NeuralSteiner: Learning Steiner Tree for Overflow-avoiding Global Routing in Chip Design

Ruizhi Liu[1,2,3], Zhisheng Zeng[1,2,4], Shizhe Ding[1,2], Jingyan Sui[1,2], Xingquan Li[4], Dongbo Bu[1,2,*]

[1]SKLP, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

{liuruizhi19s, dingshizhe19s, suijingyan18b, dbu}@ict.ac.cn

[2]University of Chinese Academy of Sciences, Beijing, China

[3]Beijing Institute of Open Source Chip, Beijing, China

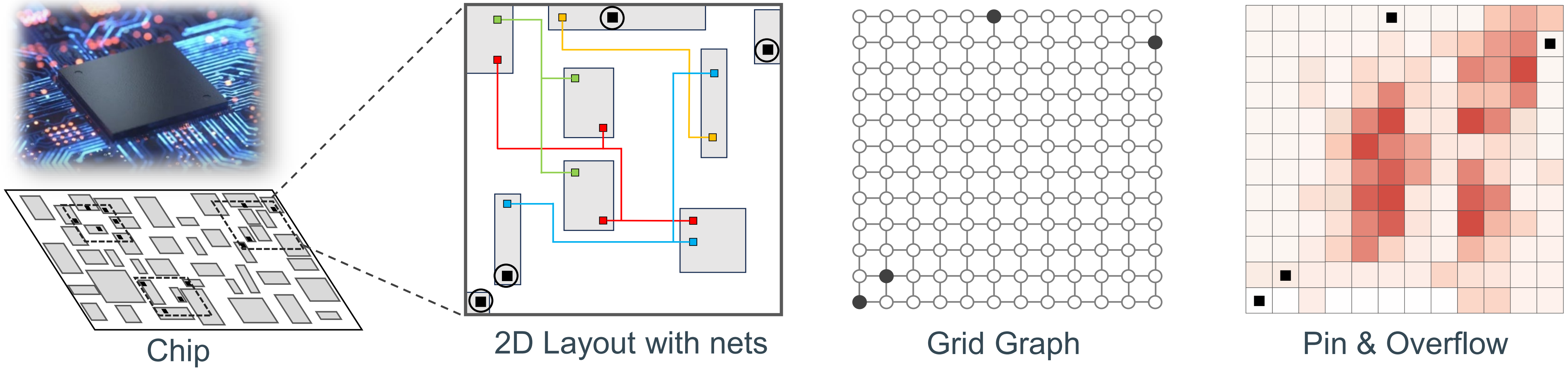
[4]Peng Cheng Laboratory, Shenzhen, Guangdong, China

{zengzhsh, lixq01}@pcl.ac.cn

Outline

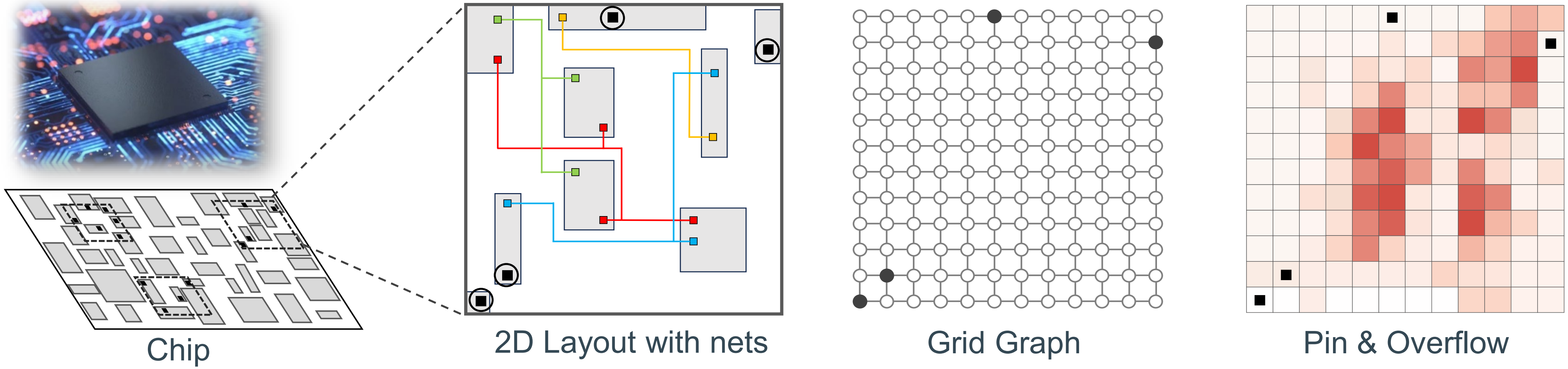
- **Introduction**
- **Learning-based Overflow-avoiding Global Routing**
- **Experimental Evaluation**

