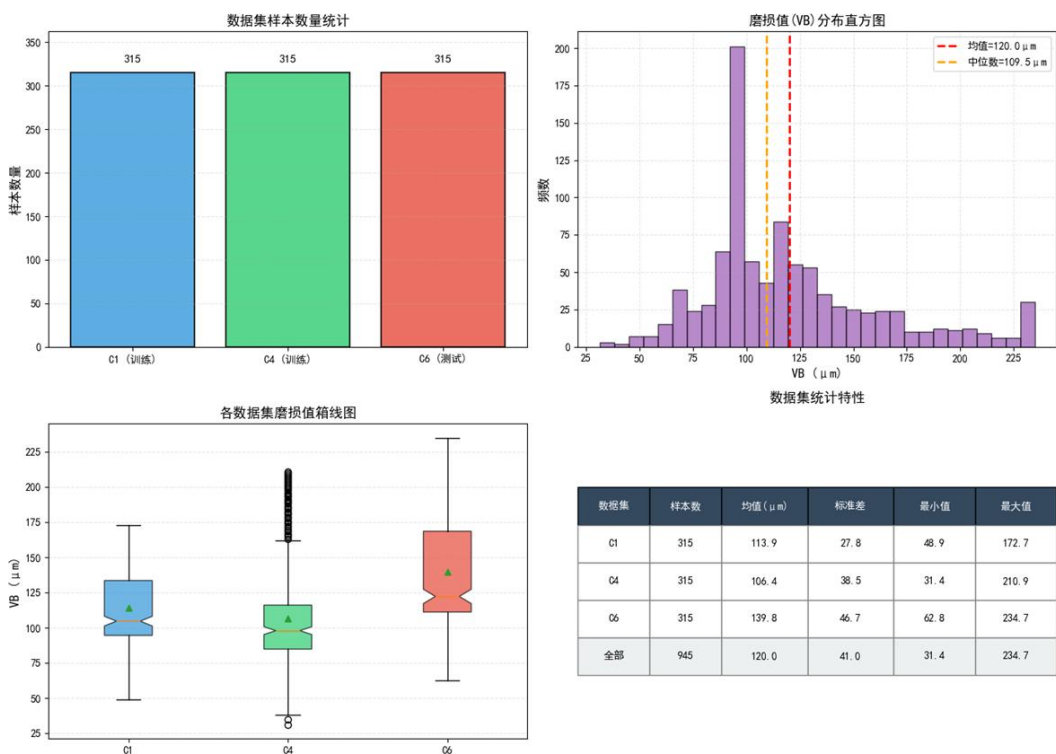


图5 数据集特征可视化

2.3 数据预处理

为提升模型训练效率并增强泛化能力，对原始信号进行了系统的预处理操作，主要包括以下四个阶段：滑动窗口分割、原始信号特征提取、特征筛选与降维、时间序列构造。

2.3.1 滑动窗口分割

由于原始信号为连续的时间序列，直接使用整段信号会导致样本数量不足且缺乏时间局部性表达。因此引入滑动窗口机制将长序列划分为多个短片段，以增加训练样本数量并捕捉动态变化趋势。具体参数设置如下：

- 窗口大小：1024个采样点，对应约66.5ms；
- 步长：512个采样点（即50%重叠）；
- 重叠策略：提高时间分辨率，捕获渐变的磨损过程。

此策略使得每条原始信号被划分为多个非独立但高度相关的子序列，显著提升了样本多样性。

2.3.2 原始信号特征提取

针对每个滑动窗口内的7通道信号，分别提取其时域与频域特征，构成综合特征向量。

(1) 时域特征提取

定义函数 `time_domain_features(x)` 提取以下六项统计量：平均值（mean）、标准差（std）、峰峰值（pp）、均方根值（rms）、偏度（skewness）、峰度（kurtosis）。

