

<https://doi.org/10.1038/s41534-026-01195-1>

Symbolic analysis of Grover search algorithm via Chain-of-Thought reasoning and quantum-native tokenization

Check for updates

Min Chen¹, Jinglei Cheng¹, Pingzhi Li², Haoran Wang², Tianlong Chen²✉ & Junyu Liu¹✉

Understanding the high-level conceptual structure of quantum algorithms from their low-level circuit representations is a critical task for verification, debugging, and education. While traditional numerical simulators can calculate output probabilities, they do not explicitly surface the underlying algorithmic logic, such as the function of an oracle or embedded symmetries. In this work, we shift the focus from numerical simulation to *symbolic analysis*, investigating whether large language models (LLMs) can automatically *interpret* quantum circuits and articulate their logic in a human-readable format. We introduce GroverGPT+, a model that leverages Chain-of-Thought reasoning and quantum-native tokenization to analyze Grover's search algorithm. We use Grover's algorithm as a controlled testbed, as its well-defined analytical properties allow for rigorous verification of the model's reasoning process. Our primary finding is that GroverGPT+ successfully identifies the oracle and its marked states directly from circuit representations. The model's key output is not a final probability, but a structured, *interpretable reasoning trace* that mirrors human expert analysis, effectively translating procedural circuit steps into conceptual insights. Furthermore, we establish a structured benchmark for this symbolic analysis task and explore its empirical extrapolation, describing the model's performance as the number of qubits increases. These findings position LLMs as powerful tools for automated quantum algorithm analysis and verification. More fundamentally, this work offers a first step towards using such models as scientific probes, suggesting that an algorithm's "learnability" by a classical model can provide a new, complementary perspective on its conceptual complexity, a topic of core interest to quantum information science.

Quantum computing has emerged as a transformative paradigm, with algorithms like Shor's¹ and Grover's² demonstrating profound theoretical advantages over classical counterparts³. However, a significant gap exists between the low-level procedural descriptions of quantum algorithms, such as quantum assembly language (QASM) code⁴, and the high-level conceptual understanding required for their design, verification, and debugging. While traditional numerical simulators are essential tools for calculating the evolution of state vectors^{5–7}, they are "semantically blind": they output final probabilities but do not explicitly surface the underlying algorithmic logic, such as the function of an oracle or embedded symmetries within the circuit. This semantic gap presents a bottleneck, motivating the development of new tools that can automatically interpret and reason about the structure of quantum algorithms.

Recently, large language models (LLMs) have demonstrated remarkable capabilities in bridging such semantic gaps in various domains, from code generation to complex scientific reasoning^{8,9}. Their ability to process and generate structured, human-readable text makes them prime candidates for a task that moves beyond numerical calculation towards conceptual interpretation. This inspires our central research question: *Can LLMs be adapted to function not as numerical simulators, but as symbolic analyzers that interpret quantum circuits and articulate their algorithmic logic in an explicit, step-by-step manner?*

In this work, we explore this question by introducing GroverGPT+, an LLM-based framework designed for the symbolic analysis of Grover's algorithm. Here, the term "GPT" (i.e., generative pre-trained transformer) is used as a functional shorthand to denote a transformer-based reasoning model adapted to a specific scientific domain, rather than implying standard

¹Department of Computer Science, The University of Pittsburgh, Pittsburgh, PA, USA. ²Department of Computer Science, The University of North Carolina at Chapel Hill, Chapel Hill, NC, USA. ✉e-mail: tianlong@cs.unc.edu; junyuliu@pitt.edu

