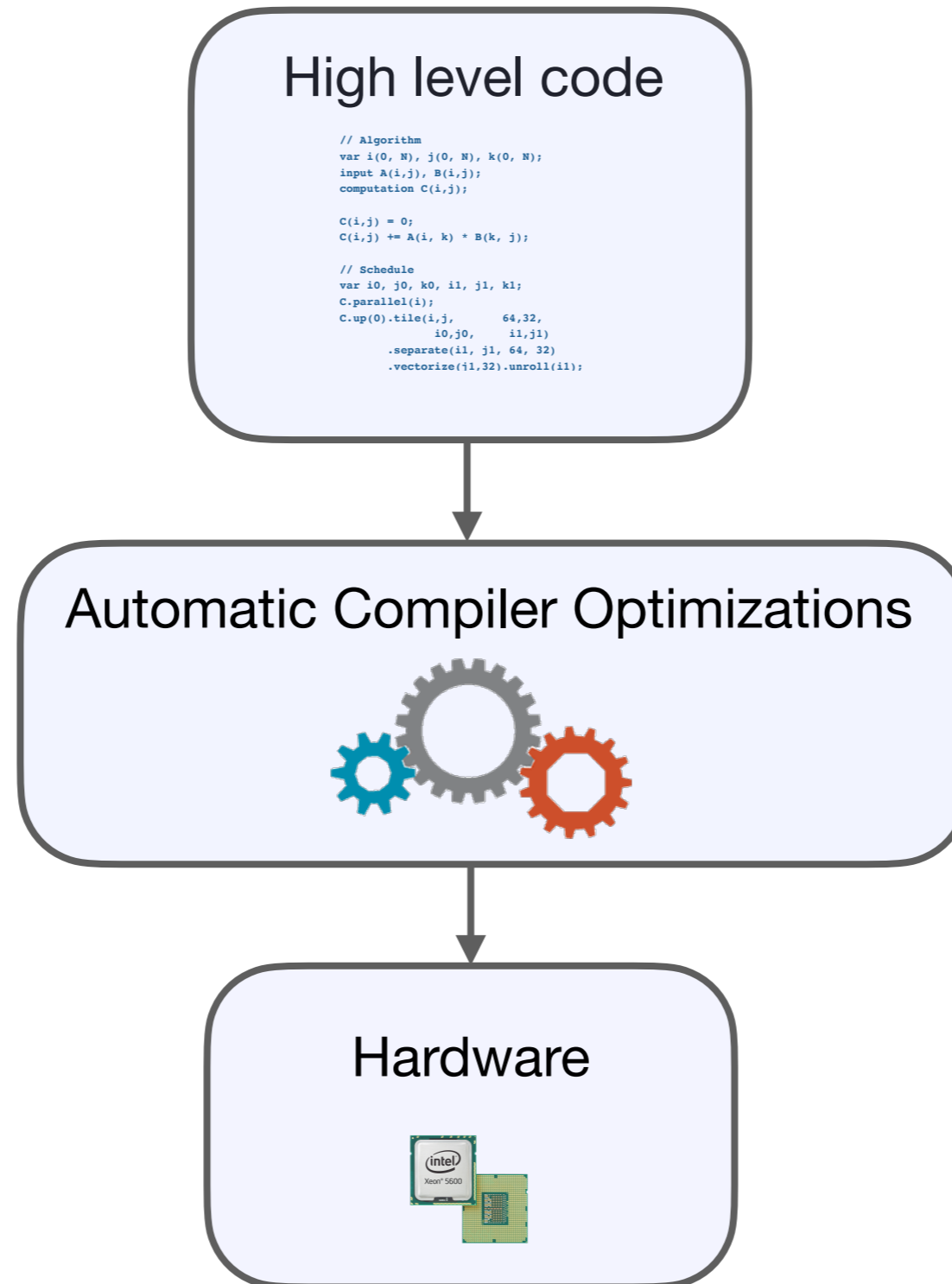


An Autoscheduler for Unimodular Transformations with a Deep Learning Cost Model

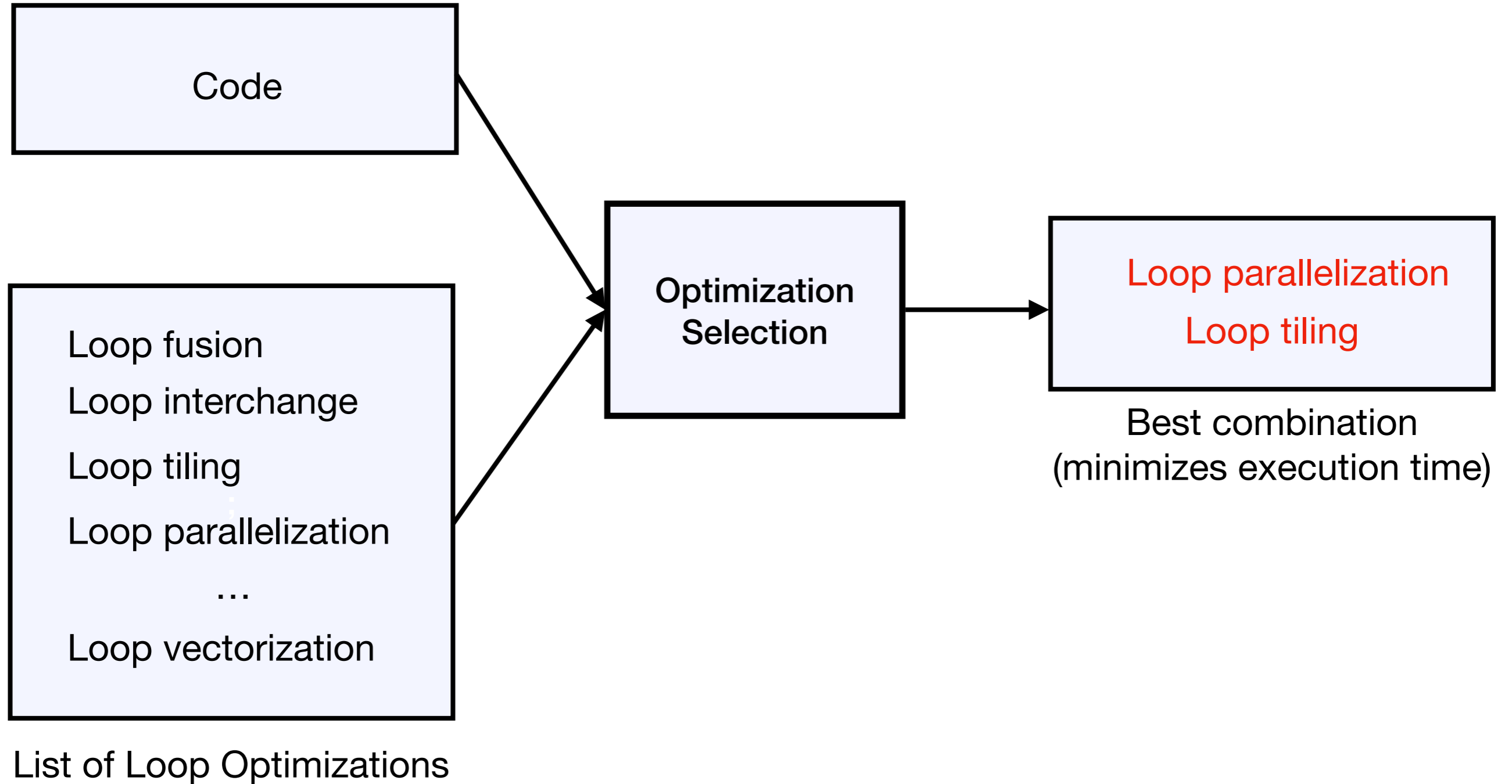
Massinissa Merouani, Khaled Afif Boudaoud, Iheb Nassim Aouadj, Nassim Tchoulak
Fatima Benbouzid-Sitayeb, Benatchba Karima, Hugh Leather, **Riyadh Baghdadi**