这些指标反映了信号的能量分布、波动强度及形态特性。

(2) 频域特征提取

利用快速傅里叶变换（FFT）分析信号频率组成，定义函数 `freq_domain_features(x)` 提取四项频域特征：

能量（energy）： $\sum |X(f)|^2$ ；

主要频率（main_freq）：幅值最大对应的频率；

平均频率（mean_freq）：加权平均频率；

频率方差（var_freq）：反映频谱分散程度

所有通道均独立进行特征提取，最终每个窗口生成 $7 \times (6+4) = 70$ 维特征向量。

(3) 时频域特征提取

实验过程中尝试用小波来进行时频域特征提取，但在训练过程中遇到了训练时间过长的问题，遂放弃，可以考虑在后续改进中再次尝试。

2.3.3 特征筛选与降维

原始特征空间维度高达70维，存在大量冗余和噪声，需进一步优化。

(1) 基于Pearson相关性的特征筛选

计算各特征与目标变量（VB磨损值）之间的Pearson相关系数，保留绝对值大于阈值0.3的特征。该方法有效去除与磨损无关或弱相关的特征，减少噪声干扰。

```
def select_features_by_correlation(X: np.ndarray, y: np.ndarray, threshold: float = CORR_THRESHOLD):
    selected = []
    for i in range(X.shape[1]):
        r, _ = pearsonr(X[:, i], y)
        if abs(r) >= threshold:
            selected.append(i)
    return X[:, selected], selected
```

经筛选后，特征维度由70降至约54维。

(2) Z-score标准化

对筛选后的特征进行Z-score标准化，使所有特征服从均值为0、标准差为1的分布。作用包括：加速神经网络梯度下降收敛；防止数值范围大的特征主导模型学习；提升模型稳定性。

```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train_sel)
```

```
X_test_scaled = scaler.transform(X_test_sel)
```

(3) 主成分分析（PCA）降维

为进一步降低维度并保留主要信息，采用PCA将特征从54维降至15维（设定N_COMPONENTS=15）。根据累计方差解释率曲线显示，前15个主成分已能解释超过95%的总方差，说明降维过程未造成显著信息损失。

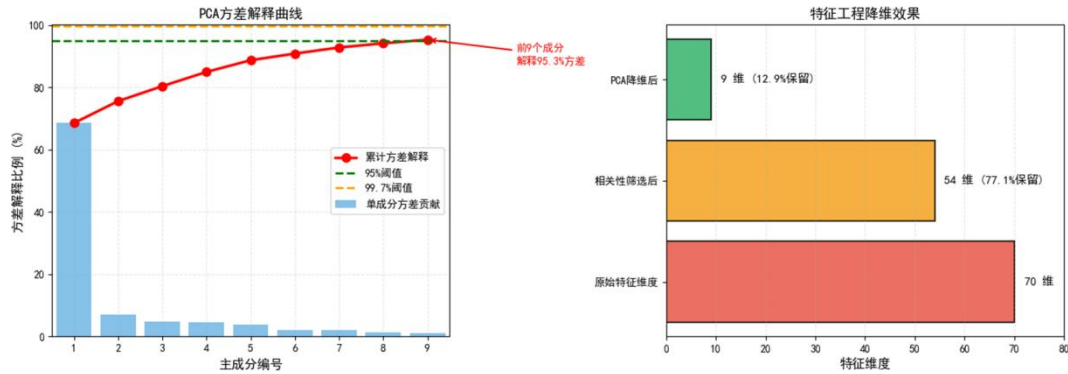


图6 PCA方差解释曲线与某次特征工程降维效果

2.3.4 时间序列构造与数据加载配置

尽管原始信号已通过滑动窗口划分成多个片段，但为了适应深度学习模型对序列结构的需求，还需将其构造“输入序列 → 输出标签”的形式。

(1) 滑动窗口序列化

将提取出的特征序列按固定长度（seq_len=10）滑动窗口构造为时间序列样本：

```
def create_sequences(X, y, seq_len=10):
    xs, ys = [], []
    for i in range(len(X) - seq_len + 1):
        xs.append(X[i:i+seq_len])
        ys.append(y[i+seq_len-1])
    return np.array(xs), np.array(ys)
```

