集成电路 EDA 设计精英挑战赛
INTEGRATED CIRCUIT EDA ELITE CHALLENGE

共圆芯梦想 2023(第五届)

集成电路EDA设计精英挑战赛总决赛

2023年12月22日-24日　南京江北新区

Team Number:

****

Team Name:

****

Contents

Part I: Problem Formulation

集成电路 EDA 设计精英挑战赛
INTEGRATED CIRCUIT EDA ELITE CHALLENGE

Introduction

Traditional Method of prediction IR drop → **Insert netlist data**, including topographical structures and parameters → **Massive matrix calculation**, usually take hundreds of hours → **Precise voltage value**

Our Method **YogurtNet** → **Insert netlist data**, same as listed above → Applying **AI image processing methods** → **Predicted voltage value**

Compare with CircuitNet Demo

CircuitNet Demo

- Rough data preprocessing
- Simplistic model structure, low precision of prediction

YogurtNet

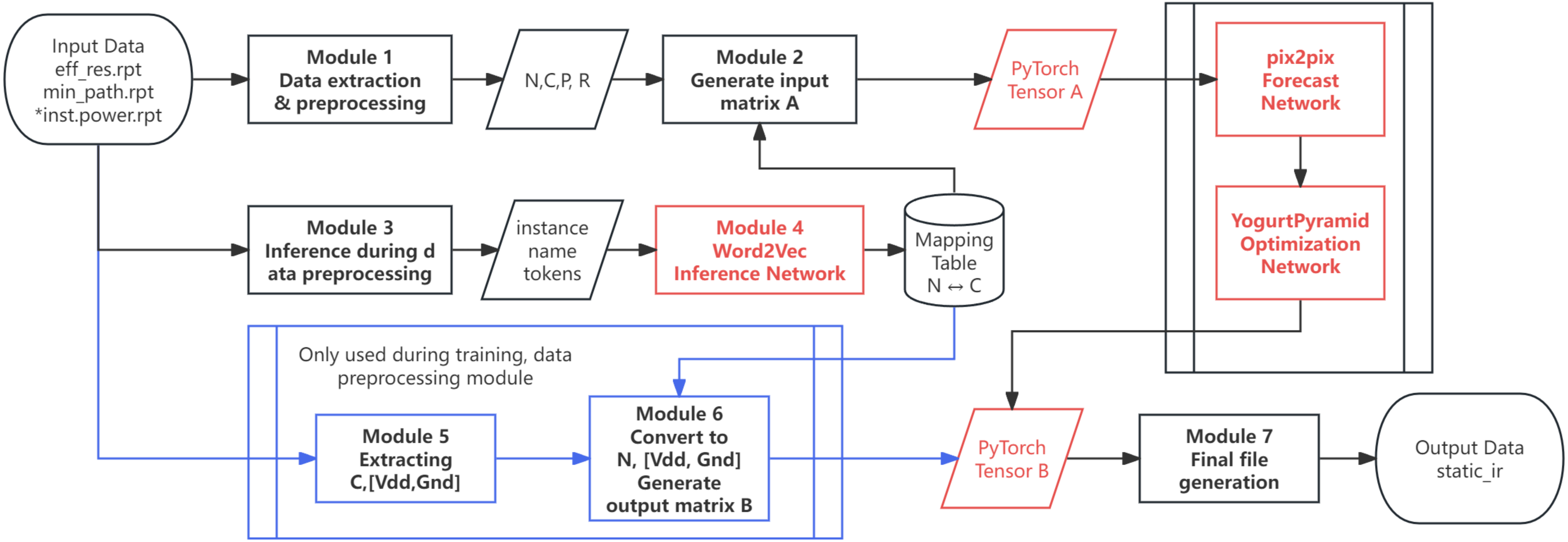
- **Data classification before input: N, C, P, R groups**
- **Multi-model application & Labor division**
  3 models: Clustering, Forecast, Optimization

