Learning Automatic Schedulers through Projective Reparameterization

Ajay Jain

Saman Amarasinghe

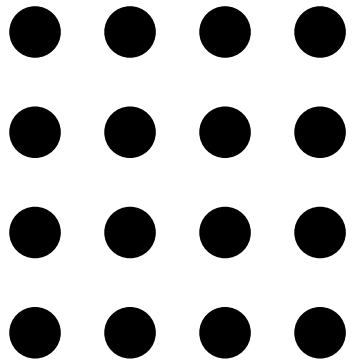
Challenges in ML for Systems

Challenges in ML for Systems

1. Discrete search spaces

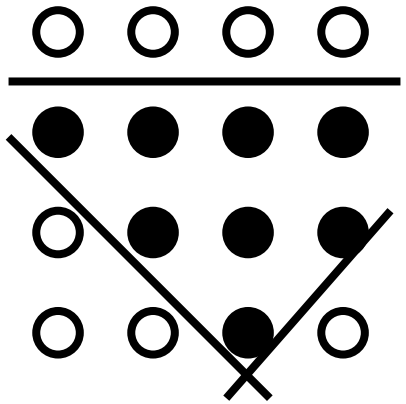
Challenges in ML for Systems

1. Discrete search spaces
2. Large, combinatorial search spaces



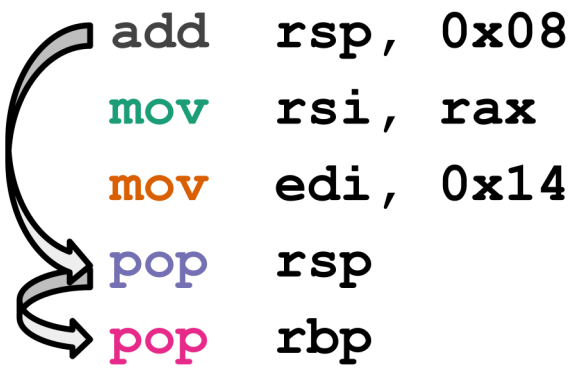
Challenges in ML for Systems

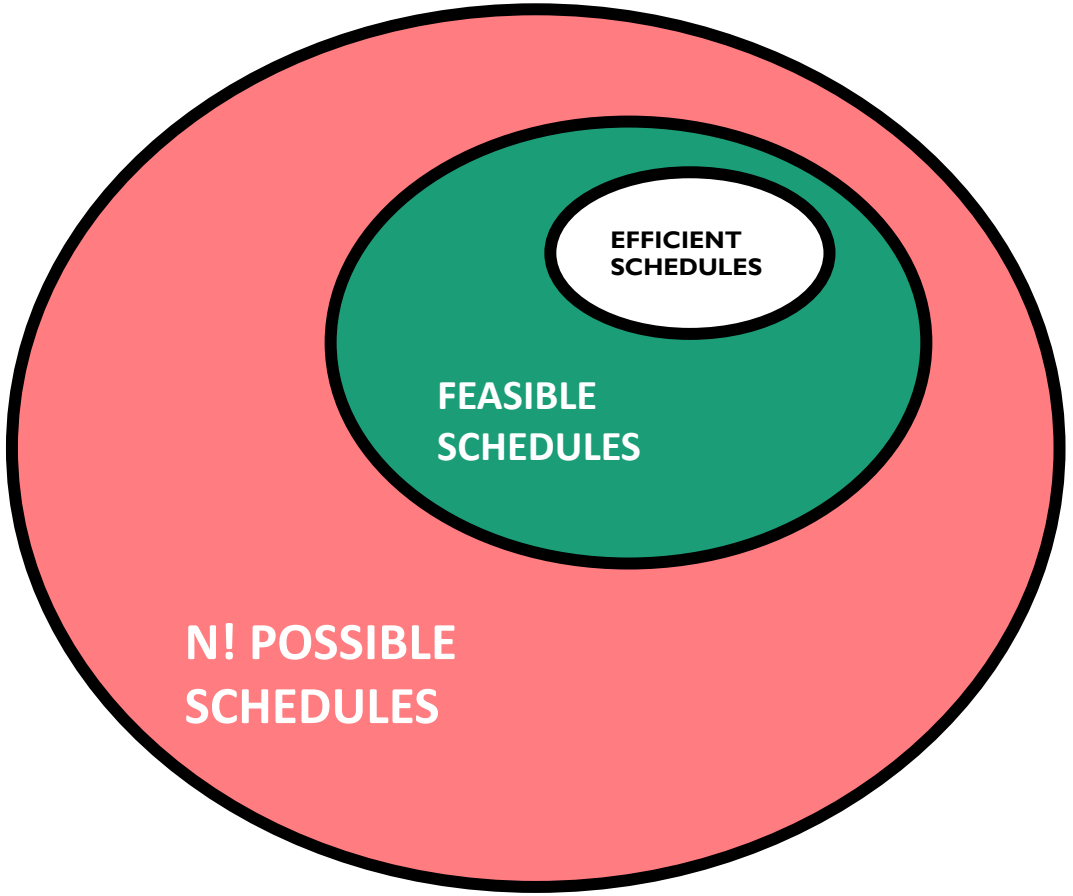
1. Discrete search spaces
2. Large, combinatorial search spaces
3. Highly constrained feasible sets



Challenging for supervised learning!

Motivating application: Instruction Scheduling





Motivating application: Instruction Scheduling

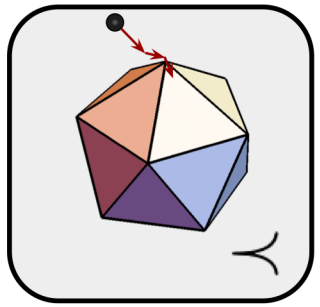
- List scheduling with heuristics
- Stochastic search & superoptimization
- Integer linear programming
- Reinforcement learning

Challenging for supervised learning!

Baseline: Sinkhorn iteration from ranking literature

→ 16% of schedules are invalid

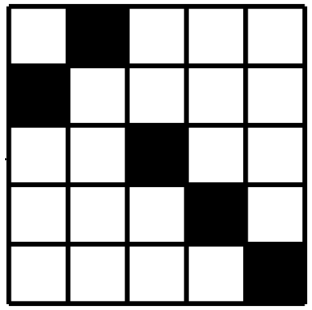
This work



- Introduce EPOCS operator
  - General approach for learning under dynamic constraints
- Formulate instruction scheduling as relaxed integer program
- Imitate GCC compiler instruction schedules

Permutation matrix representation

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 0 & 2 & 3 & 4 \end{pmatrix}$$



ranking, scheduling, packet switching, matching…

Fixed constraints

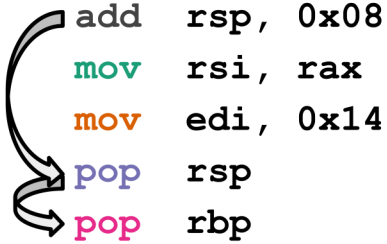
$$\sum_{i=1}^{n} \mathbf{P}_{ij} = 1$$

$$\sum_{j=1}^{n} \mathbf{P}_{ij} = 1$$

$$\mathbf{P}_{ij} \in \{0, 1\} \quad \text{(relaxed: } \mathbf{P}_{ij} \geq 0\text{)}$$

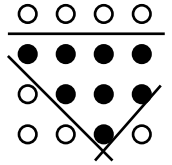
Input dependent partial order constraints

