同济大学大学生创新训练项目

季度报告

一、项目基本信息

项目名称	三维图像传感器信号增强网络的轻量化方法		
项目编号	X2024492	项目级别	国家级
起止时间（年月）	2024 年 3 月 至 2025 年 3 月		
项目负责人	林继申	所在院系	计算机科学与技术学院
学号	2250758	专业	软件工程
手机号	15143305542	邮箱	2250758@tongji.edu.cn
指导老师	曾进	所在院系	计算机科学与技术学院

二、季度报告内容

1）项目进展情况

☑ **按计划进行** ☐ **进度提前** ☐ **进度滞后**

2）项目主要研究

序号	研究阶段	研究内容	完成情况
1	第一季度	文献调研	已完成
2	第二季度	分析现有方法，记录精度及复杂度	已完成
3	第三季度	学习 Burst Denoising with Kernel Prediction Networks 方法，配置边缘设备硬件	已完成
4	第四季度	完成边缘设备软件和开发环境配置，在仿真数据集上训练，在真实数据集上测试，进行结果对比分析	已完成

3）项目研究成果

序号	季度报告成果名称	成果形式
1	完成边缘设备软件和开发环境配置，在仿真数据集上训练，在真实数据集上测试，进行结果对比分析	分析报告

4）项目季度报告

（一）边缘设备（NVIDIA Jetson Orin NX）环境配置过程

NVIDIA Orin NX 是一款高性能的边缘计算设备，广泛应用于机器人、自动驾驶和 AI 推理等领域。以下是配置 Orin NX 环境的步骤：

1. 硬件准备

Orin NX 模块：确保已安装 Orin NX 模块。

载板：如 Jetson Xavier NX 载板，确保兼容。

电源：使用官方推荐的电源适配器。

存储：安装 SSD 或 microSD 卡作为存储介质。

外设：连接显示器、键盘、鼠标等。

2. 安装操作系统

下载镜像：从 NVIDIA 官网下载适用于 Orin NX 的 JetPack SDK。

烧录镜像：使用工具如 balenaEtcher 将镜像烧录到 microSD 卡或 SSD。

启动设备：插入存储设备，启动 Orin NX，按照提示完成 Ubuntu 系统安装。

3. 安装 JetPack SDK

更新系统：

```
sudo apt update
sudo apt upgrade
```

安装 JetPack：

```
sudo apt install nvidia-jetpack
```

这将安装 CUDA、cuDNN、TensorRT 等必要组件。

4. 配置开发环境

设置环境变量：

在~/.bashrc 中添加：

export CUDA_HOME=/usr/local/cuda

export PATH=$CUDA_HOME/bin:$PATH

export LD_LIBRARY_PATH=$CUDA_HOME/lib64:$LD_LIBRARY_PATH

然后执行：

source ~/.bashrc

验证安装：

CUDA：

nvcc --version

cuDNN：

cat /usr/include/cudnn_version.h | grep CUDNN_MAJOR -A 2

TensorRT：

dpkg -l | grep tensorrt

5. 安装深度学习框架

PyTorch：

pip3 install torch torchvision torchaudio --extra-index-url https://download.pytorch.org/whl/cu113

TensorFlow：

pip3 install tensorflow

6. 测试示例

运行 CUDA 示例：

cd /usr/local/cuda/samples/1_Utilities/deviceQuery

sudo make

./deviceQuery

运行深度学习示例：

使用 PyTorch 或 TensorFlow 运行简单的模型训练或推理任务。

7. 其他配置

远程访问：配置 SSH 以便远程访问。

Docker：安装 Docker 以便容器化部署：

```
sudo apt install docker.io
sudo usermod -aG docker $USER
```

8. 性能优化

电源模式：设置为最大性能模式：

```
sudo nvpmodel -m 0
```

风扇控制：根据需要调整风扇速度：

```
sudo jetson_clocks
```

完成以上步骤后，Orin NX 设备配置完毕，可用于开发和部署 AI 应用。

（二）轻量化算法

1. 动态卷积核

动态卷积核（Dynamic Convolution）是一种通过动态生成卷积核来适应不同输入特征的方法。传统的卷积核是固定的，而动态卷积核可以根据输入的特征图动态调整卷积核的权重，从而更好地捕捉输入数据的局部特征。

