ICCAD 2023

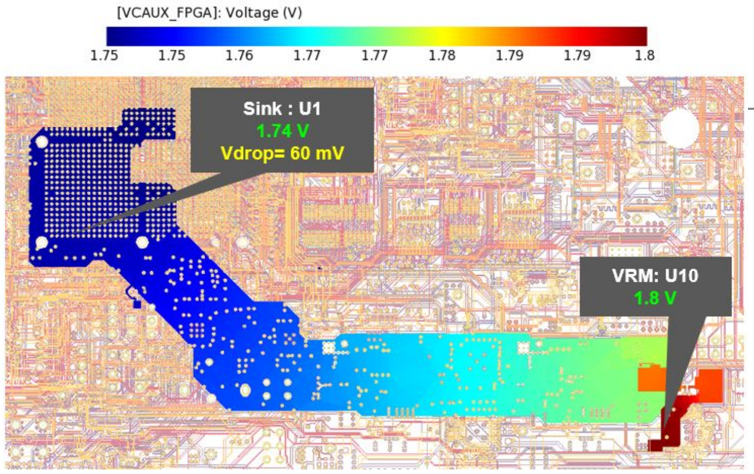
CAD Contest

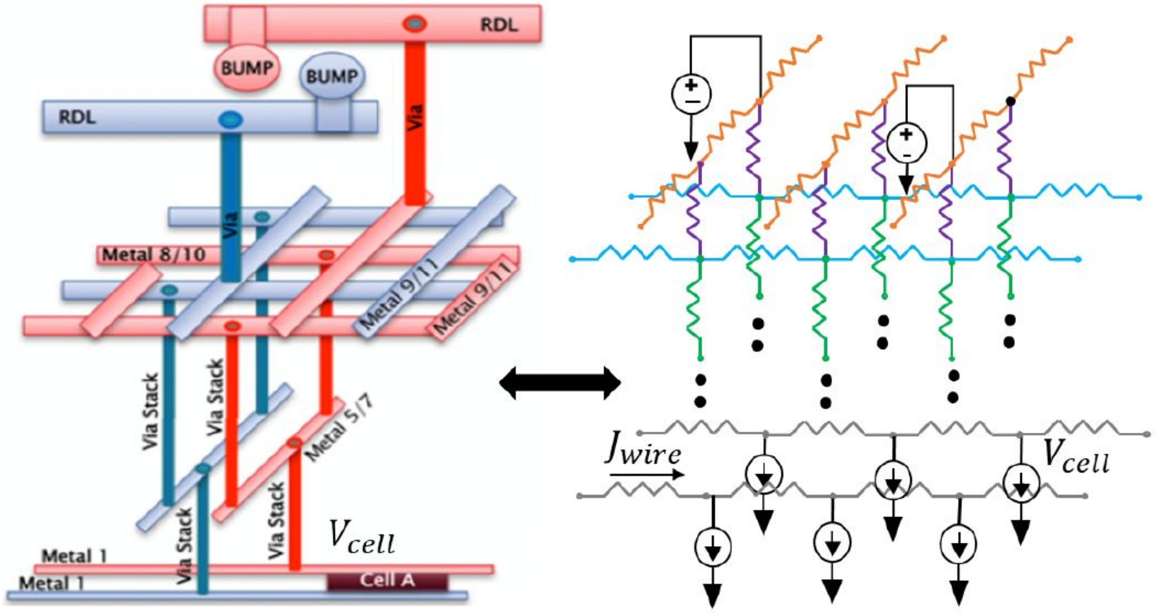
Static IR Drop Estimation Using Machine Learning

Presenter: Yu-Tung Liu, Yu-Hao Cheng, Shao-Yu Wu
Advisor: Hung-Ming Chen

PDN & IR drop

Power delivery network (PDN) is a part of chip design responsible for connecting power pad (VCC & VSS) to each cell





Power delivery network (PDN)

Due to the parasitic in the PDN, voltage drop will be introduced between power pad and cell, so we need a way to estimate voltage drop to ensure the design is feasible.

