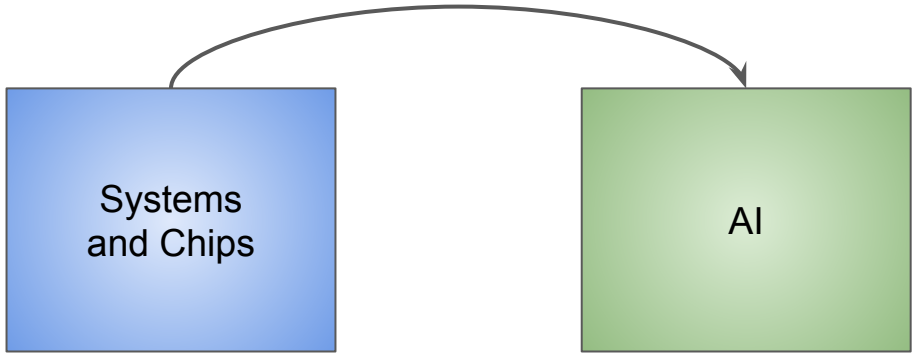
AI for AI Systems and Chips

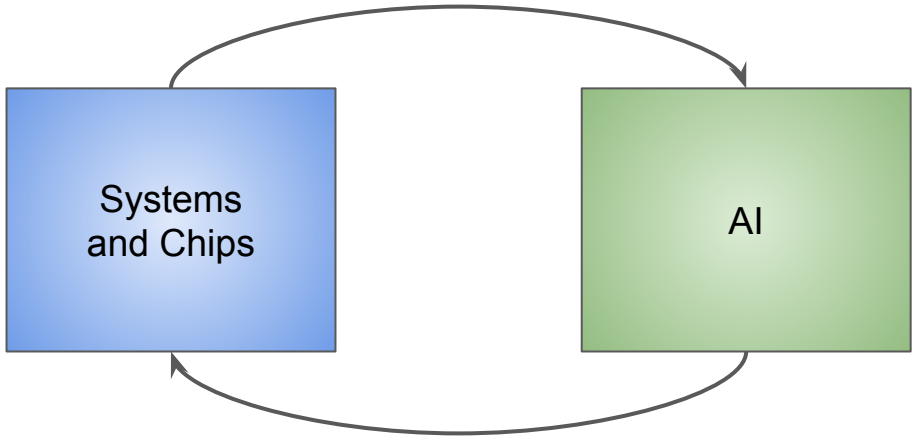
Azalia Mirhoseini
Senior Research Scientist, Google Brain

In the past decade, systems and chips have transformed AI.



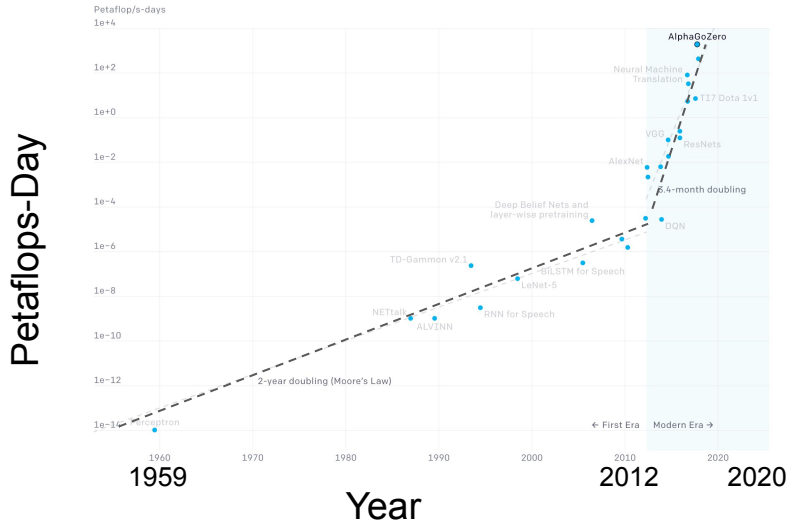
In the past decade, systems and chips have transformed AI.

Now, it's time for AI to transform the way systems and chips are made.



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

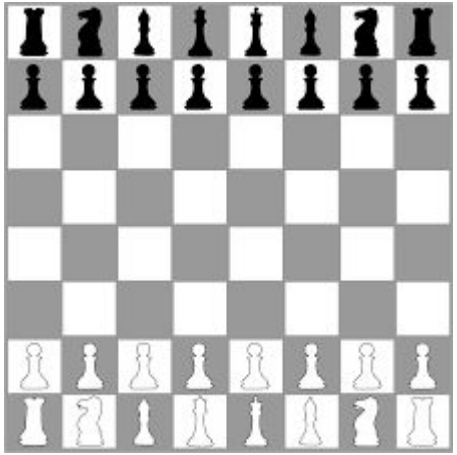
- Between 1959 to 2012, there was a 2-year doubling time for the compute used in historical AI results.
- Since 2012, the amount of compute used in the largest AI training runs doubled every 3.4 months.[1]
- By comparison, Moore's Law had an 18-month doubling period!



1.OpenAI'19

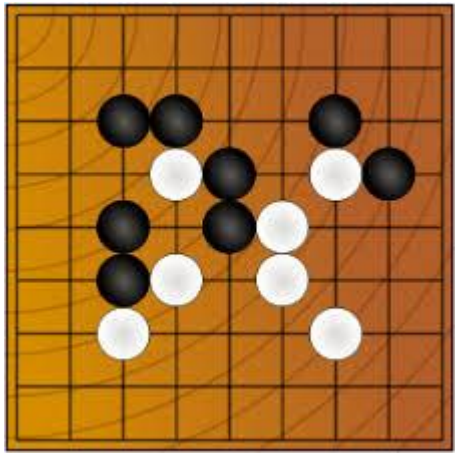
Chip design is a really complex problem and AI can help

