Intelligent OPC Engineer Assistant for Semiconductor Manufacturing

Guojin Chen[1] Haoyu Yang[2], Bei Yu[1], Haoxing Ren[2]

[1]The Chinese University of Hong Kong, [2]NVIDIA

Association for the Advancement of Artificial Intelligence

OPC Agent

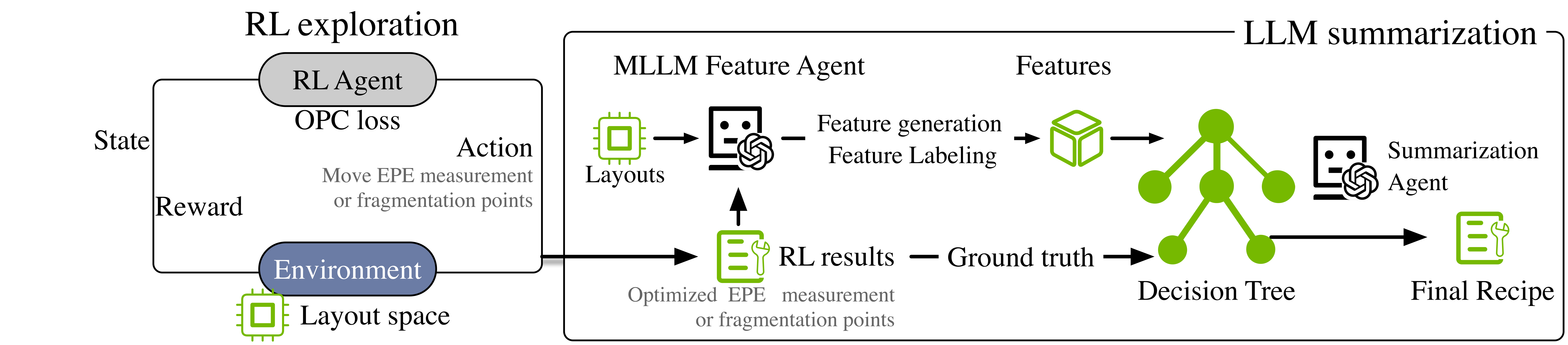


Figure 1. The two-stage approach for OPC recipe generation. The first stage employs RL to optimize OPC recipes. The second stage utilizes multi-modal LLM agents to efficiently summarize the results generated by RL and generate the final OPC recipes.

Introduction

- Recent advancements in semiconductor manufacturing have enabled increasingly complex computational capabilities, necessitating sophisticated optimization techniques.
- We present *Intelligent OPC Engineer Assistant*, a novel AI/LLM-powered methodology for optical proximity correction (OPC) that combines reinforcement learning with multi-modal agents.
- Experimental results demonstrate the efficacy of our approach in automatically generating OPC recipes across diverse chip designs, significantly reducing the manual effort required from experienced engineers.

Highlights

To enhance the efficiency of chip development,

1. We present the *Intelligent OPC Engineer Assistant*, an AI-driven framework designed to assist human engineers in the rapid development of OPC recipes.
2. This methodology integrates a reinforcement learning (RL)-based approach for optimizing objective searches and a multi-modality large language model (MLLM)-backboned agent system to facilitate spatial reasoning and recipe summarization.

