Versatile Multi-stage Graph Neural Network for Circuit Representation

Shuwen Yang, Zhihao Yang, Dong Li, Yingxue Zhang,
Zhanguang Zhang, Guojie Song* & Jianye HAO*

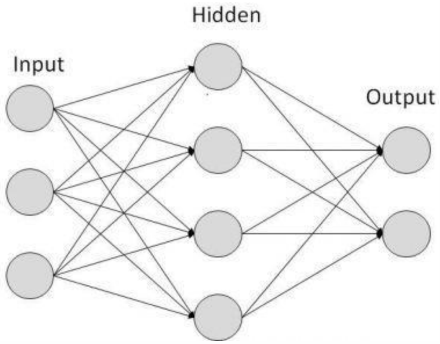
Deep Learning for EDA

Integrated circuits (ICs) are extensively used in modern electronic products like computers, smart-phones, and cars.

Due to the rapid growth in the scale of circuits, deep learning technologies have been widely exploited in Electronic Design Automation (EDA) to:

- speed up circuit design
- save labor costs

Circuit Prediction Task

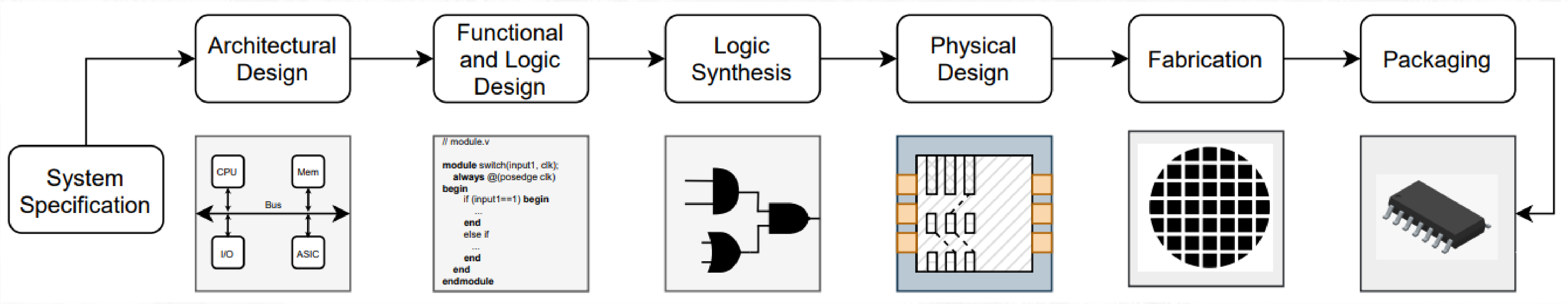


Neural Networks

Target of Circuit Prediction Task

- to find defected circuits early
- to guide circuit design

predict properties in early stages
(congestion, wire-length, …)



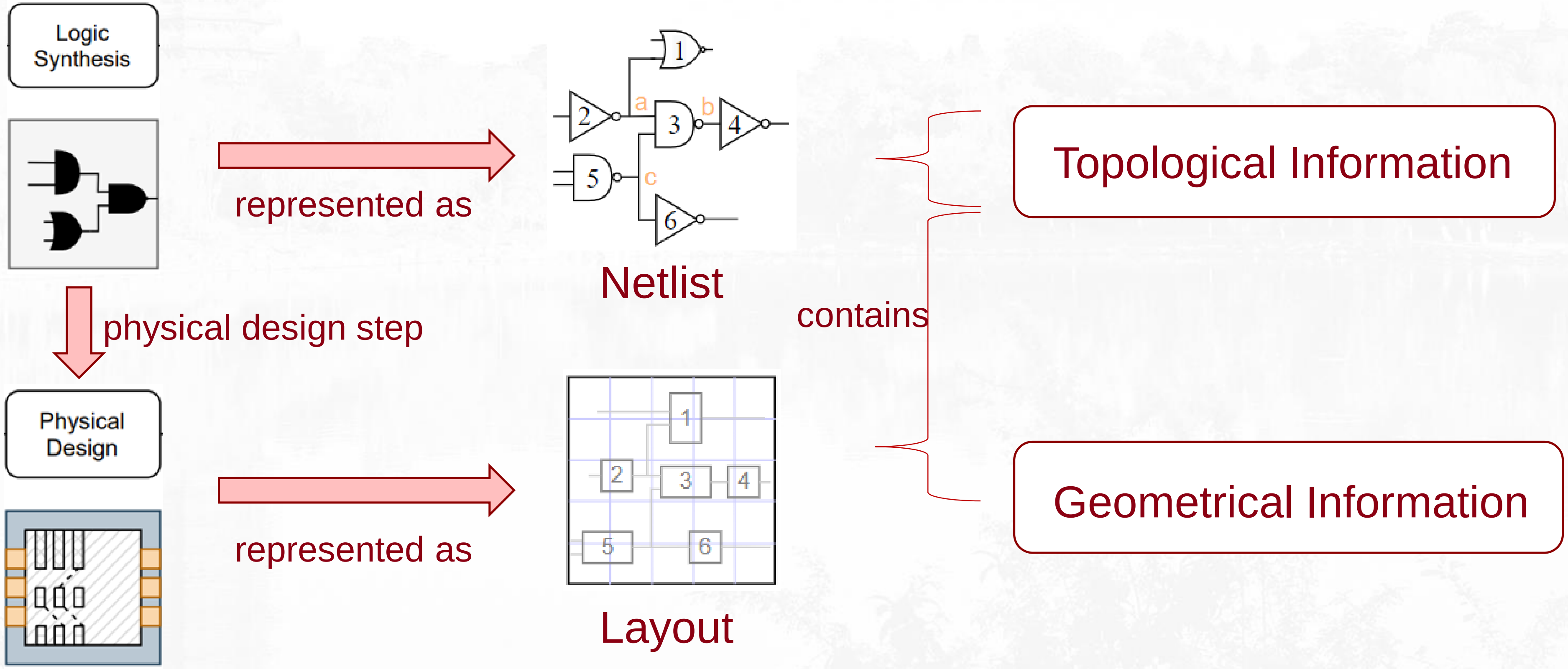
```
// module.v

module switch(input1, clk);
   always @(posedge clk)
begin
      if (input1==1) begin
         ...
      end
      else if
         ...
      end
   end
endmodule
```

