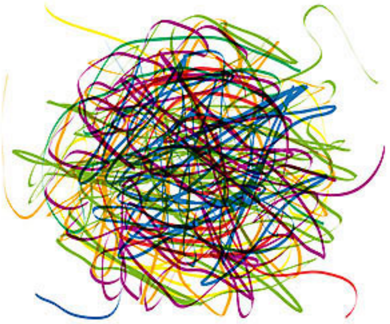
Ordering Chaos
Memory-Aware Scheduling for Irregularly Wired Neural Networks on Edge Devices



Byung Hoon Ahn, Jinwon Lee, Jamie Lin, Hsin-Pai Cheng, Jilei Hou, Hadi Esmaeilzadeh

Qualcomm

UC San Diego

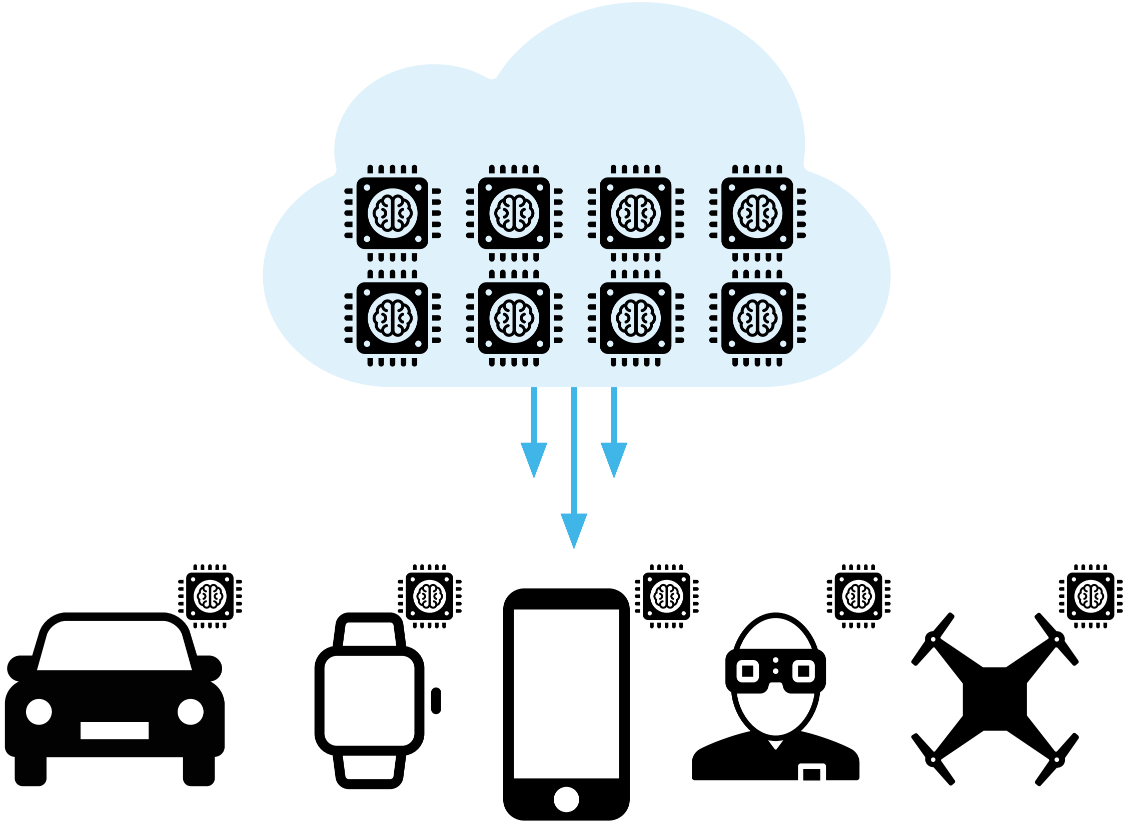
Motivation: Enabling Intelligence, Transition from Cloud to Edge

Intelligence moving
from the Cloud to the Edge

Low Latency

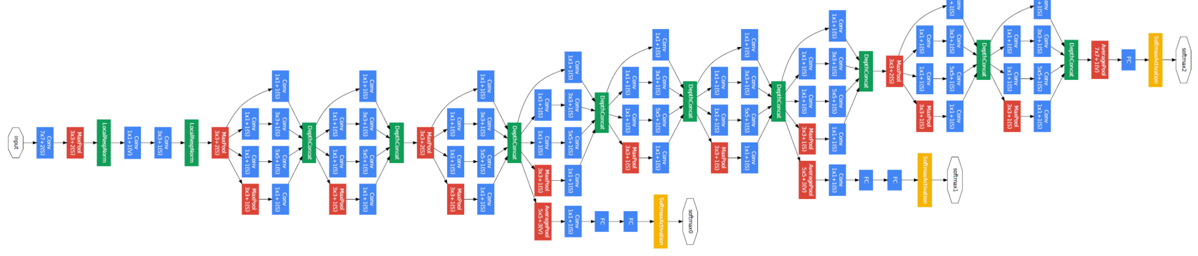
Privacy

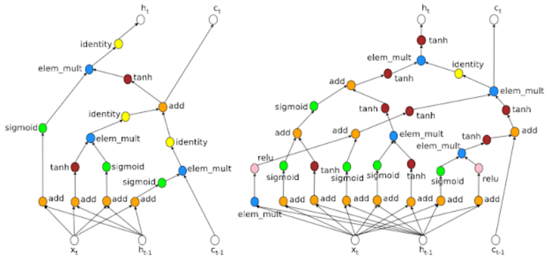
Reliability

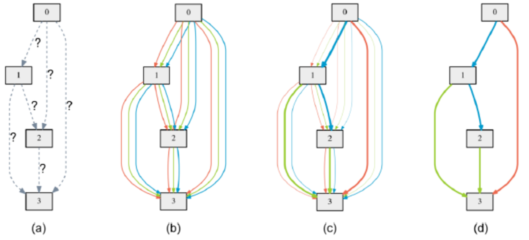


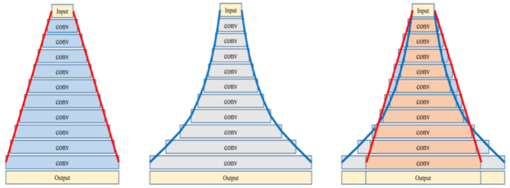
Intelligence is moving from **Cloud to Edge**
for **Low Latency**, **Privacy**, and **Reliability**

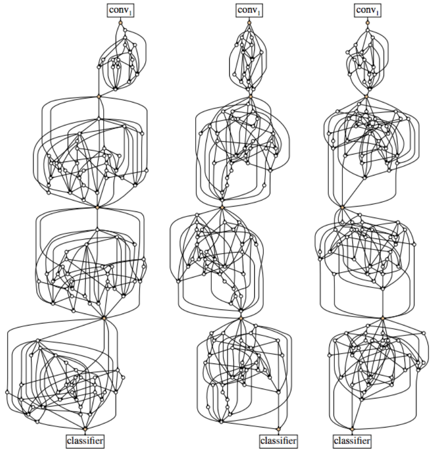
Motivation: How to Make Deep Neural Networks More Efficient?

