Explainable-DSE: Agile & Explainable Exploration of Efficient HW/SW Codesigns of DL Accelerators

Semiconductor Research Corporation

Shail Dave[1], Tony Nowatzki[2], Aviral Shrivastava[1]

[1]School of Computing and AI, Arizona State University; [2]School of Computer Science, University of California, Los Angeles

Motivation

Develop framework for explainable DSE of deep learning accelerators that reasons about underlying inefficiencies in designs, achieve efficient designs, takes short time, and can work with several domains.

Background

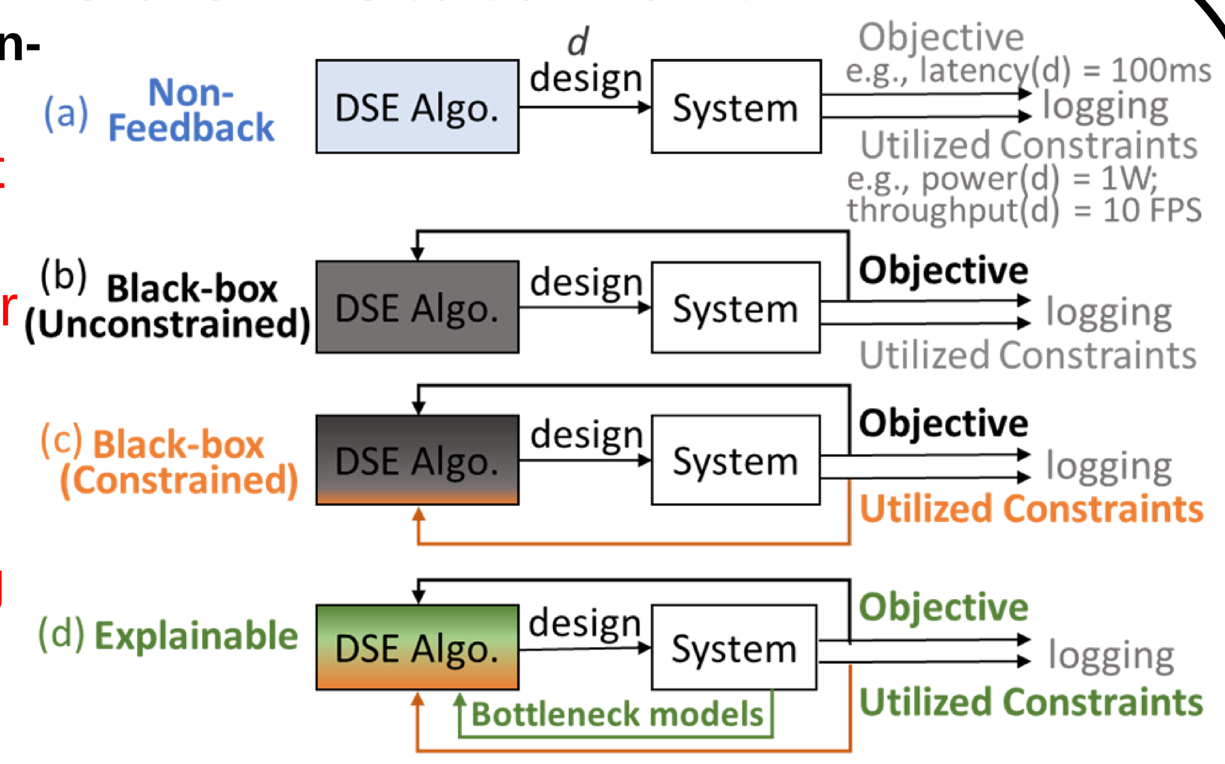
- Effective design space exploration (DSE) requires achieving efficient solutions satisfying constraints under practical exploration budgets
- Deep learning accelerator design space can be vast
  - 10^14 hardware parameter solutions
  - 10^12—10^36 mappings of DNNs layers on each AIHW architecture
- Recent industrial/academic approaches use black-box DSEs
  - Evolutionary algorithms
  - ML-based approaches
- Cost models take milliseconds-minutes; Practical explorations can afford only 1000s of iterations

Table 1: Design Space for Edge DNN Accelerators.
Data: int16; Freq. 500 MHz; Constraints: Throughput>=40/10 FPS (vision light/heavy), 120/530/176k samples/second (NLP: Transformer/BERT/wav2vec2); Area < 75 mm^2; Max. power < 4W. Objective: Minimize latency.