例如：

输入：[f1, f2, ..., f10] → 输出：y10

输入：[f2, f3, ..., f11] → 输出：y11

从而实现“过去10个状态预测当前磨损”的建模范式。

(2) 数据加载器配置

使用PyTorch的DataLoader封装训练与测试数据，设置批量大小（batch_size=32）、是否打乱顺序（shuffle=True/False），确保训练过程稳定高效。

```
train_loader = DataLoader(train_dataset, batch_size=32, shuffle=True)
test_loader = DataLoader(test_dataset, batch_size=32, shuffle=False)
```

2.4 数据集划分与实验设计

训练集：c1 + c4（共630个样本）

测试集：c6（315个样本）

这种划分方式模拟了真实工业场景中“在已知工况下训练，在未知工况下部署”的需求，更具挑战性和实用价值。

用c1+c6训练，c4测试，或者c4+c6训练，c1测试流程与之相同，后续可以进一步补充。

四、回归任务——深度学习模型建构

4.1 多层感知机（Multi-Layer Perceptron, MLP）

多层感知机是前馈人工神经网络的典型代表。其核心思想是通过堆叠多个全连接隐藏层，并引入非线性激活函数，以构建一个强大的通用函数逼近器。理论上，一个具有足够宽度和深度的MLP可以以任意精度逼近任何连续函数。给定输入特征向量

$$\mathbf{x} \in \mathbb{R}^d$$

MLP的前向传播过程可表示为：

$$\mathbf{h}^{(l)} = \sigma(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)})$$

在本研究中，经过预处理流程（滑动窗口分割、70维时频域特征提取、相关性筛选、Z-score标准化及PCA降维）后，每个时间步的输入被压缩为一个15维的特征向量。此向量已凝聚了原始多通道信号的统计与频谱信息，但其内部各维度间的复杂非线性交互关系仍需一个强表达能力的模型来解耦。MLP凭借其强大的非线性拟合能力，成为处理此类静态（或伪静态）高维特征向量的理想基线模型。其结构简洁，训练高效，能有效捕捉特征与VB磨损值之间的映射关系，为后续更复杂模型的性能提供基准参照。

本研究采用的MLP包含三个隐藏层，具体结构为：输入层（15维）→ 隐藏层1（192维，ReLU激活，Dropout=0.2）→ 隐藏层2（96维，ReLU激活，Dropout=0.2）→ 输出层（1维）。Dropout层的引入旨在防止过拟合，提升模型泛化能力。

```
class MLPRegressor(nn.Module):  
    def __init__(self, input_dim: int, hidden_size: int = HIDDEN_SIZE, dropout: float = DROPOUT):  
        super().__init__()  
        self.net = nn.Sequential(  
            nn.Linear(input_dim, hidden_size),  
            nn.ReLU(),  
            nn.Dropout(dropout),  
            nn.Linear(hidden_size, hidden_size // 2),  
            nn.ReLU(),  
            nn.Dropout(dropout),  
            nn.Linear(hidden_size // 2, 1),  
        )
```

