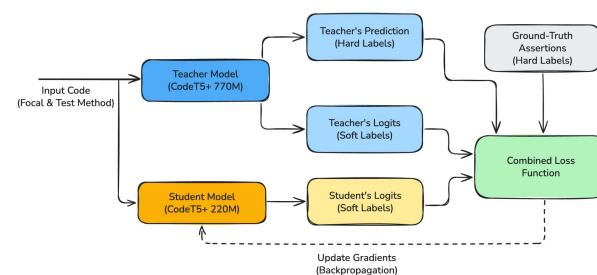


# Tiny but Mighty: Distilling Large Language Models for Efficient Test Assertion Generation

Author  
Di Wu - d.wu-7@student.tudelft.nl

Supervisor & Responsible professor  
Mitchell Olsthoorn, Annibale Panichella



## Motivation

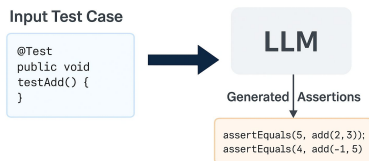
## Research Questions

## Process

Assertion generation is critical for reliable software testing. Large LLMs like CodeT5+ are effective but too heavy for local use:

- Slow inference
- High memory usage
- Unsuitable for real-time IDE integration

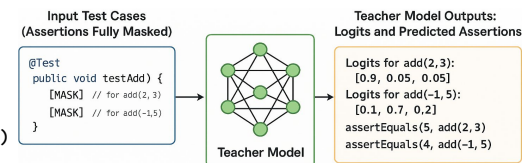
Goal: Build smaller, faster models that still generate high-quality assertions.



We aim to answer:

- How does student model size affect assertion quality?
- What are the trade-offs between quality, speed, and memory for local deployment?
- How do different distillation loss combinations affect performance?

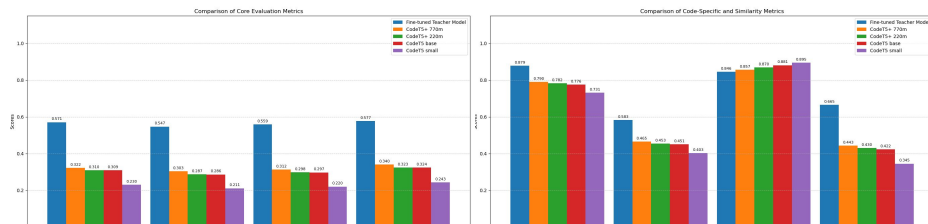
1. Fine-tune CodeT5+ on Java test assertions (teacher model).
2. Generate logits + assertions from teacher.
3. Train smaller student models using:
  - Teacher soft logits
  - Teacher predictions (hard labels)
  - Ground truth labels
4. Evaluate students across multiple quality + efficiency metrics.



## Results

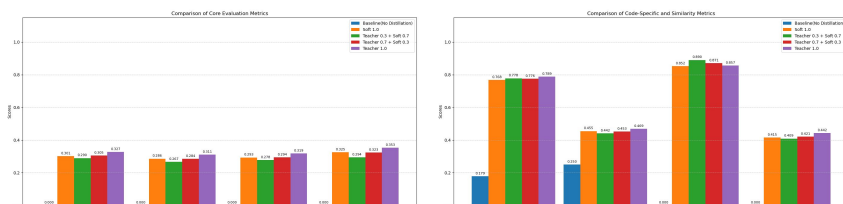
## Conclusions & Limitations

### Evaluation Metrics Across Student Model Variants



The CodeT5+ 220M student retained **~97%** of the CodeBLEU score of a much larger 770M CodeT5+ student (0.453 vs. 0.465), both being distilled using the same teacher and strategy. Interestingly, smaller student models achieved **higher AST validity**, suggesting they generate simpler, more syntactically safe code.

### Effect of Loss Weighting on Student Performance



### Inference Speed and Memory Usage

| Model           | Time / Method (ms) | Time / Assertion (ms) | Memory Usage (MB) |
|-----------------|--------------------|-----------------------|-------------------|
| Teacher (770M)  | 8216.2             | 5310.1                | 4418.89           |
| Student (220M)  | 2795.9             | 1835.1                | 2604.33           |
| Student (Base)  | 2965.9             | 1948.7                | 2658.30           |
| Student (Small) | 1225.4             | 813.4                 | 2402.34           |

The CodeT5+220M student model was nearly 3× faster per method and used ~41% less memory than the CodeT5+770M teacher. The smallest student model achieved a 6.7× speedup per method and reduced memory usage by ~46%.

Distillation was essential for learning (baseline F1 = 0.0). Surprisingly, using only the teacher's hard predictions ("Teacher 1.0") performed best, likely because the dataset is simple and the small model benefits more from clear, direct supervision than soft logits.

Conclusions:

1. Distilled a 220M model from CodeT5+ for local test assertion generation
2. Achieved ~78% CodeBLEU of the teacher at 3× faster inference, 40% less memory
3. Off-the-shelf models failed → distillation was crucial
4. Teacher hard labels alone gave best results
5. Smaller models produced more syntactically valid code

Limitations:

- Only tested on Java assertions
- Used 1 teacher model (CodeT5+)
- Stored only top-4 logits for KD (info loss)
- Did not evaluate runtime correctness (e.g., test execution)