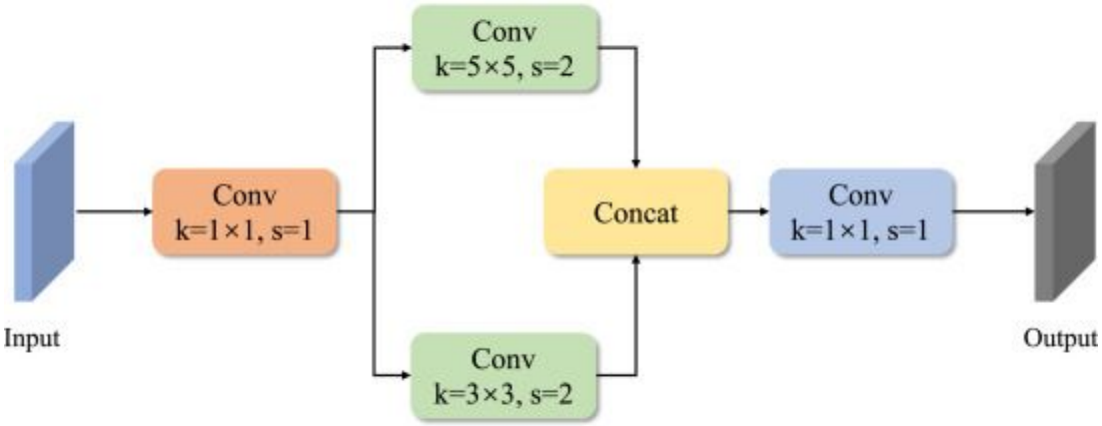
# EEINet: Optimizing Tensor Programs with Derivation-Based Transformations

Presented by
Ruquaiya Shuaibu (rs2377)

# Motivation

- A DNN is a tensor program, which is a DAG containing tensor operators performed on a set of tensors.
- They are critical in a variety of tasks, but expensive to run
- Current optimization methods only work based on predefined operators
- But this leaves very limited discovery space



# Current Optimisers

Operator level

- Uses the idea of compute/schedule
- Schedules faster kernels for each operator

tvm

Ansor

Graph level

- Reorganizes operators to optimise graph
- Enumerates possible subgraphs over predefined operators

jiazhihao/**TASO**

The Tensor Algebra SuperOptimizer for Deep Learning

thu-pacman/**PET**

PET: Optimizing Tensor Programs with Partially Equivalent Transformations and Automated Corrections

Both sides stay within a fixed vocabulary of existing operators

# Limitations

Restricted to Predefined Operator Representable (POR) Transformations

- Transformations restricted to existing ops (Conv, Matmul, Add, etc)
- Optimizers cannot create new operators
- Small optimization spaces and limited speedups

Think of a calculator that can only
add, subtract, multiply and divide

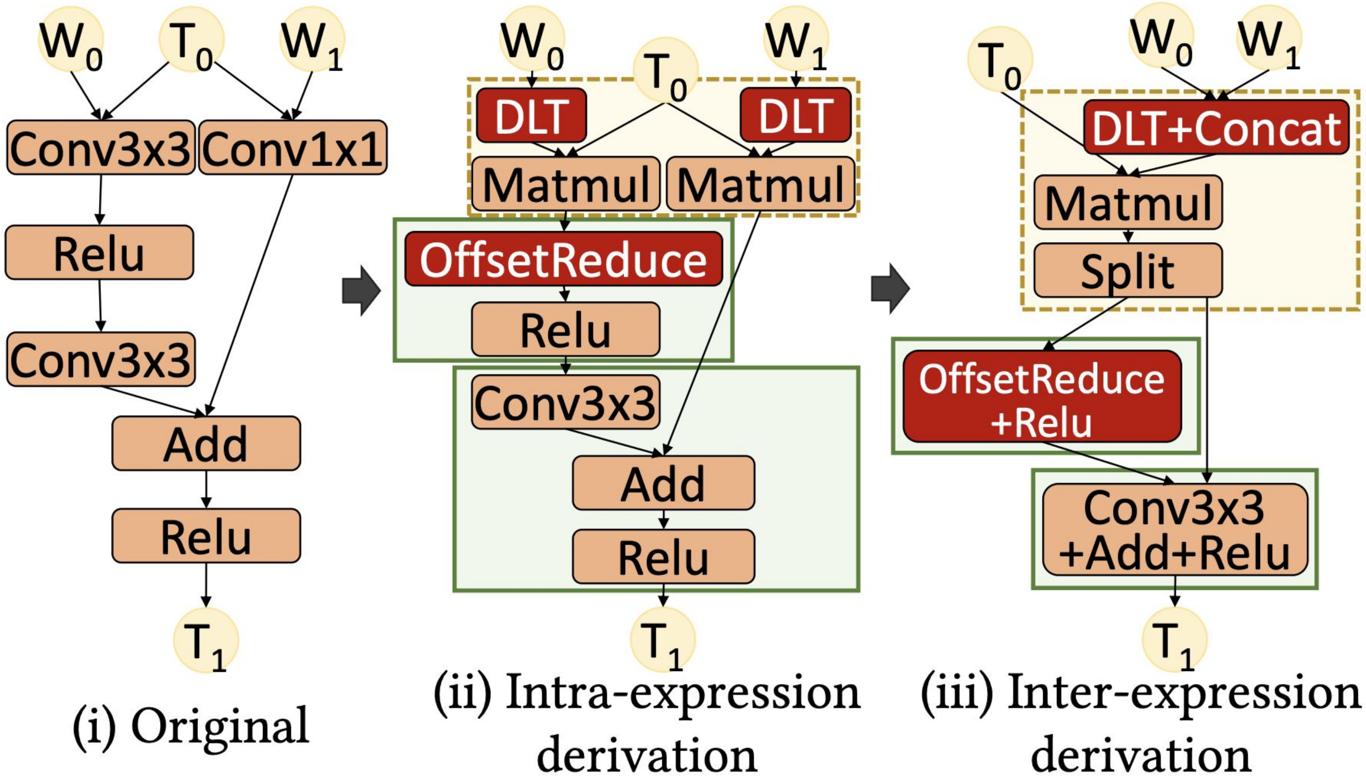
Liminting, right?

