GPUs, Deep Learning, and Chip Design

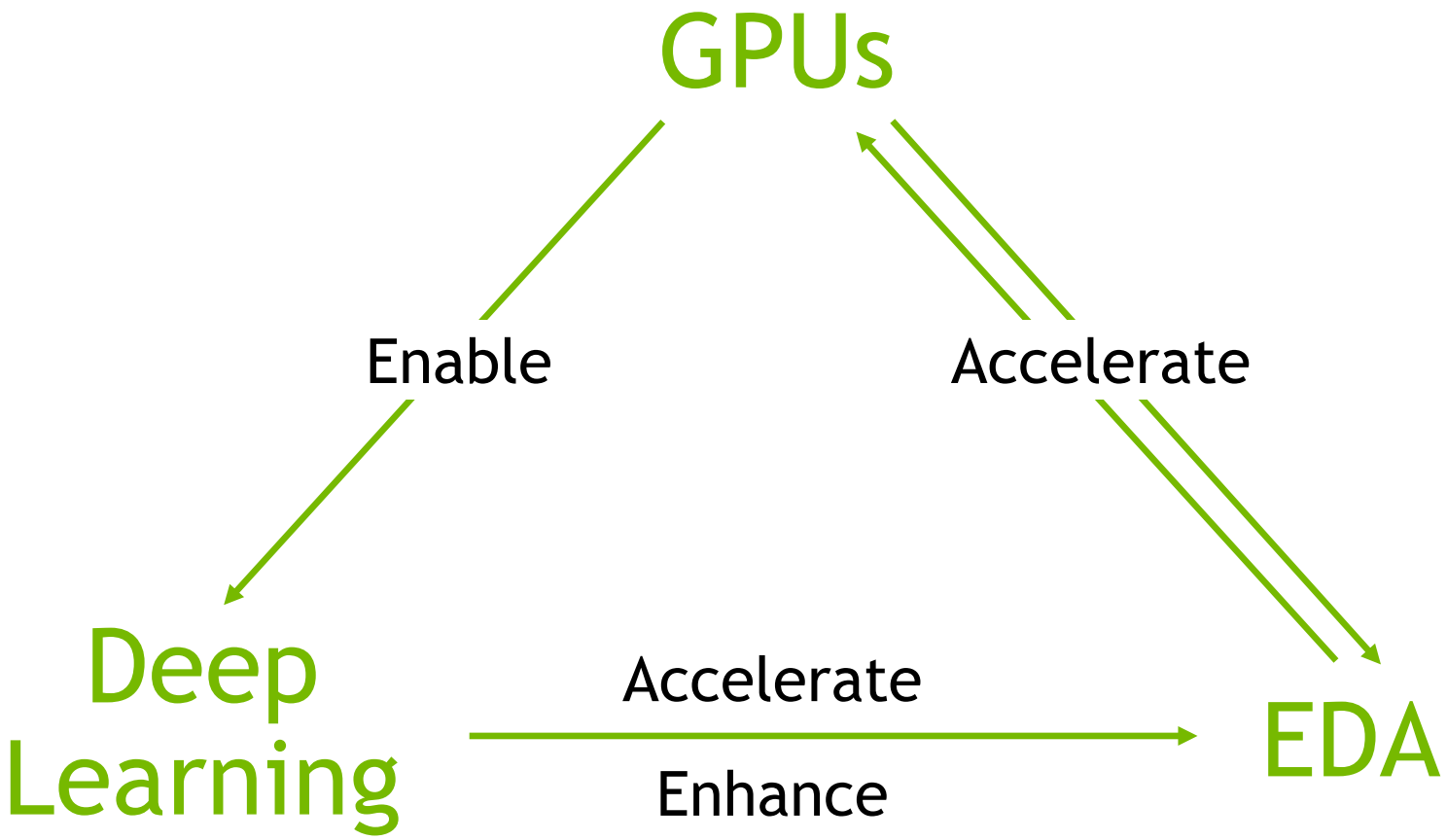
DARPA ERI

August 23, 2023

Bill Dally

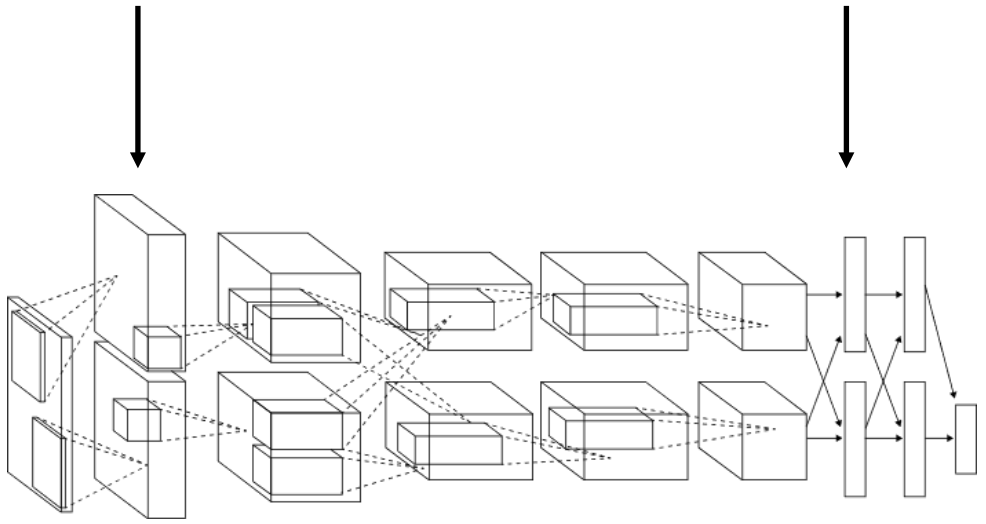
Chief Scientist and SVP of Research, NVIDIA Corporation