Part I: Problem Formulation

集成电路 EDA 设计精英挑战赛
INTEGRATED CIRCUIT EDA ELITE CHALLENGE

Overiew of YogurtNet

YogurtNet



Part II: Algorithms & Frameworks

集成电路EDA设计精英挑战赛
INTEGRATED CIRCUIT EDA ELITE CHALLENGE

Overiew of Algorithms & Frameworks

1. Data Preprocessing

2. Word2Vec Clustering

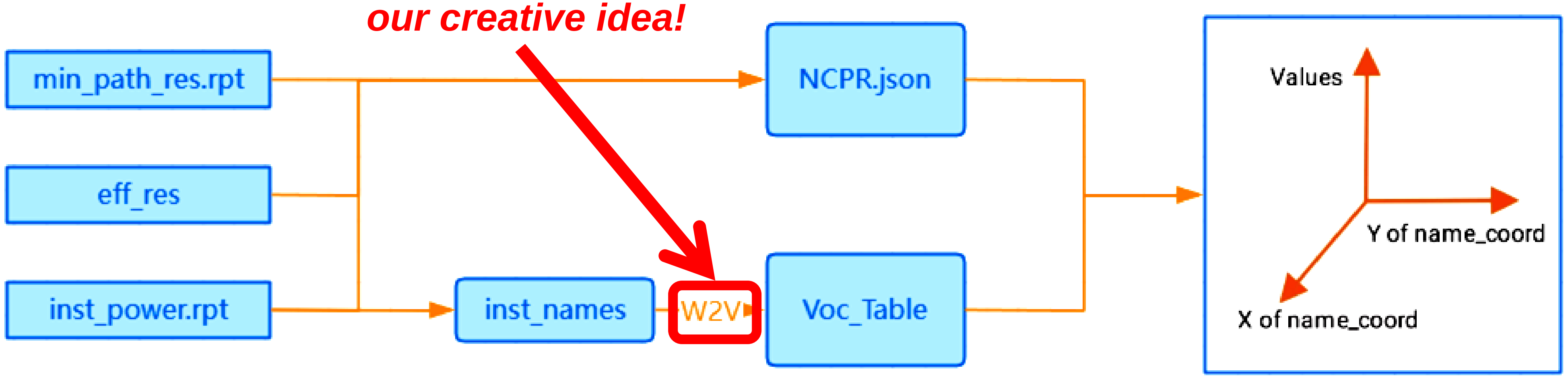
3. Post Data Processing & Program Acceleration

4. Pix2Pix Forecasting

5. YogurtPyramid Optimizing

1. Data Preprocessing

Word2Vec utilization, our creative idea!