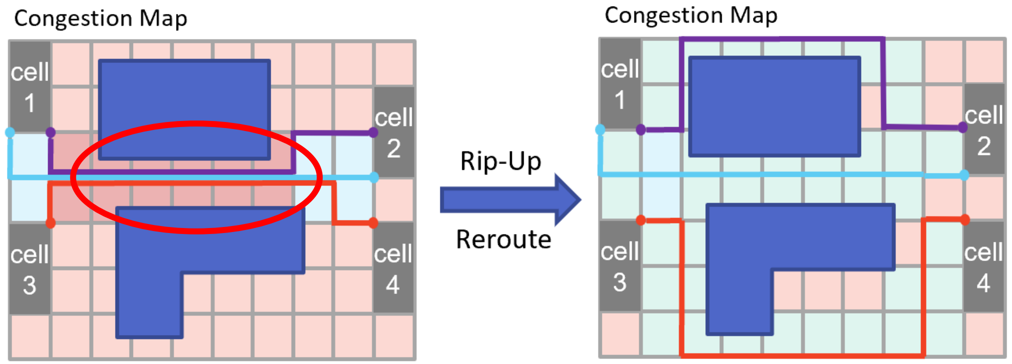
Introduction



In Very Large Scale Integration (VLSI), global routing has become one of the most complex and time-consuming steps in electronic design automation (EDA) to minimize the total wirelength of the routes (usually forming a rectilinear Steiner tree) while avoiding overflow.

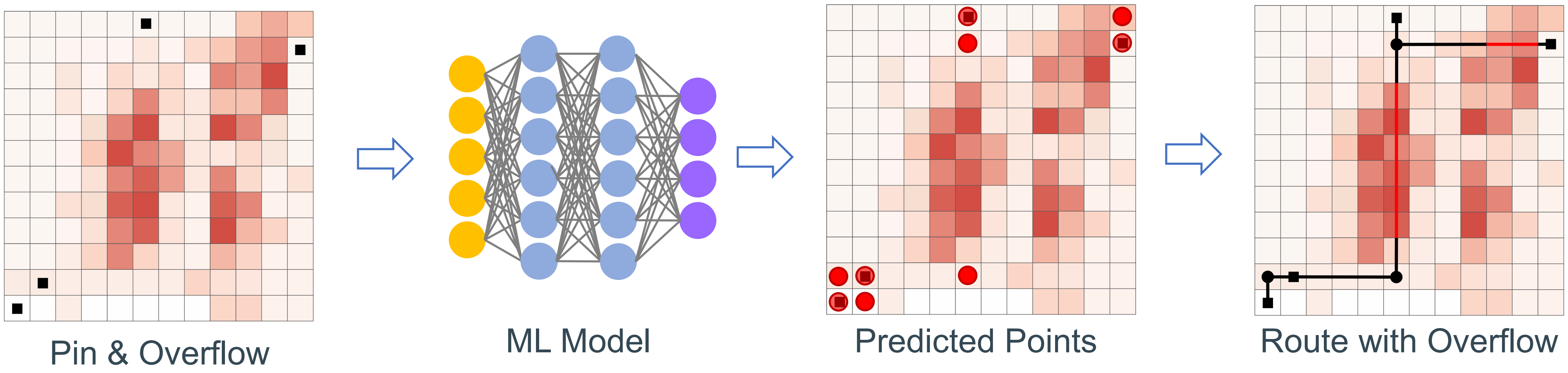
Introduction





Overflow (or congestion) occurs in areas where the number of routes exceeds the capacity.