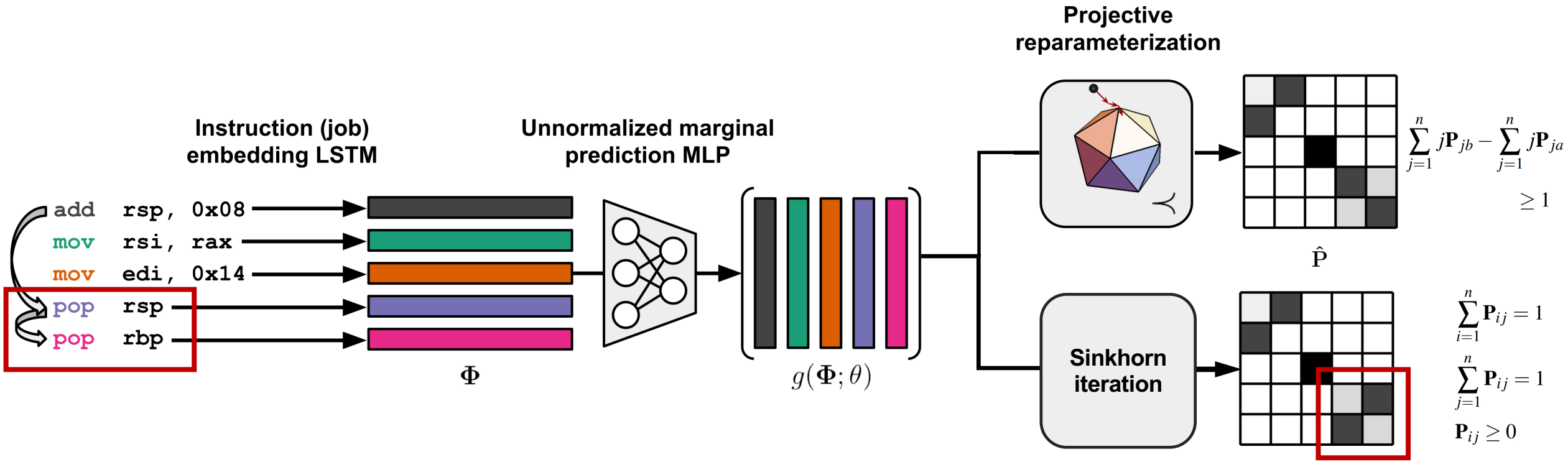
$$\sum_{j=1}^{n} j\mathbf{P}_{jb} - \sum_{j=1}^{n} j\mathbf{P}_{ja} \geq 1 \text{ if } x_a \prec x_b$$


add rsp, 0x08
mov rsi, rax
mov edi, 0x14
pop rsp
pop rbp

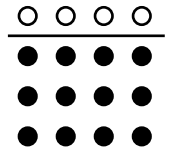
EPOCS operator

Fixed constraints
Dynamic constraints





Fixed constraints
Dynamic constraints



Algorithm 1: EPOCS provides a differentiable projective reparameterization of general dynamic constraints.

function EPOCS $(\mathbf{G}, \mathbf{A}, \vec{b})$;

Input : $\mathbf{G} \in \mathbb{R}^{n \times n}$, $\mathbf{A} \in \mathbb{R}^{m \times n^2}$, $\vec{b} \in \mathbb{R}^m$

Output : $\hat{\mathbf{G}} \in \mathbb{R}^{n \times n}$ such that $\text{vec}(\hat{\mathbf{G}})^T \mathbf{A}^T \leq \vec{b}$

$\hat{g} = \text{vec}(\mathbf{G})$;

for *it* $= 1$ *to* MAXITERS **do**

  for $c = 1$ *to* m **do**

    if $\langle \vec{a}_c, \hat{g} \rangle > b_c$ **then**

      // Project onto affine half-space boundary

      $\vec{u}_c = \vec{a}_c * \frac{b_c}{\langle \vec{a}, \vec{a} \rangle}$;

      $\hat{g} = \vec{u}_c + \text{proj}_{\vec{a}_c}(\hat{g} - \vec{u}_c)$;

    end

  end

end

return $\text{vec}^{-1}(\hat{g})$;

One iteration of EPOCS for scheduling

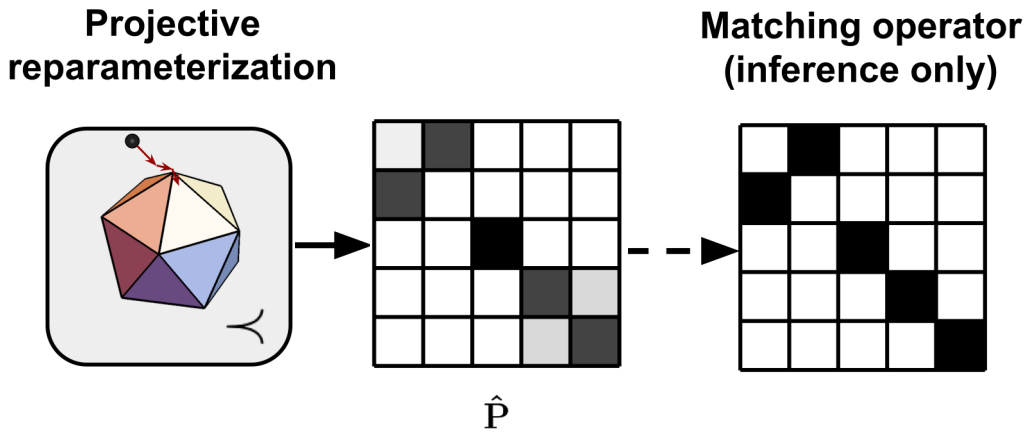
$$\sum_{i=1}^{n} \mathbf{P}_{ij} = 1$$ 1) Normalize*

$$\sum_{j=1}^{n} \mathbf{P}_{ij} = 1$$ 2) Normalize*

$$\mathbf{P}_{ij} \geq 0$$ 3) ReLU

$$\sum_{j=1}^{n} j\mathbf{P}_{jb} - \sum_{j=1}^{n} j\mathbf{P}_{ja} \geq 1 \text{ if } x_a \prec x_b$$ 4) Project

Correct the relaxation with matching
(Hungarian algorithm)

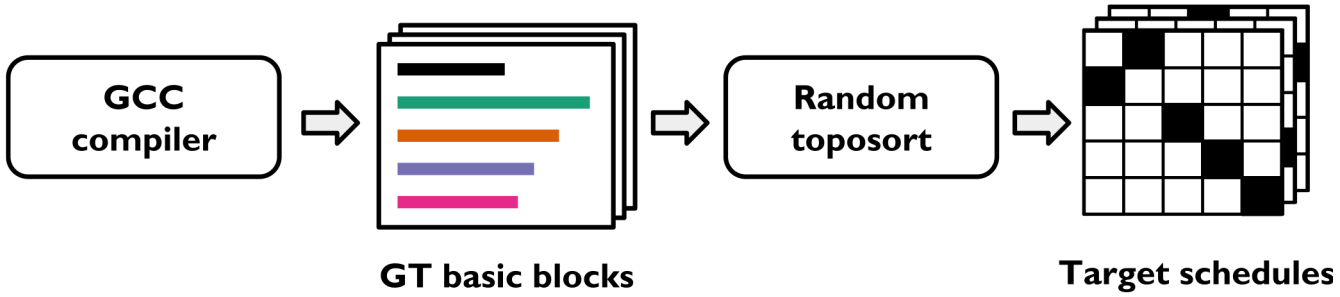


$$M(\hat{\mathbf{P}}) = \max_{\mathbf{P} \in \mathcal{P}_n} \langle \hat{\mathbf{P}}, \mathbf{P} \rangle_H$$

Evaluation

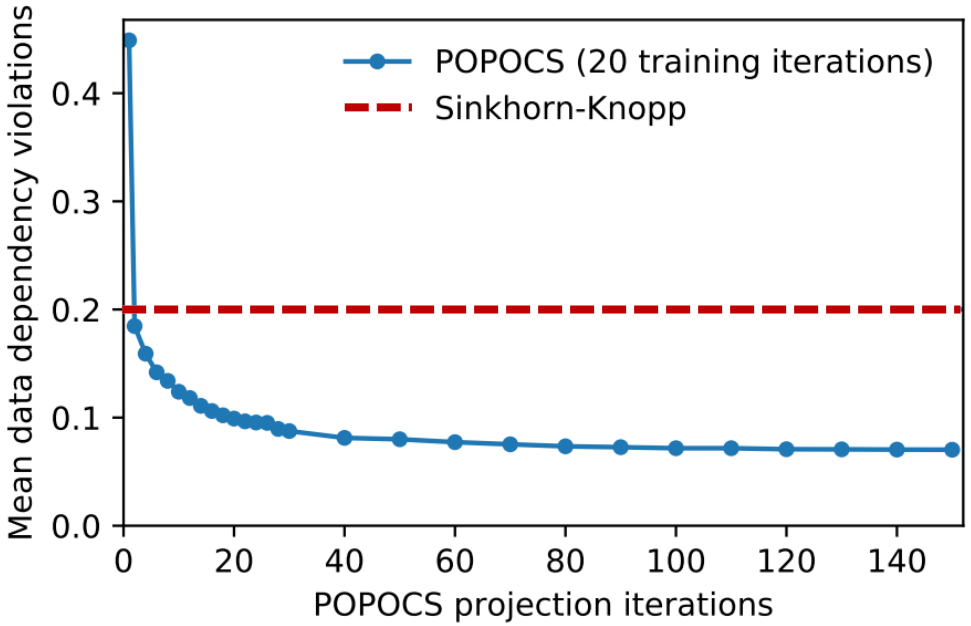
Train POCSNet to imitate GCC 4.9.4 schedules

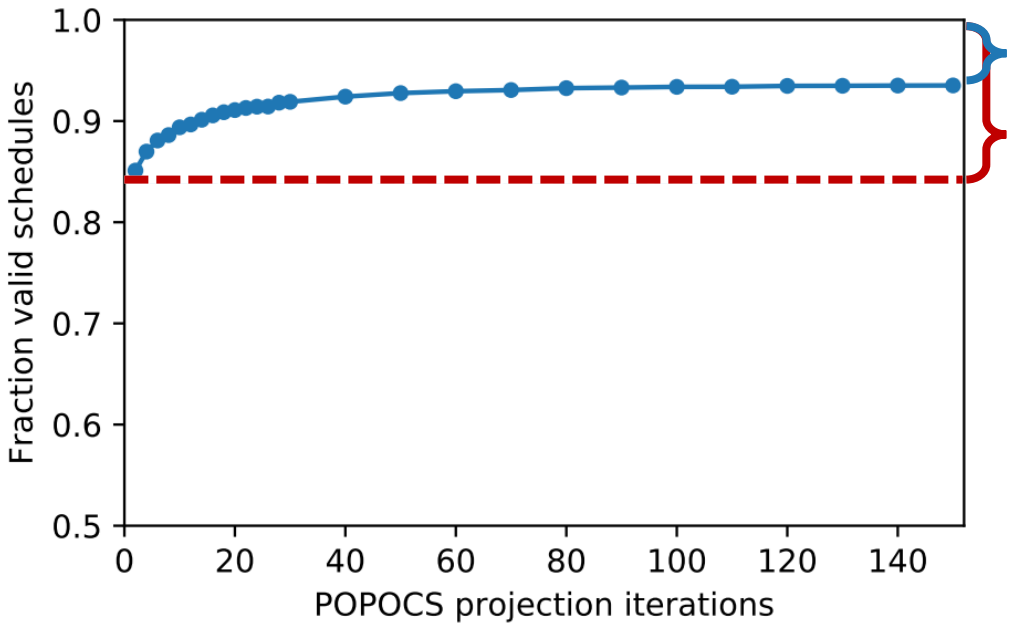
77,202 basic blocks from SPEC2006, SPEC2017 [Mendis et al 2019]



Evaluate data dependency violations, accuracy

Baseline: Sinkhorn iteration

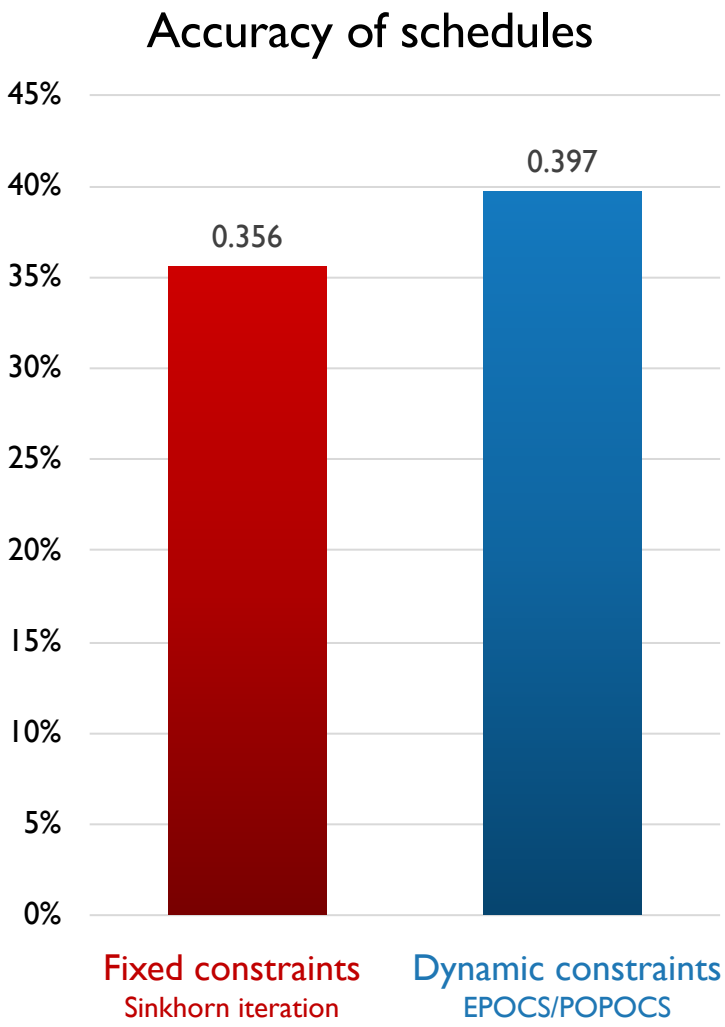




+ Dynamic constraints
with EPOCS/POPOCS

Fixed constraints only
16% of predicted schedules are infeasible

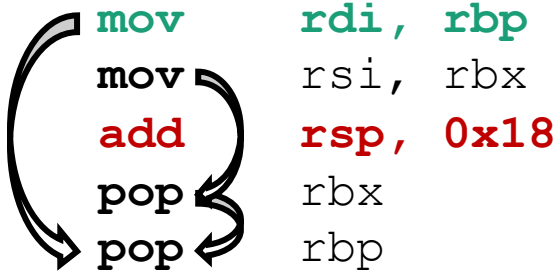
Imposing dynamic constraints reduces data dependency violations



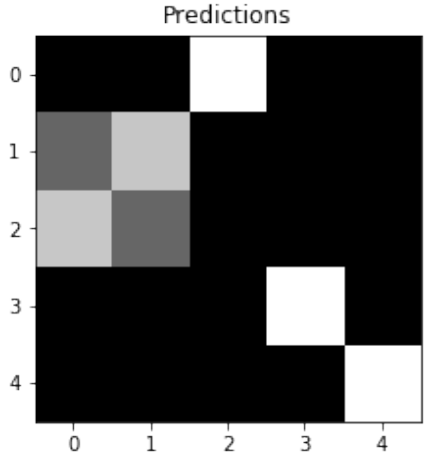
	Fixed constraints Sinkhorn iteration	**Dynamic constraints** EPOCS/POPOCS
Accuracy	35.6%	<u>39.7%</u>
Kendall tau distance	0.238	<u>0.222</u>

