PINN 正反问题中的架构设计与采样分布对精度与泛化的影响：基于 DeepXDE 的对比研究

2025 级　数学　刘倩妃

2026 年 1 月 10 日

1　实验环境

- 问题 1: 使用 DeepXDE + PyTorch，默认浮点精度为 `float64`，Adam 学习率 10^{-3}、迭代 20000 步，并切换 L-BFGS 进行二阶段优化。
- 问题 2: 使用 DeepXDE + PyTorch，默认浮点精度为 `float32`，Adam 学习率 10^{-3}、迭代 10000 步。

2　问题 1: PINN 求解 Lotka–Volterra 正问题

2.1　问题定义

Lotka–Volterra 模型常用于描述捕食-被捕食种群动力学。项目实现中对状态进行尺度化：设总体缩放常数 $U_B = 200$，时间尺度 $R_B = 20$，在区间 $t \in [0, 1]$ 上求解。令网络输出为 $\hat{y}(t) = [\hat{r}(t), \hat{p}(t)]$（均为尺度化量），并通过自动微分计算导数。对应残差形式（以项目实现为准）可写为

$$\mathcal{R}_r(t) = \frac{d\hat{r}}{dt} - \frac{R_B}{U_B}\Big(2U_B\hat{r} - 0.04(U_B\hat{r})(U_B\hat{p})\Big), \tag{1}$$

$$\mathcal{R}_p(t) = \frac{d\hat{p}}{dt} - \frac{R_B}{U_B}\Big(0.02(U_B\hat{r})(U_B\hat{p}) - 1.06(U_B\hat{p})\Big). \tag{2}$$

初值条件为

$$\hat{r}(0) = \frac{100}{U_B}, \qquad \hat{p}(0) = \frac{15}{U_B}. \tag{3}$$

2.2　PINN 训练目标：残差约束与初值约束

PINN 的典型损失由“方程残差 + 初值/边界误差”组成 [1]。本实验中重点对比了**硬初值约束（hard IC）**与不同输入特征编码方式。对于 hard IC，项目采用输出变换将初值条件结构化地写入网络输出（以实现为准）：

$$\tilde{r}(t) = y_1(t)\tanh(t) + \frac{100}{U_B}, \qquad \tilde{p}(t) = y_2(t)\tanh(t) + \frac{15}{U_B}, \tag{4}$$

其中 y_1, y_2 为网络“未约束输出”。该设计的优点是：无论参数如何更新，$t = 0$ 时总有 $\tanh(0) = 0$，从而严格满足初值；同时 $\tanh(t)$ 在小 t 区间近似线性，仍保留可训练自由度。

2.3 网络结构与特征编码

实验对比包含三类结构（均为 6 层宽度 64 的主干思路）：

- **FNN**：标准全连接网络；
- **PFNN**：分支式并行结构（在实现中体现为部分层用列表表示的并行通道）；
- **SkipFNN**：带残差/跳连的 MLP（提高优化稳定性与梯度流动效率）。

项目中对时间输入 t 使用两种特征变换：

- **sin-only**：拼接 $[t, \sin(t), \sin(2t), \ldots, \sin(6t)]$（共 7 维）；
- **Fourier 多尺度**：拼接 $[t, \sin(2\pi f t), \cos(2\pi f t)]$，频率 $f \in \{1, 2, 4, 8\}$（共 9 维）。

直观上，Lotka–Volterra 的解呈现明显的周期性与尖峰式振荡结构。Fourier 特征更贴近“谱空间表达”[3]，通常能缓解神经网络的低频偏置（spectral bias）并提升对高频/局部快速变化的拟合能力。

2.4 采样策略与训练设置

本报告展示的图均对应 **hardIC + pseudo_3k_b2** 的配置（见图标题）：

- 训练域点数：`num_domain=3000`（pseudo 分布）；
- 边界/初值点数：`num_boundary=2`（hard IC 下边界点更多是辅助稳定，而非必须满足）；
- Adam 阶段：学习率 10^{-3}，训练到 2×10^4 步量级后切换 L-BFGS 做二阶段拟合（图中 loss 横轴为 steps）。

2.5 实验结果与分析

2.5.1 Train/Test loss 的阶段性特征与异常现象

图1 展示了 Adam 阶段 train/test loss 随步数的变化。可以观察到两个值得强调的现象：

一、在训练初期（数千步以内），所有方法 loss 都快速下降若干数量级，说明 PINN 能迅速捕捉到动力学的大致趋势；之后进入平台/震荡区间，优化主要在细节与高频结构上“抠精度”。