Circuit Prediction Task

Challenges of Circuit Prediction Task:

- Complex and variant information underlying under different stages
- Generalize/transfer across tasks



Logic Synthesis

represented as

Netlist

physical design step

Physical Design

represented as

Layout

contains

Topological Information

Geometrical Information

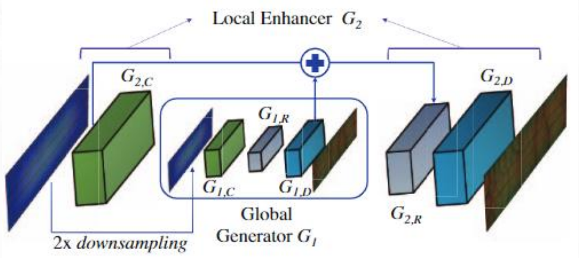
Circuit Prediction Task

Challenges of Circuit Prediction Task:

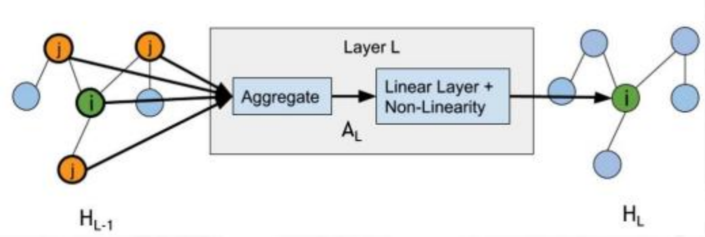
- Complex and variant information underlying under different stages

Existing methods are **stage-specific**:

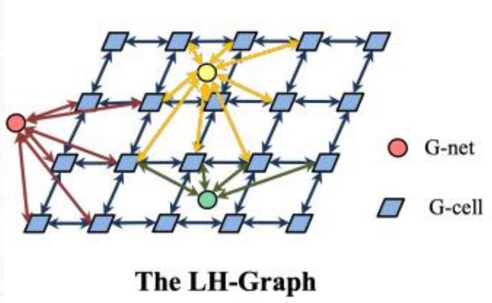
- focus on either topological/geometrical information
- not compatible with circuits on logic synthesis stage