Adjunct Professor of CS and EE, Stanford

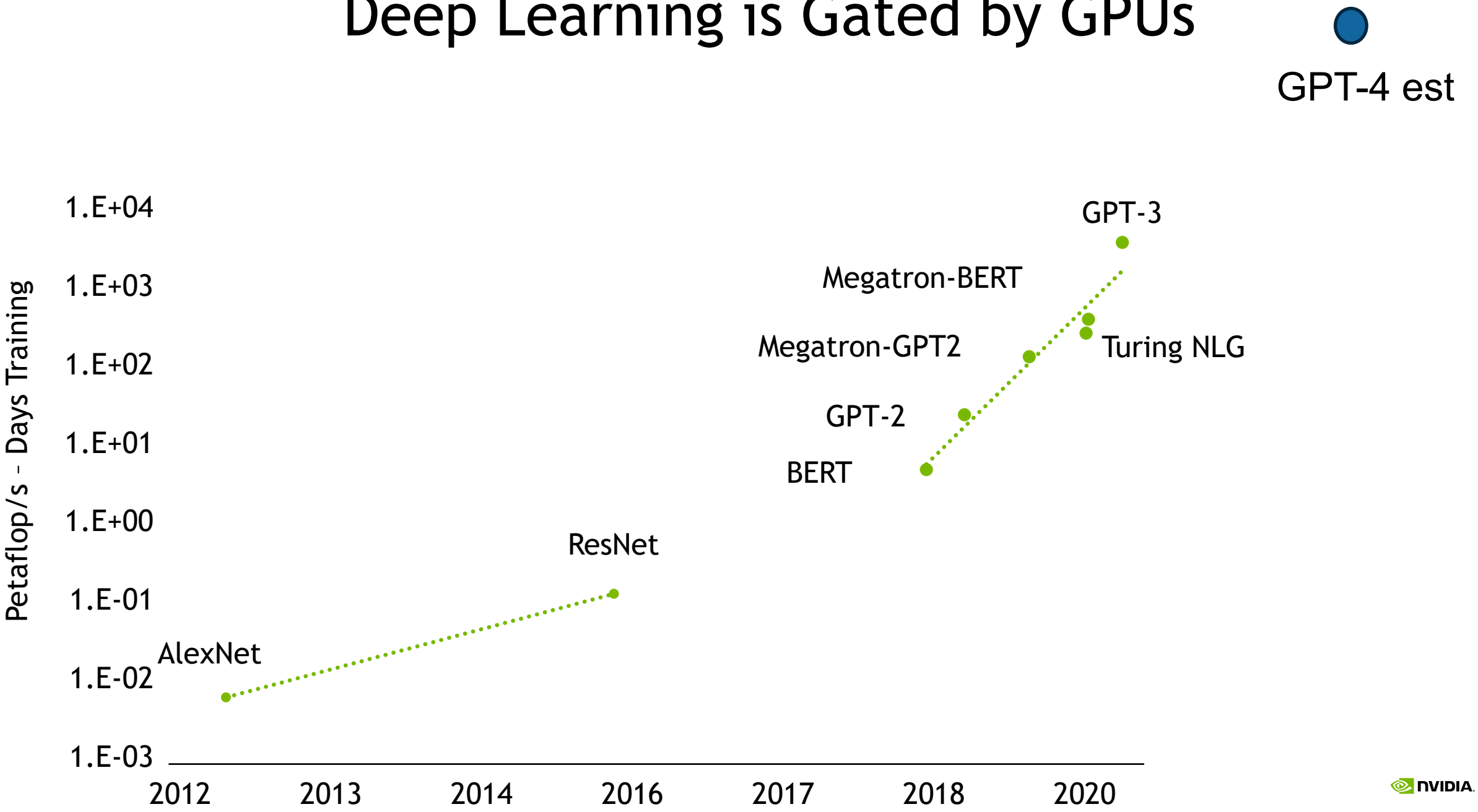

GPUs
Enable
Accelerate
Deep Learning
Accelerate
Enhance
EDA

GPUs and Deep Learning

Deep Learning was Enabled by GPUs

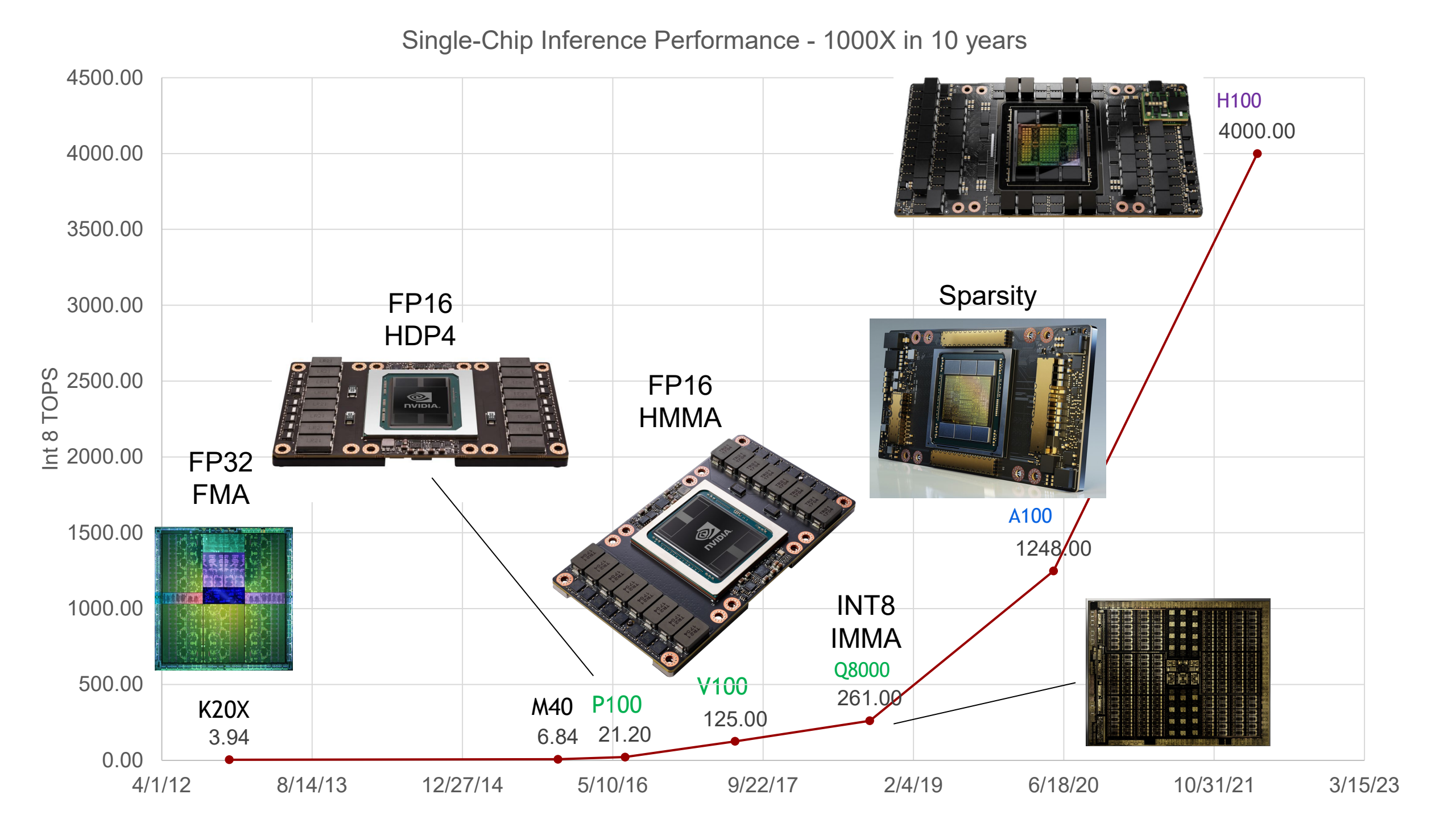


Deep Learning is Gated by GPUs



NVIDIA

The Evolution of Deep-Learning GPUs


Single-Chip Inference Performance - 1000X in 10 years
Int 8 TOPS
4500.00
4000.00
3500.00
3000.00
2500.00
2000.00
1500.00
1000.00
500.00
0.00
4/1/12
8/14/13
12/27/14
5/10/16
9/22/17
2/4/19
6/18/20
10/31/21
3/15/23
H100
4000.00
Sparsity
FP16
HDP4
FP16
HMMA
FP32
FMA
A100
1248.00
INT8
IMMA
Q8000
261.00
V100
125.00
K20X
3.94
M40
6.84
P100
21.20

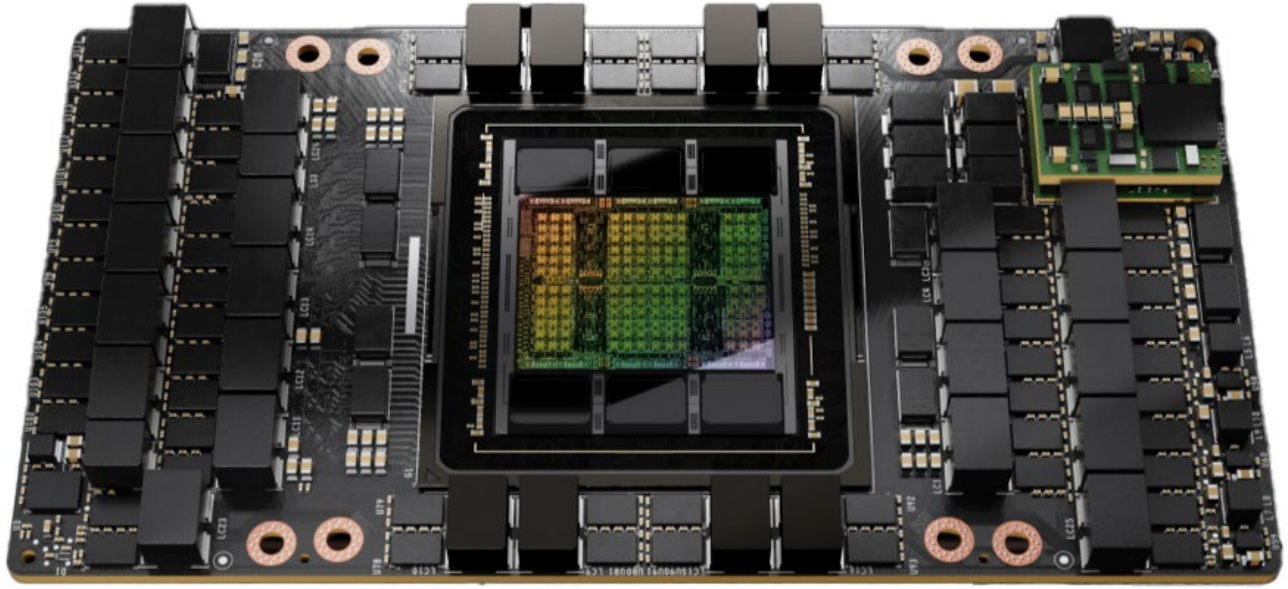
Hopper H100

1 PFLOPS (TF32)
1 / 2 PLFLOPS (FP16 or BF16) (dense/sparse)
2 / 4 PLFOPS (FP8 or Int8)

3TB/s (HBM3) 96GB
700W

Transformer Engine
Dynamic Programming Instructions

9 TOPS/W (Int8/FP8)

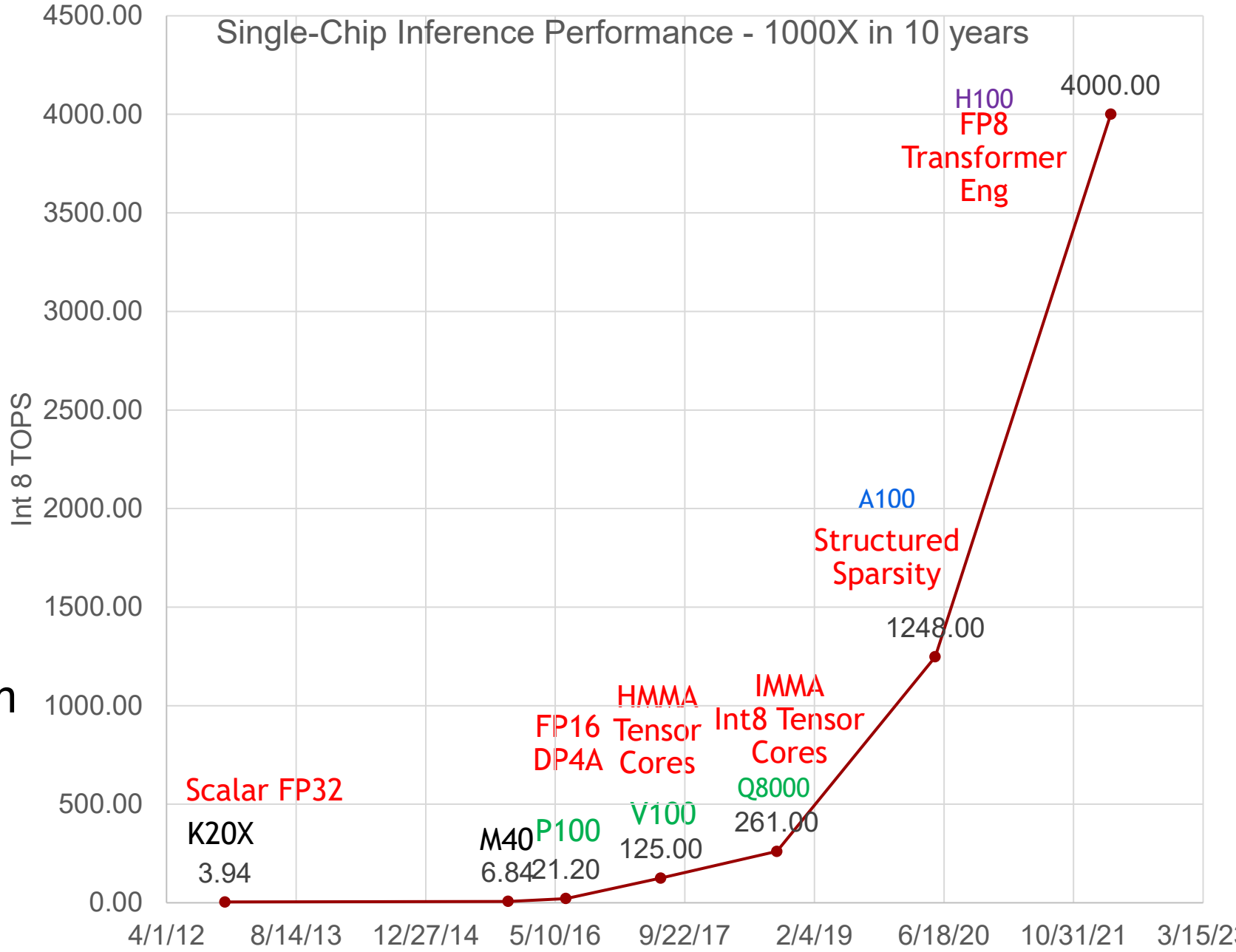


Gains from

Number representation
FP32, FP16, Int8
(TF32, BF16)

Complex instructions
DP4, HMMA, IMMA

Process
28nm, 16nm, 7nm, 5nm



Specialized Instructions Amortize Overhead

Operation	Energy**	Overhead*
HFMA	1.5pJ	2000%
HDP4A	6.0pJ	500%
HMMA	110pJ	22%
IMMA	160pJ	16%

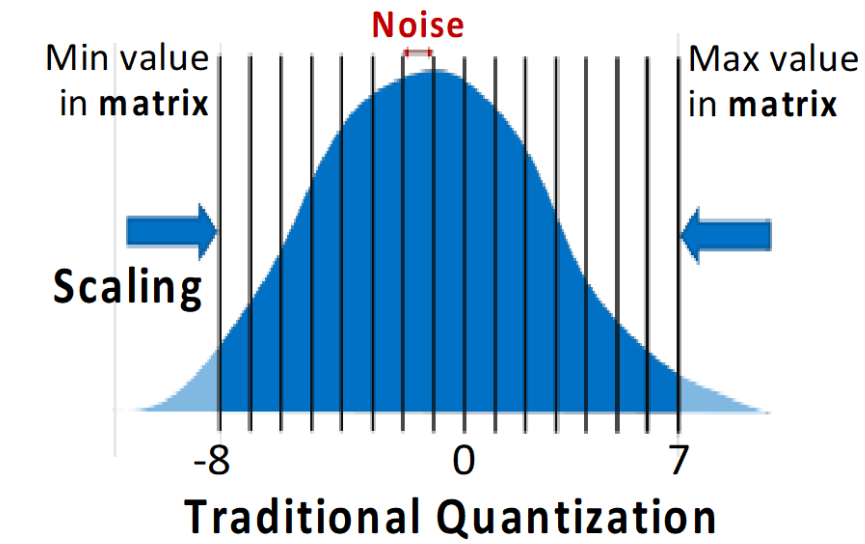
*Overhead is instruction fetch, decode, and operand fetch – 30pJ
**Energy numbers from 45nm process