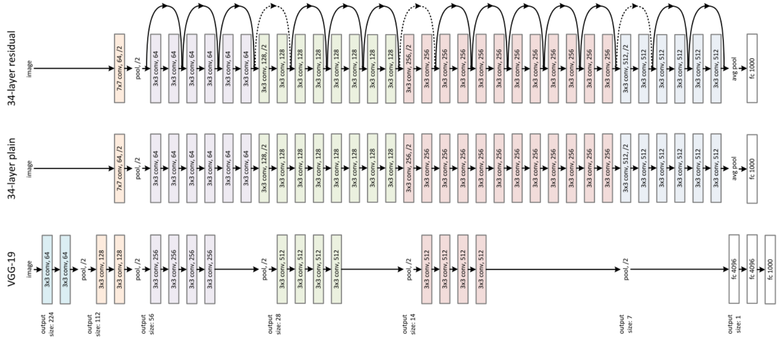


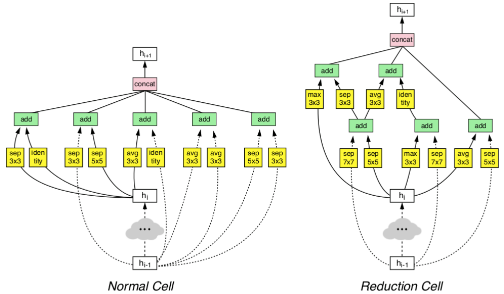


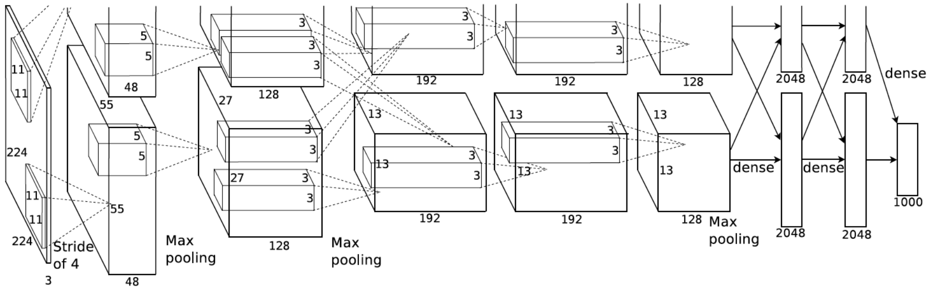


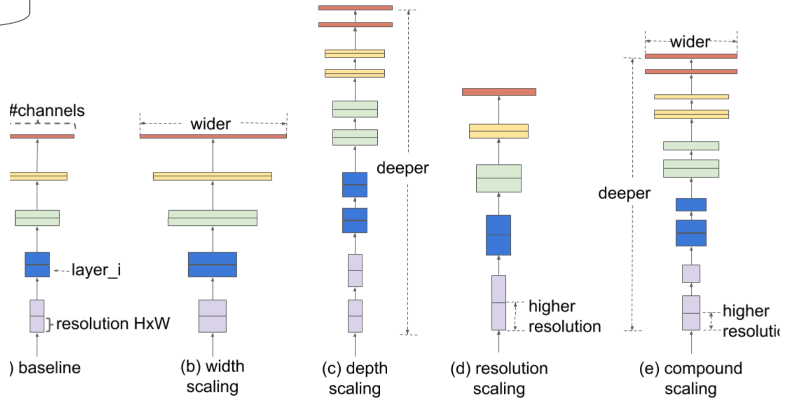


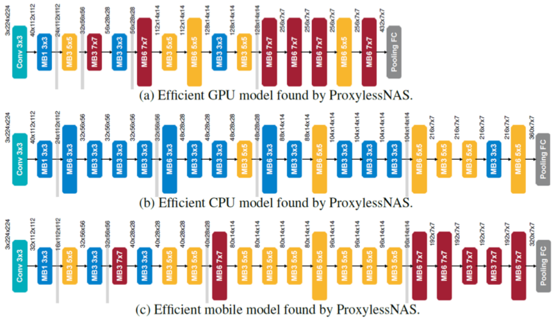




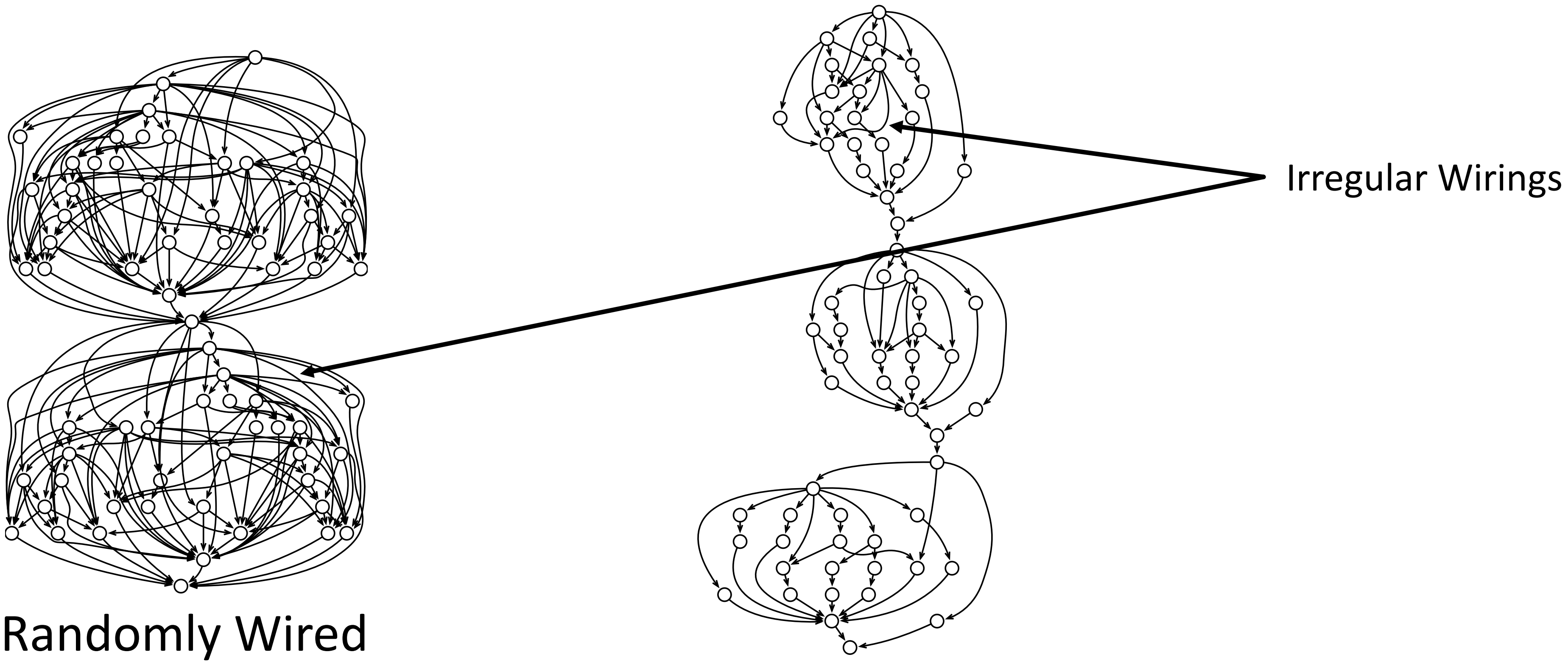








Motivation: Irregularly Wired Neural Networks



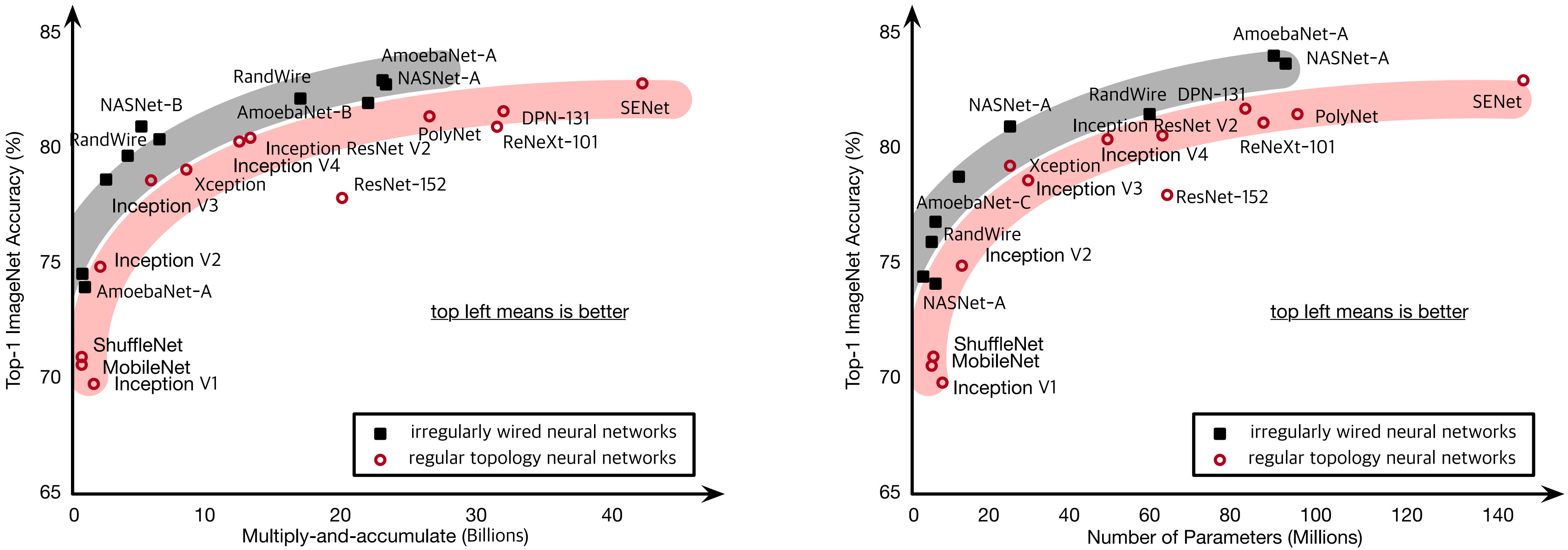
Randomly Wired Neural Network [ICCV'19]

SwiftNet [ICCV-W'19]

These **Efficient Networks** comprise of many **Irregular Wirings**

We classify them as **Irregularly Wired Neural Networks**

Motivation: Emerging Class of DNNs for Resource Constrained Scenarios

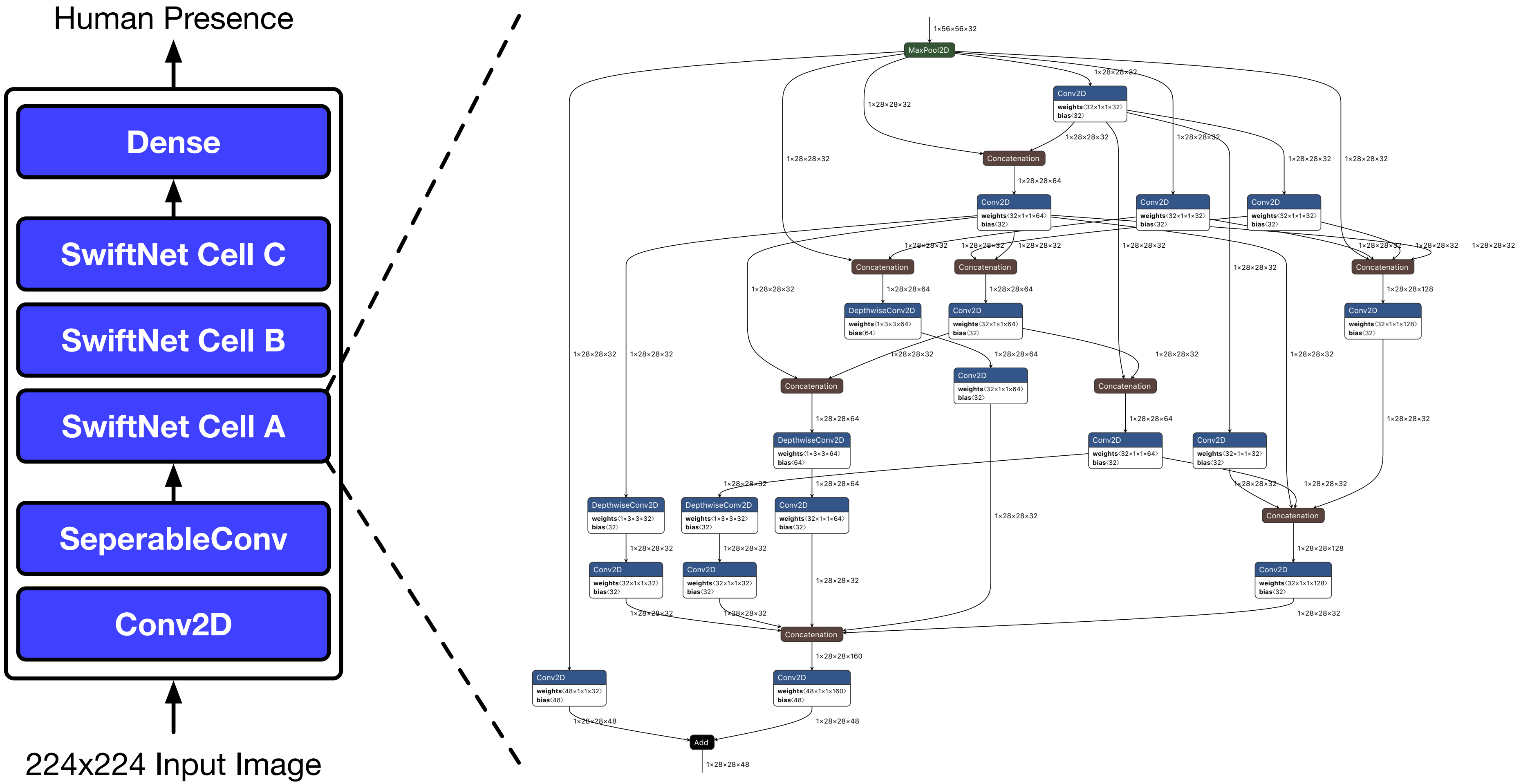


Certain class of networks require **less Resources for same Accuracy**

(a.k.a. **More Efficient Networks**)

Running Example: SwiftNet (ICCV-W'19)

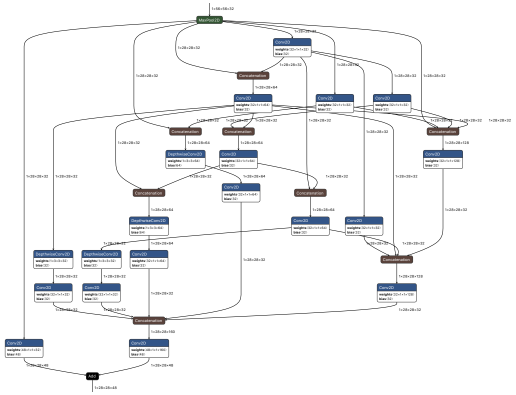
Size (8bits)	MACs	Peak Mem	ACC
249.7KB	57.4M	?	95.13%

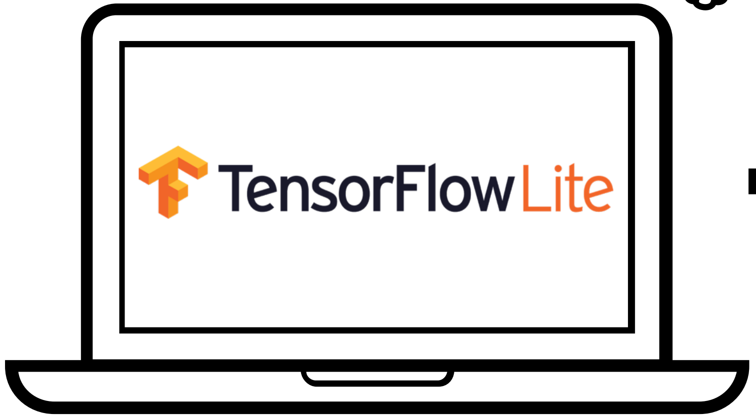


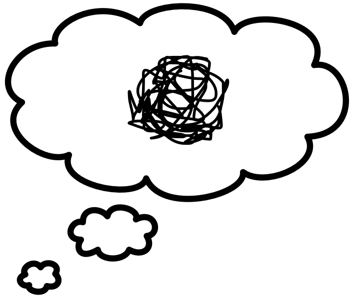
SwiftNet: Using Graph Propagation as Meta-knowledge to Search Highly Representative Neural Architectures: hsinpaic@qti.qualcomm.com; dave.cheng@duke.edu

Running Example: SwiftNet (ICCV-W'19)

Size (8bits)	MACs	Peak Mem	ACC
249.7KB	57.4M	**800KB?**	95.13%







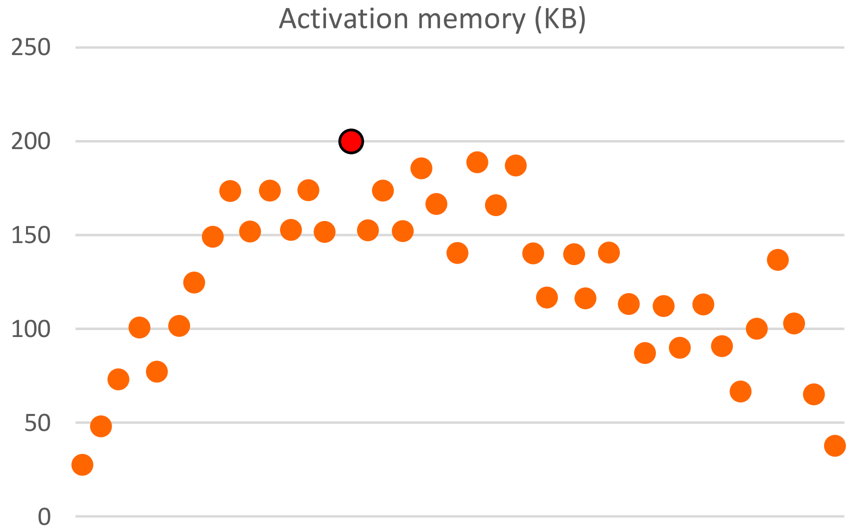
Peak Memory Footprint:
800KB (> 250KB Requirement)