pix2pix [8]
unaware of topology



CongestionNet [5]
unaware of geometry

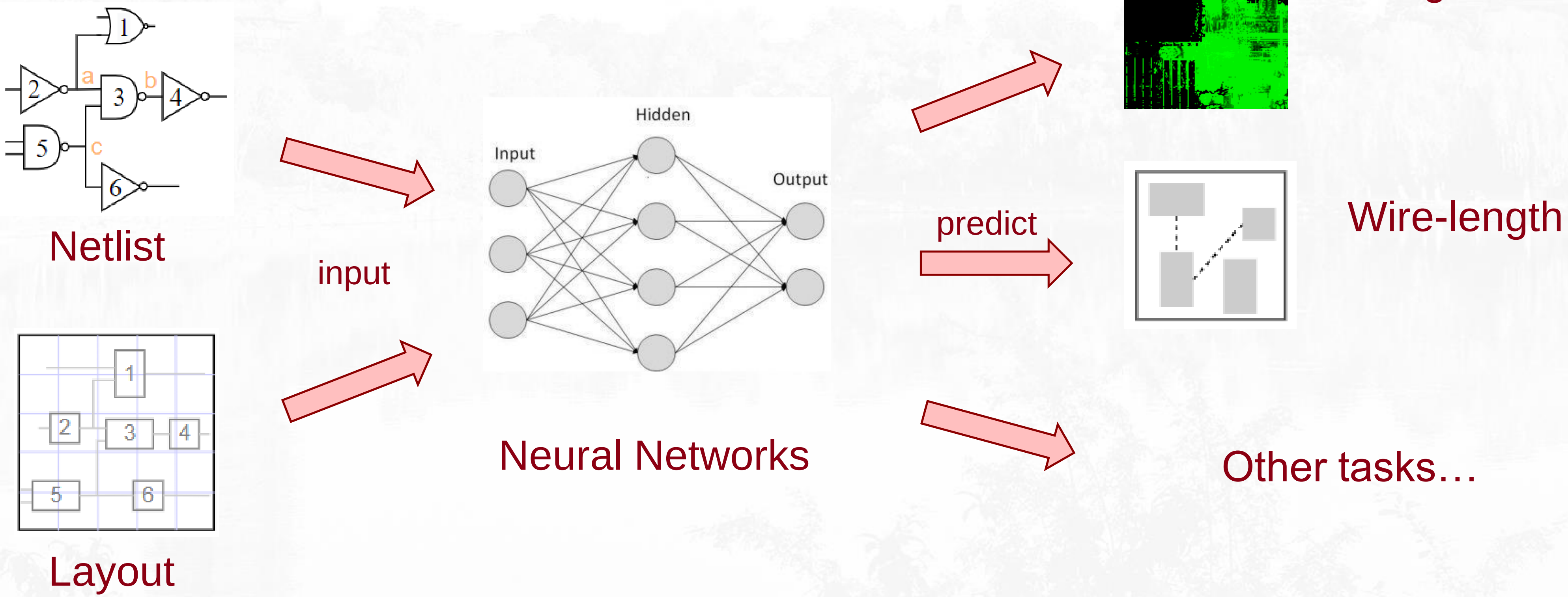


LHNN [14]
requires geometry

Circuit Prediction Task

Challenges of Circuit Prediction Task:

- Complex and variant information …
- Generalize/transfer across tasks



Netlist

Layout

input

Neural Networks

predict

Congestion

Wire-length

Other tasks…

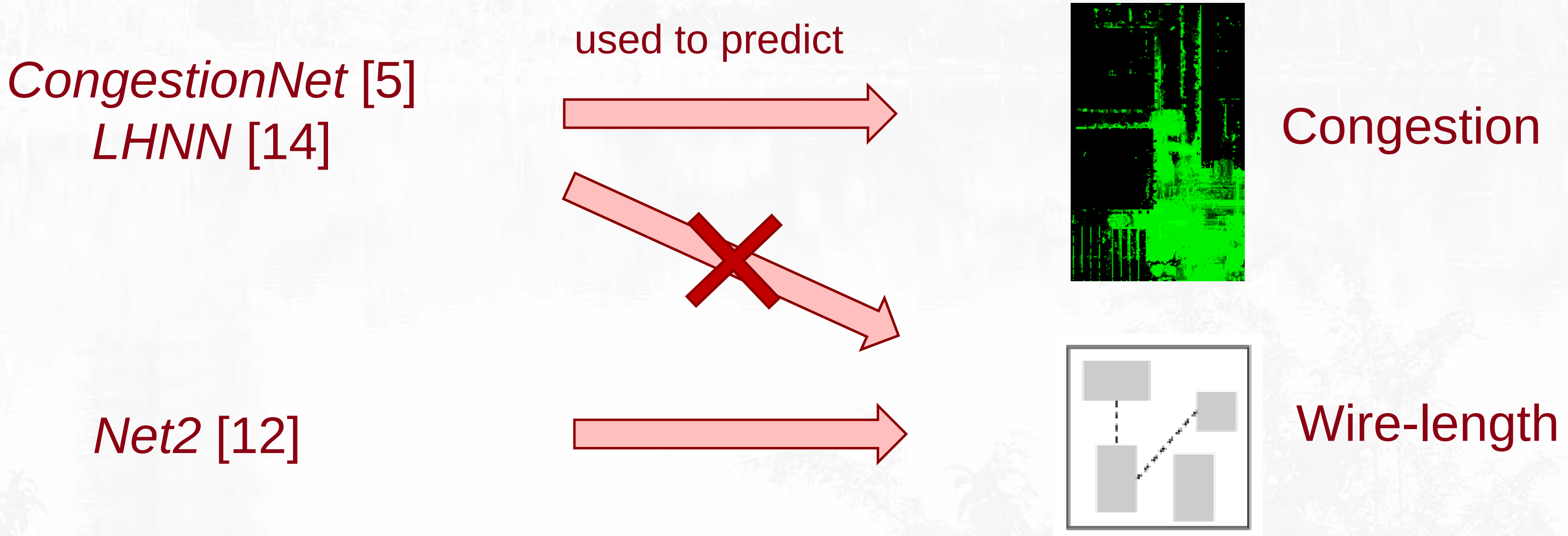
Circuit Prediction Task

Challenges of Circuit Prediction Task:

- Generalize/transfer across tasks

Existing methods are **task-specific**:

- focus on one (or few) prediction task(s)
- task-specific modules stifle knowledge transferring among tasks



CongestionNet [5]
LHNN [14]

used to predict

Congestion

Net2 [12]

Wire-length

Circuit Prediction Task

We propose a circuit representation framework:

- Data structure: **Circuit Graph**
- Neural network: **Circuit GNN**

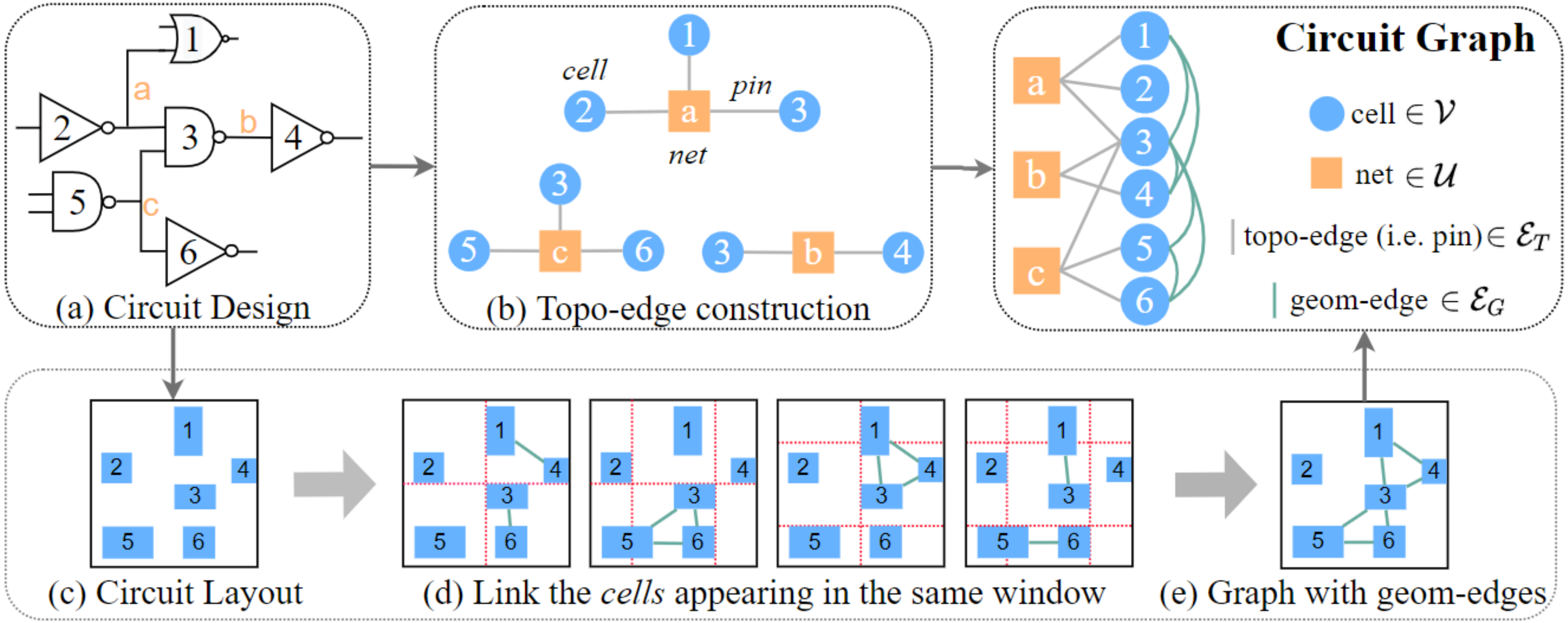
The solution is **VERSATILE**:

- **stage-adaptive**
  - can exploit and fuse topological/geometrical information
  - compatible with circuits in logic synthesis/placement stages
- **task-adaptive**
  - can be used to solve various circuit prediction tasks
  - can transfer knowledge learned from one task to others

Circuit Graph

Heterogeneous graph with two types of edges:

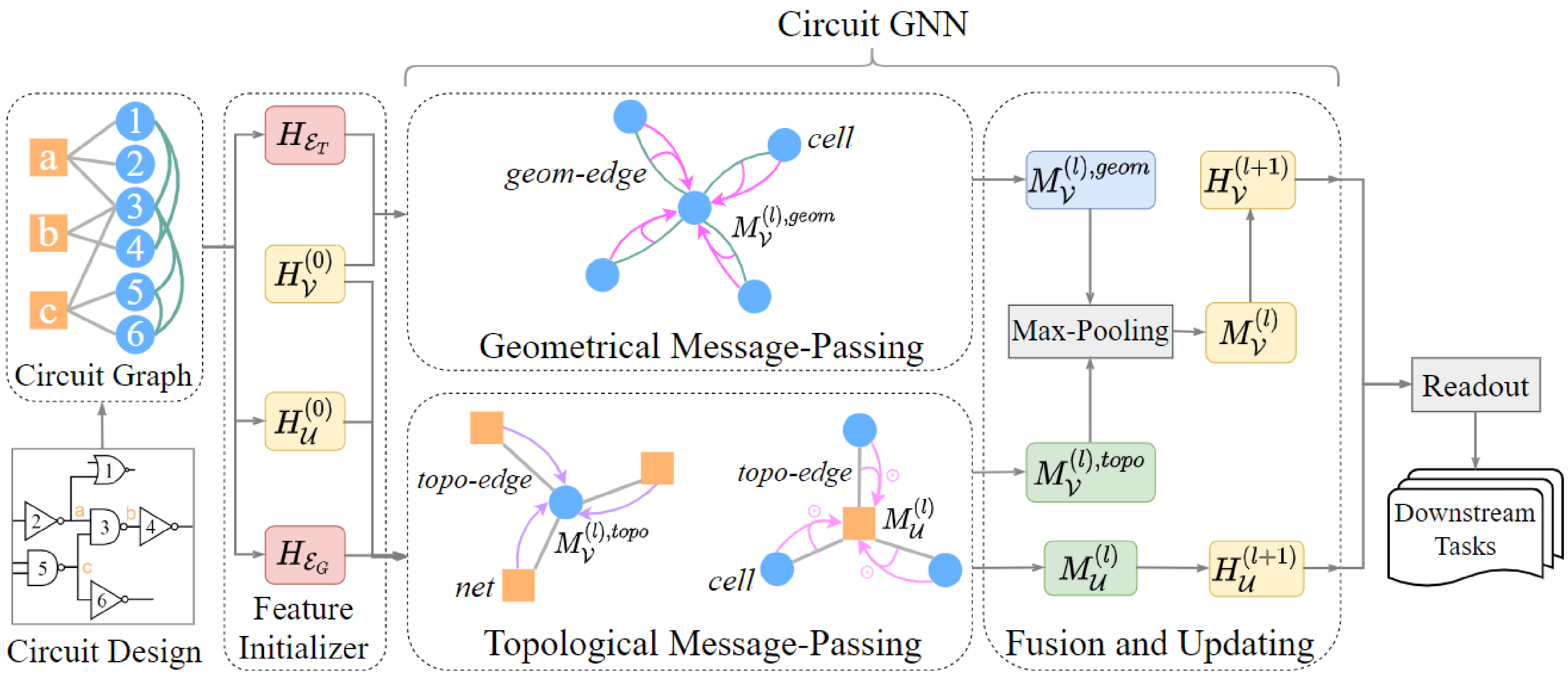
- topo-edges: topological connections between cells and nets
- geom-edges: geometrical adjacencies among cells (absent for logic synthesis circuits)


1
a
2
3
b
4
5
c
6
(a) Circuit Design
cell
pin
net
1
2
a
3
3
5
c
6
3
b
4
(b) Topo-edge construction
Circuit Graph
a
b
c
1
2
3
4
5
6
● cell ∈ $\mathcal{V}$
■ net ∈ $\mathcal{U}$
| topo-edge (i.e. pin) ∈ $\mathcal{E}_T$
| geom-edge ∈ $\mathcal{E}_G$
1
2
3
4
5
6
(c) Circuit Layout
(d) Link the *cells* appearing in the same window
(e) Graph with geom-edges

Circuit GNN

Multi-layer message-passing & fusing:

- Pass topological/geometrical messages and fuse them at the end of each layer
- Only pass topological message for logic synthesis circuits



Experiments

Congestion prediction results in logic/placement stages:

- Our solution is both efficient and effective
- Our solution is adaptive to both stages, whether or not geometry is available

