Yunxiang GE, Bing XIA, Yong DING, Wenbo LIU, 2026. Benchmarking large language models for binary function name prediction. *ENGINEERING Information Technology & Electronic Engineering*, 27(8):260017.
https://doi.org/10.1631/ENG.ITEE.2026.0017

Benchmarking large language models for binary function name prediction

Key words: Binary function name prediction; Large language models (LLMs); Reverse engineering (BE); Binary code analysis

Corresponding author: Bing XIA
E-mail: xiabing@zut.edu.cn
ORCID: https://orcid.org/0000-0001-9467-6092

Motivation

Stripped binaries lose the semantic cues that LLMs normally rely on.

Source code		**Compiled binary**		**Stripped binary**		**Decompiled C**
names + types	→	low-level form	→	symbols removed	→	noisy + sparse

- ➢Function names and symbols are removed.
- ➢Compiler optimization and instruction set architecture (ISA) differences distort the program structure.
- ➢Decompiled pseudocode is often semantically sparse and noisy.
- ➢Binary function name prediction is a representative semantic-recovery task.

Main idea

Instead of proposing another task-specific model, this paper builds a unified and realistic benchmark.

1. Realistic setting

symbol-stripped, non-obfuscated binaries across multiple projects, ISAs, and optimization levels

2. Systematic evaluation

compare general-purpose and code-oriented open-source LLMs under a unified inference protocol

3. Context analysis

separate realistically recoverable binary cues from oracle symbolic information

Key question: What are the capability boundaries of current LLMs for binary semantic recovery?

