FuzzAug: Data Augmentation by Coverage-guided Fuzzing for Neural Test Generation

Yifeng He, Jicheng Wang, Yuyang Rong, Hao Chen
University of California, Davis

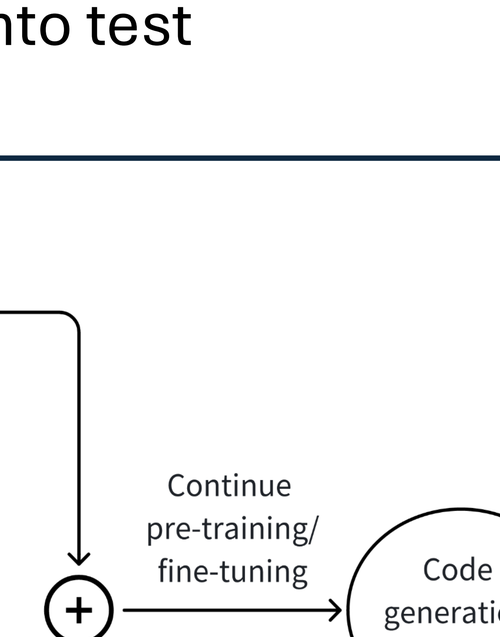
30th ANNIVERSARY EMNLP 2025
Suzhou, China | 中国苏州
November 4-9 | 11月4日–9日

Background: LLM-based unit test generation

1. Train an LLM with a massive code corpus
2. Prompt the LLM with function under test (focal function).
3. LLM generates unit test functions

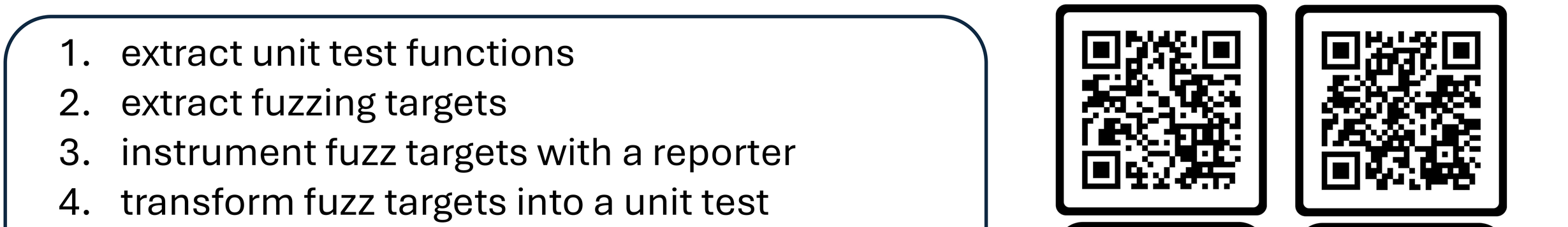
Challenge: limited training data

1. Testing code is 20% of all.
2. Focal-test pairs are even harder to find.



Our approach:

1. Use coverage-guided fuzzing to gather input data that are interesting to the focal function.
2. Use program transformation to convert existing fuzz targets into test templates with valid testing semantics.



1. extract unit test functions
2. extract fuzzing targets
3. instrument fuzz targets with a reporter
4. transform fuzz targets into a unit test
5. instantiate the templates with valid test inputs

Paper

Code on GitHub

Model	Type	Assert. CR	Acc	Model	Type	Func. CR	Cov
StarCoder2	PT	64.09	31.83	StarCoder2	PT	45.73	9.88
UnitCoder	FT	65.73	32.99	UnitCoder	FT	48.17	11.92
FuzzCoder	FT	70.98	35.50	FuzzCoder	FT	59.56	17.09
CodeLlama	IT	64.57	32.13	CodeLlama	IT	54.88	15.75
UnitLlama	FT	70.79	34.70	UnitLlama	FT	64.02	16.23
FuzzLlama	FT	75.67	37.07	FuzzLlama	FT	71.95	19.52
CodeQwen1.5	PT	66.52	41.71	CodeQwen1.5	PT	68.29	20.90
UnitQwen	FT	73.54	46.04	UnitQwen	FT	60.37	20.76
FuzzQwen	FT	80.91	52.20	FuzzQwen	FT	73.17	24.63

Models trained with FuzzAug outperforms the pre-trained baseline and models trained w/o it by large margin!

CR: compile rate
Acc: accuracy
Cov: branch coverage