二、橙色虚线（`FNN_tanh_sinfeat_hardIC__pseudo_3k_b2 | test`）在约 2.5×10^4 步后出现明显上升并长期维持较高水平；与之对比，Fourier 特征相关模型的 test loss 能继续下降或保持稳定。这一现象说明：**同样的 hard IC 与采样点数下，特征编码会显著改变模型的可泛化性**——并且这种差异不仅体现在最终误差，也体现在训练后期的稳定性上。

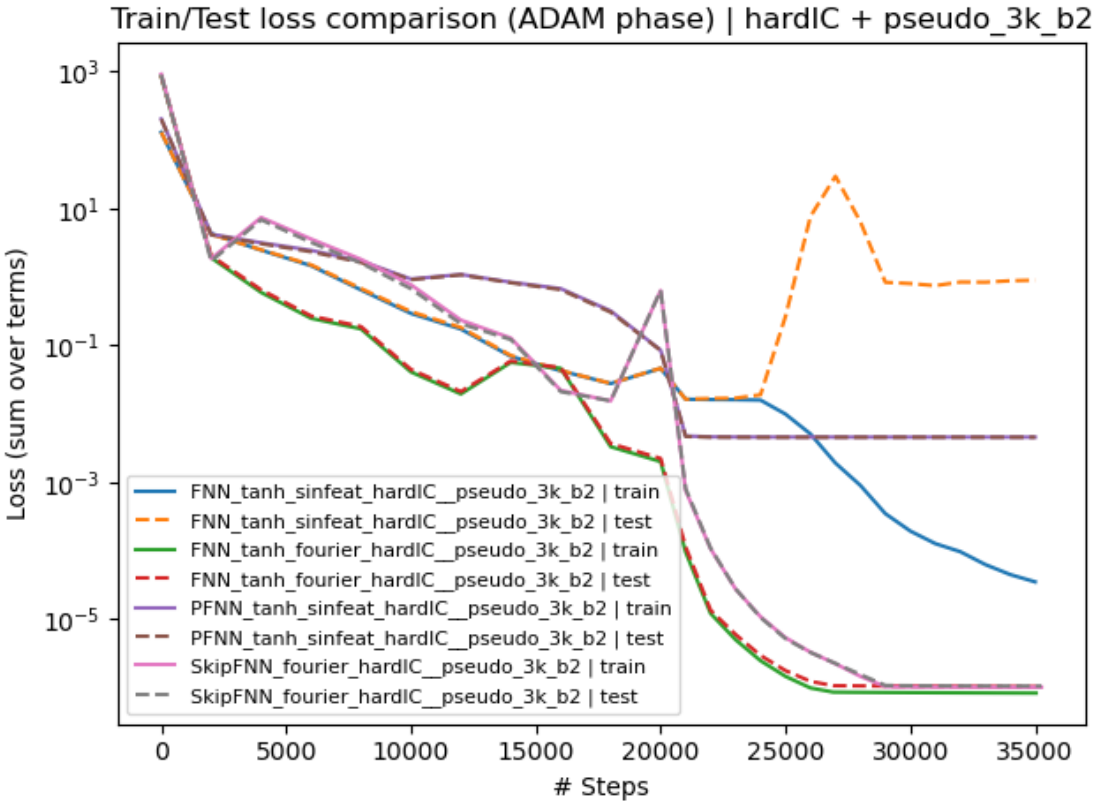

图 1: Adam 阶段 Train/Test loss 对比（hardIC + pseudo_3k_b2）。

2.5.2 预测曲线对比：Fourier 特征显著更稳健

图2 给出 $r(t)$ 与 $p(t)$ 在 $t \in [0, 1]$ 上的预测曲线。该曲线说明问题：

- **Fourier 特征 FNN（绿色虚线）与 SkipFNN（紫色虚线）**基本与真值重合，峰值位置与幅值都较准确；
- **sin-only 特征 FNN（橙色虚线）**很快塌陷到接近 0 的水平，与真实的周期峰值相去甚远；
- **PFNN（红色虚线）**能生成周期结构，但在部分峰附近存在可见的相位/幅值偏差，表明并行结构并非对该任务最优，或其优化更敏感。