Conductance matrix: $G = \frac{1}{R}$

Vector of current: J

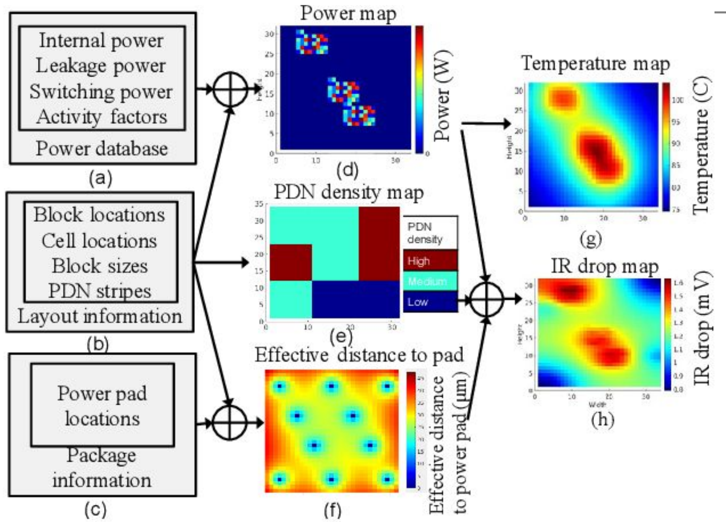
Voltage (IR drop): $V = G^{-1}J$

Traditional method: Solving Linear equations

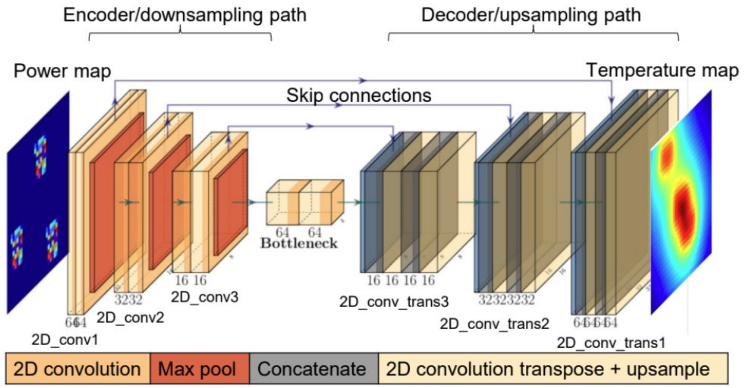
Machine Learning for IR drop

Image based method (CNN)

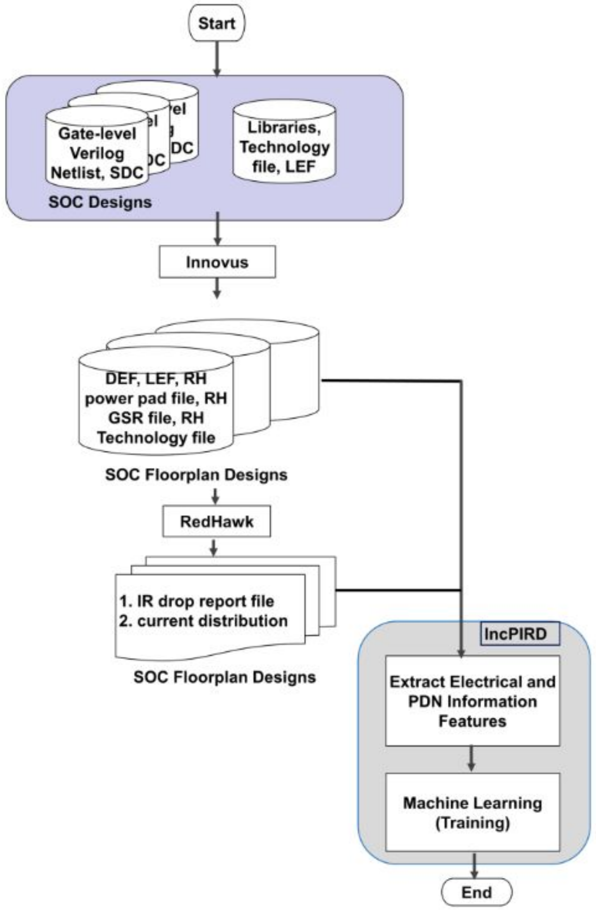
Input features



Encoder-Decoder Unet



- SPICE based method (XGBoost / GNN)



