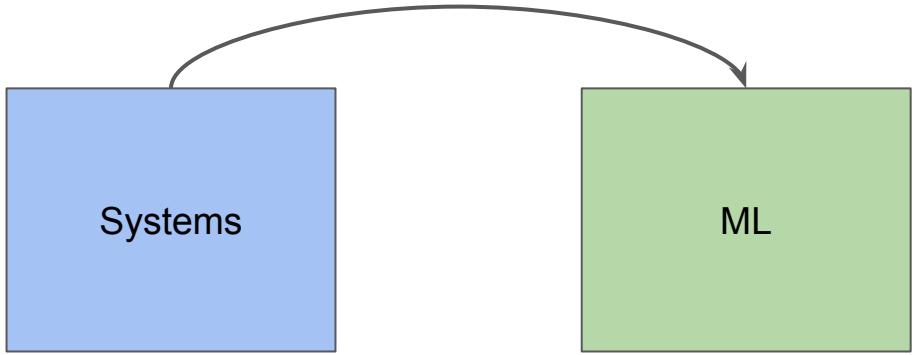
ML for Chip Floorplanning

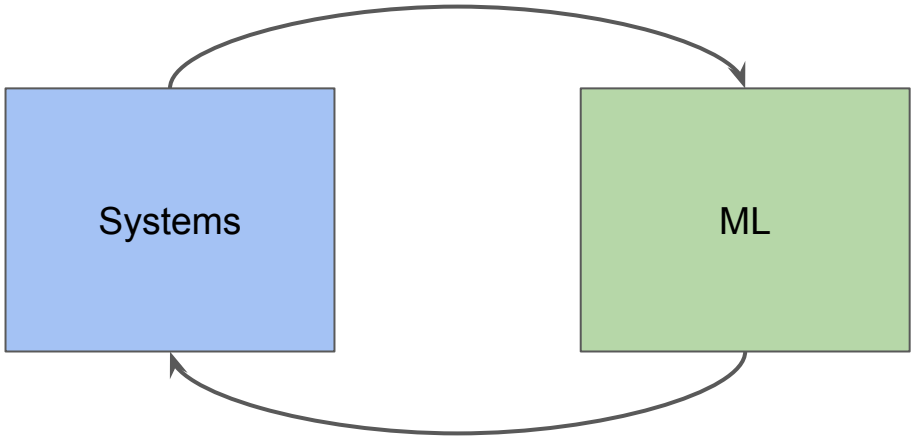
Azalia Mirhoseini & Anna Goldie
Google Research, Brain Team

In the past decade, systems and hardware have transformed ML.



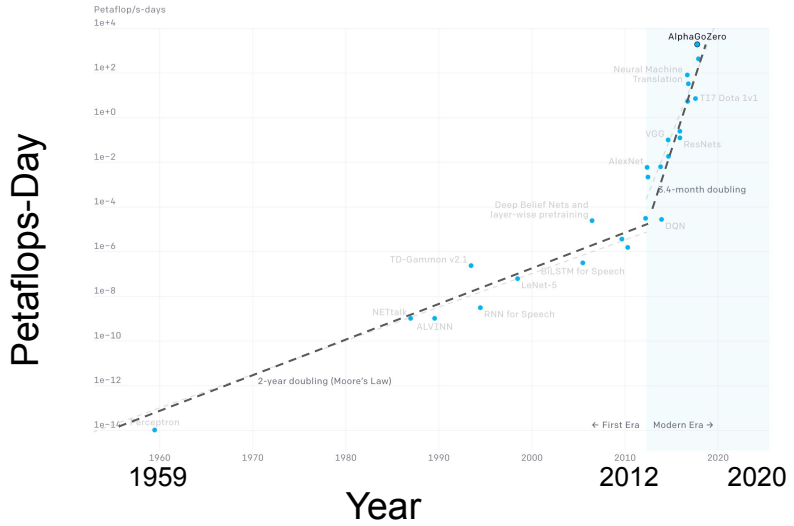
In the past decade, systems and hardware have transformed ML.

Now, it's time for ML to transform systems and hardware.



We need significantly better systems and chips to keep up with the computational demands of AI

- Between 1959 to 2012, compute usage roughly doubled every two years
- Since 2012, the amount of compute used in the largest AI training runs doubled every 3.4 months[1]
- By comparison, Moore's Law had an 18-month doubling period!



1.OpenAI'19

Key Takeaways

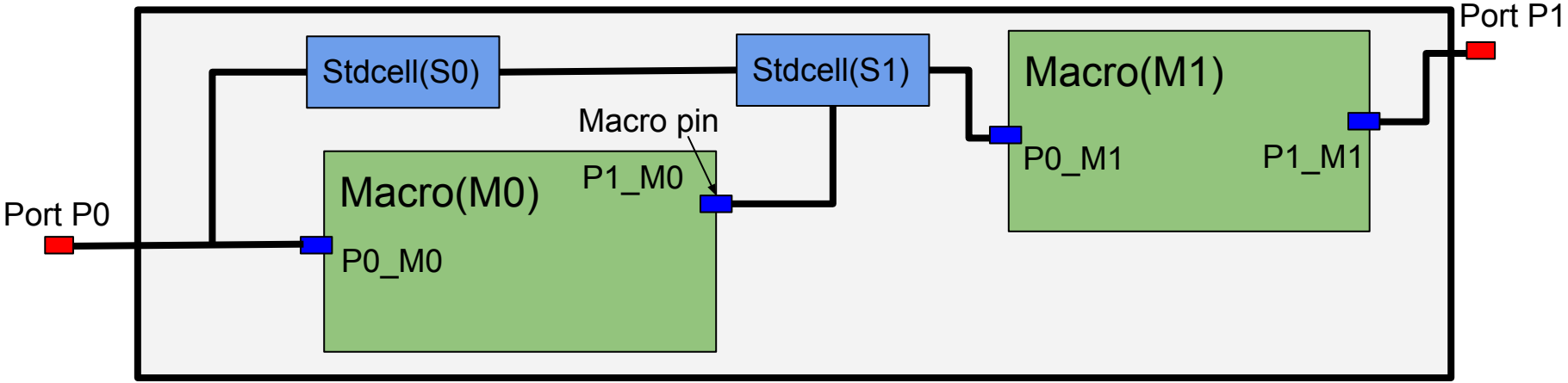
- Deep reinforcement learning method that outperforms/matches human expert performance on chip floorplanning

- Generates placements in under 6 hours, whereas human-expert baselines take weeks or months at a high operation and opportunity cost

- Superhuman chip floorplans generated by this method were used in Google's latest AI accelerator (TPU)!

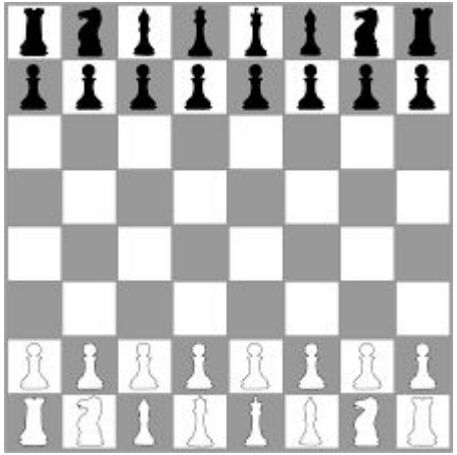
Chip Floorplanning Problem

- A form of graph resource optimization
- Place the chip components to minimize the latency of computation, power consumption, chip area and cost, while adhering to constraints, such as congestion, cell utilization, heat profile, etc.



