Architecture-Specific Evaluation of LLM-Generated Code Optimizations on x86 CPUs

Hirotaka Monya[1] Taro Watanabe[1] Sakriani Sakti[1] Hidetaka Kamigaito[1] Yusuke Sakai[1]
[1] Nara Institute of Science and Technology NLP Laboratory

Motivation / Background

- C optimization is arch-dependent and labor-intensive
- LLMs can follow textual design rules; architecture info can be prompted
- Need a safe loop: generate quickly but verify performance & correctness

Objective

- Quantify when LLM-made C beats gcc −O3 per architecture
- Map effective transforms to AVX/AVX2/AVX-512
- Provide a reproducible protocol; release prompts/variants

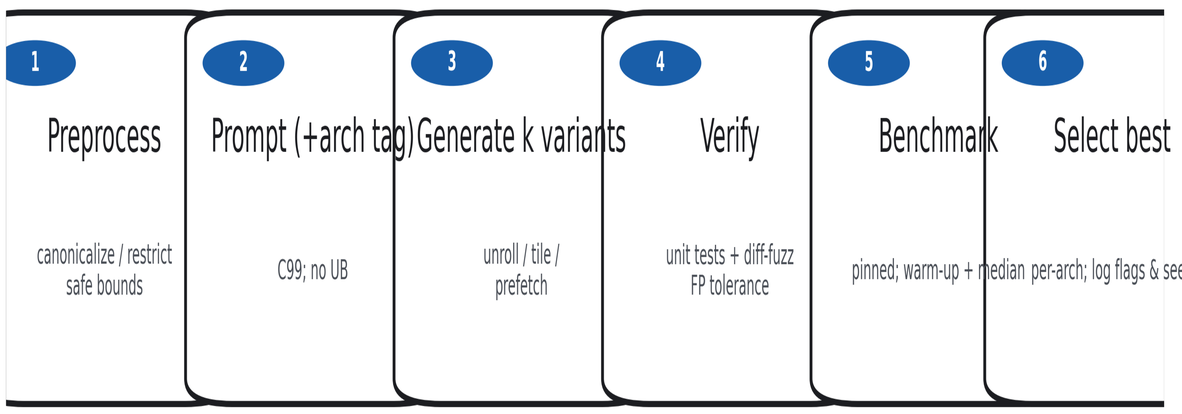
Related Work

- Auto-tuning/DSL/Superopt: Ansor, Exo/Exocompilation, Mirage
- LLM-centric: Self-Refine, LLM Compiler; program-repair survey