SPICE

↓

Feature extraction

↓

XGBoost

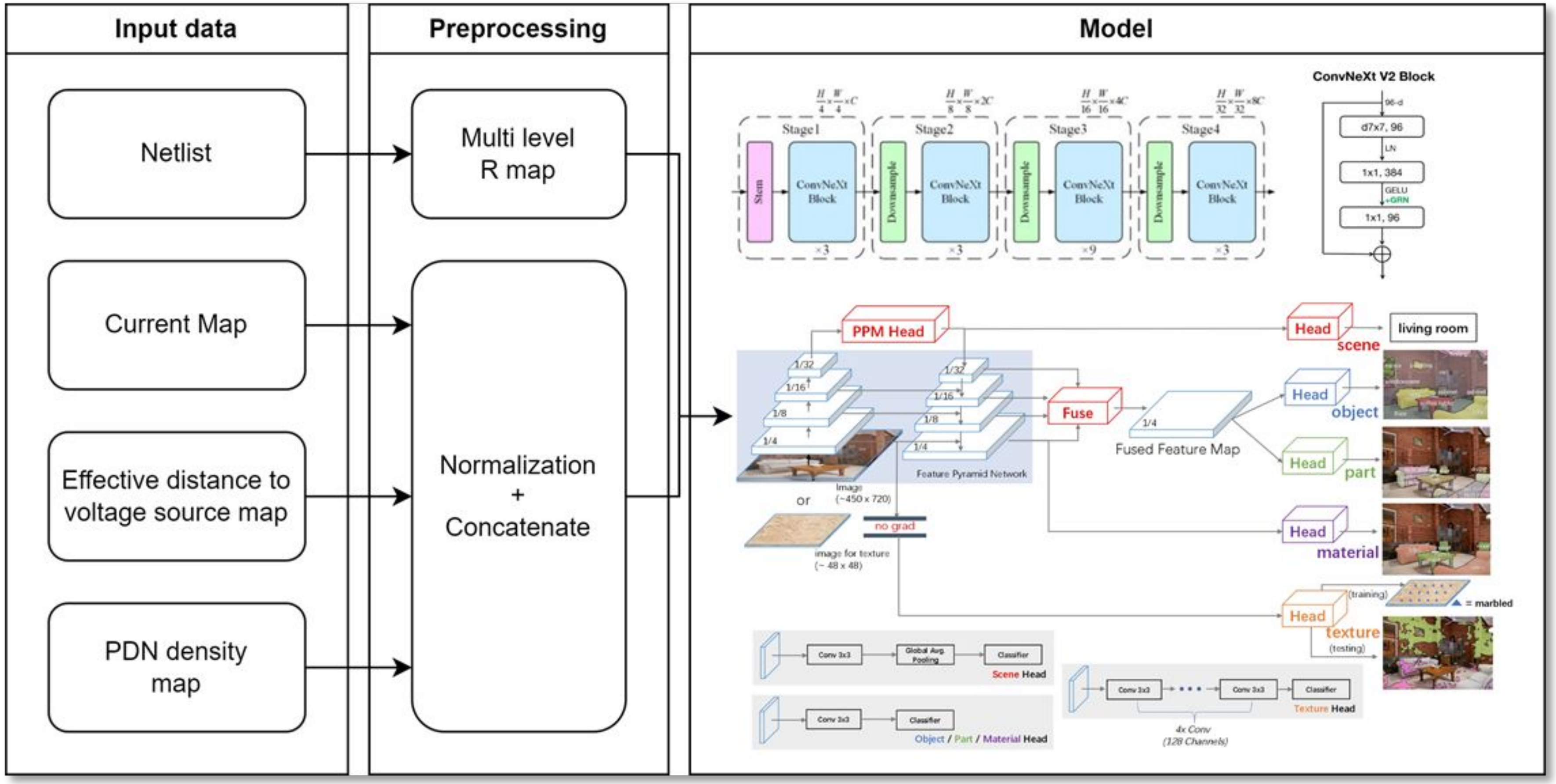
Training / Validation Data

Real circuit data for validation (10 data), Fake circuit data for training (10000 data)

