End-to-End Quantum Correlated Imaging Reconstruction with UNet

Yiwen Tang, Yufan Yao

May 2025

Project Introduction

- This project implements an end-to-end quantum correlated imaging (Ghost Imaging) reconstruction system based on UNet.
- Supports multi-frame signal/idler image stacking as multi-channel input, adapted to UNet.
- Flexible loss function combination (SSIM, MSE, perceptual loss, etc.), supports weighted loss.
- Automated experiment logging and hyperparameter search for easy comparison and reproducibility.

Data Structure and Preprocessing

Each sample directory:

- `object_dir/`
  - `signal/` % multi-frame signal images
  - `idler/` % multi-frame idler images
  - `target.JPG`

Multi-frame stacking and merging:

- Take the first max_signal/max_idler signal/idler images
- Every stack_num images are averaged to form one channel
- Final input channels:
  (max_signal//stack_num) + (max_idler//stack_num)

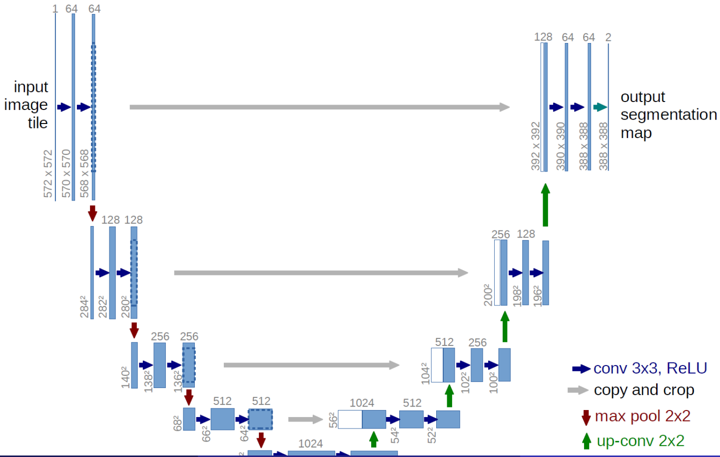
Data Flow and Input Example

Target Example

Prediction Example

Model Architecture – UNet

- Standard UNet structure, supports custom in_channels for multi-channel input
- Symmetric encoder-decoder with skip connections
- Output is single-channel reconstructed image



Loss Function and Training

Example loss function:

- `def loss_fn(output, target):`
- `    return w_ssim * (1 - ssim(output, target))`
- `      + w_mse * MSELoss(output, target)`
- `      + w_perc * perceptual_loss(output, target)`

- Supports SSIM, MSE, perceptual loss (VGG16), and weighted combination
- Automatically saves best model, loss/PSNR curves, etc.
- Each experiment is archived for comparison

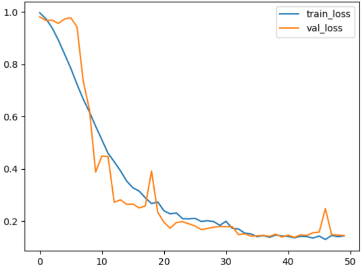
Hyperparameter Search and Experiment Logging

Optuna search example:

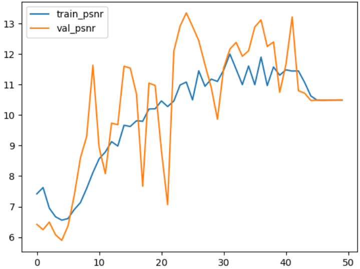
- `study = optuna.create_study(direction='minimize')`
- `study.optimize(objective, n_trials=20)`

- Supports Optuna automated hyperparameter search: stack_num, learning_rate, max_signal, loss weights, etc.
- All trial results and best params are logged
- Each run generates a unique experiment name, all results archived in results/exp_xxx/
- Each experiment saves:
  - Training/validation loss curves (losses.png)
  - PSNR curve (psnrs.png)
  - Main hyperparameters and metrics (config.json, metrics.json)
  - Best model weights (model.pth)
  - Typical predictions (pred_*.png, target_*.png)
- File names include main parameters (e.g. exp_20250531_003234_epochs50_stack2_lr0.00419_sig50)
- Can automatically extract latest experiment results for evaluation

Visualization and Example Outputs



Loss Curve



PSNR Curve (example)

- See previous page for prediction vs. target
- More examples in results/ directory

Main Parameters and Tuning

- Tune stack_num, max_signal, loss weights manually first, then use Optuna for fine-tuning
- PSNR/SSIM are for evaluation, not recommended as loss
- Code is modular for easy customization (loss, model, data, etc.)

PSNR Metric Explanation

PSNR (Peak Signal-to-Noise Ratio) is a common metric for image reconstruction quality, in dB.

- **Definition:** $\mathrm{PSNR} = 10 \log_{10}(\mathrm{MAX}^2/\mathrm{MSE})$
- **MAX** is pixel max (e.g. 1.0 or 255), MSE is mean squared error
- **Range:** Theoretical $[0, +\infty)$, practical $10 \sim 40$ dB
- **Typical intervals:**
  - < 20 dB: visible distortion
  - $20 \sim 30$ dB: acceptable
  - $30 \sim 40$ dB: high quality
  - > 40 dB: nearly perfect
- Higher PSNR means better reconstruction
- All PSNR in this project are for [0,1] normalized grayscale images

Inference Time Comparison

- **UNet-based end-to-end model:** 0.5 seconds per image (on GPU)
- **Traditional ghost imaging reconstruction:** 5–10 minutes per image (CPU, iterative algorithms)
- **Speedup:** 600x–1200x faster
- Deep learning enables real-time or near real-time quantum imaging, making practical applications feasible.

Method Comparison: PSNR and Inference Time

Method	PSNR (dB)	Time per Image
Direct Stacking	$\sim$15–20	$<$ 0.1 s (GPU/CPU)
UNet (Ours)	$\sim$28–35	$\sim$0.5 s (GPU)
Traditional Ghost Imaging	$\sim$20–28	5–10 min (CPU)

Table: Comparison of three methods on PSNR and inference/compute time.

- UNet achieves the best quality and is orders of magnitude faster than traditional algorithms.
- Direct stacking is fast but with poor quality; traditional methods are slow and moderate in quality.

Conclusion

- End-to-end quantum correlated imaging reconstruction, auto experiment logging and hyperparameter search implemented
- Welcome to discuss and contribute, see README.md