Paper ID: 205

# DiffHLS: Differential Learning for High-Level Synthesis QoR Prediction with GNNs and LLM Code Embeddings

Zedong Peng, Zeju Li, Qiang Xu, Jieru Zhao
Presenter: Zedong Peng, Shanghai Jiao Tong University, China

## Motivation

```
void vector_add(int a[8], int b[8], int c[8]) {
   for (int i = 0; i < 8; i++) {
      c[i] = a[i] + b[i];
   }
}
```

Pragma insertion

```
void vector_add(int a[8], int b[8], int c[8]) {
#pragma HLS ARRAY_PARTITION variable=a factor=2
#pragma HLS ARRAY_PARTITION variable=b factor=2
#pragma HLS ARRAY_PARTITION variable=c factor=8
   for (int i = 0; i < 8; i++) {
#pragma HLS PIPELINE OFF
#pragma HLS UNROLL factor=2
      c[i] = a[i] + b[i];
   }
}
```

**Challenge**: **Same Kernel, Different Pragma**

High-Level Synthesis (HLS) compiles C/C++ to RTL, but exploring pragma-driven optimization choices is expensive—each design point requires time-consuming synthesis. Same kernel with different pragma configurations produces different QoR metrics (DSP, FF, LUT, CP), but shares common baseline structure.

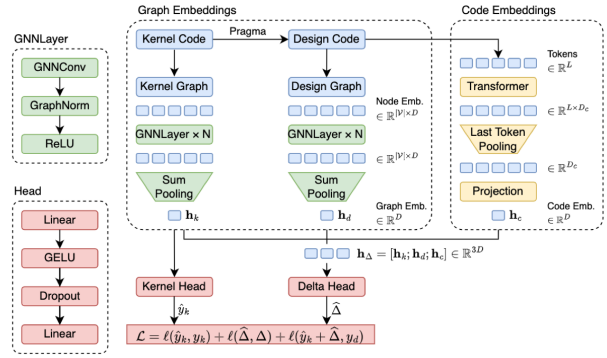
**Why Differential Learning?**

✓ **Decompose prediction**: kernel baseline + pragma-induced delta
✓ **Reduce target variance**: Focus model on pragma effects
✓ **Natural diagnostics**: Separate baseline vs. delta errors

**Why Combine Graphs + Code?**

✓ **IR graphs**: Capture control/data dependencies
✓ **Source code**: Retain pragma annotations and local patterns
✓ **LLM embeddings**: Provide pragma-aware semantic context

## Architectur




**Dual-Branch GNN Encoding**

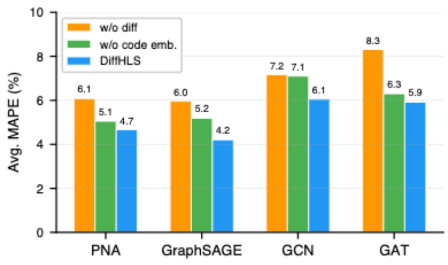
- Kernel Branch: Encodes original ke
- Design Branch: Encodes pragma-in

**LLM Code Embedding**

- Input: Design source code with pra
- Output: Last token pooling → Ligh

## Experimental Results

| Backbone | Method | DSP | FF | LUT | CP |
|---|---|---|---|---|---|
| PNA | GNN [1] | 14.71 | 26.47 | 22.86 | 8.87 |
| | HGNN [2] | 7.84 | 9.65 | 10.55 | - |
| | DiffHLS | **3.63** | **8.18** | **5.52** | **1.34** |
| GraphSAGE | GNN [1] | 17.01 | 39.11 | 28.09 | 8.25 |
| | HGNN [2] | 6.94 | 9.99 | 9.86 | - |
| | DiffHLS | **3.31** | **7.34** | **5.12** | **1.06** |
| GCN | GNN [1] | 25.30 | 38.34 | 28.64 | 8.79 |
| | HGNN [2] | **7.45** | 11.24 | 11.27 | - |
| | DiffHLS | 8.10 | **9.31** | **5.46** | **1.37** |
| GAT | GNN [1] | 28.66 | 54.73 | 46.19 | 10.32 |
| | HGNN [2] | 8.18 | 11.57 | 12.34 | - |
| | DiffHLS | **6.41** | **9.63** | **6.11** | **1.53** |




