Week-1

2025-7-4

- ICLAD comments from 正源
- ForgeHLS review&open-source
- GraphLLM experiments
- Next week plan

- Appendix
  - 2 Graph building approachs
  - Training Loss 图

- ICLAD@Standford comments from 正源
  - Jason Cong做了一个HLS合成数据集
    - Iceberg! Enhancing HLS Modeling with Synthetic Data.
  - 也有其他做HLS数据集的
    - HLS-Eval! A Benchmark and Framework for Evaluating LLMs on High-Level Synthesis Design Tasks
  - Graph enhanced LLM的工作也有, 思路和我们差不多都是Vision那一套般到graph上
    - RTL++! Graph-enhanced LLM for RTL Code Generation
    - BRIDGES! Bridging Graph Modality and Large Language Models within EDA Tasks

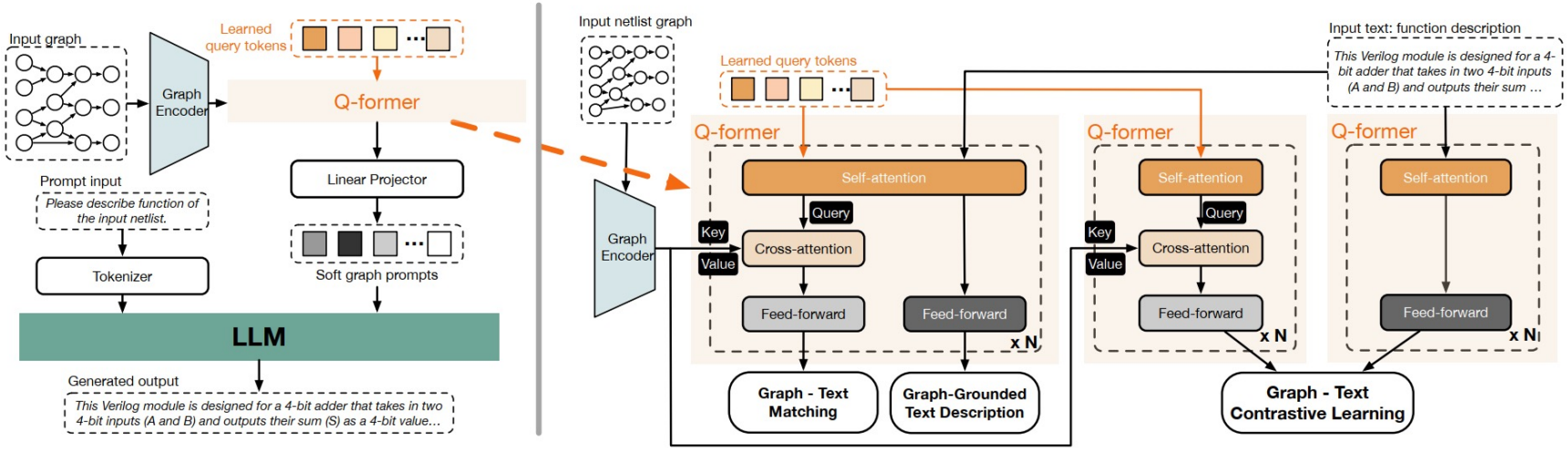


Fig. 4: **Left**: The graph-supported LLM architecture in BRIDGES. **Right**: Stage 1 pre-training of BRIDGES. The graph encoder and the cross-modal projector (Q-Former) are optimized together through three cross-modal tasks. Modules with the same color share the same parameters.

