ELE 548: Progress Report

Calvin Higgins

Department of Computer Science and Statistics
University of Rhode Island

December 11, 2025

Project Overview

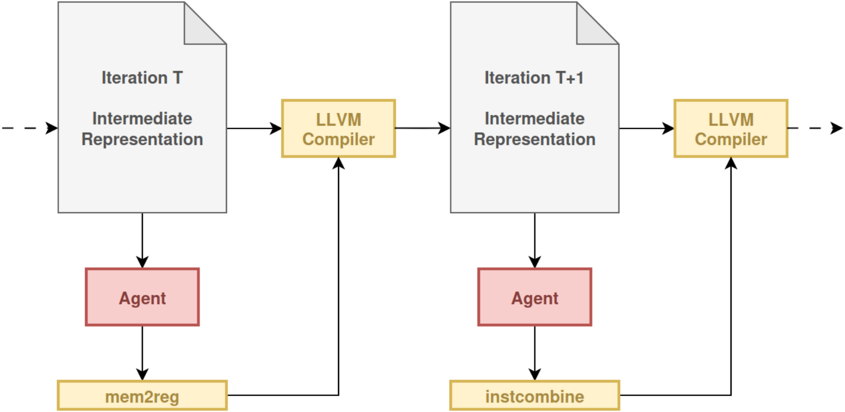


Figure: Reinforcement learning for the phase-ordering problem. Adapted from [6].

CompilerGym

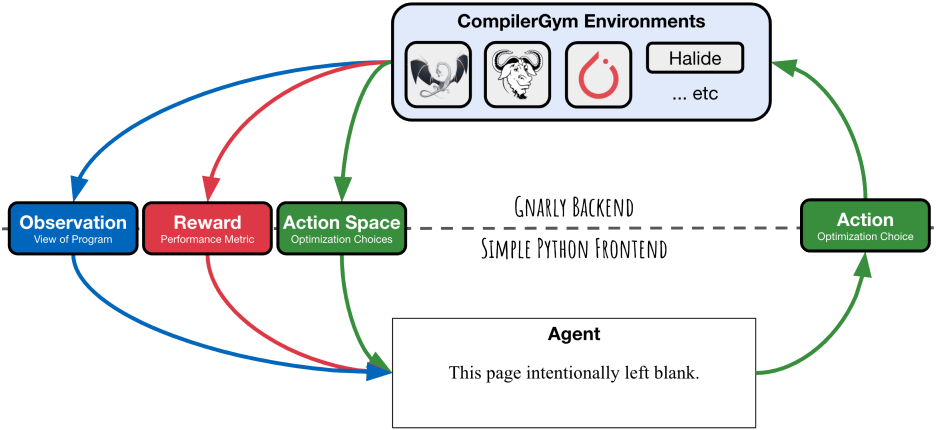
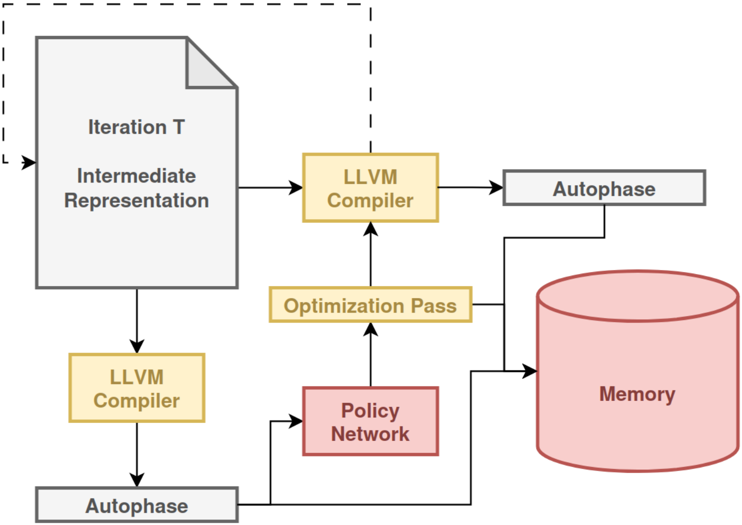
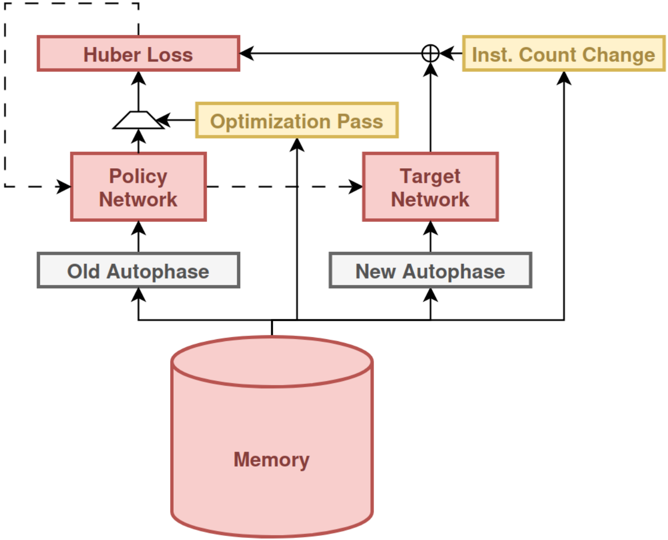


