THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

NEURAL INFORMATION PROCESSING SYSTEMS

ProfiX: Improving Profile-Guided Optimization in Compilers with Graph Neural Networks

Huiri Tan[1], Juyong Jiang[1,2], Jiasi Shen[1]

[1]HKUST, [2]HKUST(GZ)

November 7, 2025

Introduction
Background

- Compiler is a system that translates high-level code (e.g. C/C++, Rust) to low-level machine code.
- Except naïve translation, modern compilers apply various optimizations to improve performance.
- **Profile-Guided Optimization (PGO)** uses runtime profiles to guide optimizations, such as function inlining, loop unrolling, optimizig code layout.

Table 1: High-level comparison of two approach to PGO.

Type	Efficiency	Non-Intrusive	Scalable	Accuracy
Instrumentation-based PGO	✗	✗	✗	✓
Sampling-based PGO	✓	✓	✓	✗

Introduction

Problem: Profile Inference

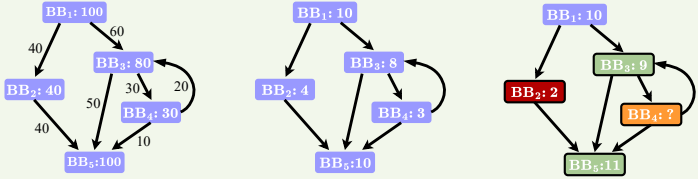

Figure 1: Motivating example of the inaccuracies in sampling-based profiles.

In the context of PGO, the profile represents the execution counts of basic blocks (BBs) and branches in control-flow graphs (CFGs).

Introduction
Our Goal

Reconstructing high-fidelity profiles from noisy / incomplete sampling-based profiles.

- **Input:** CFG $G = (V, E) \oplus$ program features $\oplus$ sampled counts $w(v)$.
- **Learn:** $F : G \to \mathbb{R}^{|V|}$ to predict BB execution frequencies.
- **Output:** High-fidelity profile with per-BB frequencies for PGO.

Methodology

The ProfiX Framework

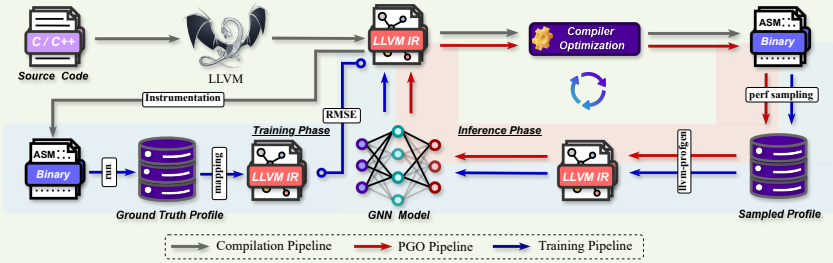

Figure 2: Overview of the ProfiX framework. The system operates in two phases.

- **Training phase:** collect sampled profiles (perf) and ground-truth (instrumentation); align to LLVM IR CFGs; train model.
- **Inference phase:** only sampled profile + CFG; infer BB frequencies; feed into LLVM PGO pipeline.
- Seamless with LLVM; supports static and post-link optimization workflows.

Methodology

Hybrid GNN (Sequence + Graph)

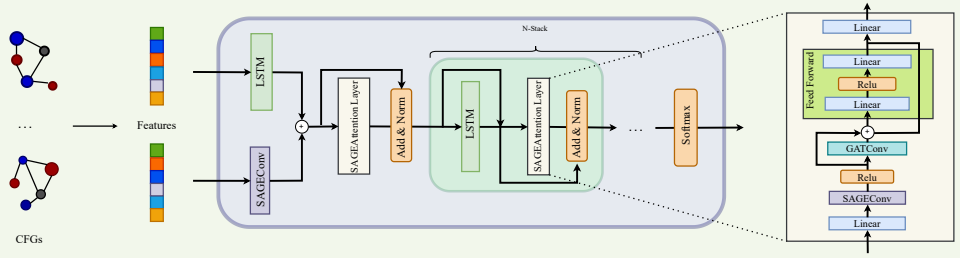

Figure 3: Overview of the model structure in ProfiX for the profile inference problem. A zoomed in view (right) illustrates the internal structure of the SAGEAttention Layer.

- **Parallel encoders:** LSTM (captures execution order) + GraphSAGE (CFG structure).
- **Fusion + SAGEAttention:** GraphSAGE aggregation with multi-head attention to focus on relevant neighborhoods.
- **Softmax head:** normalized per-BB frequencies (valid distribution).

Experiments

Dataset Construction

Table 2: Dataset statistics across five representative applications used in the paper.

Application	Source Funcs	Sampled Funcs	Filtered Funcs	Blocks / Function		Total LOC
				q-50%	q-90%	
Clang 20.0	354,365	4920	2557	98	264	9,192,184
GCC 14.2	196,337	5120	1643	92	251	6,672,238
MySQL 8.32	64,813	2467	1199	78	269	4,551,087
SQLite 3	12,794	235	87	29	81	630,516
SPEC CPU 2017	274,677	8696	575	153	611	13,815,845

- **Train/Val/Test:** Clang, GCC, MySQL, SQLite (80/10/10).
- **OOD evaluation:** SPEC CPU 2017 (unseen).
- Diverse CFG sizes/structures; strong generalization target.

Experiments

Results Highlights

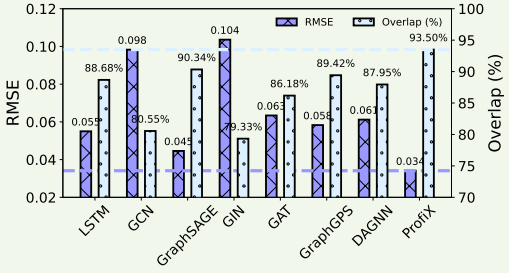

Figure 4: Performance comparison of our model and baseline methods in terms of prediction accuracy (RMSE) and consistency (overlap).

- High overlap on SPEC: up to 98.85% (imagick), 96.26% (mcf), 95.83% (perlbench).
- Improves over Profi (symbolic) and multiple neural baselines.
- Fast, stable convergence with the hybrid architecture.

Experiments

Impact on PGO

LLVM PGO for MySQL:

- **ProfiX**: **12.1%** avg speedup over native.
- Raw sampling: 8.3%; Profi-inferred: 9.7%.

Better profiles → Better optimization choices → Faster binaries.

For more details, please refer to the full paper: Huiri Tan, Juyong Jiang, and Jiasi Shen. “ProfiX: Improving Profile-Guided Optimization in Compilers with Graph Neural Networks”. In: *The Thirty-ninth Annual Conference on Neural Information Processing Systems*. 2025. URL: https://openreview.net/forum?id=CIeWaq0JDl.