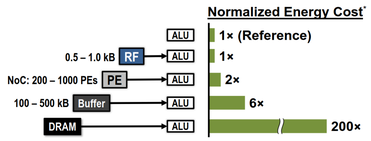
# LoopTree: Exploring the Fused-layer Dataflow Accelerator Design Space

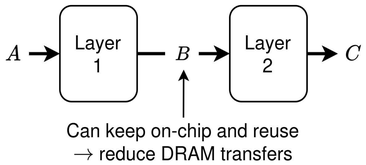
**Michael Gilbert**, Yannan Nellie Wu, Joel S. Emer, and Vivienne Sze

## Introduction

- Data movement is expensive.




- Reuse opportunity across DNN layers:

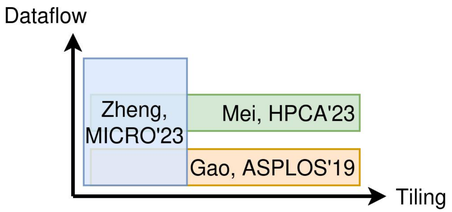



A challenge: <u>on-chip memory is limited</u>

- Address challenges of fusion via ***comprehensive exploration***.

  There are many ways to fuse. Specifically, we highlight three design choices:
  **dataflow, tiling**, **recomputation**.

- Gap: prior work is scattered and not comprehensive.

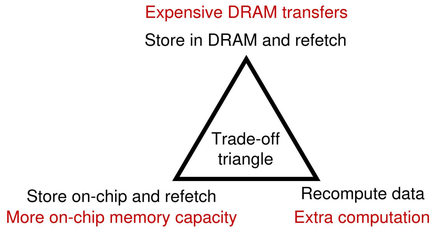



## Methodology (1)

**(1) Explore systematically!**

Choose the right representation of the design space.

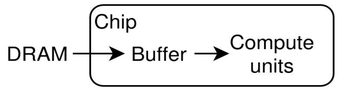
**Insight 1.1**
Choose exactly one: refetch, reuse, recompute

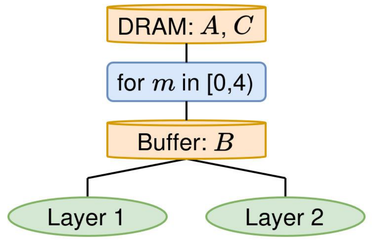



Need to represent only 2, the remainder is implied.

**Insight 1.2**
Represent fusion choices as a *LoopTree mapping*.

*E.g.,* a simple hardware:




**Loop nodes** represent **dataflow**:
Suppose *m* a dimension in *B*.
Layer 1 produce B[0]
Layer 2 consume B[0]
Layer 1 produce B[1]
Layer 2 consume B[1]
...

**Storage nodes** represent **tiling of stored data** :
Iter. 1: keep B[0] in Buffer
Iter. 2: keep B[1] in Buffer
...
Keep *A* and *C* in DRAM.

Nodes:

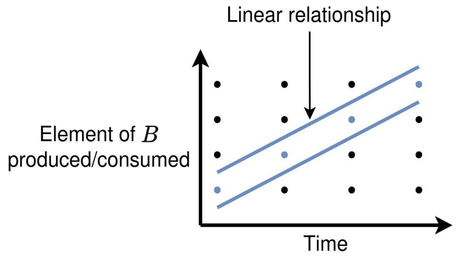

Note, many features of LoopTree not shown: parallelization, pipelining, storing data in multiple levels of memory.

## Methodology (2)

**(2) Evaluate designs with a versatile model**

Need a fast and accurate model that supports a wide design space

**Insight 2.1**
DNNs are compute- and data-intensive: Simulation takes a long time! But the operations **follow a predictable pattern**.




Can use a compact data structure to represent hardware states and actions over time.

Benefit of choosing the right representation: The LoopTree mapping abstraction makes calculating these states and actions easy.

**Insight 2.2**
Eliminate assumptions of specific dataflow or tiling

**Frame analysis as set operations on data.**

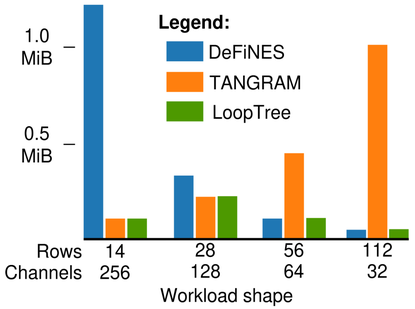
Data to reuse from Buffer
=
Data required for operations ∩ Data in Buffer

Data to refetch from DRAM
=
(Required data – Reused data) ∩ Data in DRAM

## Results

**Comprehensive exploration enables efficient fusion.**

On-chip memory capacity required for fusing ResNet (lower is better)




## Conclusion

For more efficient designs
- consider a wide design space,
- explore systematically, and
- use a versatile model

## Learn More

Appeared in
*TCAS-AI*
*ISPASS 2023*

## Acknowledgements

This work is sponsored by the MIT AI Hardware Program