Introduction

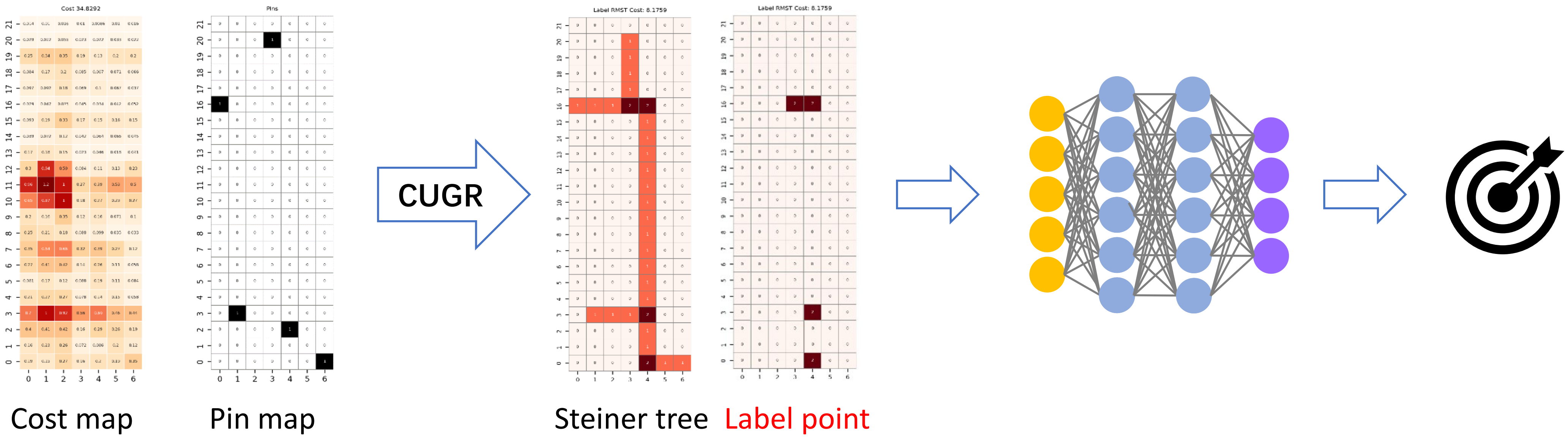


Previous ML-based routing methods mainly focus on correctness and wirelength of net, suffer from high overflow in their routing results.

Outline

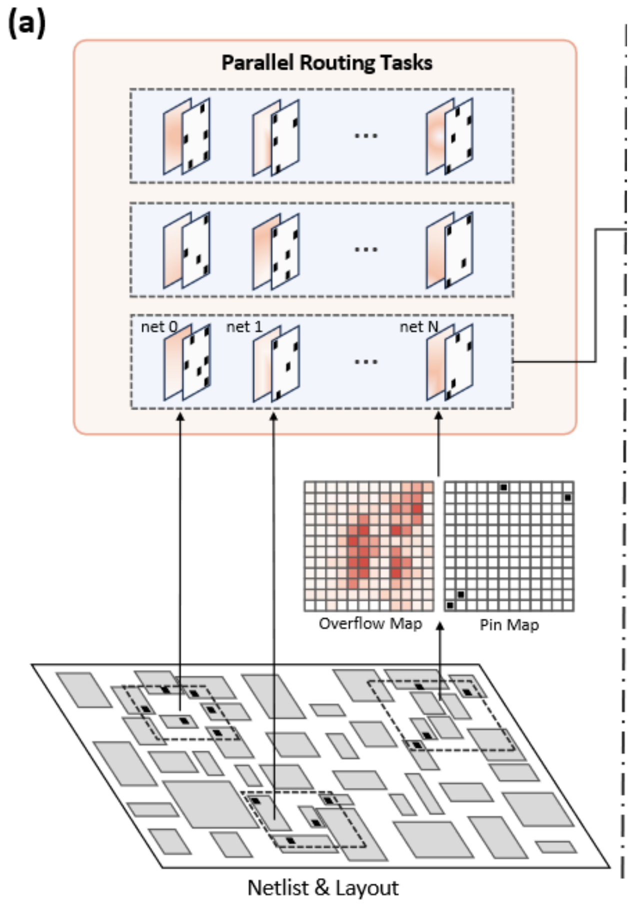
- Introduction
- **Learning-based Overflow-avoiding Global Routing**
- Experimental Evaluation