优点：

适应性更强：动态卷积核可以根据输入数据的不同部分自适应地调整卷积核的权重，从而更好地捕捉局部特征。

减少参数量：通过动态生成卷积核，可以减少固定卷积核的数量，从而降低模型的参数量。

提高模型表达能力：动态卷积核可以增加模型的表达能力，使其能够处理更复杂的任务。

缺点：

计算复杂度增加：动态生成卷积核的过程会增加额外的计算开销，尤其是在生成卷积核的过程中需要进行复杂的操作（如矩阵乘法）。

训练难度增加：动态卷积核的引入可能会增加模型的训练难度，尤其是在生成卷积核的过程中需要额外的优化步骤。

在代码中，LocalConv2d_No 类实现了动态卷积核的功能。它通过 w_gen 动态生成卷积核的权重，并根据输入的特征图进行卷积操作。这种方法可以在一定程度上减少模型的参数量，但也会增加计算复杂度。

2. 随机化剪枝

随机化剪枝（Randomized Pruning）是一种通过随机删除神经网络中的部分权重或神经元来减少模型复杂度的方法。剪枝可以分为结构化剪枝和非结构化剪枝。结构化剪枝通常删除整个卷积核或通道，而非结构化剪枝则是随机删除单个权重。

优点：

减少模型参数量：通过剪枝可以显著减少模型的参数量，从而降低模型的存储和计算需求。

加速推理：剪枝后的模型在推理时可以减少计算量，从而加速推理过程。

防止过拟合：剪枝可以作为一种正则化手段，防止模型过拟合。

缺点：

性能损失：剪枝可能会导致模型性能的下降，尤其是在剪枝比例较高时。

训练难度增加：剪枝后的模型需要重新训练或微调，以恢复部分性能。

随机性影响：随机化剪枝可能会导致模型性能的不稳定性，尤其是在不同的剪枝策略下。

3. 超参数调整

超参数调整（Hyperparameter Tuning）是指通过调整模型的超参数（如学习率、批量大小、卷积核大小、网络深度等）来优化模型的性能和效率。超参数调整可以通过手动调参、网格搜索、随机搜索或贝叶斯优化等方法进行。