Today's Frameworks are **Oblivious to "Peak Memory Footprint" Issue**
When it come to **Irregularly Wired Neural Networks**

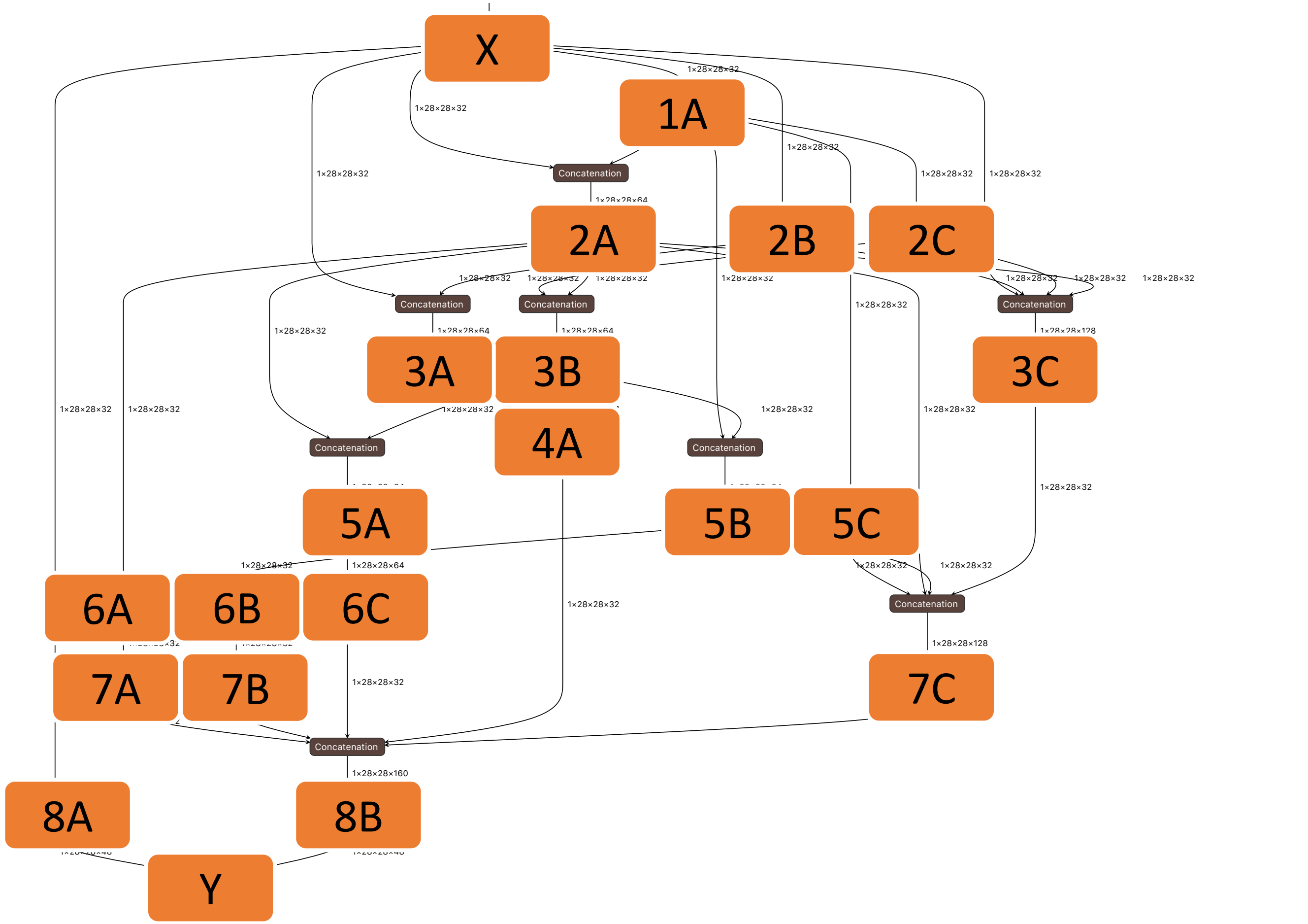
SwiftNet: Using Graph Propagation as Meta-knowledge to Search Highly Representative Neural Architectures: hsinpaic@qti.qualcomm.com; dave.cheng@duke.edu

Running Example: SwiftNet (ICCV-W'19)

Size (8bits)	MACs	Peak Mem	ACC
249.7KB	57.4M	200KB	95.13%



4x improvement in
Peak memory footprint
(c.f., today's TF Lite scheduler = 800KB)



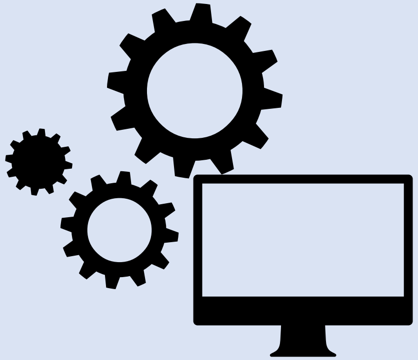
SwiftNet: Using Graph Propagation as Meta-knowledge to Search Highly Representative Neural Architectures: hsinpaic@qti.qualcomm.com; dave.cheng@duke.edu

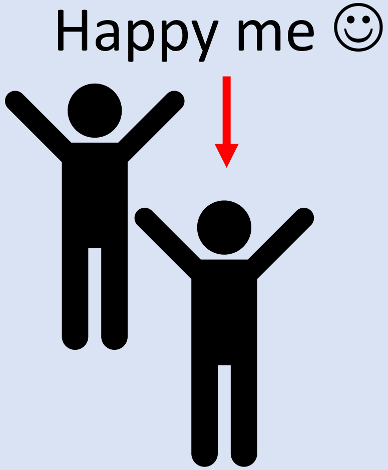
We cannot rely on human expert for scheduling all the time

Manual Work

vs

Automation



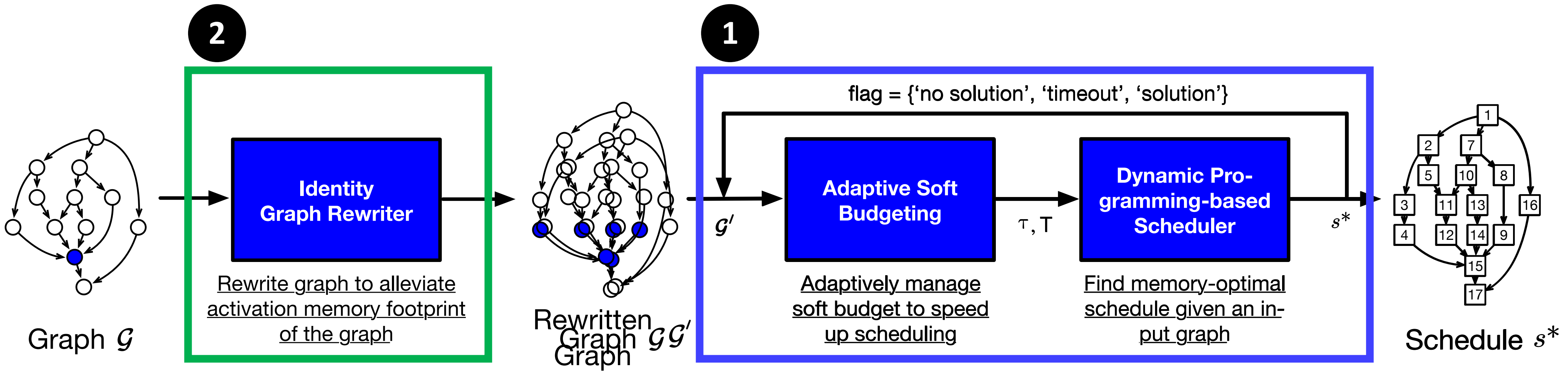


Laziness drives innovations that improve productivity

- Steven Shapiro

Our Solution

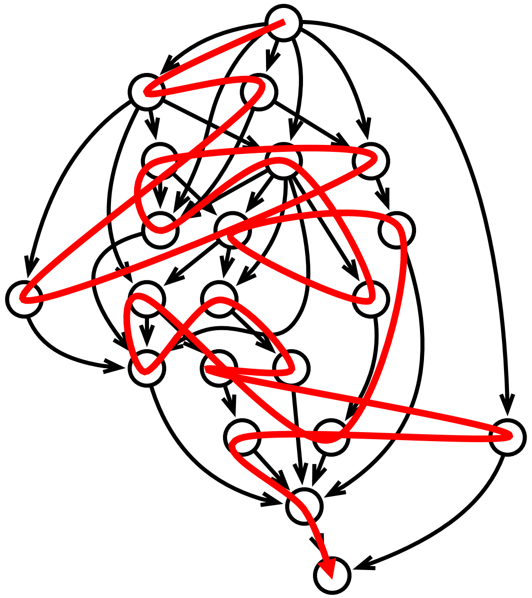
Automated Solution: Serenity (Ordering Chaos)



We propose an Automated Approach that:

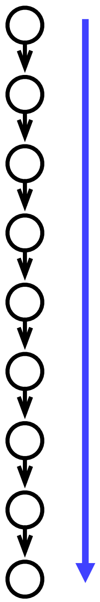
1. Quickly **finds a memory-optimal schedule** for a fixed graph
2. Explores another dimension that **alleviates the memory footprint** of the graph

Search Space: Scheduling = Topological Ordering

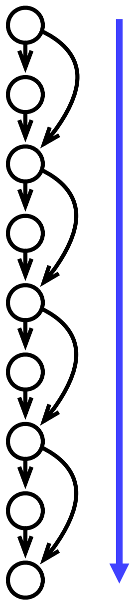


SwiftNet Cell A

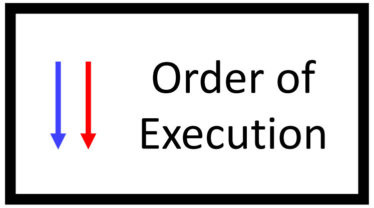
VS



AlexNet, VGGNet, …

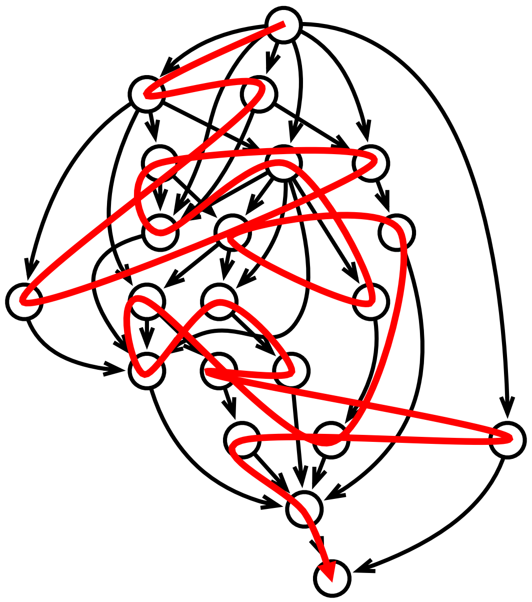


ResNet

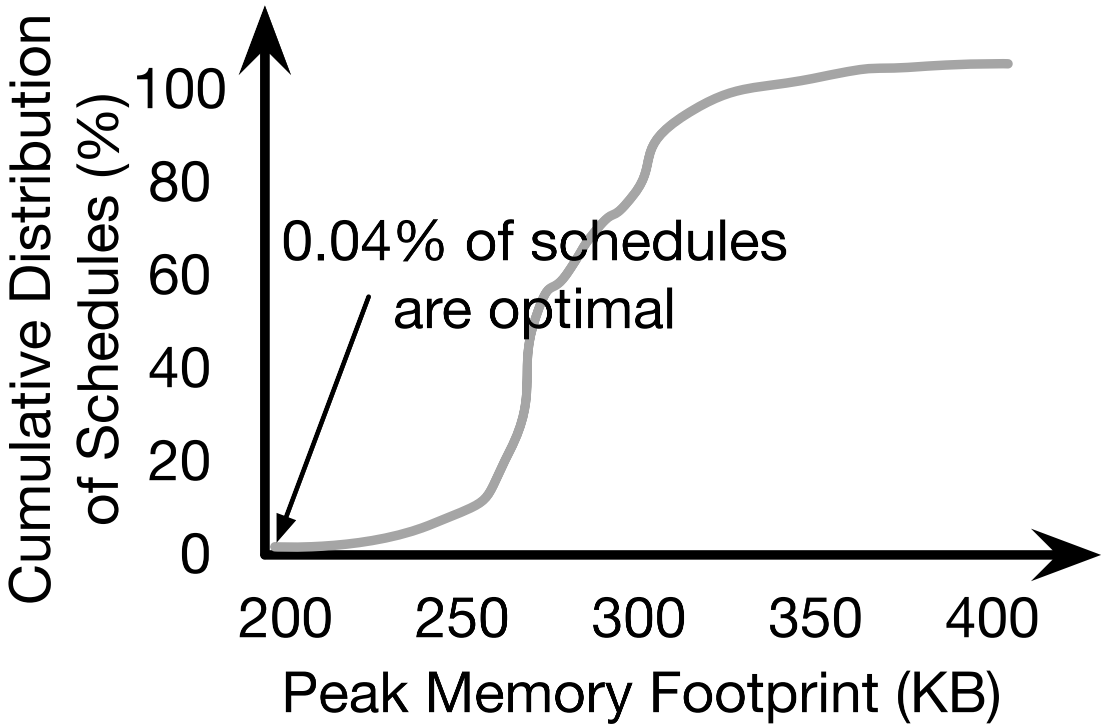


While **Conventional Network (e.g. AlexNet, …)** execution is **"streamlined"**
Irregularly Wired Neural Network execution is **"not streamlined"**