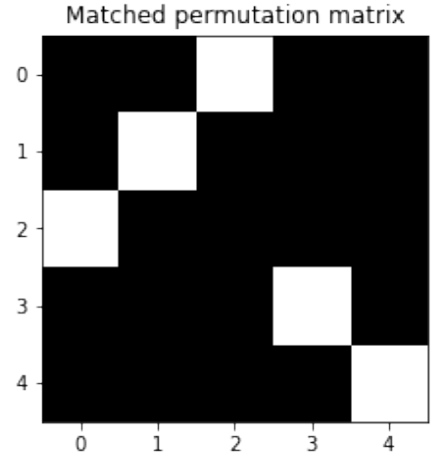
Imposing constraints improves accuracy (+4%)

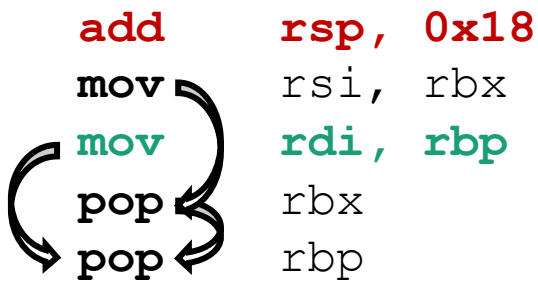
POCSNet schedule latencies are on par with GCC latencies



Shuffled input block

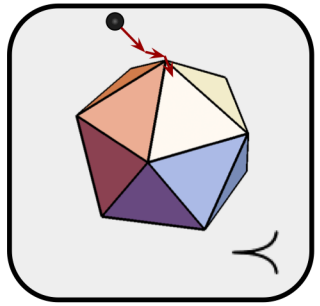






POCSNet scheduled block

Takeaways



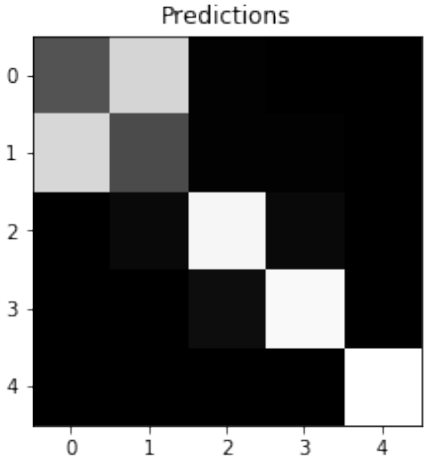
- EPOCS: General purpose op for dynamic constraints on NNs
- One application: Job scheduling problems
- A step toward correct-by-construction ML for Systems:

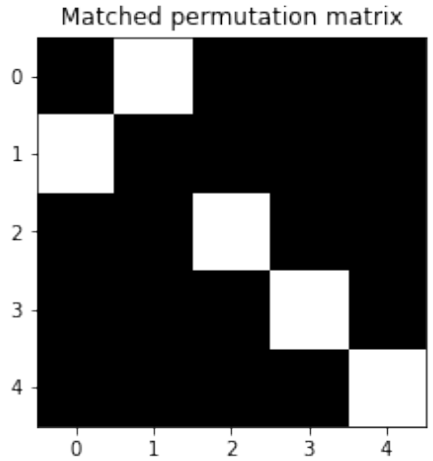
  Enforce known constraints end-to-end for accuracy boost + guarantees

Contact: ajayj@berkeley.edu

```
movsd   xmm3, qword ptr [r13+0xa0]
movsd   xmm5, qword ptr [rsp+0x20]
subsd   xmm3, xmm5
divsd   xmm3, xmm8
ucomisd xmm2, xmm3
```

Shuffled input block






```
movsd   xmm5, qword ptr [rsp+0x20]
movsd   xmm3, qword ptr [r13+0xa0]
subsd   xmm3, xmm5
divsd   xmm3, xmm8
ucomisd xmm2, xmm3
```

POCSNet scheduled block