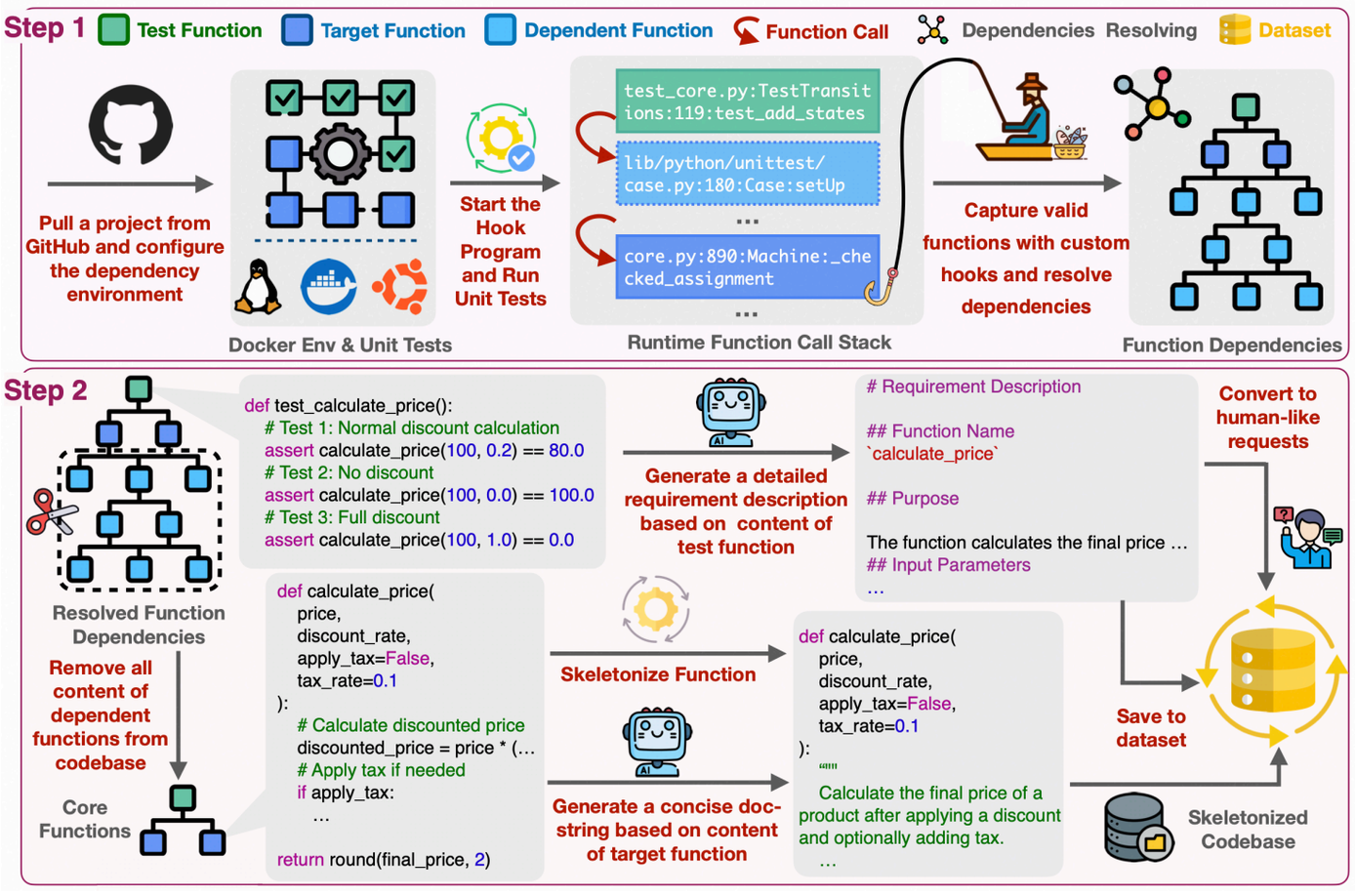
SWE-Flow: Synthesizing Software Engineering Data in a Test-Driven Manner

Lei Zhang, Jiaxi Yang, Min Yang, Jian Yang, Mouxiang Chen, Jiajun Zhang, Zeyu Cui, Binyuan Hui, Junyang Lin

Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences; University of the Chinese Academy of Sciences; Alibaba Qwen; Zhejiang University; University of Science and Technology of China

SWE-Flow: Data Synthesizing Framework



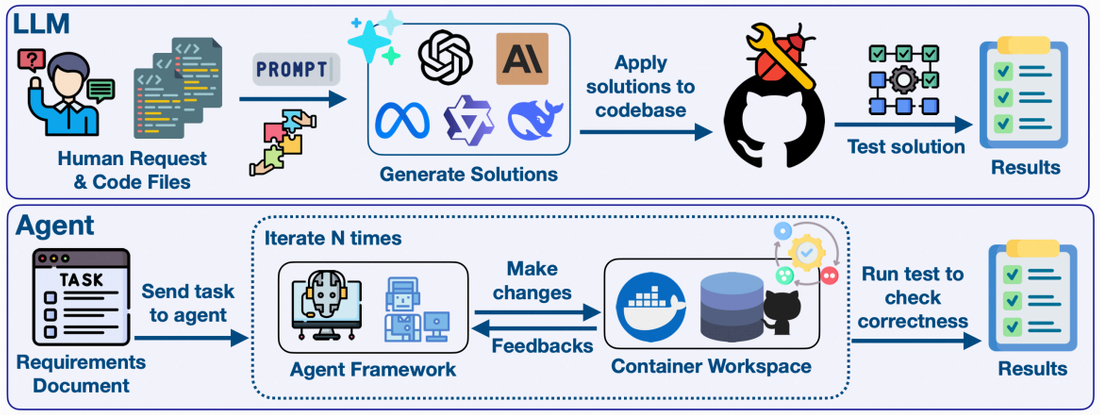
Step 1: Automated Setup and Scheduling

Given a codebase and its corresponding development environment, SWE-Flow executes unit tests, constructs the project's Runtime Dependency Graph (RDG), and generates a development schedule.

Step 2: Test-Guided Code Removal & Doc Generation

Based on the development schedule, SWE-Flow removes the implementation of core functions covered by the current step's test functions, forming an incomplete codebase for development. Additionally, it generates a development document based on the content of the test functions.

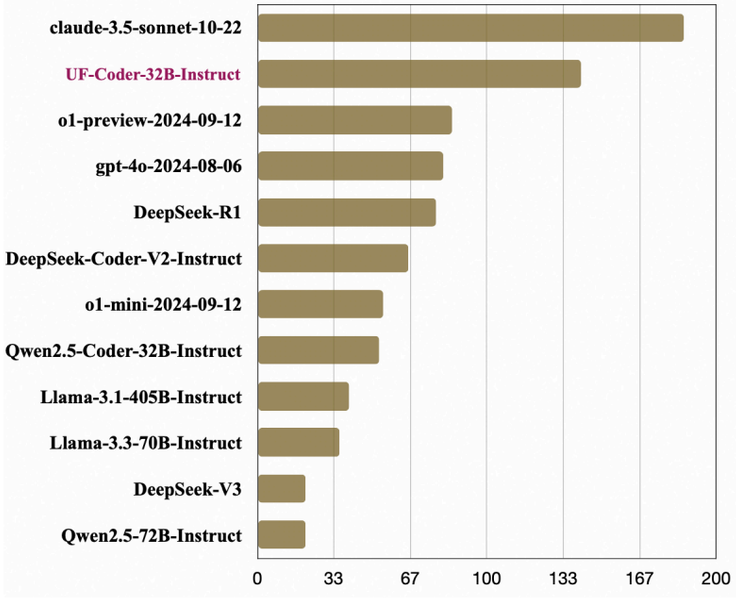
SWE-Flow-Bench: Evaluation Framework



Upper: A one-shot approach where an LLM receives the codebase, dev document, and output format in a single prompt. A post-processor applies the generated solution and runs unit tests to verify correctness.

Lower: An agent incrementally modifies an incomplete codebase based on a dev document. It develops step by step until completion or reaching an iteration limit, followed by unit test evaluation.

Experiment Results



Improvement: Finetuning Qwen-2.5-Coder-32B with data synthesized by SWE-Flow significantly improved its performance on test-driven development tasks, ranking just behind claude-3.5-sonnet.