Figure: CompilerGym architecture [5].

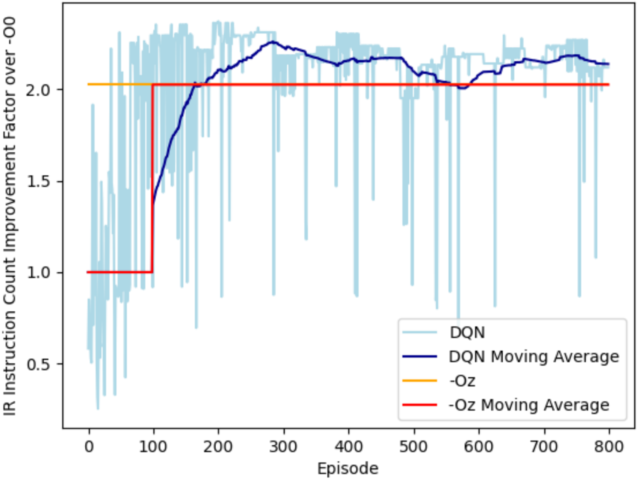
Deep Q-Network Rollout



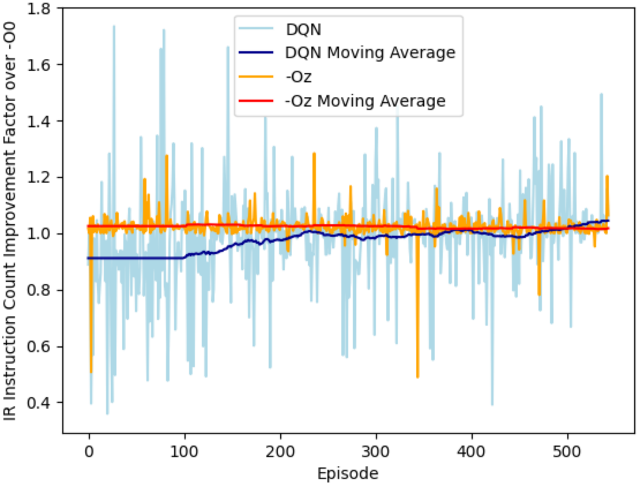
Deep Q-Network Training



QSort ($n = 1$)


IR Instruction Count Improvement Factor over -O0
2.0
1.5
1.0
0.5
0
100
200
300
400
500
600
700
800
Episode
DQN
DQN Moving Average
-Oz
-Oz Moving Average

Tensorflow [1] ($n = 1985$)