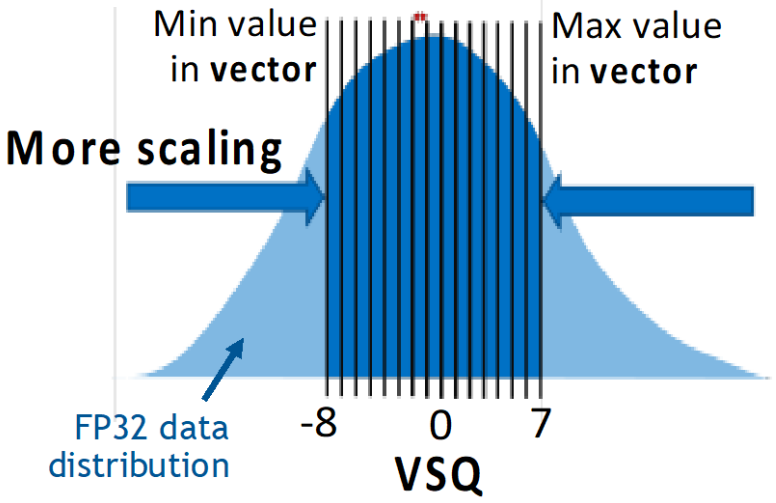
Seamless scale-up and scale-out
NVLink
Infiniband


MODULUS
CLARA MONAI
RIVA
MAXINE
NEMO
MERLIN
AVATAR
DRIVE
ISAAC
METROPOLIS
HOLOSCAN
APPLICATION FRAMEWORKS
PLATFORM
NVIDIA HPC
NVIDIA AI
NVIDIA OMNIVERSE
SYSTEM SOFTWARE
RTX
CUDA-X
PHYSX
UCF
DOCA
MAG
BASE CMD
FLEET CMD
AERIAL
HARDWARE
RTX
DGX
HGX
EGX
OVX
SUPER POD
AGX
GPU
CPU
DPU
NIC
SWITCH
SOC

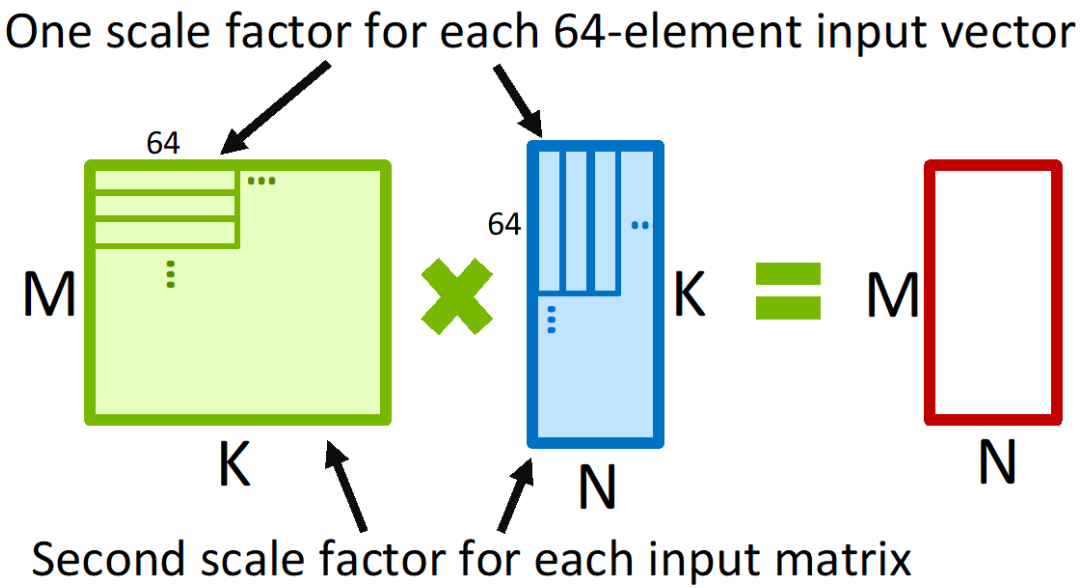
Better Number Representation

INT4 Quantization



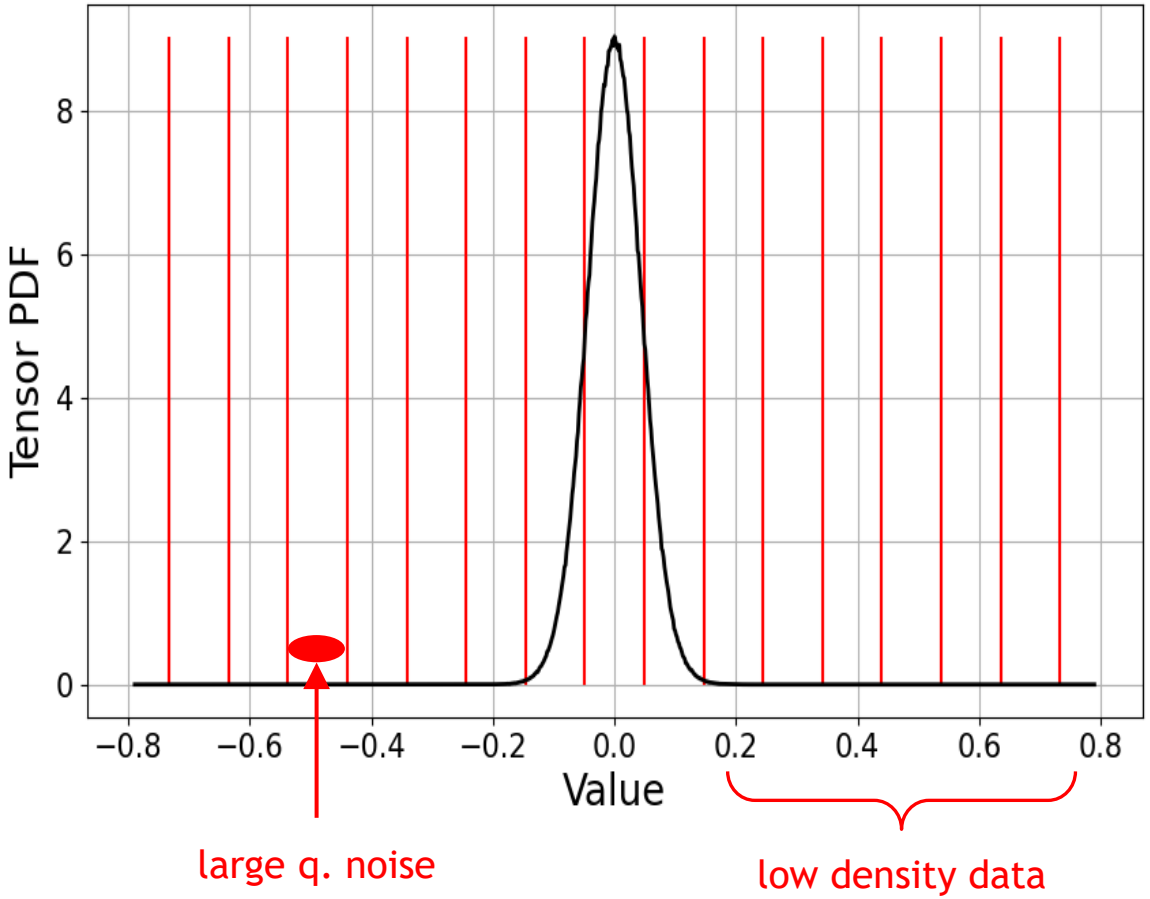


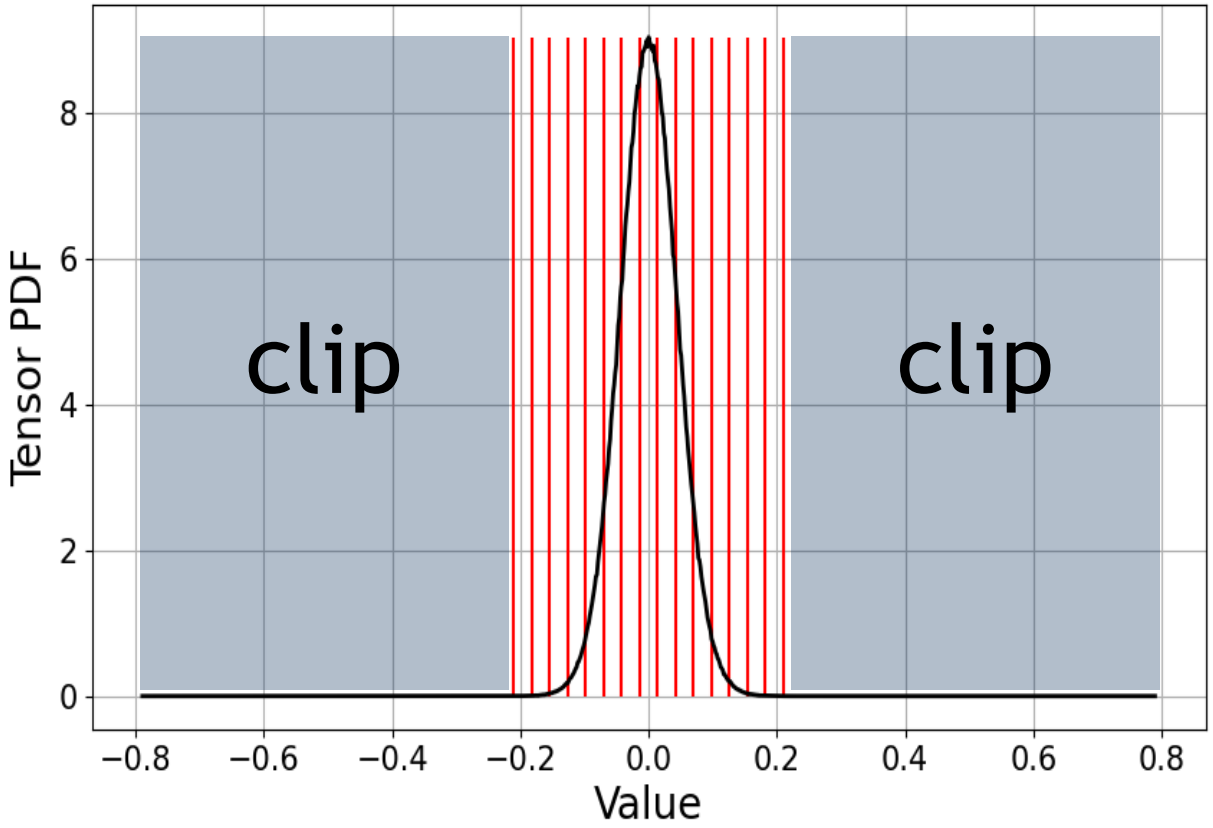
VSQ Scale Factors



Traditional Quantization	VSQ
One scale factor per matrix	Two scale factors: one per vector, one per matrix
High quantization noise	Reduced quantization noise

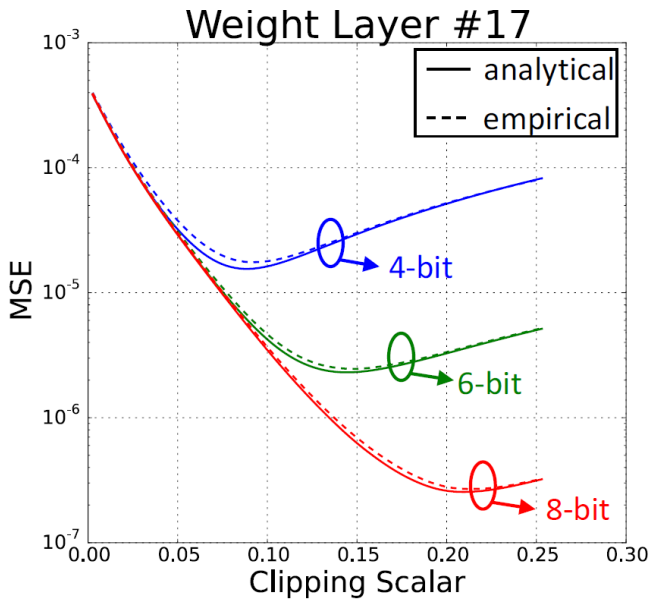
Optimal Clipping

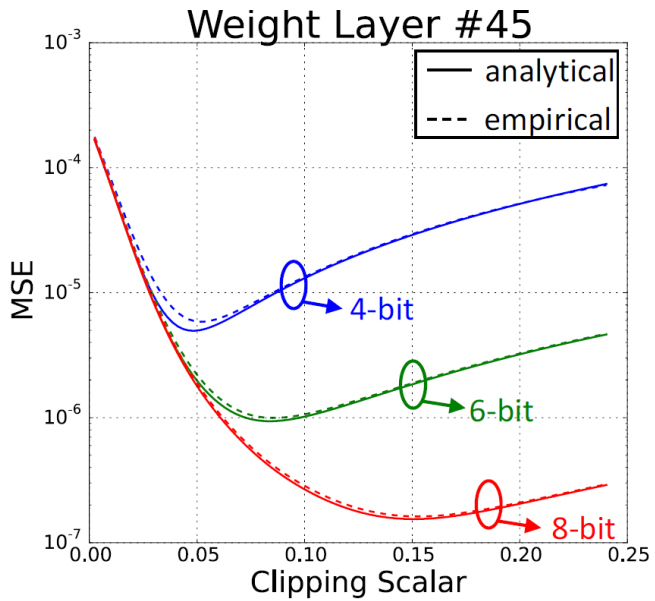


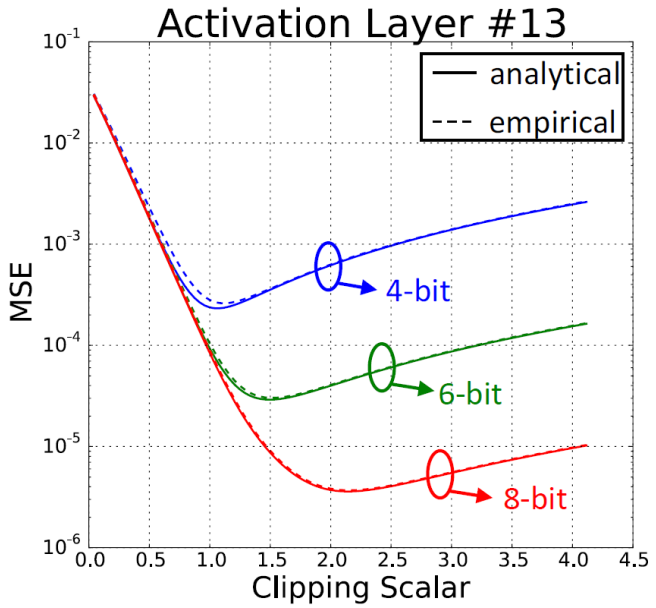


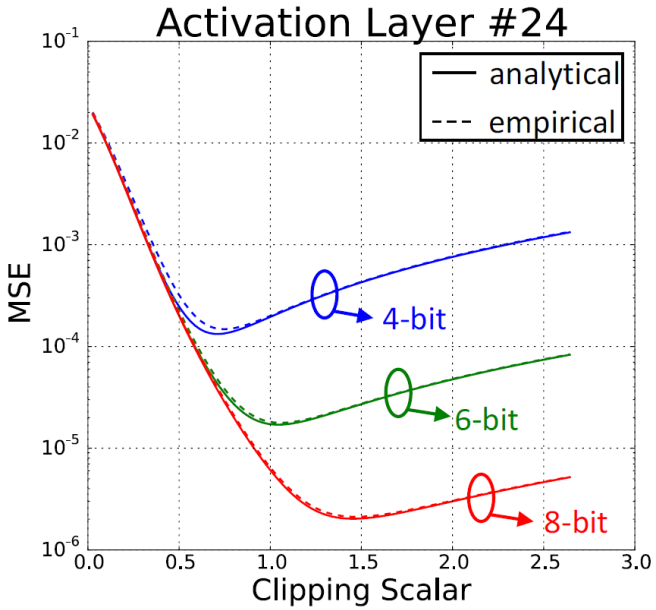
$$J = \frac{4^{-B}}{3} s^2 \int_0^s f_{|X|}(x)dx + \int_s^\infty (s-x)^2 f_{|X|}(x)dx$$

$$s_{n+1} = \frac{\boldsymbol{E}\left[|X| \cdot \mathbf{1}_{\{|X|>s_n\}}\right]}{\frac{4^{-B}}{3}\boldsymbol{E}\left[\mathbf{1}_{\{|X|<s_n\}}\right] + \boldsymbol{E}\left[\mathbf{1}_{\{|X|>s_n\}}\right]}$$





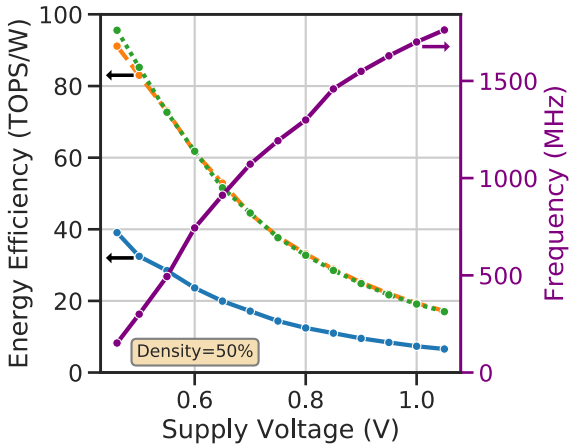




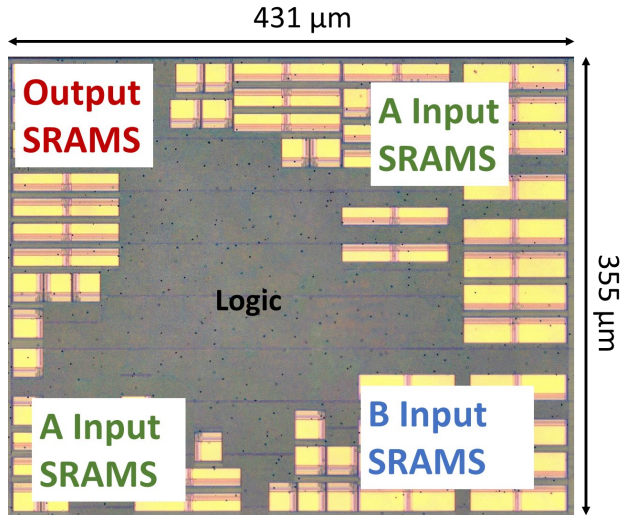
ENERGY-EFFICIENT DL INFERENCE ACCELERATOR

Transformers, VS-Quant INT4, TSMC 5nm

- Efficient architecture
  - Used MAGNet *[Venkatesan et al., ICCAD 2019]* to design a low-precision DL inference accelerator for Transformers
  - Multi-level dataflow to improve data reuse and energy efficiency
- Low-precision data format: VS-Quant INT4
  - Hardware-software techniques to tolerate quantization error
  - Enable low cost multiply-accumulate (MAC) operations
  - Reduce storage and data movement
- Special function units
  - Hardware specialization to improve efficiency of functions like Softmax that are unique to transformers

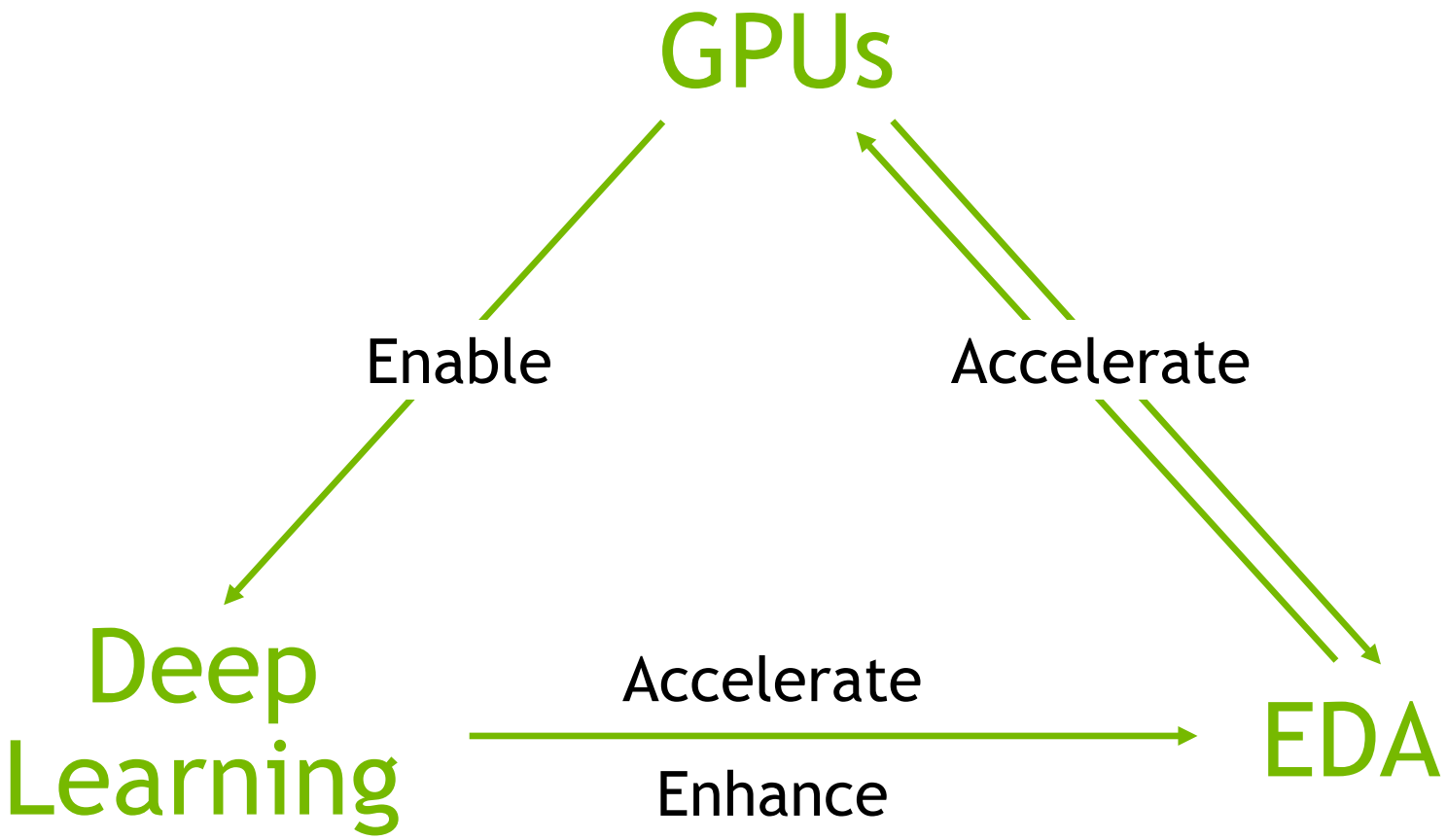


- 95.6 TOPS/W with 50%-dense 4-bit input matrices with VSQ enabled at 0.46V
- 0.8% energy overhead from VSQ support with 50%-dense inputs at 0.67V