| Head | Backbone | Code Emb. | DSP MAPE | DSP MAE | DSP $R^2$ | FF MAPE | FF MAE | FF $R^2$ | LUT MAPE | LUT MAE | LUT $R^2$ | CP MAPE | CP MAE | CP $R^2$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Kernel $\hat{y}_k$ | PNA | No | **0.43** | **0.08** | **1.000** | 0.74 | 15.53 | 0.999 | 0.55 | 13.36 | 1.000 | **0.06** | **0.004** | **0.993** |
| | | Yes | 0.84 | 0.12 | 0.999 | **0.66** | **15.08** | **1.000** | **0.38** | **8.81** | **1.000** | 0.12 | 0.008 | 0.933 |
| | GraphSAGE | No | 0.89 | 0.14 | 0.999 | 0.94 | **20.13** | 1.000 | **0.58** | **13.54** | 1.000 | 0.11 | 0.008 | 0.955 |
| | | Yes | **0.51** | **0.09** | **1.000** | **0.86** | 20.48 | 1.000 | 0.61 | 13.83 | 1.000 | **0.07** | **0.005** | **0.998** |
| | GCN | No | 1.20 | 0.15 | **0.999** | **0.67** | **13.44** | 1.000 | 0.82 | 19.48 | 0.999 | **0.06** | **0.004** | **0.996** |
| | | Yes | **0.86** | **0.14** | 0.999 | 0.74 | 16.36 | **1.000** | **0.38** | **9.05** | **1.000** | 0.24 | 0.017 | 0.379 |
| | GAT | No | **0.79** | **0.12** | **0.999** | 2.09 | 51.36 | 0.994 | **0.91** | **22.56** | **0.999** | **0.10** | **0.007** | **0.992** |
| | | Yes | 1.63 | 0.25 | 0.997 | **1.01** | **21.56** | **0.999** | 1.16 | 30.21 | 0.998 | 0.12 | 0.008 | 0.971 |
| Delta $\hat{\Delta}$ | PNA | No | 13.55 | 1.38 | **0.971** | **58.52** | 676.20 | 0.961 | 13.71 | 641.63 | 0.965 | **35.80** | **0.090** | **0.709** |
| | | Yes | **11.37** | **1.23** | 0.872 | 66.76 | **631.17** | **0.972** | **12.07** | **627.14** | **0.988** | 38.13 | 0.099 | 0.654 |
| | GraphSAGE | No | 18.59 | 1.81 | 0.931 | 53.23 | 695.03 | **0.973** | 11.83 | 576.72 | 0.990 | 36.61 | 0.092 | 0.706 |
| | | Yes | **11.92** | **1.28** | **0.956** | **46.15** | **650.10** | 0.891 | **10.63** | **511.54** | **0.991** | **31.33** | **0.078** | **0.740** |
| | GCN | No | 38.82 | 3.37 | 0.895 | 59.39 | **995.09** | **0.661** | 13.60 | 754.88 | 0.980 | **36.13** | **0.093** | **0.675** |
| | | Yes | **26.43** | **2.86** | **0.938** | **32.41** | 1002.12 | 0.384 | **11.72** | **622.58** | **0.984** | 37.80 | 0.102 | 0.638 |
| | GAT | No | 29.27 | 2.59 | 0.939 | 125.26 | **754.27** | **0.966** | **13.14** | **563.01** | **0.991** | **33.77** | **0.094** | **0.657** |
| | | Yes | **22.45** | **2.17** | **0.948** | **59.00** | 906.03 | 0.857 | 12.63 | 713.02 | 0.981 | 39.63 | 0.112 | 0.585 |

**Main Results (Top-Left)**

DiffHLS outperforms GNN and HGNN baselines across all backbones and targets. GraphSAGE achieves up to 87% MAPE reduction on CP and 81% on LUT, demonstrating the robustness of differential learning.

**Prediction Head Analysis (Top-Right)**

Kernel head achieves near-perfect accuracy (<1% MAPE, $R^2$≈1.0), while delta head handles the harder pragma-driven changes. This validates our decomposition: separating easy baseline from hard delta focuses learning on the challenging component.

**Component Ablation Study (Bottom)**

Differential learning reduces MAPE by 21-34% on FF/LUT, while code embeddings improve DSP by 35%. The two components are complementary: differential learning decomposes predictions to reduce variance, while LLM embeddings capture pragma-aware semantic cues.

**CONCLUSION**

✓ DiffHLS decomposes QoR prediction into kernel baseline + pragma-induced delta
✓ Fuses GNN encoders with LLM code embeddings for complementary signals
✓ Achieves up to 87% MAPE reduction across all metrics and backbones

上海交通大學 SHANGHAI JIAO TONG UNIVERSITY
香港中文大學 The Chinese University of Hong Kong