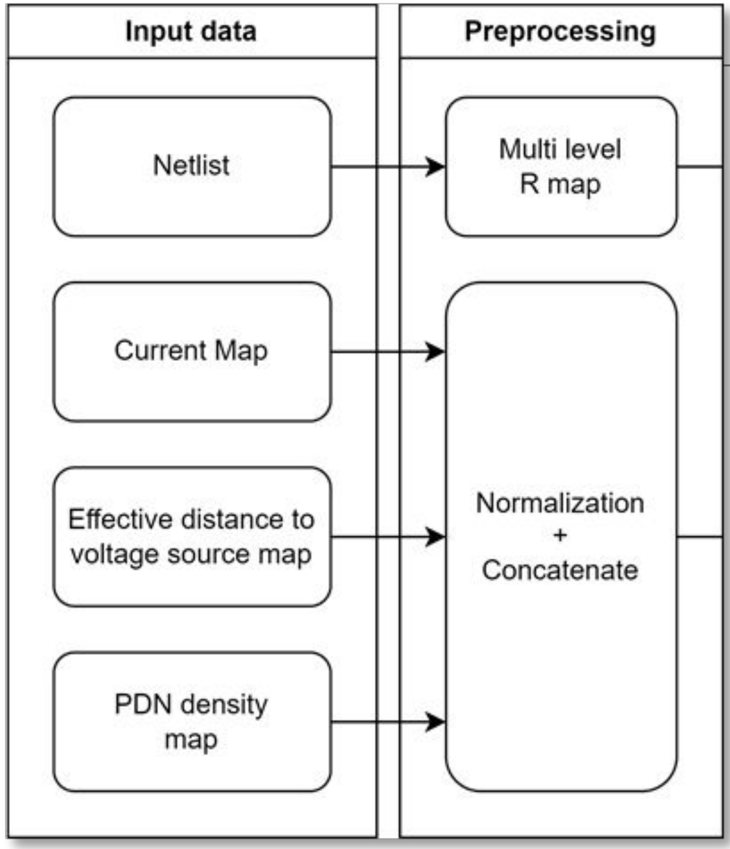
Feature	Real circuit data	Fake circuit data
Power map (Current Map)	OpenROAD-flow-scripts after placement	GAN-based power maps (no design required)
Power delivery network (PDN)	Template-based and regular PDNs in OpenROAD-flow-scripts	Template-based and regular PDNs synthesized in custom python script
Voltage source	Vary voltage source distribution in OpenROAD	Vary voltage source distribution in custom python script
SPICE netlist	Extracted from OpenROAD (PDNSim)	SPICE netlist created using python scripts
IR drop map	SPICE simulation and interpolation	SPICE simulation and interpolation

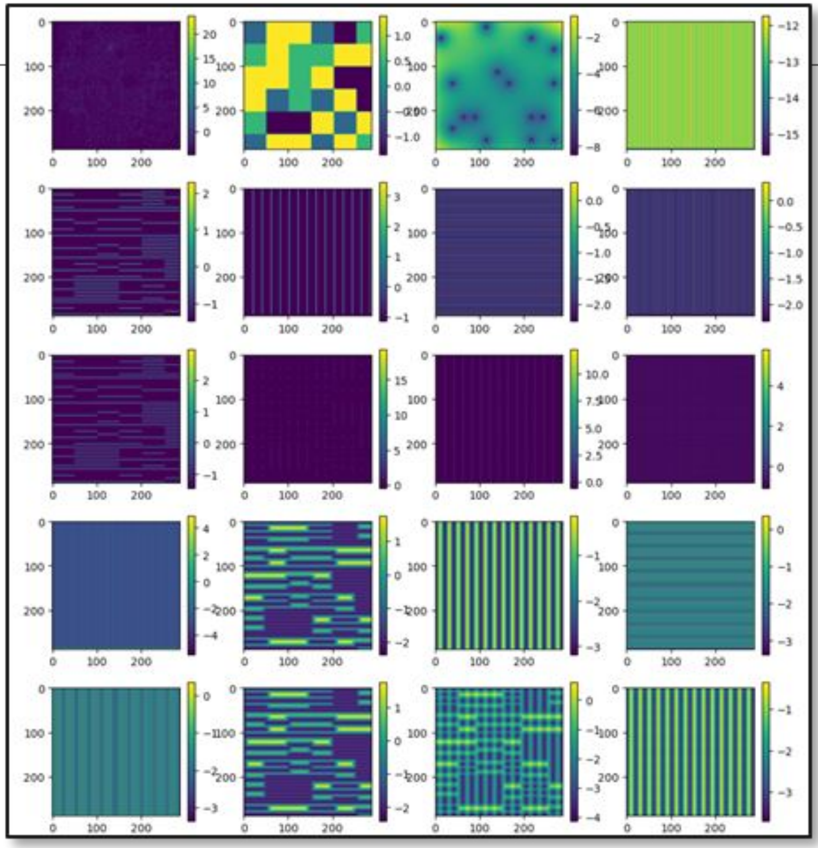
Our Method

Feature extraction + Image based CNN + FPN



Input data / Preprocessing

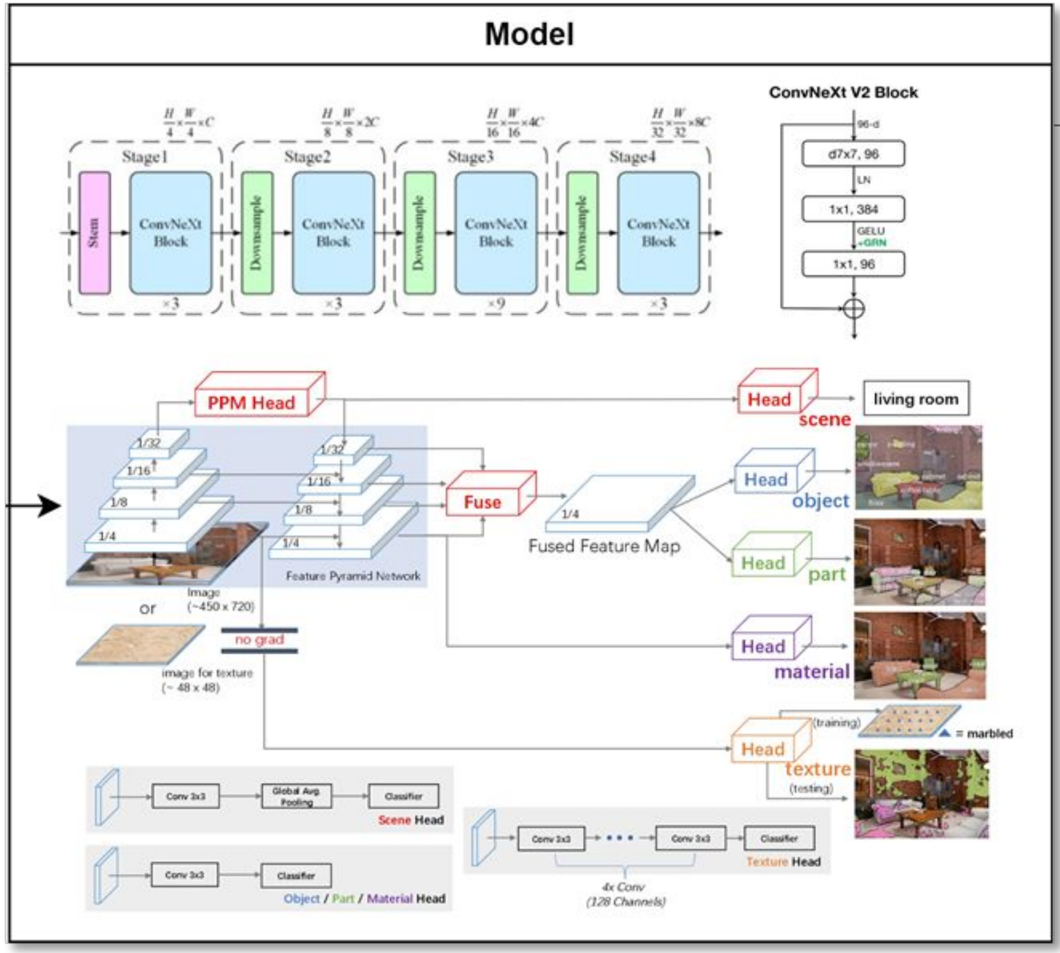




1. Produce resistance maps of every metal layer from Netlist file and their effective distance maps as part of model input.

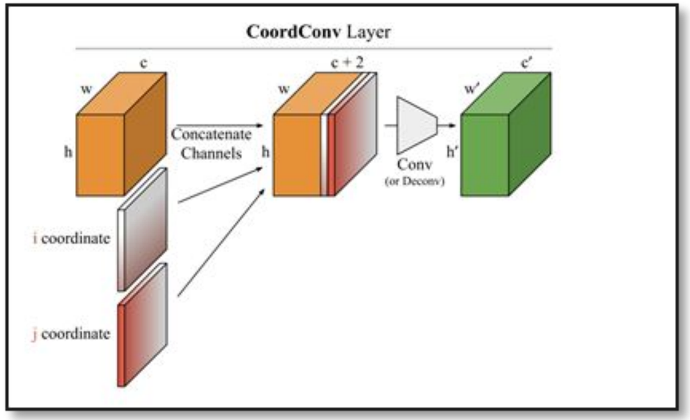
2. Normalize every map data and concatenate them as a stacked image data for model input.

Model Architecture

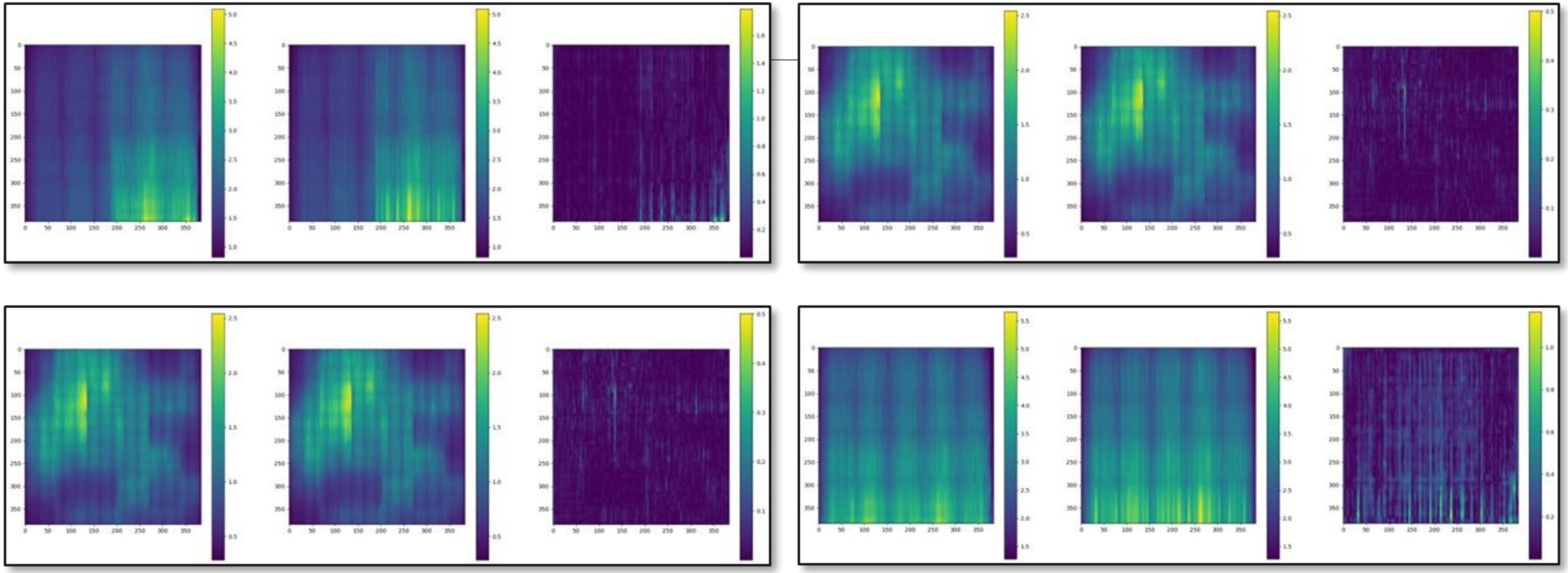


1. We use ConvNeXtV2-Nano as encoder, and custom designed UPerNet as decoder to merge multi-layer info and output the final IR-Drop prediction.

2. We also modify Stem block of ConvNeXtV2 from standard Conv to CoordConv to enhance location info.

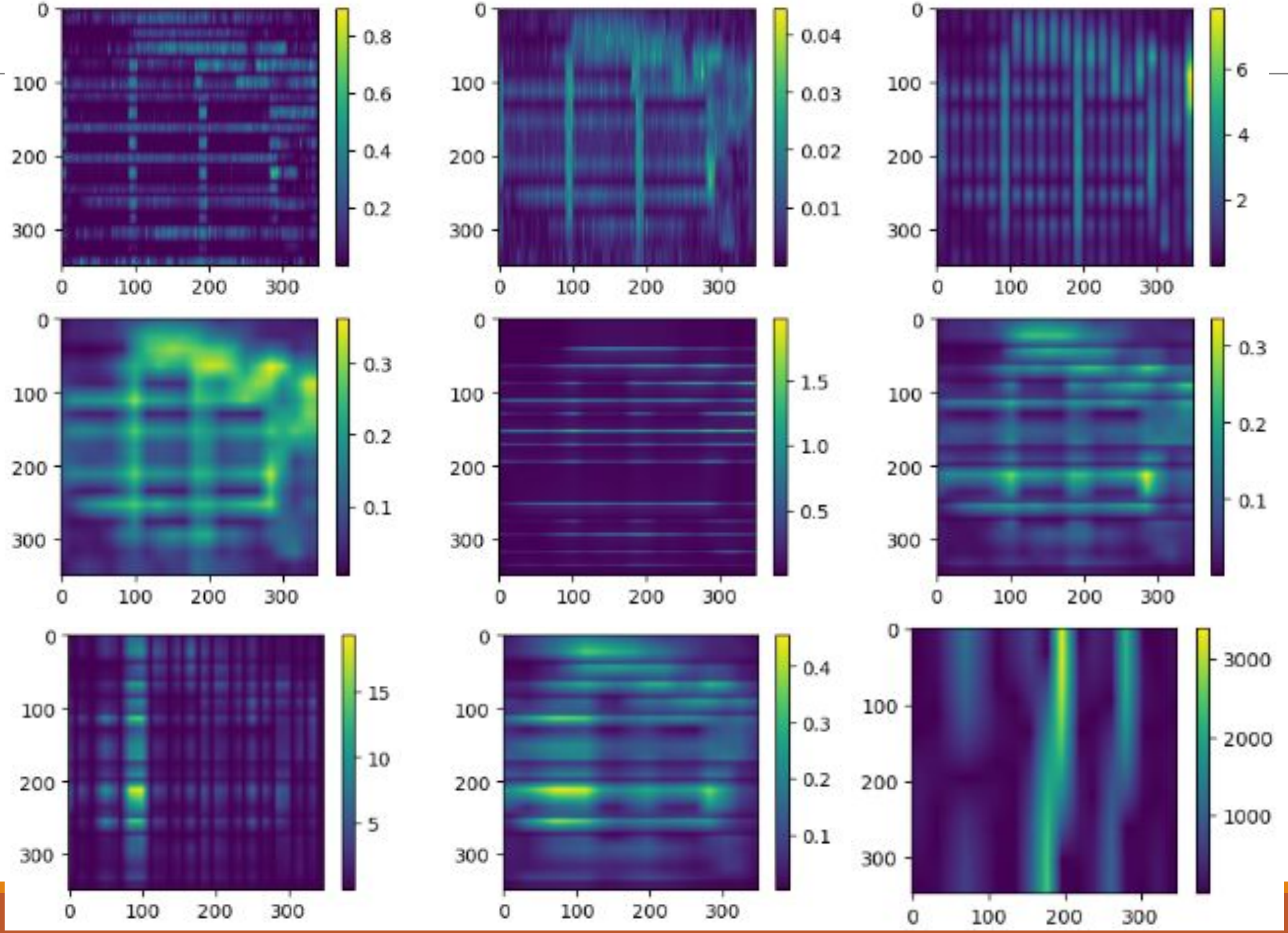


Result



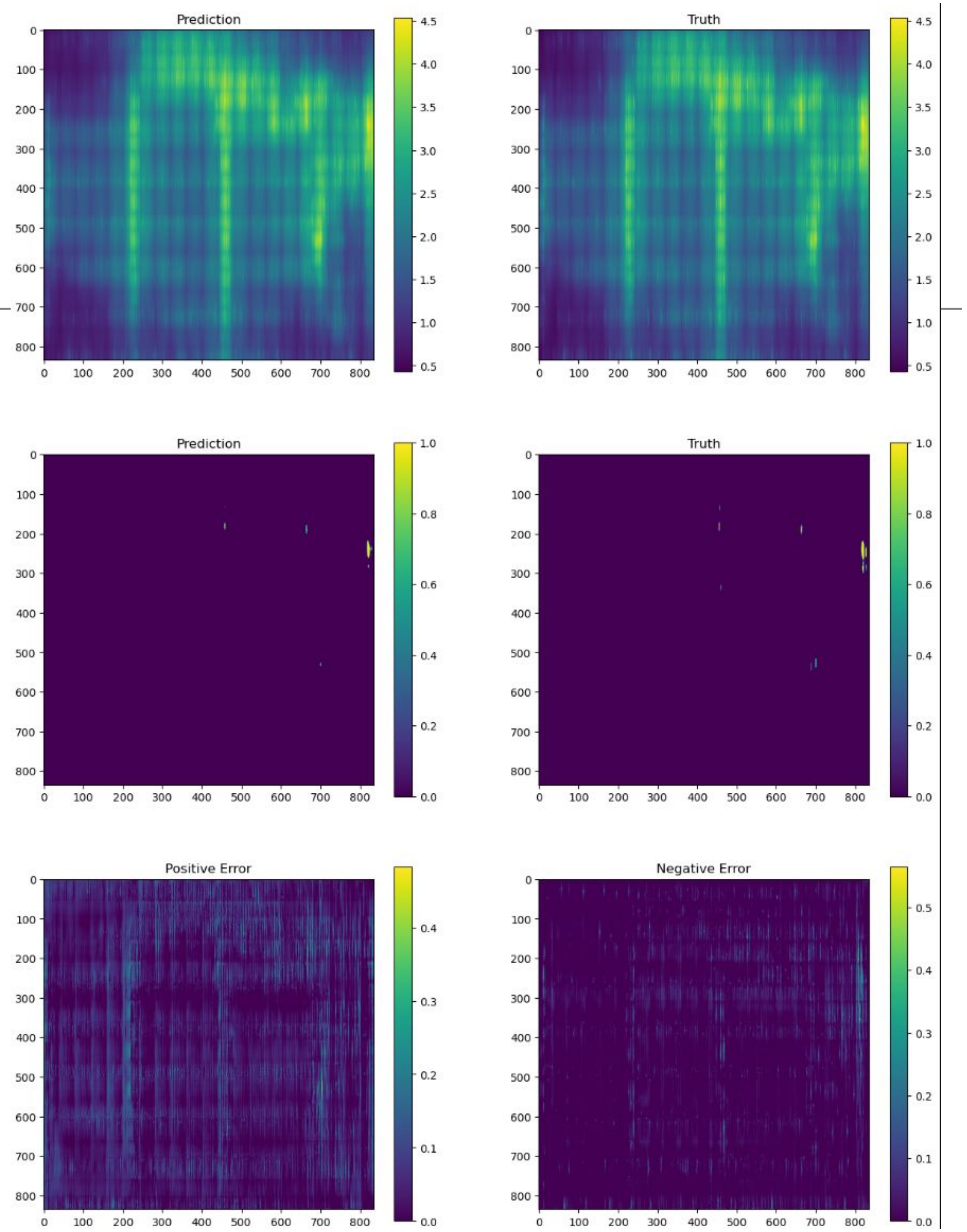
From left to right are prediction, label, and MAE, respectively. Avg MAE: 0.075mV, Avg F1-Score: 0.56

Augmented Features



Prediction Result

	MAE	
Train	0.0169mV	0.797
Validation	0.0762mV	0.634
Test	0.0843mV	0.505



Runtime

	Our Method (Python)	HSpice (C++)
Runtime (Largest Case)	15s	1min

1. The largest case is still too simple for HSpice

2. Python is much slower than C++

Discussion

1. More complex RTL code
2. How to compare to our previous work
3. Current industry EDA tool (redhawk)

Discussion on our contribution

1. The best on our dataset (open source) **but how to compare to others**

2. Runtime improved but not impressive

3. **How to integrate our method into current design flow?**