Parameter	Values	Options
PEs	64, 128, ..., 4096	7
L1 buffer (B)	8, 16, ..., 1024	8
L2 buffer (kB)	64, 128, ..., 4096	7
Offchip bandwidth (MBPS)	1024, 2048, 4096, 6400, 8192, 12800, 19200, 25600, 38400, 51200	10
NOC datawidth	16*i; i: [1, 16]	16
Physical unicast (×4)	PEs*i / 64; i: [1, 64]	64^4
Virtual unicast (×4)	2^{3i}; i: [0, 3]	4^4

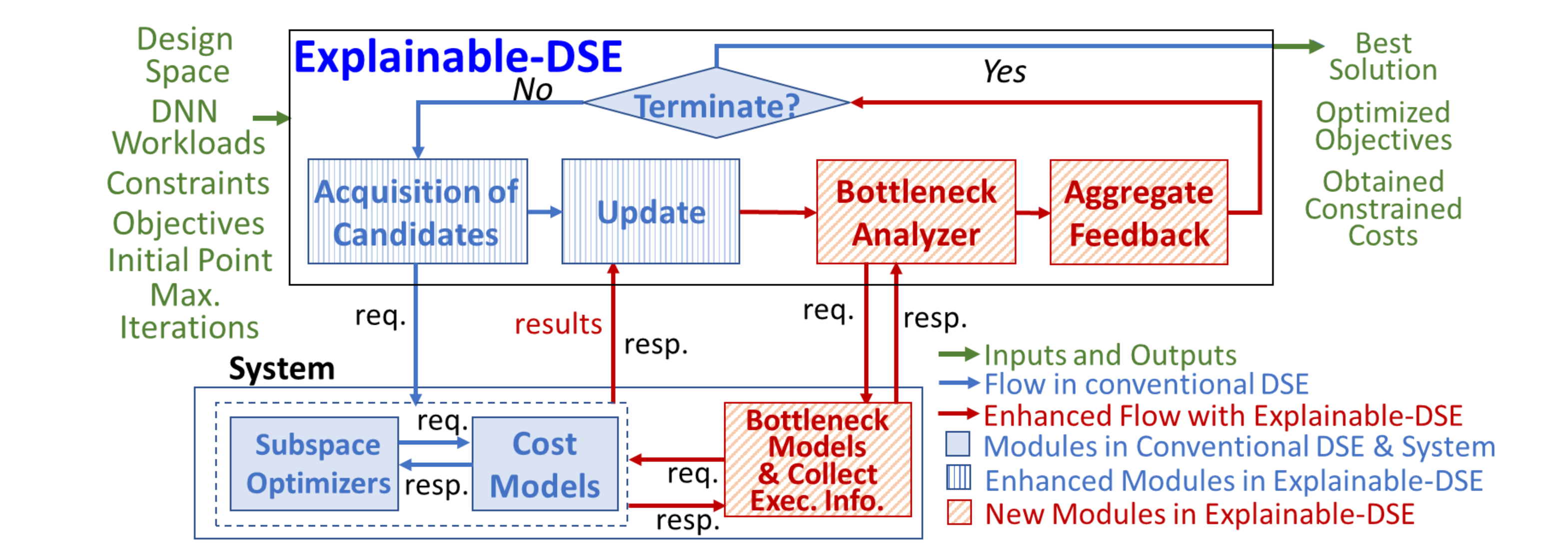
Problem Statement

- Black-box DSEs are **non-explainable**
  - Can't reason about why sampled configurations incur higher costs and how to change parameters to address underlying execution inefficiencies.