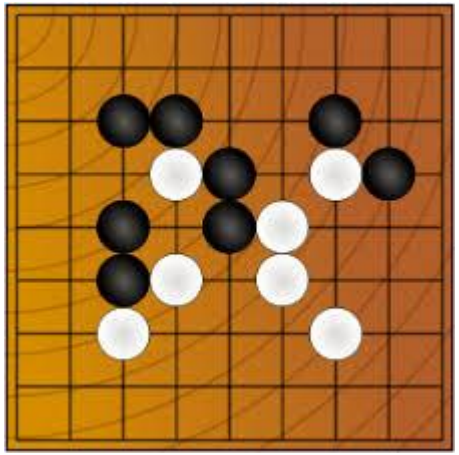
Complexity of Chip Floorplanning Problem

Chess



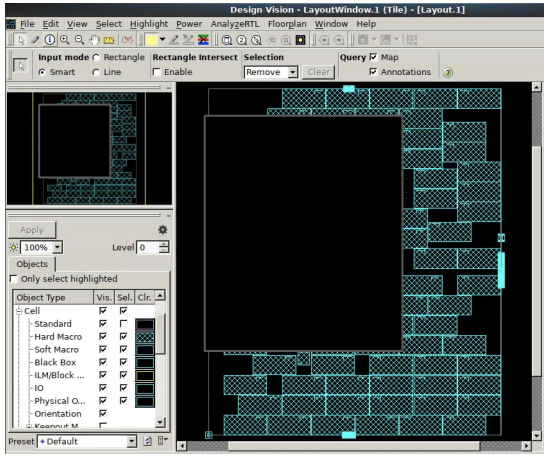
Number of states ~ 10^{123}

Go



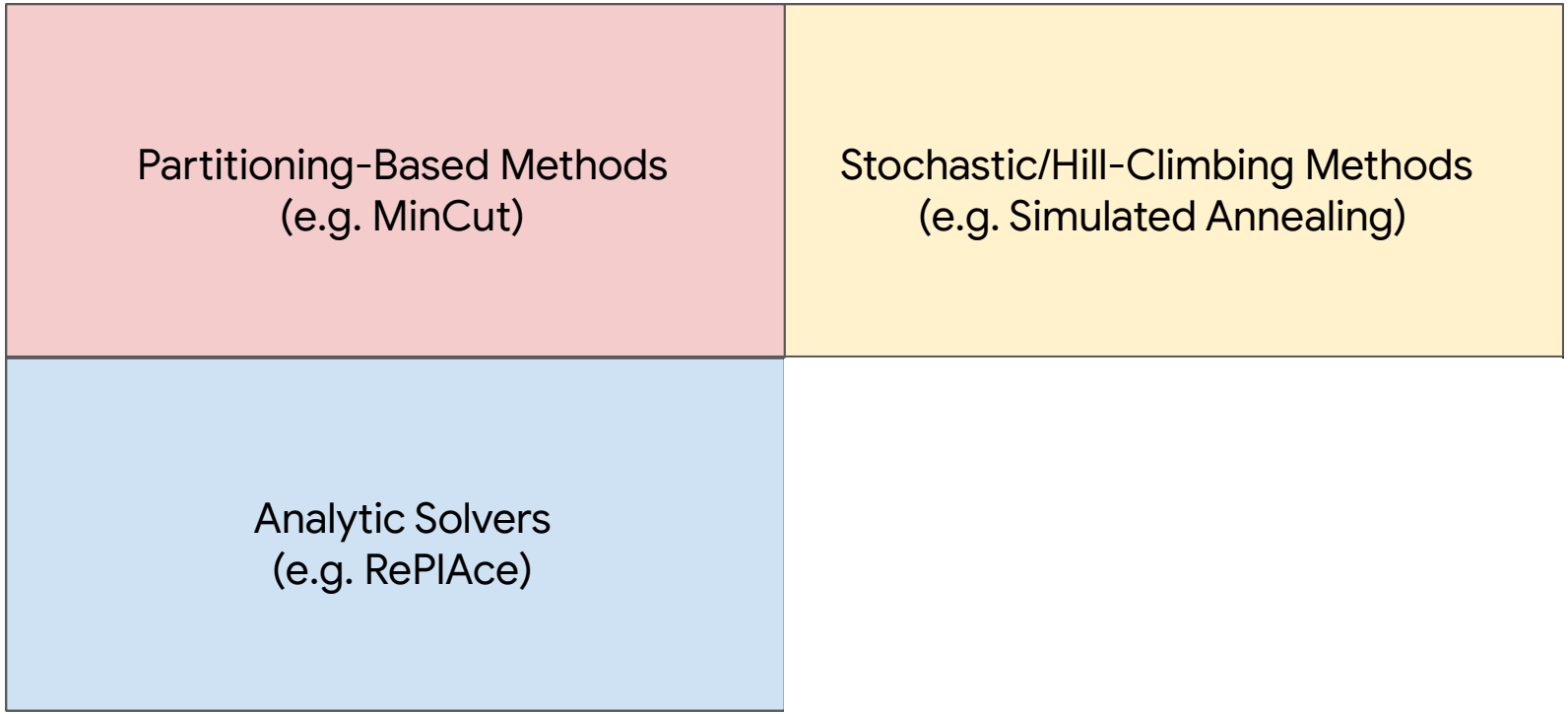
Number of states ~ 10^{360}

Chip Floorplanning

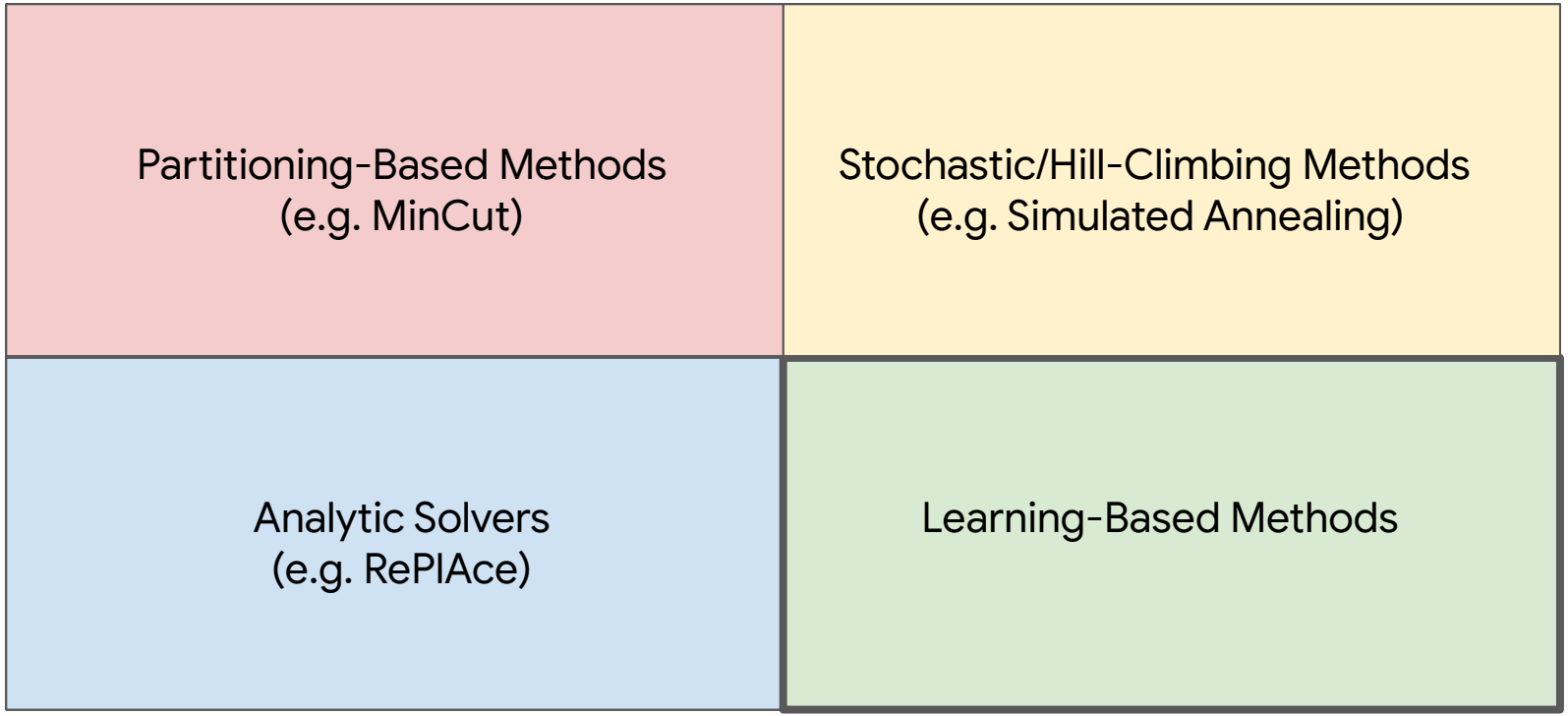


Number of states ~ 10^{9000}

Prior Approaches to Chip Floorplanning



Prior Approaches to Chip Floorplanning

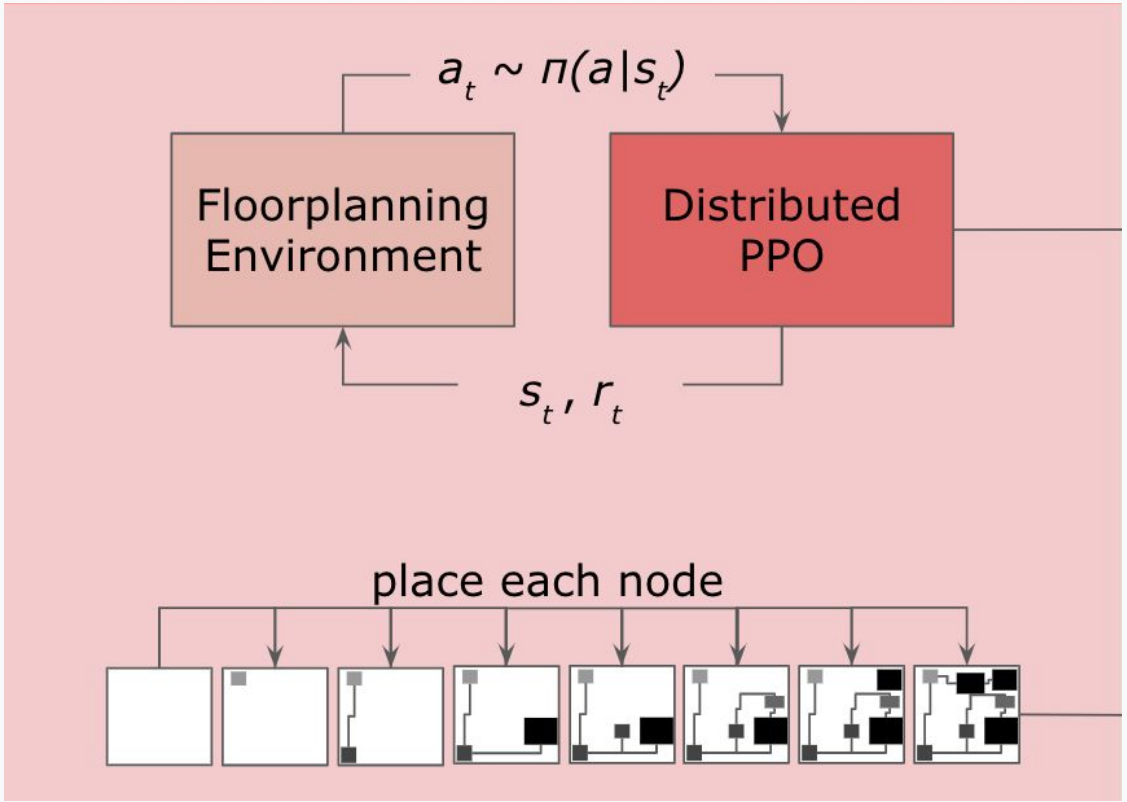


Chip Floorplanning with Reinforcement Learning

State: Graph embedding of chip netlist, embedding of the current node, and the canvas.

Action: Placing the current node onto a grid cell.

Reward: A weighted average of total wirelength, density, and congestion



Our Objective Function

Reward corresponding to placement p of netlist (graph) g