Workflow of data preprocessing

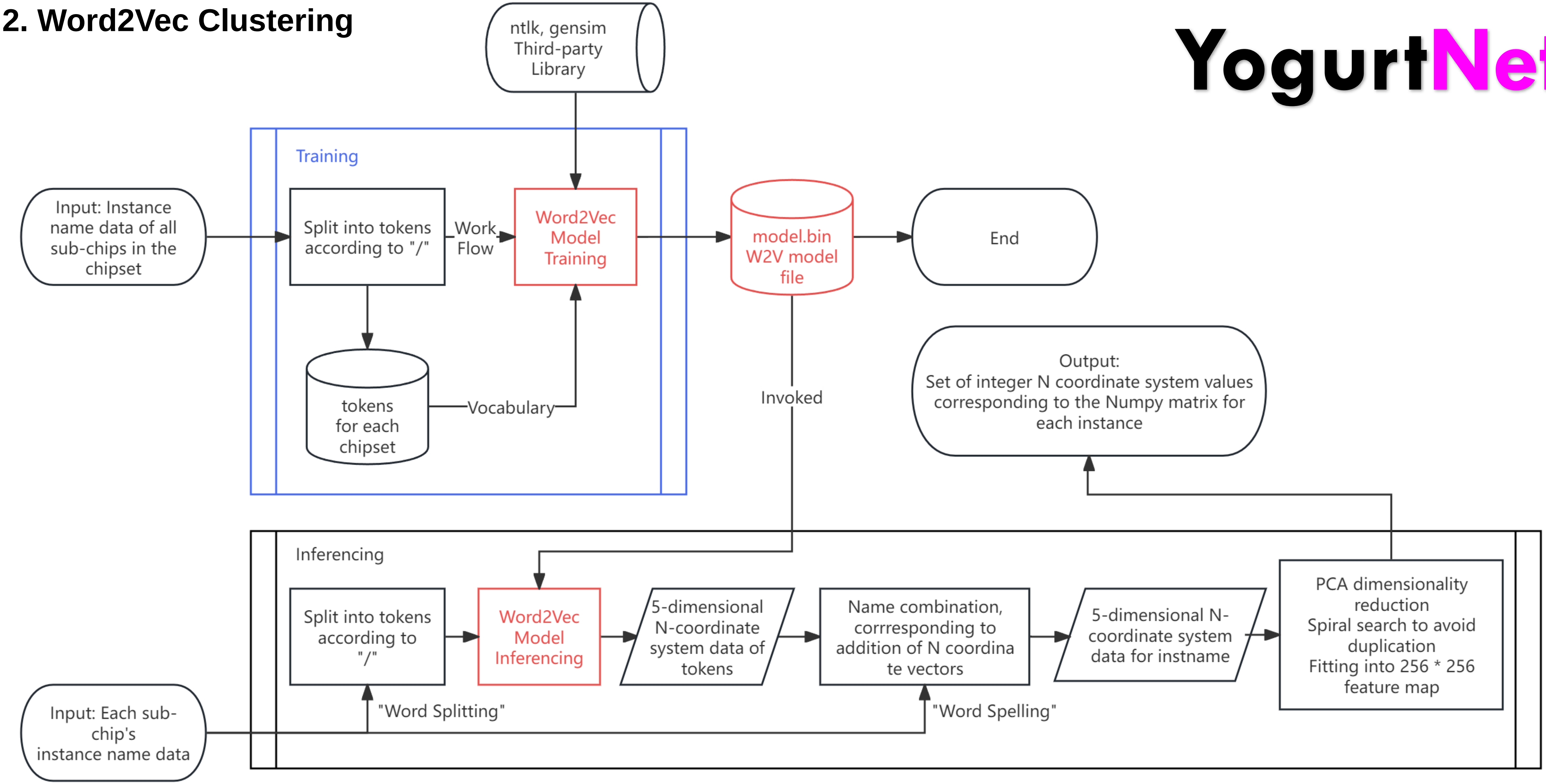

```
"peripherals_i/apb_i2c_i/byte_controller/sr_reg_-5-_": [2, 55],
"core_region_i/CORE.RISCV_CORE/id_stage_i/registers_i/U1381": [5, 55],
"core_region_i/adv_dbg_if_i/dbg_module_i/i_dbg_cpu_or1k/U247": [3, 55],
"peripherals_i/apb_spi_master_i/u_rxfifo/buffer_reg_-7-__-3-_": [2, 54],
"core_region_i/instr_mem/boot_rom_wrap_i/boot_code_i/U2289": [3, 53],
```


2-Dimensional Mapping Result
of Each Instance

Part II: Algorithms & Frameworks

2. Word2Vec Clustering

YogurtNet



2. Word2Vec Clustering

Word2Vec: Unsupervised Learning model,
typically used for natural language processing (NLP)

Applying to YogurtNet: **Data Clustering**
Instance Names → mapping → **“Name Coordinate”** N(x,y) → Feature Map

Why?

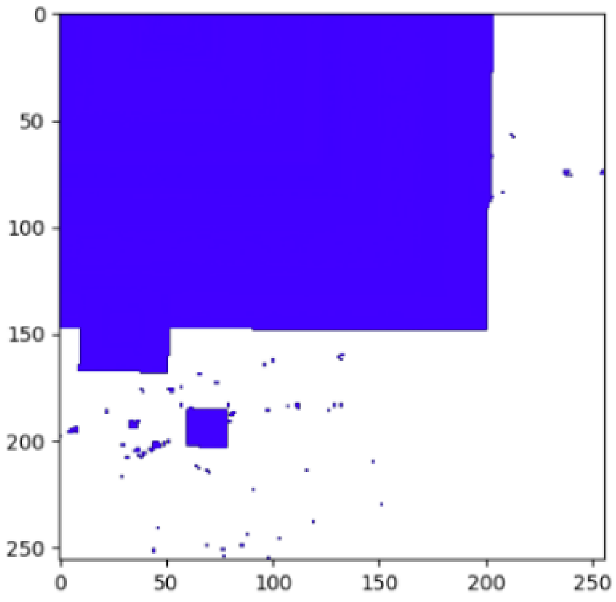
1. Assumption: there must be some **connections** between **instance names & physical layout coordinates.**