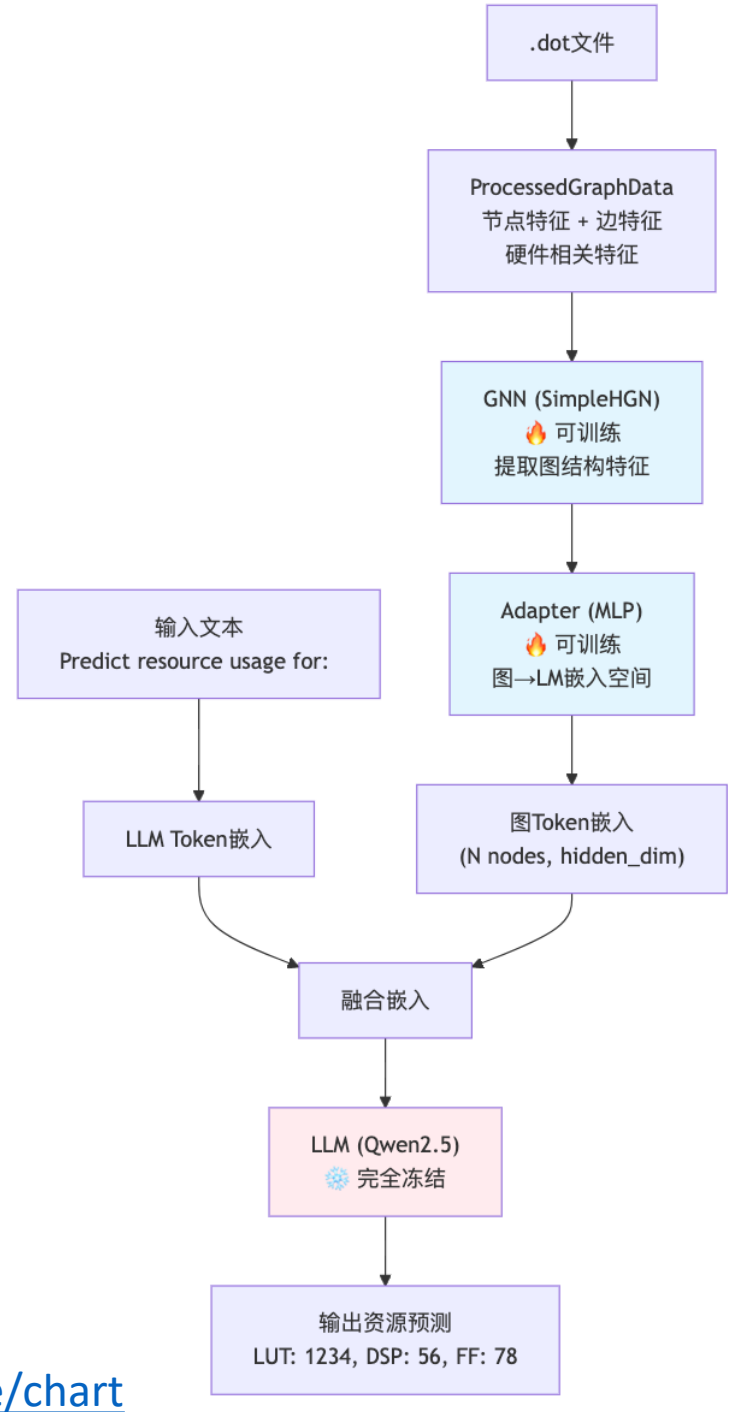
- “会场交流下来总的来说各种AI的apply都有，还是比较卷的”----zhengyuan

https://iclad.ai/program

- ForgeHLS
  - ICCAD review
  - 开源：
    - Huggingface: https://huggingface.co/datasets/zedongpeng/forgehls
      - ForgeHLS-dataset: same as paper
        - Kernels
        - Designs
      - ForgeHLS-lite-v1: 轻量版,每个kernel 100个designs (v1: 随机挑选)
        - Designs
      - ForgeHLS-benchmark: 聚类+人工+GPT4o挑选的80个kernels
    - Docs: https://zeju.gitbook.io/lcm-team with Deepcircuitx and ForgeEDA
    - Arxiv: https://arxiv.org/submit/6591950/preview
  - 转投
    - ~~ASP-DAC 明天摘要ddl，7.12(下周六) paper ddl~~
    - ~~DATE 9月 风格偏向技术创新 不友好~~
    - AAAI DB track?