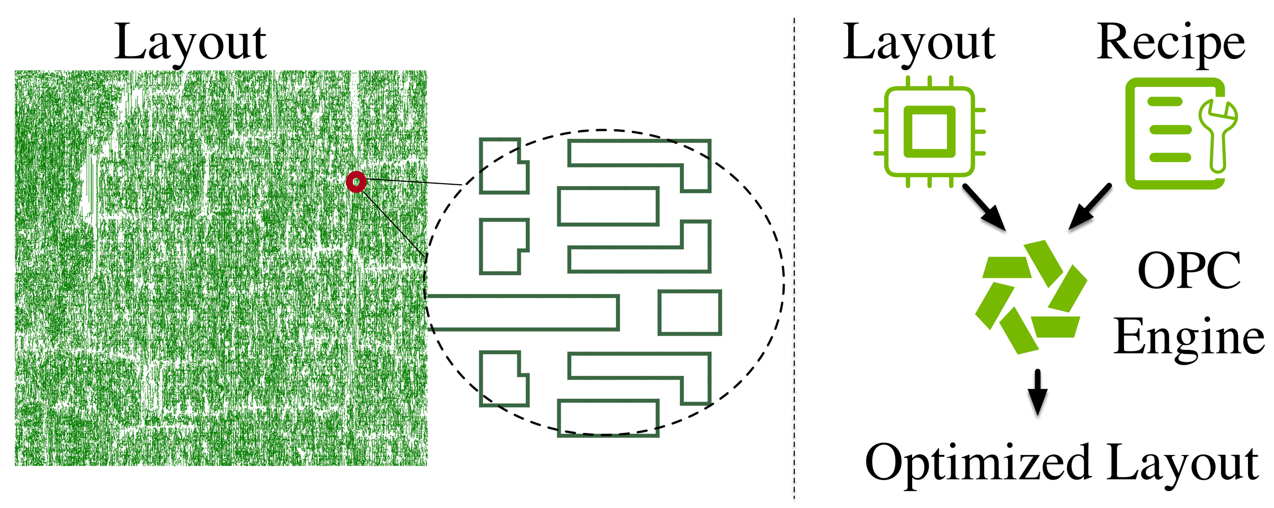
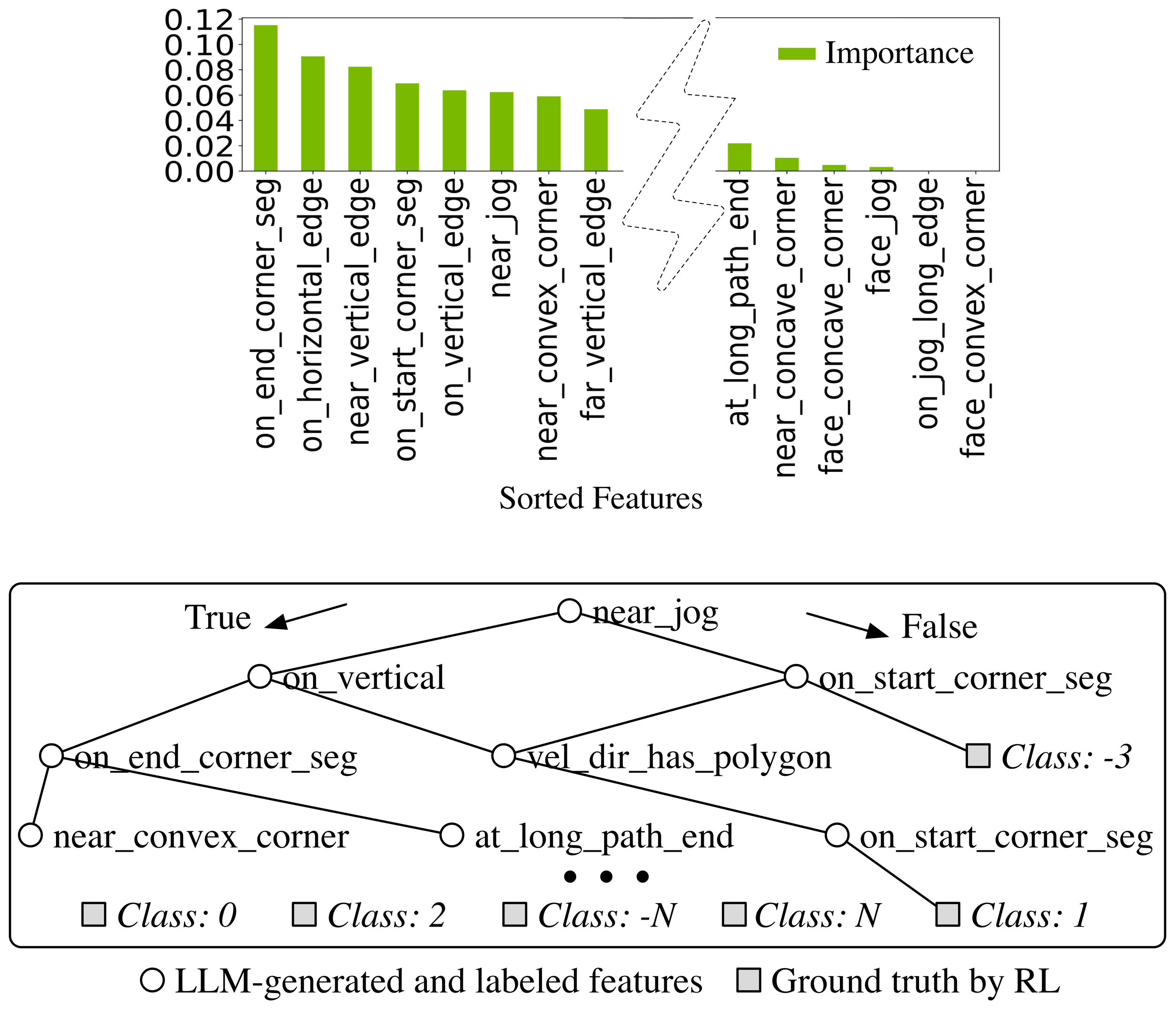


Figure 2. Left: The full-chip layout. Right: The relationship between the layout, OPC Recipe and OPC Engine.