2. Avoid Detail Loss: **every single pixel** on feature map represents **every instance** preserving **instance-based information** √
using physical coordinates as feature maps, **image compression causes detail loss** ×

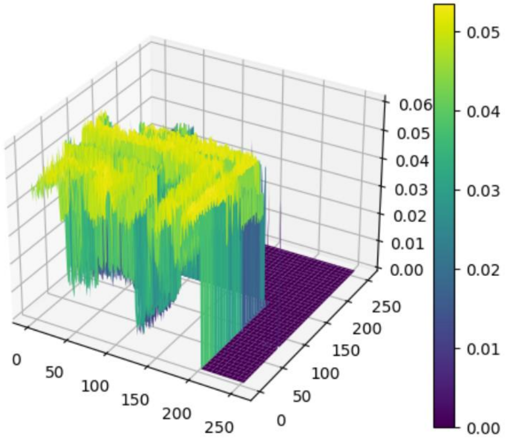
3. Proofs of Clustering Result:

a. Effective Clusterization of instance names into a **regular rectangular space**
b. Data **distribution** after clustering exhibits **strong regularity**

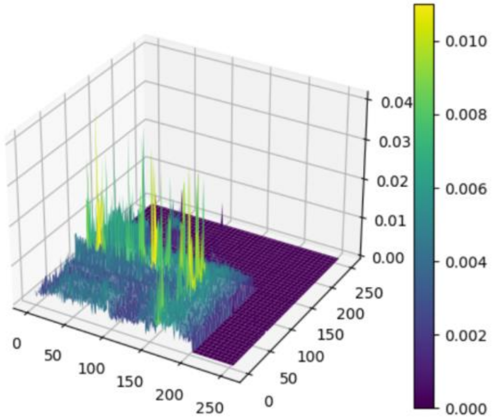
2. Word2Vec Clustering



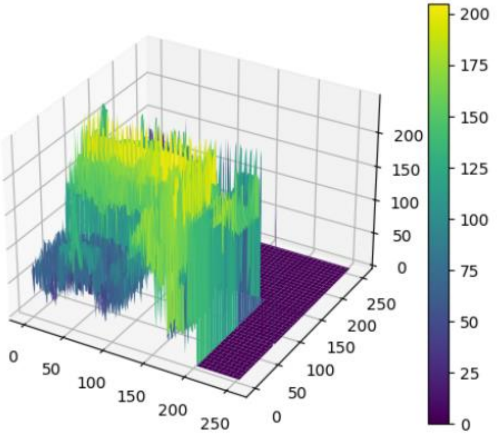
Distribution of Instances based on “Name Coordinate” N(x,y)



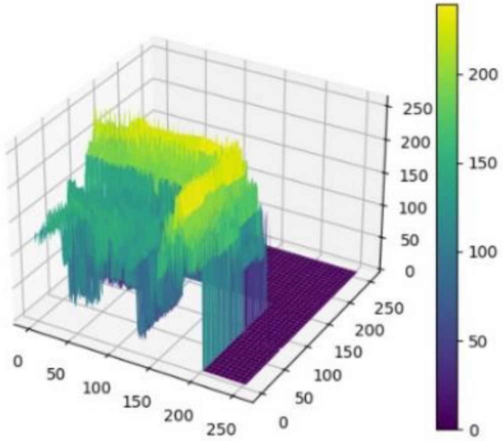
VDD_drop vs N(x,y)



GND_bounce vs N(x,y)



Physical X vs N(x,y)