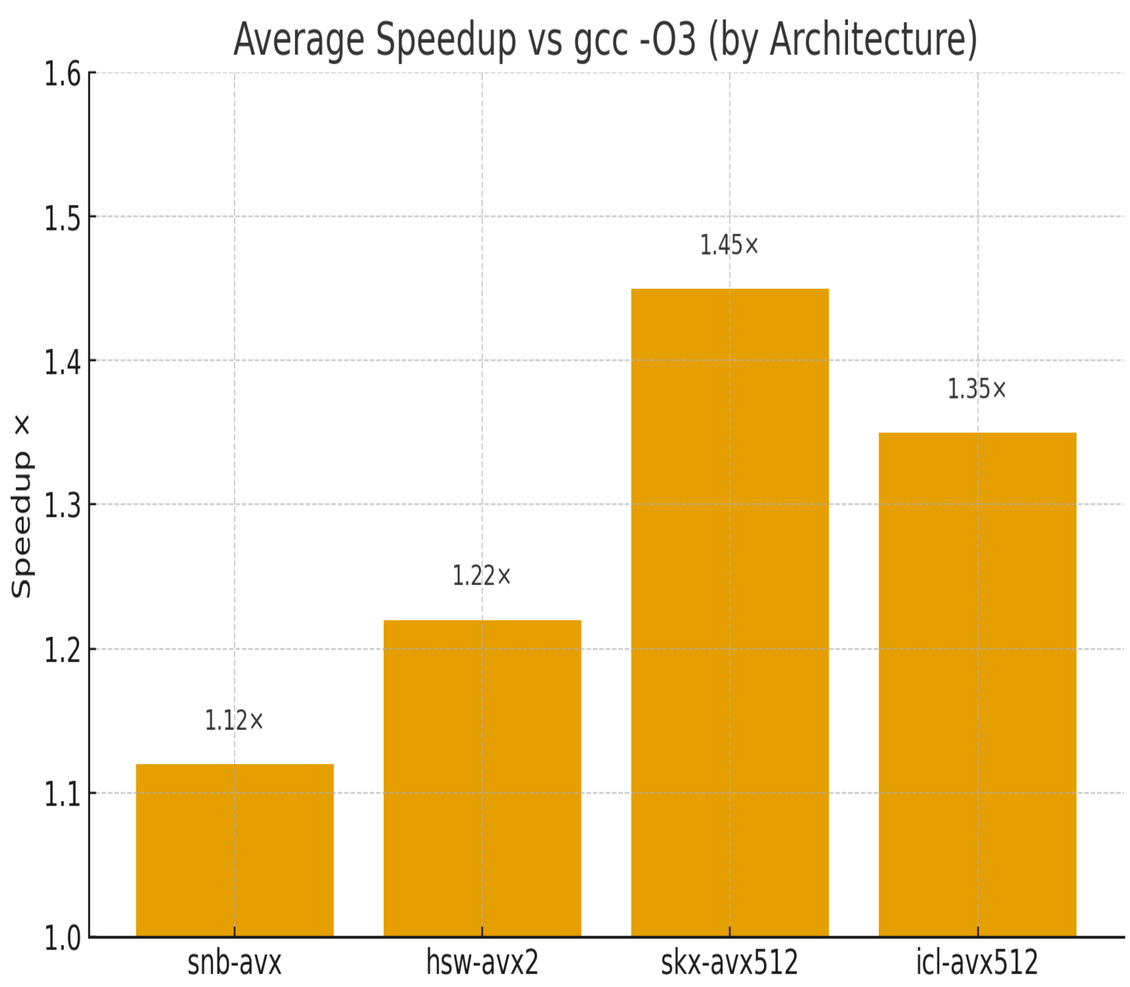
Experimental Setup

- PolyBench-C (30 kernels)
- LLMs: GPT / Claude (n-best)
- CPUs: snb-avx, hsw-avx2, skx-avx512, icl-avx512
- Baselines: gcc −O0/−O3; pinning; median-of-7

Method (Closed Loop)



Results I: Average Speedup by Architecture



Results II: Kernel-wise Best

Kernel	Category	Best arch	Best LLM	Speedup vs -O3	Key transform
gemm	BLAS-like	skx-avx512	GPT	1.48×	tile 64×64; prefetch 64; FMA
2mm	BLAS-like	skx-avx512	GPT	2.05×	tiled + unrolled; aligned loads
3mm	BLAS-like	icl-avx512	GPT	1.62×	tile 32×32; prefetch 128
syrk	BLAS-like	hsw-avx2	Claude	1.36×	unroll ×4; restrict
atax	Regular	hsw-avx2	Claude	1.22×	vectorize + unroll; bounds fix
bicg	Regular	snb-avx	GPT	1.15×	loop reorder; align
fdtd-2d	Stencil	icl-avx512	GPT	1.28×	tile 32×32; mask tails
jacobi-2d	Stencil	hsw-avx2	GPT	1.18×	unroll ×4; prefetch 64

Optimizations by Architecture

- snb-avx: small tiles, unroll×2, aligned contiguous loads
- hsw-avx2: unroll×4, 32–64 tiles, FMA-friendly schedule
- skx-avx512: 32–64 tiles, masked tails, mid-long prefetch
- icl-avx512: 32 tiles, short prefetch, keep FMA density high

Conclusion & Next Steps

Architecture-tagged prompts + verify-then-benchmark improve C performance safely. Next: profile-guided schedules, light auto-tuning, stronger equivalence (SMT/Lean), and GPU/Triton.