训练配置：

优化器：Adam（LR=7e-5, weight_decay=5e-5）

损失函数：MSELoss

批次大小：32

最大迭代次数：2000

早停机制：连续200轮未提升则停止

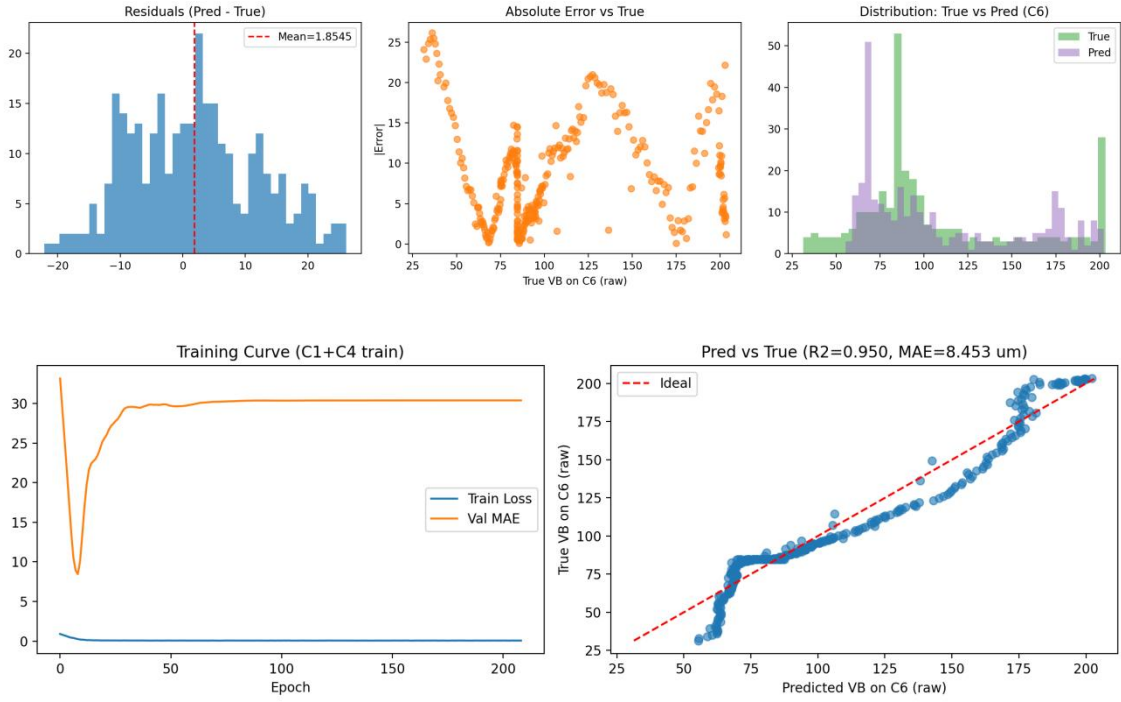


图7 MLP训练效果可视化

如图，MLP在C6测试集上取得 $R^2 = 0.950$ 、 $MAE = 8.453 \mu m$ 的结果，表明其具备较强的非线性拟合能力，能够有效捕捉磨损演化趋势。

4.2 带注意力机制的MLP（MLP+Attention）

注意力机制的核心思想是模仿人类视觉的聚焦能力，让模型能够动态地为输入特征的不同部分分配不同的权重，从而突出对当前任务更为关键的信息。在本模型中，我们采用了一种简单的加性注意力（Additive Attention）变体。首先，通过一个小型的全连接子网络（注意力模块）生成与输入维度相同的注意力权重向量。

$$\mathbf{a} = \text{softmax}(\mathbf{W}_2 \cdot \text{ReLU}(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2)$$

随后，将原始输入 \mathbf{x} 与权重向量 \mathbf{a} 进行逐元素相乘，得到加权后的特征表示

$$\mathbf{x}' = \mathbf{x} \odot \mathbf{a}$$

再将其送入标准的MLP主干网络进行回归预测。

刀具磨损过程对不同传感器信号的敏感度并非均等。例如，在磨损初期，声发射（AE）信号的能量变化可能比切削力信号更为显著；而在剧烈磨损阶段，振动信号的峰峰值和均方根值（RMS）可能成为主导特征。传统的MLP对所有输入特征一视同仁，无法自适应地关注这些“关键线索”。引入注意力机制后，模型能够在训练过程中自动学习到哪些特征维度（或其组合）在特定磨损阶段更具判别性，从而增强模型对磨损演化动态的感知能力，提升预测精度。

```
class AttnMLP(nn.Module):
    def __init__(self, input_dim: int, hidden_size: int = HIDDEN_SIZE, dropout: float = DROPOUT):
        super().__init__()
```

```

self.attn = nn.Sequential(
    nn.Linear(input_dim, hidden_size),
    nn.ReLU(),
    nn.Linear(hidden_size, input_dim),
)

self.backbone = nn.Sequential(
    nn.Linear(input_dim, hidden_size),
    nn.ReLU(),
    nn.Dropout(dropout),
    nn.Linear(hidden_size, hidden_size // 2),
    nn.ReLU(),
    nn.Dropout(dropout),
    nn.Linear(hidden_size // 2, 1),
)