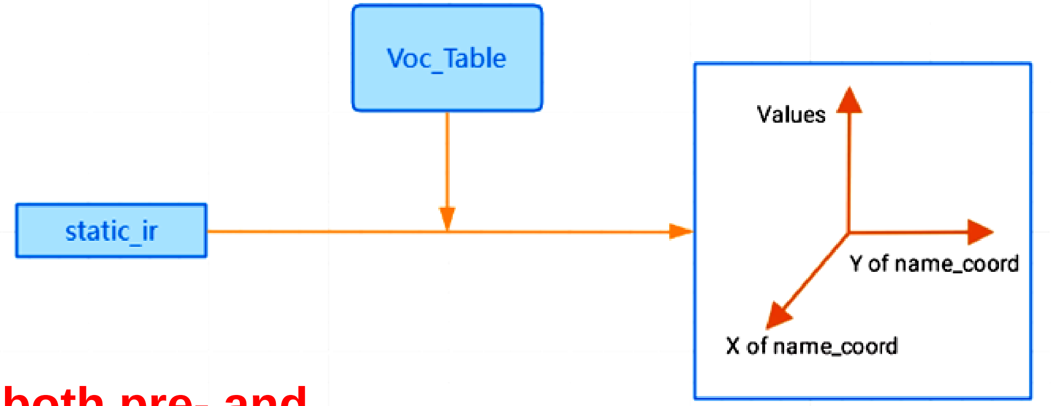


Physical Y vs N(x,y)

Data Distribution with Strong Regularity

Part II: Algorithms & Frameworks

3. Post Data Processing



Workflow of
post data processing

Program Acceleration for both pre- and post data processing

1.

```python
def parallel_run_steps1():

    with ThreadPoolExecutor() as executor:

        future1 = executor.submit(Export_NCPR.preprocess_main, fpath, inst_names_list, bbox_list, P_list)
        future2 = executor.submit(W2V_infer.W2V_infer_main, fpath, inst_names_list, size)
```

- **Multithread Implementation** with **ThreadPool** library
- Speed up **overall program execution**
- By **concurrently processing** logically independent programs

2.

```python
if __name__ == "__main__":
    process_num = 30
    main()
```

- **Program for multi-processing**
- Enabling use of multiple independent processes → perform IR drop inference on multiple chips → **speedup**

4. Pix2Pix Forecasting

Dataloader: from raw data to feature map

1. Channel Reduction

- Original channels: 6×C, 6×P, 5×R
- Criterion: **Average** & **Variance**
- Method: **K-Means**
- **17 channels → 3 channels**

2. Normalization by min-max Streching

- Minimum value: 0.0
- Maximum value: **97.5%** of all maximum values within the entire design
- range: **from -1 to 1**

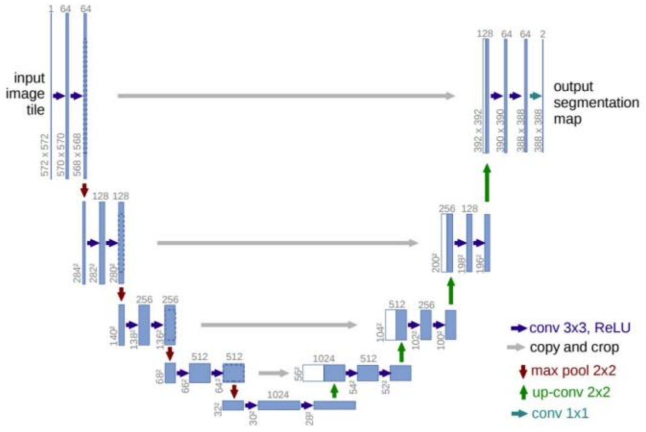
3. Filling, Filtering and Edge Process

- **Filling**: empty points, improve model's **fitting ability**
- **Filtering: balancing** burrs and regular distribution
- **Edge process**: reduce prediction errors in edge regions