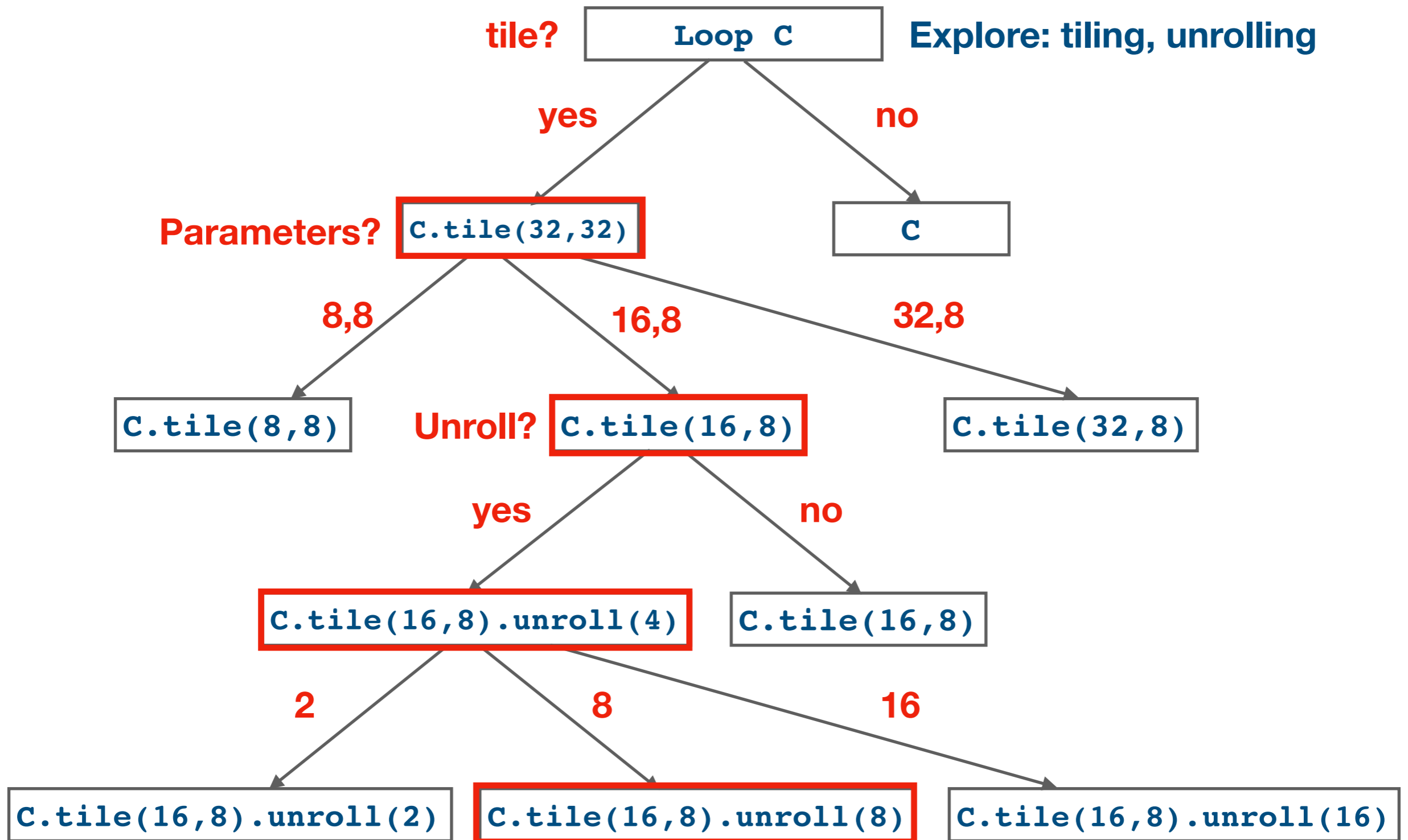
Goal: Enable Compilers to Automatically Optimize Code