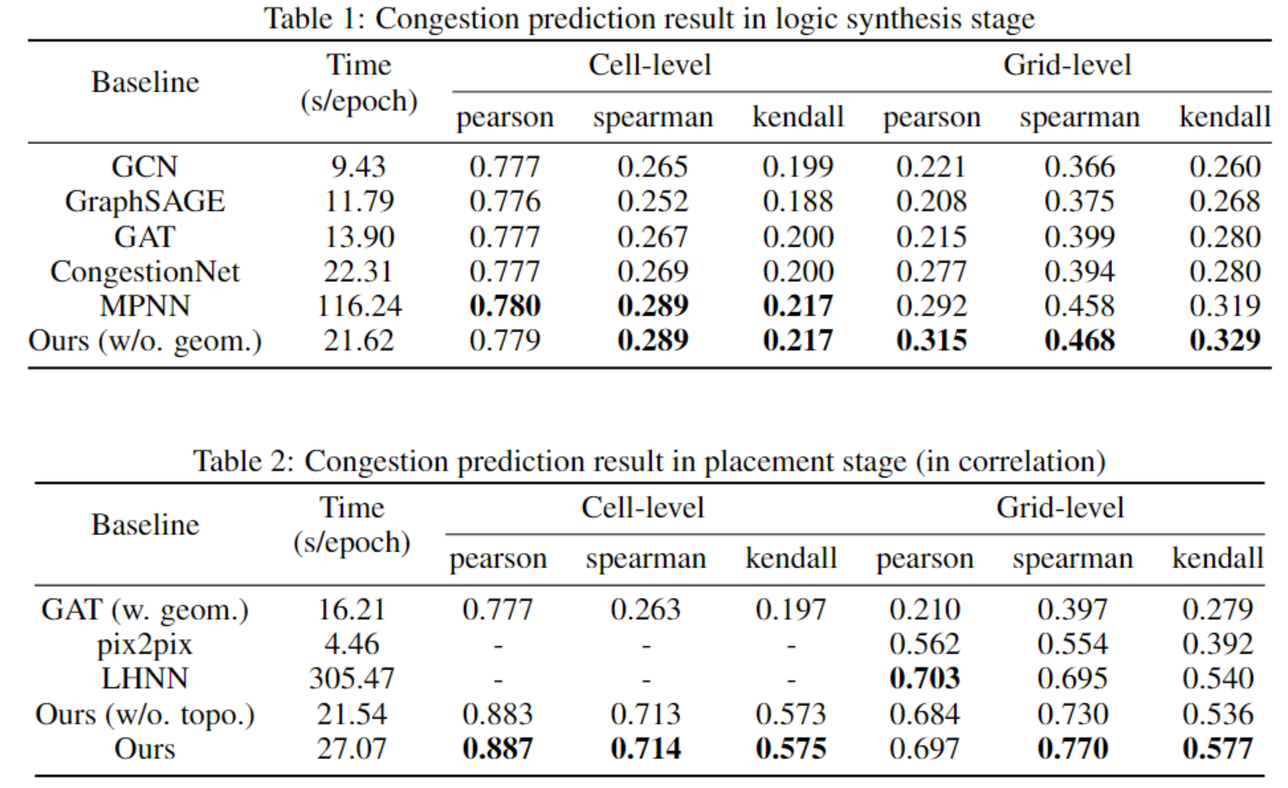


Table 1: Congestion prediction result in logic synthesis stage

Baseline	Time (s/epoch)	Cell-level			Grid-level		
		pearson	spearman	kendall	pearson	spearman	kendall
GCN	9.43	0.777	0.265	0.199	0.221	0.366	0.260
GraphSAGE	11.79	0.776	0.252	0.188	0.208	0.375	0.268
GAT	13.90	0.777	0.267	0.200	0.215	0.399	0.280
CongestionNet	22.31	0.777	0.269	0.200	0.277	0.394	0.280
MPNN	116.24	**0.780**	**0.289**	**0.217**	0.292	0.458	0.319
Ours (w/o. geom.)	21.62	0.779	**0.289**	**0.217**	**0.315**	**0.468**	**0.329**

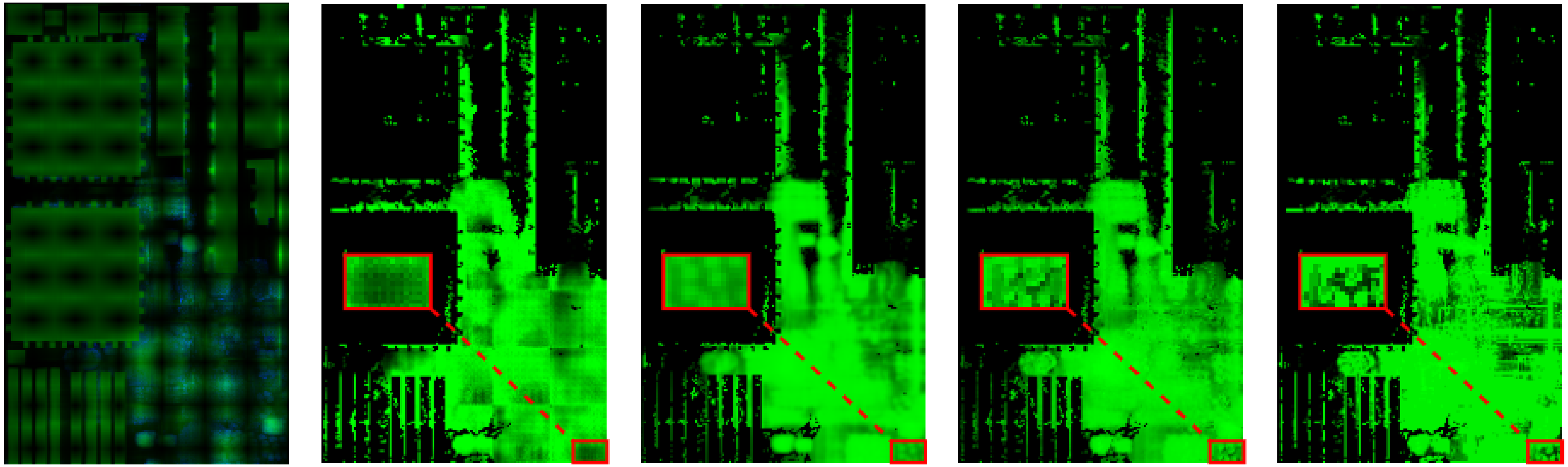
Table 2: Congestion prediction result in placement stage (in correlation)