$$J(\theta, G) = \frac{1}{K} \sum_{g \sim G} E_{g, p \sim \pi_\theta}[R_{p,g}]$$

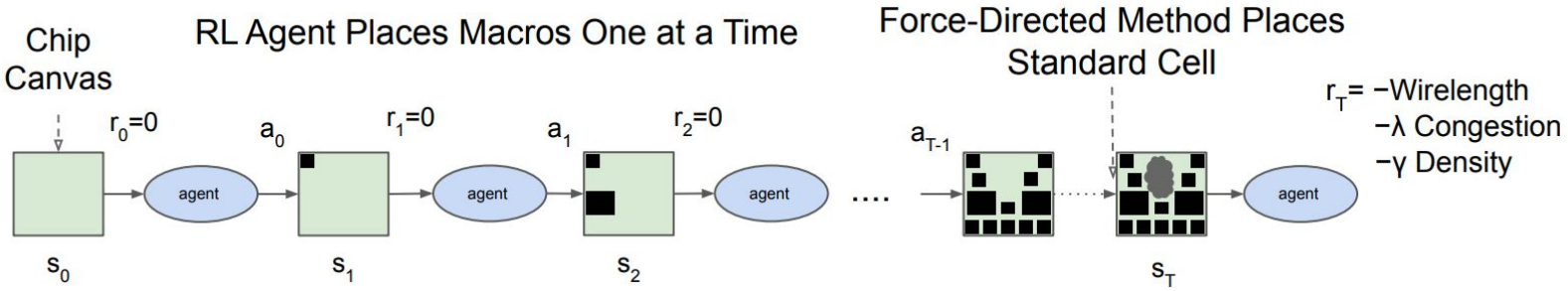
Set of training graphs G

K is size of training set

RL policy parameterized by theta

$$R_{p,g} = -Wirelength(p,g) - \lambda\, Congestion(p,g) - \gamma Density(p,g)$$

We Take a Hybrid Approach to Placement Optimization



Results on a TPU-v4 Block

White area are macros and the green area is composed of standard cell clusters
Our method finds smoother, rounder macro placements to reduce the wirelength

Human Expert

Time taken: **~6-8 weeks**
Total wirelength: 57.07m
Route DRC* violations: 1766

DRC: Design Rule Checking

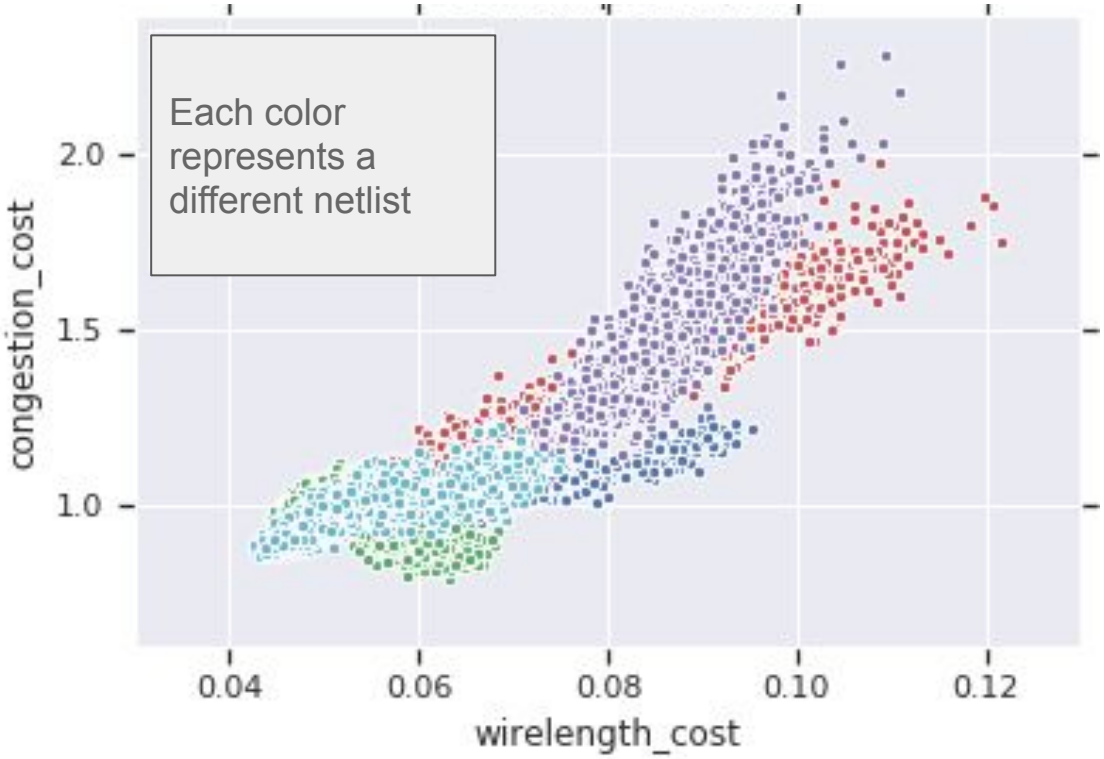
ML Placer

Time taken: **24 hours**
Total wirelength: 55.42m (-2.9% shorter)
Route DRC violations: 1789 (+23 - negligible difference)