```

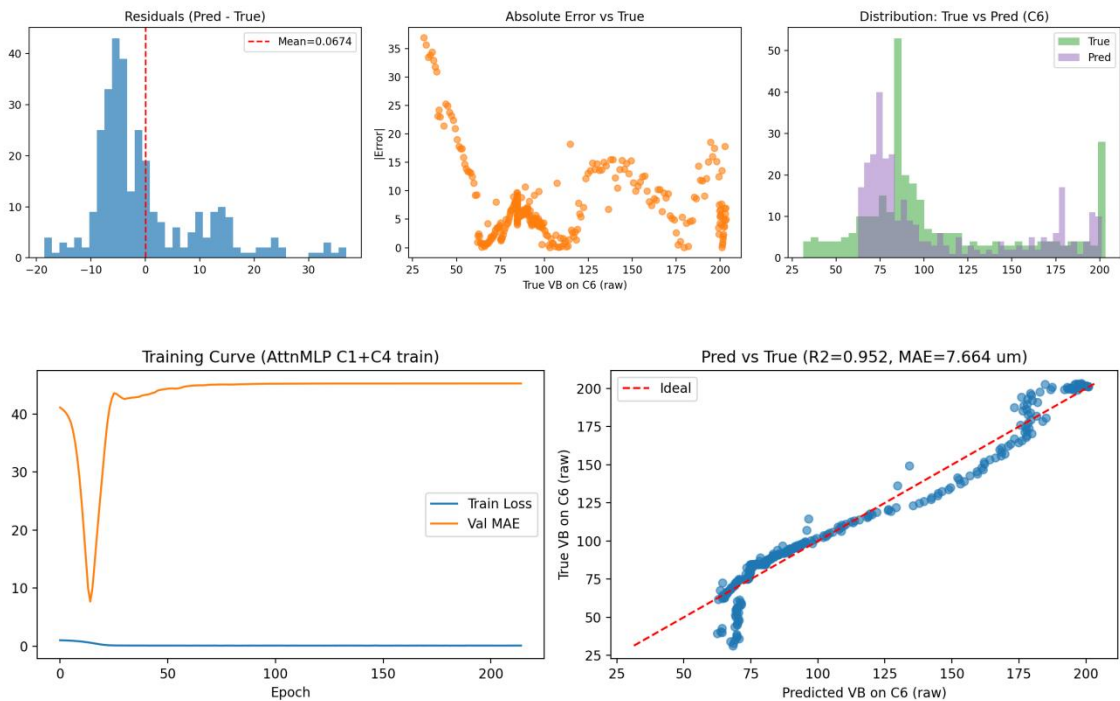


图8 MLP+Attention训练效果可视化

如图4.2所示,MLP+Attention模型进一步提升了预测精度,达到 $R^2 = 0.952$ 、 $MAE = 7.664 \mu m$, 优于原始MLP。注意力机制有效增强了模型对振动幅值、频率能量等敏感特征的响应能力。

4.3 长短期记忆网络 (Long Short-Term Memory, LSTM)

长短期记忆网络是一种特殊的循环神经网络, 专为解决标准RNN在处理长序列时存在的梯度消失/爆炸问题而设计。LSTM通过引入遗忘门、输入门、输出门等“门控机制”来控制信息在细胞状态中的流动, 从而能够选择性地记住长期依赖信息并遗忘无关信息。

```

class LSTMRegressor(nn.Module):

```

```
def __init__(self, input_dim: int, hidden_size: int = HIDDEN_SIZE, num_layers: int = NUM_LAYERS,
dropout: float = DROPOUT):
    super().__init__()
    self.lstm = nn.LSTM(
        input_size=input_dim,
        hidden_size=hidden_size,
        num_layers=num_layers,
        dropout=dropout,
        batch_first=True,
    )
    self.head = nn.Sequential(
        nn.Linear(hidden_size, hidden_size // 2),
        nn.ReLU(),
        nn.Dropout(dropout),
        nn.Linear(hidden_size // 2, 1),
    )
```

