CLNX Bridging Code and Natural Language for C/C++ Vulnerability-Contributing Commits Identification

Zeqing Qin, Yiwei Wu, Lansheng Han*

School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan, China

WHAT ARE Vulnerability-Contributing Commits (VCCs)?

Vulnerability-Contributing Commits (VCCs) are patch commits in open-source software development that unintentionally introduce security vulnerabilities into the codebase. These commits modify source code (e.g., configuration, logic) but contain flaws classified under Common Weakness Enumeration (CWE), regardless of exploit conditions.

For example, a VCC might alter file permissions in a way that enables privilege bypass. VCCs are critical for vulnerability analysis, as they reveal how and where security risks originate during code evolution.

WHY CLNX IS NEEDED?

BERT-based LLMs show promise in vulnerability detection by learning code context and dependencies. However, they struggle with C/C++—the source of 52% of open-source vulnerabilities. This gap stems from insufficient C/C++-specific pre-training and inherent challenges like **complex syntax**, **low-level operations**, and **code redundancy**. While current solutions rely on further pre-training (e.g., CodeBERT-cpp), they yield minimal gains (+2.03% accuracy) at high computational costs, proving inefficient and limited.

CLNX aims to enhance the effectiveness of LLMs in identifying C/C++ VCCs while ensuring a lightweight implementation.

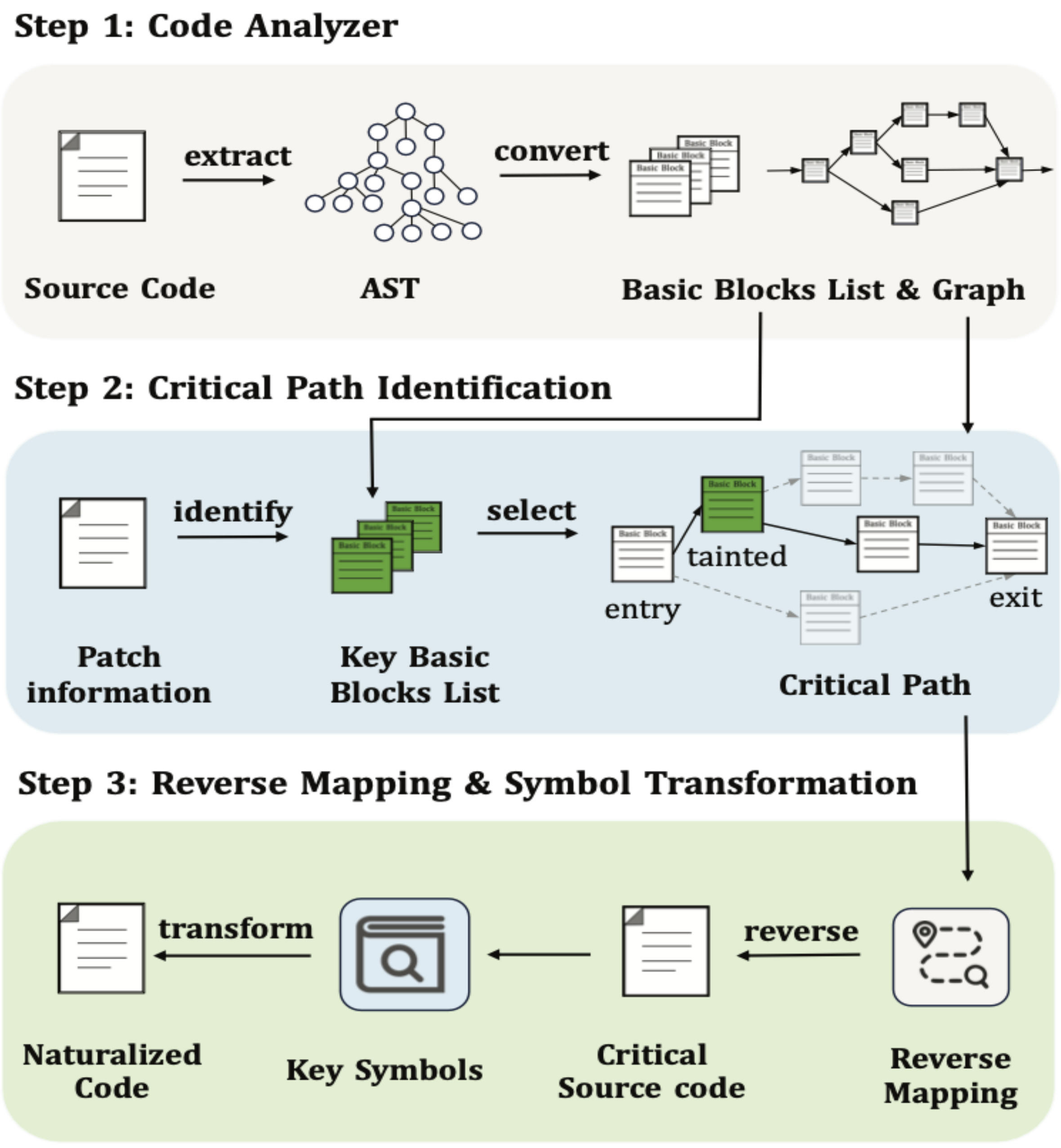
HOW CLNX WORKS?

CLNX simplifies C/C++ code analysis for LLMs through two-step naturalization, bypassing heavy pre-training:

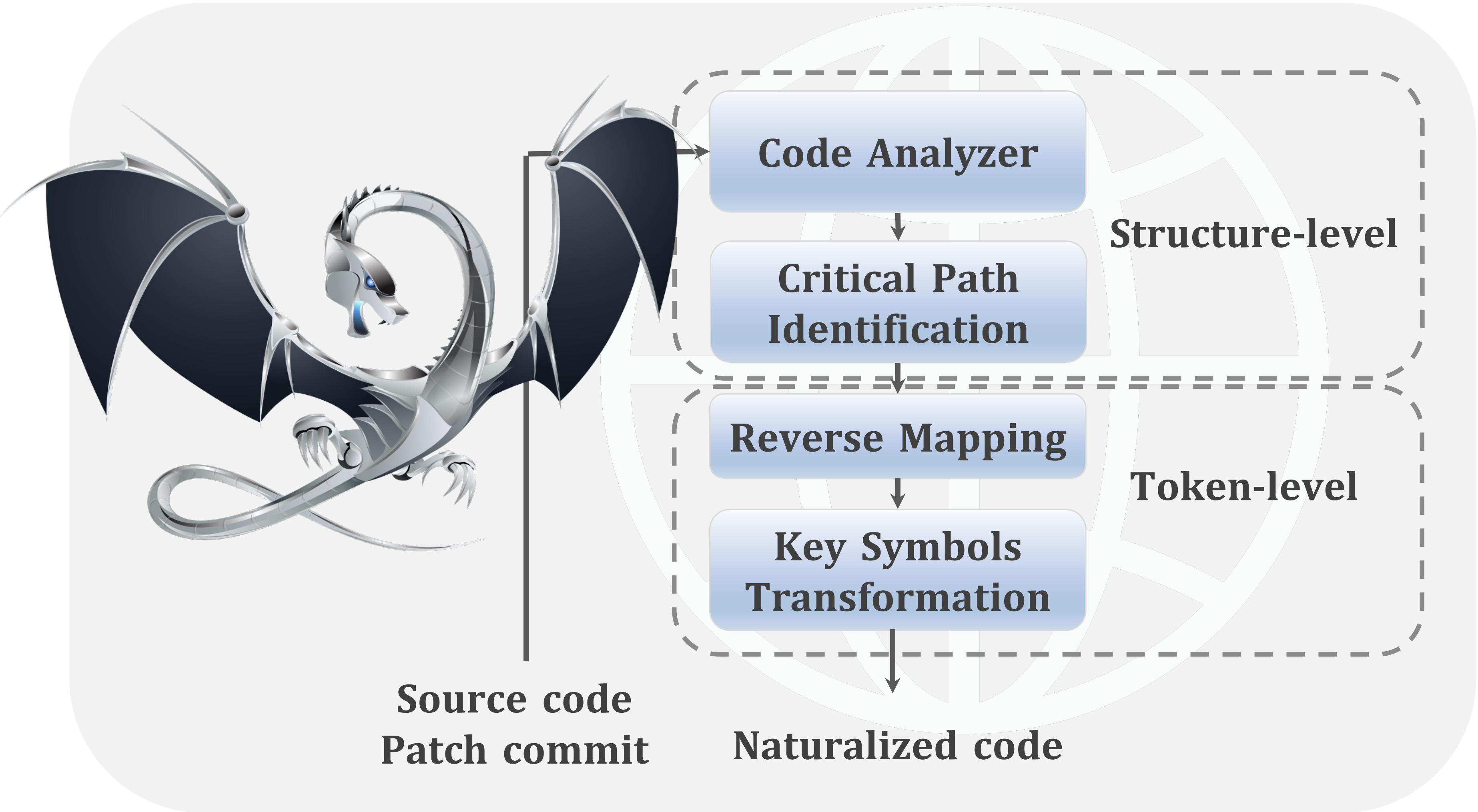
1. **Structural Naturalization:**
Transforms code into execution flow graphs and pinpoints critical paths using patch data.

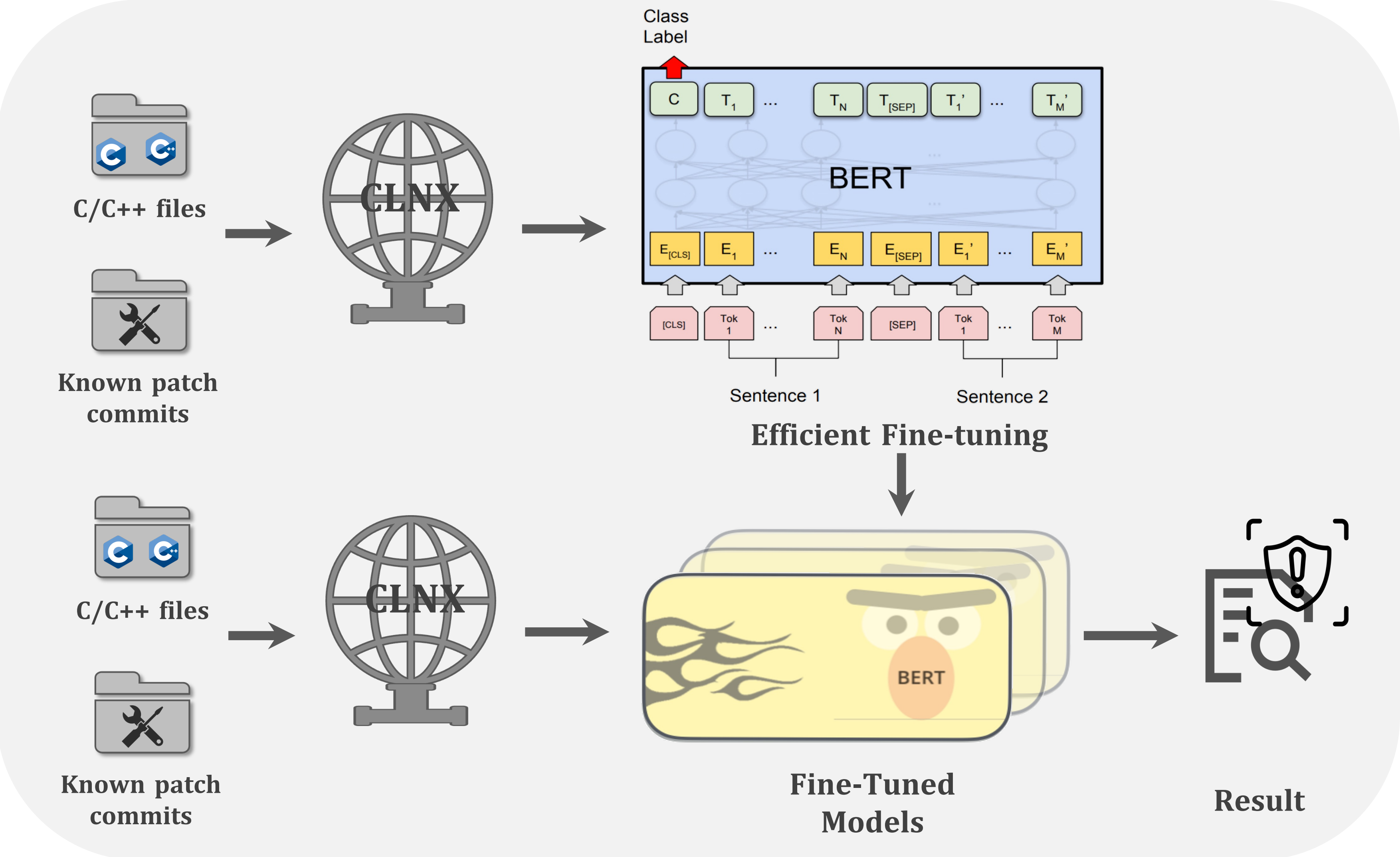
2. **Token Naturalization:**
Converts critical symbols (e.g., variables, functions) into plain-language terms (e.g., "user_input")

3. **LLM-Friendly Output:**
Delivers simplified, naturalized code for LLMs to detect C/C++ VCCs efficiently.



CLNX ARCHITECTURE



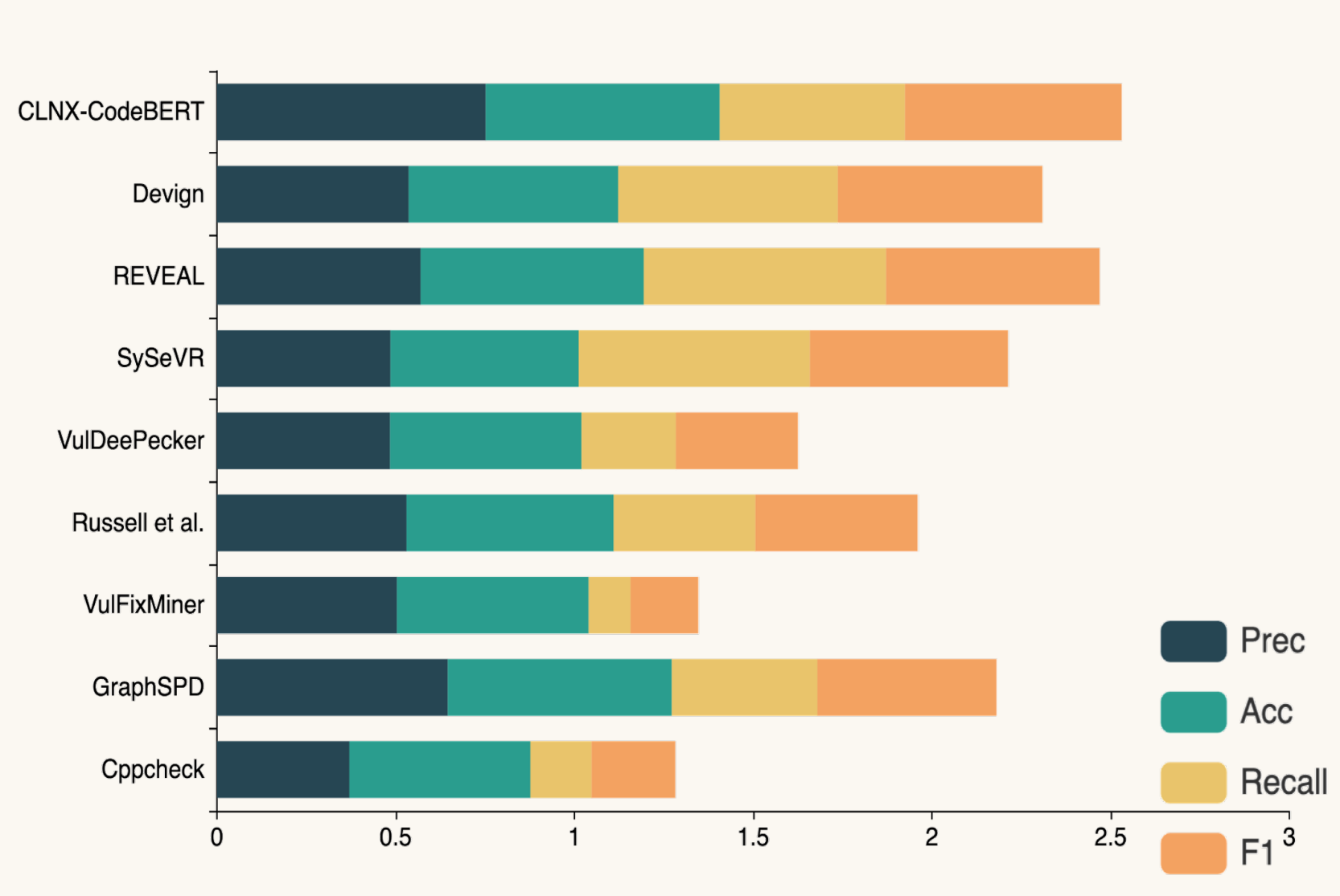


OVERVIEW

CLNX is a middleware designed to make C/C++ programs compatible with LLMs (especially BERT-based), thereby improving their ability to identify C/C++ VCCs. CLNX operates with minimal resource overhead, offering efficiency advantages over pre-training methods.

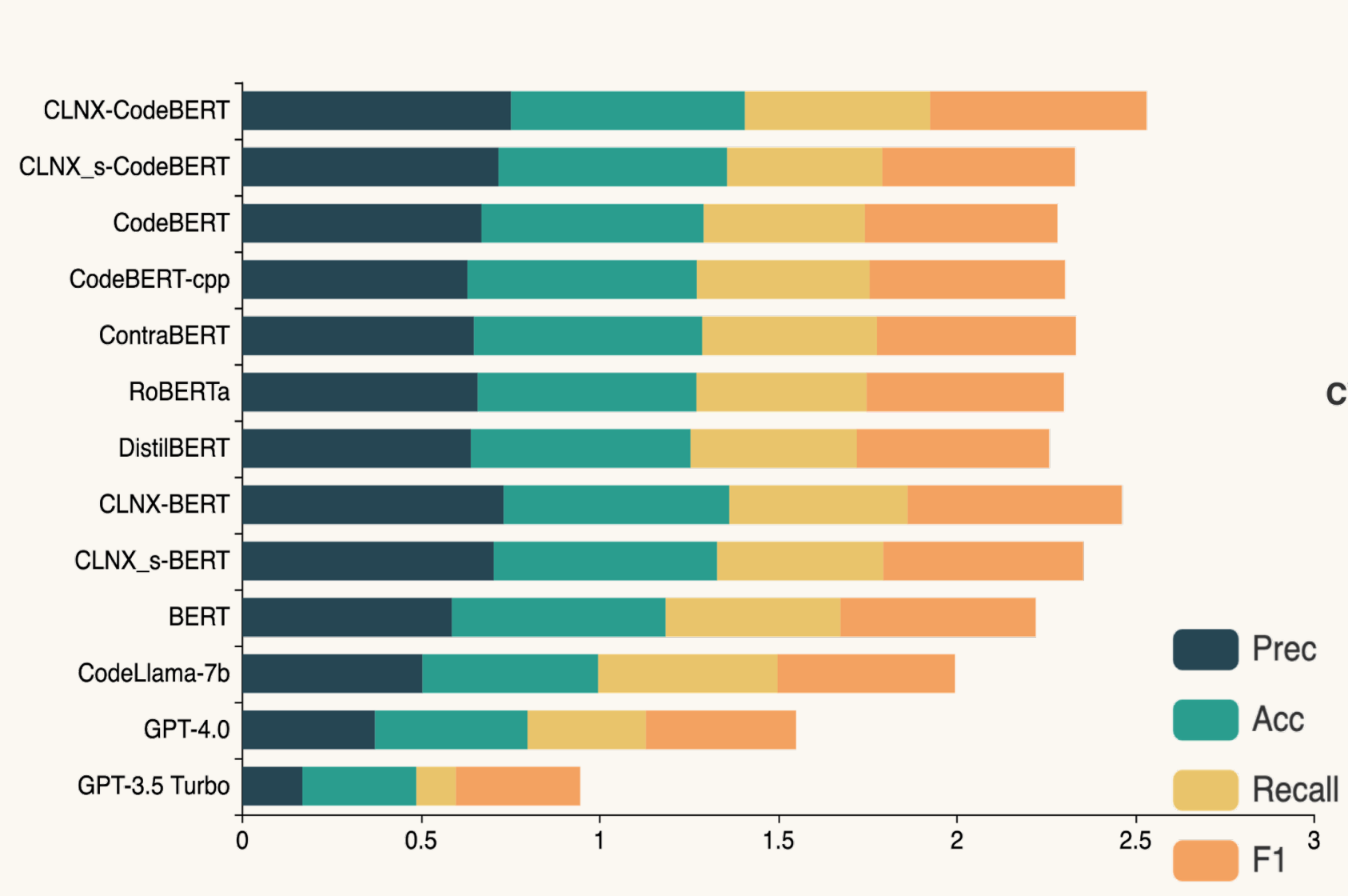
EXPERIMENTS

Results of comparative analysis

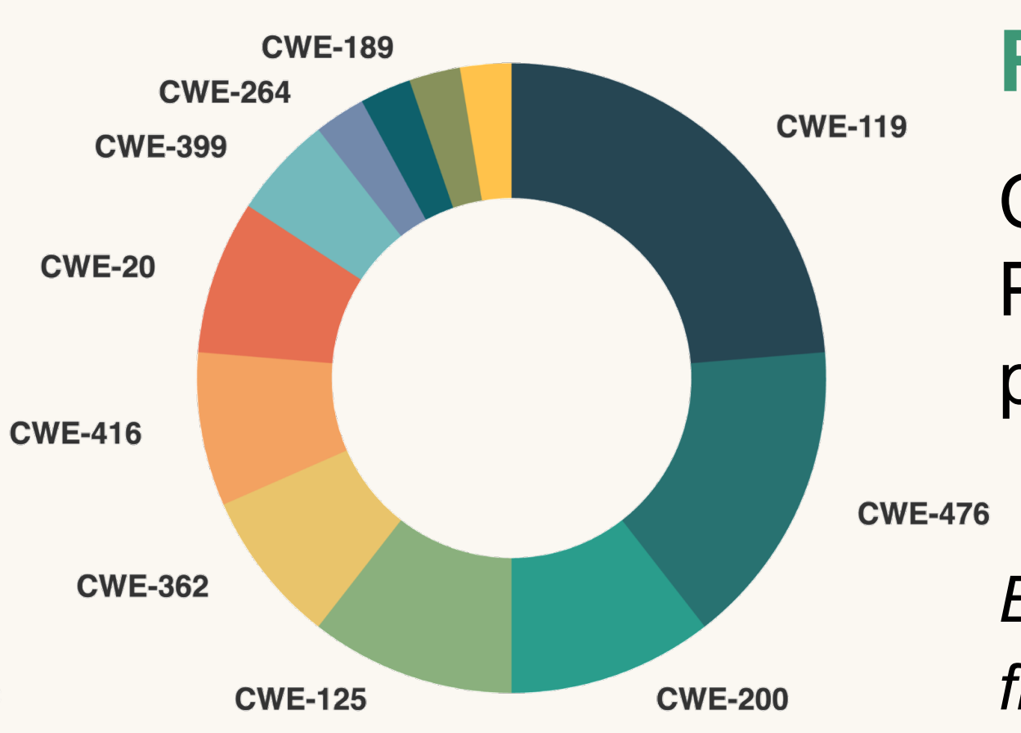


Items of Study
Effectiveness
Comparison
Read-World Application

Results of LLMs in C/C++ VCCs identification



Statistical Distribution of CWE Types



EFFECTIVENESS: HOW DOES CLNX ENHANCE LLMs FOR C/C++ VCCs IDENTIFICATION TASK?

Both the Structure-level and Token-level Naturalization phases play crucial roles in CLNX's effectiveness. CLNX boosts LLMs' performance across all metrics (Precision: BERT: +14.48%, CodeBERT: +8.27%). CLNX outperforms resource-heavy pre-training methods. Despite minor recall dips in the early stages, full CLNX integration achieves the highest recall by retaining critical vulnerability patterns.

COMPARISON: HOW DOES THE PERFORMANCE OF CLNX-LLMs COMPARE TO OTHER METHODS?

CLNX-CodeBERT surpasses graph-based models and traditional tools with 10.59% higher precision and the best F1 score, proving its lightweight code simplification outperforms complex embeddings. While recall lags slightly, its balanced efficiency and accuracy make it the top choice for C/C++ VCCs identification.

REAL WORLD: HOW DOES CLNX-LLM PERFORMS IN IDENTIFY REAL-WORLD VCCs?

CLNX-CodeBERT detected 38 verified vulnerabilities across 35 C/C++ projects, excelling at identifying common patterns like Null Pointer Dereference (6) and Buffer Errors (9) by stripping code noise. While effective for logic-driven flaws, it struggles with complex, patternless vulnerabilities (e.g., Cryptographic Issues) due to inconsistent code logic.

Experimantal Settings: Built on the public Devign dataset, our evaluation uses 25,872 real-world C/C++ function pairs (vulnerable/non-vulnerable) from FFmpeg and Qemu, split into 80% training, 10% validation, and 10% testing to ensure robust model assessment. For the real world one, we deploy the finetuned CLNX-CodeBERT to scan the repositories of 35 C/C++ open-source projects. Notably, VCCs are relatively rare in open-source software.

Made by: Zeqing Qin (Edward), Yiwei Wu

Email: Zeqing@hust.edu.cn Personal Website: https://edwardqin-creator.github.io/

Graduate Seeking LLM-Related Opportunities: Open to Diverse Roles