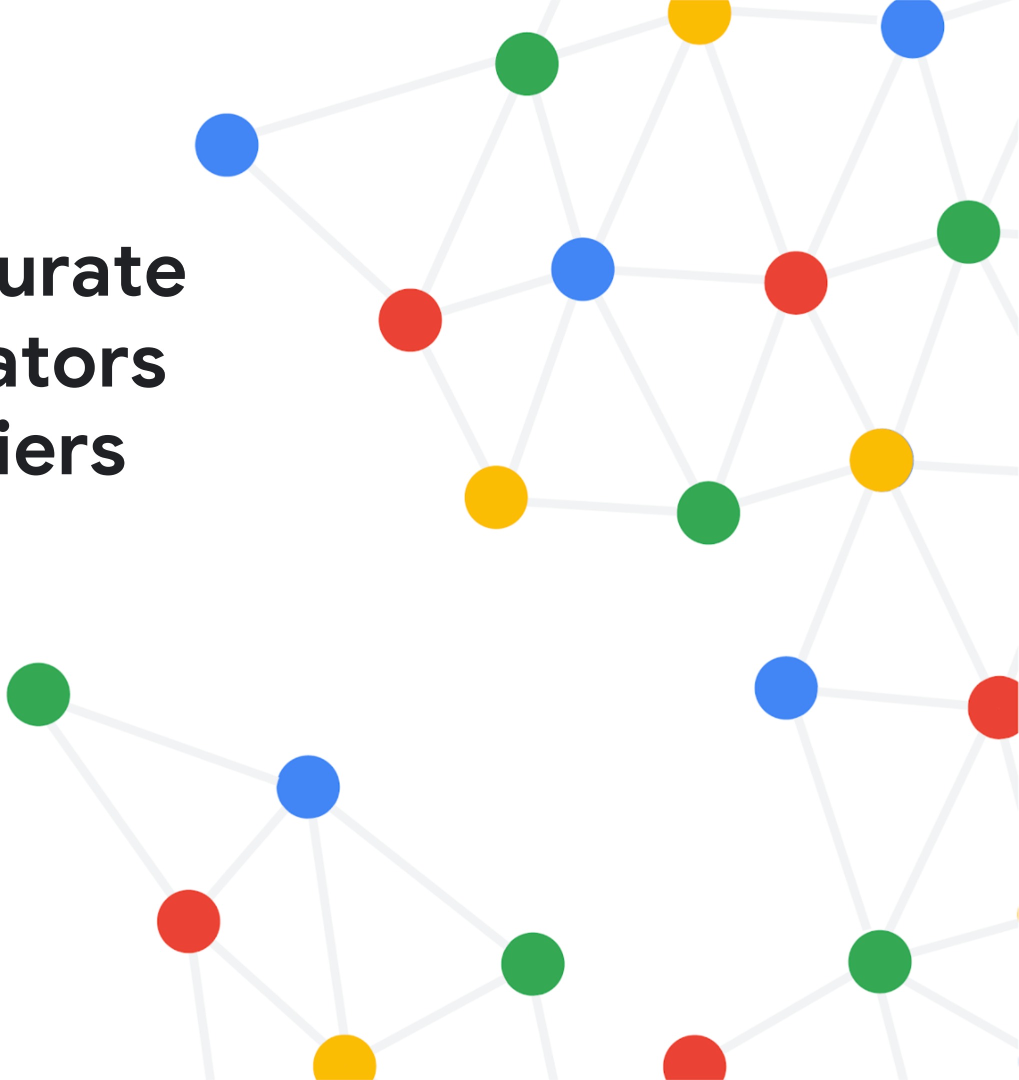
Learning to Design Accurate Deep Learning Accelerators with Inaccurate Multipliers

Paras Jain[2]

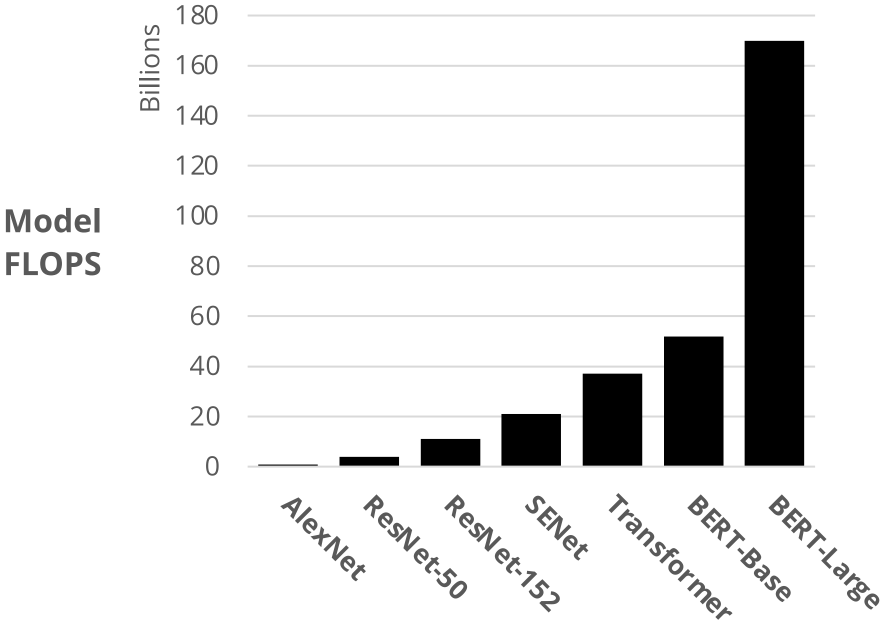
with Safeen Huda[1], Martin Maas[1], Joseph Gonzalez[2], Ion Stoica[2] and Azalia Mirhoseini[1]