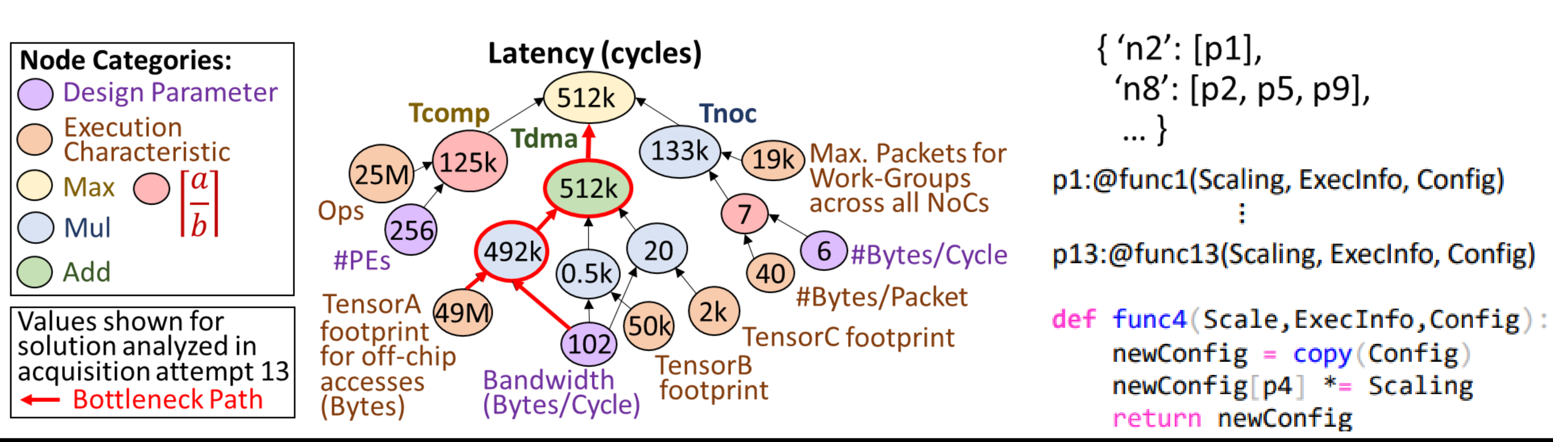


- Process single cost value for a DNN vs. per-layer costs and do not leverage information about domain-specific bottlenecks
- They require excessive trials (thousands), which leads to
  - Low Efficiency of obtained solutions (several-fold)
  - Low Feasibility (most of acquired solutions do not meet constraints)
  - Cannot do Runtime/Practical DSE (Takes days—weeks)
- Explainability can be enabled by domain-awareness and bottleneck analysis
  - Prior works lack formal specification of how to express bottleneck analysis in a generic manner for an effectual design space exploration

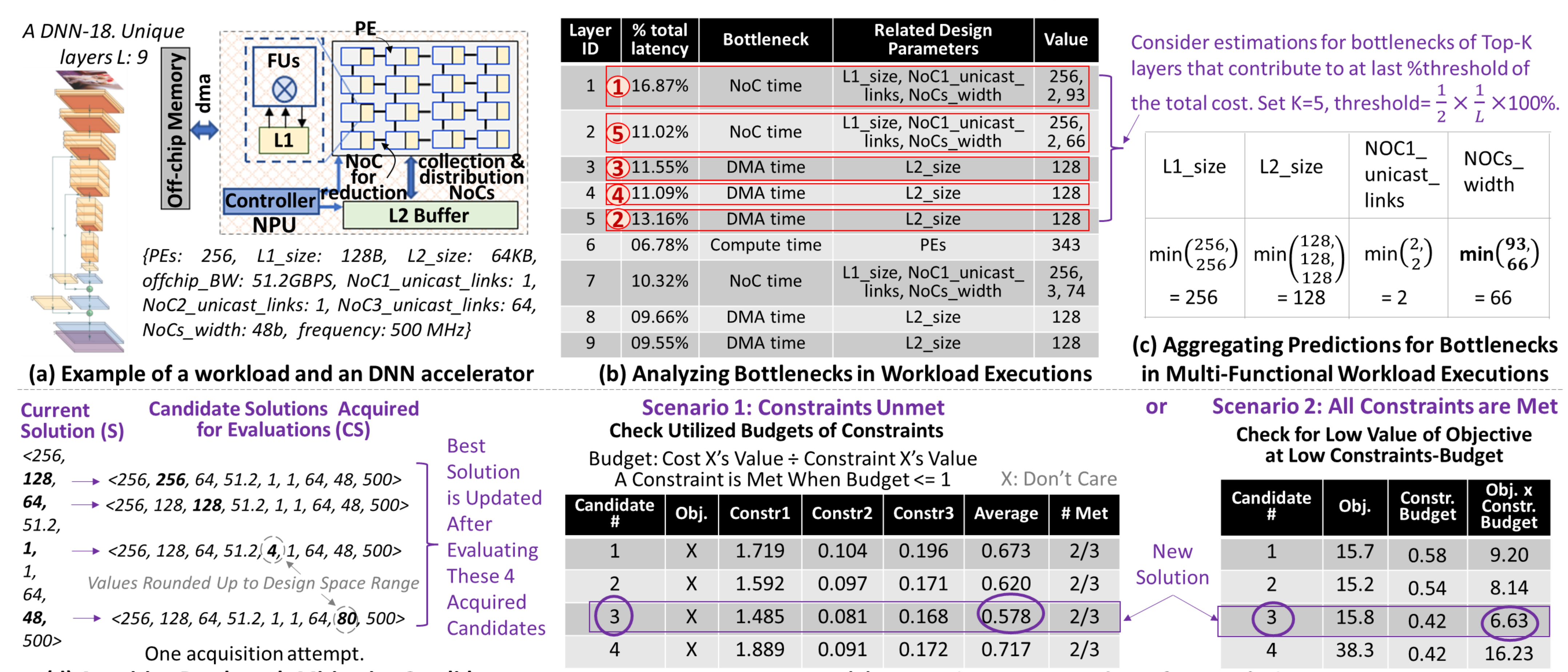
DSE Using Bottleneck Analysis & API for Expressing Domain-Specific Bottleneck Models