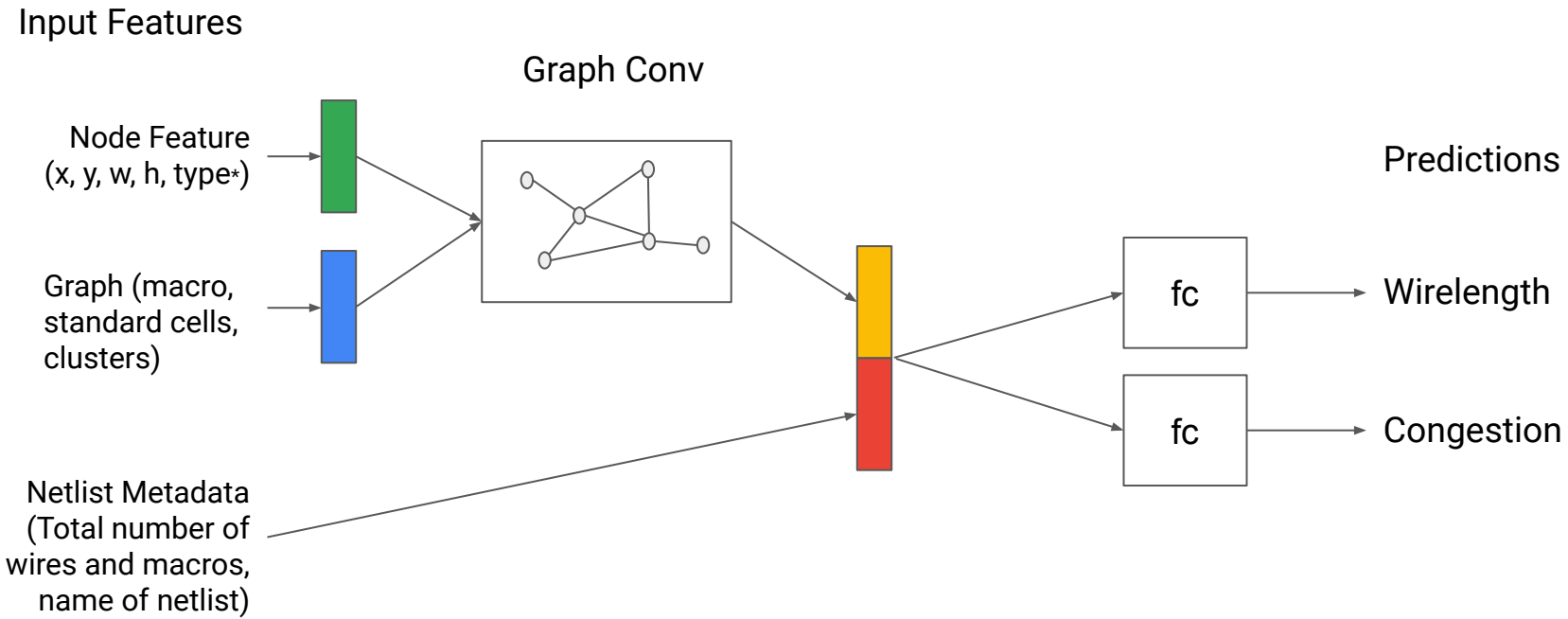
Compiling a Dataset of Chip Placements

To train a more accurate predictor, we generated a dataset of 10k placements

Each placement was labeled with their wirelength and congestion, which were drawn from vanilla RL policies.

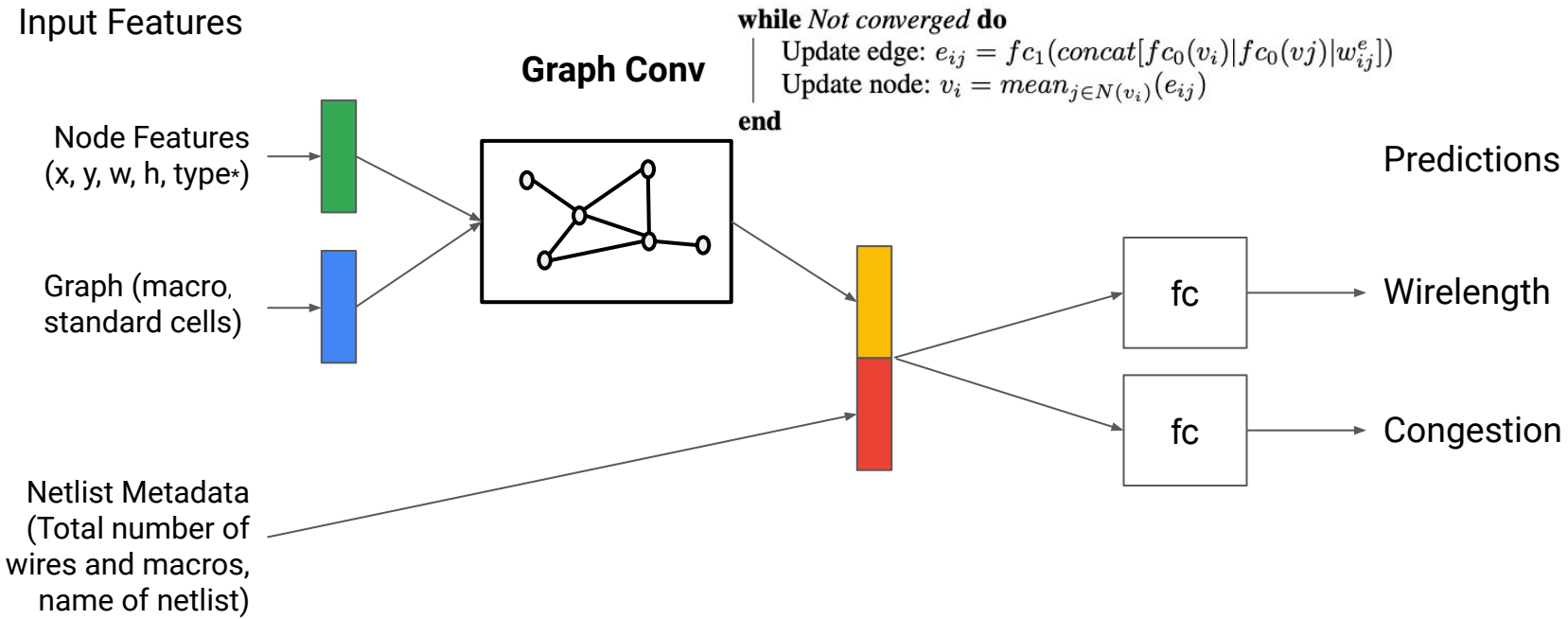


Reward Model Architecture and Features



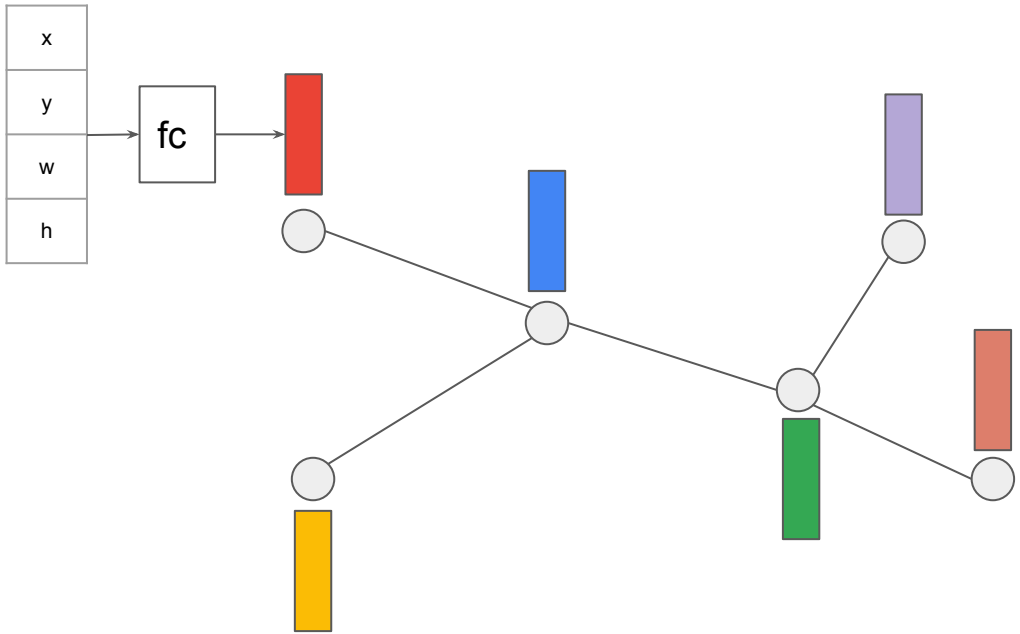
*Node type: One-hot category {Hard macro, soft macro}