Baseline	Time (s/epoch)	Cell-level			Grid-level		
		pearson	spearman	kendall	pearson	spearman	kendall
GAT (w. geom.)	16.21	0.777	0.263	0.197	0.210	0.397	0.279
pix2pix	4.46	-	-	-	0.562	0.554	0.392
LHNN	305.47	-	-	-	**0.703**	0.695	0.540
Ours (w/o. topo.)	21.54	0.883	0.713	0.573	0.684	0.730	0.536
Ours	27.07	**0.887**	**0.714**	**0.575**	0.697	**0.770**	**0.577**

Experiments

Congestion prediction results in placement stage (visualized):

- Our solution has clearer decision boundary



(a) Input (b) pix2pix (c) LHNN (d) Ours (e) Ground-truth

Figure 5: Visualization of congestion maps of circuit `ispd2011/superblue19` produced by the baselines.

Experiments

Net wire-length prediction results in placement stage:

- Our solution is also efficient and effective
- Our solution is much better than LHNN, which is especially designed to predict congestion

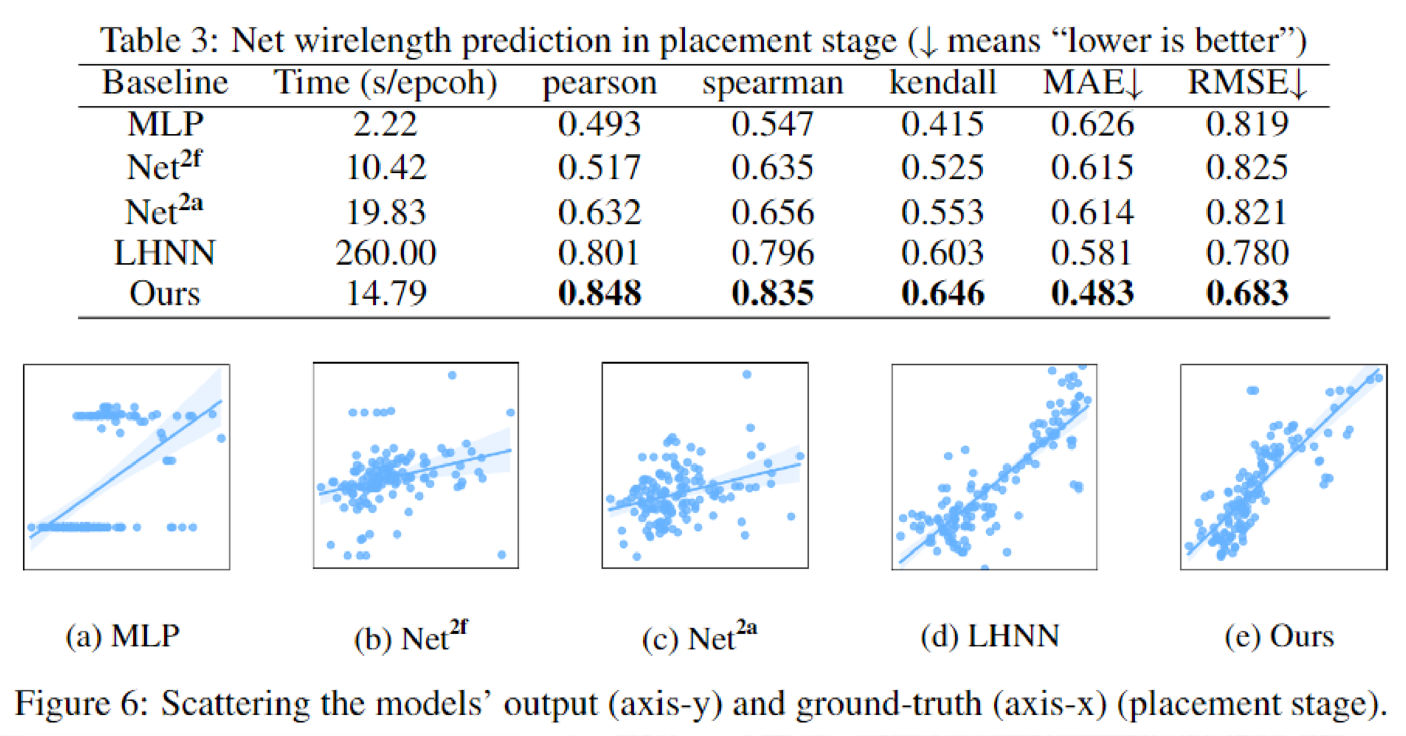


Table 3: Net wirelength prediction in placement stage (↓ means “lower is better”)