[1] Google Research

[2] Berkeley
UNIVERSITY OF CALIFORNIA

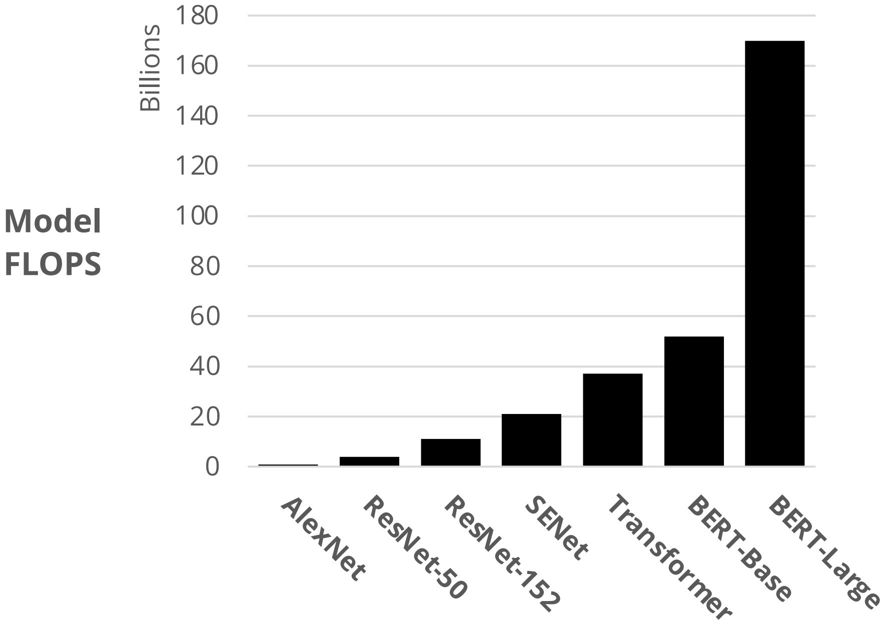


Deep learning's inference energy problem



Rise of high-parameter models

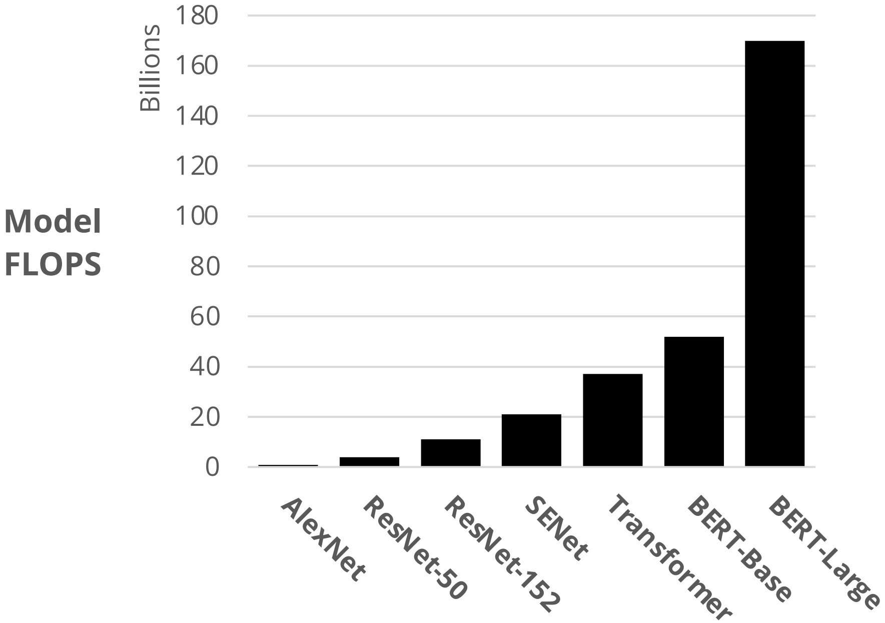
Deep learning's inference energy problem

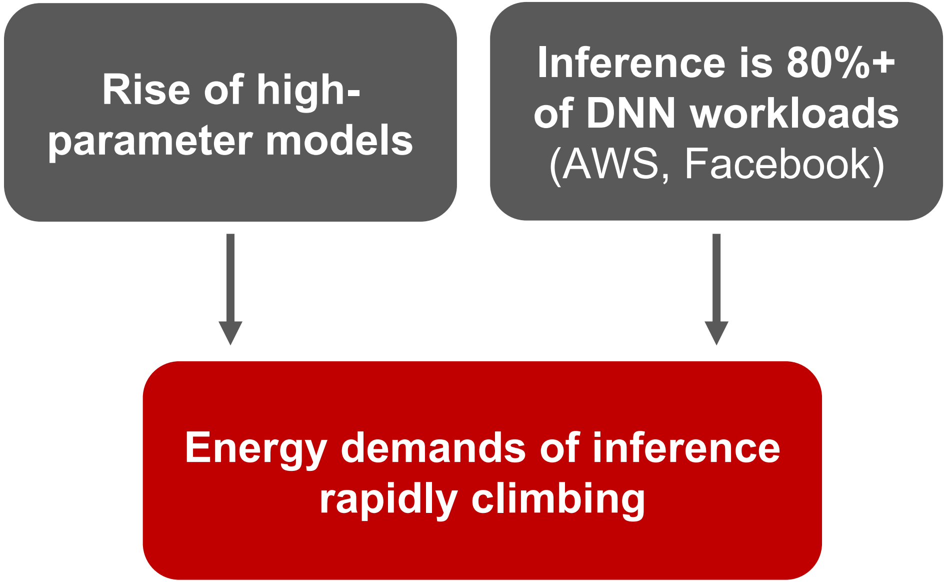


Rise of high-parameter models

Inference is 80%+ of DNN workloads (AWS, Facebook)

Deep learning’s inference energy problem





Rise of high-parameter models

Inference is 80%+ of DNN workloads (AWS, Facebook)

Energy demands of inference rapidly climbing

Approximate computing as a new way to save power on DNN accelerators

Quantization

Pruning

Approximate MACs

- Deep learning models are tolerant to approximations like quantization

Approximate computing as a new way to save power on DNN accelerators

Quantization

Pruning

Approximate MACs

This talk

- Deep learning models are tolerant to approximations like quantization
- We study: emerging approximate multipliers + adders to trade-off accuracy for power
- *Complementary* approach to quantization and sparsity
- **Challenge:** how to maintain high accuracy under approximation?

Approximate computing as a new way to save power on DNN accelerators

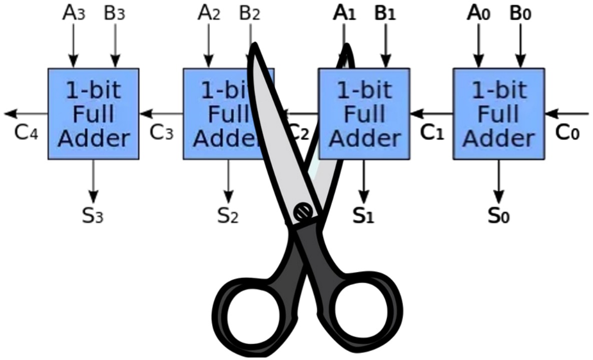
Quantization

This talk

Approximate MACs

How to achieve power savings with an approximate inference accelerator *without any accuracy loss on a large-scale dataset*?

