One Fuzz Doesn't Fit All: Optimizing Directed Fuzzing via Program State Restriction

Prashast Srivastava[1], Stefan Nagy[2], Matthew Hicks[3],
Antonio Bianchi[1], Mathias Payer[4]

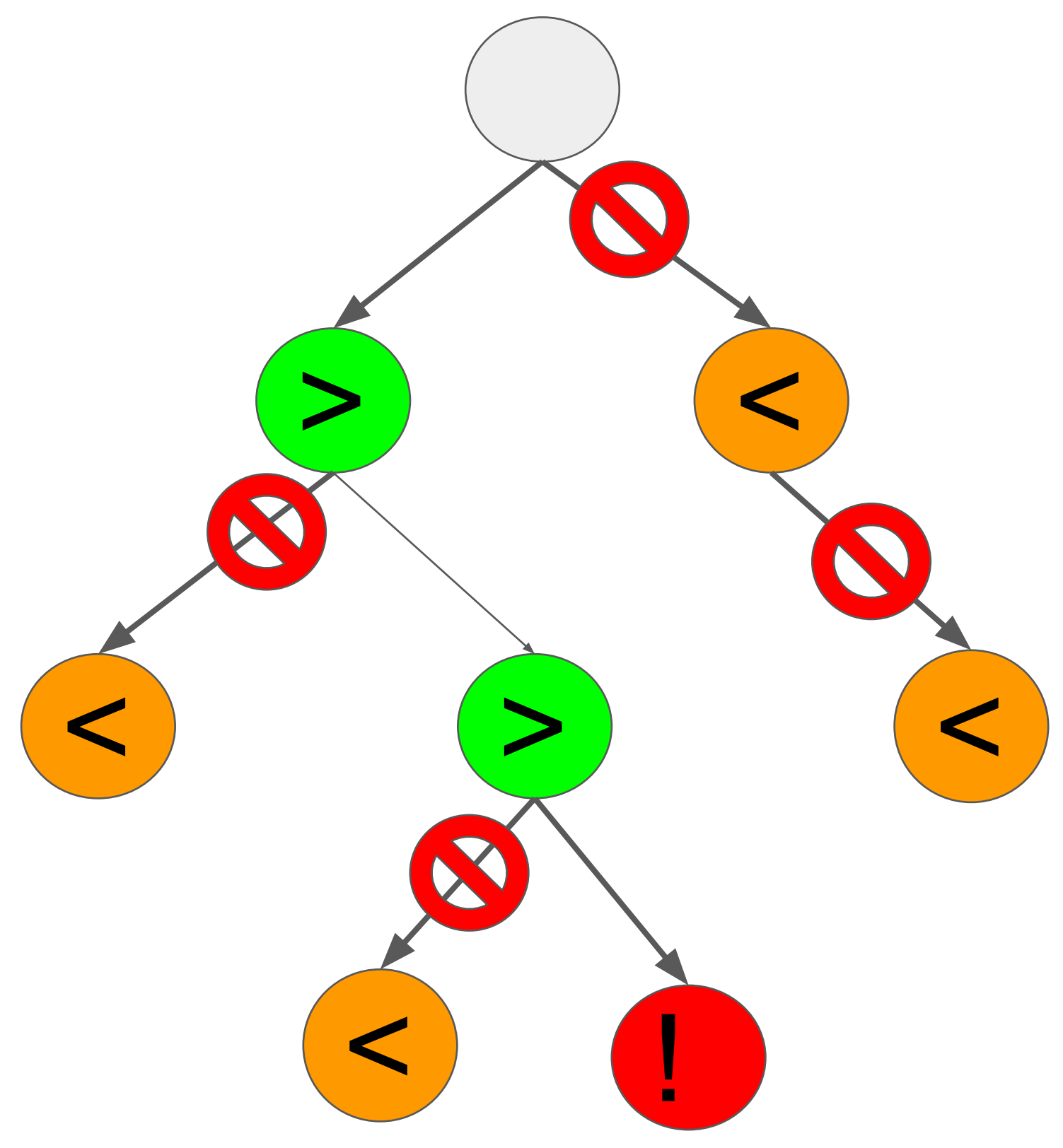
[1] Purdue University [2] University of Utah [3] Virginia Tech [4] EPFL

Inefficiency of existing directed fuzzing approaches

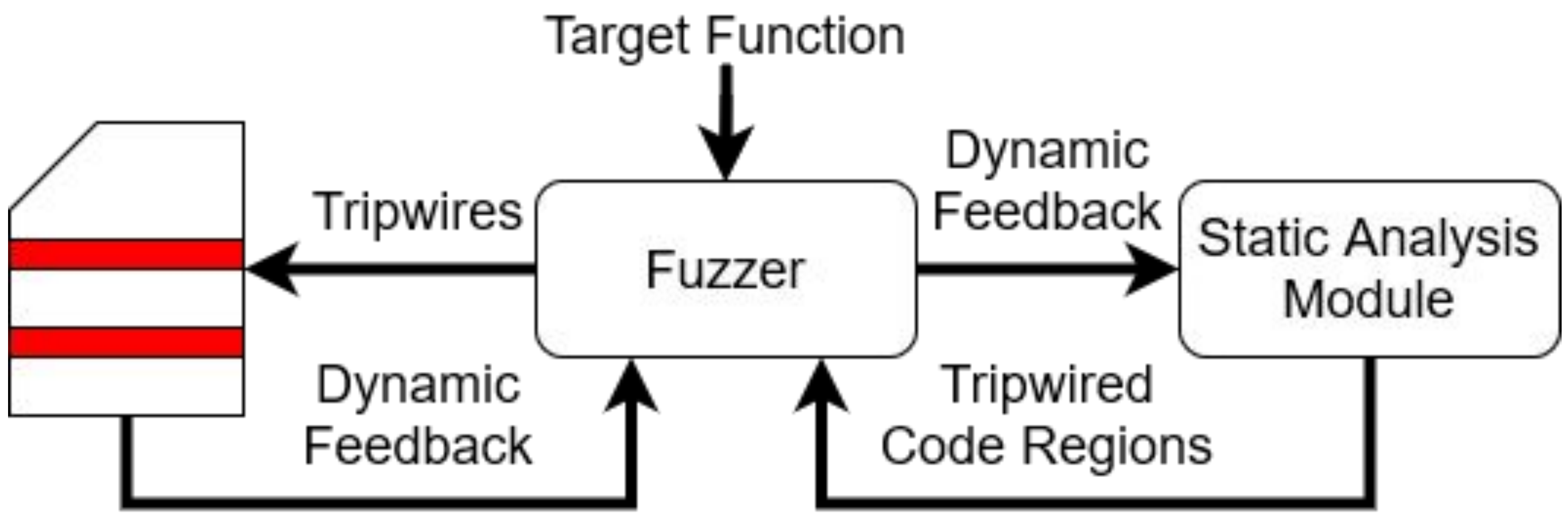
Problem: Existing distance minimization based fuzzers perform wasteful exploration of target-unreachable code regions

Observation: Current exploration schemes particularly ill-suited for *disjoint* target locations

