GeckOpt: LLM System Efficiency via Intent-Based Tool Selection

Michael Fore
Microsoft Corporation
Reston, VA, USA
mifore@microsoft.com

Simranjit Singh
Microsoft Corporation
Silicon Valley Campus, CA, USA
simsingh@microsoft.com

Dimitrios Stamoulis
Microsoft Corporation
Redmond, WA, USA
stamoulis.dimitrios@microsoft.com

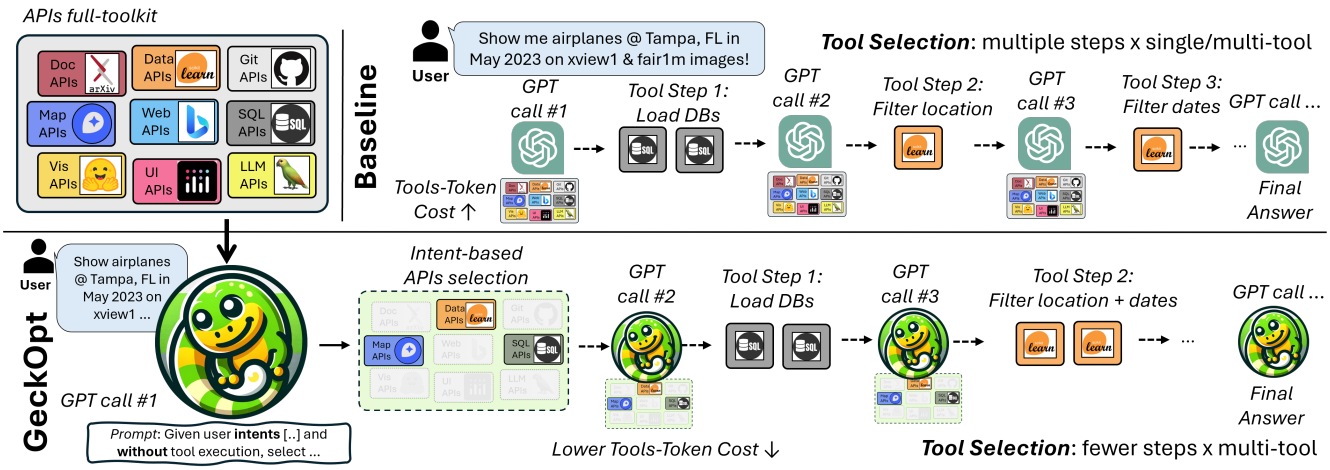

Figure 1: Intent-based tool selection with `GeckOpt`. For each user query, the agent identifies relevant API libraries - without initially executing specific tools! Preliminary results indicate that streamlining the toolset reduces token costs associated with function calls by approximately 25%, while encouraging multi-tool execution over fewer GPT requests.

ABSTRACT

In this preliminary study, we investigate a GPT-driven intent-based reasoning approach to streamline tool selection for large language models (LLMs) aimed at system efficiency. By identifying the intent behind user prompts at runtime, we narrow down the API toolset required for task execution, reducing token consumption by up to 24.6%. Early results on a real-world, massively parallel Copilot platform with over 100 GPT-4-Turbo nodes show cost reductions and potential towards improving LLM-based system efficiency.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; *Computer vision*; *Natural language processing*.

KEYWORDS

Tool-augmented Copilots, Large Language Models

GLSVLSI '24, June 12–14, 2024, Clearwater, FL, USA

ACM ISBN 979-8-4007-0605-9/24/06
https://doi.org/10.1145/3649476.3658784

ACM Reference Format:
Michael Fore, Simranjit Singh, and Dimitrios Stamoulis. 2024. GeckOpt: LLM System Efficiency via Intent-Based Tool Selection. In *Great Lakes Symposium on VLSI 2024 (GLSVLSI '24), June 12–14, 2024, Clearwater, FL, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3649476.3658784

1 PRELIMINARY INVESTIGATION

Large language models (LLMs) augment platforms by facilitating the reasoning, planning, and execution of system tools via natural language instructions, thanks to advancements in prompting techniques [2, 3, 6]. These techniques, acting as LLM-based planners, determine the sequences of tools (*i.e.*, underlying system APIs) to complete tasks. Despite the impressive LLM performance, this usually comes at considerable hardware cost.

In this initial examination, we embrace these advancements, while adopting a system-oriented view to identify bottlenecks in tool-calling processes. By analyzing an in-house large-scale Copilot system, namely `GeoLLM-Engine` [1, 4, 5], we observe that current compositional tool-calling methods often result in *multi-step* × *single-tool* executions (Fig. 1 top). This contrasts with the more desirable *multi-step* × *multi-tool* approach, which aggregates multiple API calls within a single GPT step (Fig. 1 bottom). Since each step translates to a GPT request and token consumption, the total number of steps significantly impacts system cost.

Our objective is to strike a balance between the inclination of compositional reasoning - breaking down tasks into multiple steps for task comprehension for the LLM-planner - and minimizing the system steps required for task completion. Our early analysis

Table 1: Query categorization based on *intent* following [5].