- TSMC 5nm
- 1024 4-bit MACs/cycle (512 8-bit)
- 0.153 mm^2 chip
- Voltage range: 0.46V – 1.05V
- Frequency range: 152 MHz – 1760 MHz

[Keller, Venkatesan, et al., "A 95.6-TOPS/W Deep Learning Inference Accelerator with Per-Vector Scaled 4-bit Quantization in 5nm", JSSC 2023]


GPUs
Enable
Accelerate
Deep Learning
Accelerate
Enhance
EDA

GPU Accelerated EDA

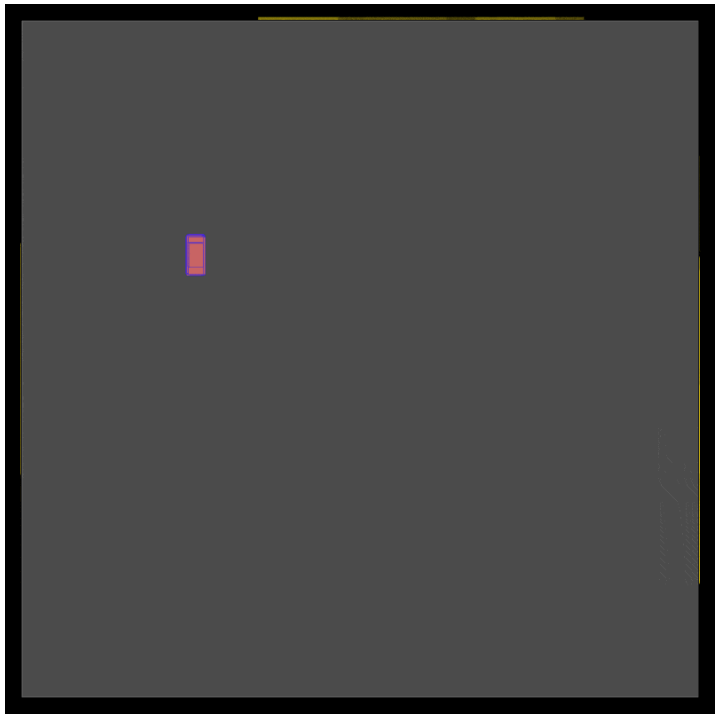
AutoDMP

Automatic Parameter Tuning for Macro Placement

- Mixed-size analytical method offers best PPA quality for macro placement
  - EDA: macro placement space explored by launching many jobs
    - e.g., Cadence Cerebrus/Synopsys DSO.ai use RL to select tool options (recipes), design constraints, library options
  - Issue: need massive compute, slow, still black-boxed

- DREAMPlace is a **superfast** mixed-size placer
  - Treats macros & standard cells similarly → macro **legalization issues**
  - High influence of parameter settings on optimization quality

- **AutoDMP → generate quickly diverse high-quality macro placements**
  - Enhance DREAMPlace: fix legalization issues & expand the design space
  - Tune DREAMPlace parameters with Multi-Objective Bayesian Optimization



Macro Placement on MemPool Design

AutoDMP: Automated DREAMPlace-based Macro Placement (ISPD 2023), https://github.com/NVlabs/AutoDMP

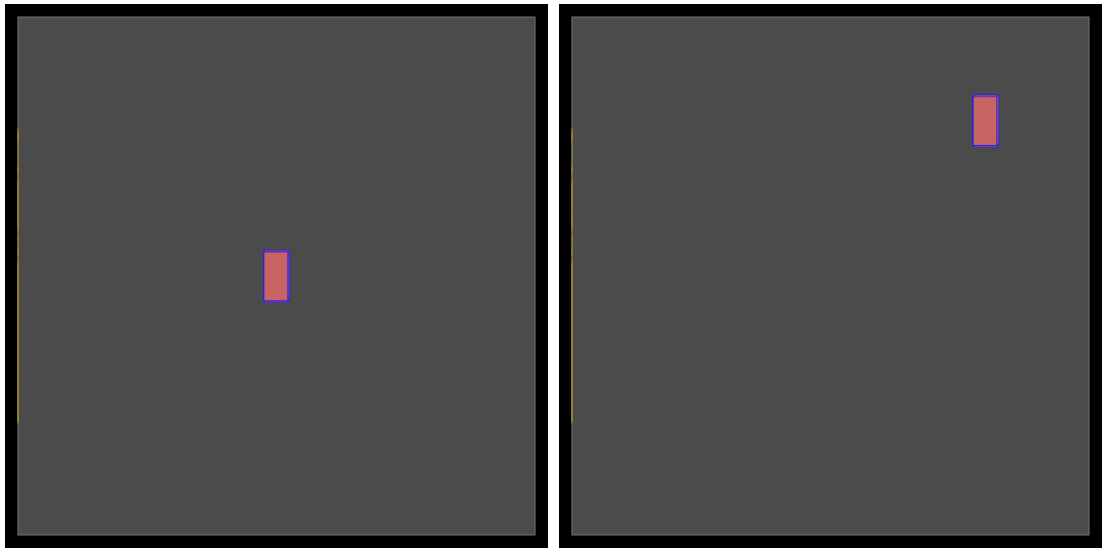
Parameter Space

AutoDMP optimizes parameters with large impact to QoR as well as algorithm convergence.

Parameter	Search Range	$\widehat{c_v}$ (%) RSMT	$\widehat{c_v}$ (%) Cong.	Divg. Rate
*horiz. initial position	[0.2, 0.8] (%)	2.2	0.9	0.0
*vert. initial position	[0.2, 0.8] (%)	2.0	1.1	0.0
*horiz. macro halo	technology dep.	1.8	1.3	0.0
*vert. macro halo	technology dep.	1.7	1.2	0.0
target density d_{target}	$[a_{\text{util}} - 0.2, a_{\text{util}}]$ (%)	-	-	-
density weight	$[1e^{-6}, 1.0]$	3.1	1.7	0.0
smooth HPWL model	{LSE, WA}	0.7	1.1	0.0
smooth HPWL initial γ_0	[0.10, 0.50]	5.1	1.9	0.0
GD initial LR lr_0	$[1e^{-4}, 1e^{-2}]$	1.4	1.0	0.0
GD LR decay	[0.99, 1.0]	6.7	2.3	53.2
GD optimizer	[Adam, Nesterov]	1.2	0.8	54.2
# horiz. global bins	{256, 512, 1024, 2048}	1.3	0.9	0.0
# vert. global bins	{256, 512, 1024, 2048}	3.1	1.3	21.1
λ update lower coeff. L	[0.90, 0.99]	4.2	1.9	0.0
λ update upper coeff. U	[1.01, 1.15]	27.0	7.5	1.8
λ update $\Delta\,\text{HPWL}_{REF}$	$[1.5e^5, 5.5e^5]$	2.3	1.2	0.0

AutoDMP extends DREAMPlace parameter space to increase diversity as well as better legalization for macro placement

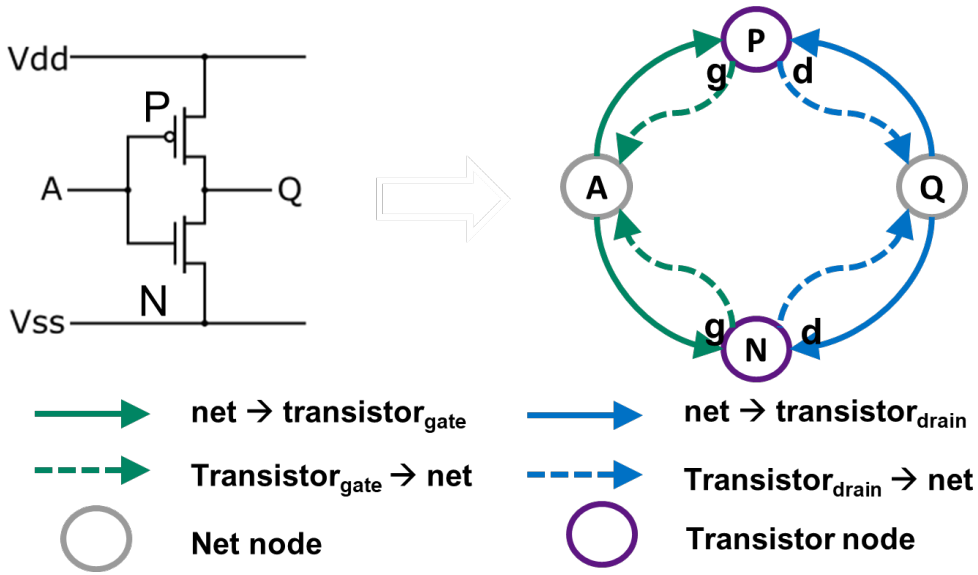
Example: Initial Positions



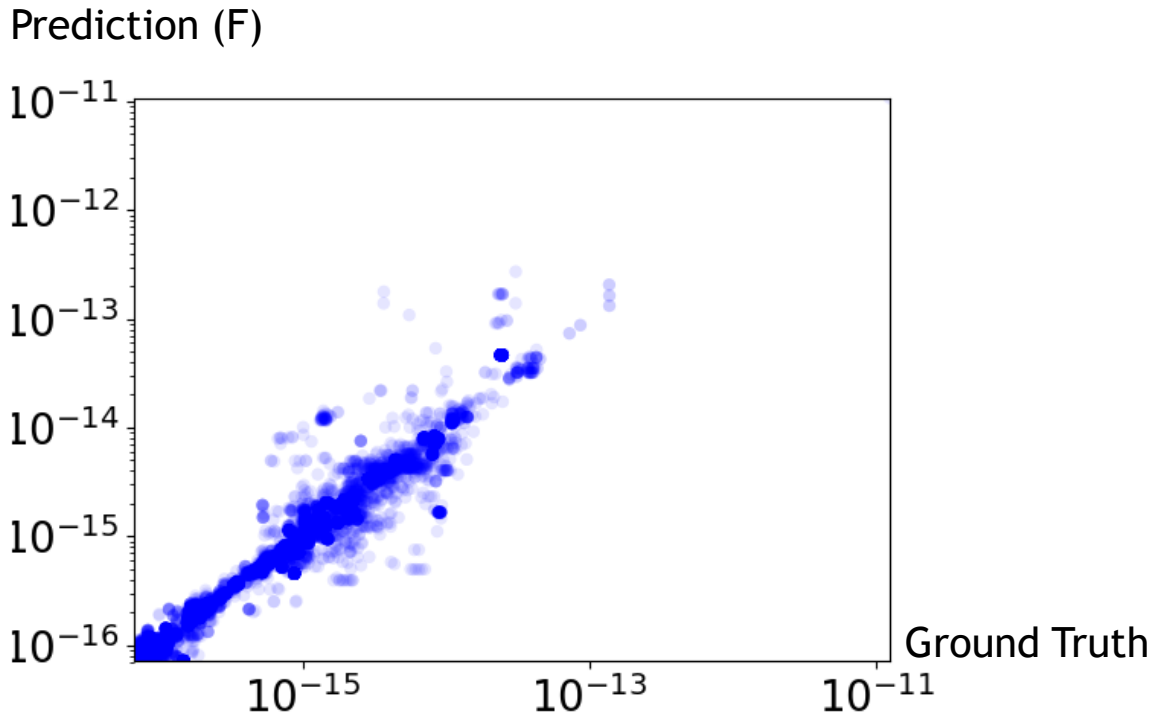
Center Position

Upper Right Position

PARASITICS PREDICTION WITH GNNS



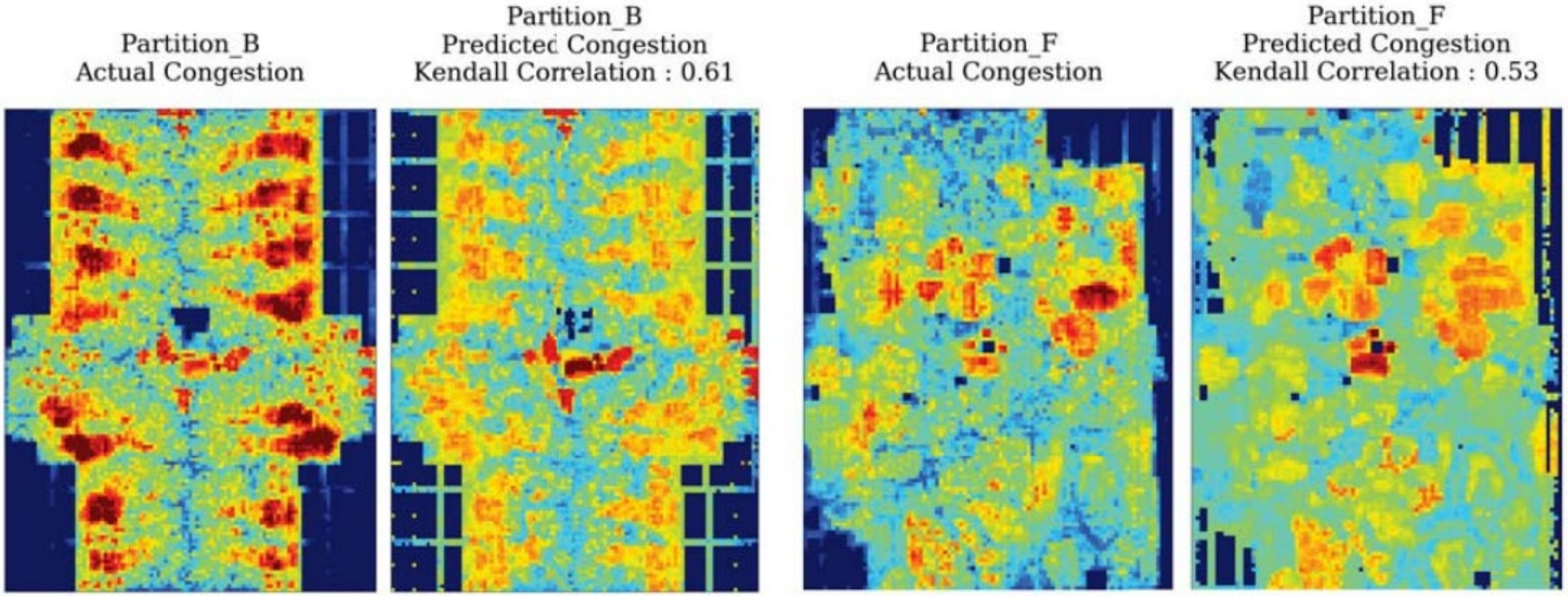
Circuit Schematics to Graph Conversion



MAE=0.852fF MAPE=15%
Simulation error reduced to <10%

H. Ren et al., "ParaGraph: Layout Parasitics and Device Parameter Prediction using Graph Neural Networks", DAC 2020.
(Research funded under Cadence's DARPA IDEA contract)

ROUTING CONGESTION PREDICTION WITH GNNS



R. Kirby et al., "CongestionNet: Routing Congestion Prediction Using Deep Graph Neural Networks", VLSI-SoC 2019

Reinforcement Learning

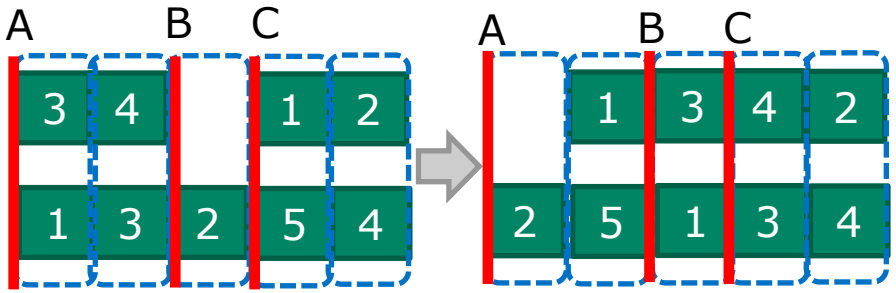
NVCELL: AUTOMATING STDCELL LAYOUT

(Ren et al., DAC 2021)

Placement: Simulated Annealing

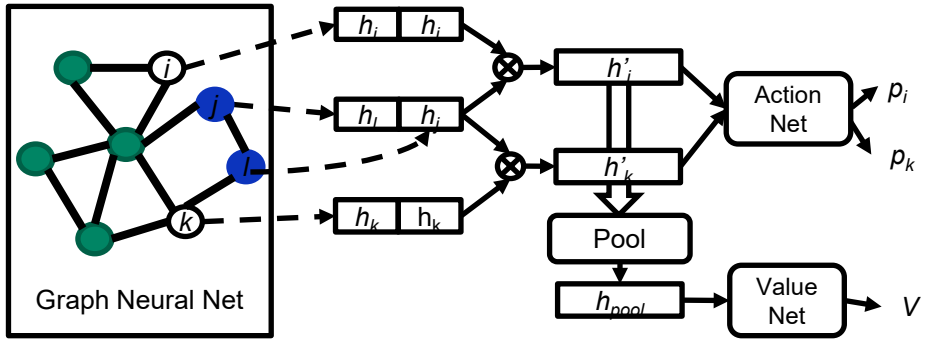
```
PMOS sequence: 3,  4,  NA, 1,  2
NMOS sequence: 1,  3,  2,  5,  4
PIN  sequence: A,  NA, B,  C,  NA
```