Chess



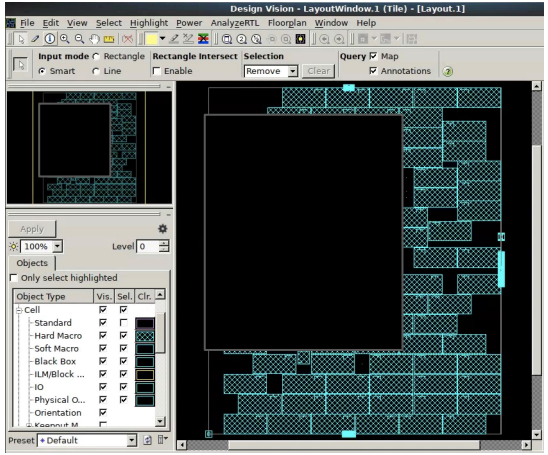
Number of states ~ 10^{123}

Go



Number of states ~ 10^{360}

Chip Placement

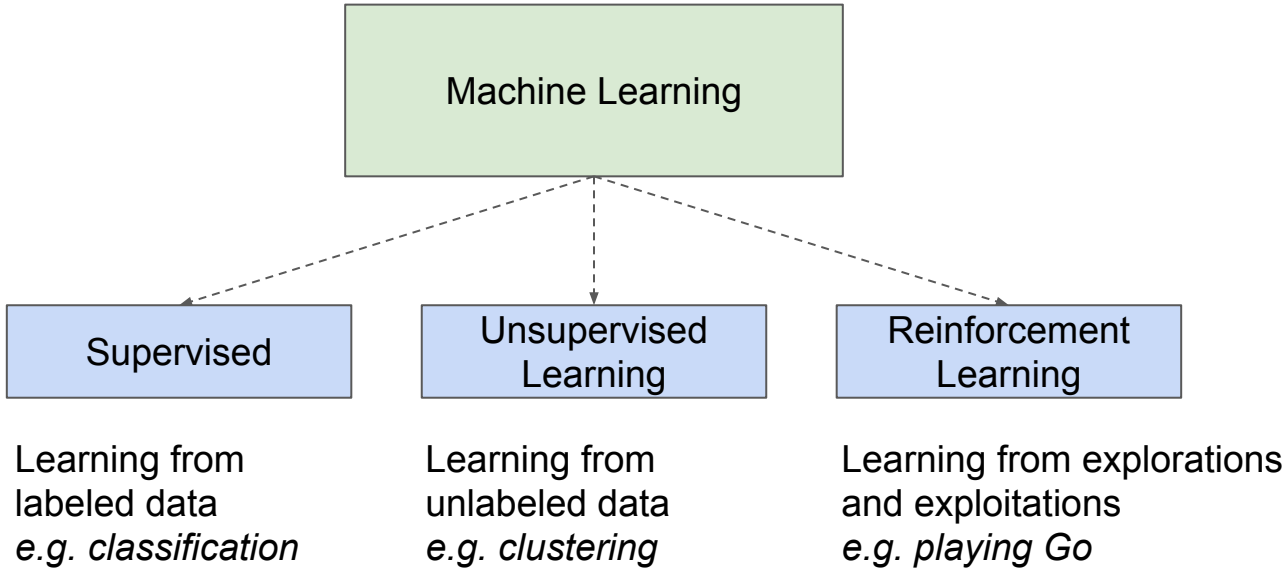


Number of states ~ 10^{9000}

This talk

Twowork on ML for Systems/Chips

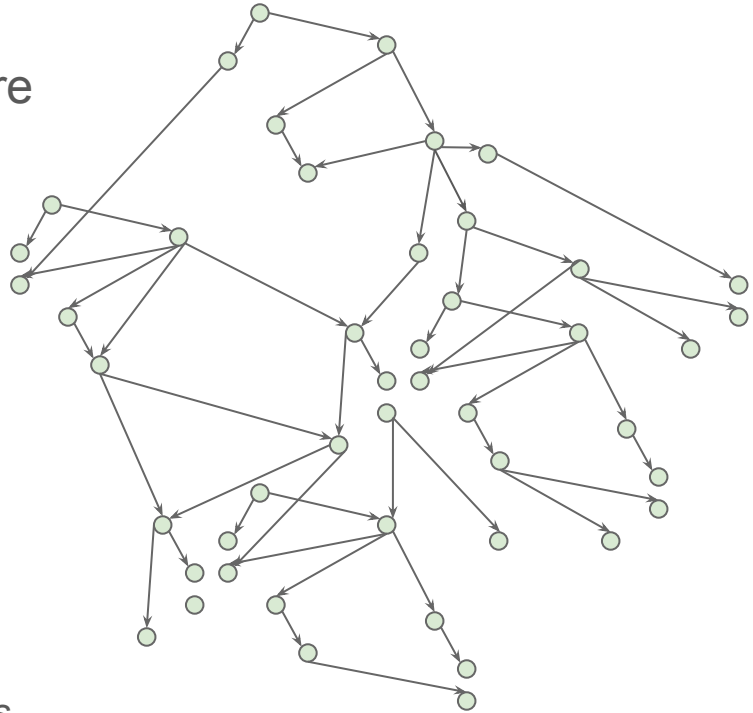
- RL for device placement
- RL for chip placement


Machine Learning
Supervised
Unsupervised Learning
Reinforcement Learning
Learning from labeled data
e.g. classification
Learning from unlabeled data
e.g. clustering
Learning from explorations and exploitations
e.g. playing Go

RL for systems and chips

Many different problems in systems and hardware require decision-making optimization:

- Computational graph placement:
  - Input: A TensorFlow graph
  - Objective: Placement on GPU/TPU/CPU platforms
- Chip placement:
  - Input: A chip netlist graph
  - Objective: Placement on 2D or 3D grids
- Datacenter resource allocation:
  - Input: A jobs workload graph
  - Objective: Placement on datacenter cells and racks
- ...



Some resources for RL

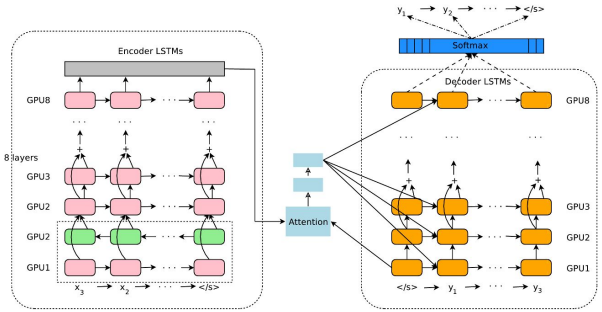
- Reinforcement Learning: An Introduction, Sutton & Barto 2018 (textbook)
  - Thorough definitions & theory, 2nd edition draft available online
- Online courses with lecture slides/videos:
  - David Silver's RL Course (video lectures)
  - UC Berkeley (rll.berkeley.edu/deeprlcourse)
  - Stanford (cs234.stanford.edu)
- Open-Source Reinforcement Learning Examples
  - Tf-agents: An RL library built on top of TensorFlow.
  - github.com/openai/baselines, gym.openai.com/envs
  - github/carpedm20/deep-rl-tensorflow

This talk

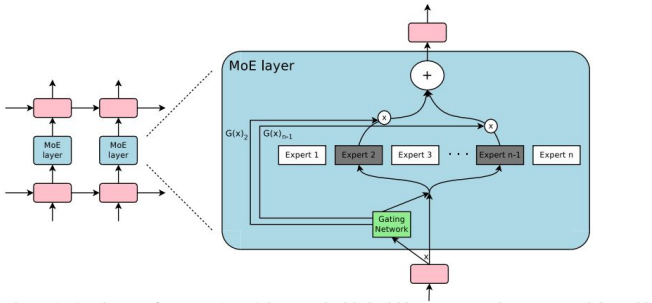
- **RL for device placement**
- RL for chip placement

What is device placement and why is it important?

