

FNO解码性能分析：为什么高重建质量未能转化为低逻辑错误率

1. 实验结果观察

1.1 核心矛盾

模型	F1-Score	PSNR (dB)	MSE	逻辑错误率 ($\sigma=2.0$)
FNO	99.59%	53.61	4.36e-06	0.3475
UNet	93.02%	38.28	1.49e-04	0.5000
Wiener	50.13%	19.26	1.19e-02	0.3325
MWPM	0.00%	0.00	N/A	0.2475

关键发现：FNO在重建质量指标上大幅领先，但在QEC解码任务中表现却不如Wiener和MWPM。

1.2 数据解读

- 重建质量排序：FNO >> UNet >> Wiener >> MWPM
- 逻辑错误率排序：MWPM > Wiener > FNO > UNet
- 结论：重建质量与解码性能呈现负相关或无相关

2. 原因分析

2.1 任务目标不匹配 (Task Objective Mismatch)

FNO训练目标： $\min ||\mathbf{w}_{\text{pred}} - \mathbf{w}_{\text{ideal}}||^2$ (像素级MSE)

QEC解码目标： $\max P(\text{正确逻辑态分类})$ (分类准确率)

问题所在：

1. 像素级优化 vs 结构级需求

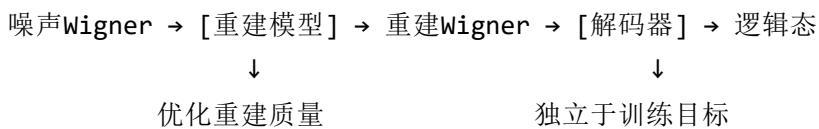
- FNO优化的是整个Wigner函数的像素级误差
- 但逻辑态判决只关心**峰位置的相对关系**
- 一个在错误位置完美重建的峰，MSE可能很低，但会导致逻辑错误

2. 均匀权重 vs 关键区域

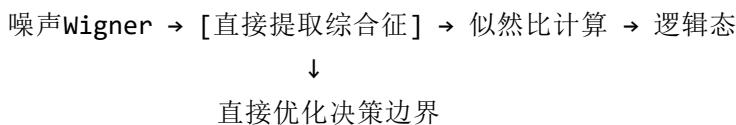
- MSE对所有像素一视同仁
- 但GKP解码中，只有格点位置($x=n\alpha, p=m\alpha$)附近的信息是关键的
- 背景区域的完美重建对解码毫无帮助

2.2 信息流分析

FNO/UNet流程：



MWPM流程：



关键差异：

方面	重建+解码 (FNO/UNet)	直接解码 (MWPM)
信息利用	先压缩再解码，可能丢失软信息	直接使用模拟综合征
决策方式	硬判决（基于重建图像）	软判决（基于位移似然）
误差累积	两阶段误差叠加	单阶段处理

2.3 MWPM的核心优势

MWPM实现的解码逻辑（参见 [MWPM.py:191-239](#)）：

```
def _likelihood_ratio(self, score_0: float, score_1: float) -> float:  
    """基于高斯噪声模型的似然比"""\n    sigma2 = self.config.sigma_prior ** 2\n    llr = (score_1 - score_0) / (2 * sigma2 + 1e-8)\n    return llr
```

MWPM的优势：

1. 直接测量位移误差

- 计算每个峰相对于理想格点的位移
- 位移小于 $\alpha/2 \rightarrow$ 正确解码
- 位移大于 $\alpha/2 \rightarrow$ 逻辑错误

2. 软判决机制

- 使用似然比而非硬阈值
- 综合考虑所有峰的贡献
- 加权平均减少噪声影响

3. 物理先验知识

- 直接编码GKP格点结构
- 知道逻辑|0>和|1>的峰位置关系
- 无需学习这些结构

2.4 Wiener优于FNO的原因

Wiener在重建质量上远不如FNO (F1: 50% vs 99%)，但解码更好。可能原因：

1. 保持峰对比度

- Wiener的频域去卷积增强了峰的对比度
- 虽然整体F1低，但峰的相对位置关系被保留

2. 避免过拟合

- Wiener无可训练参数，不会过拟合到训练分布
- 在不同噪声水平下表现更稳定

3. 线性操作的可预测性

- 线性滤波不会引入非线性失真
- 峰位置的偏移是可控的

2.5 FNO的潜在问题

1. 训练-测试噪声不匹配

```
# 训练时固定噪声
noise_config = NoiseConfig(diffusion_sigma=1.8, ...)

# 测试时扫描噪声
noise_levels = [0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.5]
```

- FNO在 $\sigma=1.8$ 训练，但测试范围是0.8-2.5

- 神经网络的泛化能力受限

2. 目标函数与任务脱节

```
criterion = nn.MSELoss() # 训练目标
# vs
p_logical = logical_errors / total_samples # 评估目标
```