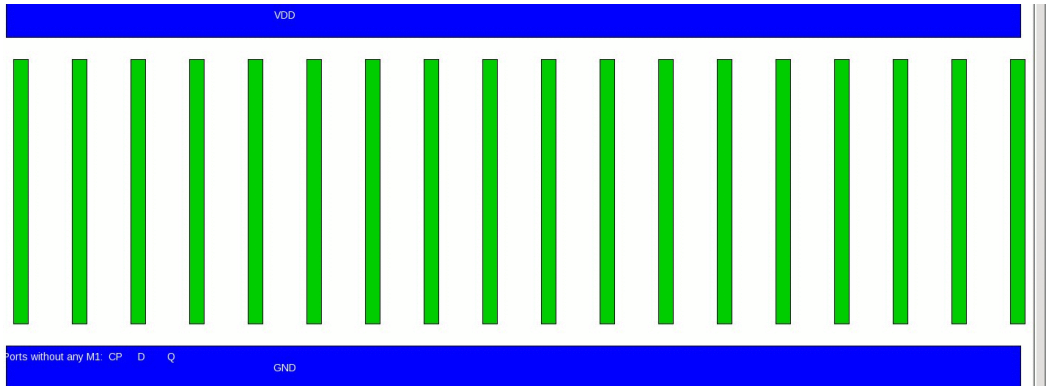
Placement representation



Transforms: Swap PMOS/NMOS pair segment, etc

Placement: Reinforcement learning

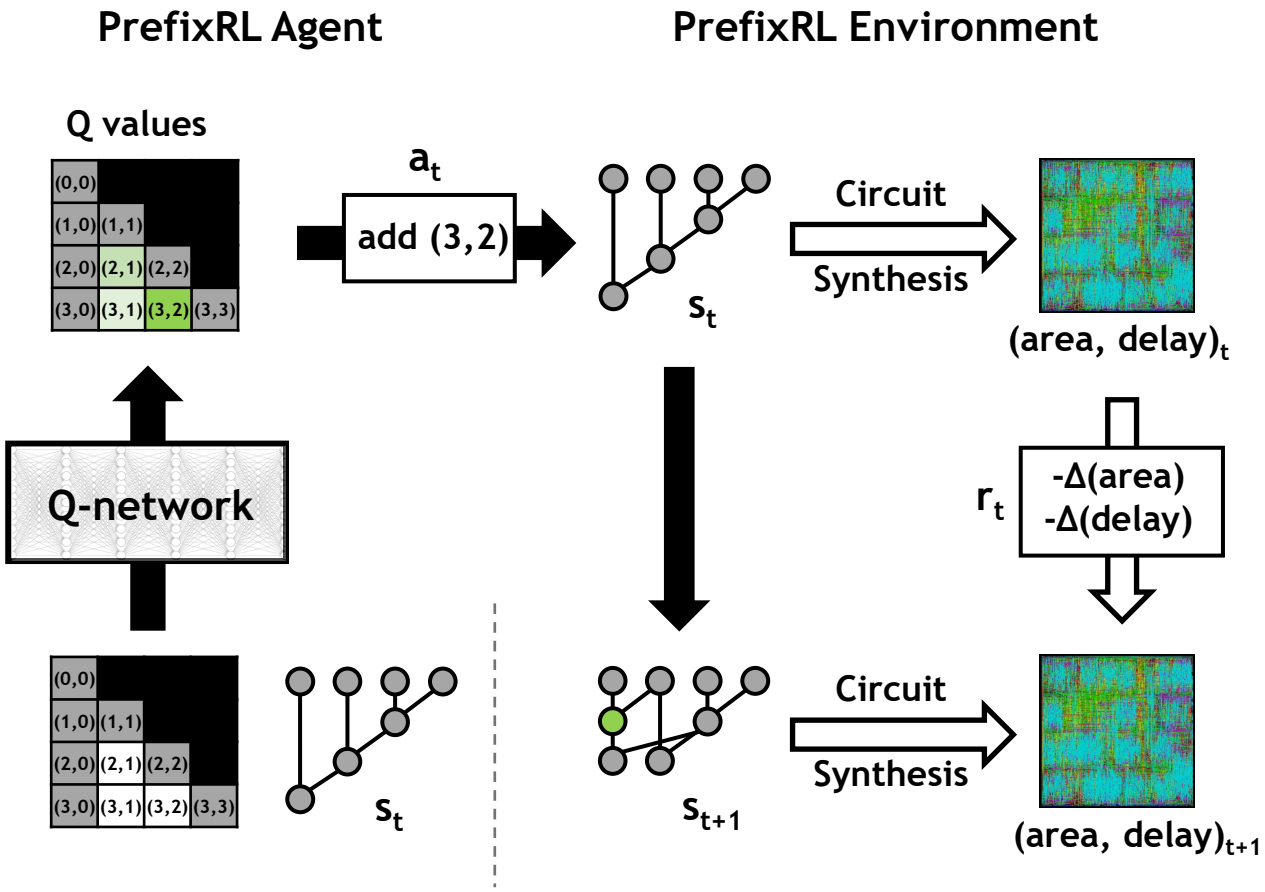


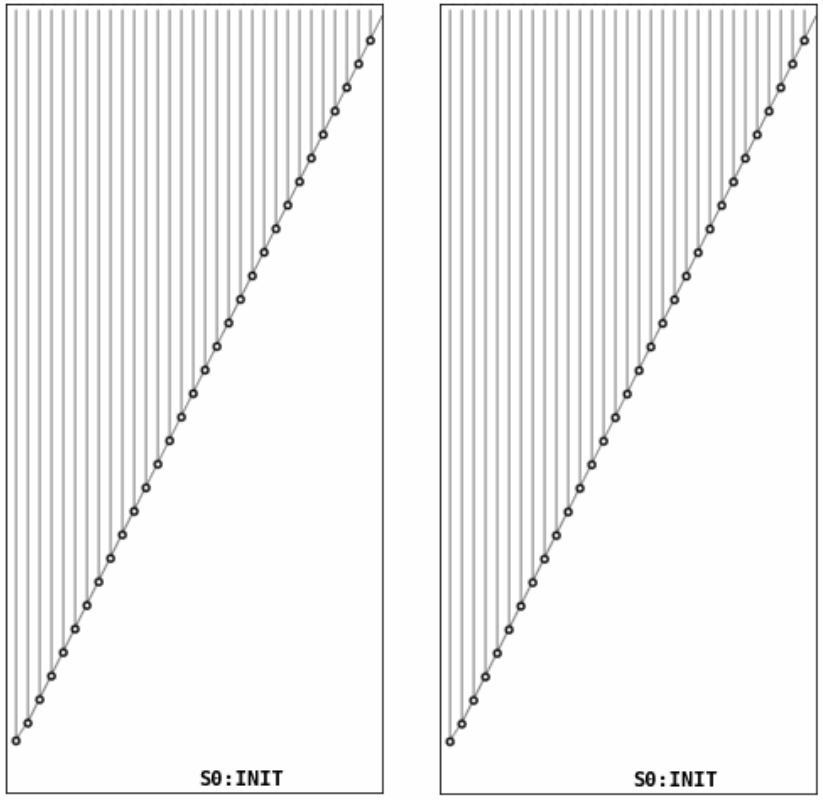


RL Placement Game Sequence (as good as SA on 97% cells)

PREFIXRL: RL FOR PARALLEL PREFIX CIRCUITS

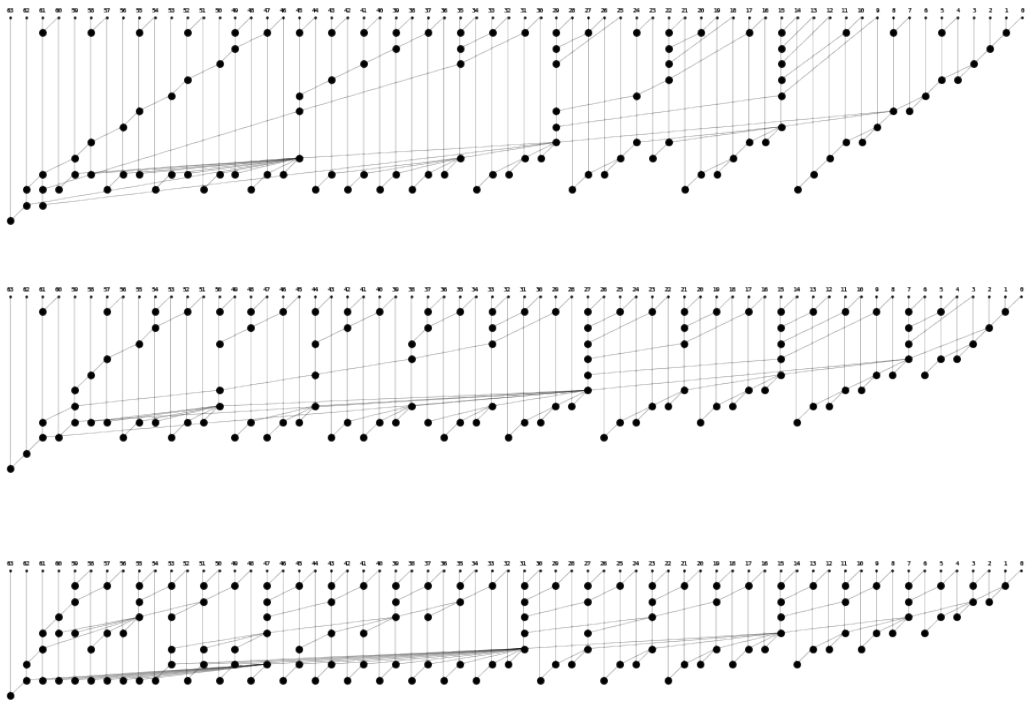
Adders, priority encoders, custom circuits

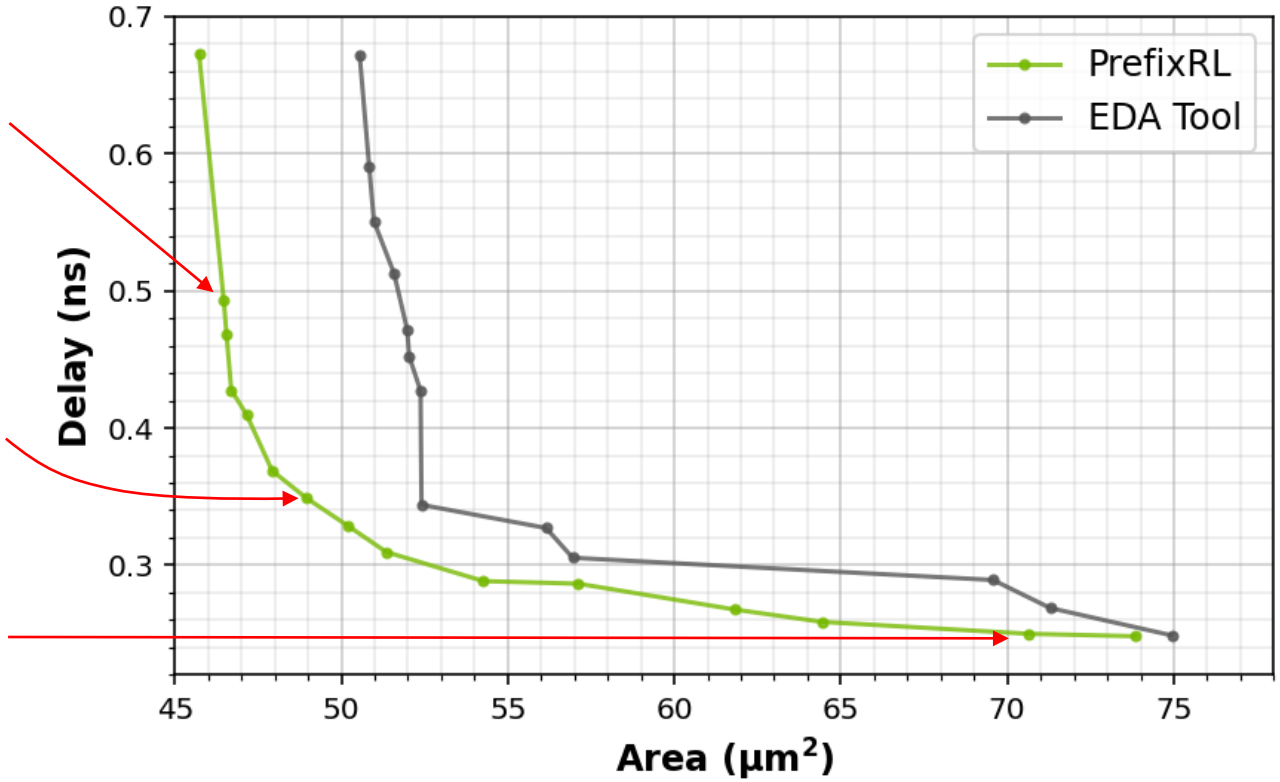




PREFIXRL: RESULTS

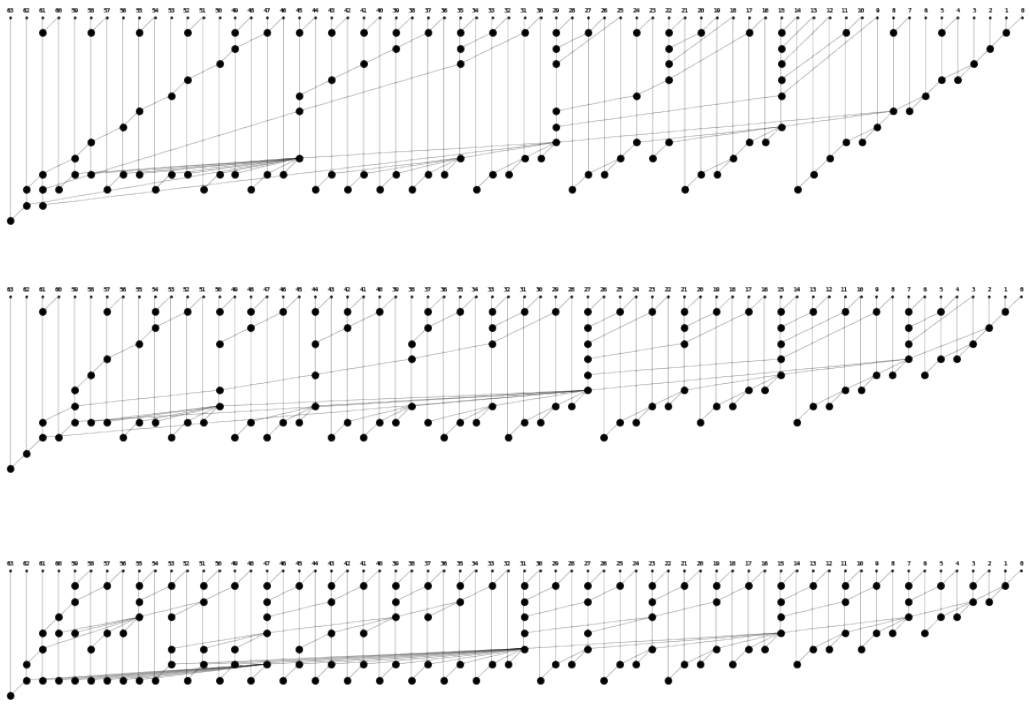
64b adders, commercial synthesis tool, latest technology node