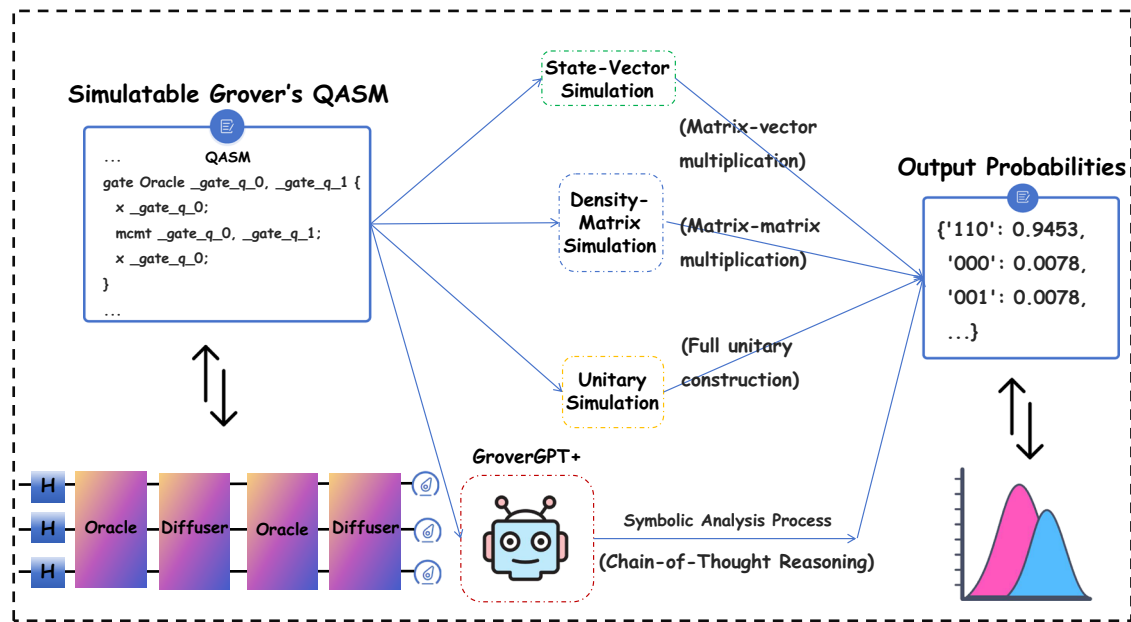


Fig. 1 | A comparison of two distinct tasks: classical simulation and symbolic analysis. Traditional methods (top path), such as state-vector simulation, take a QASM input and perform numerical operations like matrix–vector multiplication to directly compute the final output probabilities. In contrast, our approach with

GroverGPT+ (bottom path) first performs symbolic analysis using Chain-of-Thought reasoning to generate an interpretable reasoning trace that explains the circuit’s logic. The final probabilities are then inferred from this analysis.

GPT-style pretraining. To enable the model to fluently “read” the language of quantum circuits, we introduce *quantum-native tokenization*, a method for tokenizing QASM representations by extending the vocabulary of a base tokenizer with quantum-specific operations. To compel the model to “think step-by-step” and externalize its analysis, we curate a large corpus of training data and employ *Chain-of-Thought (CoT) supervised fine-tuning*. We choose Grover’s algorithm as our primary testbed as its non-trivial, well-defined structure provides an ideal, controlled environment where the accuracy of the model’s symbolic reasoning can be rigorously verified against a known analytical solution.

Our results demonstrate that GroverGPT+ can successfully analyze quantum circuits from their QASM representations. Instead of merely outputting final probabilities, the model generates structured, interpretable reasoning traces that correctly identify high-level algorithmic structures, including the oracle and its marked states. This work establishes a benchmark for the task of automated symbolic analysis of quantum circuits and provides empirical evidence of the model’s extrapolation behavior. Ultimately, our findings position LLMs as a new class of complementary tools for quantum information science: they are not for replacing numerical simulators, but for aiding in tasks requiring conceptual understanding, such as automated verification, debugging, and education. This work opens a new direction for AI systems that reason about the logic of quantum algorithms, suggesting that an algorithm’s “learnability” can itself provide a new lens for understanding its conceptual complexity, a topic of core interest to quantum information science.

Results

Details of the tasks

The central task of this work is the symbolic analysis of quantum circuits. Given a circuit representation (e.g., in QASM format), the primary objective is to generate a human-readable reasoning trace that identifies the circuit’s high-level algorithmic components. For Grover’s algorithm (see Supplementary Information 1 for a detailed introduction), this corresponds to correctly identifying the oracle and its marked states. This task is distinct from numerical simulation as its principal output is not a final probability distribution, but rather the symbolic, conceptual insight derived from the circuit’s structure. A comparison between our symbolic analysis task and

traditional classical simulation is illustrated in Fig. 1. Besides, our model also generates a final probability distribution over the computational basis states. We utilize this numerical output for two purposes: (i) as a scalable metric to quantitatively evaluate the model’s ability to identify the marked states, and (ii) as a method for end-to-end validation, where a correct symbolic understanding should lead to a high-fidelity final state distribution. Formally, let C denote a Grover circuit. The goal of symbolic analysis is to produce a reasoning trace $R(C)$ that explicitly identifies the set of marked states, \mathcal{M}_{true} . The predicted probability distribution $g_{\theta}(C)$ serves to verify the correctness of $R(C)$. Below, we introduce the evaluation metrics:

We first introduce search accuracy (SA). Given $k = |\mathcal{M}_{true}|$, we sort candidates by their predicted probabilities p_i^{model} in descending order, and in case of ties by the integer value of the binary state in ascending order. Let T_k be the top- k states. To avoid spurious hits under near-uniform predictions, we require a minimum confidence threshold τ :

$$\widehat{\mathcal{M}}_{model} = \{x \in T_k : p_x^{model} \geq \tau\}. \tag{1}$$

We then define

$$SA = \frac{|\widehat{\mathcal{M}}_{model} \cap \mathcal{M}_{true}|}{k}. \tag{2}$$

If $|\widehat{\mathcal{M}}_{model}| < k$, the remaining slots are treated as misses. Notably, the definition above applies generally to arbitrary m . In our experimental setting, the number of marked states satisfies $k \leq 3$, we thereby set $\tau = 0.3$ by default. Meanwhile, the models are prompted or trained to output at least the top-30 candidates, so truncation never affects SA, i.e., we have ensured $k \leq t$ denoted as the number of truncated states in practice. If a model outputs fewer candidates, any truncated marked states are treated as misses.

Besides, we introduce classical fidelity (CF). Our evaluation focuses on the similarity between *measurement amplitude probability distributions* produced by a method and by an ideal simulator. Given two probability distributions over the computational basis, $p = (p_1, \dots, p_d)$ and $q = (q_1, \dots, q_d)$ with $d = 2^n$, we use the *classical fidelity (CF)*^{3,10–12} (see Supplementary

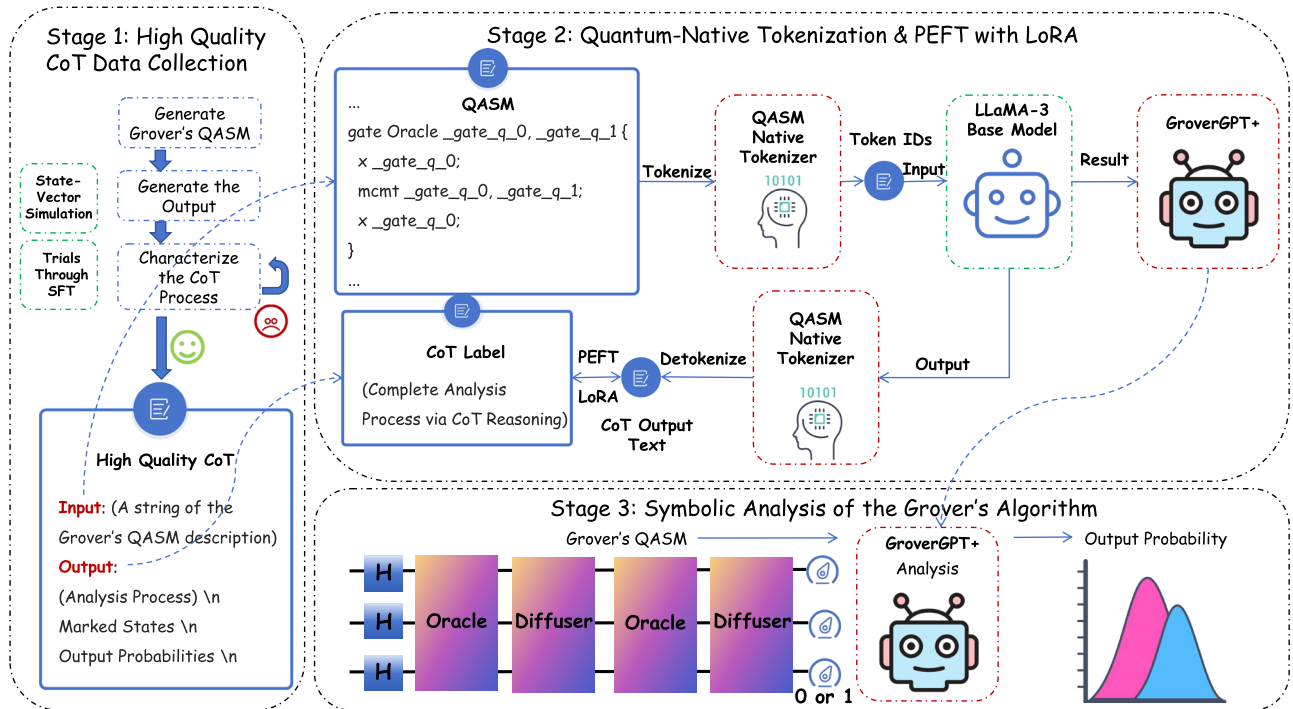


Fig. 2 | The overall framework of GroverGPT+ and its application for the symbolic analysis of Grover's algorithm consists of three stages. Stage 1: We initiate by collecting high-quality CoT data tailored for Grover's algorithm. This involves generating Grover's QASM circuits, performing classical simulations via the state-vector simulation method, and labeling the output distributions along with marked states as CoT supervision targets. Stage 2: The collected QASM-CoT pairs