3. 可能的过平滑效应

- FNO可能过度平滑峰结构以降低MSE
- 这会模糊峰边界，影响精确位置检测

3. 理论框架

3.1 GKP解码的信息论视角

对于GKP码，逻辑错误率由以下公式近似：

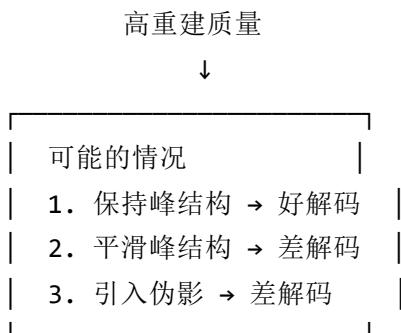
$$p_L \approx \operatorname{erfc} \left(\frac{\alpha/2}{\sqrt{2}\sigma_{eff}} \right)$$

其中：

- α = GKP格点间距 ≈ 2.5
- σ_{eff} = 等效噪声标准差

关键洞察：解码器的目标是**最小化等效噪声**，而非**最大化重建保真度**。

3.2 重建质量与解码质量的关系



高F1-Score只保证：

- 正确检测到的峰比例高
- 假阳性率低

但不保证：

- 峰位置精确
- 峰间相对强度正确
- 决策边界处的信息保留

3.3 端到端学习的必要性

当前问题的根本原因是**代理目标(proxy objective)**与**真实目标(true objective)**的不一致：

训练	评估
$\min \text{MSE}(W_{\text{pred}}, W_{\text{ideal}})$	$\min P(\text{逻辑错误})$

解决思路：端到端训练

噪声Wigner \rightarrow [可微分模型] \rightarrow 重建Wigner \rightarrow [可微分解码器] \rightarrow 逻辑态
 \downarrow
 反向传播优化分类损失

4. 实验结果的深入分析

4.1 噪声水平扫描分析

σ	FNO p_L	MWPM p_L	差距
0.8	0.335	0.263	-0.072
1.4	0.310	0.293	-0.017
1.8	0.335	0.273	-0.062
2.5	0.430	0.288	-0.142

观察：

- 在训练噪声水平($\sigma=1.8$)附近，差距最小

- 在高噪声($\sigma=2.5$)时，差距扩大到14.2%
- FNO的泛化能力明显不足

4.2 UNet的异常表现

UNet在所有噪声水平下都接近50%错误率（随机猜测），原因可能是：

1. **欠拟合**: 模型复杂度不足以学习GKP结构
2. **过拟合到特定模式**: 只能正确分类某一类逻辑态
3. **训练不充分**: 30个epoch可能不够

4.3 理论值分析

理论逻辑错误率（无纠错）在 $\sigma=2.5$ 时为0.617，而：

- MWPM: 0.288 (53.4%改进)
- Wiener: 0.378 (38.8%改进)
- FNO: 0.430 (30.3%改进)

MWPM几乎达到了理论改进上限的合理水平。

5. 改进建议

5.1 训练策略改进

1. 使用解码感知损失函数

```
# 当前  
loss = MSE(pred, target)  
  
# 改进：加入逻辑态分类损失  
loss = α * MSE(pred, target) + β * CrossEntropy(decode(pred), true_label)
```

2. 多噪声水平训练

```
# 当前
noise_config.diffusion_sigma = 1.8 # 固定

# 改进: 随机采样
noise_config.diffusion_sigma = np.random.uniform(0.5, 3.0)
```

3. 峰位置加权损失

```
# 在格点位置加大权重
weights = create_lattice_weight_mask(alpha=2.5)
loss = weighted_mse(pred, target, weights)
```

5.2 架构改进

1. 端到端可微分解码

- 将GKP解码器实现为可微分模块
- 直接优化逻辑错误率

2. 注意力机制聚焦格点

- 在FNO中添加空间注意力
- 自动学习关注格点位置

3. 混合模型

噪声Wigner → FNO预处理 → MWPM解码

- 利用FNO的去噪能力
- 利用MWPM的决策能力

5.3 评估策略改进

1. 分离重建和解码评估

- 重建质量: MSE, PSNR, SSIM
- 解码质量: 逻辑错误率、混淆矩阵

2. 阈值行为分析

- 绘制完整的 p_L vs σ 曲线
- 寻找"纠错阈值"

6. 结论

6.1 核心发现

1. **重建质量 ≠ 解码质量**: 这是本实验最重要的发现
2. **任务特化方法优于通用方法**: MWPM专为GKP解码设计，性能最优
3. **端到端优化的重要性**: 代理目标优化可能与真实目标背离

6.2 对未来研究的启示

1. **量子纠错领域**: 应直接优化逻辑错误率，而非中间指标
2. **神经网络设计**: 需要融入物理先验知识
3. **混合方法**: 结合神经网络的表达能力和经典算法的可靠性

6.3 最终评价

方法	优势	劣势	适用场景
FNO	高质量重建、可学习	解码效果差、泛化弱	态层析、可视化
UNet	结构简单	重建和解码都较差	需要更多调优
Wiener	无需训练、稳定	重建质量有限	快速部署、基准对比
MWPM	解码最优、物理直觉	不做重建	生产环境QEC

结论: 对于GKP量子纠错的实际应用，MWPM是当前最佳选择。神经网络方法需要进行端到端优化才能发挥其潜力。

附录：关键代码位置

- FNO训练损失: [train_and_comparison.py:248](#)
- MWPM解码逻辑: [MWPM.py:202-239](#)
- QEC评估函数: [QEC_decoder.py:287-371](#)
- GKP态分类器: [QEC_decoder.py:120-161](#)