Part II: Algorithms & Frameworks

4. Pix2Pix Forecasting

input feature map:
256 × 256 × 3



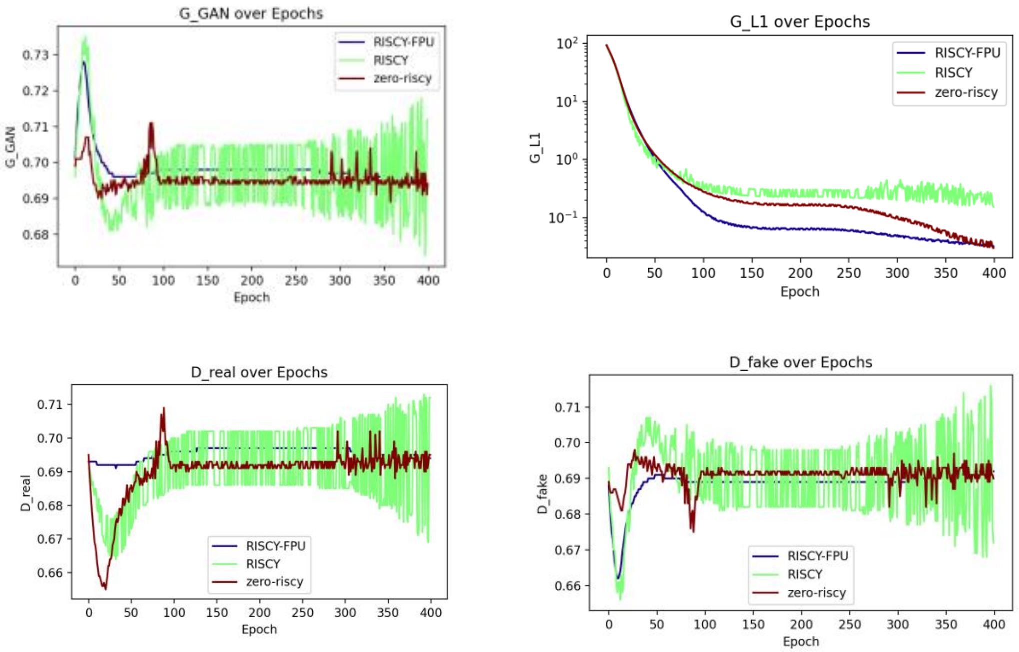
output feature map:
256 × 256 × 2

pix2pix： U-Net backbone
GAN training method

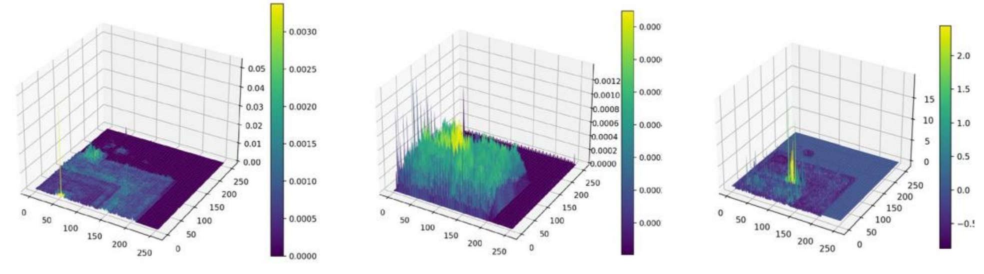
for chips with larger instance numbers:
feature map = *batchsize × 128×128×3*

Model Training Results:

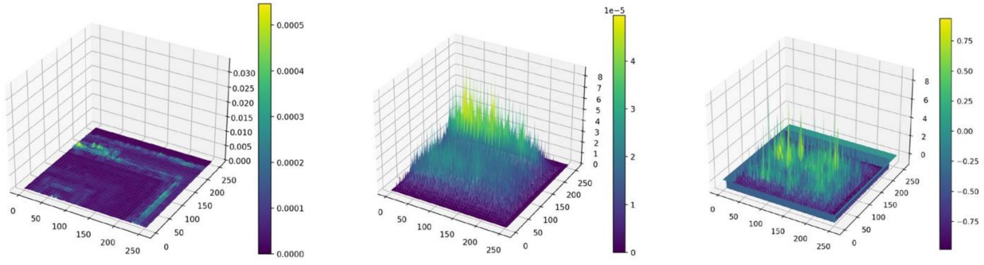
G_GAN, G_L1, D_real, D_fake metrics



Pix2Pix forecasting example results

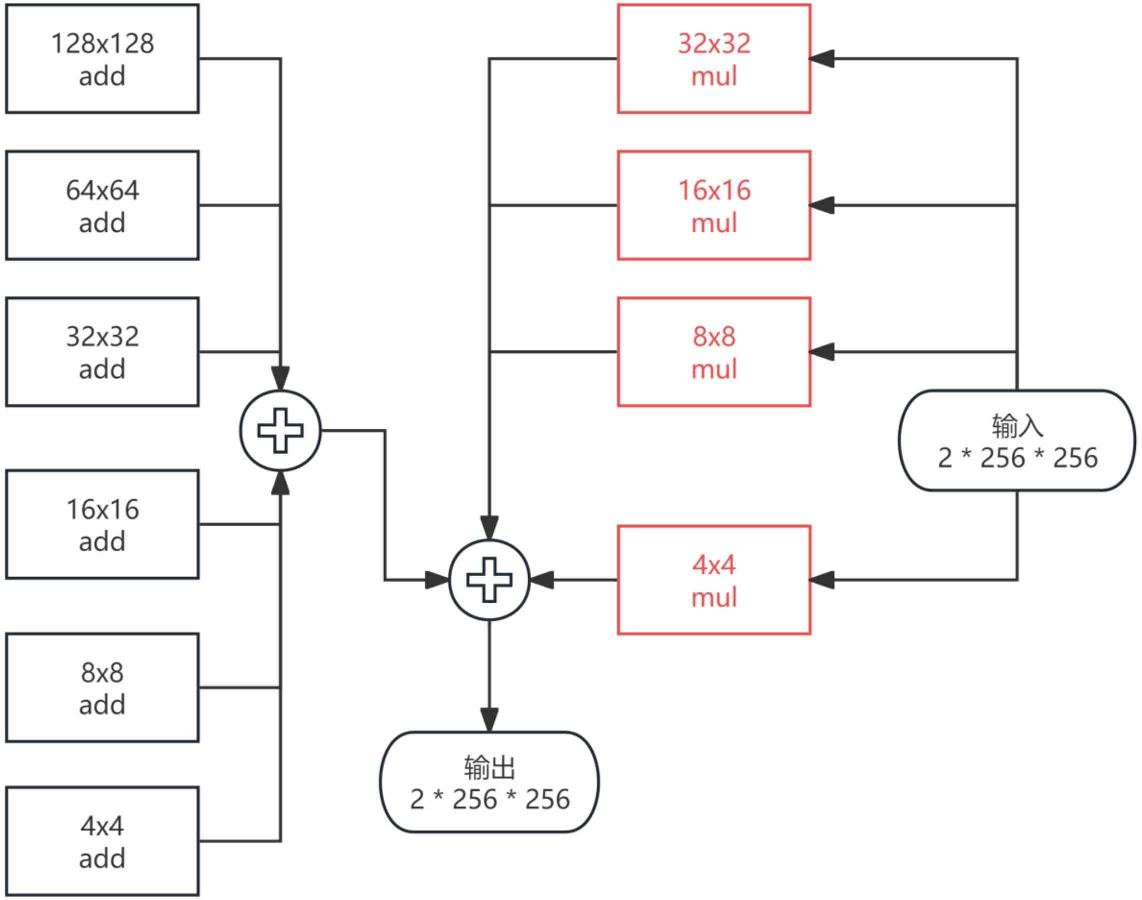


Zero-riscy 目标事实　Zero-riscy 预测结果　Zero-riscy 偏差比例



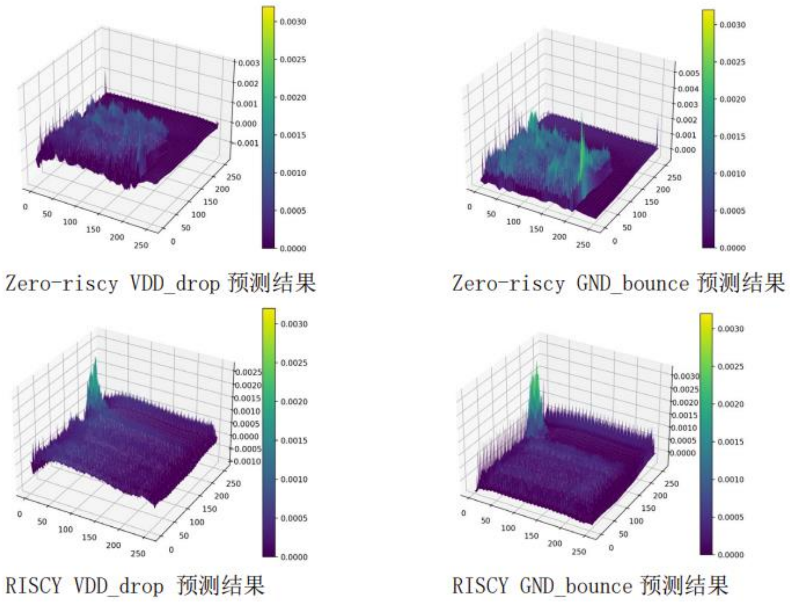
RISCY-FPU 目标事实　RISC-FPU 预测结果　RISC-FPU 偏差比例

5. YogurtPyramid Optimizing



Basic Structure of YogurtPyramid

YogurtPyramid optimization example results:



L1 loss metrics:

before	after
L1 loss: 0.074649	L1 loss: 0.0297581
进程已结束,退出代码0	进程已结束,退出代码0

though it's not a mature network, but we've proved the feasibility of hybrid network structures.

Part III: Experimental results

集成电路 EDA 设计精英挑战赛
INTEGRATED CIRCUIT EDA ELITE CHALLENGE

Testing Configuration Info

CPU:
Intel Xeon Platinum 8368 @ 2.40GHz × 152 cores
GPU:
NVIDIA A100, with 80GB memory
× 2 processors
Operating system:
Ubuntu 20.04.3 LTS, Linux kervel ver 5.4.0
Software configuration:
Python 3.8.10, installed on Anaconda 2022.10
PyTorch ver 2.0.1, CUDA 11.8, CUDNN 8.7.10
Third-party library:
gensim 4.3.0, ntlk 3.8.1

Analysis:
most **time-consuming** part:
Word2Vec inferencing, specifically
PCA dimensionality reduction

Source:
1. zero-riscy_freq_50_mp_1_fpu_55_fpa_1.5_p_6_fi_ap_
2. zero-riscy_freq_50_mp_1_fpu_60_fpa_2.0_p_4_fi_ar_
3. zero-riscy_freq_50_mp_1_fpu_65_fpa_1.0_p_3_fi_ar_