- Common review
  - Strength
    - Synthetic code
    - Automated workflow
    - Valuable dataset
  - Weakness
    - Lack comparison of downstream task
    - Lack perfect coverage metrics
    - lack of novelty outside the dataset. 审稿人建议略写低novelty内容

- Graph enhanced LLM
  - Dataset: ForgeHLS-kernels w/o pragma
  - Graph: Build Graph from IR (same as Wunan)
    - C ---(Vitis HLS)---> IR ----(python script)---> .dot
  - Model:
    - SimpleHGN, 5 layers 512 dims
    - Qwen2.5-Coder-1.5B-Instruct
  - Train log:
    - Paragrams refers to swanlab

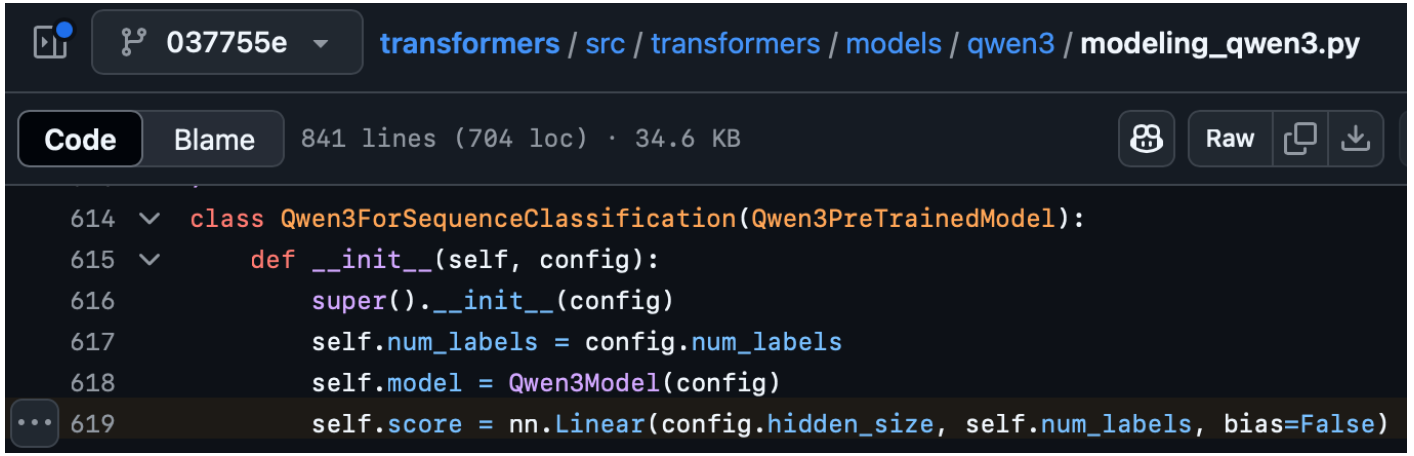


https://swanlab.cn/@zedong/GraphLLM-spec_generation/runs/s07z1woiwyq1gnm1oam7e/chart

- Graph enhanced LLM
  - GraphLLM 在 DSP 和 LUT 上表现优异
  - GNN训练经过10+次调参，forgehls-kernels数据集上难拟合
  - 整体MAPE较高，希望在大数据集上表现更好(SOTA MAPE<10%)

Model	DSP_MAPE	FF_MAPE	LUT_MAPE	DSP_RMSE	FF_RMSE	LUT_RMSE
GraphLLM	54.56%	126.75%	51.88%	40.32	5722.37	4823.72
GraphLLM+Code	55.23%	131.28%	47.38%	38.89	5958.19	5048.97
Graph_Text+Code	3446.67%	3370.63%	90.08%	559.69	3758.39	438.74
Graph_Text	64.67%	84.96%	98.29%	8.05	256.89	251.66
Code	322308.17%	6074.65%	98.03%	164312.98	331208.58	7800.54
Code-Lora	144.09%	86.27%	102.12%	31.70	2320.31	6821.32
GAT	60.25%	99.54%	99.71%	20.66	4368.80	4344.83
RGCN	205.72%	1425.85%	67.49%	120.84	4051.98	1877.79
SAGE	134.16%	1218.65%	65.03%	104.40	2282.19	2969.52

- NEXT MOVE
  - 1. GraphLLM
    - Train on ForgeHLS-designs (For example, PolyBench)
    - Train ProgSG/other SOTA
    - For regression tasks with a decoder-only model:
      - Replace the classification head
      - Replace loss
  - 2. ForgeHLS
    - 调研添加SOTA实验
    - 添加合成数据集上的训练实验
    - 发邮件询问previous数据集缺少的信息