Reward Model Architecture and Features

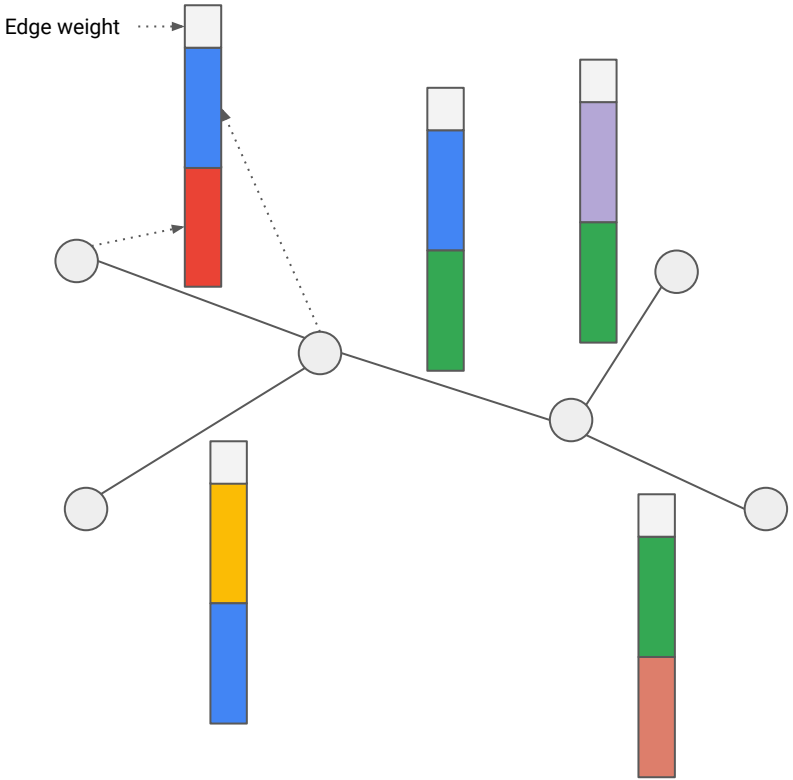


*Node type: One-hot category {Hard macro, soft macro}

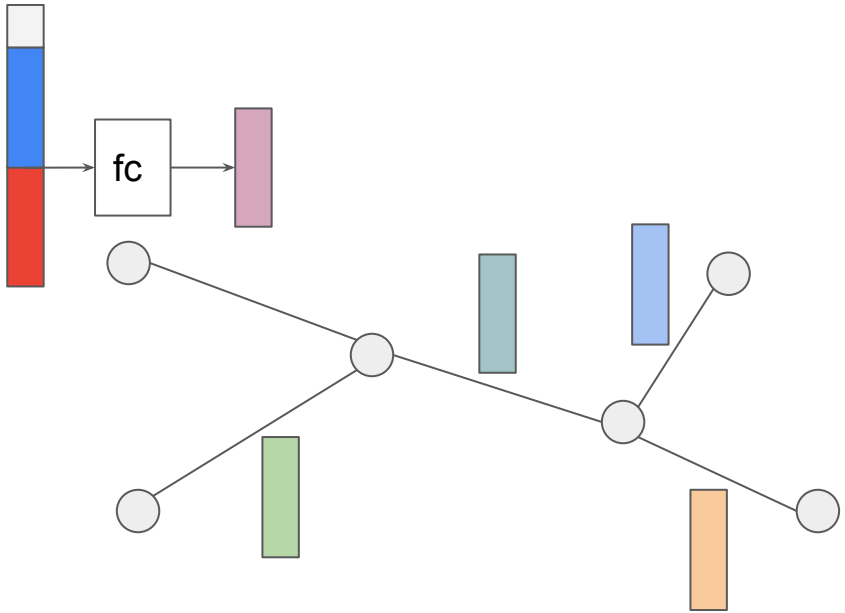
Edge-based Graph Convolution: Node Embeddings



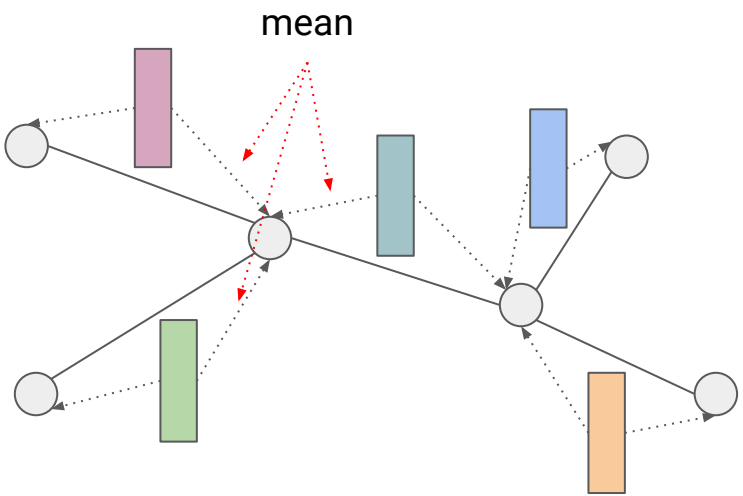
Edge-based Graph Convolution: Edge Embedding



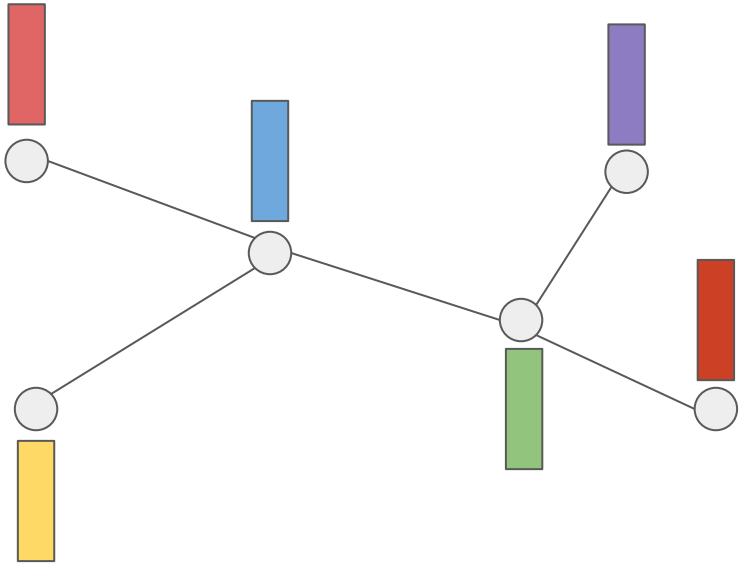
Edge-based Graph Convolution: Edge Embedding