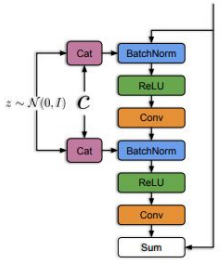
Trend towards many-device training, bigger models, larger batch sizes



Google neural machine translation'16
300 million parameters,
trained on 128 GPUs



Sparsely gated mixture of experts'17
130 billion parameters,
trained on 128 GPUs

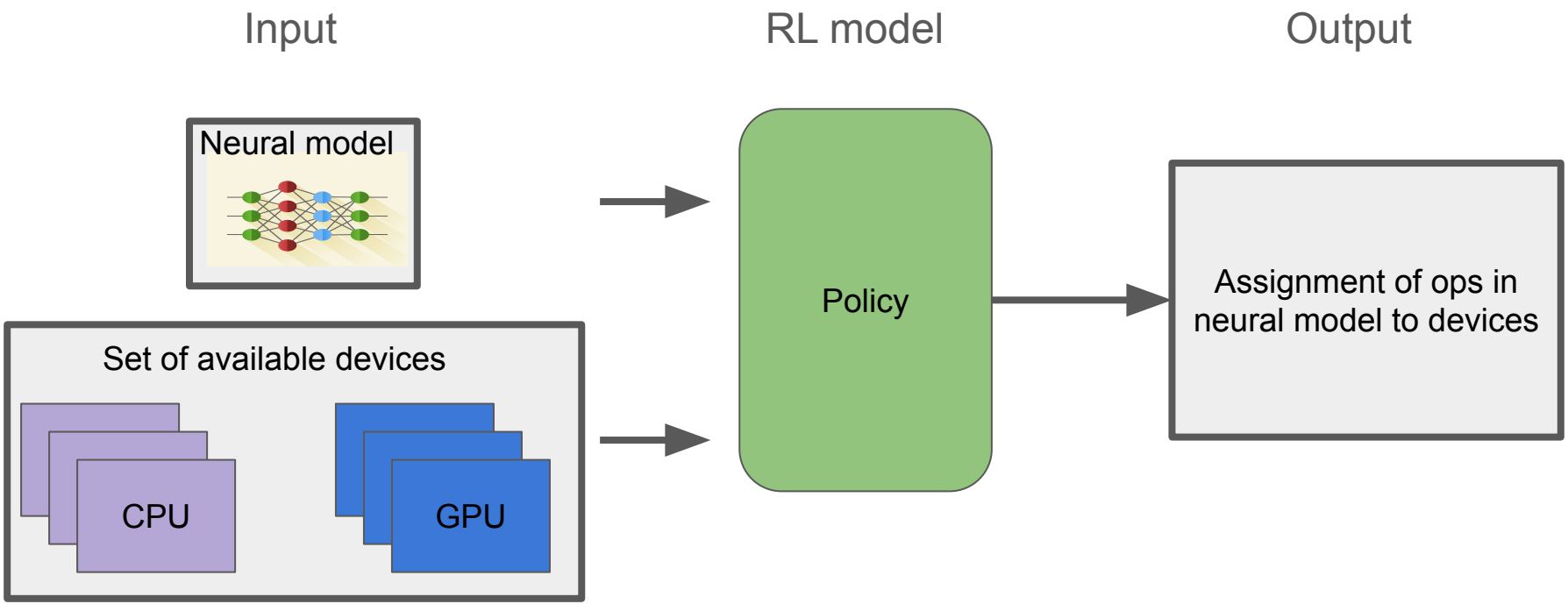


BigGAN'18
355 million parameters,
trained on 512 TPU cores

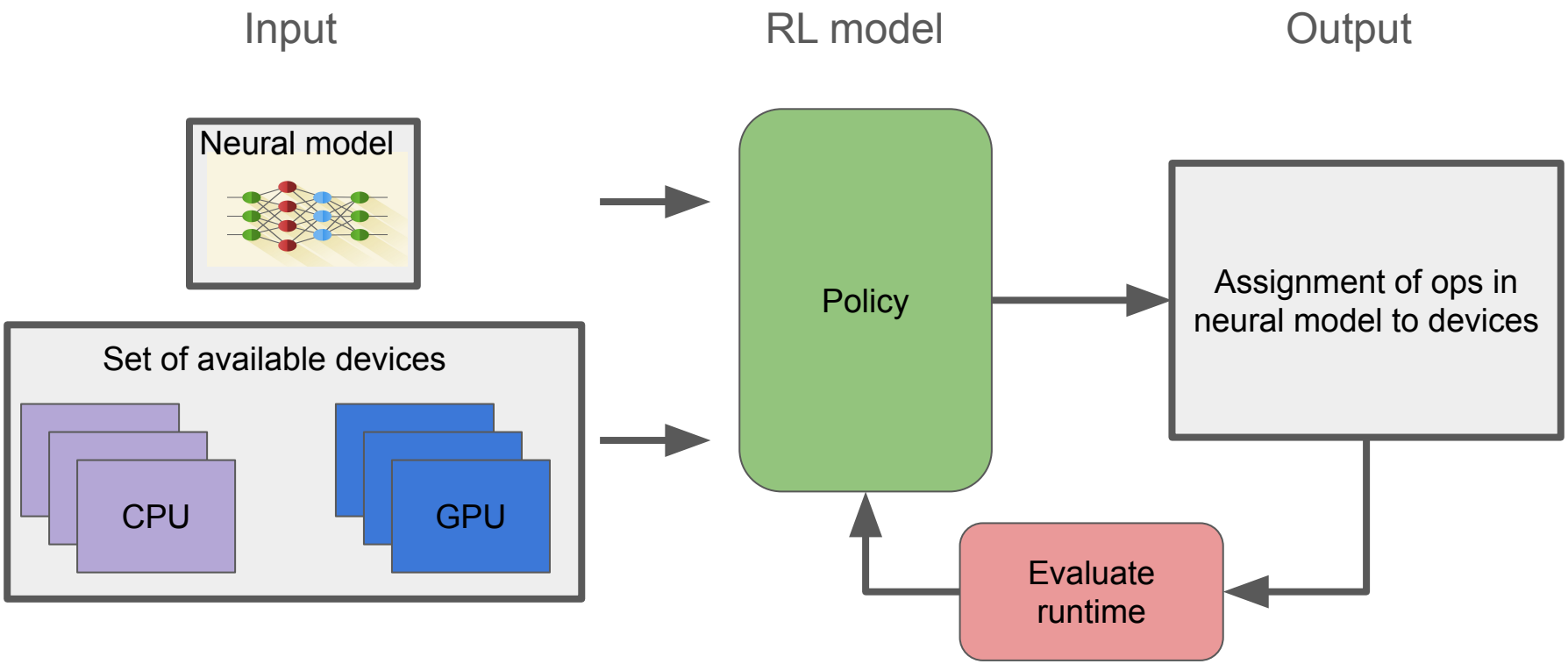
Standard practice for device placement

- Often based on greedy heuristics
- Requires deep understanding of devices: nonlinear FLOPs, bandwidth, latency behavior
- Requires modeling parallelism and pipelining
- Does not generalize well