NeuralSteiner:
Learning-based Overflow-avoiding Global Routing



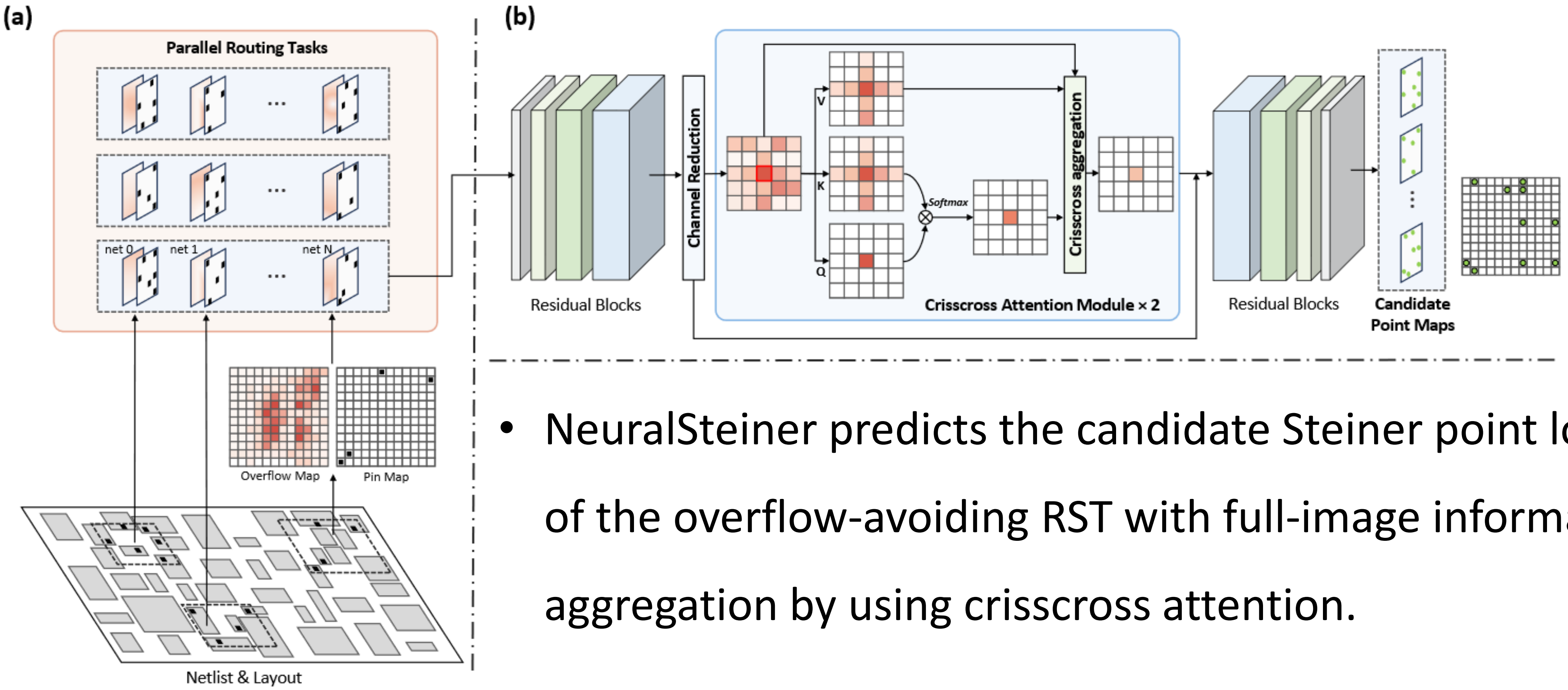
Learning to predict Steiner points of the Steiner tree which can balance the wirelength and overflow well.

NeuralSteiner:
Learning-based Overflow-avoiding Global Routing



- Accelerate routing by grouping the non-overlapping nets into one batch to parallel routing process.