Search Space: Scheduling = Topological Ordering

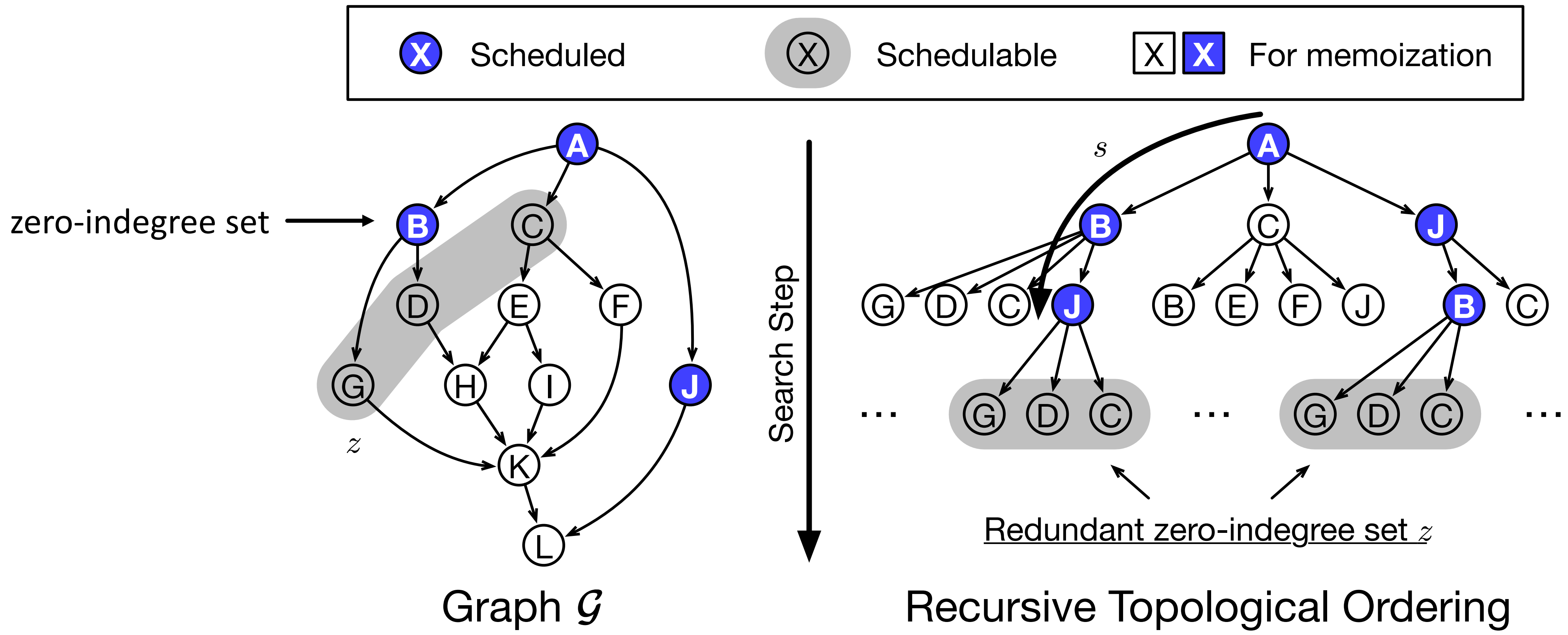


SwiftNet Cell A



Search space is **exponentially large** and
Optimal solutions account for **very very small fraction** of the entire space

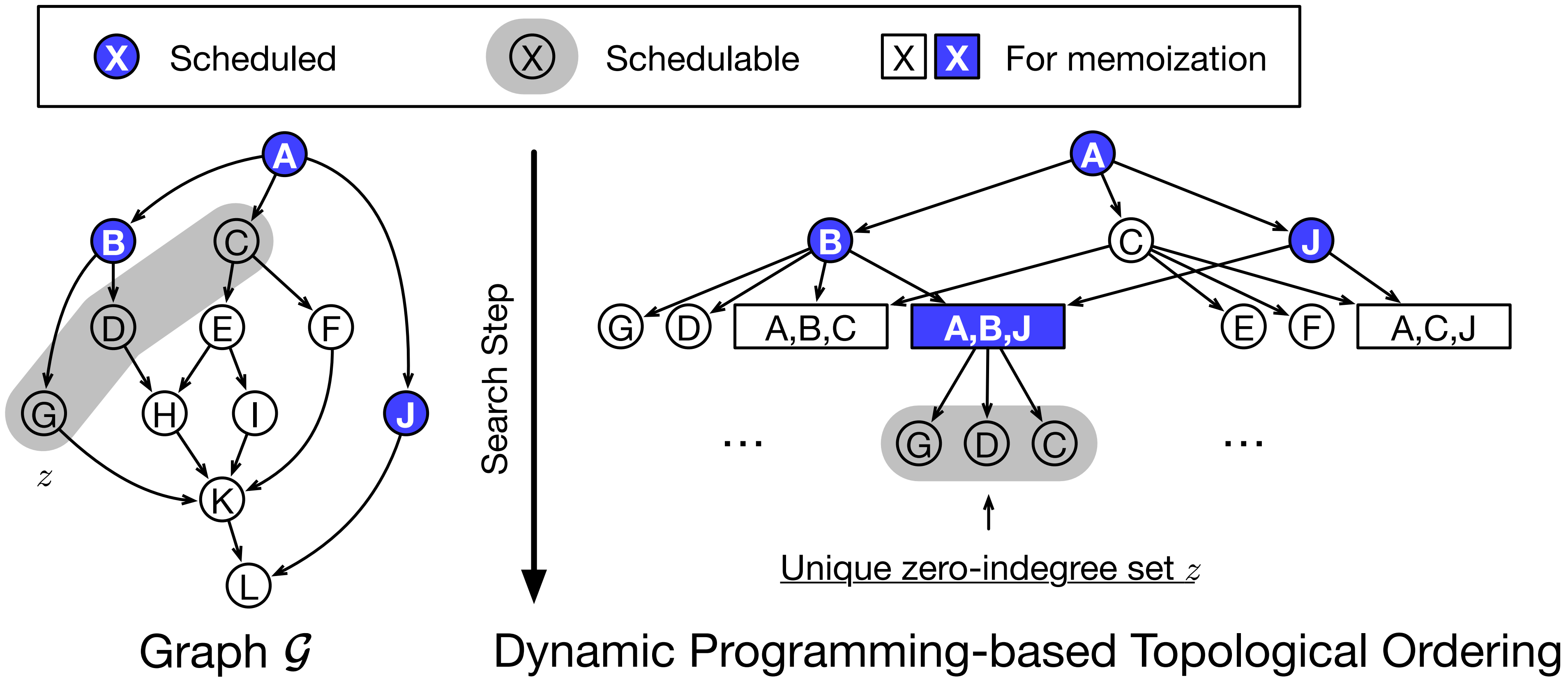
Brute Force Algorithm for Topological Ordering



Many **zero-indegree sets** are **redundant**

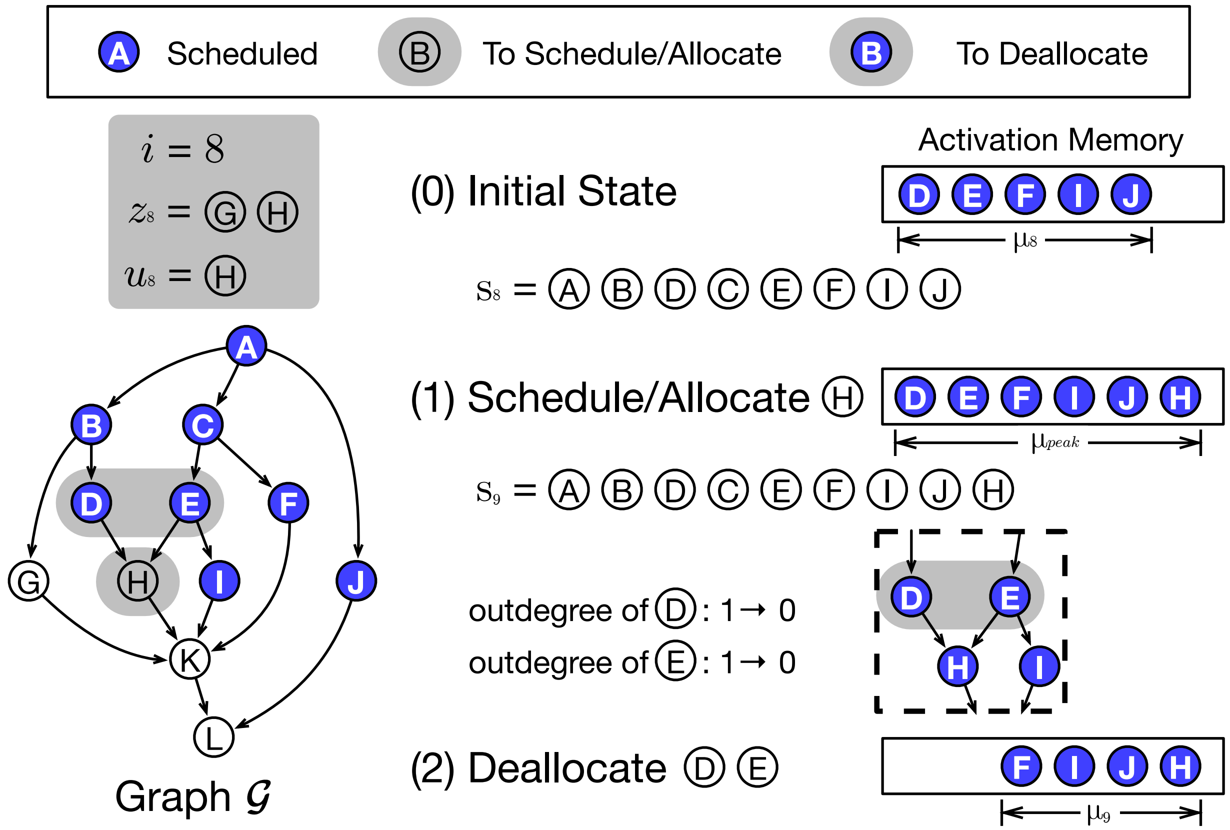
Optimizing this **eliminates redundancy**

Dynamic Programming Algorithm for Topological Ordering



Dynamic Programming-based Topological Ordering
can **speed-up the traversal of schedules significantly**