Baseline	Time (s/epcoh)	pearson	spearman	kendall	MAE↓	RMSE↓
MLP	2.22	0.493	0.547	0.415	0.626	0.819
Net^{2f}	10.42	0.517	0.635	0.525	0.615	0.825
Net^{2a}	19.83	0.632	0.656	0.553	0.614	0.821
LHNN	260.00	0.801	0.796	0.603	0.581	0.780
Ours	14.79	**0.848**	**0.835**	**0.646**	**0.483**	**0.683**

(a) MLP (b) Net^{2f} (c) Net^{2a} (d) LHNN (e) Ours

Figure 6: Scattering the models’ output (axis-y) and ground-truth (axis-x) (placement stage).

Experiments

Transfer task (congestion -> wire-length):

- LHNN/Ours: trained on wire-length task
- LHNN/Ours(evaluate): trained on congestion task and evaluated on wire-length task
  - the regressor is re-trained, similar for (fine-tune)
- LHNN/Ours(fine-tune): trained on congestion task and fine-tuned on wire-length task

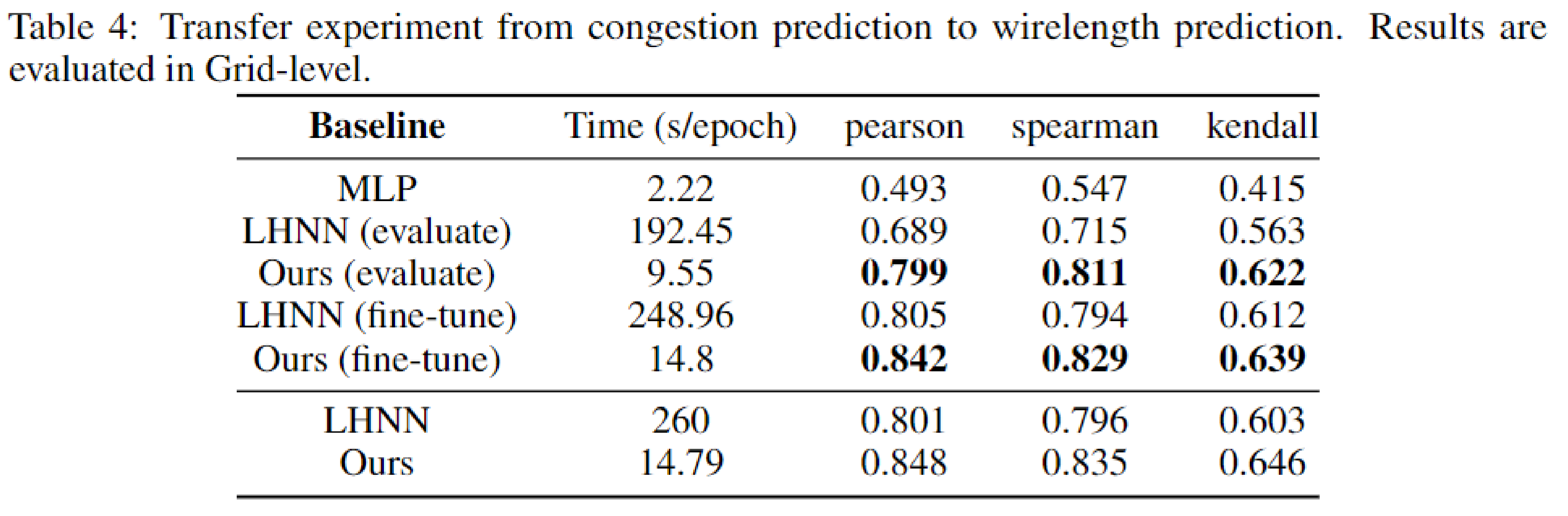


Table 4: Transfer experiment from congestion prediction to wirelength prediction. Results are evaluated in Grid-level.

Baseline	Time (s/epoch)	pearson	spearman	kendall
MLP	2.22	0.493	0.547	0.415
LHNN (evaluate)	192.45	0.689	0.715	0.563
Ours (evaluate)	9.55	**0.799**	**0.811**	**0.622**
LHNN (fine-tune)	248.96	0.805	0.794	0.612
Ours (fine-tune)	14.8	**0.842**	**0.829**	**0.639**
LHNN	260	0.801	0.796	0.603
Ours	14.79	0.848	0.835	0.646

Limitation & Future Work

1. There is still a gap between the novel machine learning algorithms and their application in commercial tools.

2. Our solution is only applicable to netlists (in logic synthesis stage) and layouts (in placement stage), while data-flow graphs or And-Inverter Graphs (AIGs) in other stages are not supported.

Thank you

Shuwen Yang @ 2022