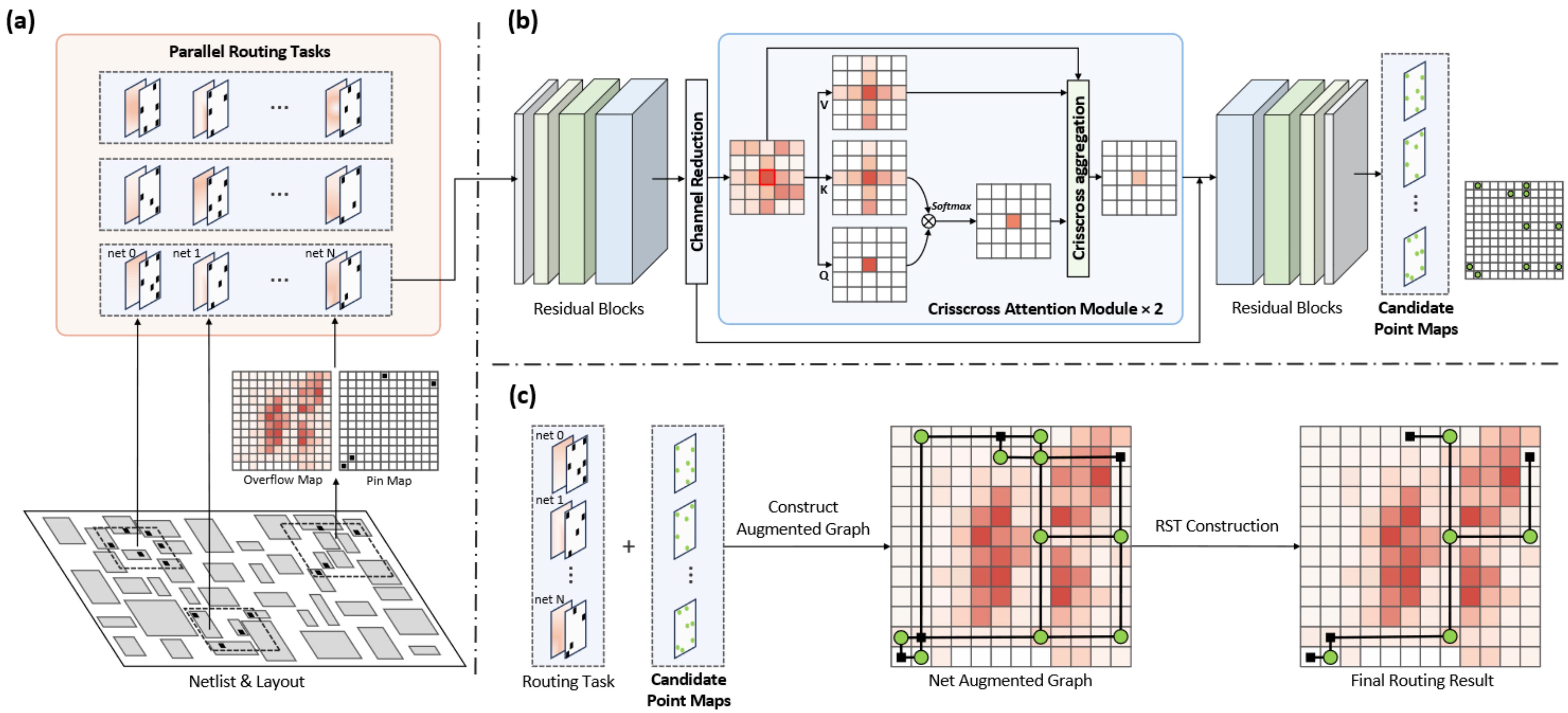
NeuralSteiner:
Learning-based Overflow-avoiding Global Routing



- NeuralSteiner predicts the candidate Steiner point locations of the overflow-avoiding RST with full-image information aggregation by using crisscross attention.

NeuralSteiner: Learning-based Overflow-avoiding Global Routing

NEURAL INFORMATION PROCESSING SYSTEMS



- The post-process algorithm constructs the net augmented graphs based on the predicted candidate points and generates overflow-avoiding RSTs.