Overlaying Problem Constraints



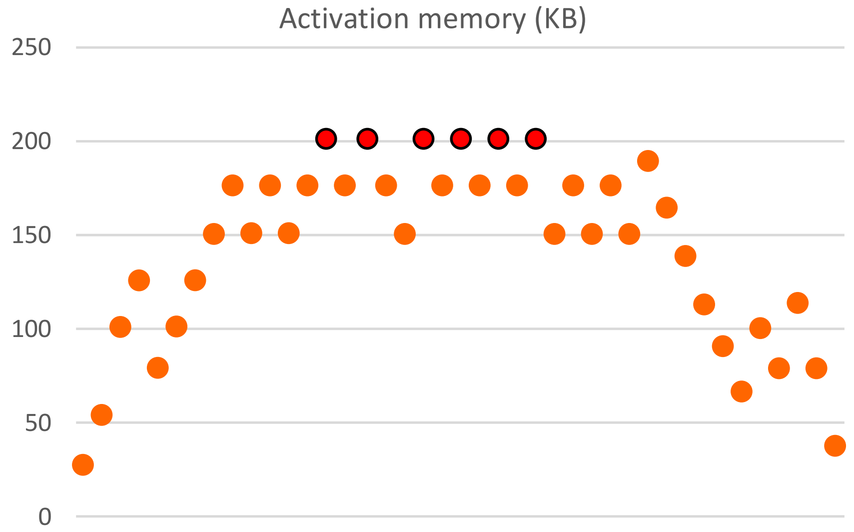
Overlaying these constraints gives

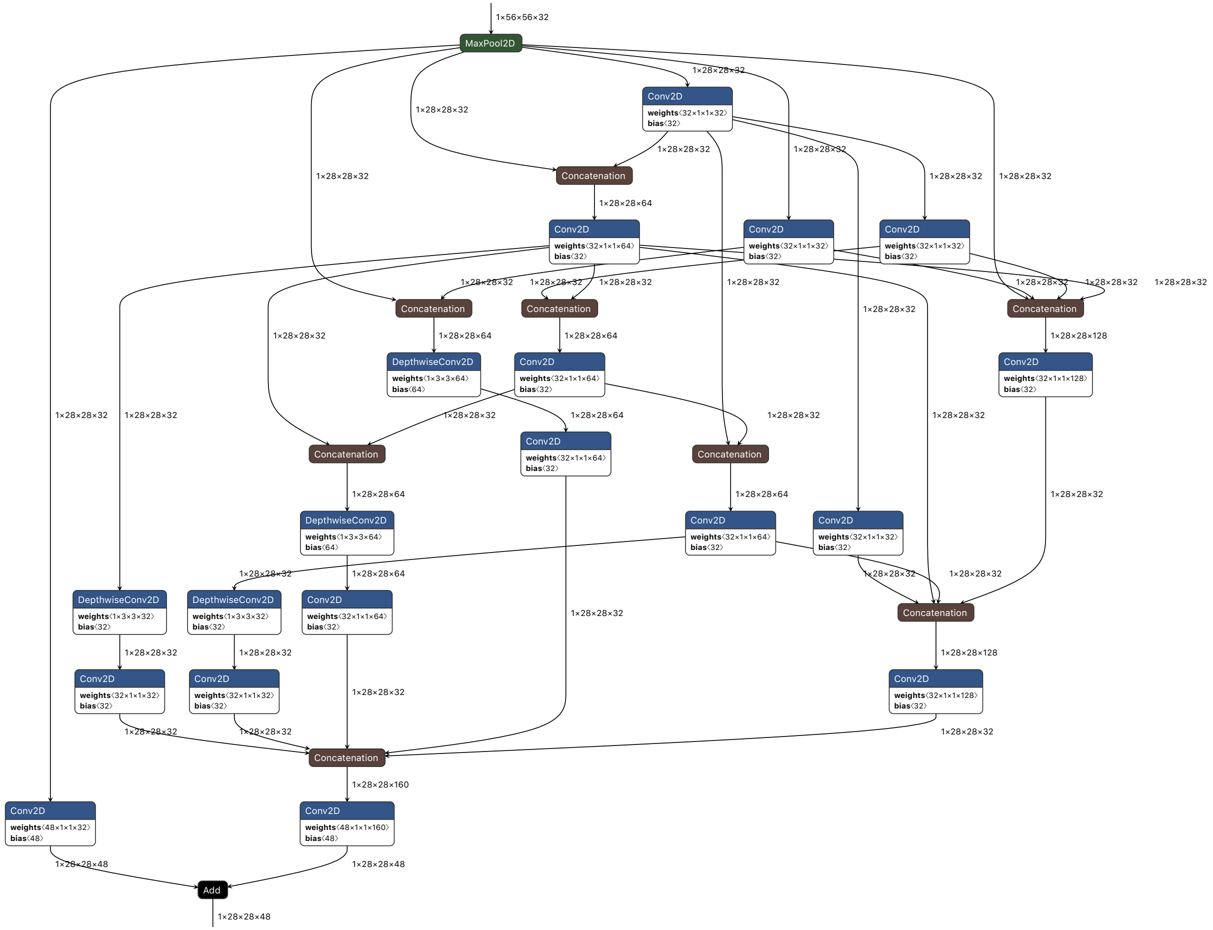
Memory-optimal schedule of the nodes

Dynamic Programming-based Scheduling

Output Activations
In memory

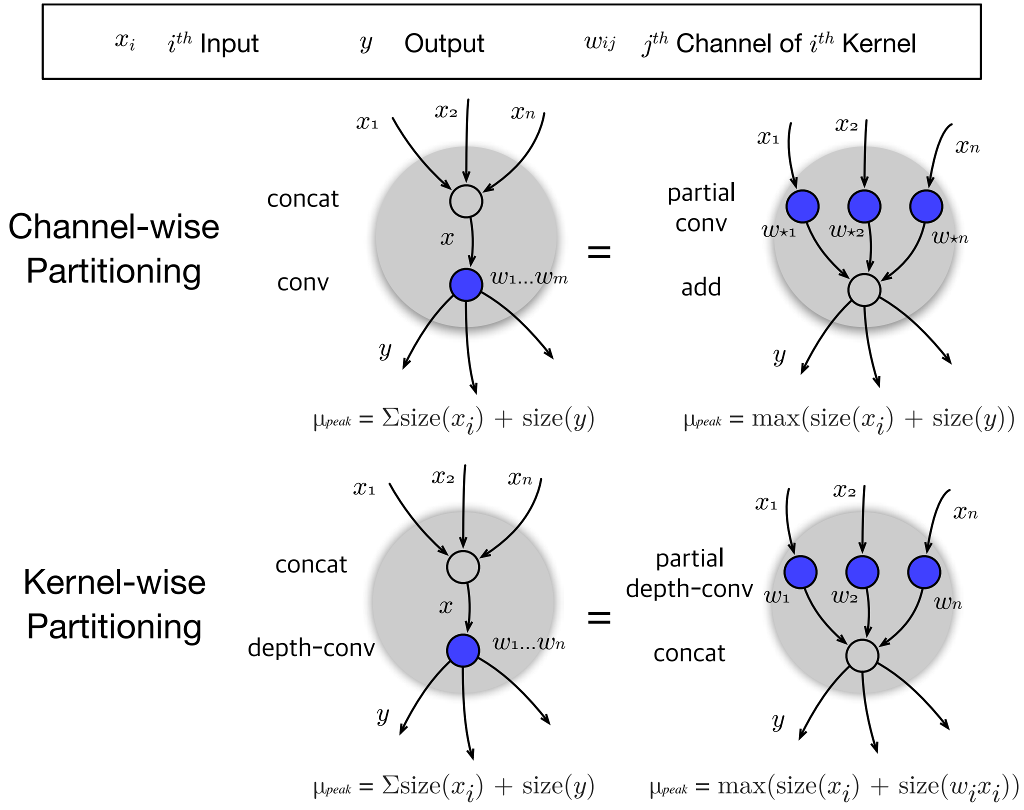
Size (8bits)	MACs	Peak Mem	ACC
249.7KB	57.4M	200KB	95.13%





4x Improvement in
Peak Memory Footprint
(c.f., today's TF Lite scheduler = 800KB)

Identity Graph Rewriting

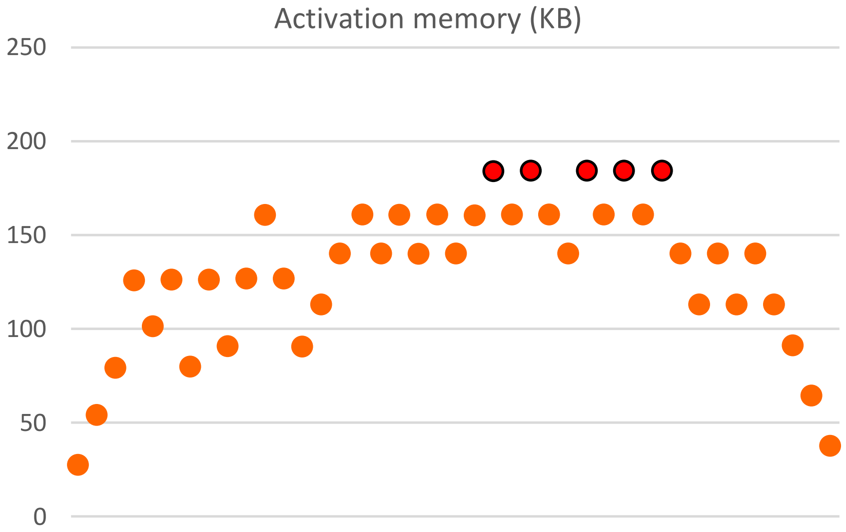


Graph Rewriting while **maintaining the mathematical integrity** allows further **reduction in Peak Memory Footprint**

Dynamic Programming-based Scheduling + Graph Rewriting

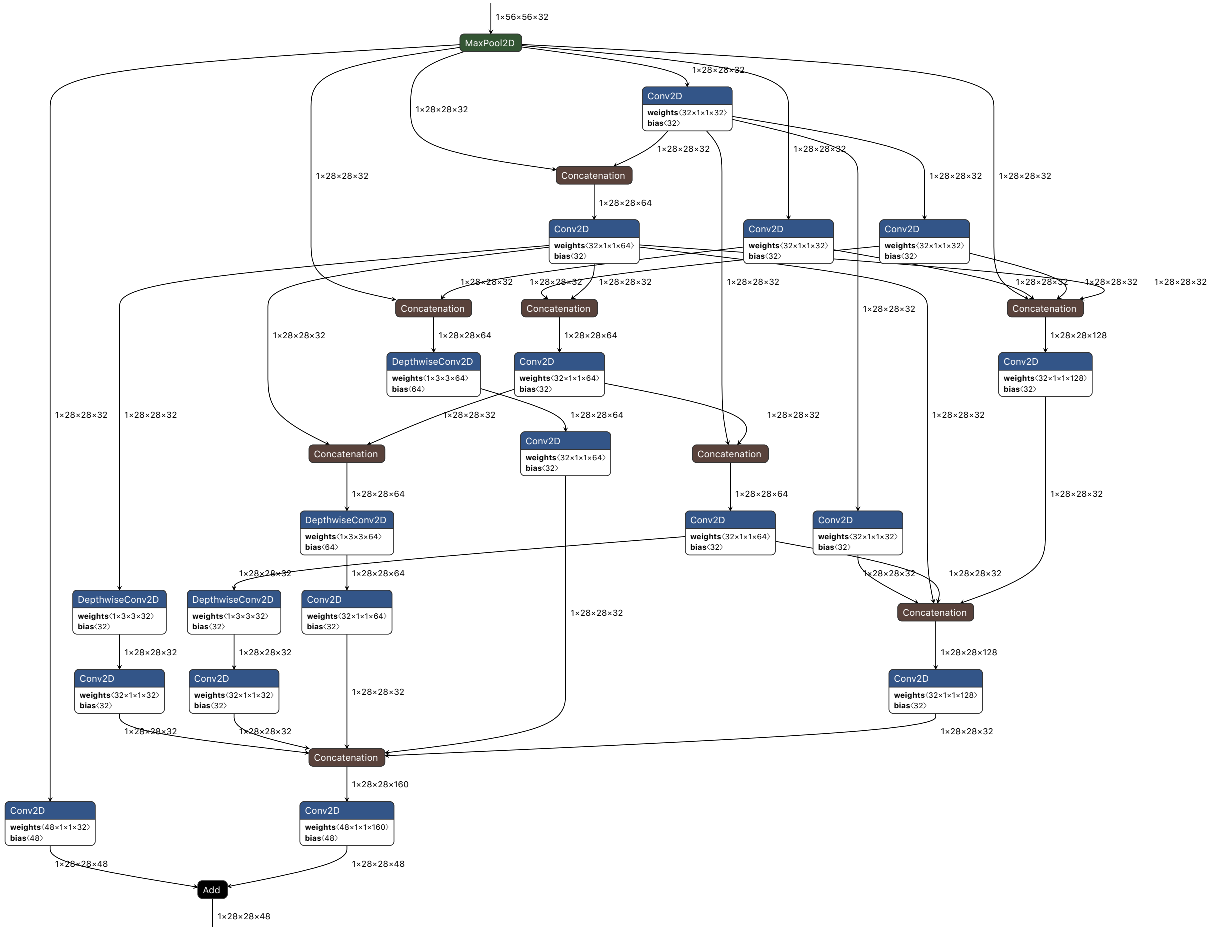
Output Activations In memory

Size (8bits)	MACs	Peak Mem	ACC
249.7KB	57.4M	188KB	95.13%

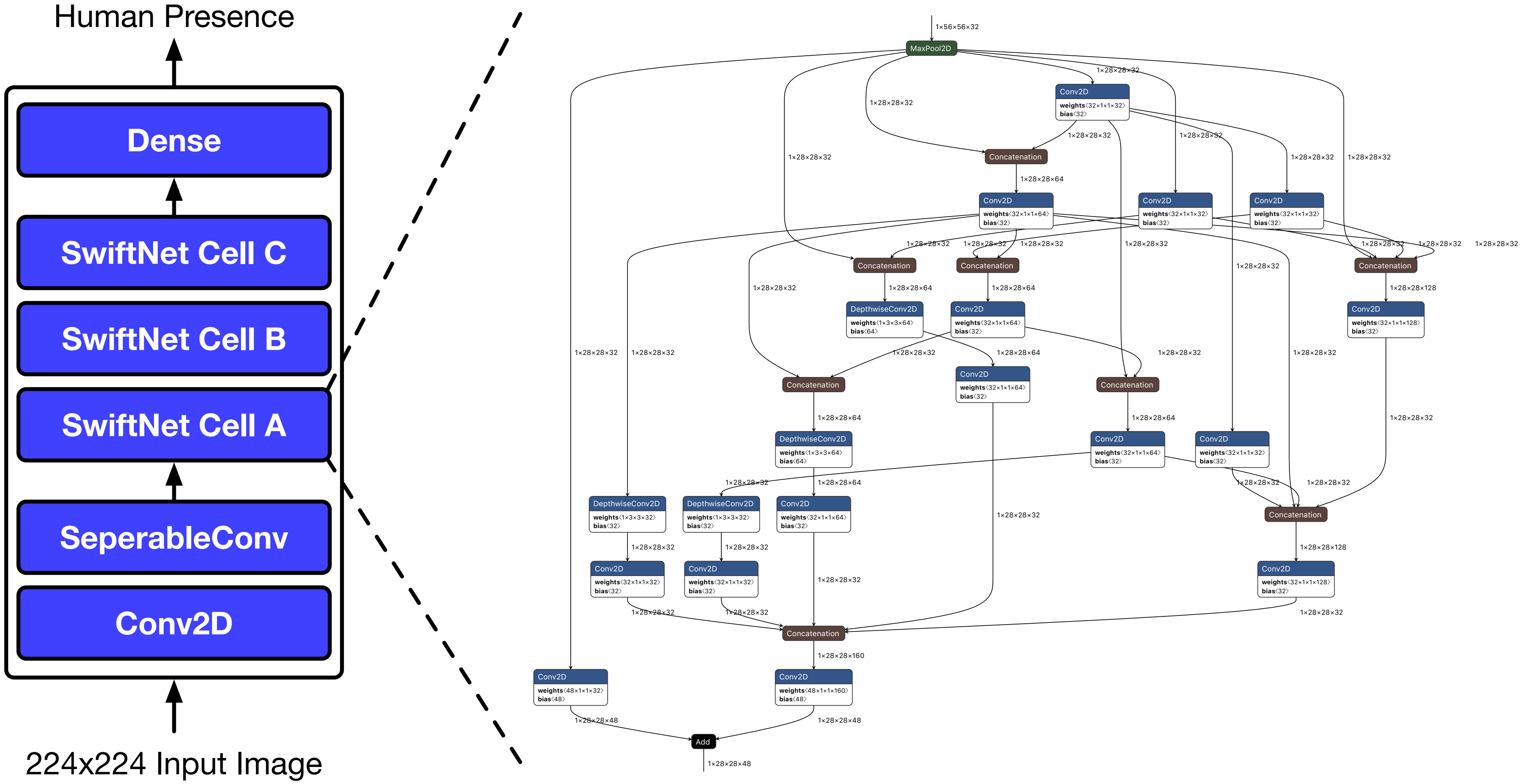


12KB Further Improvement with Graph Rewriting

(c.f., today's TF Lite scheduler = 800KB)

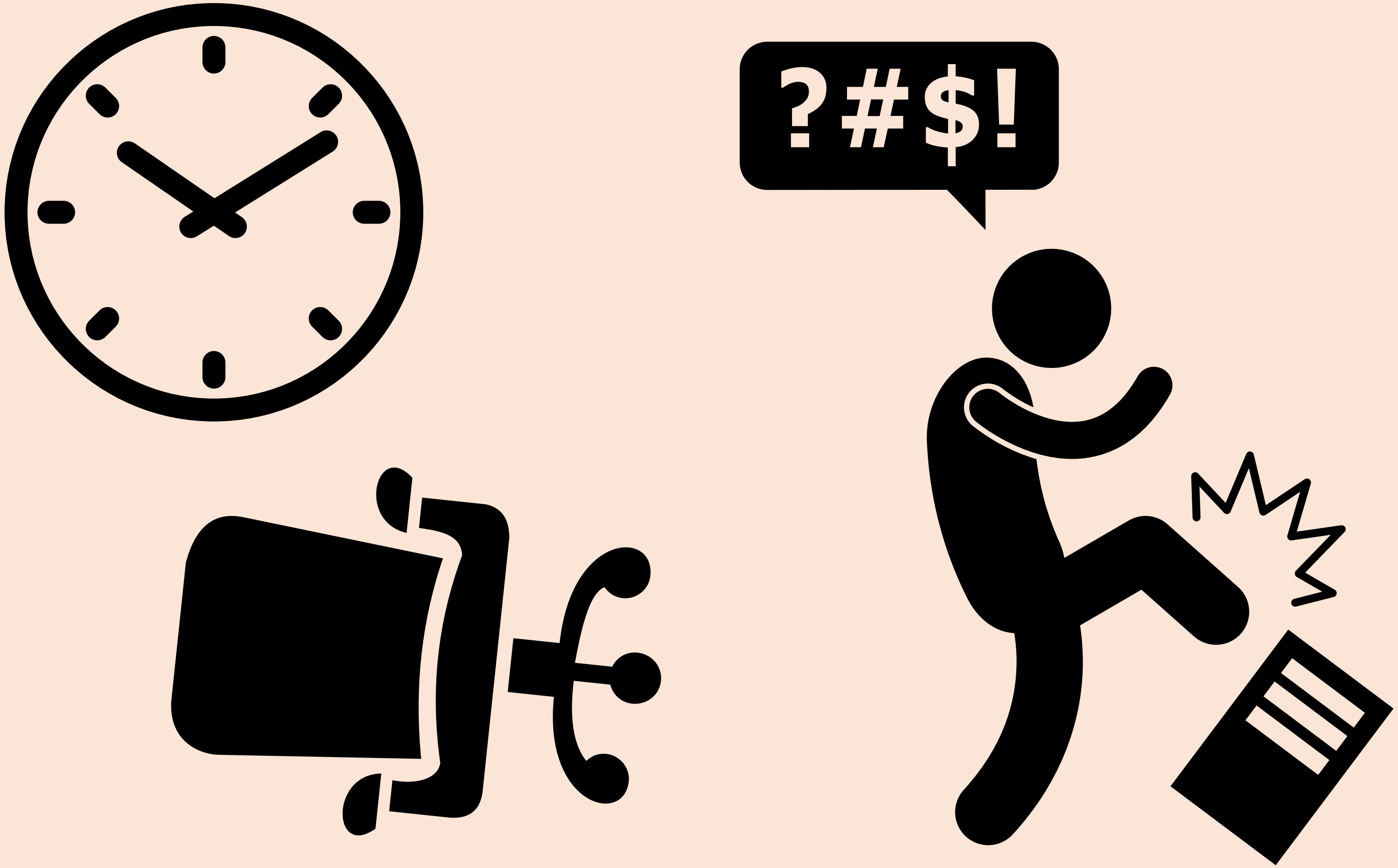


Peak memory performance for different scheduling

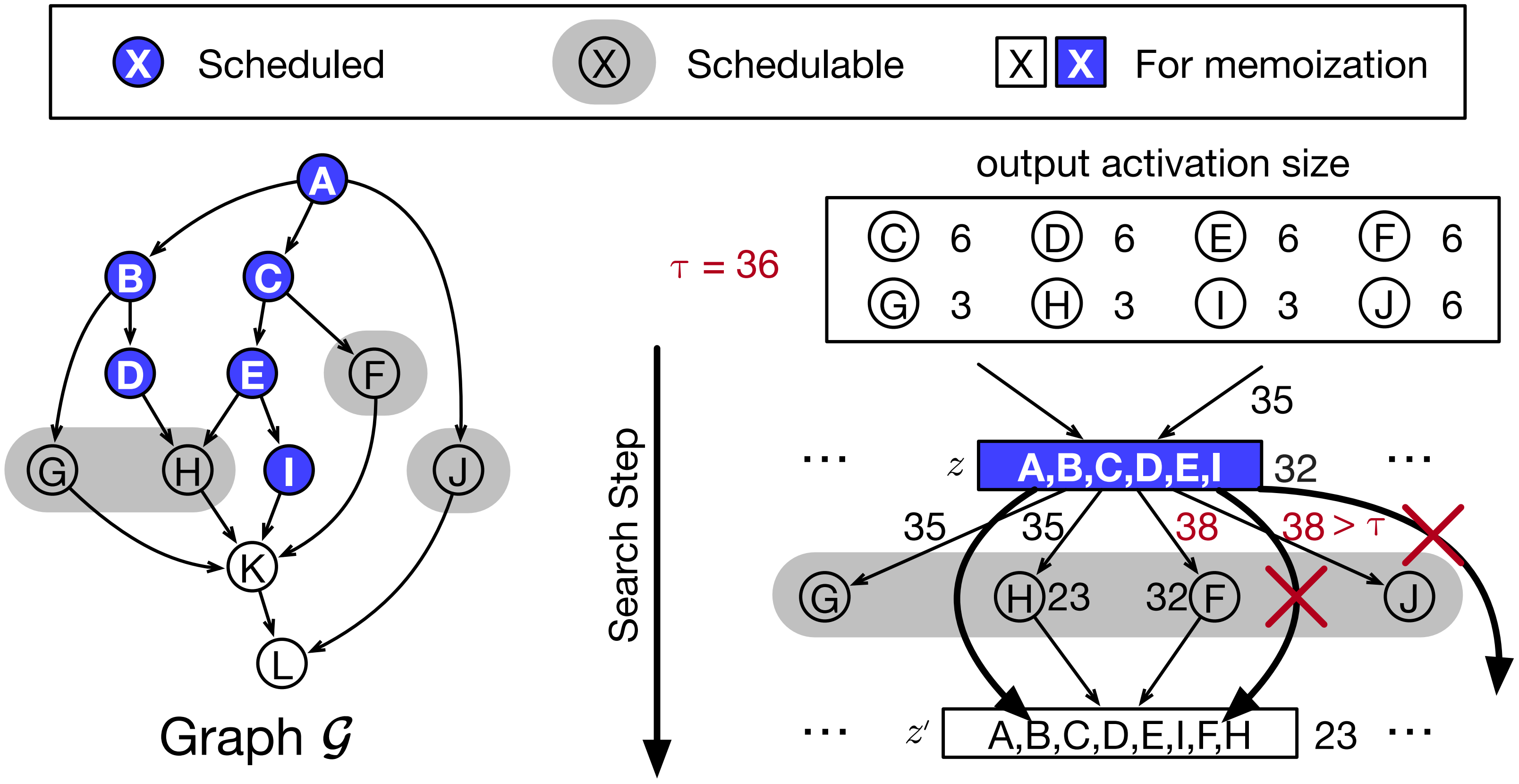


Scheduling Strategy	Peak Mem	Time
Manual Optimization + Partial Convolution	200KB	2 days
(Automatic) Dynamic Programming-based Scheduling	200KB	?
(Automatic) Dynamic Programming-based Scheduling + Graph Rewriting	188KB	?