Optimization Selection

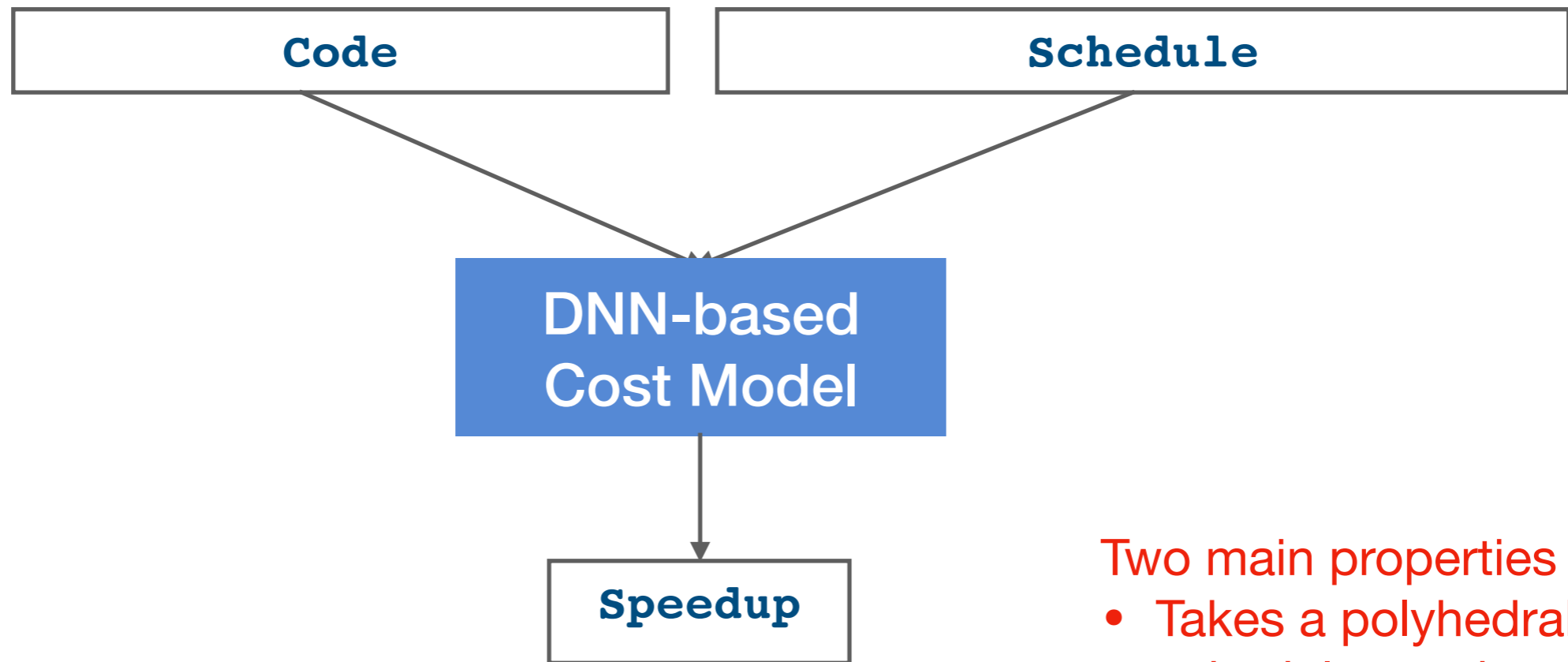


Optimization Selection



Goal

Build a **deep learning model** for speedup prediction



- Two main properties
- Takes a polyhedral schedule matrix
 - Support affine unimodular transformations

Difference with Previous Work (Baghdadi et al., 2021)

Schedule

(Baghdadi et al., 2021)