Outline

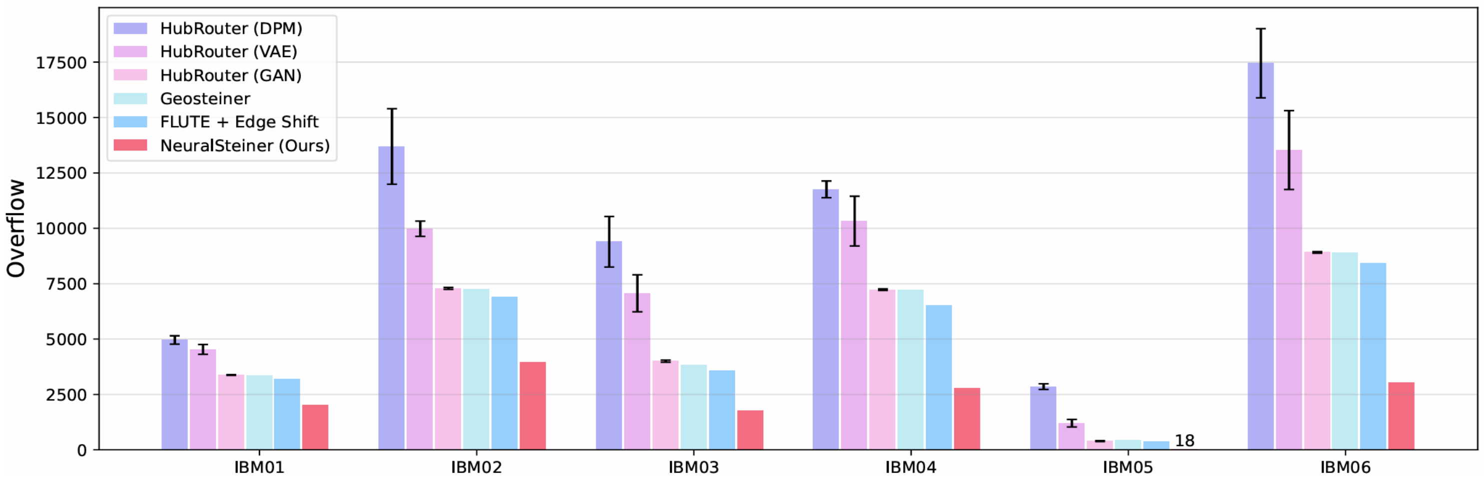
- Introduction
- Learning-based Overflow-avoiding Global Routing
- **Experimental Evaluation**

Experimental Evaluation

NEURAL INFORMATION PROCESSING SYSTEMS

Comparisons with baselines on ISPD98

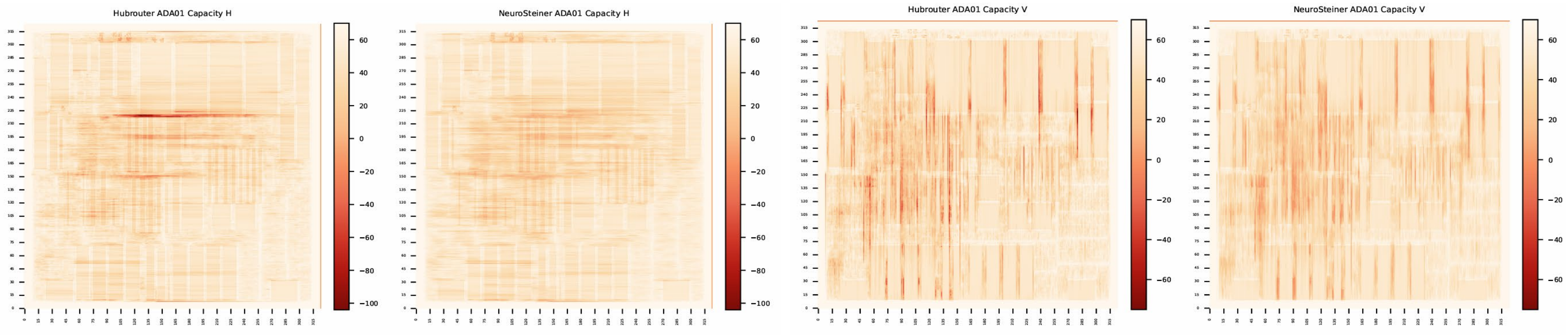
Metric	Model	ibm01	ibm02	ibm03	ibm04	ibm05	ibm06
WL	GeoSteiner	**60142**	**165863**	**145678**	**162734**	**409709**	**275868**
	Boxrouter	62659	171110	146634	167275	410614	277913
	FLUTE+ES	61492	169251	146287	167547	411936	280477
	HR-VAE	64812 ± 1252	176838 ± 6419	161032 ± 3231	179018 ± 4791	440302 ± 4577	301035 ± 5836
	HR-DPM	66575 ± 1394	190142 ± 2511	168550 ± 2103	183051 ± 1946	474463 ± 6674	320423 ± 2958
	HR-GAN	60971 ± 290	167316 ± 578	146893 ± 315	164084 ± 299	411887 ± 4529	277977 ± 514
	NeuralSteiner	61735	170405	148036	166648	415684	283727
Time (Sec)	GeoSteiner	**1.00**	**2.21**	**1.68**	**2.19**	**3.69**	**3.38**
	Boxrouter	5.33	9.76	8.42	31.69	10.75	24.94
	FLUTE+ES	2.90	4.71	5.87	17.16	6.83	13.64
	HR-VAE	8.41 ± 0.03	8.47 ± 0.06	8.59 ± 0.04	10.85 ± 0.04	12.44 ± 0.18	15.83 ± 0.11
	HR-DPM	1701.57 ± 34.19	2589.93 ± 19.63	2669.28 ± 22.77	3593.04 ± 24.10	3995.47 ± 19.57	4305.82 ± 132.85
	HR-GAN	37.40 ± 0.37	41.55 ± 0.51	50.84 ± 2.84	59.94 ± 2.75	69.42 ± 4.03	81.96 ± 3.98
	NeuralSteiner	27.18	34.79	46.24	50.37	75.99	70.32