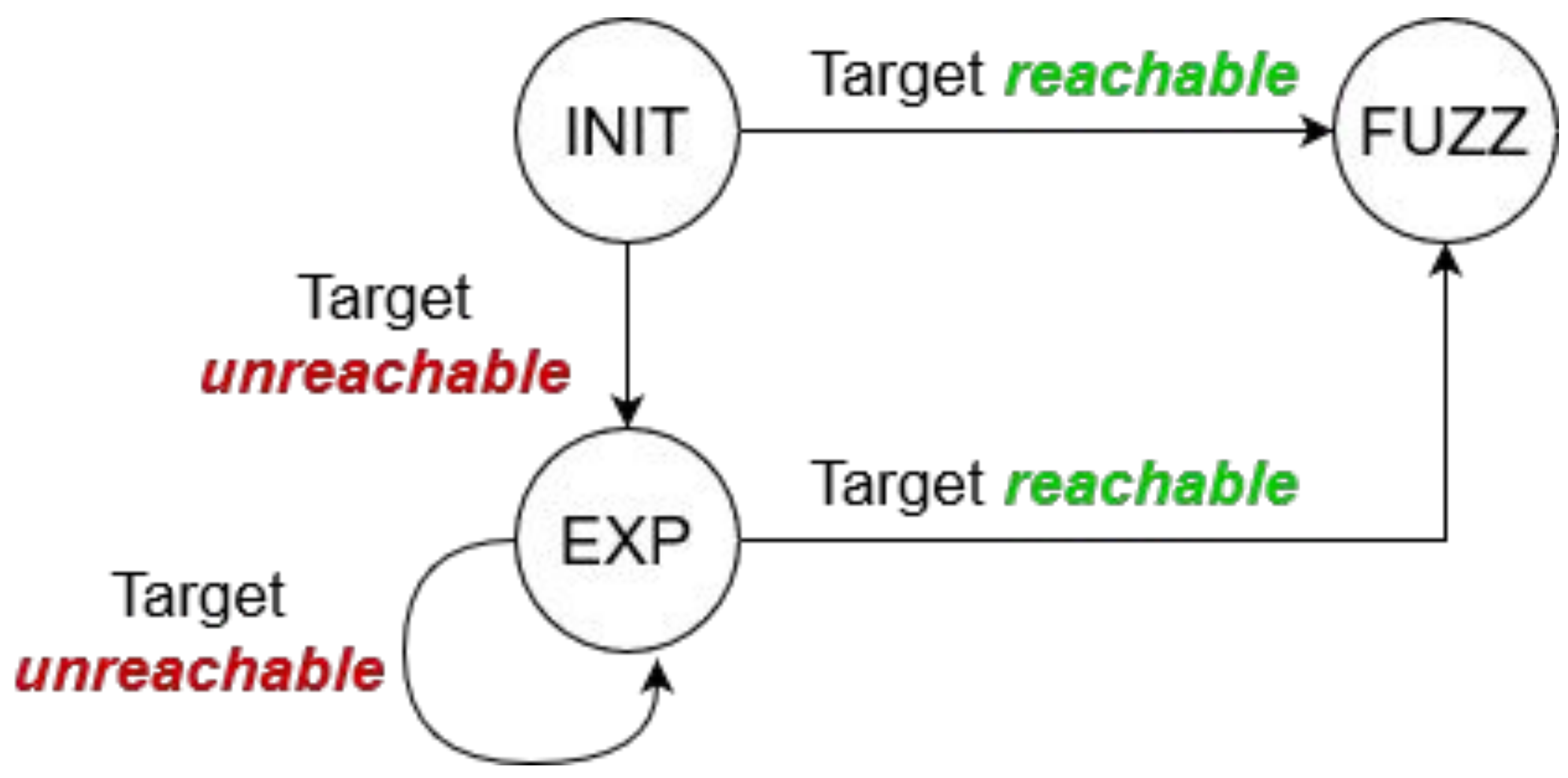
Solution: Preemptively terminate execution of target-unreachable code regions — Tripwiring



