GPU Kernel Scientist

An LLM-Driven Framework for Iterative Kernel Optimization

ICML International Conference On Machine Learning

Martin Andrews **Sam Witteveen**

ES-FoMo Workshop
Efficient Systems for Foundation Models

Background

GPU Kernels :

- Challenging to write
  - Specialised skill-set
  - Public information Nvidia-focused

AMD Developer Challenge :

- 'GPUmode' Kaggle-like competition
  - Single GPU target : AMD MI300
  - 3 DeepSeek-inspired kernels

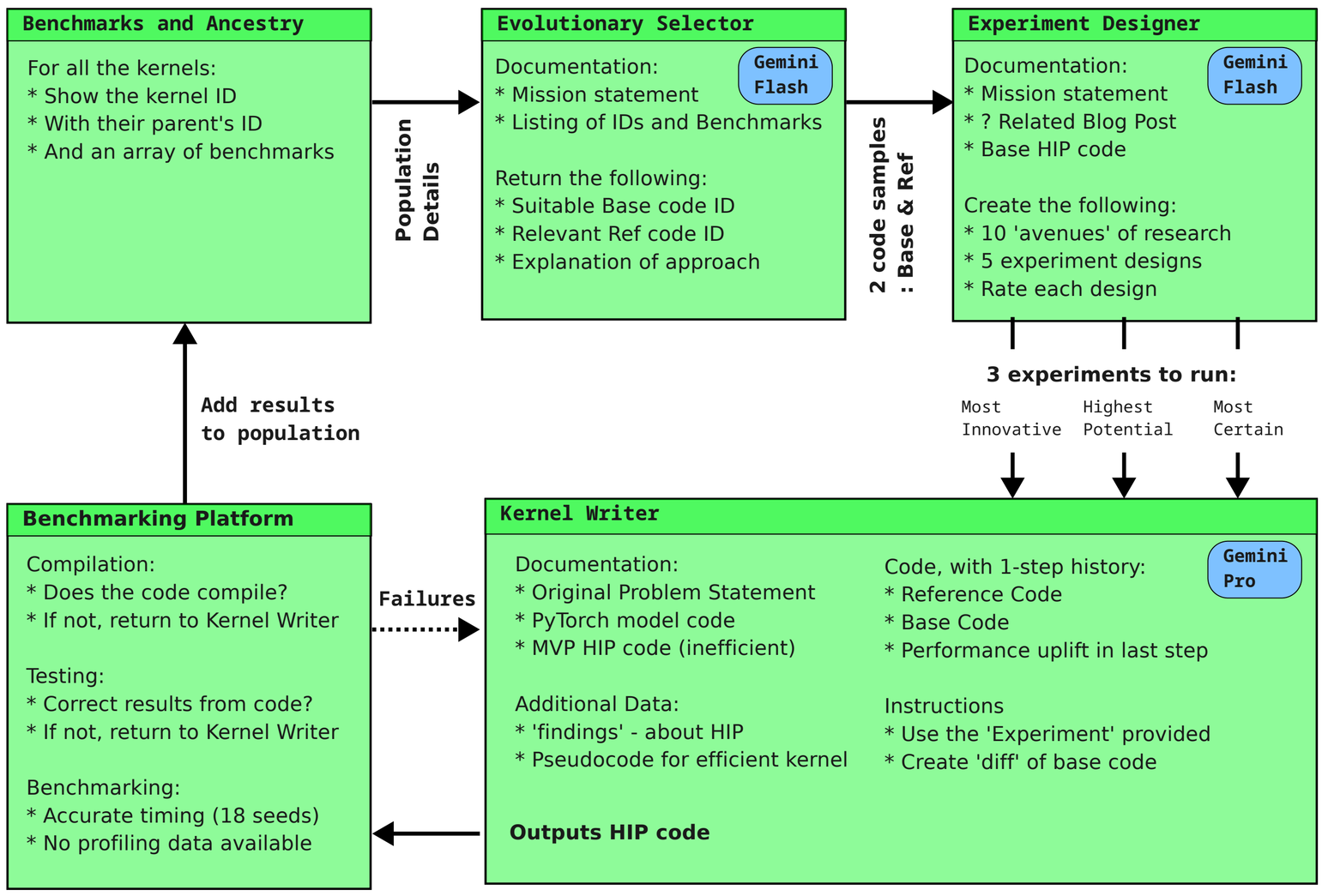
The Focus : `amd-fp8-mm`

- Block-Scaled fp8 matrix multiply
  - 18 specific input sizes
  - PyTorch version (accuracy check)
- No restrictions to approach

Competition Limitiations

- Only REST access to server
  - Returned data very limited
- No access to profiler tooling
- Code limited to ~30KB

GPU Kernel Scientist Process



Key Decisions

LLM-Driven Approach

- Used Gemini 2.5 Flash & Pro models
- Human input (necessarily) limited

LLM-only Evolution/Science Process

- Base (and reference) kernel choice
- Design of experiments to perform
- Writing next generation of kernels

Key Strategies

- Prompt for reasoning about selection
- Prompt for multiple experiments
- "Innovation" scoring for diversity

Results

Competition Timings :

- Naïve HIP: ~5000µs
- PyTorch base-case: ~860µs
  - (uses optimised fp16)
- Winning human entry: ~105µs
  - (top-8 had actual access to MI300)
- This work's LLM-only entry: 450µs

Main Hypotheses Validated :

- LLMs are capable of writing kernels
  - ... even without much public data
- Gemini Pro for exploration/debugging
- Human specialists still have edge
  - But now have good examples

Discussion

Key Take-aways :

- In-context code examples work
- Overall process >> single LLM
- Diversity preservation important

Future directions :

- Approach not limited to AMD ...
  - Many new Hardware devices
  - ... often Software constrained
- Use good code cross-domain
- Selection of in-context materials

Key References

- "Illuminating Search Spaces by Mapping Elites" - Mouret & Clune (2015)
- "AlphaEvolve: A coding agent for scientific and algorithmic discovery." - Google DeepMind (2025)
- "The AI CUDA Engineer: Agentic CUDA Kernel Discovery, Optimization and Composition" - Lange et al. (2025)

Contact

Support for this research was provided by the Google AI Developer Programs team, including access to the Gemini models and GPUs on Google Cloud Platform.