Tensorflow [1] ($n = 1985$)

```
Traceback (most recent call last):
  File "dqn2.py", line 357, in <module>
    main(cfg)
  File "dqn2.py", line 352, in main
    train(sys, cfg, stat)
  File "/workspace/.venv/lib/python3.7/site-packages/torch/autograd/grad_mode.py
", line 27, in decorate_context
    return func(*args, **kwargs)
  File "dqn2.py", line 278, in train
    train_episode_batch(sys, cfg, stat)
  File "/workspace/.venv/lib/python3.7/site-packages/torch/autograd/grad_mode.py
", line 27, in decorate_context
    return func(*args, **kwargs)
  File "dqn2.py", line 200, in train_episode_batch
    for initial_cost, episode_cost in zip(initial_costs, episode_costs)
  File "dqn2.py", line 200, in <genexpr>
    for initial_cost, episode_cost in zip(initial_costs, episode_costs)
ZeroDivisionError: division by zero
```

Some programs are optimized to size zero?!

Example Zero Size Program

```c
struct item_head { int dummy; };

int IH_SIZE;
int memcpy (
    struct item_head *to,
    const struct item_head *from,
    int size
);

inline void copy_item_head(
    struct item_head *to,
    const struct item_head *from
) {
    memcpy(to, from, IH_SIZE);
}
```

Only inline function definitions and globals...

ISO C Standard (n3467) Section 6.7.5.8

Any function with internal linkage can be an inline function. For a function with external linkage, the following restrictions apply: **If all the file scope declarations for a function in a translation unit include the inline function specifier without extern, then the definition in that translation unit is an inline definition. An inline definition does not provide an external definition for the function** *and does not forbid an external definition in another translation unit.*

Inline functions are only available inside the file where they are defined!

Hyperparameter Sweep

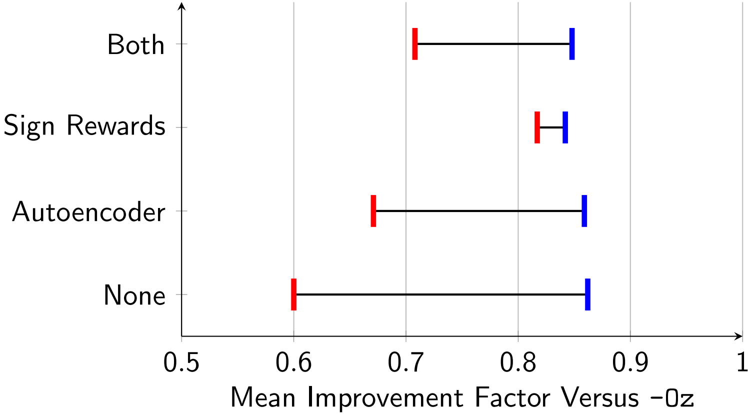
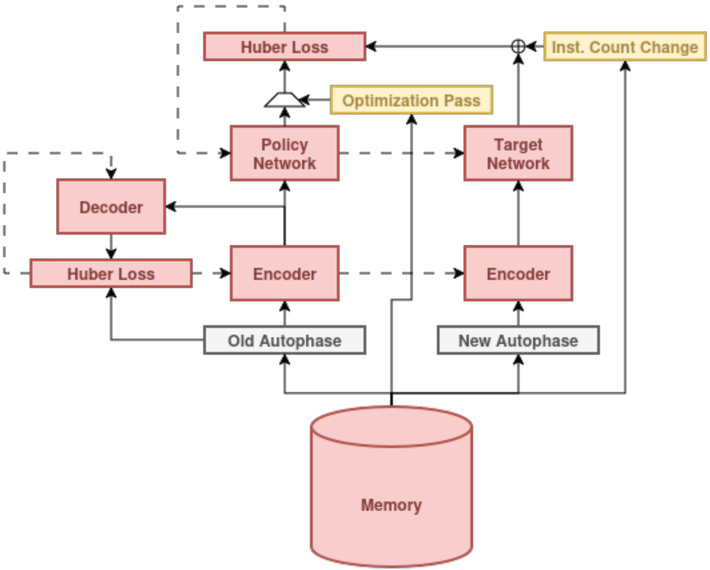
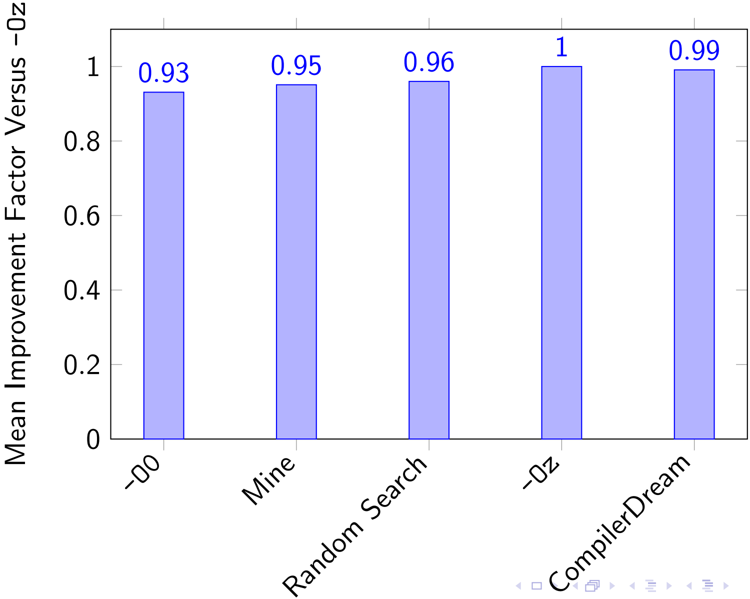


