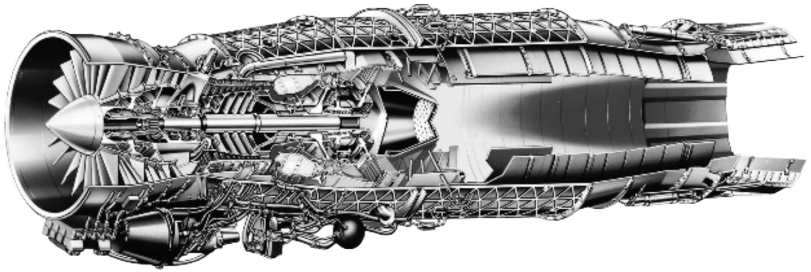
NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE



Afterburner: *Reinforcement Learning Facilitates Self-Improving Code Efficiency Optimization*

Mingzhe Du *(G2204045F)*
College of Computing and Data Science (CCDS)

1 Background & Motivation

Given an array of integers `nums`, sort the array in ascending order and return it.

```python
# Solution A
def sortArray(self, nums):
    i = 0
    while i < len(nums)-1:
        j = i + 1
        while j < len(nums):
            if nums[i] > nums[j]:
                nums[i],nums[j] = nums[j], nums[i]
            j += 1
        i += 1
    return nums
```

Runtime
5714 ms 🦕

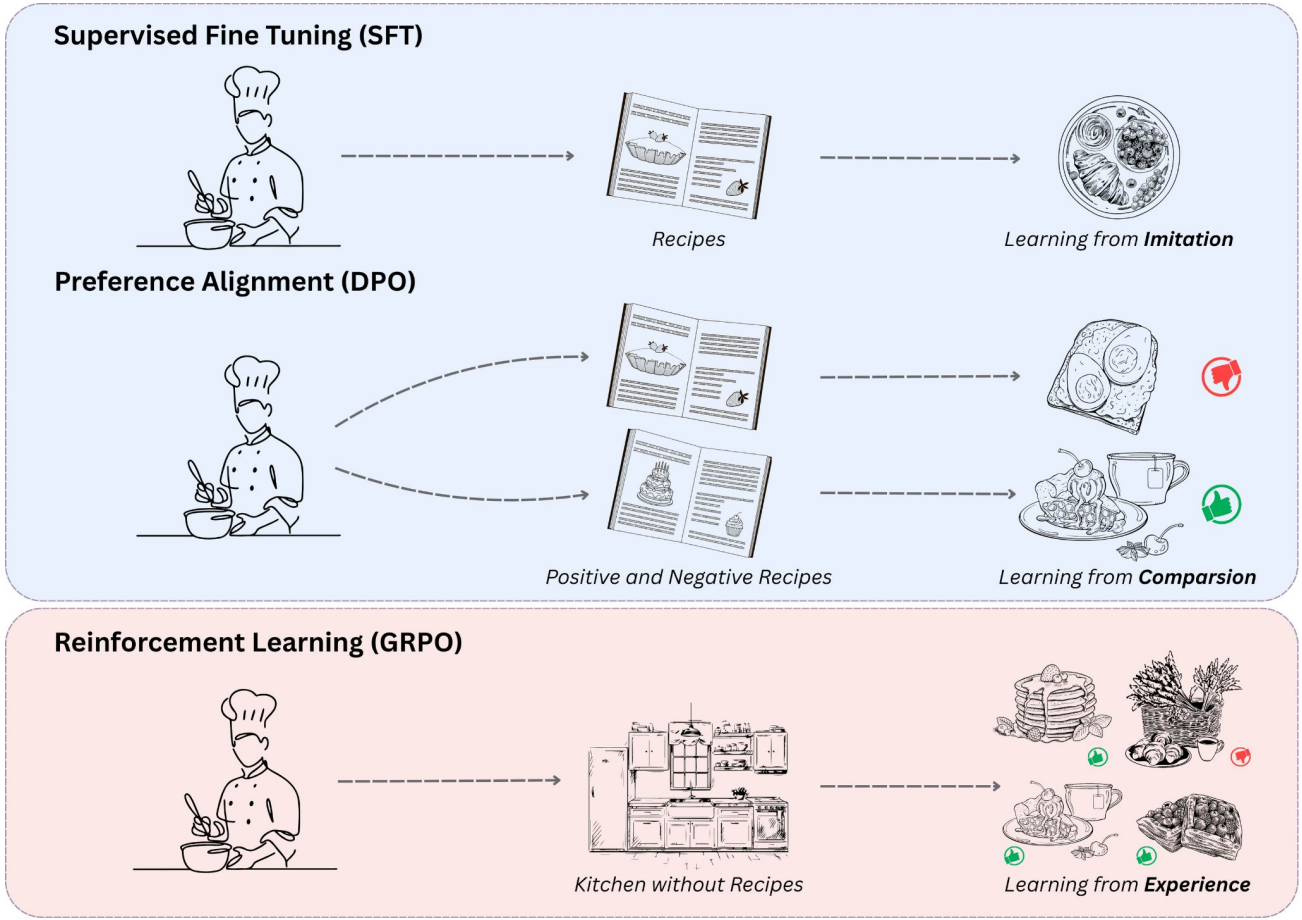
Functional Correctness: **Passed** ✅
Computational Efficiency: **Slow** 😕

```python
# Solution B
def sortArray(self, nums):
    def quicksort(nums, l, r):
        if r - l <= 1: return
        # Function partition not shown for clarity
        pivot = partition(nums, l, r)
        quicksort(nums, l, pivot)
        quicksort(nums, pivot+1, r)
    quicksort(nums, 0, len(nums))
    return nums
```