Intent Example	Related Query	APIs
Load→Filter→Plot	Plot xview1 images around Tampa Bay, FL, USA	`SQL_apis()`, `data_apis()`, `map_apis()`
UI/Web Navigation	Search Bing for "System-efficient LLM prompting"?	`web_apis()`, `UI_apis()`
Information Seeking	Which model to use for airplane detection?	`wiki_apis()`

Table 2: `GeckOpt` intent-based tool-selection; enhanced agent performance on the `GeoLLM-Engine-10k` baselines [5].

GPT-4 Turbo (0125)	Correct. Rate ↑	Success Rate ↑	Obj. Det F1 ↑	LCC R ↑	VQA Rouge-L ↑	Tokens/Task ↓
CoT Zero-Shot [5]	80.88%	77.35%	87.99%	96.56%	65.29%	23.6k
CoT Zero-Shot + `GeckOpt`	79.13%	77.03%	86.03%	96.46%	63.21%	**18.48k**
CoT Few-Shot [5]	84.01%	80.00%	88.40%	**99.89%**	67.65%	25.8k
CoT Few-Shot + `GeckOpt`	83.11%	79.26%	88.19%	98.53%	68.66%	**19.45k**
ReAct Zero-Shot [5]	84.27%	80.03%	**89.34%**	98.83%	68.11%	26.7k
ReAct Zero-Shot + `GeckOpt`	83.87%	79.46%	88.02%	98.74%	68.73%	**20.38k**
ReAct Few-Shot [5]	**84.31%**	**81.11%**	83.85%	99.63%	**69.37%**	32.5k
ReAct Few-Shot + `GeckOpt`	84.10%	80.17%	83.31%	99.01%	69.26%	**25.14k**

reveals that, as is to be intuitively expected, presenting the LLM with a narrower selection of tools, combined with a clearer understanding of the task, encourages the aggregation of more tools per step. To this end, we draw inspiration from the concept of *LLM intent*, where state-of-the-art methodologies demonstrate LLMs' capability to discern "user intent" and utilize it to categorize tasks into granular user-defined templates [8].

`GeckOpt`: Our methodology begins with an offline phase where tasks are mapped to intents and associated tools with minimal human involvement (Table 1). Next, **at runtime** and for each prompt, we first engage the LLM agent to determine the task's intent and to identify the relevant subsets of API libraries - ahead of pinpointing specific tools at this stage! This intent-driven "gating," while it incurs the minor cost of an extra API call, effectively narrows down the tool selection pool for all subsequent compositional prompting to proceed as usual. The result is a reduction in token requirements and a move towards executing multiple tools within a single API call, optimizing both token usage and system resource allocation. Given that our approach is *fully* GPT-driven, it adeptly handles failures where the initial API selection is incorrect, with the agent being instructed via prompting to revert to the full toolset.

2 PRELIMINARY RESULTS

Experimental setup: as a representative Copilot platform, we use our `GeoLLM-Engine` framework [5] that spans several remote sensing tasks, open-source datasets, and API tools for loading, filtering, processing, and visualizing data. We report agent performance metrics on the `GeoLLM-Engine-5k` benchmark for CoT [6] and ReAct [7] prompting. We refer the reader to [5] for further details.

Results: Table 2 shows that `GeckOpt` reduces token usage across various baselines up to 24.6% with small performance variation. While we note slight deviations, possibly attributable to employing non-zero temperature settings in LLM function calling to foster creative answers in visual question answering, we observe that cost reductions come at negligible performance degradation within 1% in terms of success rates. Extrapolated across numerous user sessions over our cloud-native platform, such token reduction by almost a quarter can culminate in substantial cloud cost savings.

Limitations - Future Work: We note that this is a preliminary investigation where geospatial tasks are well-suited to intent-based granularity. An important next step is to confirm the generalizability of the methodology across various function-calling benchmarks. Last, as our study considers only cloud endpoints, we are currently expanding to local LLM execution. Given the correlation between token usage and operational costs, we anticipate additional improvements for LLM system utilization, underscoring the potential of intent-based tool selection for enhancing hardware efficiency.

REFERENCES

[1] Yanan Jian, Fuxun Yu, Simranjit Singh, and Dimitrios Stamoulis. 2023. Stable Diffusion For Aerial Object Detection. In *NeurIPS 2023 Workshop on Synthetic Data Generation with Generative AI.*

[2] Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. 2023. Chameleon: Plug-and-Play Compositional Reasoning with Large Language Models. arXiv:2304.09842 [cs.CL]

[3] Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2023. ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs. arXiv:2307.16789 [cs.AI]

[4] Simranjit Singh, Michael Fore, and Dimitrios Stamoulis. 2024. Evaluating Tool-Augmented Agents in Remote Sensing Platforms. In *ICLR 2024 Workshop: 2nd Machine Learning for Remote Sensing Workshop.*

[5] Simranjit Singh, Michael Fore, and Dimitrios Stamoulis. 2024. GeoLLM-Engine: A Realistic Environment for Building Geospatial Copilots. In *CVPR 2024 Workshop EARTHVISION 2024.*

[6] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903 [cs.CL]

[7] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. arXiv:2210.03629 [cs.CL]

[8] Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. 2023. WebArena: A Realistic Web Environment for Building Autonomous Agents. arXiv:2307.13854 [cs.AI]