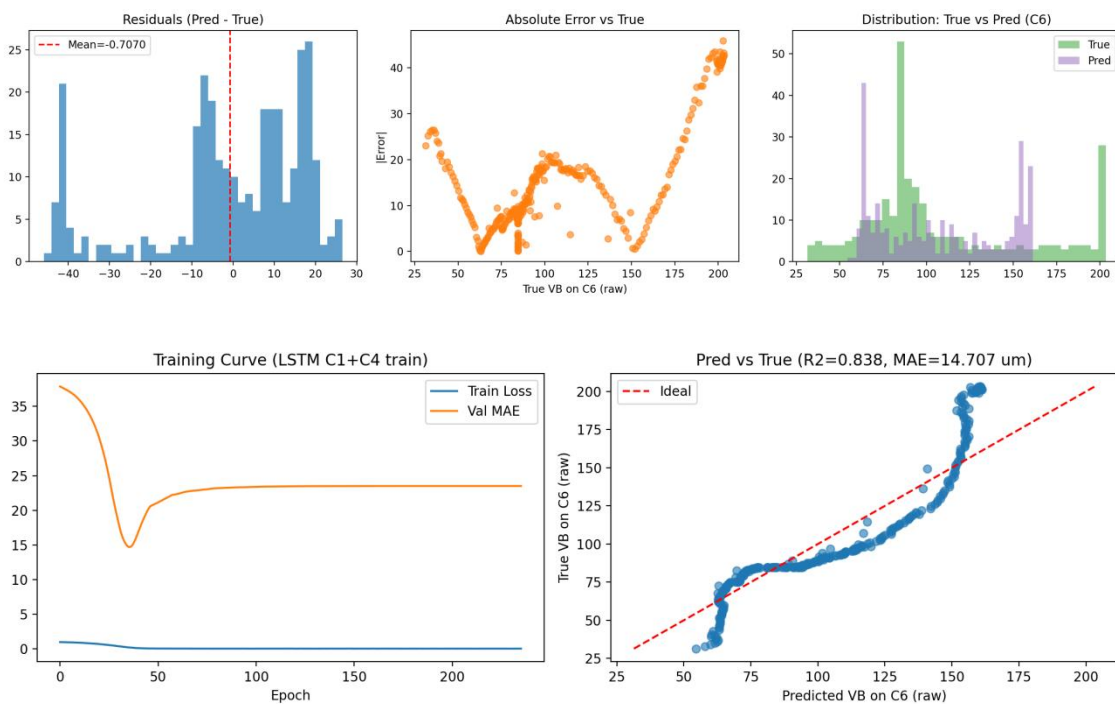


图9 LSTM训练效果可视化

本次实验中， $R^2 = 0.838$ 、 $MAE = 14.707 \mu m$ ，性能显著下降，原因分析如下：

理论上，刀具磨损是一个典型的时序累积过程，历史状态对当前磨损值有重要影响。因此，LSTM这类序列模型是自然的选择。然而，本研究的数据预处理流程存在一个关键点：在特征提取阶段，我们已经通过滑动窗口（长度1024点）将原始长时序信号转换为了一个个独立的、富含局部统计信息的12维特征向量。虽然我们在最后一步通过`create_sequences`函数构造了长度为10的序列样本，但这个“序列”的时间粒度（每个点代表约66.5ms的统计摘要）与LSTM擅长处理的原始采样点级别的细粒度时序存在本质差异。LSTM的优势在于挖掘原始信号波形中的动态模式，而我们的输入已经是高度抽象的统计特征。这种“特征工程先行”

的策略在一定程度上削弱了LSTM的序列建模优势，使其退化为一个带有内部记忆的复杂前馈网络，这可能是其性能未达预期的根本原因。

4.4 残差连接MLP（ResMLP）

残差网络通过引入“跳跃连接”（Skip Connection）或“恒等映射”（Identity Mapping）解决了深度网络训练中的退化问题。其核心思想是让网络学习残差函数 $F(\mathbf{x}) = H(\mathbf{x}) - \mathbf{x}$ ，而非直接学习目标映射 $H(\mathbf{x})$ 。一个基本的残差块（Residual Block）的输出可表示为：

$$y = F(\mathbf{x}, \{\mathbf{W}_i\}) + \mathbf{x}$$

其中 $F(\mathbf{x}, \{\mathbf{W}_i\})$ 通常由两层或三层FC层构成，从而缓解梯度消失问题，使得训练极深的网络成为可能。

```
class ResBlock(nn.Module):
    def __init__(self, dim: int, dropout: float):
        super().__init__()
        self.block = nn.Sequential(
            nn.Linear(dim, dim),
            nn.ReLU(),
            nn.Dropout(dropout),
            nn.Linear(dim, dim),
        )
        self.norm = nn.LayerNorm(dim)

    def forward(self, x: torch.Tensor) -> torch.Tensor:
        out = self.block(x)
        out = self.norm(out + x)
        return out
```