are tokenized using our QASM-native tokenizer. We then adopt PEFT using the LoRA technique to specialize the base LLM for the symbolic analysis of quantum circuits while maintaining training efficiency. Stage 3: GroverGPT+ can now serve as a tool for symbolic analysis: given a Grover's QASM circuit, it generates an interpretable reasoning trace that identifies the marked states and infers the final-state probability distribution through CoT reasoning.

Information 4 for the relation to quantum state fidelity).

$$CF(p, q) = \left(\sum_{i=1}^d \sqrt{p_i q_i} \right)^2 \tag{3}$$

When a model outputs a truncated distribution (e.g., top-30 states for baseline LLMs), we treat any state not present in the model's output as having zero probability. Specifically, for the model's predicted distribution p , if a computational basis state i is not included in the output, we set $p_i = 0$. The CF is then computed using Eq. (3) over all $d = 2^n$ states, where missing states contribute zero to the sum. This approach ensures that truncation does not artificially inflate fidelity scores, as states omitted from the output are penalized through their zero-probability assignment.

Overview of GroverGPT+

Figure 2 presents the overall framework of GroverGPT+. It is an LLM with 8 billion parameters supervised fine-tuned on the base Llama-3¹³ model. To conduct our task, we first develop GroverGPT+ through stages, including high-quality CoT data collection (Stage 1), quantum-native tokenization and parameter-efficient fine-tuning (PEFT) with low-rank adaptation (LoRA) (Stage 2), and then perform symbolic analysis of Grover's algorithm (Stage 3). Below are the details for each stage:

In Stage 1, we first generate high-quality CoT training data. Grover's QASM circuits are generated starting from 2 qubits, marking 1–3 target states. For each circuit size, the number of marked states never exceeds the number of qubits and is capped at three. See Supplementary Information 12 for the detailed experimental setup. Corresponding probability amplitudes are computed using brute-force state-vector simulation. CoT processes are then annotated based on outputs from an intermediate

supervised fine-tuned LLM. We finalize the curation of the dataset once desirable CoT processes are observed.

In Stage 2, we supervised fine-tune GroverGPT+ using PEFT with LoRA. Initially, collected QASM descriptions are tokenized into token IDs using our quantum-native tokenizer (detailed in Section IVA and Supplementary Information 6). These token IDs serve as inputs to the LLaMA-3 base model, whose outputs are then detokenized into a text format. PEFT with LoRA is conducted for higher training efficiency.

In Stage 3, once trained, GroverGPT+ accepts Grover's QASM descriptions as input and performs symbolic analysis via CoT reasoning. The model outputs structured text including intermediate reasoning steps, marked states, and the output probability amplitudes of all computational basis states. Specifically, the complete CoT process is detailed in Supplementary Information 8.

When analyzing Grover's algorithm, GroverGPT+ only requires a pure QASM description of a quantum circuit as input, without additional information, while general-purpose LLMs need a meticulous prompt design to guide the LLM to output correct results. Therefore, GroverGPT+ offers a more streamlined and efficient workflow for this analysis task. Below briefly introduces how this is achieved:

Firstly, GroverGPT+ extracts the Oracle entity from the whole bunch of long QASM for searching the marked computational basis states in the following steps. Secondly, GroverGPT+ reasons about each corresponding marked state according to the oracle construction extracted before. GroverGPT+ leverages how the target states are marked according to Grover's algorithm design. Thirdly, following the second step, GroverGPT+ outputs the probabilities of the marked states and the unmarked states according to the reasoned information. It is achieved through a learned mapping from basic information, including the number of qubits, the number of marked states, and the searched results from the previous steps, to the probability amplitudes for each computational basis state.

Fig. 3 | Performance of GroverGPT+ against baseline LLMs in the symbolic analysis of Grover’s algorithm across varying numbers of qubits. Both the SA (a) and CF (b) serve as the evaluation metrics. Solid lines show means; discrete error bars indicate uncertainty (mean \pm std (σ) or where noted, mean \pm std).