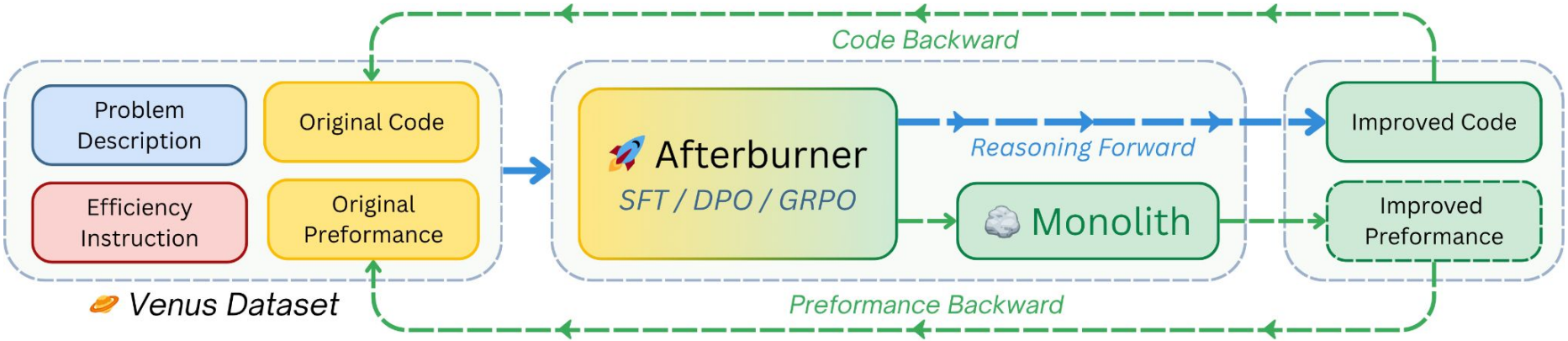
Runtime
121 ms 🐇

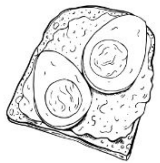
Functional Correctness: **Passed** ✅
Computational Efficiency: **Fast** 🙂

2 How to Improve Code Efficiency?



3 Iterative Optimization Framework





Original Food

Chef

Critic

Improved Food

4 Experiment Results

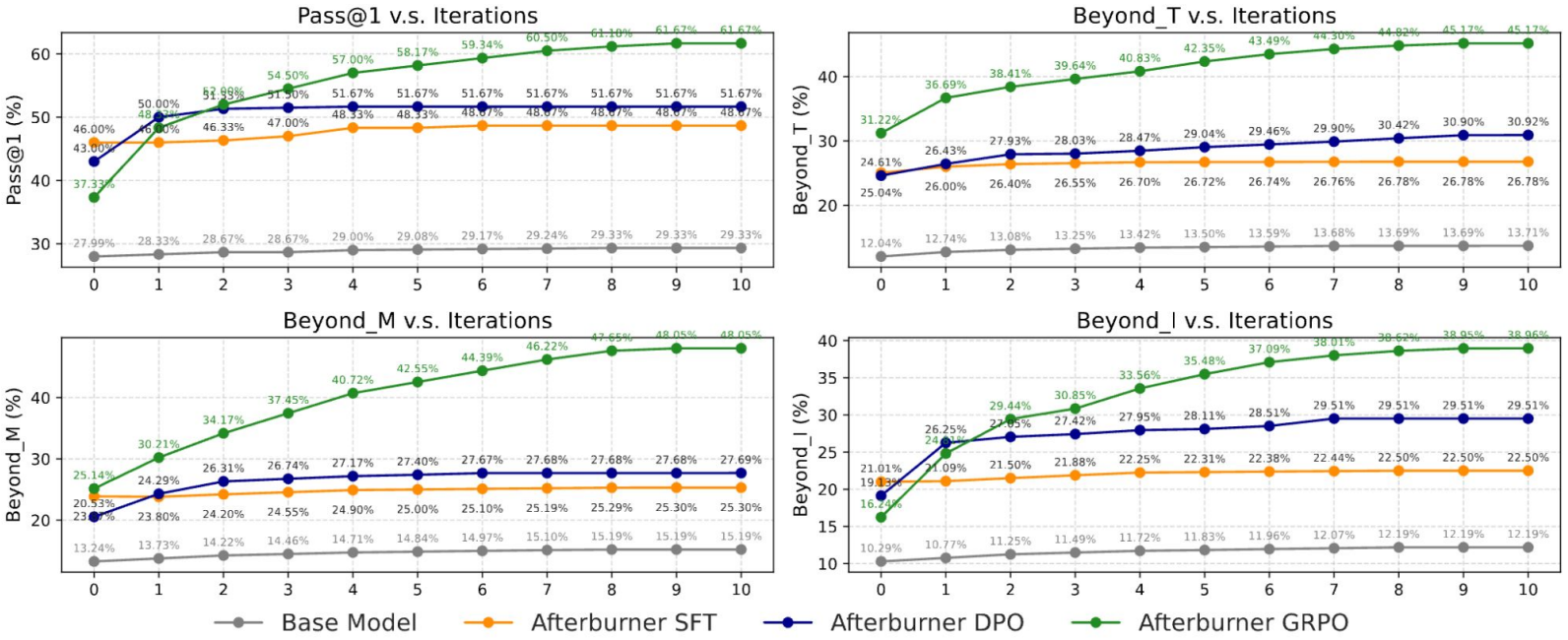


Figure 5: Iterative Optimization with an Efficient Instruction *'both time and memory efficient'*.

Thank you!