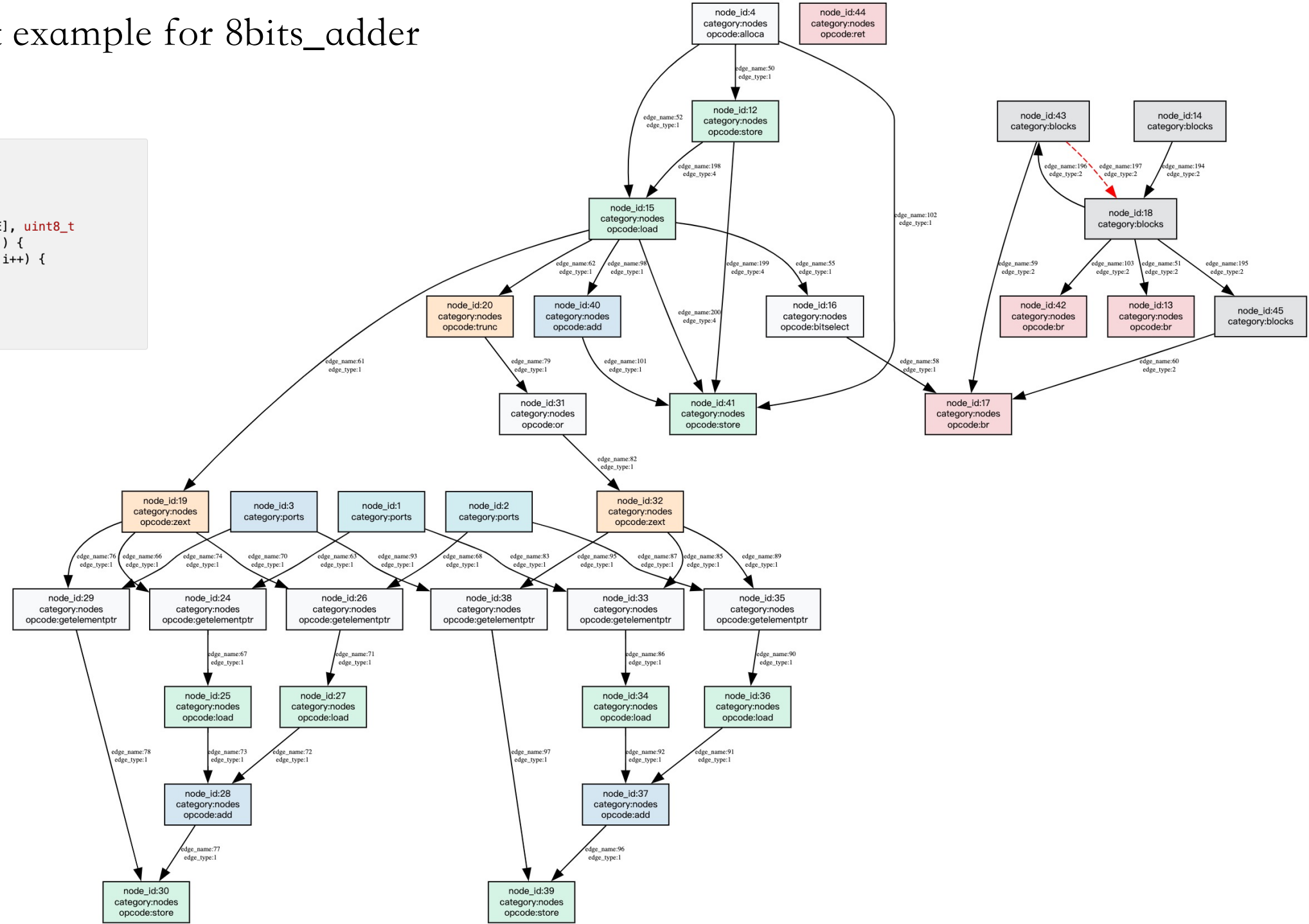
https://github.com/huggingface/transformers/blob/037755ed54208eefa77673b0af2a0b13e51f2fb1/src/transformers/models/qwen3/modeling_qwen3.py#L619

Appendix

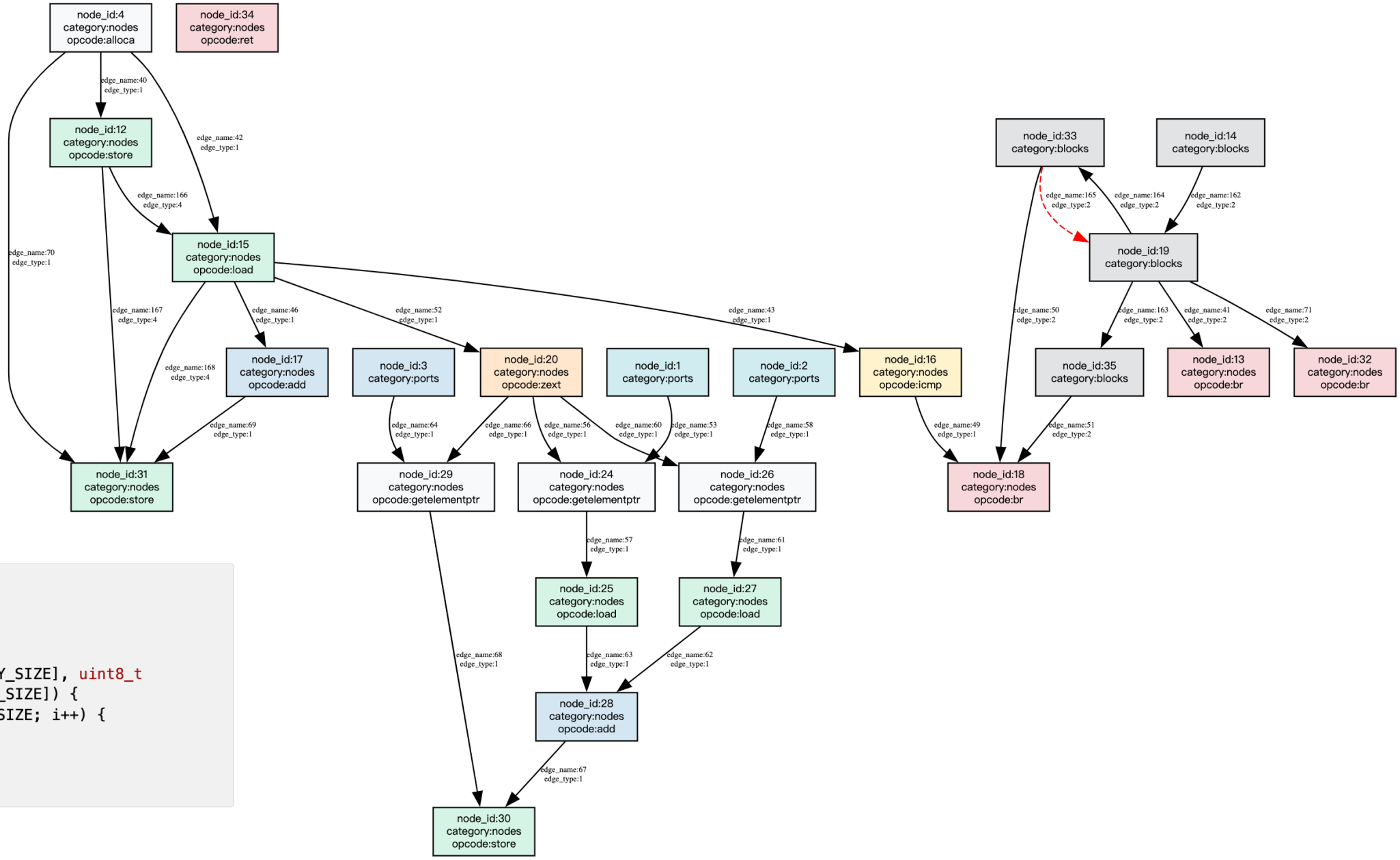
- HLS unroll 2 .dot example for 8bits_adder

hls unroll Vitis HLS -->adb --> graph

```
#include <stdint.h>

#define ARRAY_SIZE 1024

void adder_8bit(uint8_t A[ARRAY_SIZE], uint8_t
B[ARRAY_SIZE], uint8_t C[ARRAY_SIZE]) {
    for (int i = 0; i < ARRAY_SIZE; i++) {
        #pragma HLS unroll factor=2
        C[i] = A[i] + B[i];
    }
}
```



- No pragma .dot example for 8bits_adder



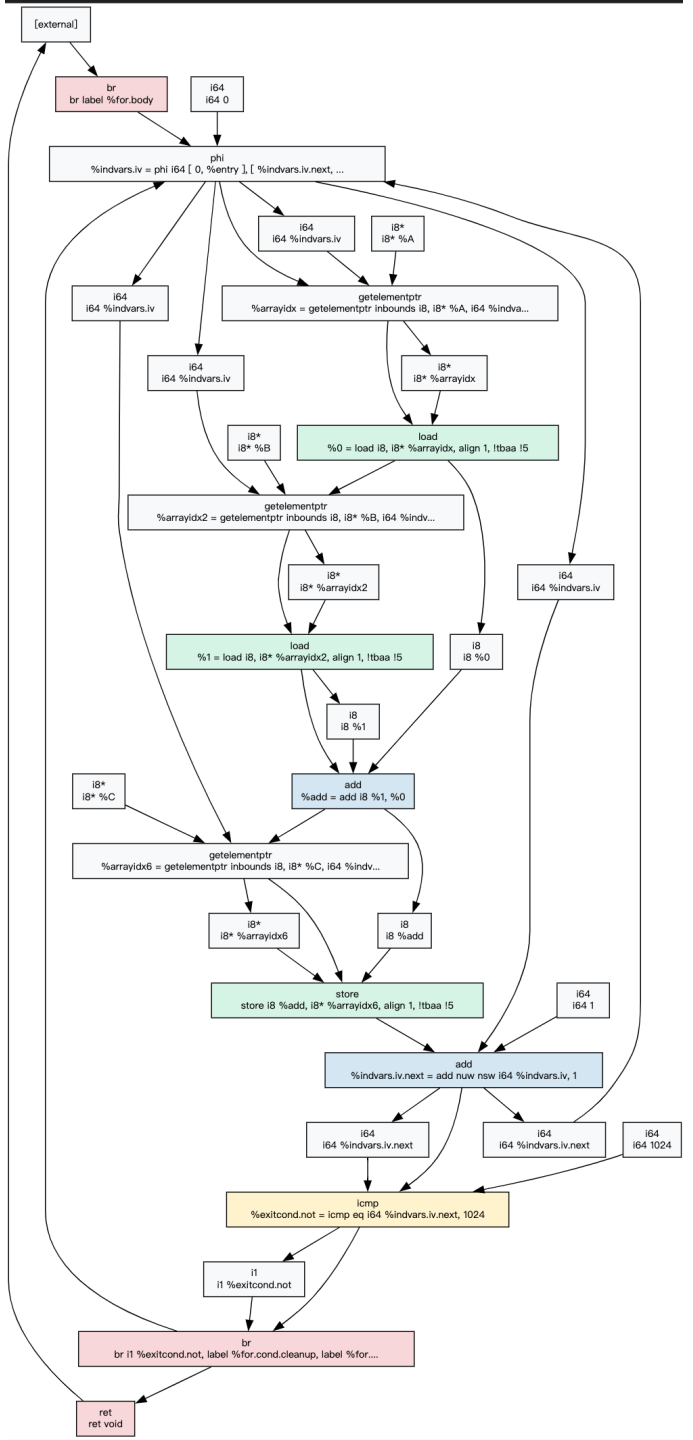
no pragma

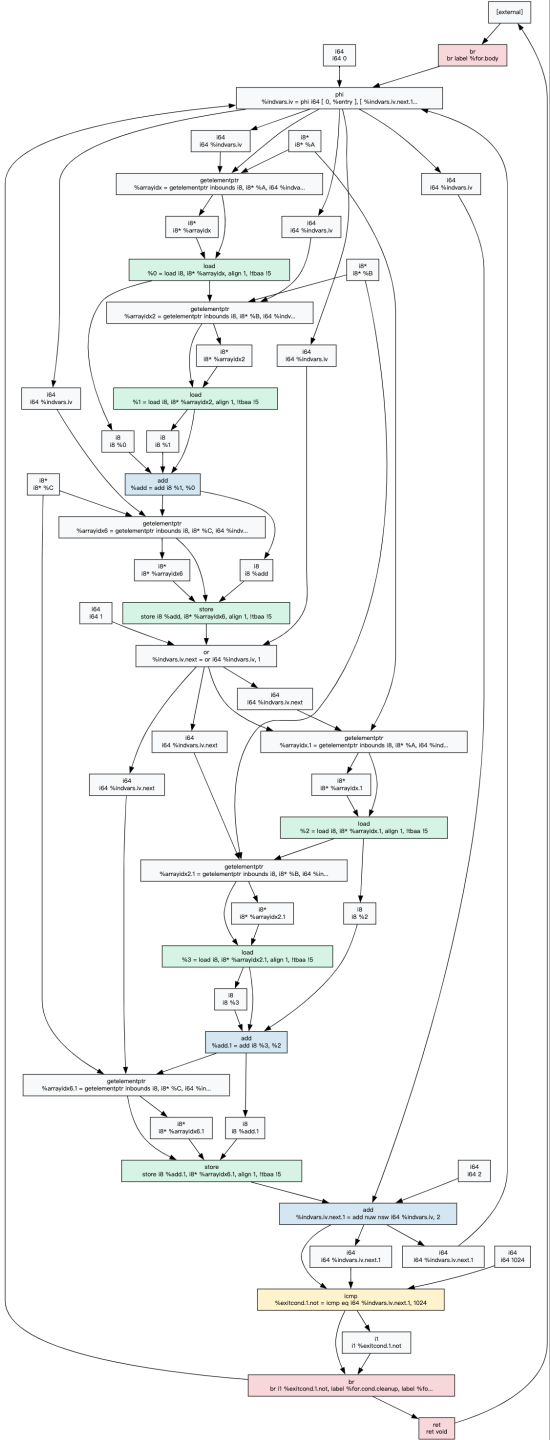
```c
#include <stdint.h>

#define ARRAY_SIZE 1024

void adder_8bit(uint8_t A[ARRAY_SIZE], uint8_t
B[ARRAY_SIZE], uint8_t C[ARRAY_SIZE]) {
    for (int i = 0; i < ARRAY_SIZE; i++) {
        C[i] = A[i] + B[i];
    }
}
```

- c unroll

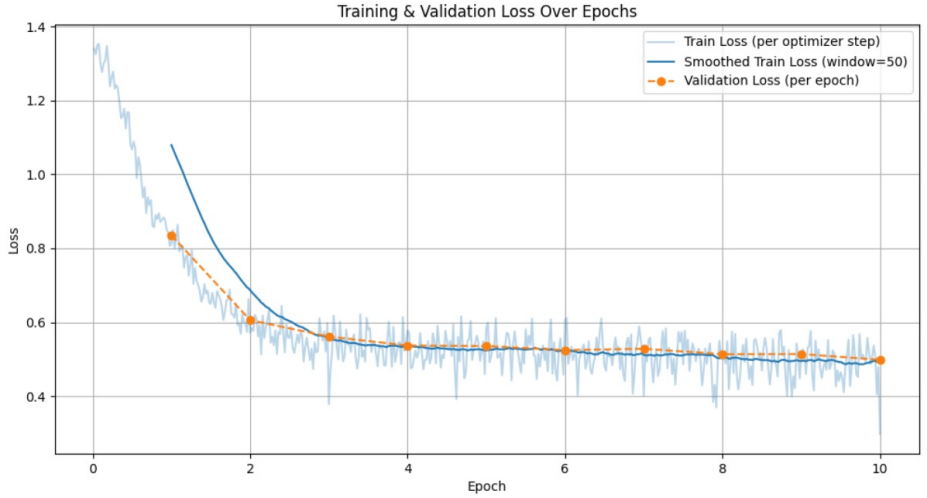
c unroll c --> llvm IR --> pragraml.from_llvm_ir -->graph

```c
#include <stdint.h>

#define ARRAY_SIZE 1024

void adder_8bit(uint8_t A[ARRAY_SIZE], uint8_t
B[ARRAY_SIZE], uint8_t C[ARRAY_SIZE]) {
    #pragma unroll 2
    for (int i = 0; i < ARRAY_SIZE; i++) {
        C[i] = A[i] + B[i];
    }
}
```





GiT/zedong/Code-Verification/train/output/my_codegen_training_run_25-07-03-23-45/loss_plot.png



llama_facroty_sft/yaml/saves/qwen/lora/sft/training_loss.png

GNN/saves/gat/dsp/all_numerical_forgehls_kernels/default/20250704_035655/comprehensive_training_analysis.png