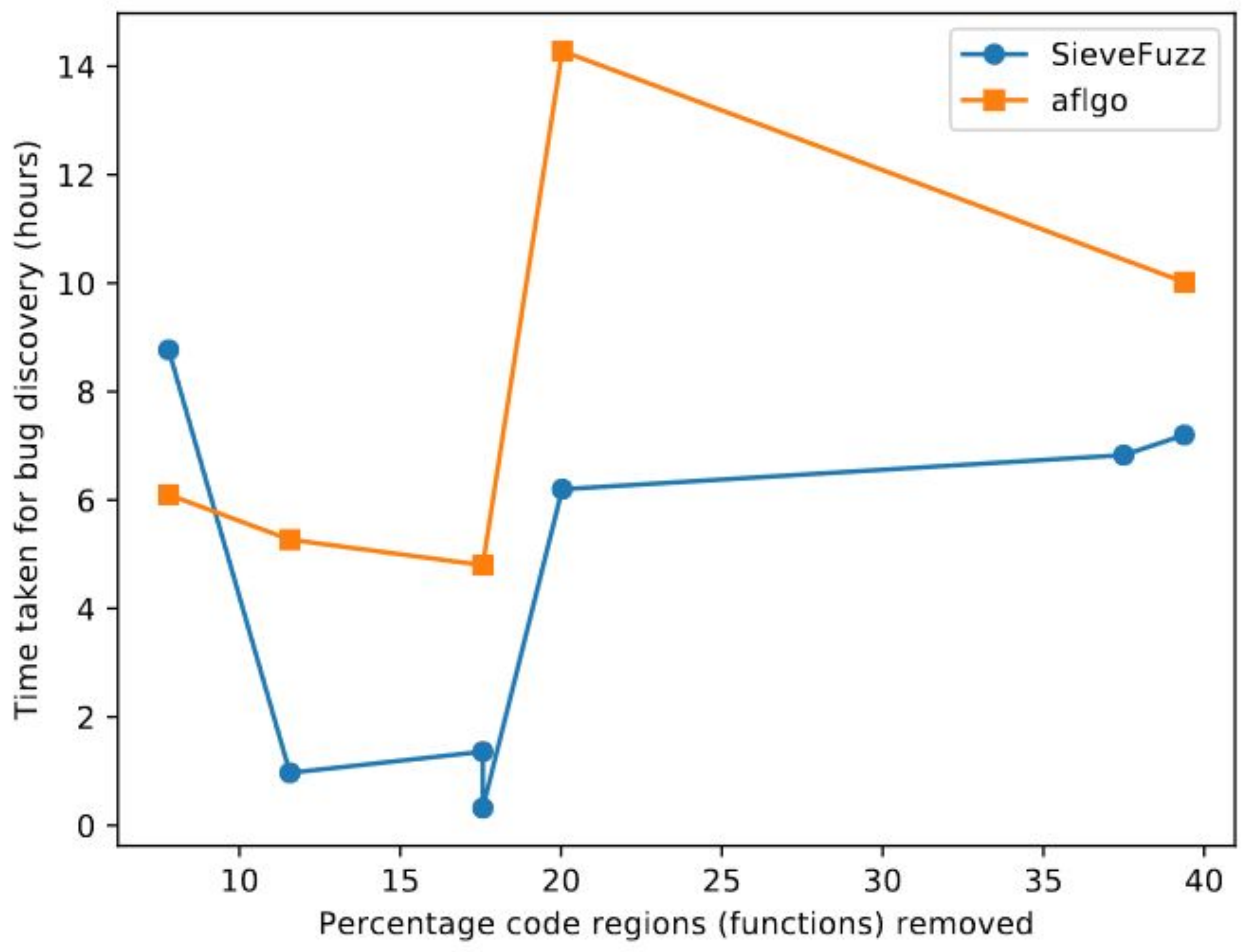
Tripwiring-directed Fuzzing



SieveFuzz Workflow



Bug Discovery Performance



Takeaway

Tripwiring is an optimal strategy for fuzzing target locations which exhibit disjointness

SieveFuzz can trigger bugs on average **47% more consistently** and **117% faster** than existing state-of-the art undirected (AFL++) and directed fuzzers (AFLGo, BEACON)

erc
European Research Council
Established by the European Commission