Figure: Geometric mean improvement factor versus -Oz across six benchmark suites. Blue (red) line is maximum (minimum) improvement factor in hyperparameter sweep. All models trained on AnghaBench [11] with Autophase [9] features (1024 episodes, 64 steps).

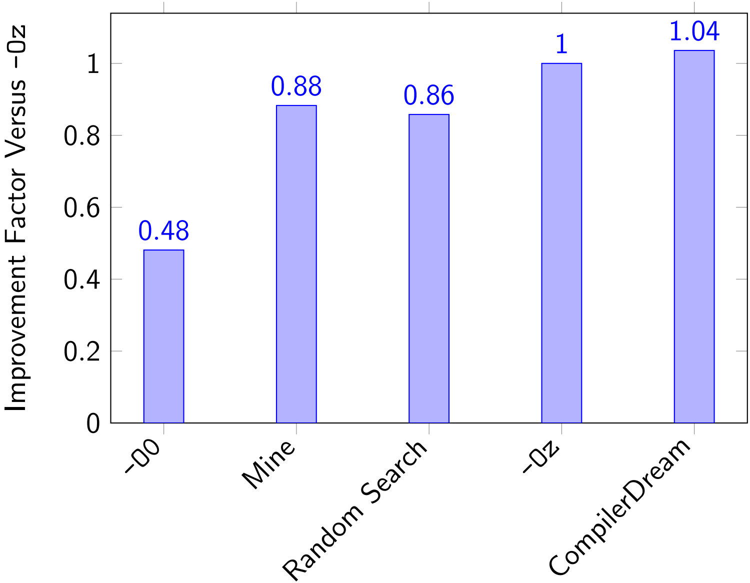
Deep Q-Network Training with Autoencoder