Long Compile Time is Not Good for Mental Health



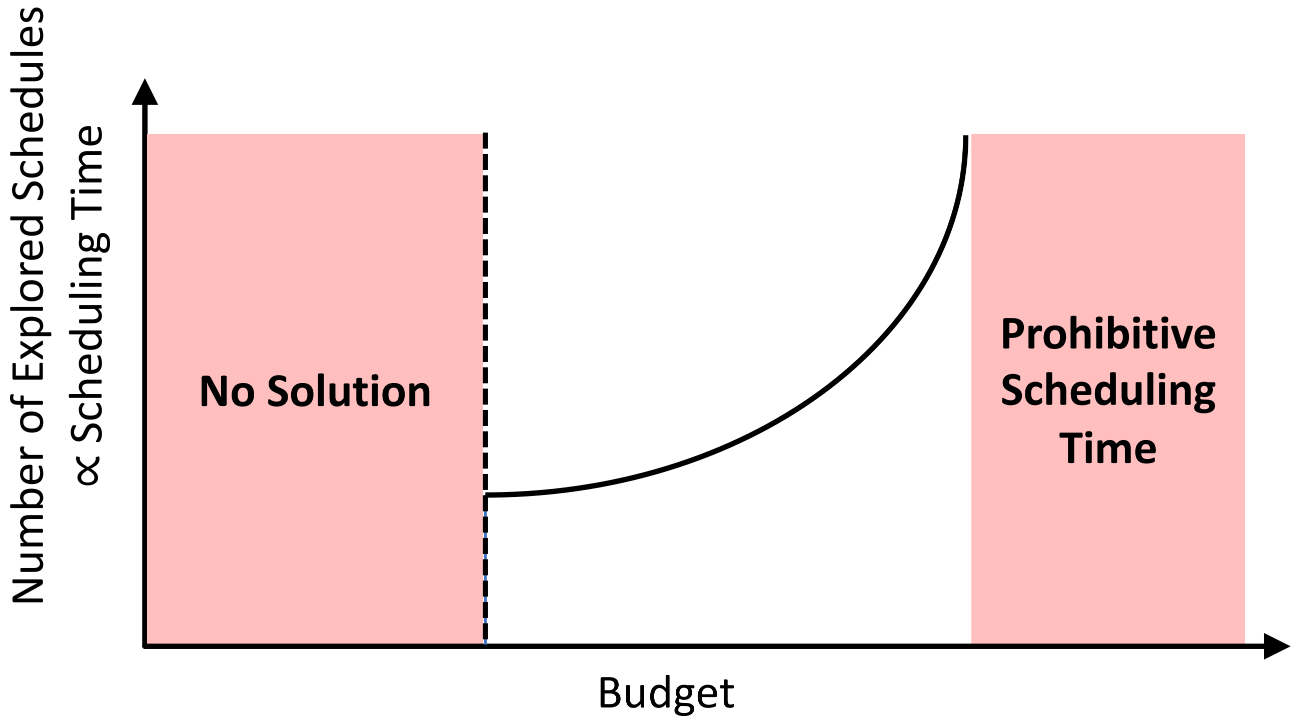
Pruning without Affecting Optimality



By **setting an appropriate threshold**,

some paths can be pruned without affecting optimality

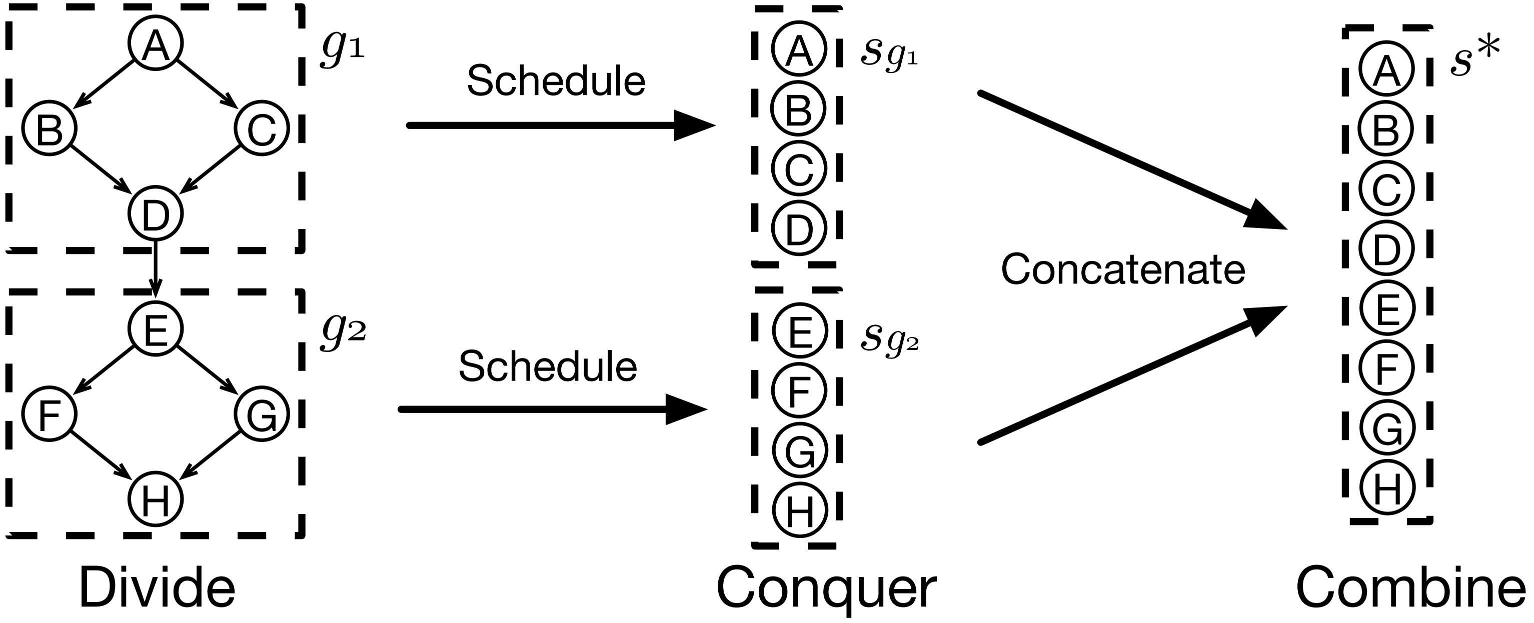
Adaptive Soft Budgeting



Adaptive Soft Budgeting finds appropriate threshold

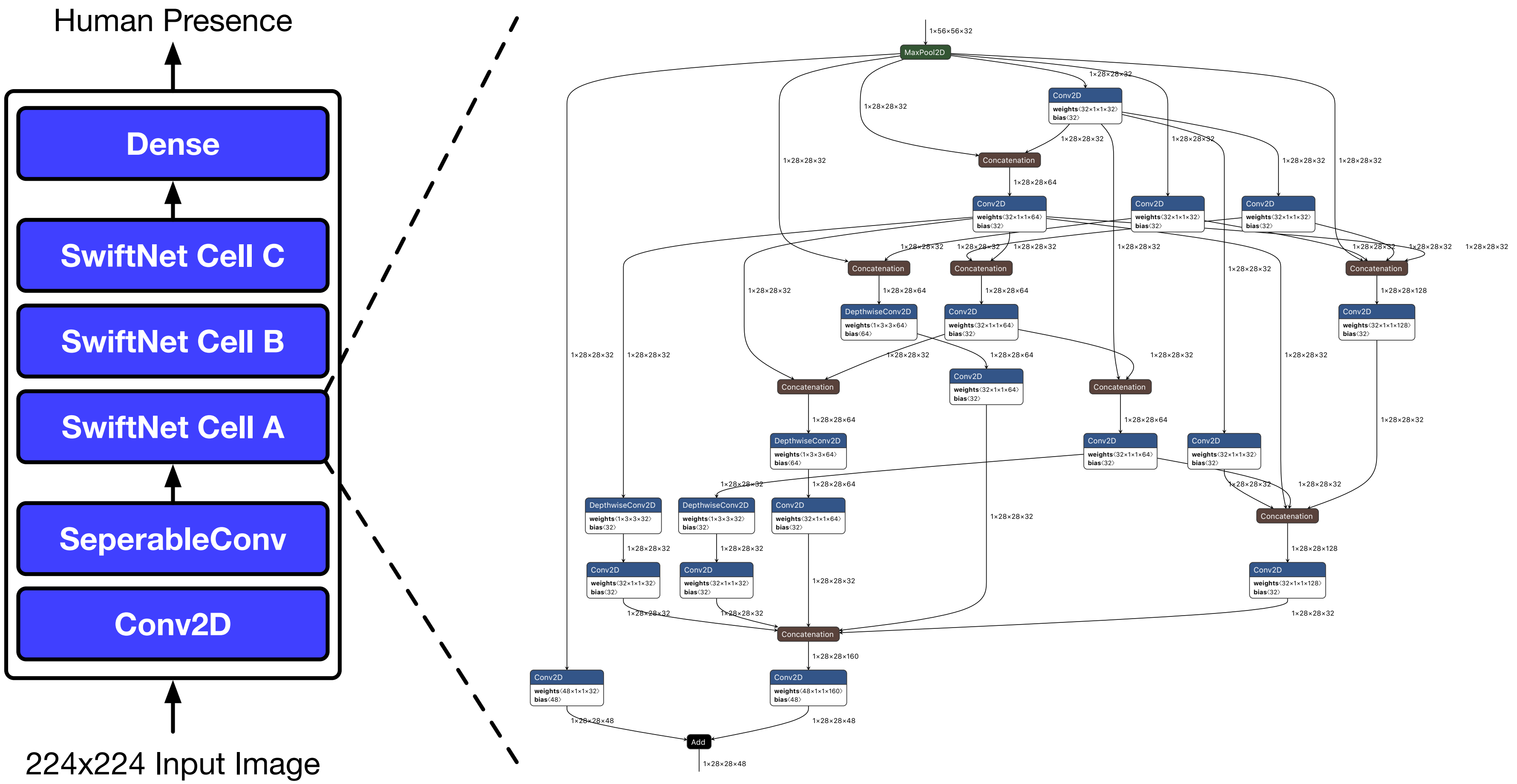
reducing the scheduling time significantly

Accelerating Automated Approach: Divide and Conquer



Many **Irregularly Wired Neural Networks** are **Hourglass-shaped** that enables **Divide-and-Conquer**

Peak memory performance for different scheduling



Scheduling Strategy	Peak Mem	Time
Manual Optimization + Partial Convolution	200KB	2 days
(Automatic) Dynamic Programming-based Scheduling	200KB	seconds
(Automatic) Dynamic Programming-based Scheduling + Graph Rewriting	188KB	minutes

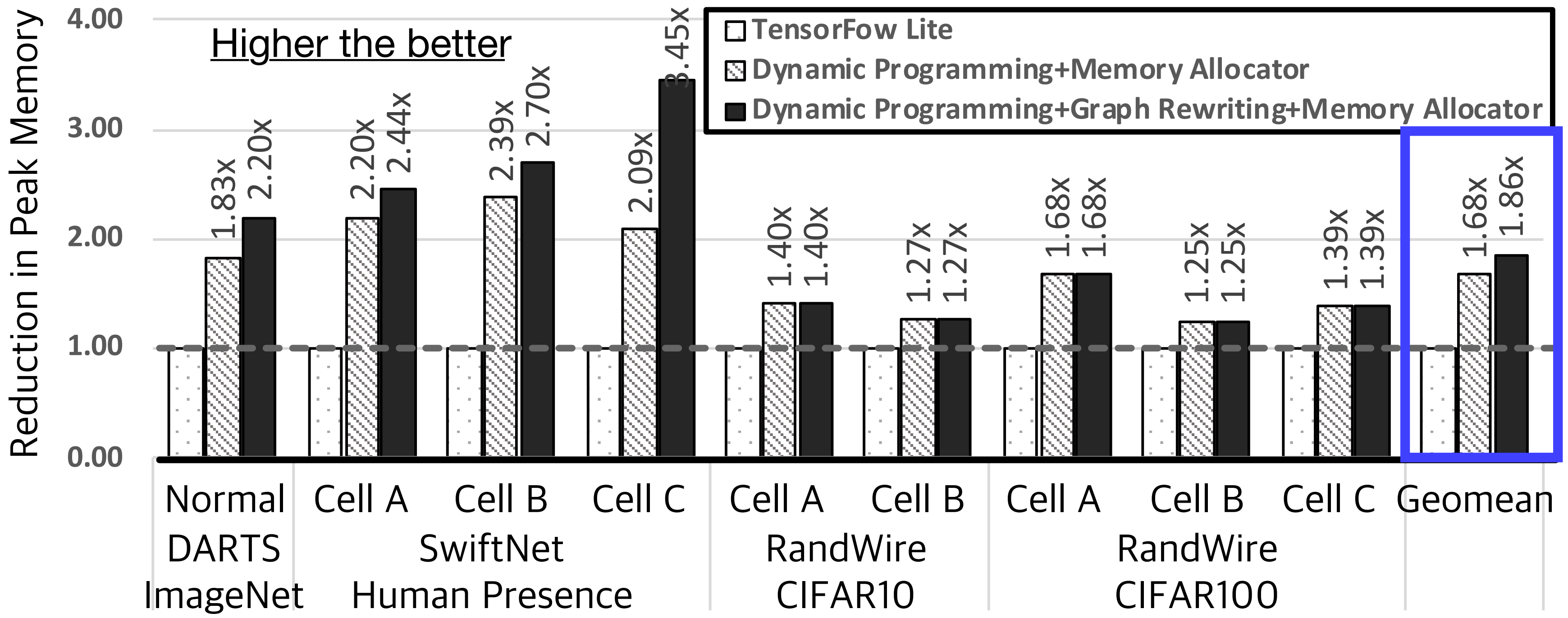
Evaluation

Evaluation: Benchmark Irregularly Wired Neural Networks

Network	Type	Dataset	# MAC	# Weight	Top-1 Accuracy*
DARTS [ICLR'19]	Neural Architecture Search	ImageNet	574.0M	4.7M	73.3%
SwiftNet [CVPR-C'19, ICCV-W'19]		Human Presence Detection	57.4M	249.7K	95.1%
Randomly Wired Neural Networks [ICCV'19]	Random Network Generators	CIFAR10	111.0M	1.2M	93.6%
		CIFAR100	160.0M	4.7M	74.5%