Evaluation framework

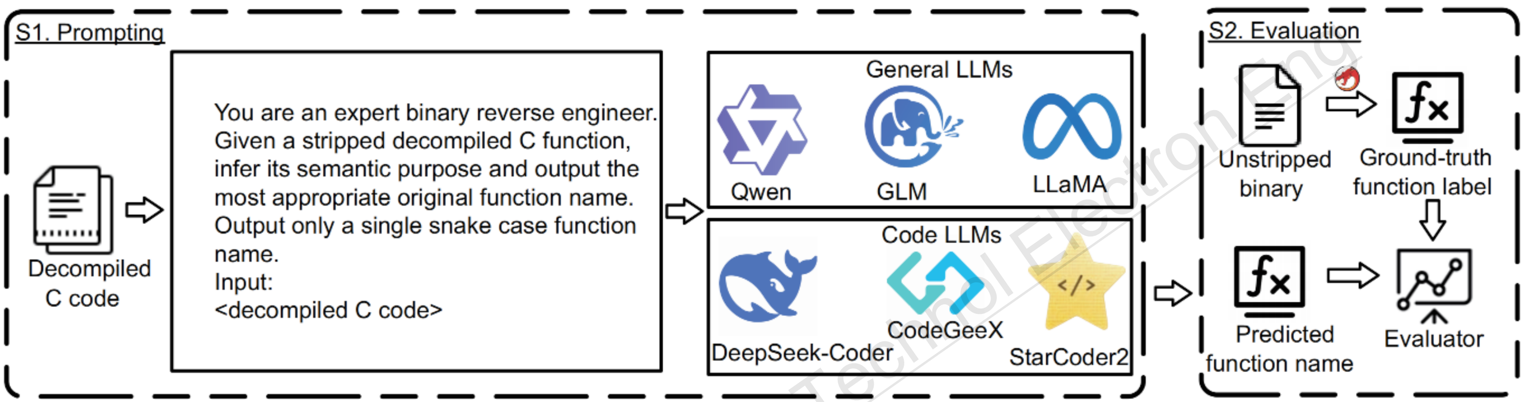

Fig. 1 Overview of the function name prediction workflow

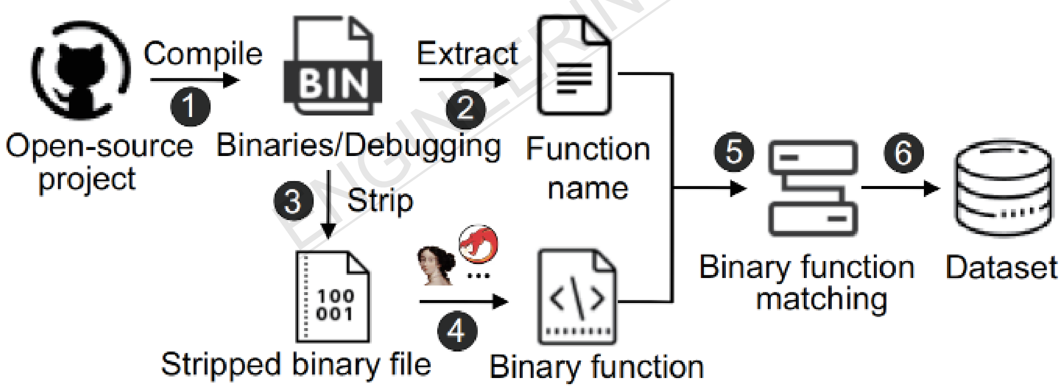

Fig. 2 Overview of the dataset construction pipeline

Dataset

49 935 functions from eight open-source projects
Benchmark subset: 1600 functions (200/project)

ISAs

x86 / x64 / ARM / MIPS

Setup

Zero-shot; local open-source LLMs; F1 evaluation

Major results

(model domain and scale)

Code-oriented pretraining matters more than the parameter count.

17.03%

F1 of Qwen2.5-Coder:7B under stripped binaries

Stripped-setting F1

Qwen2.5-Coder:7B	**17.03%**
LLaMA-3.1:70B	16.75%
Qwen2.5-Coder:14B	14.27%
Qwen2.5-Coder:32B	15.10%
Qwen2.5:72B	15.39%

- Code-oriented LLMs consistently outperform general-purpose LLMs.
- Larger models do not produce monotonic gains.
- The main limitation is cross-domain semantic mismatch, not model capacity alone.

Major results

(input length)