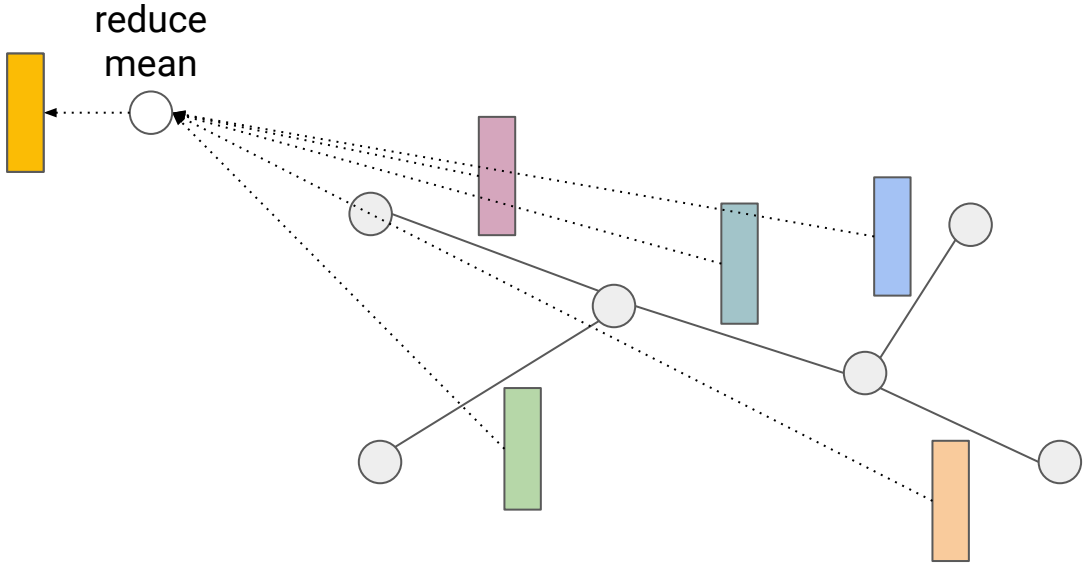
Edge-based Graph Convolution: Propagate



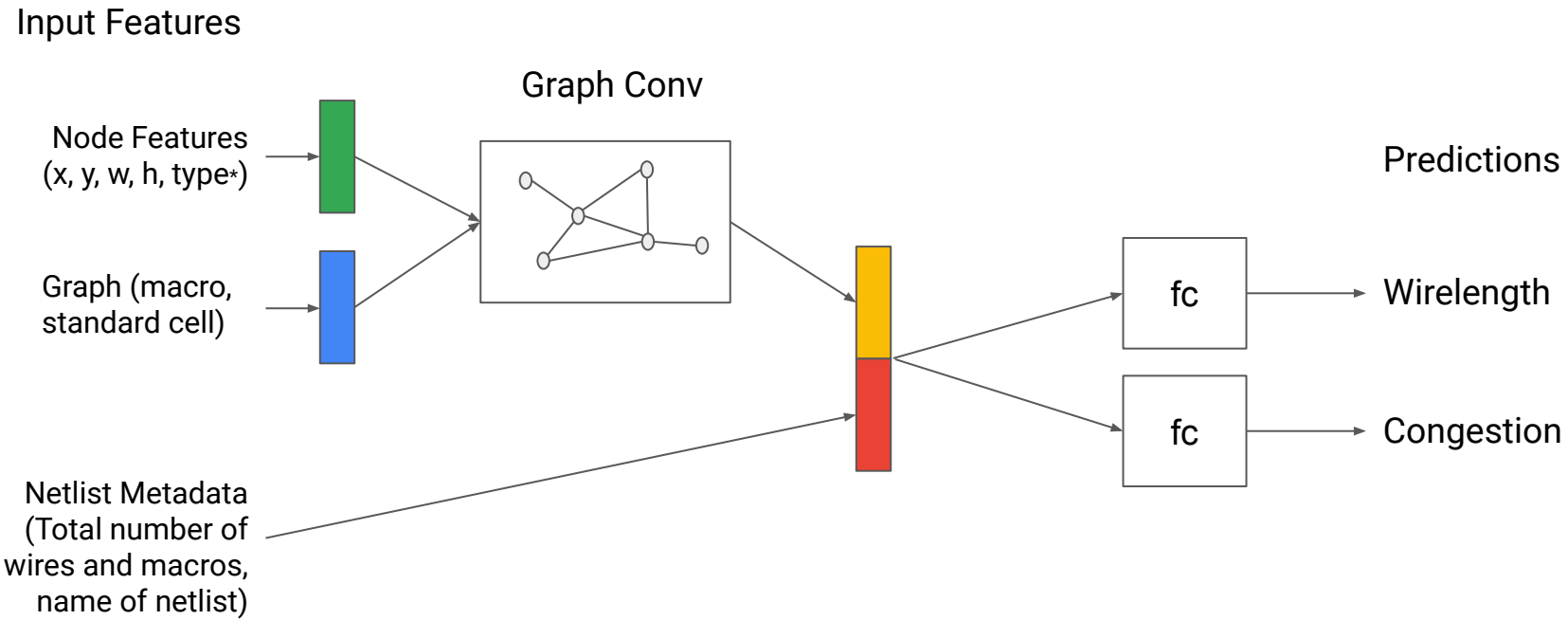
Edge-based Graph Convolution: Repeat



Final Step: Get Graph Embedding

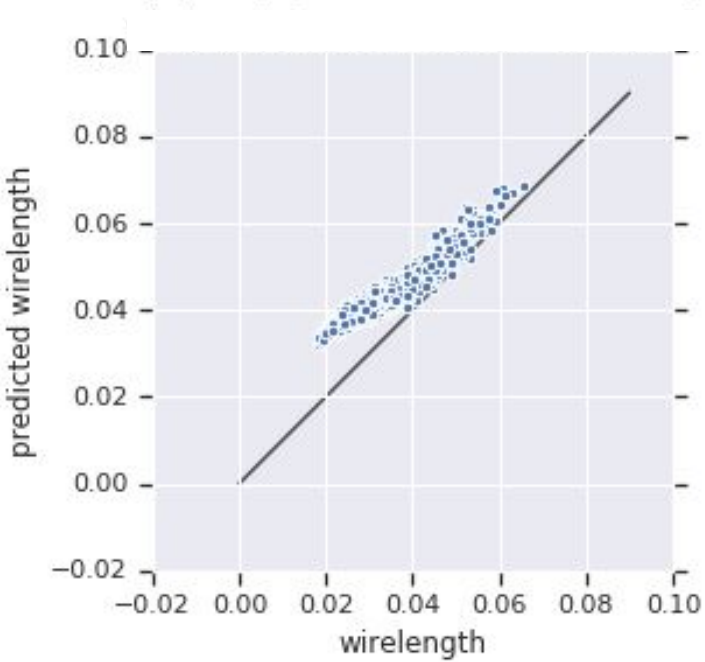


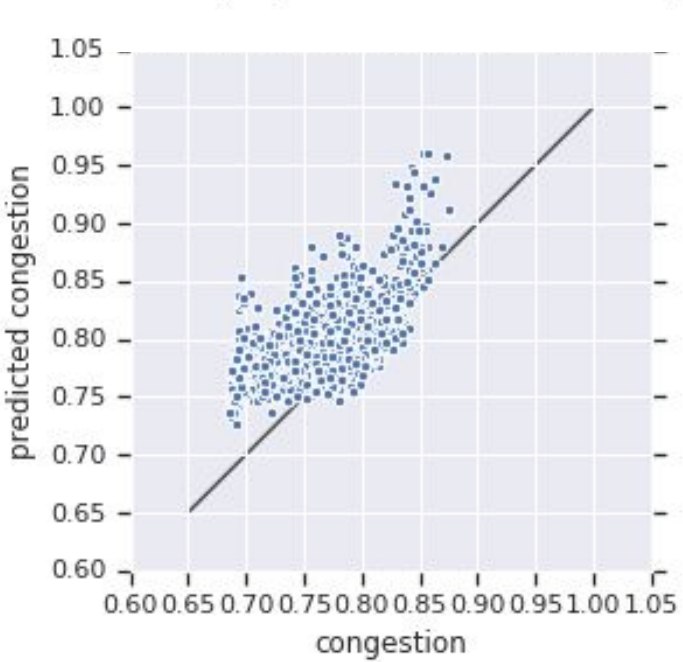
Reward Model Architecture and Features



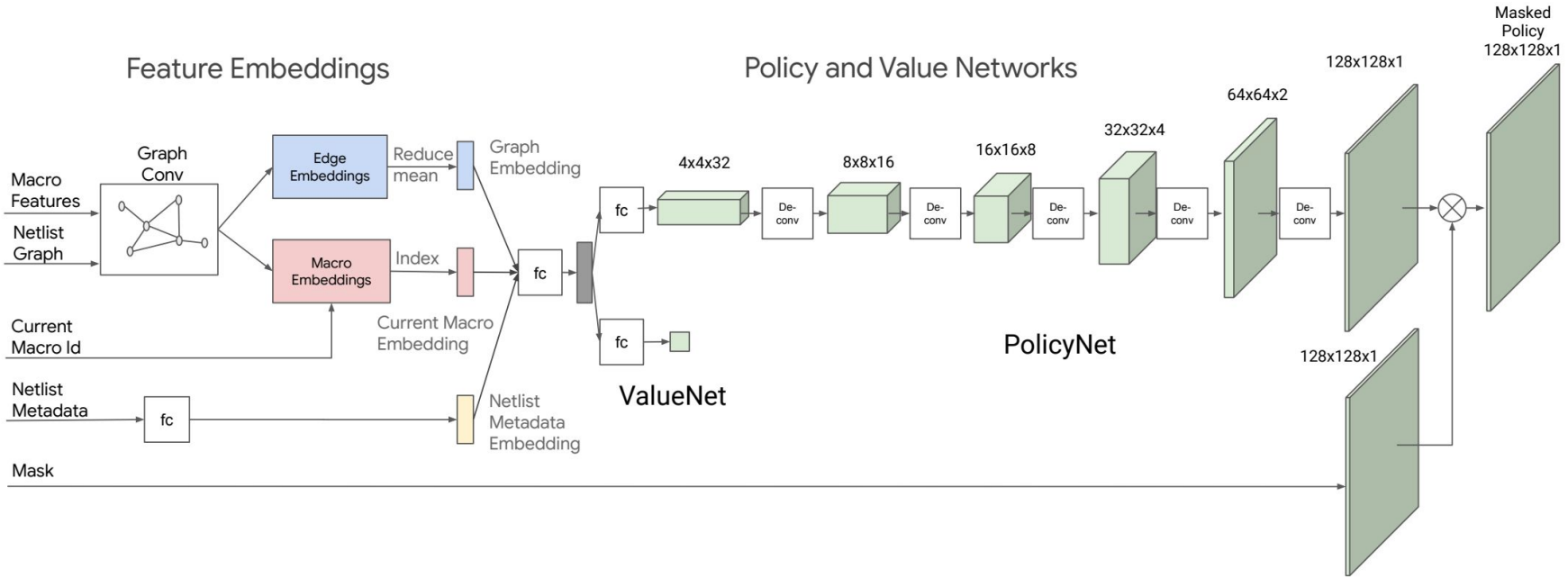
*Node type: One-hot category {Hard macro, soft macro}