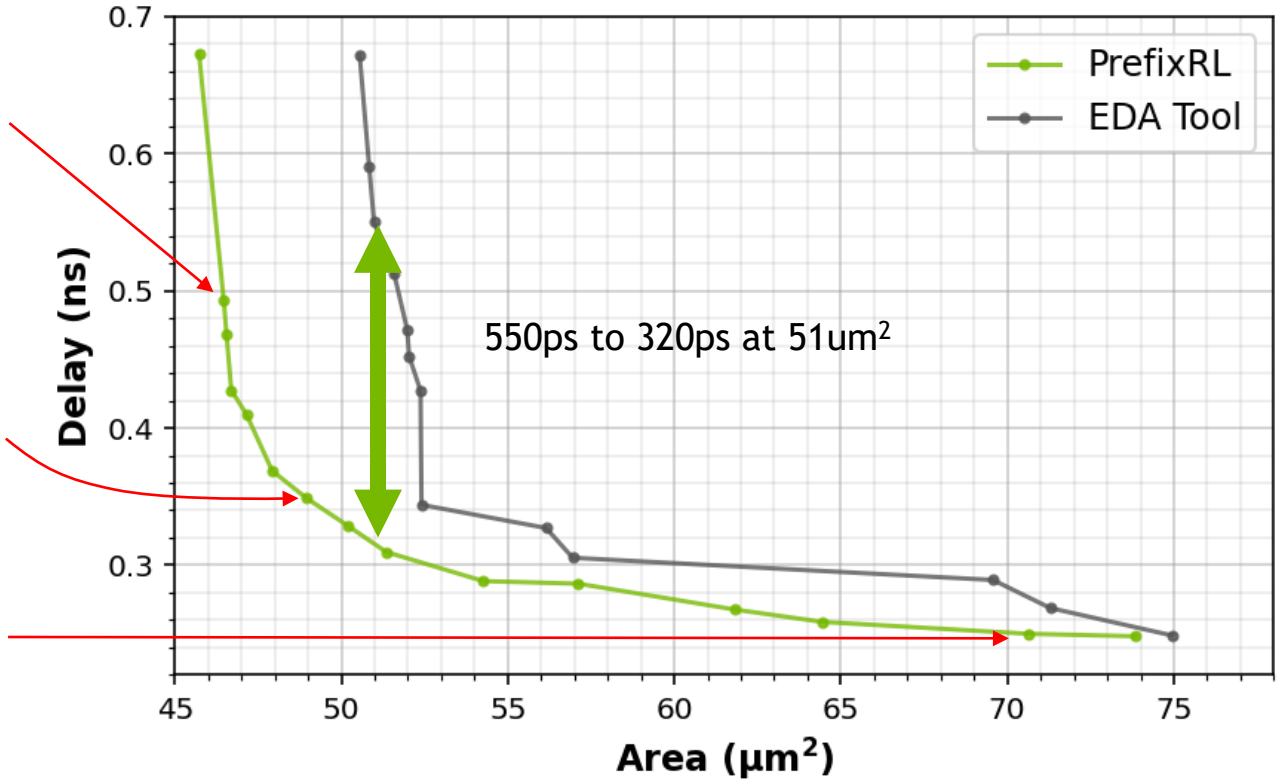




PREFIXRL: RESULTS

64b adders, commercial synthesis tool, latest technology node





Wednesday, December 8th

3:30pm -3:50pm PST

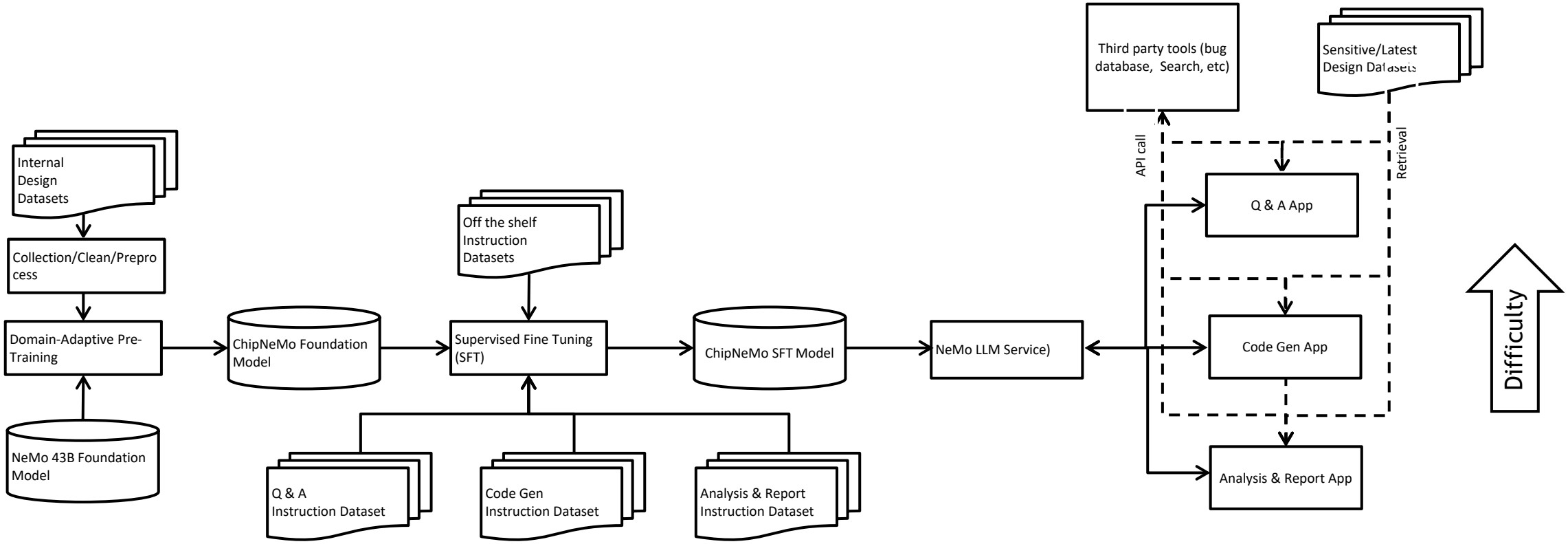
PrefixRL: Optimization of Parallel Prefix Circuits using Deep Reinforcement Learning

Authors: Rajarshi Roy, Jonathan Raiman, Neel Kant, Ilyas Elkin, Robert Kirby, Michael Siu, Stuart Oberman, Saad Godil, Bryan Catanzaro

LLMs

ChipNeMo

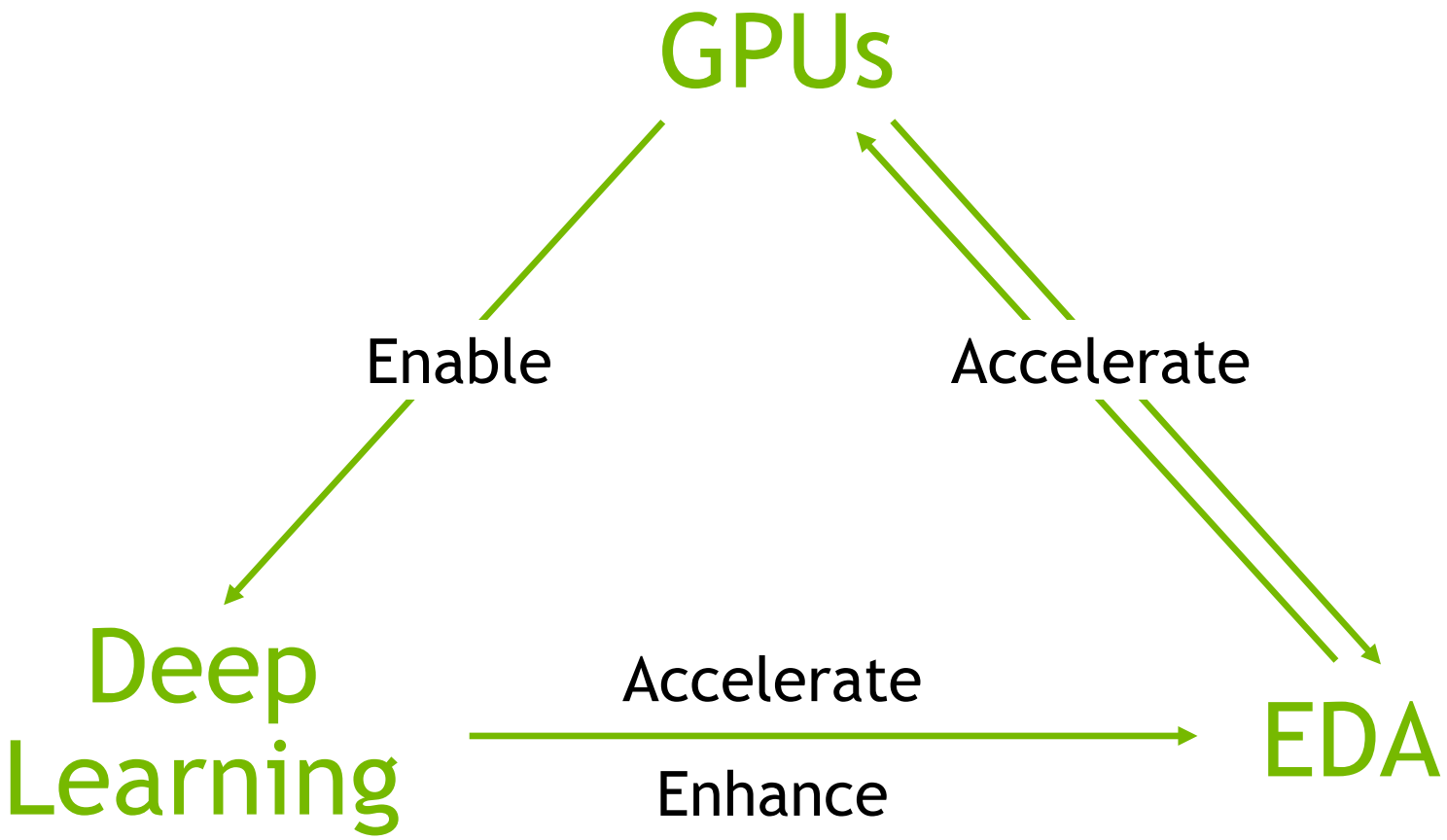
Chip Design Domain Adaptive Model



Use cases:

- Triage a design problem (debug a regression problem, how to fix a bug, etc)
- Code generation (Internal tools scripts and configs, testbenchs, assertions, Verilog, Viva, C++, TCL, Python, etc)
- Hardware design Q & A (Questions about infrastructures, internal tools, flows, ASIC/VLSI/Analog domains)
- Analysis and reporting (summarization, visualization of design data, etc)

Conclusion


GPUs
Enable
Accelerate
Deep Learning
Accelerate
Enhance
EDA

- GPUs are the **engines of deep learning**
- GPU inference performance is **doubling every year**
  - Number systems, complex instructions, sparsity, plumbing
  - Accelerators test new concepts
- GPUs give **>1,000x speedup** on EDA
  - Logic simulation, placement
- ML on GPUs gives fast, accurate **prediction**
  - DRC hot spots, parasitic estimation, power dissipation, routing congestion
- RL on GPUs gives **super-human design** and productivity
  - Prefix RL, NV Cell
- LLMs for design