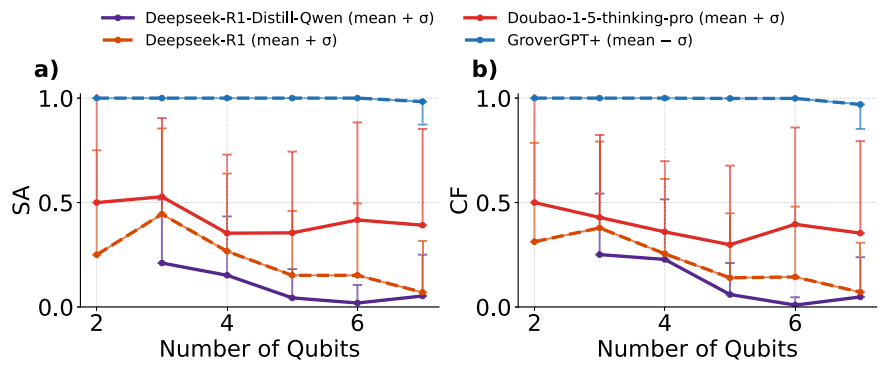
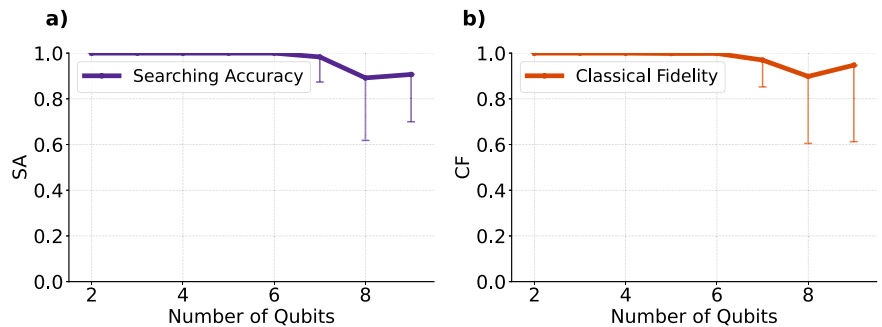


Fig. 4 | Nearby OOD performance of GroverGPT+ when scaling up to 8 and 9 qubits (beyond the training range). Both the SA (a) and CF (b) serve as the evaluation metrics. Solid lines show means; discrete error bars indicate uncertainty (mean \pm std).



Experimental settings

We first introduce the general experimental settings, then we conduct empirical studies of GroverGPT+ with respect to its in-distribution and nearby out-of-distribution (OOD) performance. Furthermore, we also evaluate its computational scalability. Notably, evaluations with respect to its CoT advantage, the technique of quantum-native tokenization, and its extrapolation performance are respectively shown in Supplementary Information 13–15. The hyperparameter settings for this section can be found in Supplementary Information 16.

We first detail the general experimental settings. We evaluate GroverGPT+ under different input formats and quantum circuit settings. Specifically, we consider two types of inputs: *Full-circuit Input* and *Oracle-only Input*, each designed to probe different aspects of model capability. The experiments vary the number of qubits and marked states to comprehensively assess performance, while also exploring the model’s scaling behavior under increasing circuit sizes. For detailed configurations, see Supplementary Information 12. Notably, for the oracle-only input setting, the target distribution is defined using the analytically optimal iteration number k_{opt} , which guarantees the maximum success probability for a given (n, t) configuration, where n is the number of qubits and t is the number of marked states. This ensures that the output distributions are consistently defined even when the iteration number k is not explicitly present in the oracle-only input. In all plots, solid lines denote the mean over runs, and discrete error bars indicate mean \pm one standard deviation (std) for both SA and CF.

Empirical study of GroverGPT+ in analyzing Grover’s algorithm

In this section, we empirically evaluate GroverGPT+’s performance in the symbolic analysis of Grover’s quantum search algorithm. We first test its ability using *full-circuit inputs* for qubit counts $n \in \{2, 3, \dots, 7\}$, corresponding to the training data. Given that the maximum token length is exceeded at $n = 9$ (see Supplementary Information 15), we separately assess GroverGPT+’s generalization performance at $n = \{8, 9\}$ and compare it with the trained scenarios. Additionally, we evaluate the performance on analyzing *oracle-only inputs* across a broader range, $n \in \{2, 3, \dots, 13\}$. The results

are, respectively, shown in Figs. 3–5. Below are observations regarding the results.