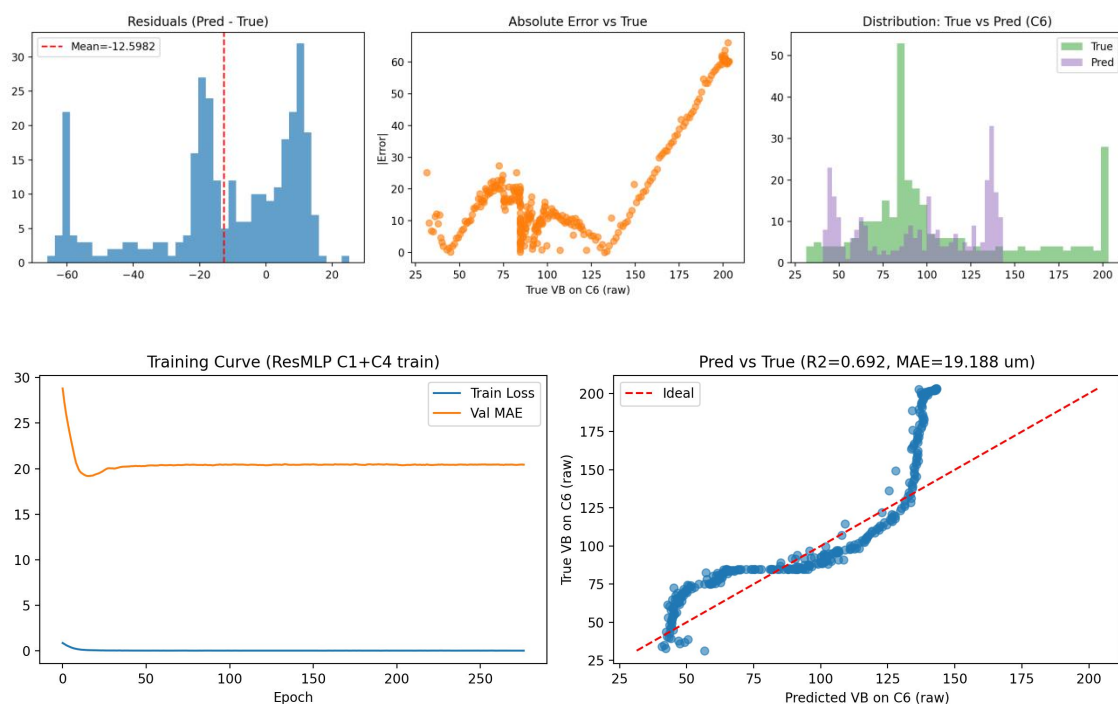


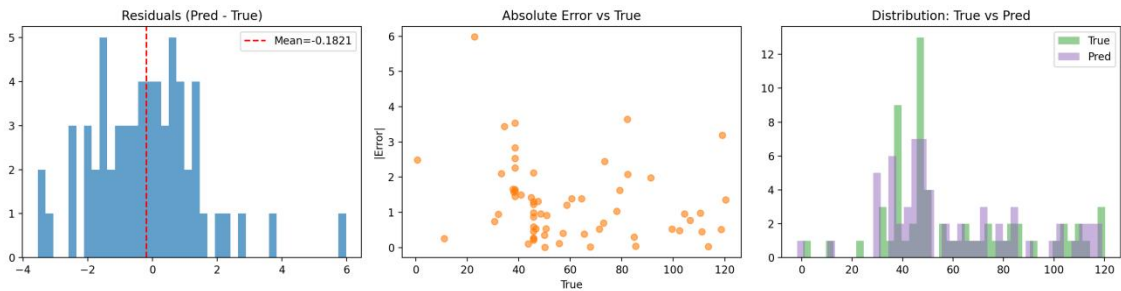
图10 ResMLP训练效果可视化

实验结果中， $R^2 = 0.692$ 、 $MAE = 19.188 \mu m$ ，效果不如基础MLP。主要原因在于任务与模型复杂度的不匹配。经过PCA降维后，输入特征维度仅为15维，这是一个相对低维且信息高度浓缩的空间。在此空间上构建过于复杂的残差结构（包含LayerNorm、多层非线性变换），极易导致模型过拟合。模型将过多的容量用于拟合训练集中的噪声，而非学习泛化的磨损规律，从而在测试集上表现出较差的泛化能力。这表明，对于特征工程充分、输入维度较低的任务，盲目增加模型深度并非良策。

4.5 集成学习（Ensemble Learning, EL）

集成学习通过结合多个“基学习器”的预测结果，利用“群体智慧”来获得比单一模型更优、更稳定的性能。其有效性源于两个方面：一是降低模型的方差，二是降低模型的偏差。本研究采用简单平均法作为集成策略，即对MLP、MLP+Attention、LSTM和ResMLP四个异构模型的预测输出进行算术平均：该策略简单有效，无需额外训练，且能有效平滑掉单个模型的极端错误预测。

```
def load_and_predict(models: List[str], X: np.ndarray, y_scaler, seq_mode=False, input_len=None):
    preds_all = []
    for name in models:
        net = model_dict[name]()