- Through API, architects/tools can Specify Bottleneck Graph of target costs and appropriate Scaling of Parameters



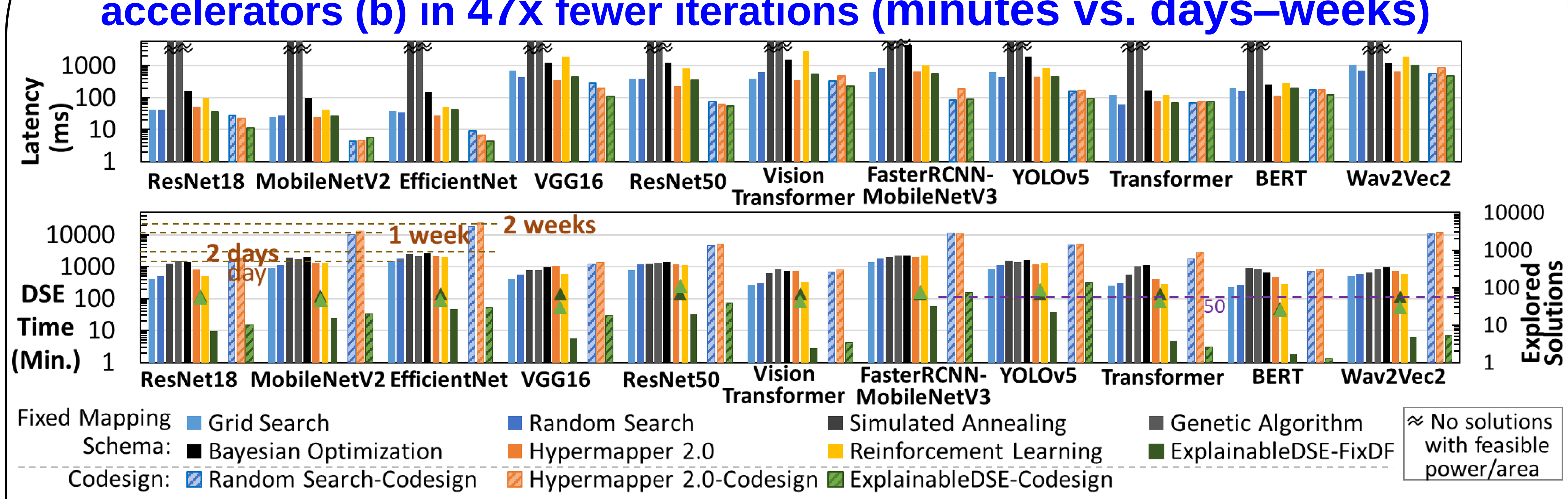
- Workflow example for Solution Acquisitions addressing Multiple Bottlenecks and Constraints-Budget-Awareness