Figure 3 illustrates that baseline LLMs exhibit relatively low SA and fidelity values (around 0.2–0.5) with substantial standard deviations, particularly for qubit counts between 5 and 7. For instance, at 7 qubits, baseline models achieve SA and fidelity below 0.4, indicating unstable performance. Conversely, GroverGPT+ consistently attains high SA and fidelity values approaching 1.0 with minimal variability, highlighting its stability and superior performance. This advantage likely results from GroverGPT+’s specialized fine-tuning using high-quality CoT data and quantum-native tokenization, enhancing result consistency across varying circuit sizes.

Figure 4 evaluates GroverGPT+ on qubit counts slightly beyond the training range by training the model up to 7 qubits and testing it on 8–9 qubits, i.e., a nearby OOD by qubit count setting. We can observe a mild drop in SA \approx 0.89 with 8 qubits and SA \approx 0.91 with 9 qubits, while the CF remains above 0.90. These results indicate that the model is not merely memorizing the training distribution and exhibits potential to scale to larger circuits within the tested range (2–9 qubits).

Figure 5 further examines the Oracle-only input format on 10–13 qubits, where both SA and the CF remain close to 1.0. We view this as encouraging evidence that the compact Oracle-only representation supports evaluation at larger qubit counts under context-length constraints within the tested range (2–13 qubits), indicating potential scalability within this tested regime.

Notably, the evaluations are based on both single-target and multi-target Grover circuits. Compared to the single-target case, multi-target Grover’s algorithm requires multiple oracle blocks, each responsible for flipping the phase of one marked state. This leads to longer and structurally more complex QASM circuits. Consequently, the CoT reasoning produced by the model must also handle multiple oracle blocks, increasing the length and potential variability of reasoning chains. This may, in turn, reduce both SA by missing one or more marked states and CF by spreading probability mass over incorrect states. While single- and multi-target Grover circuits differ in their QASM representations, we emphasize that our evaluation does not hinge on this distinction. We are tackling an analysis scenario

Fig. 5 | The performance of GroverGPT+ under the Oracle-only input setting with the number of qubits $n = \{2, 3, \dots, 13\}$. Both the SA (a) and the CF (b) serve as the evaluation metrics. Solid lines show means; discrete error bars indicate uncertainty (mean \pm std).

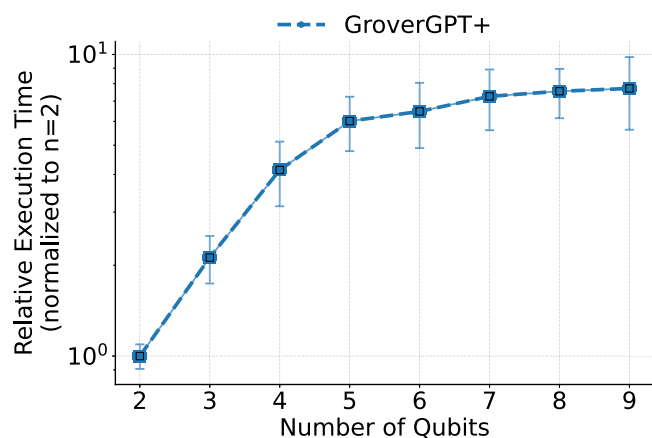
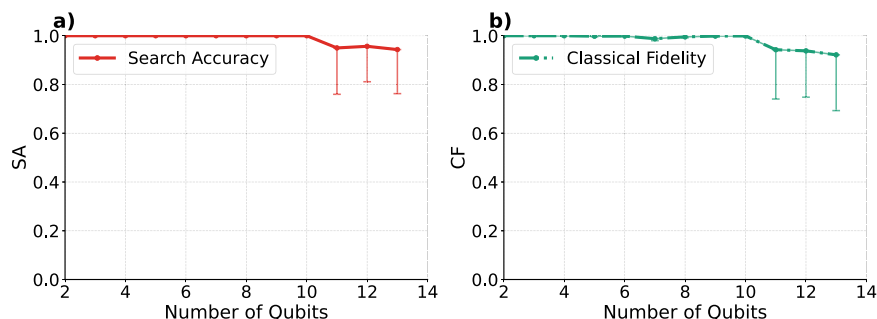


Fig. 6 | Computational scalability of GroverGPT+. The plot shows the relative inference time for the symbolic analysis task, normalized to the runtime at 2 qubits, as a function of the number of qubits. Solid lines show the mean over three runs; discrete error bars indicate mean \pm one standard deviation.

where the inputs are directly simulatable QASMs, thereby the number of marked states is not known a priori, and hence the task setting naturally mixes single- and multi-target instances, which makes the evaluation closer to realistic scenarios. Accordingly, when reporting the SA and CF, we aggregate results by computing the mean and standard deviation across both single- and multi-target instances rather than isolating them.