这说明，对于带有尖峰的周期动力学，仅用 $\sin(kt)$（且不含 cos 互补项、也不含 2π 归一化频率）可能不足以提供稳定的“相位表达基底”；Fourier（sin+cos）多尺度特征能够提供更完整的周期基表示，从而让网络更容易逼近真实轨道。

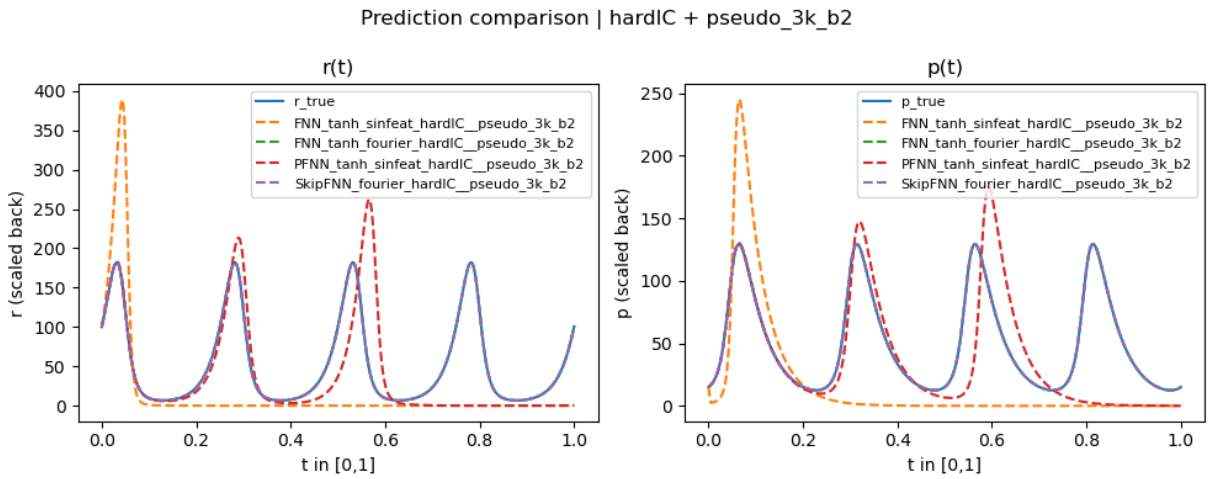

图 2: $r(t)$ 与 $p(t)$ 的预测对比（hardIC + pseudo_3k_b2）。

2.5.3 最优模型与误差量级

图3 展示了最优配置：`FNN_tanh_fourier_hardIC__pseudo_3k_b2`，其图中给出的整体相对误差为 `err_all=6.620e-04`。从曲线看，$r(t)$ 与 $p(t)$ 在多个周期峰的峰值、谷值、回落段均能与真值紧密贴合，说明此配置在表达能力、优化稳定性与约束实现上达成了较好的平衡。

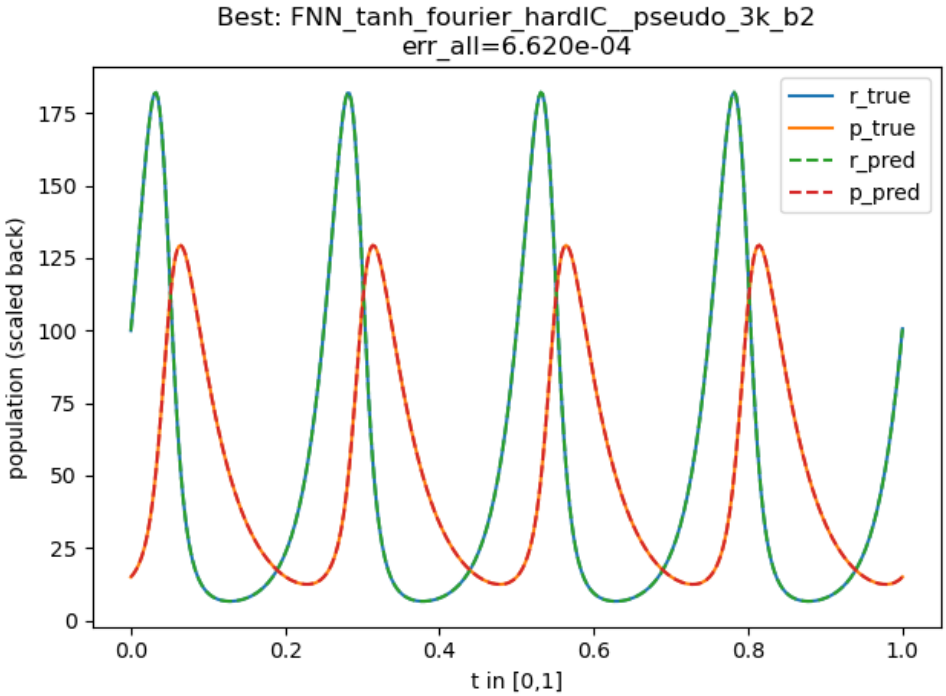

图 3: 最优模型预测效果（err_all=6.620e-04）。

2.6 讨论：为何 Fourier + hardIC 更有效?

2.6.1 谱偏置与周期系统的"频域捷径"

经典现象是：标准 MLP 在训练早期更容易学习低频成分（谱偏置） [4]。而 Lotka–Volterra 的解虽然整体周期，但峰附近存在较强的局部高频变化。Fourier 特征将输入映射到更适合表示周期的空间，使网络更容易在较少参数更新下捕捉相位与频率结构，从而减少后期不稳定振荡或塌陷。

2.6.2 硬约束降低可行域维度，提升后期稳定性

hard IC 在结构上保证初值严格满足，相当于从优化角度减少了"先拟合初值再拟合残差"的竞争关系，使得优化更多专注于动力学残差。对敏感系统而言，初值的微小偏差会随时间放大；hard IC 能显著降低误差传播起点的不确定性，这对长程稳定预测尤其关键。

2.6.3 结构选择：SkipFNN 的优势与 PFNN 的敏感性

从图2可见 SkipFNN（紫色虚线）与 Fourier FNN（绿色虚线）都表现稳定。一般来说，残差连接有助于缓解深层网络训练中的梯度消失/鞍点停滞，使得在相同深度下更容易优化；而 PFNN 的并行结构在该实验中未体现出明显优势，可能原因包括：并行通道之间的特征分工未被充分利用，或超参数（宽度/深度/初始化）对 PFNN 更敏感。

2.7 小节总结

在相同采样与 hard IC 约束下，输入特征编码对 PINN 的训练稳定性与最终精度具有决定性影响；Fourier（sin+cos，多尺度）特征显著优于 sin-only 特征，并能与 FNN/SkipFNN 结构形成稳定组合。

3 问题 2: DeepONet 学习算子（Antiderivative）

3.1 问题定义

目标算子为

$$\mathcal{G} : f(x) \mapsto u(x) = \int_0^x f(s)\, ds, \qquad x \in [0, 1]. \tag{5}$$

DeepONet 的基本形式为 [2]

$$\hat{u}_\theta(x; f) = \sum_{k=1}^{p} b_\theta^{(k)}(f(x_1), \ldots, f(x_m))\; t_\theta^{(k)}(x), \tag{6}$$

其中 Branch 网络将离散化的输入函数值映射为潜在表征向量，Trunk 网络将位置 x 映射为基函数表征。本次实验使用 `deepxde.nn.tensorflow.deeponet` 模块实现，将一个 batch 的多函数输入与一个 trunk 点集进行笛卡尔积组合预测。

对不定积分而言，天然边界条件为 $u(0) = 0$。项目在 E6 中引入 hard IC，通过输出变换把该条件结构化写入：

$$\tilde{u}(x) = x \cdot \hat{u}(x). \tag{7}$$

由于 $x = 0$ 时 $\tilde{u}(0) = 0$ 恒成立，从而严格满足边界条件。这种约束同样具有"减少优化冲突、提高边界一致性"的作用，但它也会改变函数的有效表达空间（例如在 x 很小时会放大相对误差敏感性），因此需要与采样与正则化配合。

在 E6 中，trunk 输入还做了两步处理：

- 将 $x \in [0, 1]$ 线性映射到 $[-1, 1]$；
- 拼接 Fourier 特征：$[x, \sin(kx), \cos(kx)]$，$k = 1, \ldots, 8$（附带比例系数）。

该设计增强了 trunk 基函数的表达能力，使其更容易生成复杂的函数形状；但在 trunk 采样稀疏时，也更容易出现插值不稳定。

3.2 实验设计：E1–E6 的对比维度

本次实验使用训练/测试数据集：`antiderivative_aligned`。训练数据集包含 $n_{\text{train}} = 150$ 个函数样本，测试数据集包含 $n_{\text{test}} = 1000$ 个函数样本；每个输入函数在 $m = 100$ 个 sensors 位置采样，输出 $u(x)$ 在 100 个 trunk 位置给出真值（full trunk）。问题 2 的实验对比围绕四个关键因素：

1. **sensors 数量**：100（full）vs 20；
2. **trunk 数量**：100（full）vs 20；
3. **trunk 采样分布**：uniform vs betaBoundary（更偏向边界附近采样）；
4. **Branch 架构与特征**：MLP / CNN branch / ResMLP branch + trunk Fourier + hardIC。

表 1 汇总对照实验 E1–E6 6 种配置：

表 1: E1–E6 实验配置

实验	sensors	trunk	trunk 采样	网络与关键设置
E1	100	100	uniform	MLP-MLP，latent=64，ReLU，no hardIC，no Fourier
E2	20	100	uniform	MLP-MLP，latent=64，ReLU，no hardIC
E3	20	20	uniform	MLP-MLP，latent=64，ReLU，no hardIC
E4	20	20	betaBoundary	MLP-MLP，latent=64，ReLU，no hardIC
E5	20	20	uniform	CNN Branch，latent=64，ReLU，no hardIC
E6	20	20	betaBoundary	ResMLP Branch，latent=128，ReLU，hardIC，trunk: scale+Fourier(8)

所有实验采用 Adam 优化器、学习率 10^{-3}、训练 10000 步。以固定间隔抽样记录 train loss、test loss 以及一个在 trunk 点上的相对误差指标（mean rel L2）。

项目中使用的 mean relative L2（对每个函数样本先算相对 L2，再对样本取均值）：

$$\mathrm{RelL2} = \frac{1}{N}\sum_{i=1}^{N}\frac{\|u_i - \hat{u}_i\|_2}{\|u_i\|_2 + \varepsilon}. \tag{8}$$

根据最终数据进一步统计 full grid 上的均值相对误差。

3.3 实验结果与分析

图 4 汇总了 E1–E6 的三条曲线（train loss、test loss、trunk 指标）。可以看到：

- 所有实验在 0–1000 步内迅速下降，说明该算子学习任务在数据驱动下较易被拟合；
- 之后进入平台期，各方法的 test loss 与 trunk 指标差异并不巨大；
- 从 trunk 指标看，E6 在中后期甚至可能呈现较优水平（曲线更低）。