优点：

优化模型性能：通过调整超参数，可以找到最优的模型配置，从而提高模型的性能。

提高模型效率：通过调整超参数（如批量大小、学习率等），可以加速模型的训练过程，并减少资源消耗。

灵活性高：超参数调整可以根据具体的任务需求进行灵活配置，适用于不同的应用场景。

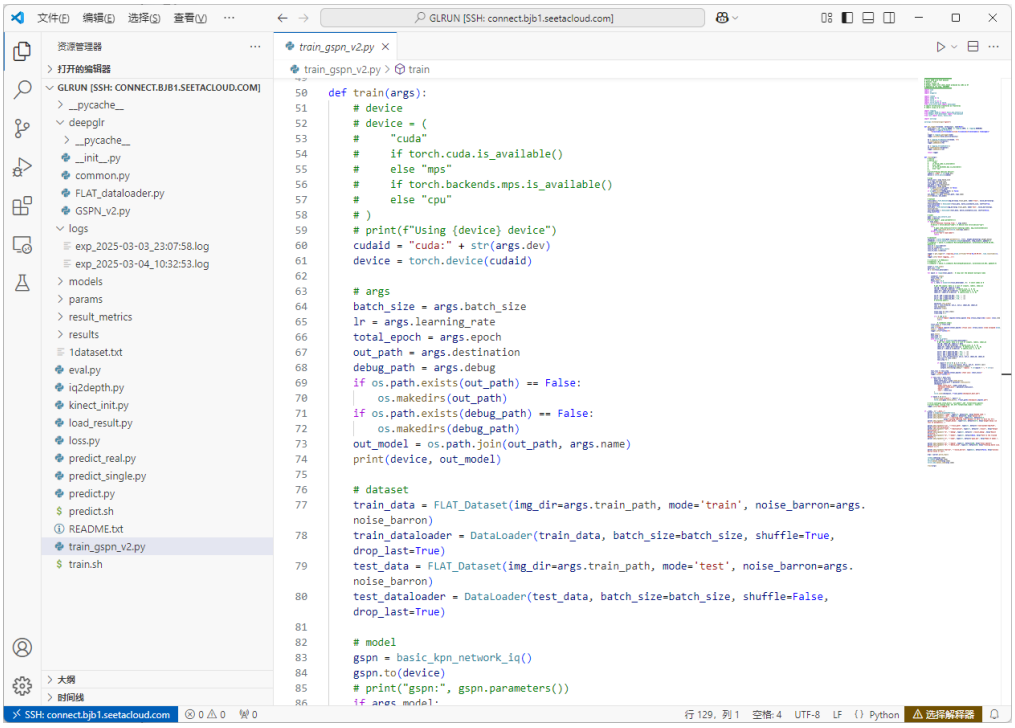
缺点：

耗时：超参数调整通常需要大量的实验和计算资源，尤其是在使用网格搜索或贝叶斯优化时。

局部最优：超参数调整可能会陷入局部最优，尤其是在搜索空间较大时。

依赖经验：超参数调整的效果很大程度上依赖于调参者的经验和对模型的理解。

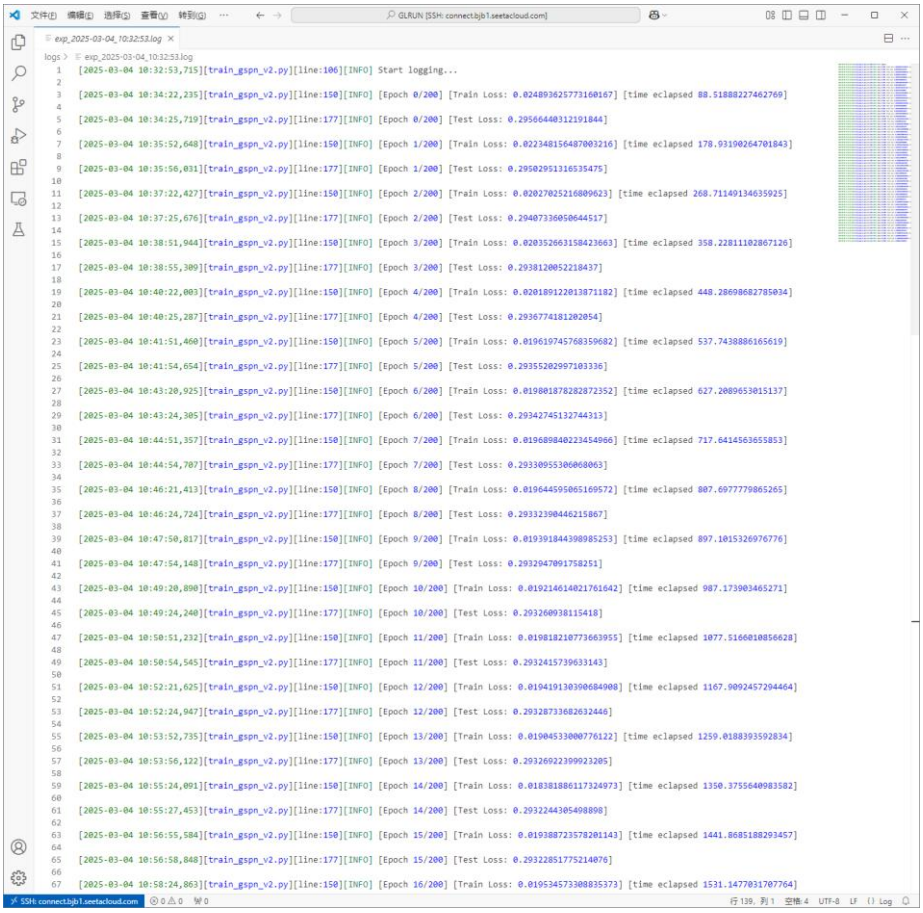
（三）在仿真数据集上训练，在真实数据集上测试



训练代码

描述	值
随机种子	42
使用的 GPU 设备 ID	0
学习率	1.00E-04
权重衰减（L2 正则化）	1.00E-05
训练的总轮数	200
训练的批量大小	10
优化器	Adam
学习率调度器	StepLR（每 15 个 epoch 乘以 0.7）
训练损失函数	GLoss
测试损失函数	GLoss_test