* **Serenity** does not affect accuracy

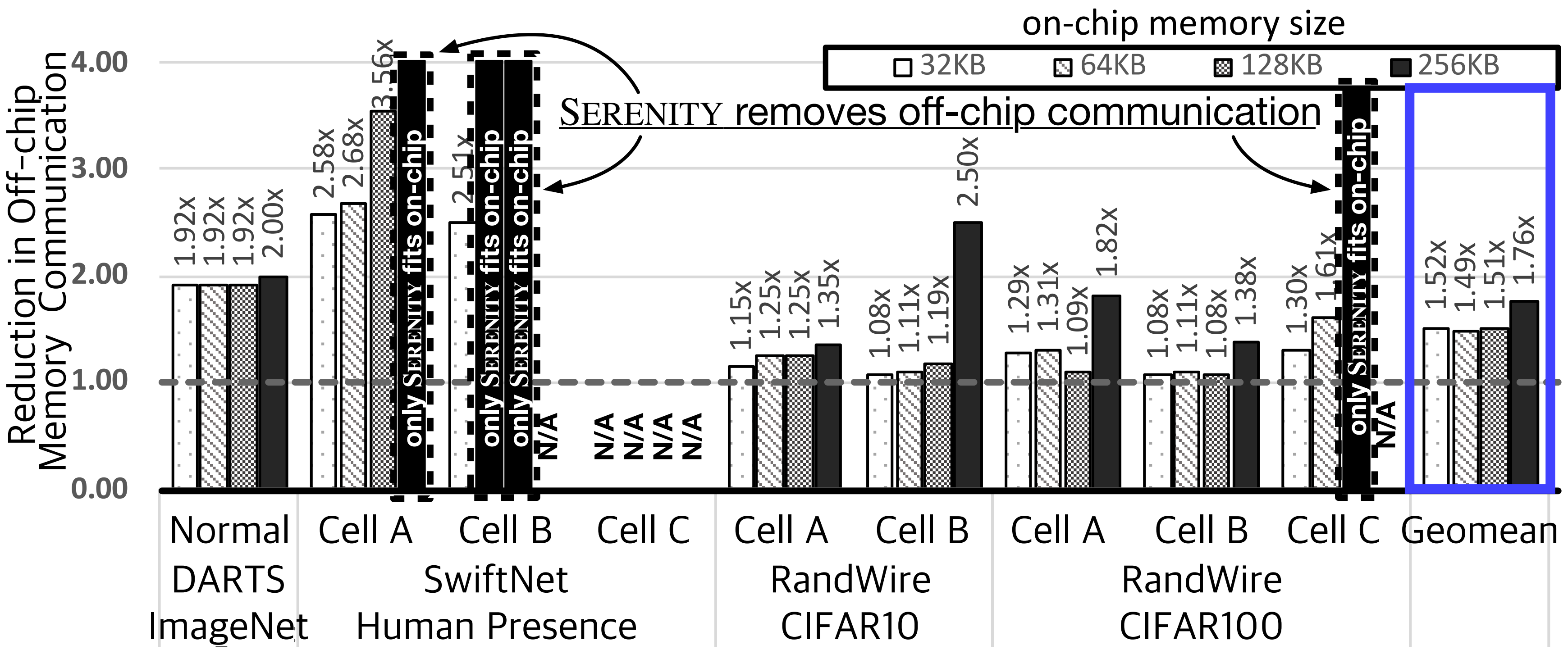
Evaluation: Reduction in Peak Memory Footprint



Serenity reduces the **Peak Memory Footprint**

by **1.68x** without Graph Rewriting and **1.86x** with Graph Rewriting

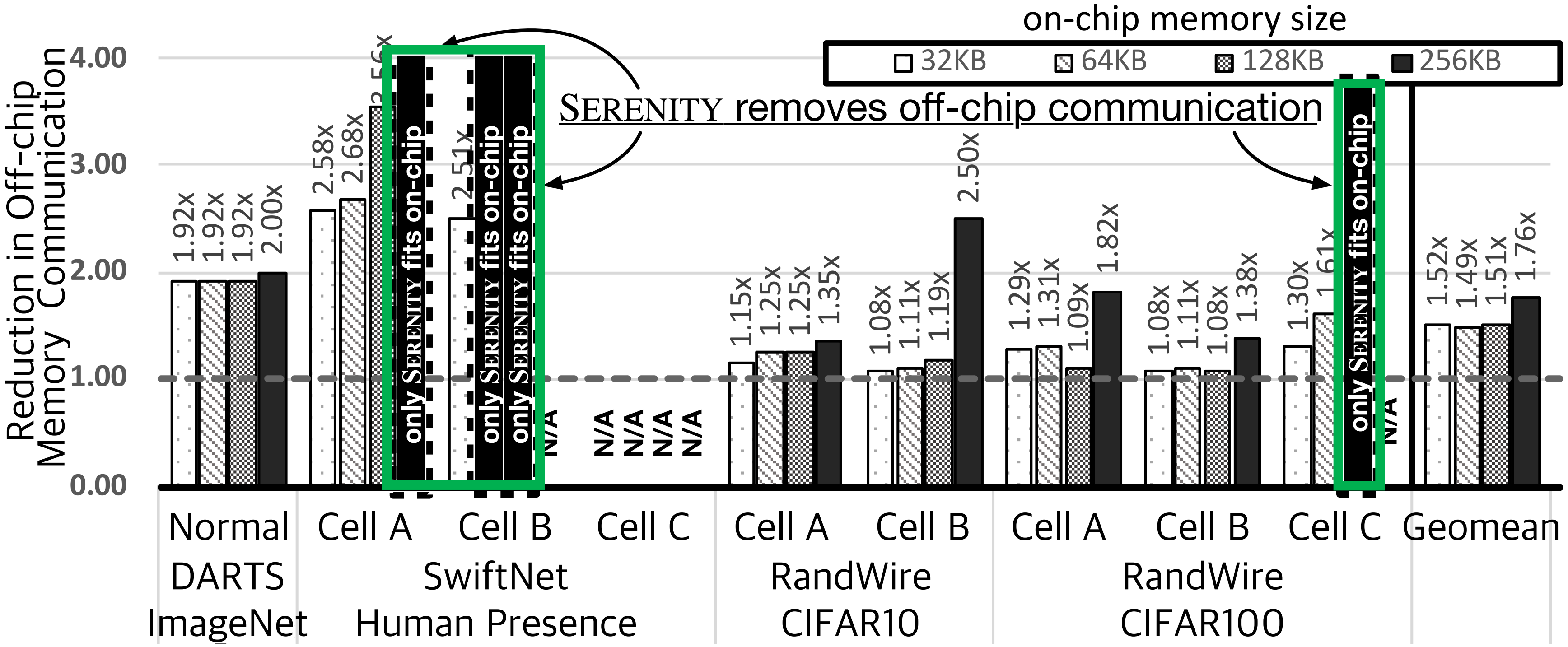
Evaluation: Reduction in Off-Chip Memory Communication



Serenity also **reduces off-chip memory communication**

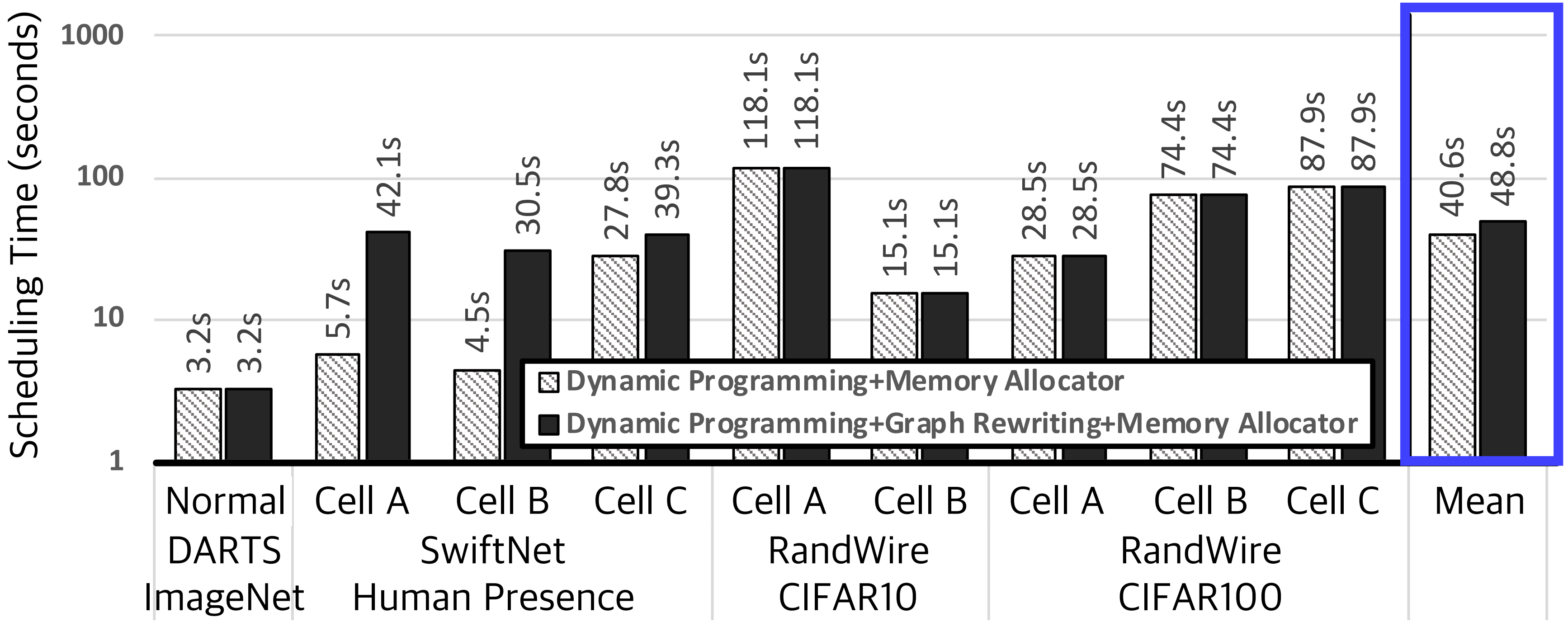
by **1.52x**, **1.49x**, **1.51x**, and **1.76x** for 32KB, 64KB, 128KB, and 256KB, respectively

Evaluation: Reduction in Off-Chip Memory Communication



Serenity even **eradicates off-chip memory communication**

Evaluation: Scheduling Time



Average scheduling time of **Serenity** is **under a minute** for the benchmark models

Can be further improved by **Porting from Python to C/C++**

Summary and Takeaways

1. **Irregularly Wired Neural Networks** are emerging class of Network Architectures with **many upsides in terms of efficiency**, but current deep learning frameworks are oblivious to the **Peak Memory Footprint challenge** they introduce.

2. We leverage **Dynamic Programming-based Scheduling** to find an **optimal schedule**; devise a **Identity Graph Rewriting** to **further reduce Peak Memory Footprint**; and develop **Adaptive Soft Budgeting** and **Divide-and-Conquer** to **minimize overhead**

Future Directions

1 **Expanding Applications** or **Revisiting** the **classical algorithms** or **compiler heuristics**:
- Problems of optimizing memory communication and inference time can also benefit from similar dynamic programming formulation

2. Using **Machine Learning** techniques to find good schedules in **one-shot**:
- Graph Neural Networks to parse and extract information from the graph
- Reinforcement Learning and other intelligent algorithms for scheduling

3. Exploring **Other Dimensions** of **reducing intermediate activations**:
- Quantization and Pruning are popular compression techniques
- Lossy/Lossless compression for intermediate activations are interesting future path