Test Results

Object	Wall time
Data Preprocessing (Word2Vec inferencing included)	**191.65s**
Pix2Pix Inferencing	10.79s
YogurtPyramid inferencing & post data generation	5.71s
Total (all 3 chips)	209.82s
Average time (single chip)	**69.94s**

CC Metric (technical score)	**0.36891373**

Object	MAE metrics
zero-riscy_freq_50_mp_1_fpu_55_fpa_1.5_p_6_fi_ap_	3.2457
zero-riscy_freq_50_mp_1_fpu_60_fpa_2.0_p_4_fi_ar_	9.1438
zero-riscy_freq_50_mp_1_fpu_65_fpa_1.0_p_3_fi_ar_	3.4459
Total MAE (all 3 chips)	15.8354
Average MAE (single chip)	**5.2784**

Part IV: Conclusion & Discussion

集成电路EDA设计精英挑战赛
INTEGRATED CIRCUIT EDA ELITE CHALLENGE

Advantages

1. Basic improvements of **prediction accuracy**
2. **Innovation** of utilizing a NLP model -- **Word2Vec** for IR drop prediction
3. Enhance **generalization ability** by applying multiple network models for specific purposes

Potential Improvements

1. Poor **time-consuming** performance, **bottleneck: PCA dimensional reduction** → by designing and applying more effective methods
2. **Word2Vec model**: require **finer parameter tuning** → **better adaptation**
3. Replace existing YogurtPyramid with **Diffusion CNN model** for better fitting performance.

集成电路EDA设计精英挑战赛
INTEGRATED CIRCUIT EDA ELITE CHALLENGE

Thanks!