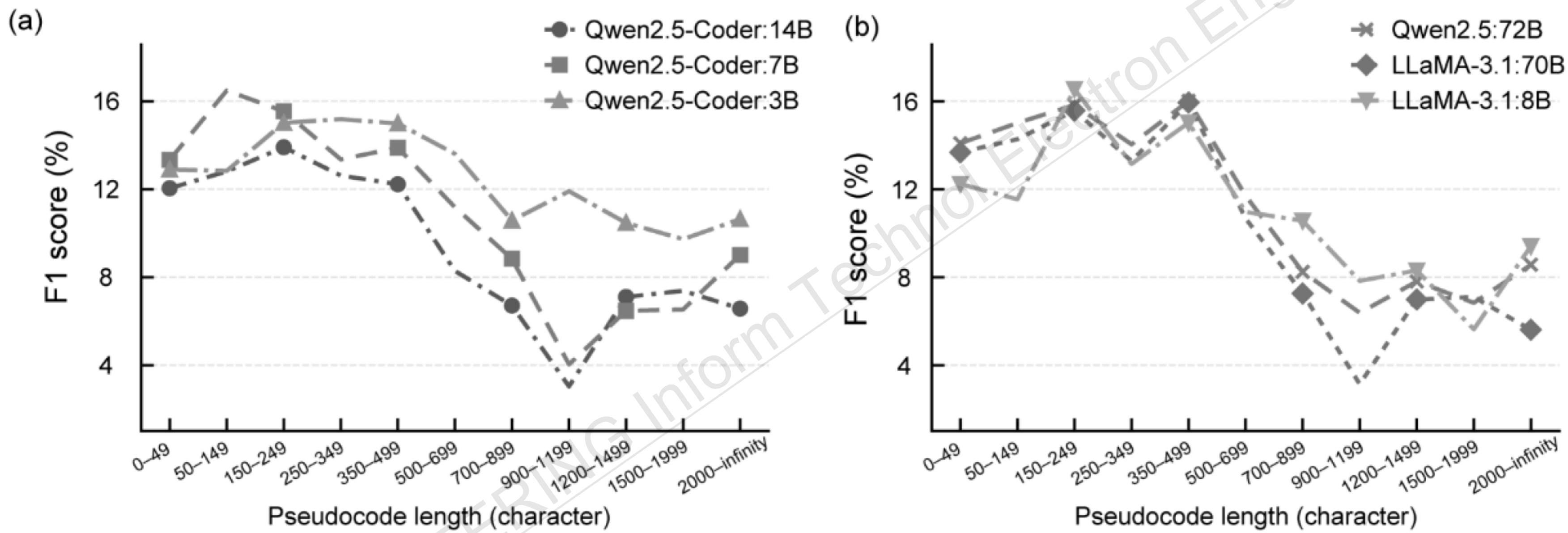

Fig. 3 Impact of pseudocode length on function name prediction performance: (a) code-oriented LLMs; (b) general-purpose LLMs

- **Longer pseudocode generally degrades performance.**
- **Shorter inputs often preserve the most information-rich local cues.**

Major results

(ISA and optimization)

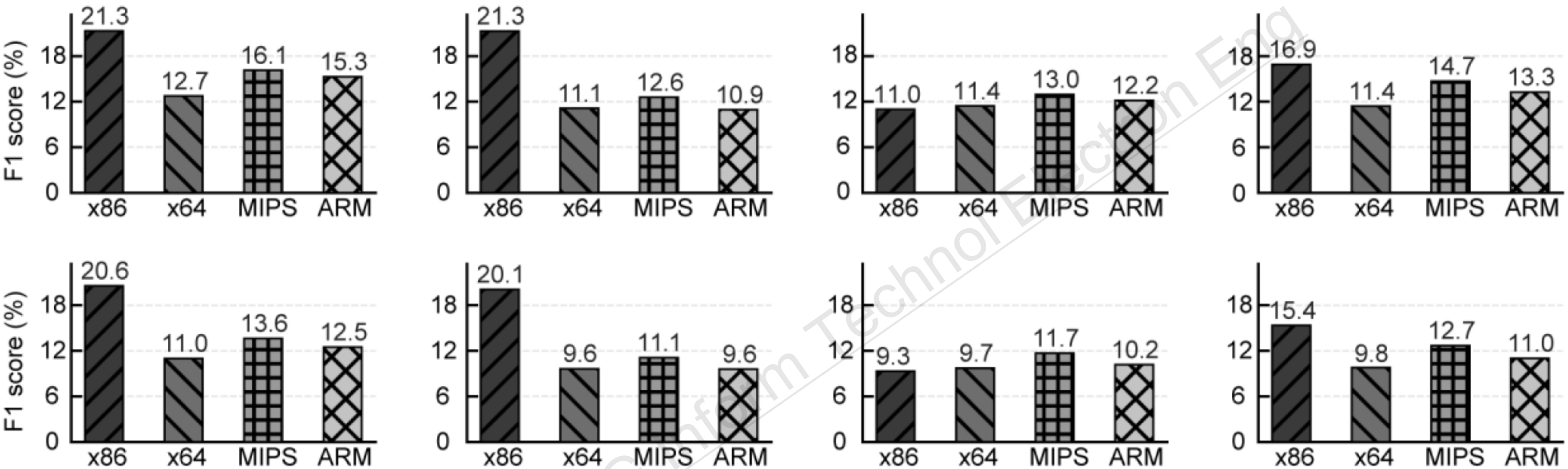

Fig. 4 F1 score comparison of different models across instruction set architectures and compiler optimization levels. The first row corresponds to Qwen2.5-Coder:7B and the second row corresponds to LLaMA-3.1:8B. From left to right, the four columns correspond to O0, O1, O2, and O3, respectively

- ➢ **Higher optimization levels generally degrade semantic recovery.**
- ➢ **Code-oriented models show stronger robustness across architectures and optimization settings.**

Major results

(context matters)

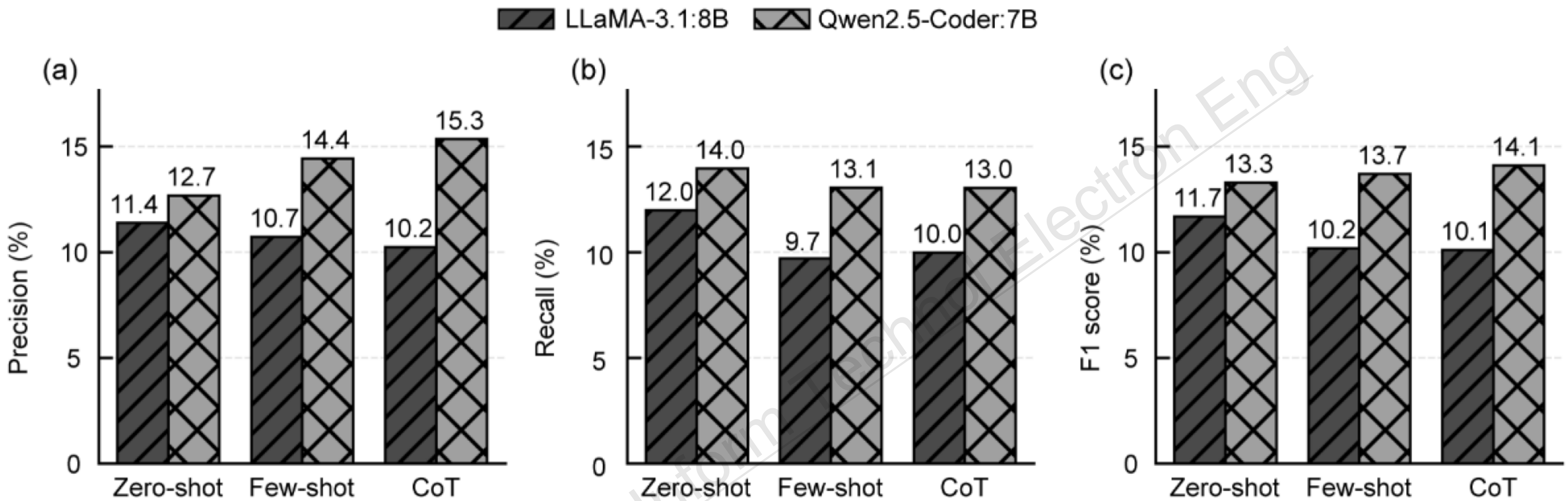

Fig. 5 Impact of different prompt designs on function name prediction performance: (a) Precision; (b) Recall; (c) F1 score

Qwen2.5-Coder:7B

Pseudocode	17.10%
+ API	18.40%
+ String	17.90%

+ Topology	19.70%
+ Realistic context	**20.80%**
+ Oracle context	**29.10%**

Prompt engineering provides only the modest gains; structurally recoverable cross-function context is much more effective.

Conclusions

- Code pretraining helps, but source-code semantics does not reliably transfer to stripped binaries.
- Scaling and prompt engineering alone cannot overcome semantic sparsity.
- Realistically recoverable context, especially the call-graph structure, offers a stronger path forward.
- Future binary LLM research should focus on contextual modeling and binary-specific representation learning.

Yunxiang GE received his M.S. degree from Zhongyuan University of Technology, Zhengzhou, China, in 2024. His research interests include large language models and binary reverse engineering.

Bing XIA is a distinguished member of CCF and an associate professor with the School of Cyberspace Security, Zhongyuan University of Technology, Zhengzhou, China. He received his Ph.D. degree from Information Engineering University, Zhengzhou, China. He serves as a deputy director of the Office of the CCF Technical Committee on Computer Applications. He has presided over multiple national and provincial or ministerial research projects, and supervised students receiving awards in national-level cybersecurity competitions. His research interests lie in agent-based cybersecurity and system security.