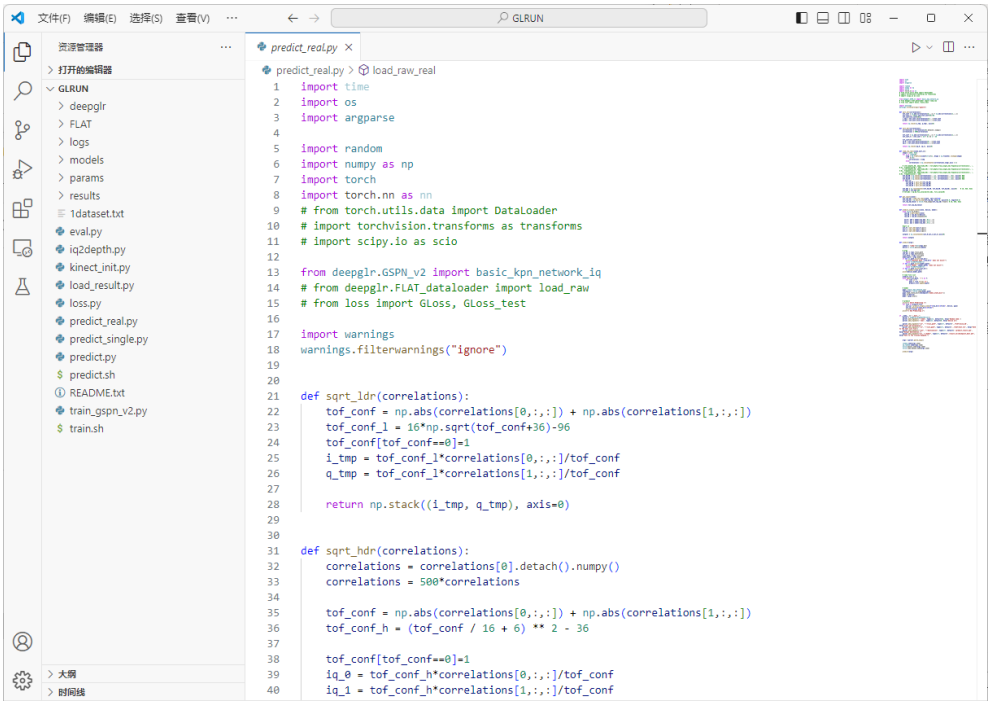
超参数



```
[2025-03-04 10:32:53,715][train_gspn_v2.py][line:106][INFO] Start logging...

[2025-03-04 10:34:22,235][train_gspn_v2.py][line:150][INFO] [Epoch 0/200] [Train Loss: 0.024893625773160167] [time eclapsed 88.51888227462769]

[2025-03-04 10:34:25,719][train_gspn_v2.py][line:177][INFO] [Epoch 0/200] [Test Loss: 0.29566440312191844]

[2025-03-04 10:35:52,648][train_gspn_v2.py][line:150][INFO] [Epoch 1/200] [Train Loss: 0.022348156487003216] [time eclapsed 178.93190264701843]

[2025-03-04 10:35:56,031][train_gspn_v2.py][line:177][INFO] [Epoch 1/200] [Test Loss: 0.29502951316535475]

[2025-03-04 10:37:22,427][train_gspn_v2.py][line:150][INFO] [Epoch 2/200] [Train Loss: 0.02027025216809623] [time eclapsed 268.71149134635925]

[2025-03-04 10:37:25,676][train_gspn_v2.py][line:177][INFO] [Epoch 2/200] [Test Loss: 0.29407336050644517]

[2025-03-04 10:38:51,944][train_gspn_v2.py][line:150][INFO] [Epoch 3/200] [Train Loss: 0.020352663158423663] [time eclapsed 358.22811102867126]

[2025-03-04 10:38:55,309][train_gspn_v2.py][line:177][INFO] [Epoch 3/200] [Test Loss: 0.2938120052218437]

[2025-03-04 10:40:22,003][train_gspn_v2.py][line:150][INFO] [Epoch 4/200] [Train Loss: 0.020189122013871182] [time eclapsed 448.28698682785034]

[2025-03-04 10:40:25,287][train_gspn_v2.py][line:177][INFO] [Epoch 4/200] [Test Loss: 0.2936774181202054]

[2025-03-04 10:41:51,460][train_gspn_v2.py][line:150][INFO] [Epoch 5/200] [Train Loss: 0.019619745768359682] [time eclapsed 537.7438886165619]

[2025-03-04 10:41:54,654][train_gspn_v2.py][line:177][INFO] [Epoch 5/200] [Test Loss: 0.29355202997103336]

[2025-03-04 10:43:20,925][train_gspn_v2.py][line:150][INFO] [Epoch 6/200] [Train Loss: 0.019801878282872352] [time eclapsed 627.2089653015137]

[2025-03-04 10:43:24,305][train_gspn_v2.py][line:177][INFO] [Epoch 6/200] [Test Loss: 0.29342745132744313]

[2025-03-04 10:44:51,357][train_gspn_v2.py][line:150][INFO] [Epoch 7/200] [Train Loss: 0.019689840223454966] [time eclapsed 717.6414563655853]

[2025-03-04 10:44:54,707][train_gspn_v2.py][line:177][INFO] [Epoch 7/200] [Test Loss: 0.29330955306068063]