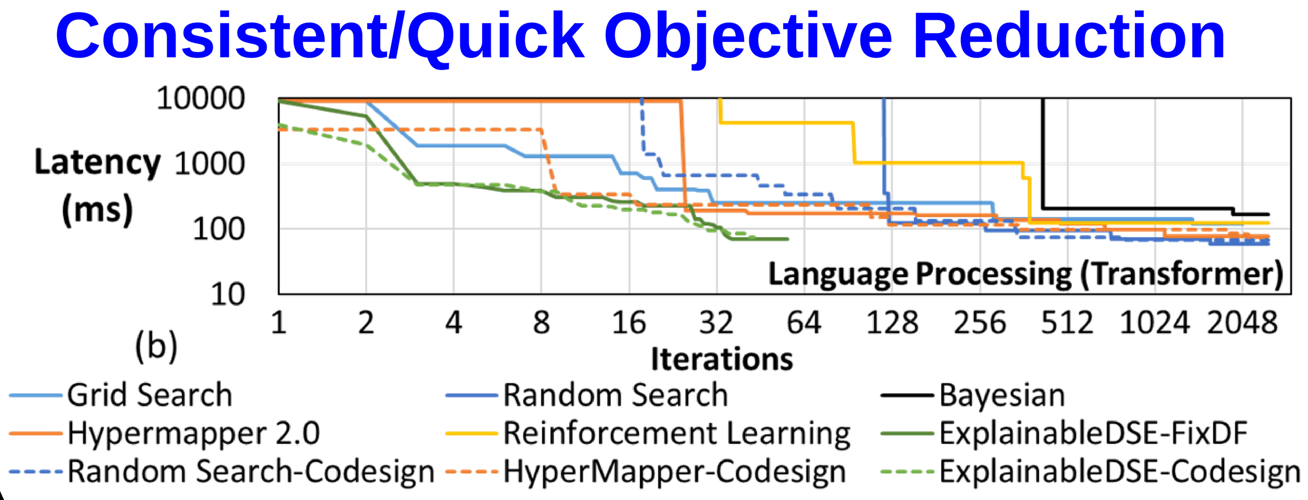
(a) Example of a workload and an DNN accelerator
(b) Analyzing Bottlenecks in Workload Executions
(c) Aggregating Predictions for Bottlenecks in Multi-Functional Workload Executions
(d) Acquiring Bottleneck-Mitigating Candidates
(e) Constraints-Aware Update of Best Solution

Results

Explainable-DSE achieved (a) 6x lower latency codesigns for edge DL accelerators (b) in 47x fewer iterations (minutes vs. days–weeks)



Consistent/Quick Objective Reduction



- At every acquisition, Explainable-DSE reduces objective by 30% vs ~1% by non-explainable DSEs
- Constraints-budget-aware DSE majorly explores feasible solutions (87% vs. 21% for non-explainable DSEs)
- Including software design space in the exploration enables 4.24x better solutions and tightly coupled codesign (search time increases from 21 to 64 minutes; from 16 to 21 minutes for most DNNs)

Conclusions

Know what designs you explore and why!

- Black-box DSE cannot explain why solutions incur high costs and underlying inefficiencies or goodness of potential configurations to try.
- Explainability can be enabled with bottleneck analysis of costs and gray-box estimations for mitigations.
- Formalizing specification for bottleneck models can help applying DSE to broader workload domains.
- Address diverse bottlenecks in multi-workload executions by aggregating mitigation predictions.
- With Explainability, DSE can find several-fold efficient solutions at runtime!

Future Directions

- Automate bottleneck model generation for a variety of domain-specific architectures
- Introducing bottleneck mitigations in ML-based DSE
- Overcome local-optima-convergence due to greediness for mitigating primary bottlenecks

Publications

Explainable-DSE: Agile & Explainable HW/SW Codesign Exploration using Bottleneck Analysis. In ACM International Conference on Architectural Support for Programming Languages & Operating Systems (**ASPLOS**)

Website and Related Material: