

GKP量子纠错实验分析文档

1. 实验概述

1.1 研究背景

GKP (Gottesman-Kitaev-Preskill) 码是一种将离散量子比特编码到连续变量 (Continuous Variable, CV) 系统中的量子纠错码。它将逻辑量子比特编码在谐振子的无限维希尔伯特空间中，通过相空间中的周期性结构实现纠错。

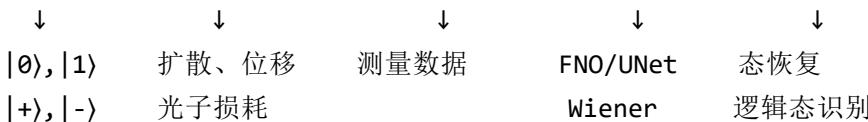
1.2 实验目的

本实验的核心目标是：

- GKP态重建 (GKP State Reconstruction)**：从噪声测量数据中恢复理想的GKP量子态
- GKP解码 (GKP Decoding)**：识别噪声态对应的逻辑量子比特状态
- 方法对比**：比较神经网络方法 (FNO、UNet) 与传统信号处理方法 (Wiener滤波) 的性能

1.3 实验流程

理想GKP态 → 噪声通道 → 噪声Wigner函数 → 解码器 → 重建态/逻辑判决



2. 物理模型详解

2.1 GKP逻辑态

理想GKP码的逻辑态定义在相空间中：

逻辑态	相空间结构	Wigner函数峰位置
$ 0_L\rangle$	x方向偶数格点	$x = 2n\alpha, p = m\alpha$

逻辑态	相空间结构	Wigner函数峰位置
$ 1_L\rangle$	x方向奇数格点	$x = (2n+1)\alpha, p = m\alpha$
$ +_L\rangle$	p方向偶数格点	$x = n\alpha, p = 2m\alpha$
$ -_L\rangle$	p方向奇数格点	$x = n\alpha, p = (2m+1)\alpha$

其中 $\alpha = \sqrt{(2\pi)} \approx 2.507$ 是GKP格点间距。

2.2 有限能量GKP态

实际物理系统中，GKP态必须是有限能量的。我们通过高斯包络函数实现：

$$W_{finite}(x, p) = W_{ideal}(x, p) \cdot \exp\left(-\frac{x^2 + p^2}{2\sigma_{env}^2}\right)$$

2.3 噪声模型

本实验模拟了以下物理噪声过程：

噪声类型	物理来源	数学描述
扩散噪声	有限挤压、退相干	高斯卷积 σ_{diff}
位移误差	外部场扰动	相空间平移 $(\delta x, \delta p)$
旋转误差	振荡器频率偏移	相空间旋转 θ
热噪声	环境热激发	背景高斯 n_{th}
光子损耗	振幅阻尼通道	相空间收缩 $\sqrt{\eta}$

3. 任务类型分析

3.1 态重建 vs 解码

本实验实际上包含两个相关但不同的任务：

3.1.1 GKP态重建 (State Reconstruction)

目标：从噪声Wigner函数恢复理想Wigner函数

输入：噪声测量 $W_{\text{noisy}}(x,p)$

输出：重建态 $W_{\text{reconstructed}}(x,p)$

评估：与理想态 $W_{\text{ideal}}(x,p)$ 的相似度

这是一个**图像去噪/反卷积**问题，当前代码主要解决的是这个任务。

3.1.2 GKP解码 (Decoding)

目标：从噪声态判断逻辑量子比特状态

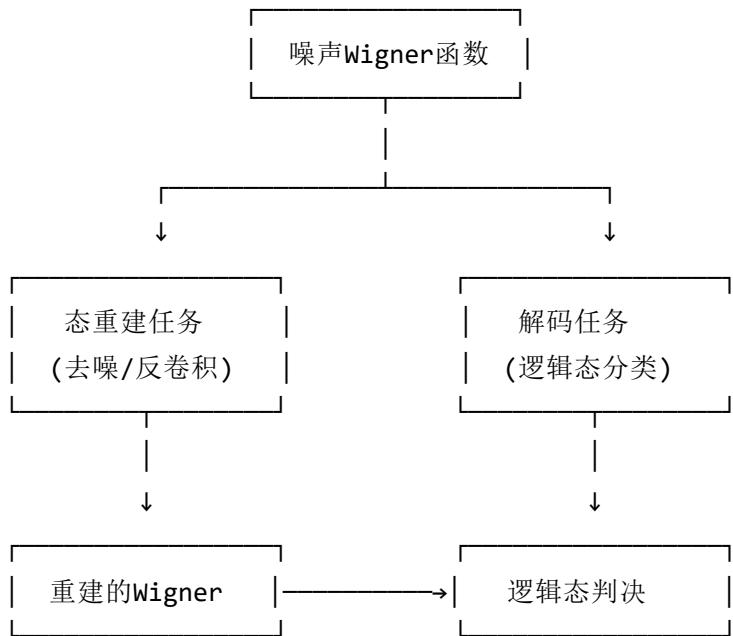
输入：噪声测量 $W_{\text{noisy}}(x,p)$

输出：逻辑态标签 $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$

评估：逻辑判决的正确率

这是一个**分类**问题，需要从相空间结构推断逻辑信息。

3.2 两者的关系



好的态重建通常会带来好的解码性能，但两者并非完全等价。

4. 评估指标分析

4.1 当前使用的指标

指标	定义	物理意义
MSE	$\sum(W_{\text{pred}} - W_{\text{ideal}})^2 / N$	重建误差的平方平均
PSNR	$10 \cdot \log_{10}(1/\text{MSE})$	信号质量的对数度量
F1-Score	$2 \cdot P \cdot R / (P+R)$	峰检测的综合性能
Precision	$TP / (TP+FP)$	检测到的峰的准确性
Recall	$TP / (TP+FN)$	真实峰的召回率
SSIM	结构相似性	结构保持程度

4.2 量子纠错领域的标准指标

在量子纠错领域，最重要的两个指标是：

4.2.1 物理错误率 (Physical Error Rate, p)

定义：底层物理系统发生错误的概率

在GKP码中的对应：

- 相空间中的位移误差大小： $\sigma_{\text{displacement}}$
- 噪声强度参数： σ_{noise}
- 单次操作的保真度损失

计算方式：

```
# 方法1: 基于位移误差
p_physical = 1 - exp(-σ_displacement² / (2·α²))

# 方法2: 基于噪声强度
p_physical = σ_noise / α # 简化估计
```

4.2.2 逻辑错误率 (Logical Error Rate, p_L)

定义：纠错后逻辑量子比特发生错误的概率

在GKP码中的对应：

- 解码后逻辑态判决错误的比例
- 位移跨越决策边界导致比特翻转的概率

计算方式：

```
# 从实验数据计算
p_logical = (错误分类的样本数) / (总样本数)

# 或从重建态计算
for sample in test_data:
    reconstructed = model(noisy_sample)
    predicted_logical_state = classify_gkp_state(reconstructed)
    if predicted_logical_state != true_logical_state:
        logical_errors += 1
p_logical = logical_errors / total_samples
```

4.3 能否使用物理/逻辑错误率？

答案：可以，但需要适当定义和实现。

4.3.1 当前代码的局限性

当前代码的指标（MSE、F1等）更适合评估**态重建质量**，而非直接衡量**量子纠错能力**。

4.3.2 添加逻辑错误率的方案

```
def compute_logical_error_rate(model, test_loader, gkp_config):
    """
    计算逻辑错误率

    GKP解码决策规则：
    - 测量重建态在x=0和x=α/2处的积分强度
    - 如果x=0附近更强 → |0⟩
    - 如果x=α/2附近更强 → |1⟩
    """

    logical_errors = 0
    total = 0

    for noisy, ideal, true_label in test_loader:
        reconstructed = model(noisy)

        # 从重建的Wigner函数推断逻辑态
        predicted_label = classify_gkp_state(reconstructed, gkp_config)

        # 统计错误
        logical_errors += (predicted_label != true_label).sum()
        total += len(true_label)

    return logical_errors / total

def classify_gkp_state(wigner, config):
    """
    GKP态分类器

    基于相空间峰位置判断逻辑态

    # 计算x方向偶数/奇数格点的强度
    even_x_intensity = integrate_at_positions(wigner, x_positions='even')
    odd_x_intensity = integrate_at_positions(wigner, x_positions='odd')

    # 计算p方向偶数/奇数格点的强度
    even_p_intensity = integrate_at_positions(wigner, p_positions='even')
    odd_p_intensity = integrate_at_positions(wigner, p_positions='odd')

    # 决策逻辑
    if even_x_intensity > odd_x_intensity:
        if even_p_intensity > odd_p_intensity:
```

```
    return LogicalState.ZERO # |0> or |+>, 需要更细致判断
# ... 完整决策树
```

4.3.3 物理错误率与逻辑错误率的关系

对于理想GKP码，存在理论上的阈值行为：

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

其中 σ 是等效噪声标准差。当 $\sigma < \alpha/(2\sqrt{2})$ 时，逻辑错误率被指数压制。

神经网络解码器的目标是：

1. **降低等效噪声**：通过去噪减小有效的 σ
2. **软判决**：比硬阈值判决更好地利用信息

5. 建议的改进方案

5.1 添加逻辑错误率指标

在 train_and_comparison.py 中添加

```
def compute_logical_error_rate(model, test_loader, threshold=0.5):
    """计算逻辑错误率"""
    model.eval()
    correct = 0
    total = 0

    with torch.no_grad():
        for noisy, ideal, labels in test_loader:
            reconstructed = model(noisy.to(device))
            predicted_labels = classify_from_wigner(reconstructed)
            correct += (predicted_labels == labels.to(device)).sum().item()
            total += len(labels)

    logical_error_rate = 1 - correct / total
    return logical_error_rate
```

5.2 完整的QEC评估框架

建议添加以下指标体系：

层级	指标	意义
底层	重建MSE	态恢复的数值精度
中层	峰检测F1	格点结构的保持
顶层	逻辑错误率	量子纠错的最终性能

5.3 噪声扫描实验

绘制逻辑错误率 vs 物理噪声强度曲线：

```
noise_levels = [0.5, 1.0, 1.5, 2.0, 2.5, 3.0]
results = {model_name: [] for model_name in ['FNO', 'UNet', 'Wiener']}

for sigma in noise_levels:
    noise_config.diffusion_sigma = sigma
    test_loader = create_test_loader(noise_config)

    for name, model in models.items():
        p_logical = compute_logical_error_rate(model, test_loader)
        results[name].append(p_logical)

# 绘制 p_L vs σ 曲线
```

6. 总结

6.1 当前实验的定位

本实验主要解决的是 **GKP态重建** 问题，即从噪声测量中恢复理想的Wigner函数表示。这是一个信号处理/图像去噪任务，使用的MSE、PSNR、F1等指标是合适的。

6.2 与量子纠错的关系

方面	当前实现	完整QEC框架
输入	噪声Wigner函数	同左
输出	重建的Wigner函数	逻辑态 + 综合征
主要指标	MSE, F1	逻辑错误率
评估角度	信号重建质量	纠错能力

6.3 结论

- 当前指标的有效性：**MSE、F1等指标能够反映模型的去噪能力，与量子纠错性能正相关，但不直接等价。
- 物理/逻辑错误率的可行性：**可以在当前框架上添加逻辑错误率计算，这将更直接地衡量模型的量子纠错能力。
- 建议：**
 - 保留当前的重建质量指标
 - 添加逻辑错误率作为主要QEC性能指标
 - 进行噪声扫描实验，绘制阈值曲线

附录：符号表

符号	含义
α	GKP格点间距， $\approx \sqrt{2\pi}$
σ	噪声标准差
η	光子传输效率
n_{th}	热光子数
p	物理错误率
p_L	逻辑错误率
$W(x,p)$	Wigner函数