[2025-03-04 10:46:21,413][train_gspn_v2.py][line:150][INFO] [Epoch 8/200] [Train Loss: 0.019644595065169572] [time eclapsed 807.6977779865265]

[2025-03-04 10:46:24,724][train_gspn_v2.py][line:177][INFO] [Epoch 8/200] [Test Loss: 0.29332390446215867]

[2025-03-04 10:47:50,817][train_gspn_v2.py][line:150][INFO] [Epoch 9/200] [Train Loss: 0.019391844398985253] [time eclapsed 897.1015326976776]

[2025-03-04 10:47:54,148][train_gspn_v2.py][line:177][INFO] [Epoch 9/200] [Test Loss: 0.2932947091758251]

[2025-03-04 10:49:20,890][train_gspn_v2.py][line:150][INFO] [Epoch 10/200] [Train Loss: 0.019214614021761642] [time eclapsed 987.173903465271]

[2025-03-04 10:49:24,240][train_gspn_v2.py][line:177][INFO] [Epoch 10/200] [Test Loss: 0.293260938115418]

[2025-03-04 10:50:51,232][train_gspn_v2.py][line:150][INFO] [Epoch 11/200] [Train Loss: 0.019818210773663955] [time eclapsed 1077.5166010856628]

[2025-03-04 10:50:54,545][train_gspn_v2.py][line:177][INFO] [Epoch 11/200] [Test Loss: 0.2932415739633143]

[2025-03-04 10:52:21,625][train_gspn_v2.py][line:150][INFO] [Epoch 12/200] [Train Loss: 0.019419130390684908] [time eclapsed 1167.9092457294464]

[2025-03-04 10:52:24,947][train_gspn_v2.py][line:177][INFO] [Epoch 12/200] [Test Loss: 0.2932873368632644]

[2025-03-04 10:53:52,735][train_gspn_v2.py][line:150][INFO] [Epoch 13/200] [Train Loss: 0.019045330007761 22] [time eclapsed 1259.0188393592834]

[2025-03-04 10:53:56,122][train_gspn_v2.py][line:177][INFO] [Epoch 13/200] [Test Loss: 0.2932692239992320 5]

[2025-03-04 10:55:24,091][train_gspn_v2.py][line:150][INFO] [Epoch 14/200] [Train Loss: 0.018381886117324973] [time eclapsed 1350.3755640983582]

[2025-03-04 10:55:27,453][train_gspn_v2.py][line:177][INFO] [Epoch 14/200] [Test Loss: 0.2932244305409898]

[2025-03-04 10:56:55,584][train_gspn_v2.py][line:150][INFO] [Epoch 15/200] [Train Loss: 0.019388723578201143] [time eclapsed 1441.8685188293457]

[2025-03-04 10:56:58,848][train_gspn_v2.py][line:177][INFO] [Epoch 15/200] [Test Loss: 0.2932285177521407 6]

[2025-03-04 10:58:24,863][train_gspn_v2.py][line:150][INFO] [Epoch 16/200] [Train Loss: 0.019534573308835373] [time eclapsed 1531.1477031707764]
```