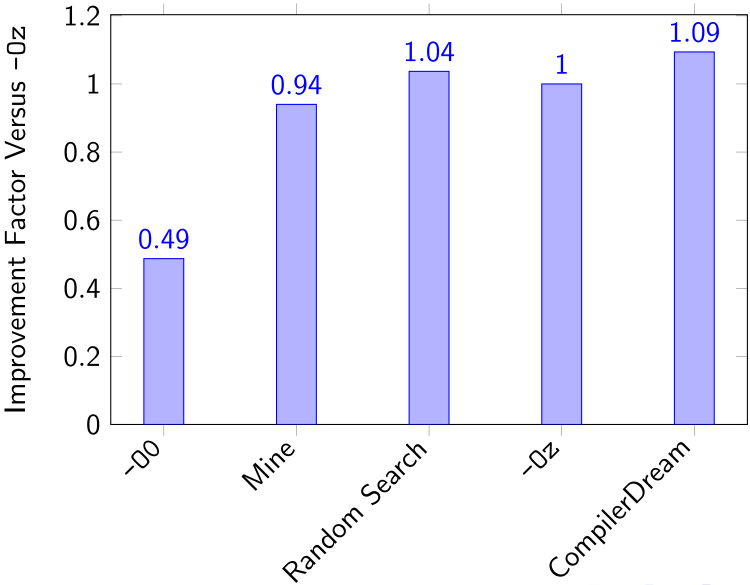
Final Results: BLAS [2] ($n = 300$)



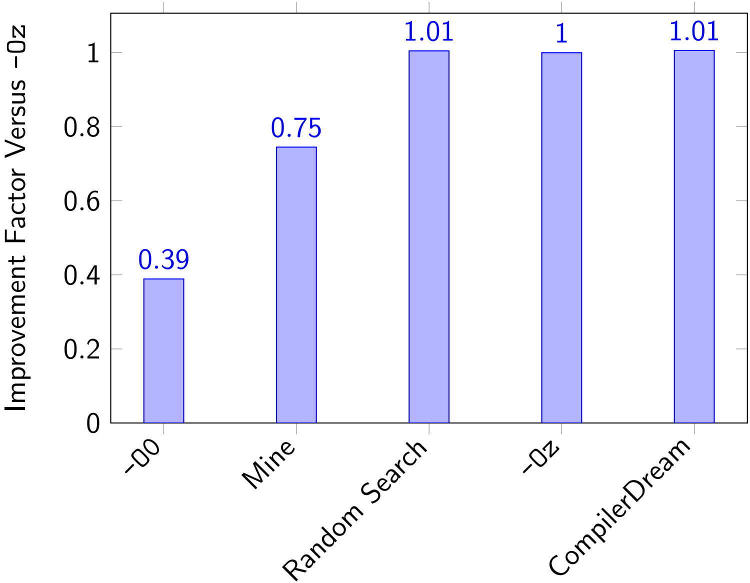
Final Results: cBench [7] ($n = 23$)



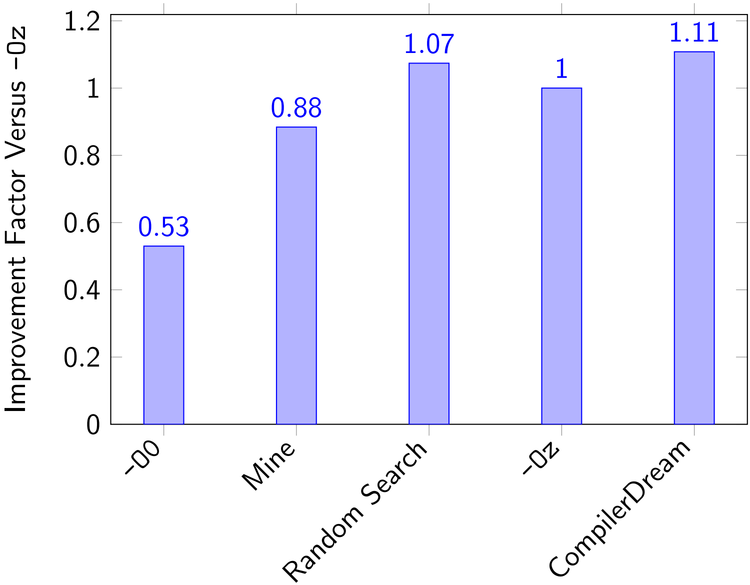
Final Results: CHStone [10] ($n = 12$)