Posing device placement as an RL problem



Posing device placement as an RL problem



Problem formulation for hierarchical placement

$$J(\theta_g, \theta_d) = \mathbf{E}_{\mathbf{P}(\mathbf{d};\theta_\mathbf{g},\theta_\mathbf{d})}[R_d] = \sum_{g\sim\pi_g} \sum_{d\sim\pi_d} p(g;\theta_g)p(d|g;\theta_d)R_d$$

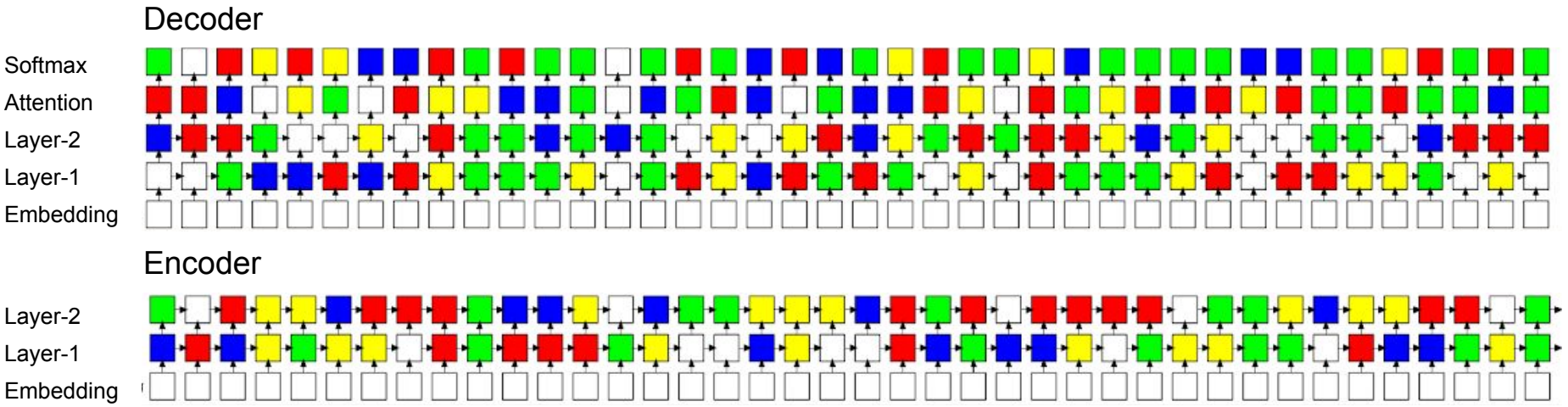
$J(\theta_g, \theta_d)$: expected runtime

θ_g: trainable parameters of Grouper

θ_d: trainable parameters of Placer

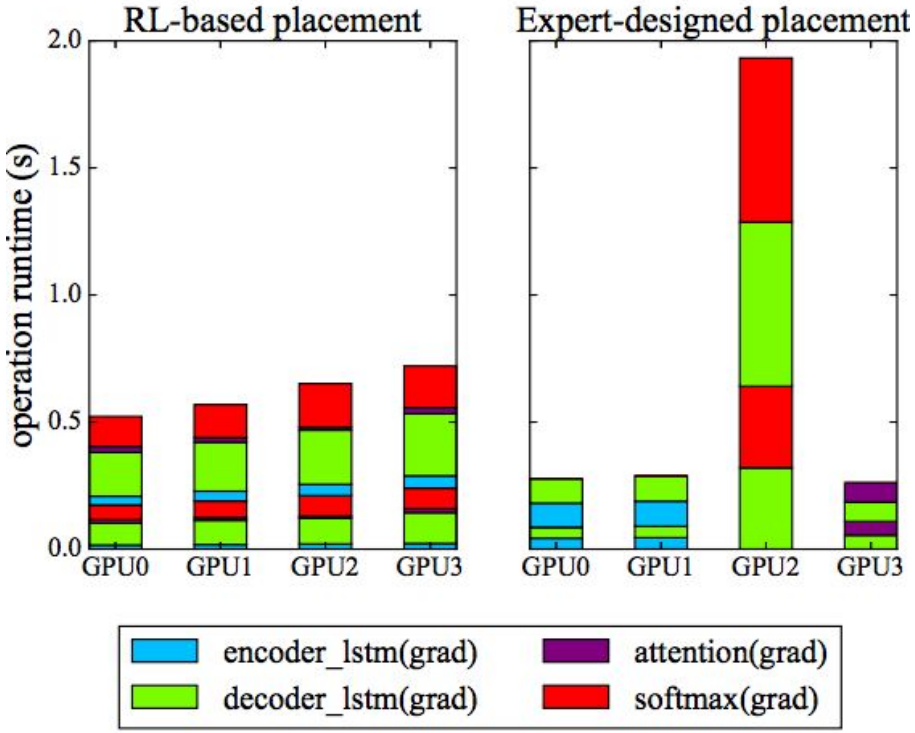
R_d: runtime for placement d

Learned placement on NMT

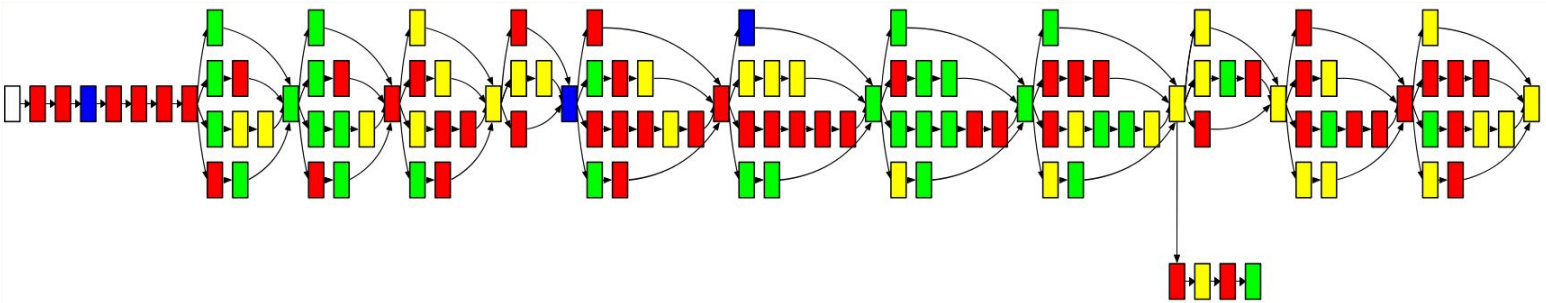


White represents CPU (Ixion Haswell 2300)
Each other color represents a separate GPU (Nvidia Tesla K80)
Searching over a space of 5^280 possible assignments