如果只依据这组三图，直觉上会认为：E6（更强的 branch、更强的 trunk 表达、更硬的边界约束）应当是“更强模型”。

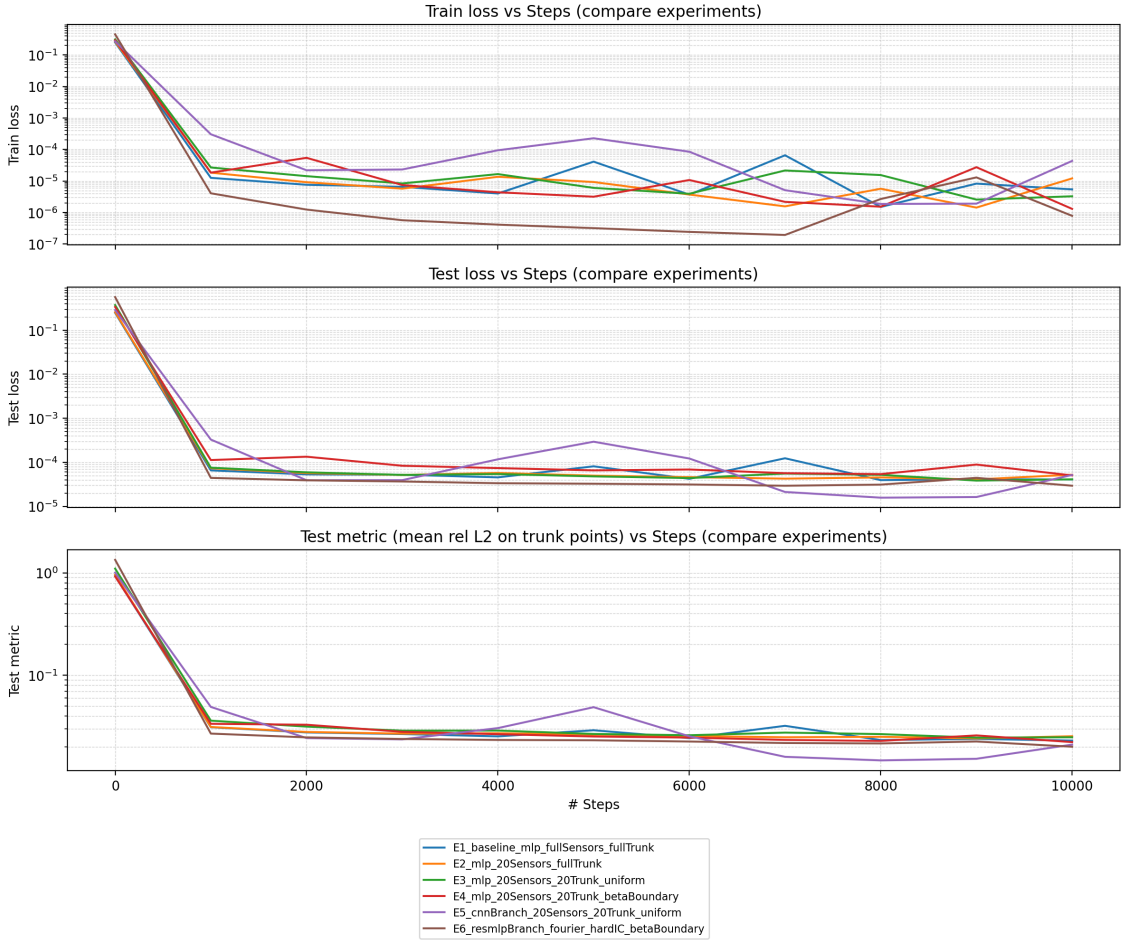

图 4: E1–E6 的 Train loss / Test loss / trunk 指标对比

图 5 展示 train loss 的 running-min 曲线。E6 在全程显著低于其他实验（最低可到 10^{-7} 量级附近），说明它在训练目标上具有更强的优化与拟合能力。一般而言，这对应于更大的模型容量（latent=128）、更强的特征表达（trunk Fourier）以及 hardIC 对边界误差的消除。

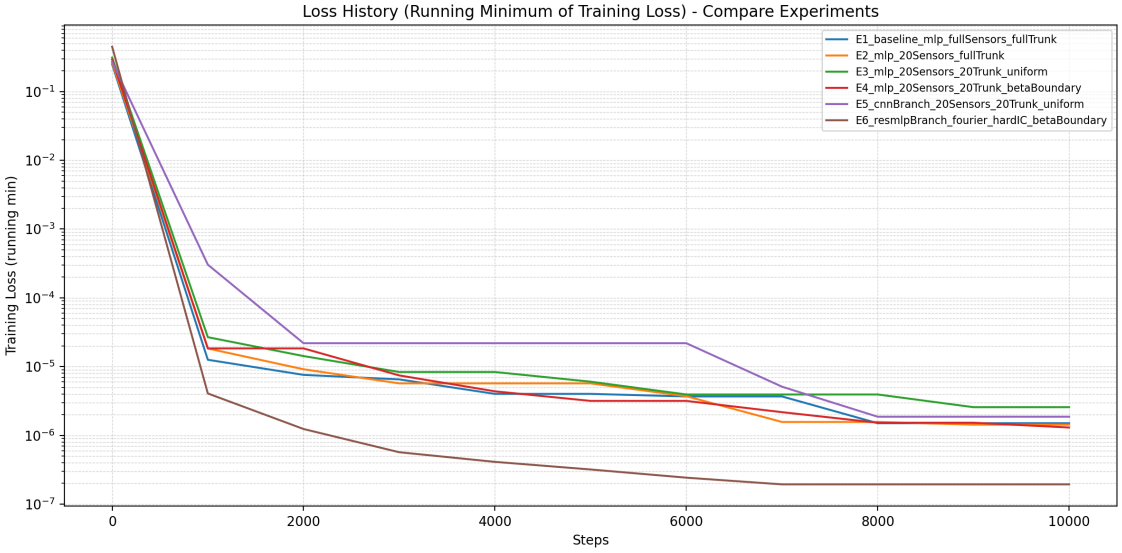

图 5: 训练损失 running-min 对比

图 6 在同一个测试样本上对比 full grid 的 $\hat{u}(x)$ 曲线。

结果非常关键：E1–E5 的预测几乎与真值重合（只有轻微偏差），而 E6（粉色虚线）出现明显的高频振荡、尖峰和局部偏离。

这说明：**E6 可能在训练所采样的 trunk 点上拟合得很好，但在点与点之间插值/外推时产生了非物理的高频波动**。由于不定积分 $u(x)$ 本应相对平滑（若 f 有界），这种振荡通常不是目标函数真实结构，而是模型自由度过大、采样过稀、缺少平滑先验共同作用的结果。