# EINNet’s Key Idea

General tensor algorithms to rewrite the math itself

Examples

1. Rewrite an op to do the same thing, but faster
2. Replacing old ops with new ones + customised ones
3. Reorganize graphs more deeply




(a) Optimizations found by EINNET

# The Three Main Challenges Addressed

Automatically discovering transformations over general expressions

- There are infinitely possible algebraic expressions
- Cannot rely on manually-written rules or superoptimzation
- EINNET uses derivation rules to systematically rewrite expressions

Turning expressions into kernels

- General expressions may not match any known operator
- Need a way to:
  - Match parts existing highly optimised kernels
  - Auto-generate kernels for the rest (eOperators)

## Searching hygge space efficiently

- Transformations may require long sequences of derivations
- Brute-force search is impossible
- So a two-stage, distance guided search is used to find promising transformations

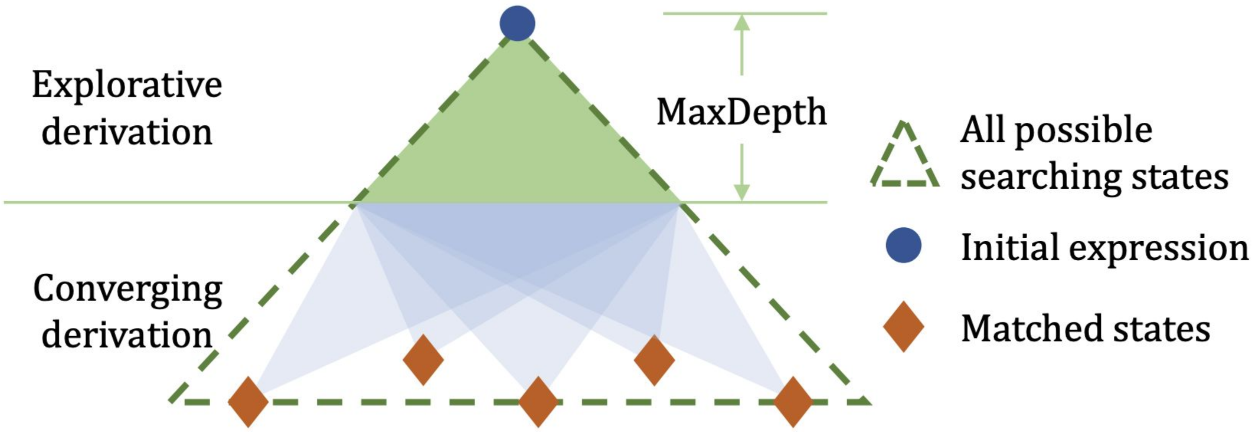
# Search Strategy

Exploration Derivation Stage

- Apply all possible derivation rules to expand search space

Converging Derivation Stage

- Use “expression distance” to guide search toward target operators



# Evaluation

On 7 DNN models:

- Up to 2.72x speedup over best existing optimizer
- Tested on both A100 and V100 GPUs
- Particularly strong on Conv-heavy models (ResNet, DCGAN, FSRCNN)
- EINNet discovers optimizations these systems cannot even represent

# Key Takeaways/Contributions

- Extending the POR optimization space
- Present the first attempt to explore a significantly larger expression search space
- Built EINNET achieving 2.72x speedup over existing tensor program optimizers

# Weakness/Criticisms

- Compilation time
- Requires non-trivial search time
- Some generated operators may still be memory-bound
- Can it handle dynamic graphs?

# Questions?