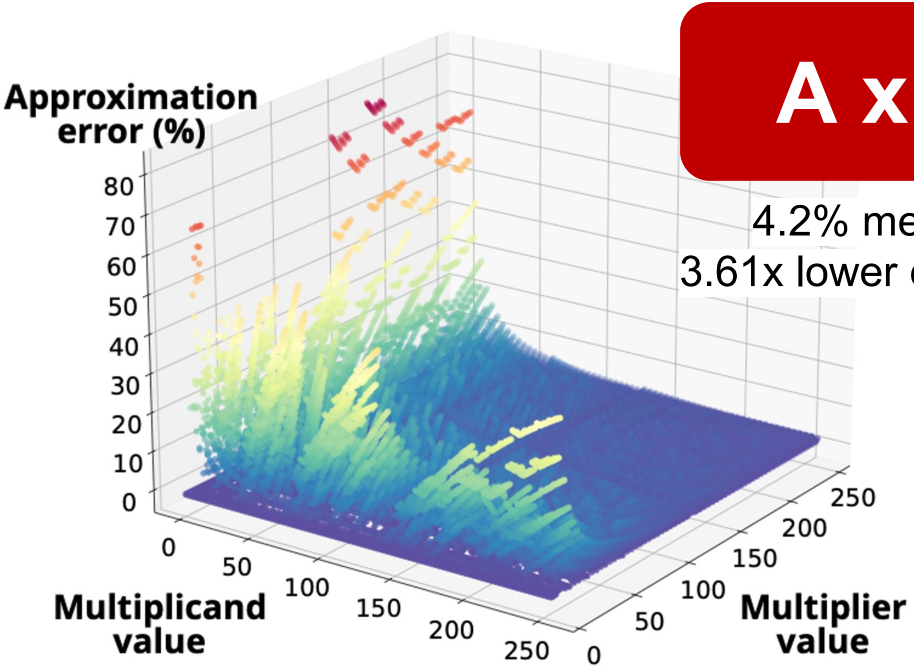
Background: approximate MACs to trade-off power and accuracy



- Parts of fully-accurate circuits can be removed to trade-off **accuracy for better power efficiency**
- Example: truncate the carry chain in an 8-bit adder
- Extensive prior work to produce such multipliers/adders [1] [2] [survey].
- Functionally approximate circuits only

[1] https://dl.acm.org/doi/10.1145/2228360.2228509
[2] https://ieeexplore.ieee.org/abstract/document/7926993
[survey] https://www.osti.gov/pages/servlets/purl/1286958

Background: approximate MACs to trade-off power and accuracy



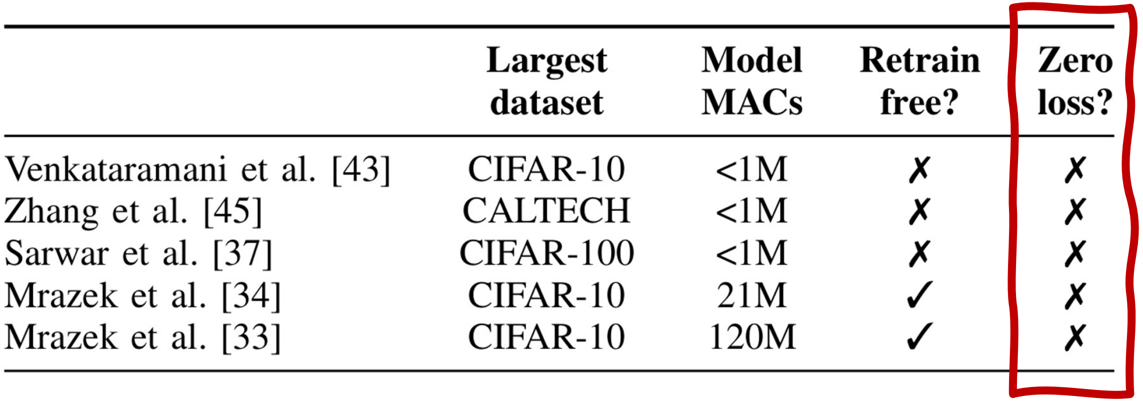
A x B ≈ C

4.2% mean relative error

3.61x lower energy consumption

V. Mrazek, R. Hrbacek, Z. Vasicek and L. Sekanina, EvoApprox8b: Library of approximate adders and multipliers for circuit design and benchmarking of approximation methods. Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017

Challenge: Prior designs with approximate MACs degrade accuracy

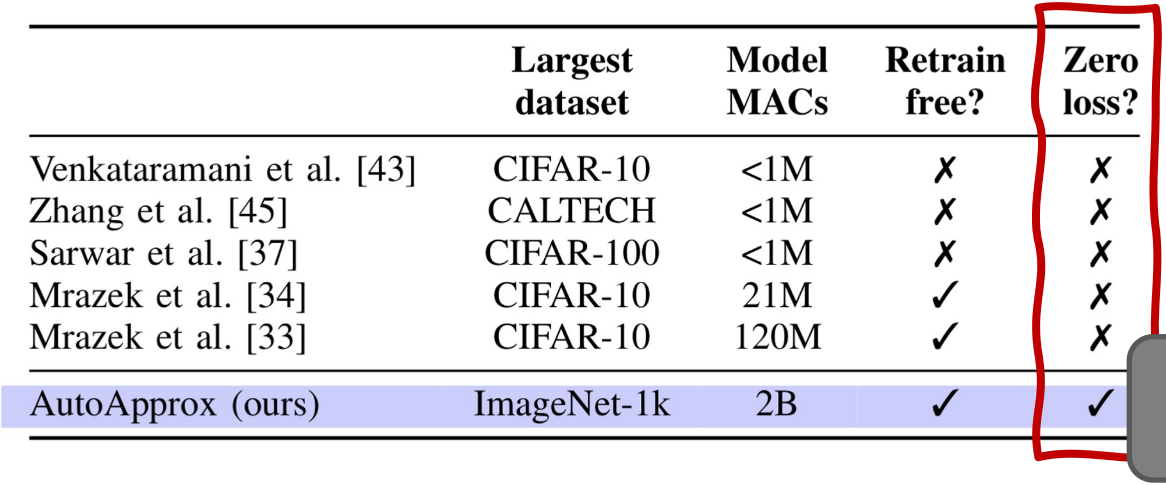


	Largest dataset	Model MACs	Retrain free?	Zero loss?
Venkataramani et al. [43]	CIFAR-10	<1M	✗	✗
Zhang et al. [45]	CALTECH	<1M	✗	✗
Sarwar et al. [37]	CIFAR-100	<1M	✗	✗
Mrazek et al. [34]	CIFAR-10	21M	✓	✗
Mrazek et al. [33]	CIFAR-10	120M	✓	✗

Must incur accuracy penalty!

Evaluated on CIFAR w/ small models

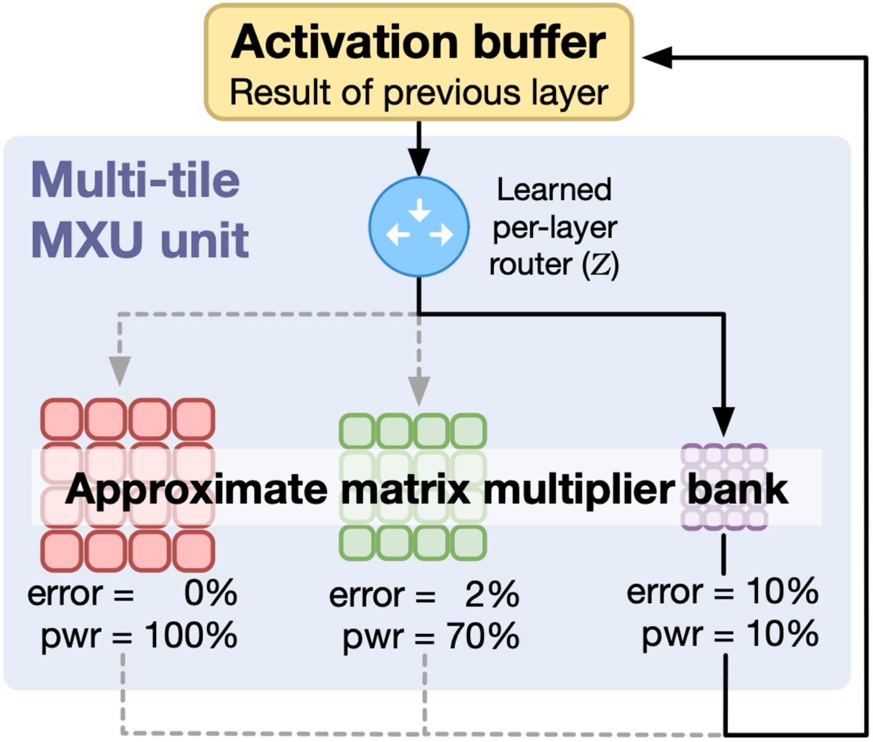
This work: We show it is possible to use approximation and maintain accuracy



	Largest dataset	Model MACs	Retrain free?	Zero loss?
Venkataramani et al. [43]	CIFAR-10	<1M	✗	✗
Zhang et al. [45]	CALTECH	<1M	✗	✗
Sarwar et al. [37]	CIFAR-100	<1M	✗	✗
Mrazek et al. [34]	CIFAR-10	21M	✓	✗
Mrazek et al. [33]	CIFAR-10	120M	✓	✗
AutoApprox (ours)	ImageNet-1k	2B	✓	✓

10^3 more data
(bytes)

Key Insight: Add additional approximate units next to exact units as a low-power "fast-path"



At inference, router selects <u>one systolic array</u>

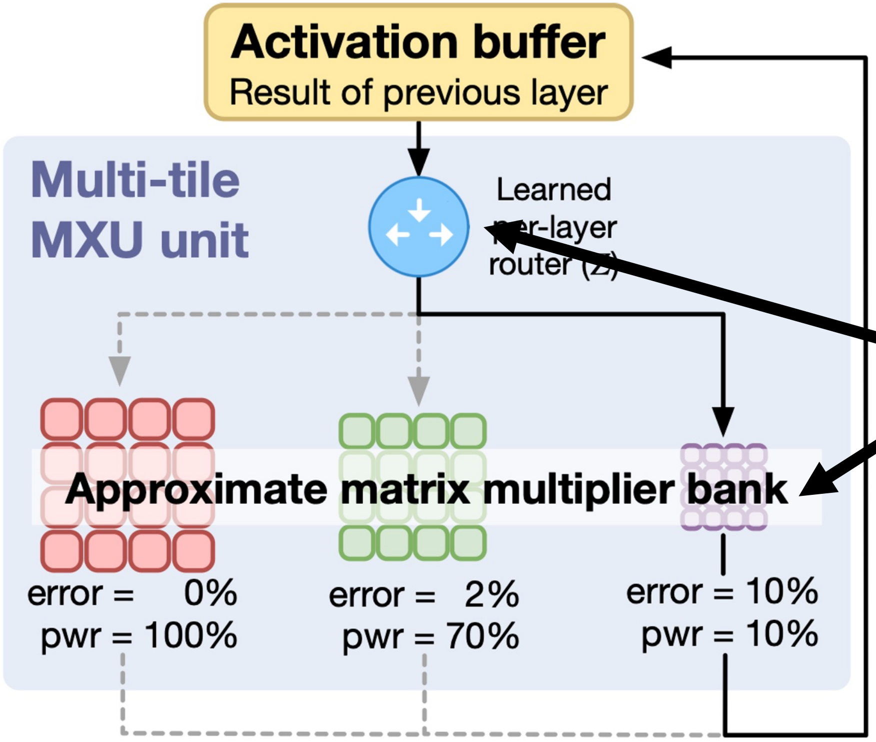
Error-tolerant workloads:

→ Save power by using approximate MAC

Sensitive workloads:

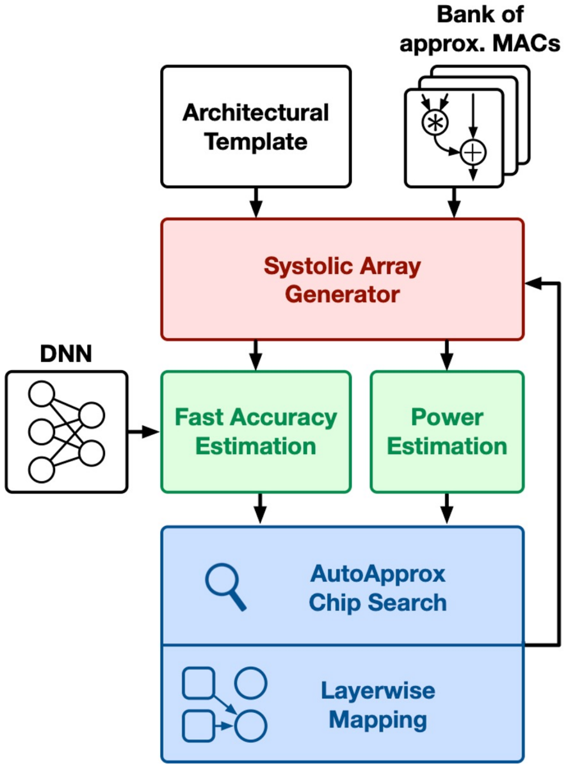
→ Maintain accuracy by using exact MAC

Key Insight: Add additional approximate units next to exact units as a low-power "fast-path"



How to select MACs + mapping?
Up to $\underline{O(2^{268})}$ search space

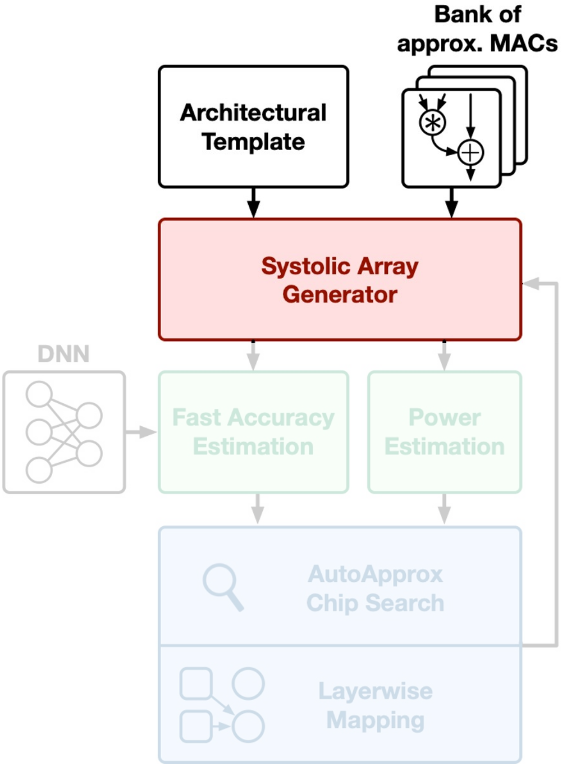
AutoApprox: full-stack framework to design zero-loss approximate accelerators



Contributions:

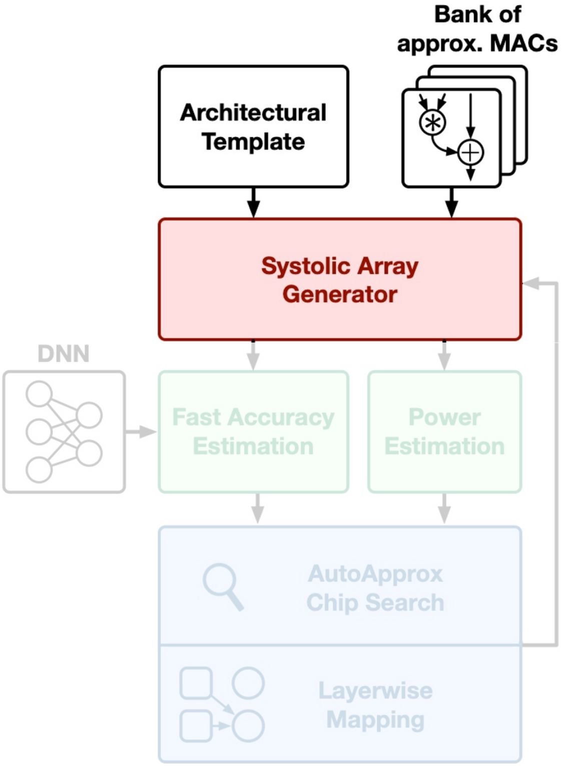
1. **Approx. TPU architecture** w/ exact fallback
2. **Fast e2e accuracy simulation:** 7000x simulation speedup over Verilator
3. **ML-guided search:** Novel Bayesian optimizer for large combinatorial space of circuits

Candidate hardware generation

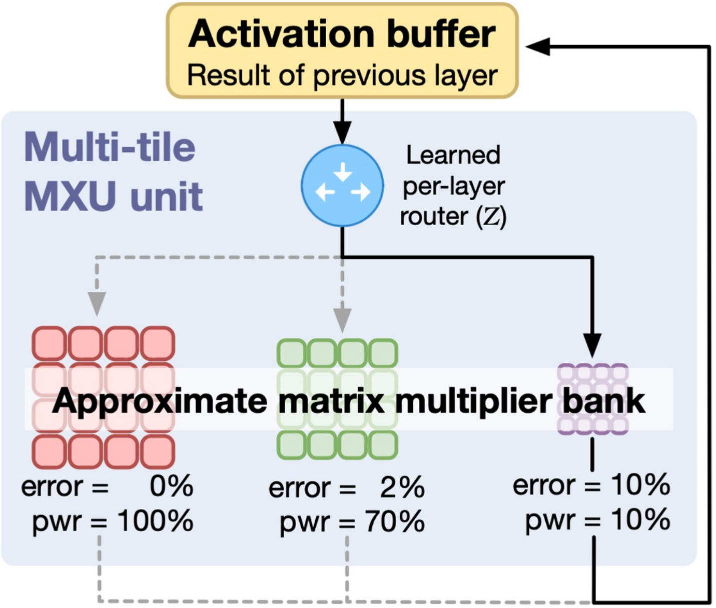


- Systolic array generator instantiates diverse set of approximate TPU designs
- **Architectural template**: TPU w/ sister approximate matrix multipliers
- **Approximate MAC bank**: 36 MACs from prior work, can be augmented w/ new designs

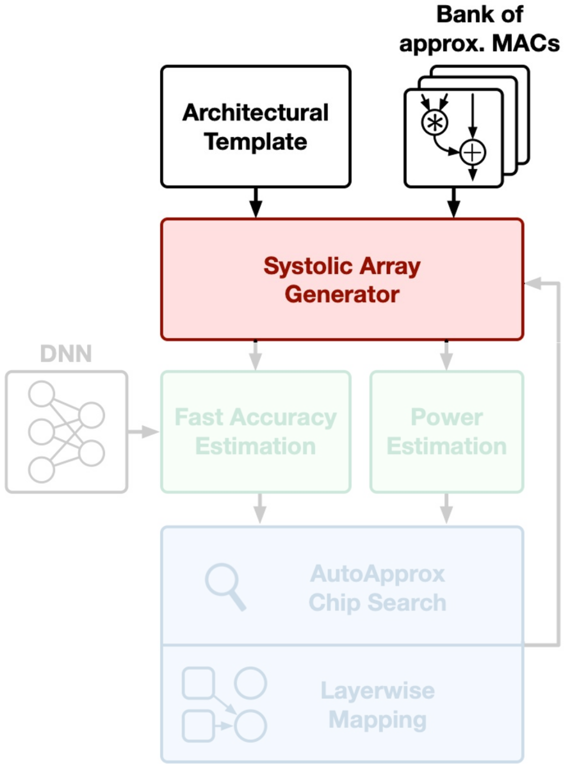
Candidate hardware generation

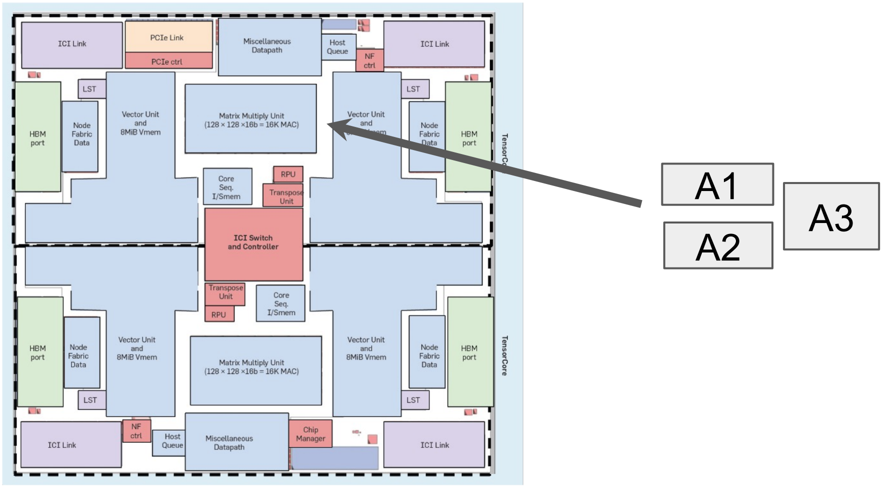


Template: At inference, router selects one systolic array



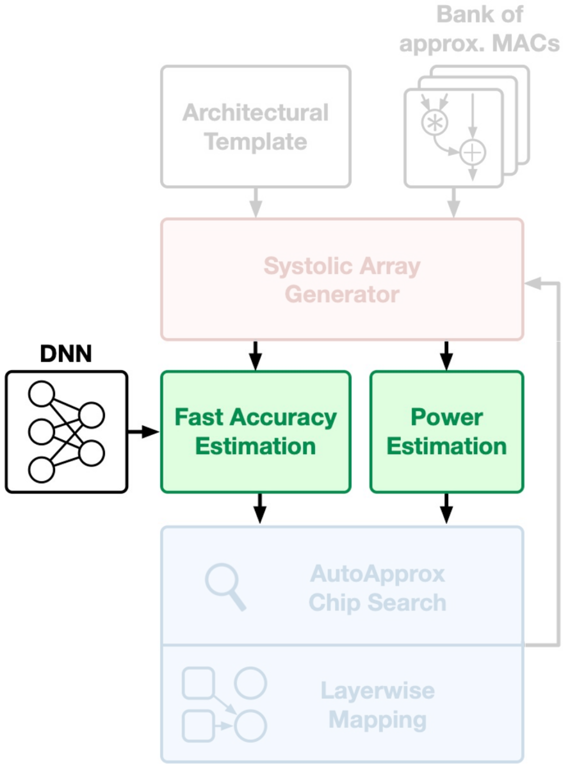
Candidate hardware generation





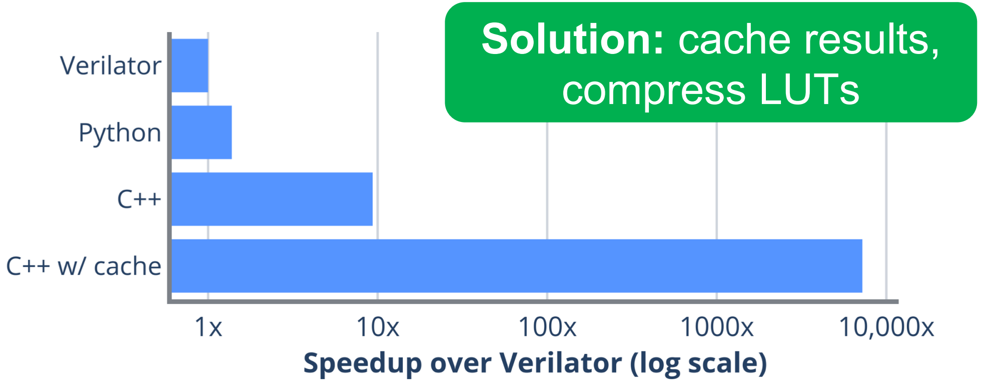
- Systolic array generator instantiates diverse set of approximate TPU designs
- **Architectural template**: TPU w/ sister approximate matrix multipliers
- **Approximate MAC bank**: 36 MACs from prior work, can be augmented w/ new designs

Scoring candidates by accuracy and PPA



VERILATOR

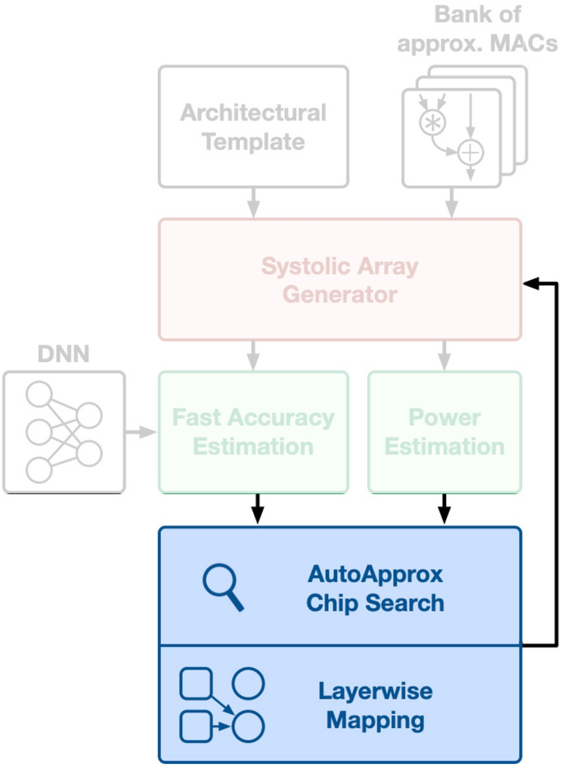
Evaluating single inference with Verilator takes 4.2hrs



Solution: cache results, compress LUTs

ML-guided search

with pruning, continuous relaxation

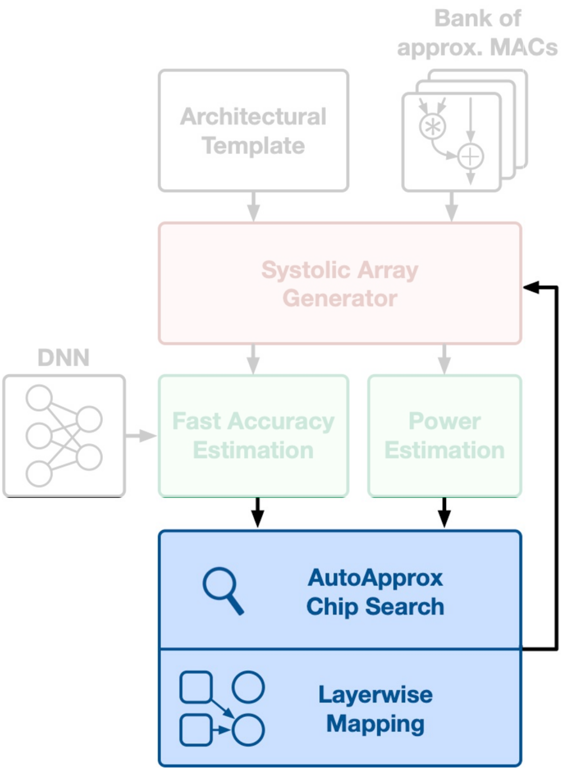


$$
\begin{aligned}
\min_{z} \quad & \sum_{i=1}^{N} q_i^{\intercal} Z_i \\
\text{s.t.} \quad & \text{ACC}(Z) \geq \tau \\
& \text{AREA}(Z) \leq \phi \\
& \sum_{j=1}^{K} Z_{ij} = 1 \qquad \forall i \in \{1, \ldots, N\} \\
& Z \in \{0,\, 1\}^{N \times K}
\end{aligned}
$$

Search space O(2^{268})

ML-guided search

with pruning, continuous relaxation



Search space O(2^{268})

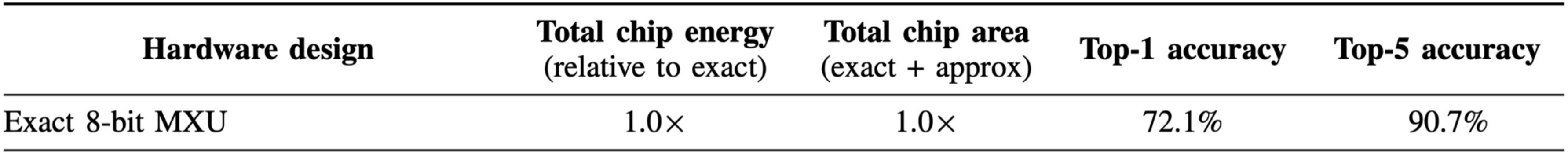
(a) Bayesian optimization to balance exploration + exploitation w/ learned surrogate cost function

(b) Prune catastrophic trials using greedy lower bound

(c) Relax combinatorial search space into continuous space

Results: Evaluating AutoApprox on large-scale workload + dataset

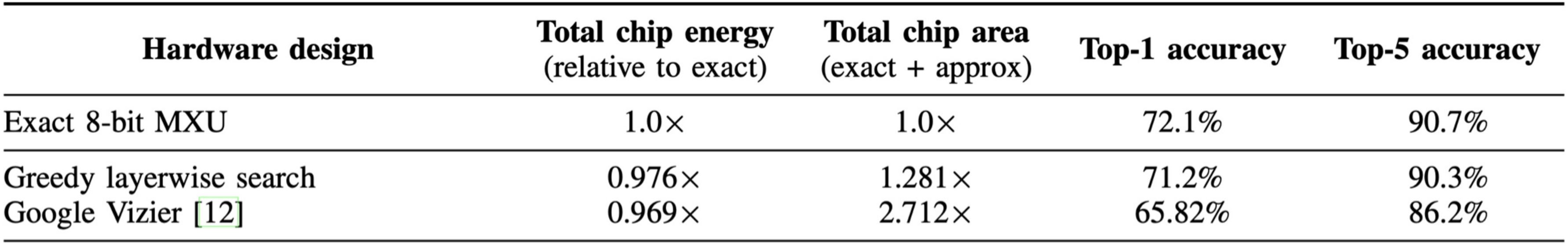
Workload: ResNet-50 on ImageNet-1k
Evaluating routed TPU design w/ approximate cores
Energy, perf. and area evaluated at <10nm PDK



Hardware design	**Total chip energy** (relative to exact)	**Total chip area** (exact + approx)	**Top-1 accuracy**	**Top-5 accuracy**
Exact 8-bit MXU	1.0×	1.0×	72.1%	90.7%

Results: Evaluating AutoApprox on large-scale workload + dataset

Workload: ResNet-50 on ImageNet-1k
Evaluating routed TPU design w/ approximate cores
Energy, perf. and area evaluated at <10nm PDK

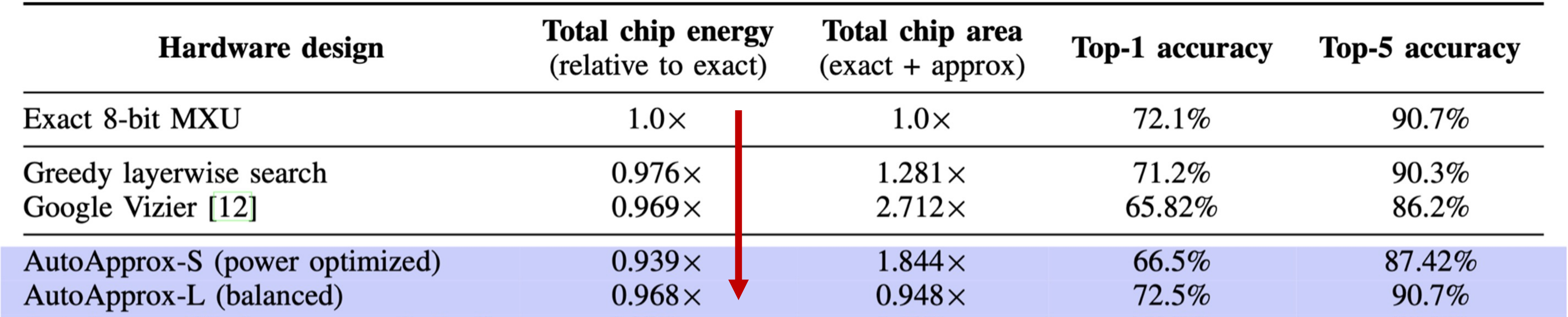


Hardware design	**Total chip energy** (relative to exact)	**Total chip area** (exact + approx)	**Top-1 accuracy**	**Top-5 accuracy**
Exact 8-bit MXU	1.0×	1.0×	72.1%	90.7%
Greedy layerwise search	0.976×	1.281×	71.2%	90.3%
Google Vizier [12]	0.969×	2.712×	65.82%	86.2%

1%-6% lower accuracy than baseline

Results: Evaluating AutoApprox on large-scale workload + dataset

Workload: ResNet-50 on ImageNet-1k
Evaluating routed TPU design w/ approximate cores
Energy, perf. and area evaluated at <10nm PDK

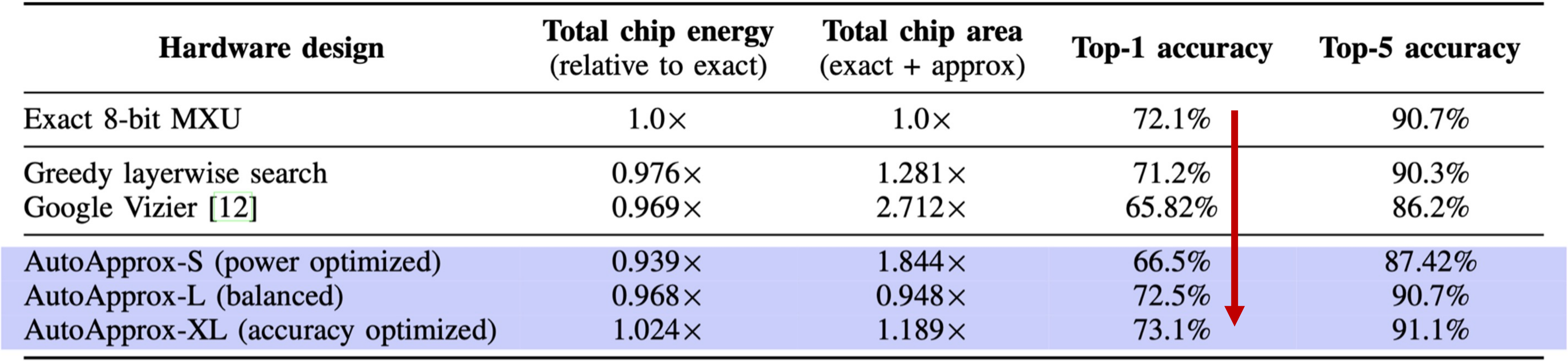


Hardware design	Total chip energy (relative to exact)	Total chip area (exact + approx)	Top-1 accuracy	Top-5 accuracy
Exact 8-bit MXU	1.0×	1.0×	72.1%	90.7%
Greedy layerwise search	0.976×	1.281×	71.2%	90.3%
Google Vizier [12]	0.969×	2.712×	65.82%	86.2%
AutoApprox-S (power optimized)	0.939×	1.844×	66.5%	87.42%
AutoApprox-L (balanced)	0.968×	0.948×	72.5%	90.7%

3.2% - 6.1% energy savings!

Results: Significant energy savings for TPU with zero accuracy loss

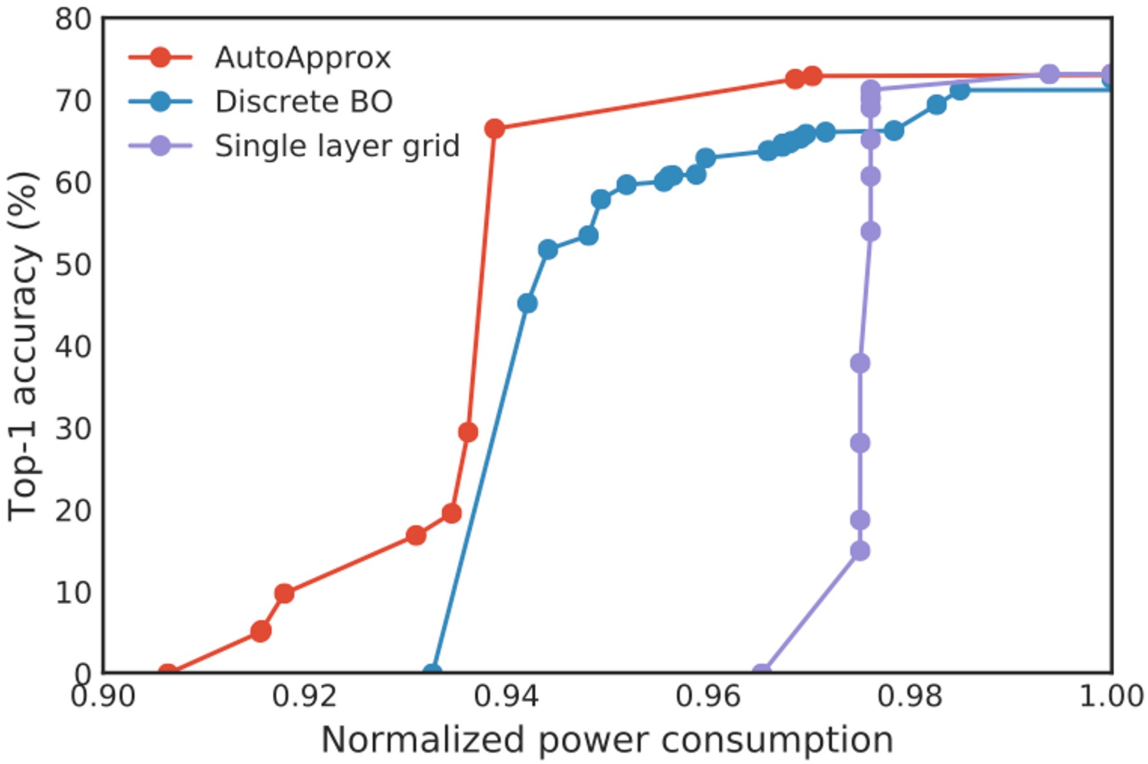
Workload: ResNet-50 on ImageNet-1k
Evaluating routed TPU design w/ approximate cores
Energy, perf. and area evaluated at <10nm PDK



Hardware design	**Total chip energy** (relative to exact)	**Total chip area** (exact + approx)	**Top-1 accuracy**	**Top-5 accuracy**
Exact 8-bit MXU	1.0×	1.0×	72.1%	90.7%
Greedy layerwise search	0.976×	1.281×	71.2%	90.3%
Google Vizier [12]	0.969×	2.712×	65.82%	86.2%
AutoApprox-S (power optimized)	0.939×	1.844×	66.5%	87.42%
AutoApprox-L (balanced)	0.968×	0.948×	72.5%	90.7%
AutoApprox-XL (accuracy optimized)	1.024×	1.189×	73.1%	91.1%

Improve accuracy by 1%

Results: AutoApprox system pareto optimal to baselines



Synthesizing Zero-loss Low-Power Approximate DNN Accelerators with Large-Scale Search

Paras Jain, Safeen Huda, Martin Maas, Joseph Gonzalez, Ion Stoica, Azalia Mirhoseini

Please reach out!
parasj@berkeley.edu

Problem: Leverage DNN tolerance to approximation to improve TPU perf/TCO via approximately accurate circuits.

Approach: Pack heterogenous approximate MXUs as sidekicks to a fallback exact MXU.

Contributions:

- Approx. TPU architecture w/ exact fallback
- Fast e2e accuracy simulation
- ML-guided search

Key results:

- Save up to 6% MXU power end-to-end on real TPU design (<10nm)
- Method significantly outperforms competitive baselines
- Opens new orthogonal avenue for chip efficiency beyond quantization + sparsity