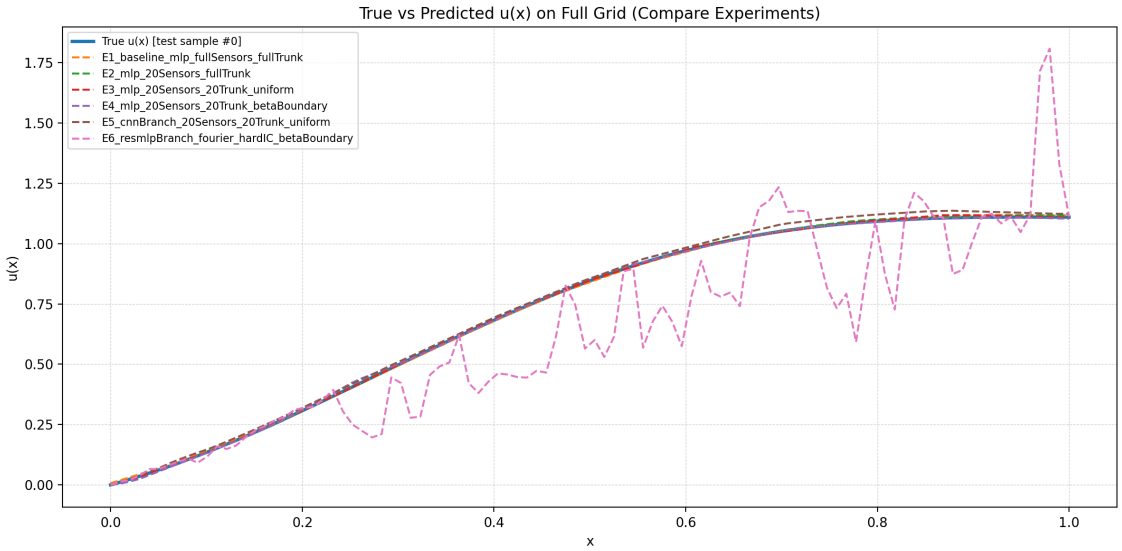

图 6: full grid 上 True vs Pred 对比

图 7 给出 full grid 的 mean relative L2 error。可以看到 E6 的误差条远高于其他实验（约 0.4 量级），而 E1–E5 都处于较小水平。

因此，可以得出：

训练损失或稀疏 trunk 点上的 test metric 并不必然等价于全域泛化质量；当模型表达过强而 trunk 采样稀疏时，可能出现“点上对、点间错”的高频振荡过拟合。