Experimental Evaluation

Comparisons with baselines on ISPD07

Metric	Method	adaptec01_2d	adaptec02_2d	adaptec03_2d	adaptec04_2d	adaptec05_2d	newblue01_2d	newblue02_2d	newblue03_2d
OF	GeoSteiner	35945	53848	142254	45050	102300	1734	1832	584761
	FLUTE+ES	32518	50947	137104	42306	957704	1348	1713	558047
	HR-GAN	35441	53652	142131	45230	102108	1516	1857	583901
	NeuralSteiner	**82**	**255**	**728**	**97**	**431**	**5**	**35**	**10343**
WL	GeoSteiner	**3389601**	**3209172**	**9330748**	**8865643**	**9784471**	**2320456**	**4595235**	**7371273**
	FLUTE+ES	3418461	3235803	9417934	8896007	9886249	2347941	4651033	7454720
	HR-GAN	3407033	3229110	9355980	8888775	9832110	2339204	4623006	7391055
	NeuralSteiner	3438717	3247429	9459117	9003952	9915795	2365499	4668079	7480679
Time (Sec)	GeoSteiner	**83.17**	**111.92**	**320.08**	**267.13**	**261.43**	**124.68**	**183.82**	**315.48**
	FLUTE+ES	118.48	187.03	396.51	376.72	360.68	169.36	223.55	438.79
	HR-GAN	593.02	780.44	1324.81	1387.01	1384.96	849.34	1221.16	1526.86
	NeuralSteiner	347.20	461.35	1351.91	1138.66	1106.54	390.34	446.68	1225.79



(a) (b) (c) (d)

The Overflow Distribution on different directions after routing by HubRouter (a), (c) and NeuralSteiner (b), (d).

Experimental Evaluation

NEURAL INFORMATION PROCESSING SYSTEMS

Generalization of NeuralSteiner w. CUGR on ISPD18/19

Design	Wire Length ($\times 10^7$)		Via ($\times 10^5$)		Short		Space	
	CUGR	CUGR+NS	CUGR	CUGR+NS	CUGR	CUGR+NS	CUGR	CUGR+NS
ispd18_t5m5	2.878	**2.874**	**9.154**	9.180	389.5	**362.5**	16	**7**
ispd18_t8m5	**6.653**	6.671	22.466	**22.452**	414.6	**412.8**	66	**65**
ispd19_t7	**12.556**	12.621	**40.446**	40.956	2117.6	**2042.3**	7084	**6472**
ispd19_t7m5	**11.273**	11.303	**40.356**	40.613	2368.5	**2219.0**	7715	**6964**
Average	**1**	1.002	**1**	1.005	1	**0.956**	1	**0.809**

When combined with the leading traditional methods CUGR and tested on much larger benchmarks ISPD18/19, NeuralSteiner achieves **4.4%** and **19.1% reduction** on average in **shorts** and **spaces**, with minimal losses in wirelength and vias.

Thanks!

- For more details and results, please refer to the paper: https://openreview.net/pdf?id=oEKFPSOWpp
- Our Code will be released at: https://github.com/liuruizhi96/NeuralSteiner