训练日志

（四）结果对比分析



```python
import time
import os
import argparse

import random
import numpy as np
import torch
import torch.nn as nn
# from torch.utils.data import DataLoader
# import torchvision.transforms as transforms
# import scipy.io as scio

from deepglr.GSPN_v2 import basic_kpn_network_iq
# from deepglr.FLAT_dataloader import load_raw
# from loss import GLoss, GLoss_test

import warnings
warnings.filterwarnings("ignore")


def sqrt_ldr(correlations):
    tof_conf = np.abs(correlations[0,:,:]) + np.abs(correlations[1,:,:])
    tof_conf_l = 16*np.sqrt(tof_conf+36)-96
    tof_conf[tof_conf==0]=1
    i_tmp = tof_conf_l*correlations[0,:,:]/tof_conf
    q_tmp = tof_conf_l*correlations[1,:,:]/tof_conf

    return np.stack((i_tmp, q_tmp), axis=0)


def sqrt_hdr(correlations):
    correlations = correlations[0].detach().numpy()
    correlations = 500*correlations

    tof_conf = np.abs(correlations[0,:,:]) + np.abs(correlations[1,:,:])
    tof_conf_h = (tof_conf / 16 + 6) ** 2 - 36

    tof_conf[tof_conf==0]=1
    iq_0 = tof_conf_h*correlations[0,:,:]/tof_conf
    iq_1 = tof_conf_h*correlations[1,:,:]/tof_conf
```

预测代码

Baseline：

```
[ Start Predicting ]
Elapsed time : 0.9651293754577637 sec, Average processing time : 0.008936383106090405 sec
```

```
rmse_mean, mae_mean, irmse_mean, imae_mean: 0.0741 0.0135 0.0448 0.0116
rel mean, del 1 mean, del 2 mean, del 3 mean: 0.0327 0.9984 0.9989 0.9990
```

轻量化：

```
[ Start Predicting ]
Elapsed time : 1.0434322357177734 sec, Average processing time : 0.009661409589979384 sec
```

```
rmse_mean, mae_mean, irmse_mean, imae_mean: 0.0695 0.0108 0.0291 0.0086
rel mean, del 1 mean, del 2 mean, del 3 mean: 0.0209 0.9990 0.9992 0.9992
```

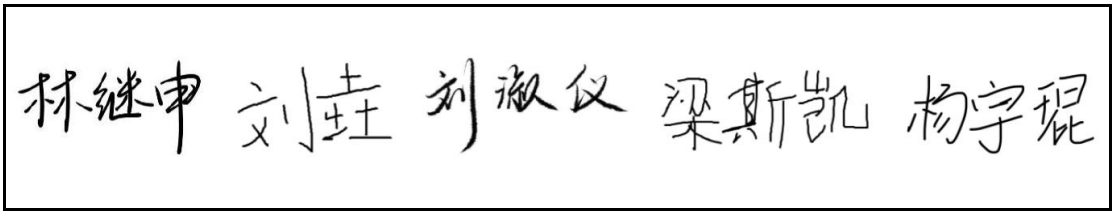
5）经费开支情况

名目	金额（元）	用途	备注
1. 业务费	0	无	无
（1）计算、分析、测试费	0	无	无
（2）能源动力费	0	无	无
（3）会议、差旅费	0	无	无
（4）文献检索费	0	无	无
（5）论文出版费	0	无	无
2. 仪器设备购置费	0	无	无
3. 实验装置试制费	0	无	无
4. 材料费	0	无	无

6）项目后期具体工作计划

a) 撰写结题报告

三、项目组成员签名



四、指导老师意见

导师签字： 年　　月　　日

五、院系意见

教学负责人（签章）： 年　　月　　日

六、学校大学生创新创业训练计划专家组意见

负责人（签章）： 年　　月　　日