Label Prediction Results on Test Chips



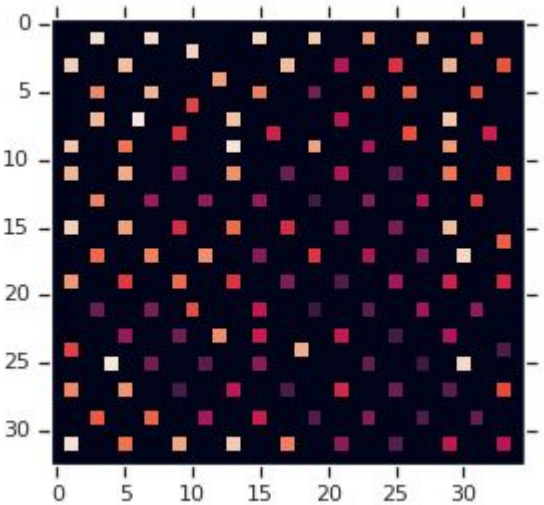


Policy/Value Model Architecture

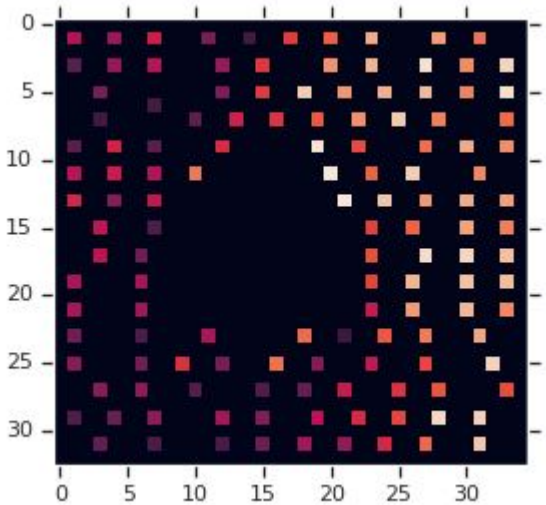


Ariane (RISC-V) Placement Visualization

Training policy from scratch



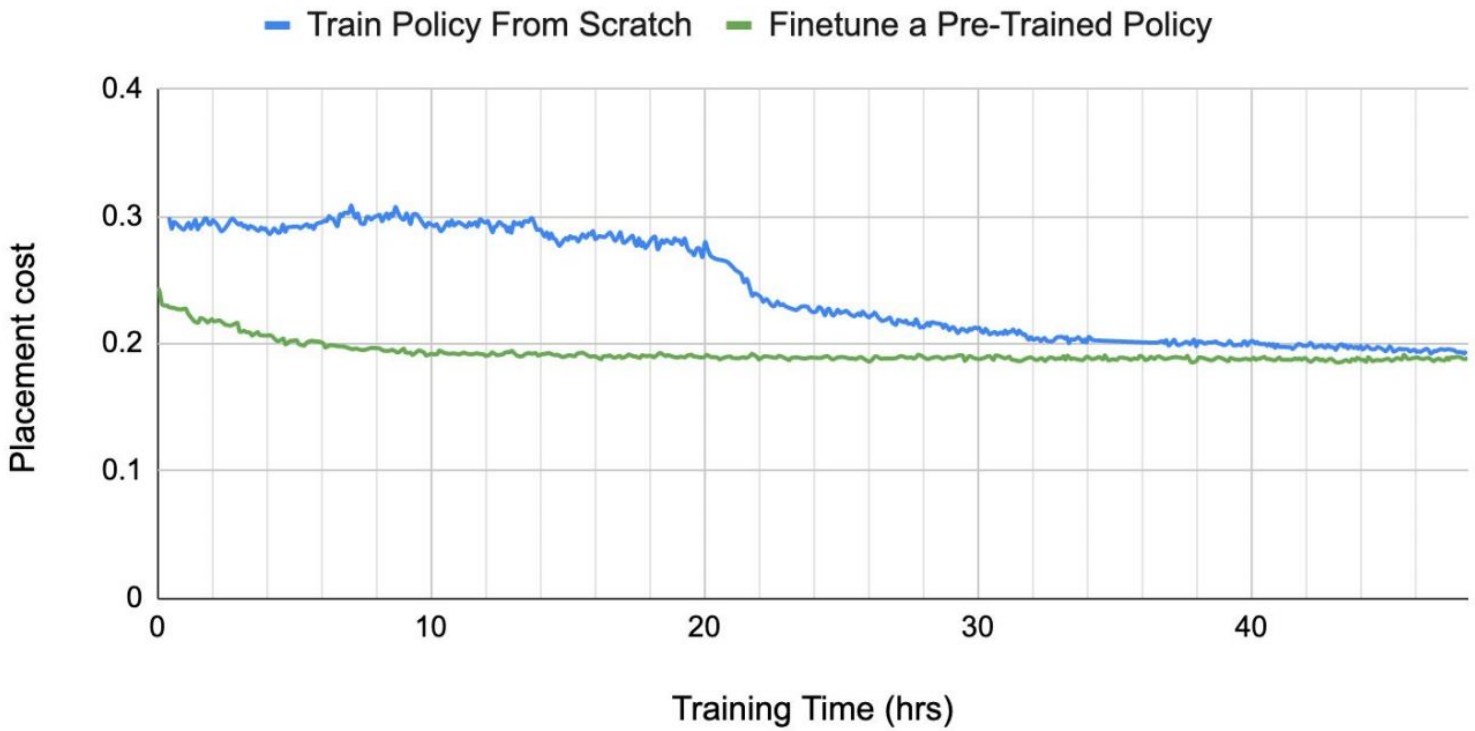
Finetuning a pre-trained policy



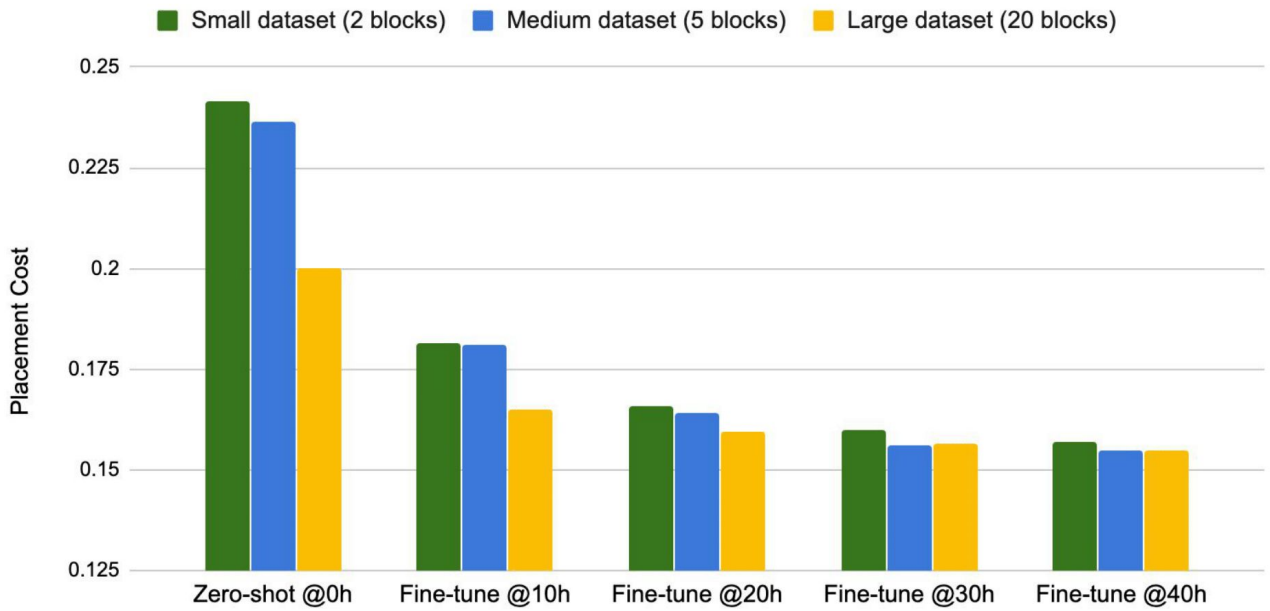
The animation shows the macro placements as the training progresses. Each square shows the center of a macro.