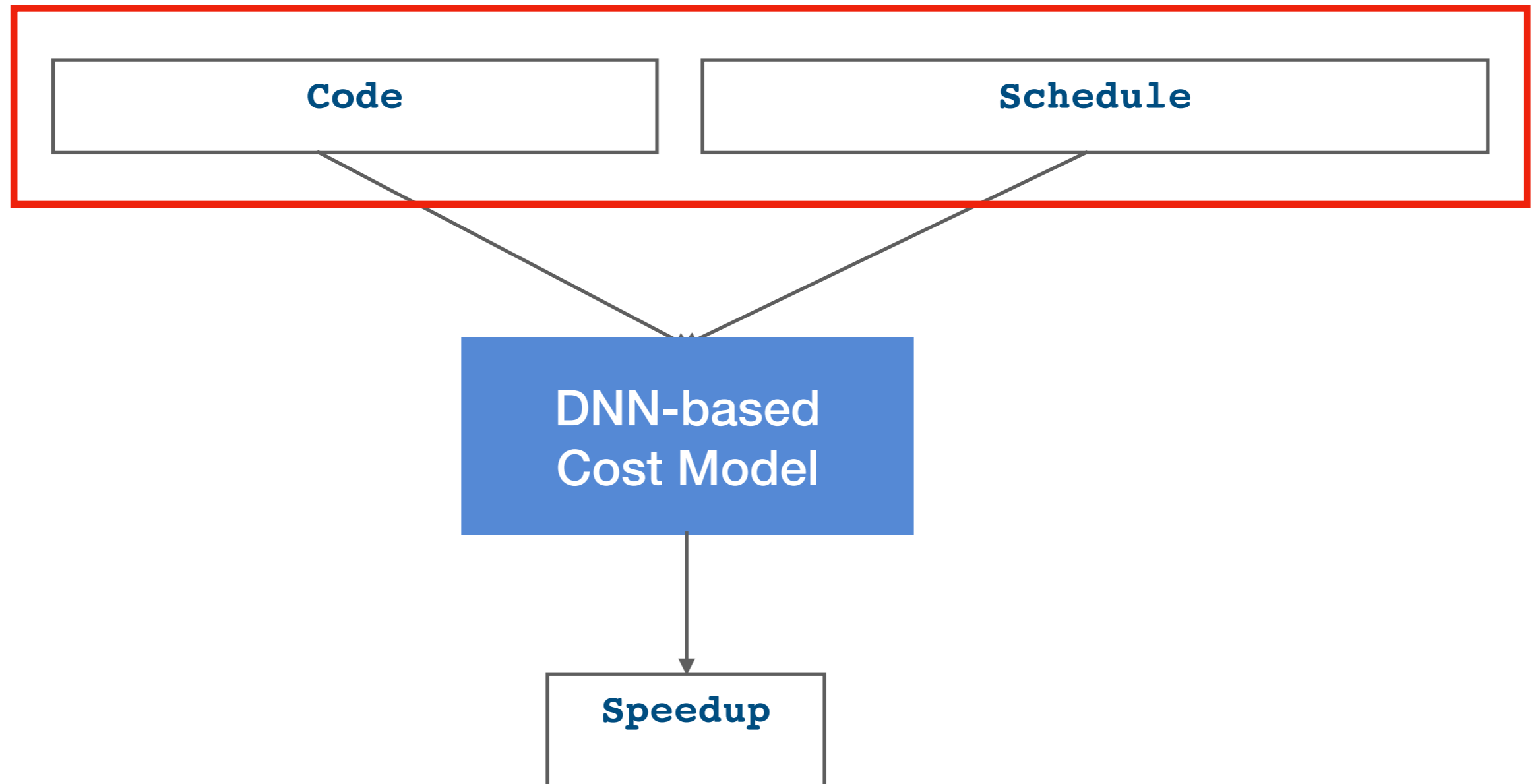
This work

| | Tiled | Interchanged | Parallel | Vectorized | Unrolled |
|----------|-------|--------------|----------|------------|----------|
| i | 1 | 0 | 1 | 0 | 0 |
| j | 1 | 0 | 0 | 0 | 0 |

| | Tiled | Parallel | Vectorized | Unrolled | Schedule Matrix | | | | | | | | | | | | | | | | |
|----------|-------|----------|------------|----------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| i | 1 | 1 | 0 | 0 | <table border="1"> <tr><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> </table> | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | 0 | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | 0 | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 1 | | | | | | | | | | | | | | | | | | |
| j | 1 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | |

DNN-based Cost Model

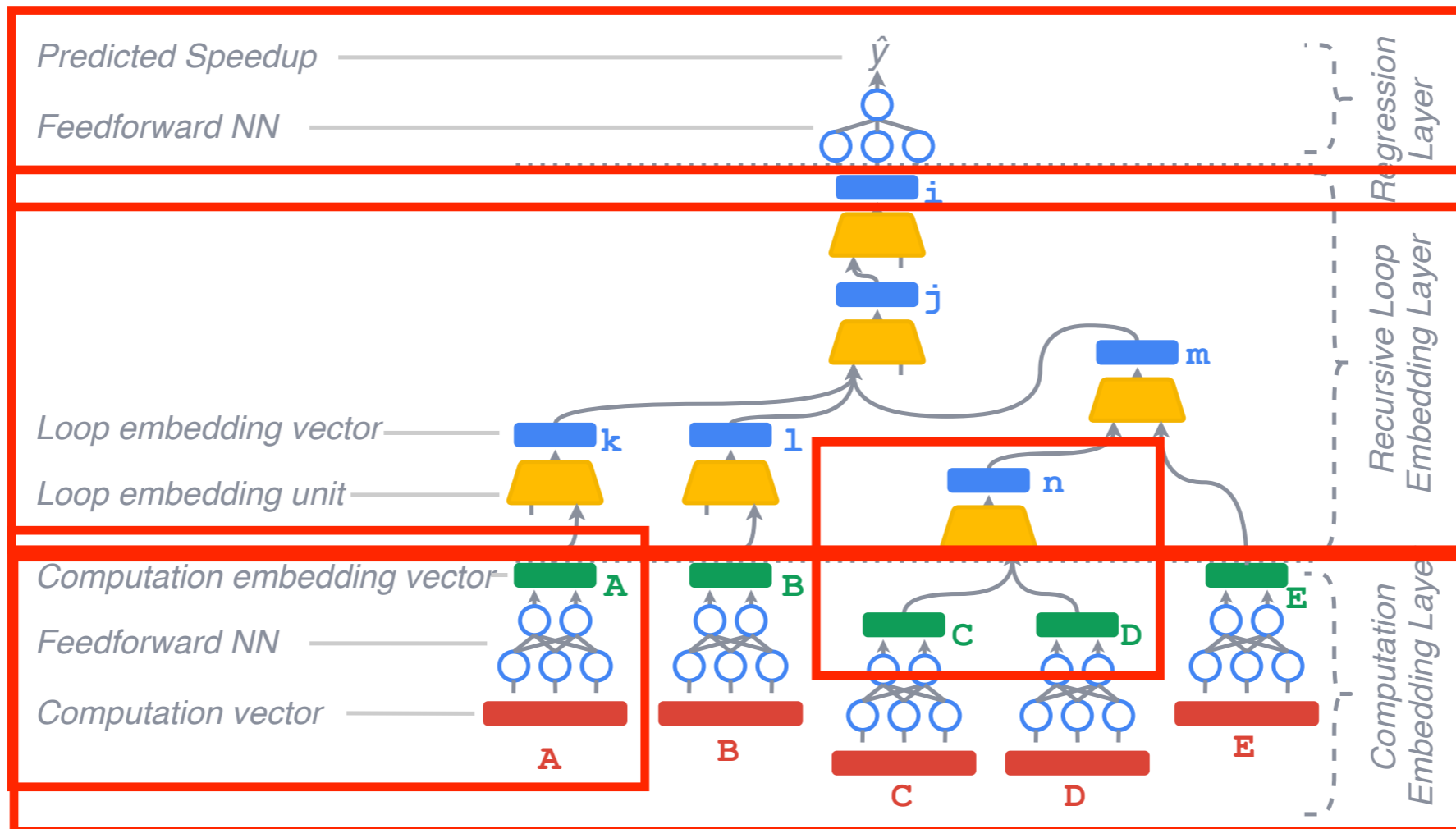
Model



Representation for Code & Optimizations

```
for i:  
  for j:  
    for k:  
      computation A  
    for l:  
      computation B  
    for m:  
      for n:  
        computation C  
        computation D  
      computation E
```

Neural Network Architecture



Training Dataset

- Data generated
 - 13M datapoints
 - Data generation took 8 months on 16 nodes (48 cores)

Preliminary Results

Automatic Optimization

■ Beam Search with Executions ■ Beam Search with Cost Model — Pluto --tile --parallel (Baseline)