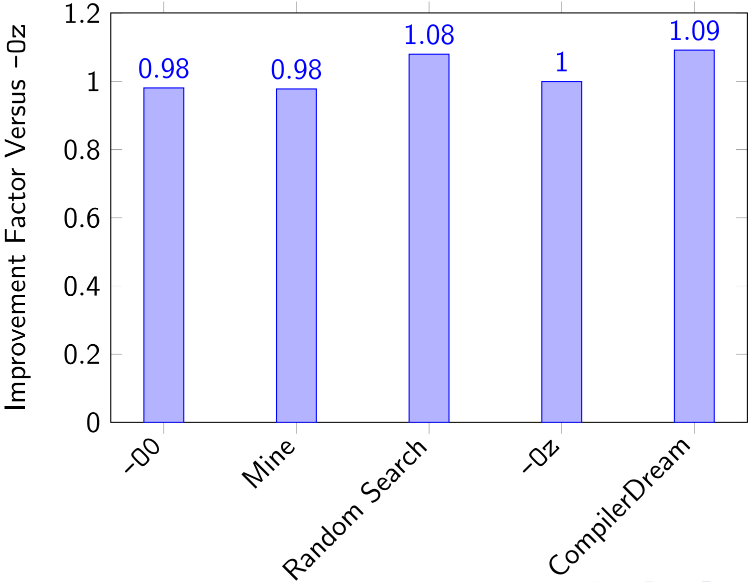
Final Results: MiBench [8] ($n = 40$)



Final Results: NPB ($n = 122$)



Final Results: OpenCV [4] ($n = 442$)



Future Work

CompilerGym:

1. Fix race conditions
2. Parallelize compilation
3. Improve performance (inst2vec [3])

Program Representation:

1. Try Transformer-based embedding models

Architecture:

1. Implement EfficientZero [12]

References I

[1] Martín Abadi et al. "TensorFlow: a system for large-scale machine learning". In: *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*. 2016. ISBN: 9781931971331.

[2] "An updated set of basic linear algebra subprograms (BLAS)". In: *ACM Transactions on Mathematics Software* (2002). DOI: 10.1145/567806.567807.

[3] Tal Ben-Nun, Alice Shoshana Jakobovits, and Torsten Hoefler. "Neural code comprehension: a learnable representation of code semantics". In: *International Conference on Neural Information Processing Systems (NeurIPS)*. 2018.

[4] Ivan Culjak et al. "A brief introduction to OpenCV". In: *International Convention MIPRO*. 2012.

References II

[5] Chris Cummins et al. "CompilerGym: robust, performant compiler optimization environments for AI research". In: *International Symposium on Code Generation and Optimization (CGO)*. 2022. DOI: 10.1109/CGO53902.2022.9741258.

[6] Chaoyi Deng et al. "CompilerDream: Learning a Compiler World Model for General Code Optimization". In: *Conference on Knowledge Discovery and Data Mining (KDD)*. 2025. DOI: 10.1145/3711896.3736887.

[7] Grigori Fursin. "Collective Tuning Initiative". In: *GCC Developers Summit*. 2009.

[8] M.R. Guthaus et al. "MiBench: A free, commercially representative embedded benchmark suite". In: *IEEE International Workshop on Workload Characterization (WWC)*. 2001. DOI: 10.1109/WWC.2001.990739.

References III

[9] Ameer Haj-Ali et al. "AutoPhase: Juggling HLS Phase Orderings in Random Forests with Deep Reinforcement Learning". In: *Machine Learning and Systems (MLSys)*. 2020.

[10] Yuko Hara et al. "CHStone: A benchmark program suite for practical C-based high-level synthesis". In: *International Symposium on Circuits and Systems (ISCAS)*. 2008. DOI: 10.1109/ISCAS.2008.4541637.

[11] Anderson Faustino da Silva et al. "AnghaBench: a suite with one million compilable C benchmarks for code-size reduction". In: *International Symposium on Code Generation and Optimization (CGO)*. 2021. DOI: 10.1109/CGO51591.2021.9370322.

[12] Weirui Ye et al. "Mastering Atari Games with Limited Data". In: *International Conference on Neural Information Processing Systems (NeurIPS)*. 2021.