Generation Example



(a) Decision tree example. The leaf nodes of the decision tree are labeled with ground truth based on RL results. The non-leaf nodes are features generated and labeled by the LLM.

```
...{condition: [near_jog, on_vertical, not vel_dir_has
_polygon, not on_start_corner_seg ], type: EPE, class: 1}
{condition: [not near_jog, not on_start_corner_seg ], type:
EPE, class: 3} {condition: [near_convex_corner, next_to_concave
_corner ], type: FRAG, class: -2} ...
```

(b) LLM-generated recipe example in `jsonl` format. Conditions are tree feature labels, and the type determines the tasks. The class represents the RL ground truth.

```
NEWTAG edge A short_len corner1 convex corner2 convex -out
line_end
NEWTAG neighbor both line_end short_len corner convex -out
line_end_adj
fragment_corner A convex concave mid_length 0.03
fragment_corner convex concave long_lenght 0.04
breakinhalf
retarget_layer A pattern0 curve0 pattern_epe curve1
emulate
```

(c) Downstream OPC software recipe example also includes statements for defining feature labels and movement distances.

Figure 3. Decision tree and recipe generation example. (a) Decision tree constructed from LLM-labeled features and RL ground truth. (b) Simplified LLM-generated recipe example in `jsonl` format. (c) Downstream OPC [?] software recipe example in Tcl language.

Evaluation on Accuracy

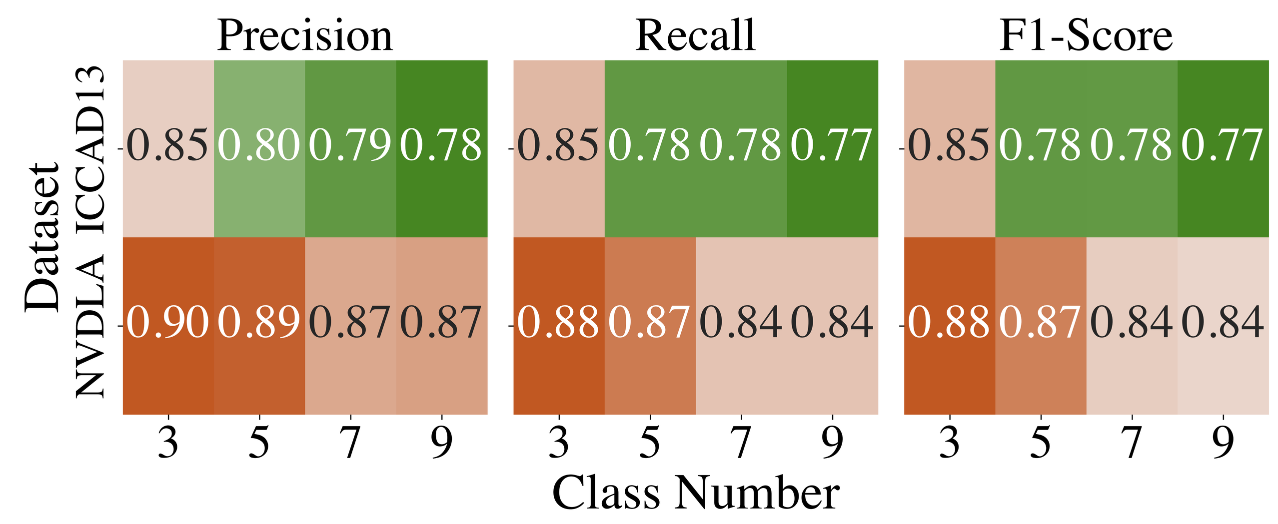


Figure 4. Decision Tree Efficiency on both ICCAD13 and NVDLA datasets.

	ICCAD13		
	OPC	OPC+LLM	OPC+RL
PVBand	53328	51271	50060
ratio	1.00	0.96	0.94
EPE N	119.70	107.00	105.60
ratio	1.00	0.89	0.88
EPE D	693.10	561.70	525.90
ratio	1.00	0.81	0.76
Runtime	4s	4s	3hr
	NVDLA		
	OPC	OPC+LLM	OPC+RL
PVBand	170899	161056	162096
ratio	1.00	0.94	0.95
EPE N	159.45	146.95	139.40
ratio	1.00	0.92	0.87
EPE D	869.25	766.10	741.05
ratio	1.00	0.88	0.85
Runtime	6s	6s	3hr

Table 1. Performance of the framework. The RL stage results are shown in the OPC+RL column, and the results related to the final LLM-generated recipe are shown in the OPC+LLM column.