        net.load_state_dict(safe_load(ckpt))
        net.eval()
        with torch.no_grad():
            p = net(torch.tensor(X)).cpu().numpy().flatten()
            p_raw = y_scaler.inverse_transform(p.reshape(-1, 1)).flatten()
            preds_all.append(p_raw)
    return np.mean(preds_all, axis=0)
```



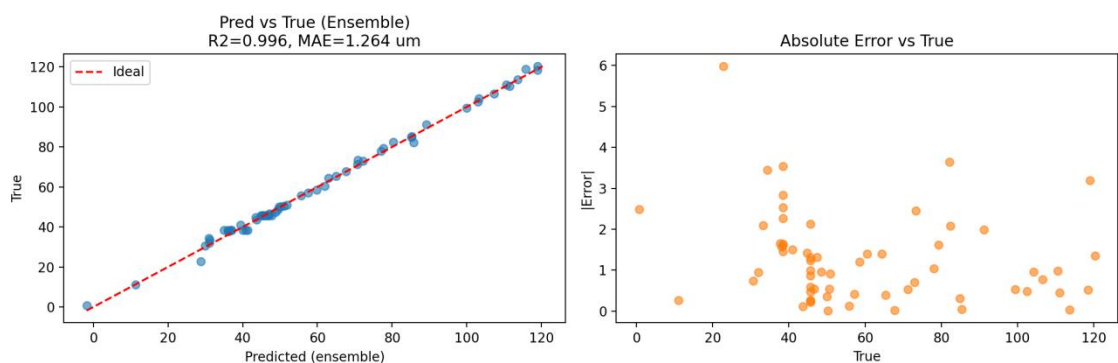


图11 集成学习训练效果可视化

集成模型取得了最佳性能： $R^2 = 0.996$ 、 $MAE = 1.264 \mu m$ ，超过所有单模型，其平均绝对误差低；残差接近零，分布集中；预测值与真实值几乎完全重合。

4.6 模型对比

模型	MAE (μm)	RMSE (μm)	R^2
MLP	8.453	10.412	0.95
ResMLP	19.188	25.912	0.692
MLP+Attention	7.664	10.185	0.952
LSTM	14.707	18.773	0.838
集成学习	1.264	1.674	0.996

由表格，MLP与MLP+Attention 表现优异，前者结构简洁，后者表现效果有提升；LSTM 因输入非时序特性而表现欠佳；ResMLP 过度复杂化导致性能下降；集成学习凭借多样性与稳定性，实现了性能质的飞跃。