Profiling placement on NMT

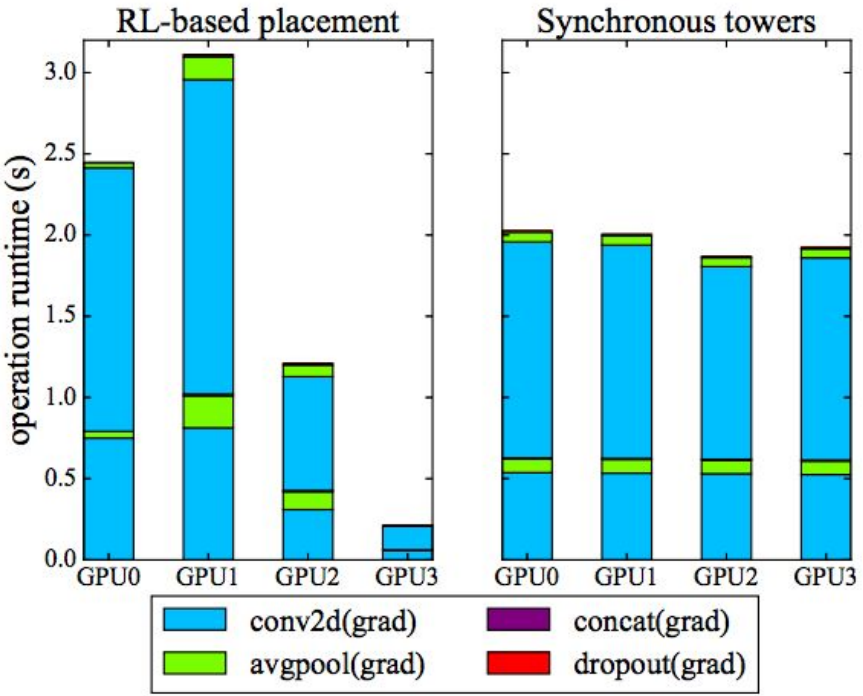


Learned placement on Inception-V3

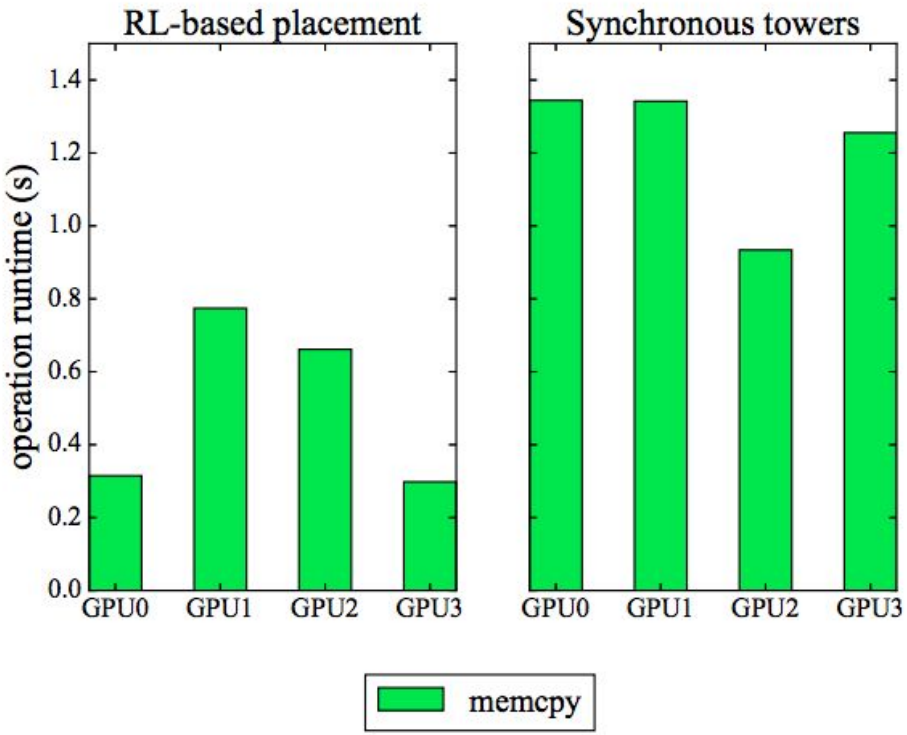


White represents CPU (Ixion Haswell 2300)
Each other color represents a separate GPU (Nvidia Tesla K80)
Searching over a space of 5^83 possible assignments

Profiling placement on Inception-V3

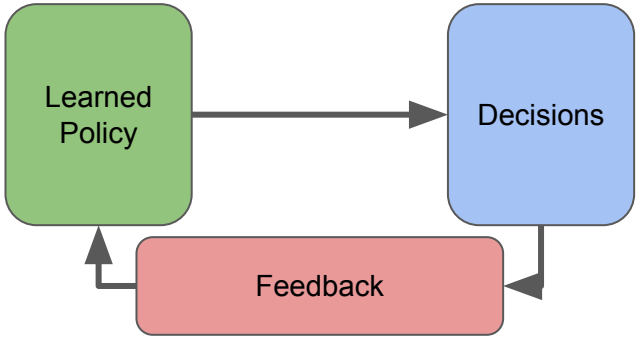


Profiling placement on Inception-V3



Policy optimization for device placement

Model (#devices)	**GDP-one** (s)	HP (s)	METIS (s)	HDP (s)	Run time speed up over HP / HDP	Search speed up
2-layer RNNLM (2)	0.234	0.257	0.355	0.243	9.8% / 4%	2.95x
4-layer RNNLM (4)	0.409	0.48	OOM	0.490	17.4% / 19.8%	1.76x
2-layer GNMT (2)	0.301	0.384	OOM	0.376	27.6% / 24.9%	30x
4-layer GNMT (4)	0.409	0.469	OOM	0.520	14.7% / 27.1%	58.8x
8-layer GNMT (8)	0.649	0.610	OOM	0.693	-6% / 6.8%	7.35x
2-layer Transformer-XL (2)	0.386	0.473	OOM	0.435	22.5% / 12.7%	40x
4-layer Transformer-XL (4)	0.580	0.641	OOM	0.621	11.4% / 7.1%	26.7x
8-layer Transformer-XL (8)	0.748	0.813	OOM	0.789	8.9% / 5.5%	16.7x
Inception (2)	0.405	0.418	0.423	0.417	3.2% / 3%	13.5x
AmoebaNet (4)	0.394	0.44	0.426	0.418	26.1% / 6.1%	58.8x
2-stack 18-layer WaveNet (2)	0.317	0.376	OOM	0.354	18.6% / 11.7%	6.67x
4-stack 36-layer WaveNet (4)	0.659	0.988	OOM	0.721	50% / 9.4%	20x
GEOMEAN	-	-	-	-	**16% / 9.2%**	**15x**



1- Azalia Mirhoseini*, Hieu Pham*, Quoc V. Le, Benoit Steiner, Rasmus Larsen, Yuefeng Zhou, Naveen Kumar, Mohammad Norouzi, Samy Bengio, Jeff Dean, ICML'17: Device Placement Optimization with Reinforcement Learning
2- Azalia Mirhoseini*, Anna Goldie*, Hieu Pham, Benoit Steiner, Quoc V. Le and Jeff Dean, ICLR'18: A Hierarchical Model for Device Placement
3- Yanqi Zhou, Sudip Roy, Amirali Abdolrashidi, Daniel Wong, Peter C. Ma, Qiumin Xu Ming Zhong, Hanxiao Liu, Anna Goldie, Azalia Mirhoseini, James Laudon, arxiv 2019 "GDP: generalized device placement for dataflow graphs"

This talk

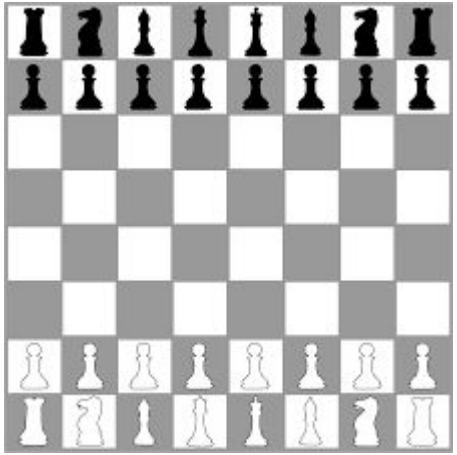
- RL for device placement
- **RL for chip placement**

Machine Learning for ASIC Chip Placement

Tech/Research Leads: Anna Goldie and Azalia Mirhoseini
Engineering Leads: Joe Jiang and Mustafa Yazgan

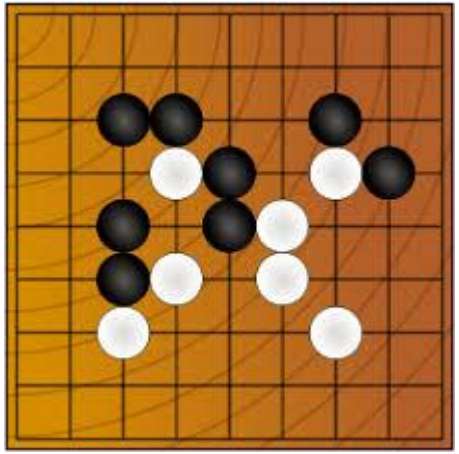
Collaborators: Anand Babu, Jeff Dean, Roger Carpenter, William Hang, Richard Ho, James Laudon, Eric Johnson, Young-Joon Lee, Azade Nazi, Omkar Pathak, Quoc Le, Sudip Roy, Amir Salek, Kavya Setty, Ebrahim Songhori, Andy Tong, Emre Tuncer, Shen Wang, Amir Yazdanbakhsh

Chess



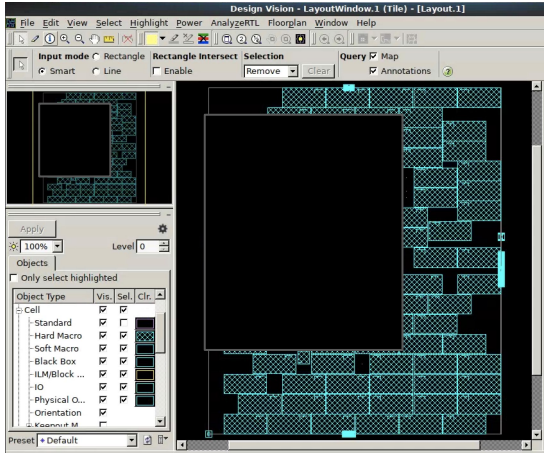
Number of states ~ 10^{123}

Go



Number of states ~ 10^{360}

Chip Placement



Number of states ~ 10^{9000}

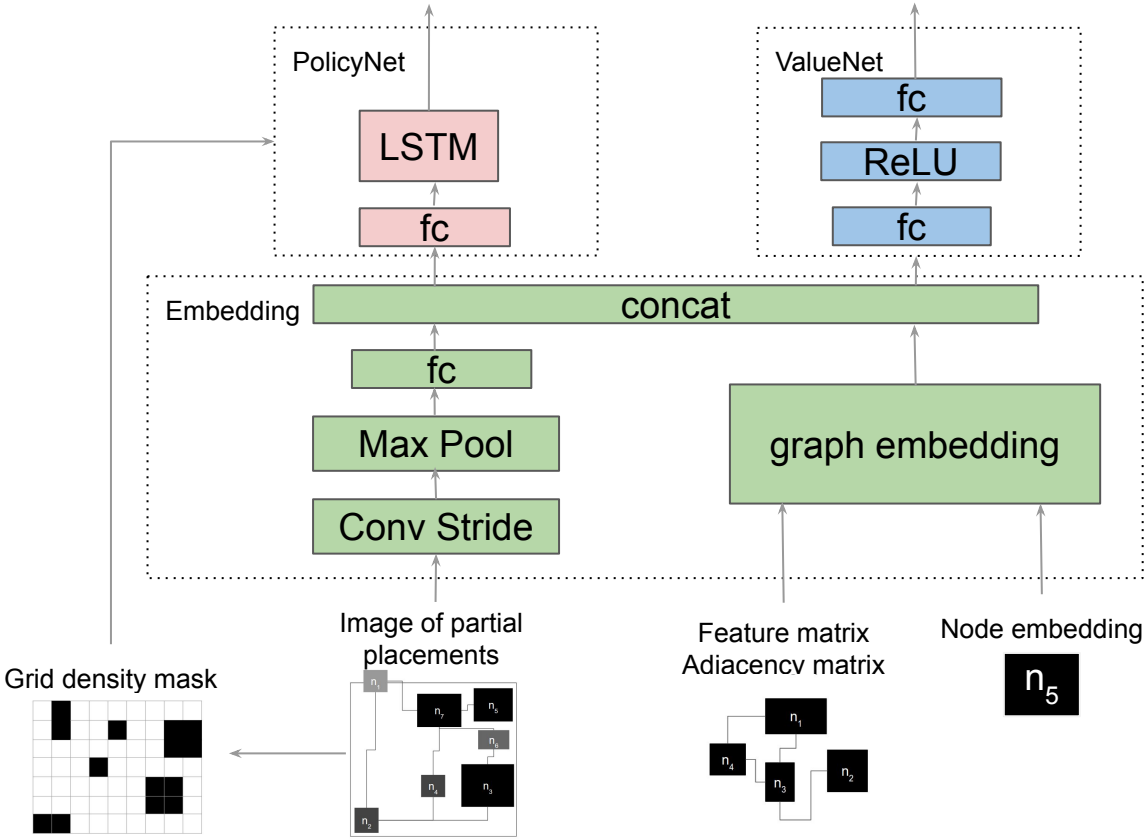
A Few Complexities

Problem size is very large (millions or billions of items)

Multiple objectives: area, timing, congestion, design rules, etc.

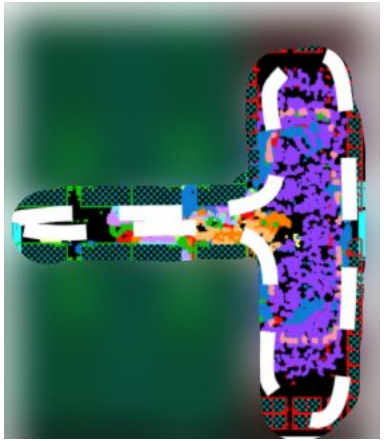
True reward function is very expensive to evaluate (many hours)

Policy architecture

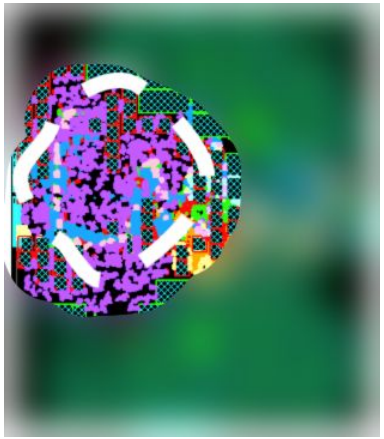


Results on a Low Power ML Accelerator Chip

Human Expert



ML Placer



	Proxy Congestion	Proxy Wirelength
Human Expert	0.76060	0.10135
ML Placer	0.60646	0.07430
Improvement	**20.2%**	**26.7%**

Blurred for confidentiality

Results on a TPU Design Block

White blurred area are macros (memory) and green blurred area are standard cell clusters (logic)
ML placer finds smoother, rounder macro placements to reduce the wirelength

Human Expert

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

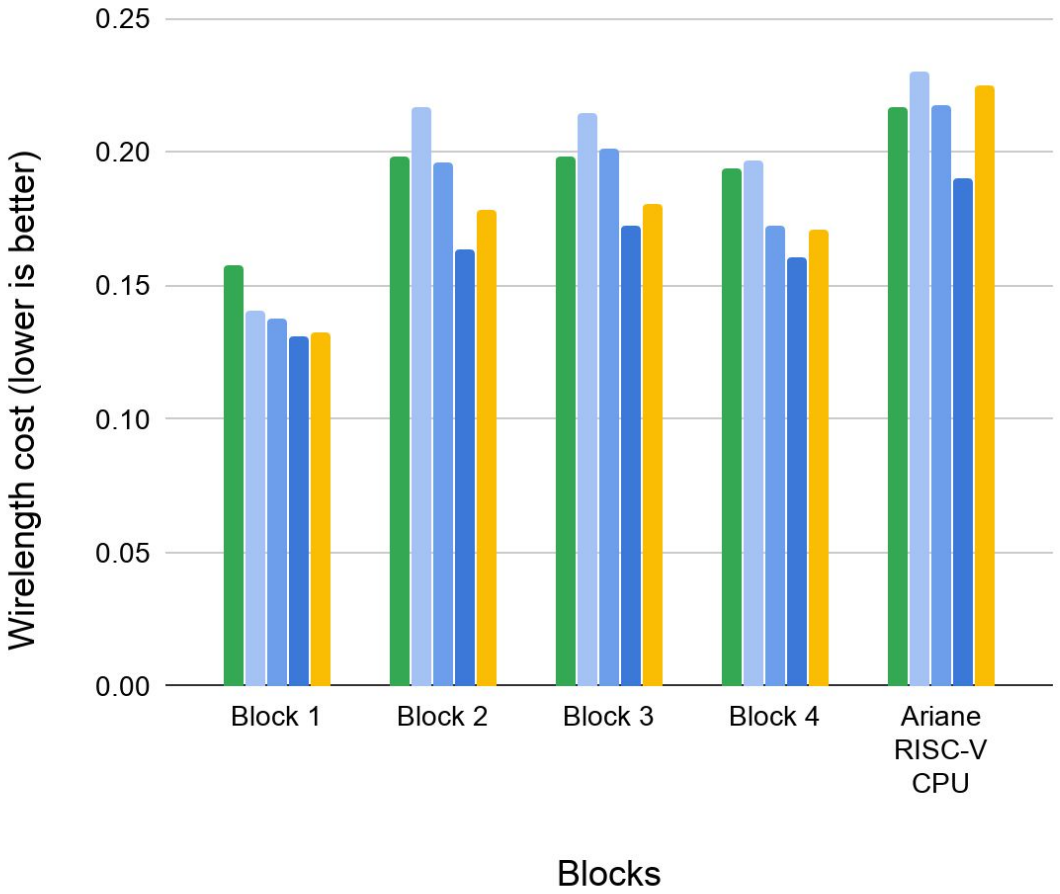
DRC: Design Rule Checking

ML Placer

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

Generalization Results

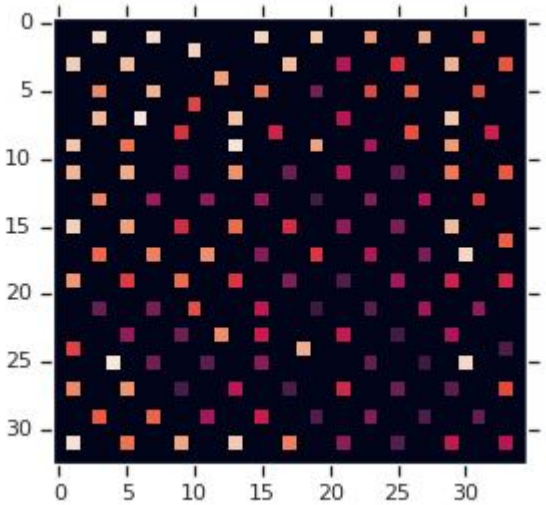
Google Research



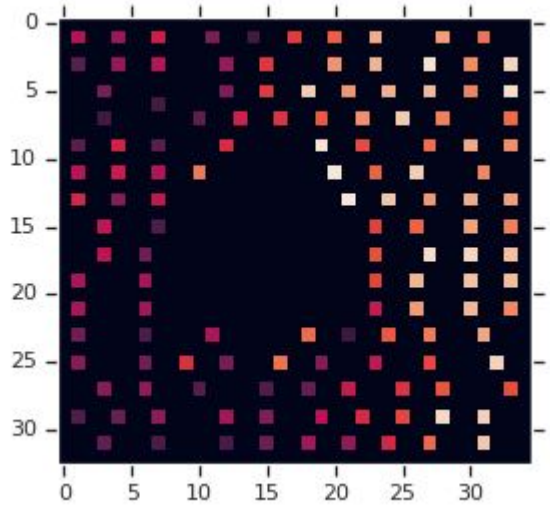
- The zero shot policy is trained on a set of unrelated blocks for ~24 hrs.
- Placement is done using 16 Tesla P100 GPUs.
- Block 1-4 are real TPU blocks.

Ariane (RISC-V) Placement Visualization

Training policy from scratch



Finetuning a pre-trained policy

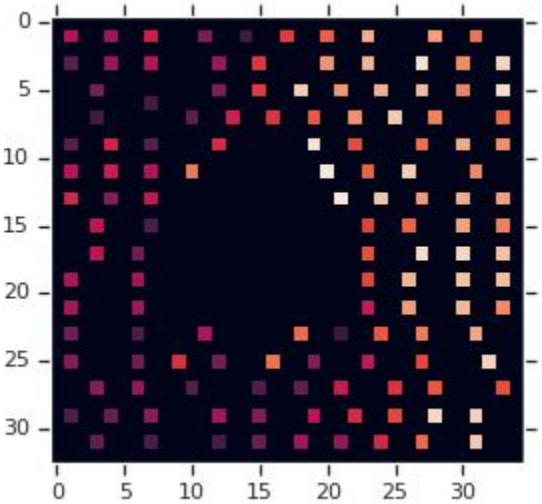


The animation shows the macro placements as the training progresses. Each square shows the center of a macro.

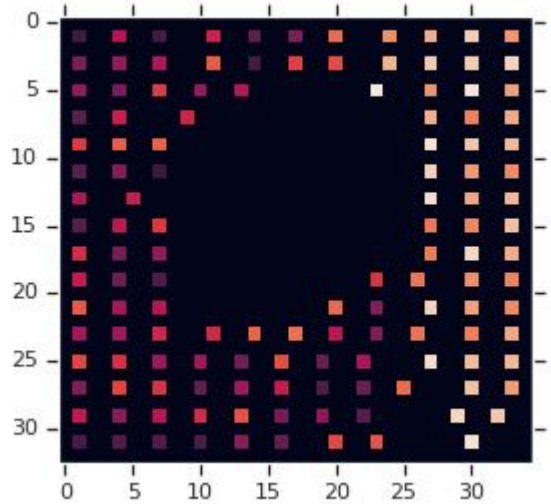
Ariane is an open-source RISC-V processor. See: https://github.com/pulp-platform/ariane

Ariane (RISC-V) Block Final Placement

Placement results of the pre-trained policy (Zero Shot)



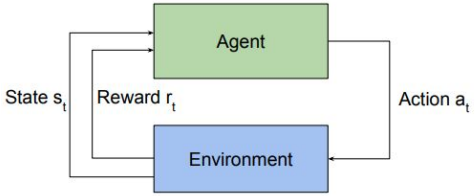
Placement results of the Finetuned policy

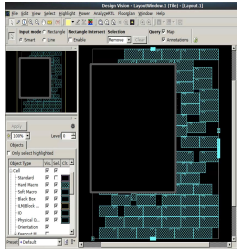


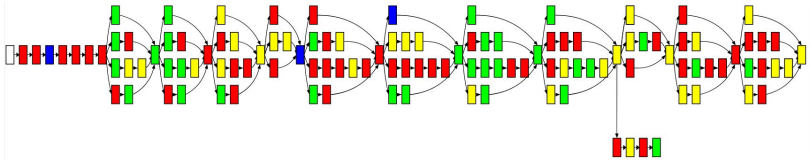
Ariane is an open-source RISC-V processor. See: https://github.com/pulp-platform/ariane

We have gotten comparable or superhuman results on all the blocks we have tried on so far

Block	Version	Timing		Area (sq. um)	
		Worst Negative Slack (WNS) (ps)	Total Negative Slack (TNS) (ns)	Buf + Inv	Total
A	Manual	**72**	97.4	49741	830799
	ML Placer	123	**75.1**	**31888**	**799507**
B	Manual	58	17.9	22254	947766
	ML Placer	**27**	**7.04**	**21492**	**946771**
C	Manual	-6	-0.3	**10226**	871617
	ML Placer	**-8**	-0.3	12746	**868098**

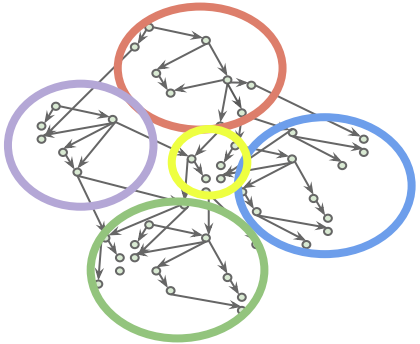


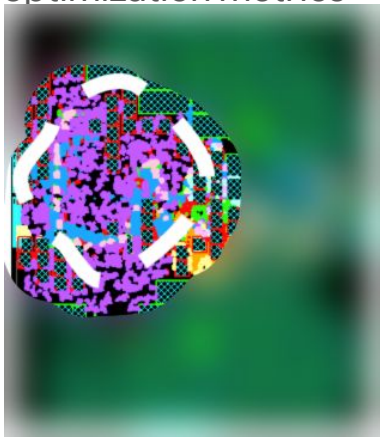




- ML/RL for systems and chip design
  - Improve engineering efficiency by automating and optimizing various stages of the pipeline and potentially enabling global optimization
  - Enable transfer of knowledge across multiple chips/systems
  - Automatically generate designs that explore trade-offs between various optimization metrics

- Recap of this talk:
  - RL for device placement
  - RL for chip placement





Contact:
azalia@google.com
Twitter: azaliamirh@