Computational scalability of GroverGPT+

In this section, we characterize the *computational scalability* of the symbolic analysis task performed by GroverGPT+. We define this as the growth of the model’s inference time with respect to the number of qubits, n . To highlight the scaling trend independently of hardware-specific overheads, we measure the relative execution time. Formally, let $T(n)$ denote the mean inference time for an n -qubit circuit. We define the scalability metric as

$$S(n) = \frac{T(n)}{T(2)}, \quad (4)$$

which normalizes the runtime to its value at $n = 2$. The absolute execution times are also evaluated in Supplementary Information 7 for completeness.

The experiments were conducted by measuring the inference time of GroverGPT+ across qubit sizes ranging from 2 to 9. For each qubit count, three runs were performed on an NVIDIA RTX A6000 GPU with 48 GB GDDR6 VRAM to compute the mean and standard deviation. The reported runtimes refer to inference time only and do not include the one-time cost of model training. Refer to Supplementary Information 7 for more details.

The results are plotted on a logarithmic scale in Fig. 6. Notably, all scaling observations reported in this section are limited to the tested range. We observe that the execution time of GroverGPT+ exhibits a gentle and

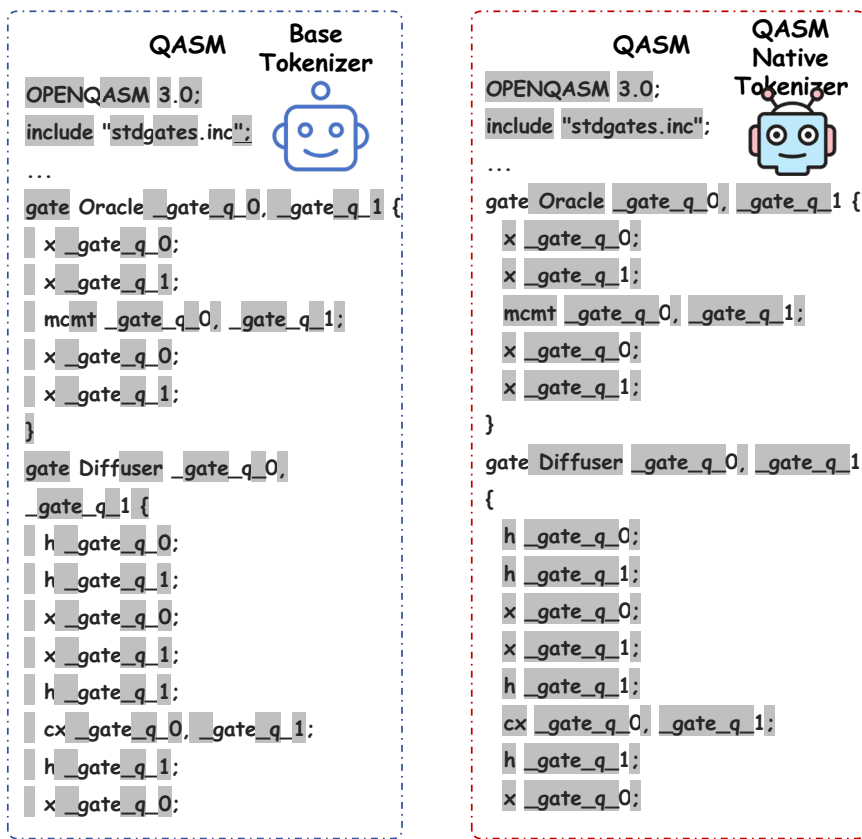
notably sub-linear growth trend with respect to the number of qubits. The relative execution time remains consistently within a 1–10 \times range compared to its baseline at 2 qubits, and the variance remains stable. This favorable scaling is a direct consequence of our symbolic analysis approach. Instead of performing tensor-based state evolution, which is inherently exponential, GroverGPT+’s CoT reasoning operates on the symbolic structure of the QASM input. This allows the model to avoid the exponential computational overhead associated with methods that must track the full 2^n -dimensional state vector, i.e., GroverGPT+ can also serve as a useful tool for finding the final output probability given the simulatable QASM within the tested regimes. To further explore, since GroverGPT+ directly operates on simulatable QASM inputs and finally generates the output amplitude distributions, we also compare this end-to-end latency with some traditional classical simulations. Refer to Supplementary Information 7 for details.