Ariane is an open-source RISC-V processor. See: https://github.com/pulp-platform/ariane

Convergence Curve: Training from Scratch vs. Finetuning



Effects of Training Set Size on Convergence



Comparisons with Manual and SOTA Baselines

Name	Method	Timing		Area	Power	Wirelength	Congestion	
		WNS (ps)	TNS (ns)	Total (μm^2)	Total (W)	(m)	H (%)	V (%)
Block 1	RePlAce	374	233.7	1693139	3.70	52.14	1.82	0.06
	Manual	136	47.6	1680790	3.74	51.12	0.13	0.03
	Ours	84	23.3	1681767	3.59	51.29	0.34	0.03
Block 2	RePlAce	97	6.6	785655	3.52	61.07	1.58	0.06
	Manual	75	98.1	830470	3.56	62.92	0.23	0.04
	Ours	59	170	694757	3.13	59.11	0.45	0.03
Block 3	RePlAce	193	3.9	867390	1.36	18.84	0.19	0.05
	Manual	18	0.2	869779	1.42	20.74	0.22	0.07
	Ours	11	2.2	868101	1.38	20.80	0.04	0.04
Block 4	RePlAce	58	11.2	944211	2.21	27.37	0.03	0.03
	Manual	58	17.9	947766	2.17	29.16	0.00	0.01
	Ours	52	0.7	942867	2.21	28.50	0.03	0.02
Block 5	RePlAce	156	254.6	1477283	3.24	31.83	0.04	0.03
	Manual	107	97.2	1480881	3.23	37.99	0.00	0.01
	Ours	68	141.0	1472302	3.28	36.59	0.01	0.03

- We freeze the macro placements generated by each method and report the place opt results by the commercial EDA.
- RePlAce: C. Cheng, A. B. Kahng, I. Kang and L. Wang, "RePlAce: Advancing Solution Quality and Routability Validation in Global Placement," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2018

IEEE SPECTRUM

Google Invents AI That Learns a Key Part of Chip Design

AI helps designs AI chip that might help an AI design future AI chips

By **Samuel K. Moore**

MIT Technology Review

Google is using AI to design chips that will accelerate AI

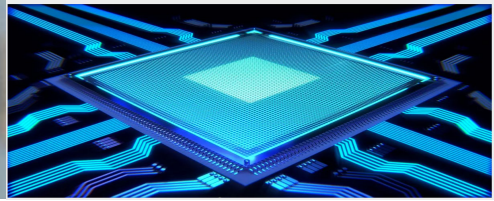
PC GAMER

Google is using AI to design AI processors much faster than humans can

By Paul Lilly 10 days ago

Chips making chips.

COMMENTS



Google Proposes AI as Solution for Speedier AI Chip Design

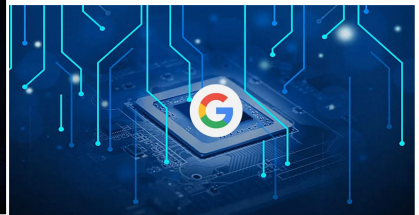
Google uses artificial intelligence to optimize AI chip production

By **Mario McKellop** - April 2, 2020 140 0

Google Hoping The Next AI Chips Will Be Designed By AI

Company researchers have come up with an AI system that can design other AI chips. The goal is to help improve AI with the help of AI.

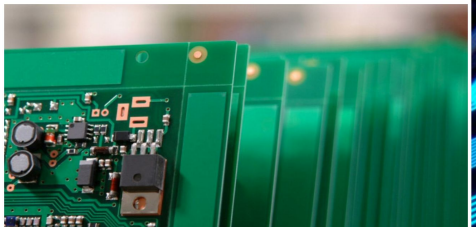
BY ASWIN NARAYANAN
MAR 31, 2020



Google Researchers Create AI-ception with an AI Chip That Speeds Up AI

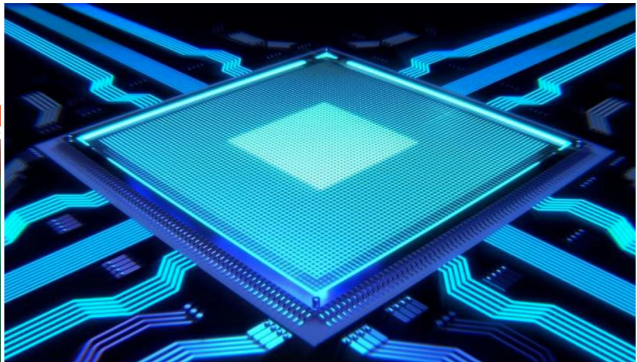
Using a reinforcement-learning algorithm, the AI has learnt to optimize the placement of components on a computer chip.

By Fabienne Lang
March 30, 2020



Google trains chips to design themselves

by Peter Grad , Tech Xplore



This work was a collaboration between Google Brain and the TPU team!

I wanted to end this talk by thanking all of my collaborators and mentors who helped make this possible!

Mustafa Yazgan, Joe Jiang, Ebrahim Songhori, Shen Wang, Young-Joon Lee, Eric Johnson, Omkar Pathak, Azade Nazi, Jiwoo Pak, Andy Tong, Kavya Srinivasa, William Hang, Emre Tuncer, Quoc V Le, James Laudon, Richard Ho, Roger Carpenter, Jeff Dean

Google AI Blog: https://ai.googleblog.com/2020/04/chip-design-with-deep-reinforcement.html

Paper: Chip Placement with Deep Reinforcement Learning

Other work on placement/combinatorial optimization with ML by our group:

1. Placement Optimization with Deep Reinforcement Learning, Anna Goldie, Azalia Mirhoseini, ISPD, 2020.
2. GAP: Generalizable Approximate Graph Partitioning Framework. Azade Nazi, Will Hang, Anna Goldie, Sujith Ravi, Azalia Mirhoseini, ICLR workshop on graph representation, 2019.
3. Generalized Clustering by Learning to Optimize Expected Normalized Cuts, Azade Nazi, Will Hang, Anna Goldie, Sujith Ravi, Azalia Mirhoseini, Neurips workshop on sets and partitions, 2019.
4. A Reinforcement Learning Driven Heuristic Optimization Framework, Qingpeng Cai, Will Hang, Azalia Mirhoseini, George Tucker, Jingtao Wang, Wei Wei, KDD workshop on deep reinforcement learning for knowledge discovery, 2019.
5. GDP: generalized device placement for dataflow graphs, Yanqi Zhou, Sudip Roy, Amirali Abdolrashidi, Daniel Wong, Peter C. Ma, Qiumin Xu Ming Zhong, Hanxiao Liu, Anna Goldie, Azalia Mirhoseini, James Laudon, 2019.
6. A Hierarchical Model for Device Placement, Azalia Mirhoseini*, Anna Goldie*, Hieu Pham, Benoit Steiner, Quoc V. Le and Jeff Dean, ICLR, 2018.
7. Device Placement Optimization with Reinforcement Learning, Azalia Mirhoseini*, Hieu Pham*, Quoc V. Le, Benoit Steiner, Rasmus Larsen, Yuefeng Zhou, Naveen Kumar, Mohammad Norouzi, Samy Bengio, Jeff Dean, ICML, 2017.

Thank you!