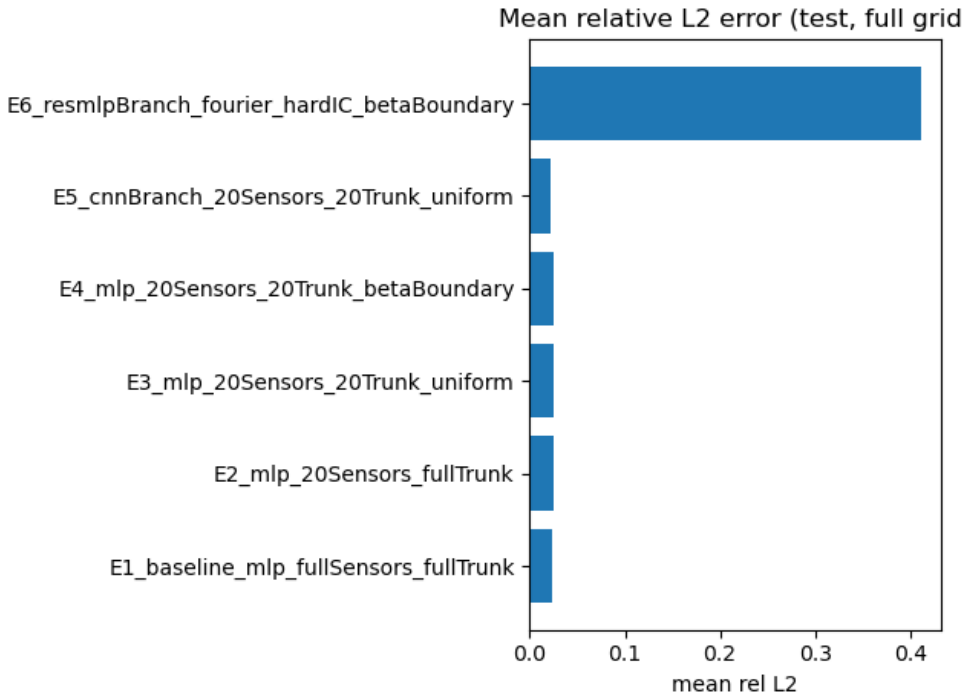

图 7: full grid mean relative L2 error 对比

E6 为什么会训练更好但泛化更差?

trunk Fourier 特征提升表达，也提升了振荡风险。Fourier 特征等价于给 trunk 提供了一组高频基函数。若 trunk 点数足够多，它可以更好逼近复杂函数；但当 trunk 仅有 20 个点时，模型可能利用高频自由度在这些点上“对齐”，同时在点间产生快速摆动，从而导致 full grid 误差显著上升。

betaBoundary 采样强调边界，可能弱化内部约束。betaBoundary 的直观目标是让模型更重视边界附近的精度（对积分问题边界确实重要）。但当 trunk 总点数很少时，过度偏向边界会减少内部区间的有效约束密度，使内部插值更不稳定。

hardIC 强化边界一致性，但不等价于“整体平滑先验”。hardIC 仅保证 $u(0) = 0$（式 $），它并不直接抑制 $u(x)$ 的振荡。换言之，hardIC 是“边界正确”的保证，但不是“形状合理”的保证。若没有额外的平滑正则或更密集的 trunk 覆盖，高频自由度仍可能导致不合理形状。

为使更强表达（ResMLP + Fourier + hardIC）的优势真正转化为 full grid 的收益，分析得出如下改进方向：

- 在训练过程中定期用更密集的 trunk（例如 100 或 200 点）计算验证误差，而不是只用 20 个 trunk 点；
- 在每个 step 随机重采样 trunk 点（stochastic trunk sampling），使模型无法只记住固定点位。
- 对积分问题，边界与内部都重要。可以采用混合采样：

$$x_{\text{trunk}} \sim \lambda \cdot \text{Uniform}(0,1) + (1-\lambda) \cdot \text{BetaBoundary}, \tag{9}$$

  在保证边界覆盖的同时不牺牲内部密度。
- 对 trunk Fourier 频率做截断（减小 K 或降低 Fourier scale）；
- 对输出曲线加入平滑正则（例如对离散网格上的二阶差分 $\|\Delta^2 u\|$ 做惩罚）；
- 更贴合算子结构的方式：利用 $u'(x) = f(x)$，在训练中加入导数一致性项（软约束）：

$$\mathcal{L}_{\text{der}} = \|\partial_x \hat{u}(x) - f(x)\|^2,$$

  该项能够直接抑制“积分结果出现高频波动但导数不匹配”的情况。

3.4 小节总结

在算子学习/函数学习任务中，如果验证仅发生在稀疏点集上，强表达模型可能通过高频振荡在点上取得低误差，从而掩盖真实的全域泛化失败；full grid 评估与平滑先验是必要的。

4 结论

综合两个任务，可以得到一条贯穿性的经验规律：

1. **表达方式要与问题结构匹配**：周期/振荡系统（问题 1）更适合 Fourier 类特征；算子学习的 trunk（问题 2）引入 Fourier 可提升容量，但需同时提升采样密度或引入平滑先验。
2. **硬约束是“边界正确”的工具，而非“全域正确”的保证**：hard IC 能显著减少边界误差与训练冲突，但仍可能出现点间振荡或形状不合理，需要采样与正则协同。
3. **评价指标必须覆盖真实使用场景**：问题 2 表明仅看 train/test loss 或稀疏 trunk metric 可能被误导；full grid 评估更能反映实际泛化质量。

参考文献

[1] M. Raissi, P. Perdikaris, and G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *Journal of Computational Physics*, 2019.

[2] L. Lu, P. Jin, and G. E. Karniadakis, Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators, *Nature Machine Intelligence*, 2021.

[3] M. Tancik et al., Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains, *NeurIPS*, 2020.

[4] R. Rahaman et al., On the Spectral Bias of Neural Networks, *ICML*, 2019.