Discussion

Our work began with a central research question: *Can LLMs be adapted to function not as numerical simulators, but as symbolic analyzers that interpret quantum circuits and articulate their algorithmic logic in an explicit, step-by-step manner?* Our findings provide an answer for this. Through the development of GroverGPT+, we have demonstrated that an LLM, when equipped with domain-specific techniques like quantum-native tokenization and structured Chain-of-Thought training, can successfully parse low-level QASM code and produce high-level, interpretable reasoning traces. These traces are not merely a byproduct; they are the primary output, revealing the model’s “understanding” of algorithmic components like the oracle and its marked states. Our work thus is complementary to prior studies such as GroverGPT¹⁴ (see Supplementary Information 3 for detailed comparison) by shifting the objective from reproducing numerical outcomes to elucidating the underlying logical process, establishing a stringent benchmark for the automated analysis of quantum algorithms.

For the domain of quantum information science, our study points towards more than a practical tool: it suggests a promising avenue for future research in evaluating algorithmic complexity through a new conceptual lens. While traditional metrics focus on physical resources like gate count and circuit depth, a compelling future direction would be to investigate whether the ‘learnability’ of an algorithm by a general-purpose reasoner like an LLM can serve as a proxy for its descriptive or conceptual complexity. Our findings on computational scalability provide initial evidence for this direction. We observed that the inference time for analyzing Grover’s algorithm, known for its highly regular and iterative logic, scales sub-linearly with the number of qubits. This favorable scaling indicates the model’s effort might be tied to the algorithm’s low conceptual complexity, not the exponential size of its Hilbert space. This framework opens new research questions for quantum information theory: would analyzing a complex variational quantum eigensolver (VQE) ansatz or identifying stabilizer generators in a quantum error correction (QEC) code reveal a higher conceptual complexity through this new lens? This approach begins to reframe the LLM from a simple tool into a scientific instrument for probing the nature of quantum algorithms themselves, marking an early and encouraging

Fig. 7 | Comparison of the base tokenizer and the quantum-native tokenizer on QASM input. Each grey and non-grey segment represents a distinct token. The base tokenizer fragments the syntax into subword units, while the quantum-native tokenizer preserves gate operations and qubit references as cohesive tokens, resulting in more compact and efficient representations.



exploration into a new, AI-driven approach to theoretical quantum information science.

Besides, our work contributes a generalizable methodology for applying LLMs to the symbolic analysis of quantum circuits. For example, our *quantum-native tokenization* is not specific to Grover’s algorithm. Instead, it can be adopted to efficiently represent any QASM-described circuit in a way that is semantically meaningful to a transformer architecture. Similarly, our strategy for *structure-aware CoT fine-tuning* serves as a template for teaching an LLM the specific logical steps of other quantum algorithms (see Supplementary Information 8). Together, these techniques form a foundational toolkit that enables future research into the automated analysis of a much wider range of quantum algorithms, from the Quantum Fourier Transform to complex variational circuits, thereby paving the way for more sophisticated AI-driven tools in the quantum domain.

Methods

Quantum-native tokenization

The LLaMA model is primarily trained on approximately 1.4T English-language tokens sourced from the Internet¹⁵, lacking native support for quantum-specific languages such as QASM. As a result, it struggles to tokenize QASM code effectively, leading to fragmented subword sequences that ignore the language’s syntactic and semantic structure. This inefficient tokenization increases input length and memory usage. As illustrated in Fig. 7, the base tokenizer breaks down QASM statements into disjointed pieces based on natural language rules, rather than recognizing them as coherent units. To overcome these limitations, we propose a quantum-native tokenizer tailored to the structure of quantum programming languages. Specifically designed for QASM, this tokenizer captures key elements—such as gate operations, qubit identifiers, and block constructs—as discrete, semantically meaningful tokens. By aligning with the intrinsic structure of QASM, it achieves more compact and efficient tokenization, reducing context length and improving

memory efficiency in downstream tasks. The development process is detailed as follows:

Firstly, we collect a large-scale dataset encompassing a comprehensive range of QASM circuit descriptions, covering quantum circuits with qubit numbers ranging from $n = 2$ to $n = 9$. Secondly, to systematically process and analyze these QASM circuits, we develop a set of custom parsing rules tailored to the unique syntactic structure of QASM. These rules tokenize each line of the QASM files to accurately extract quantum gate definitions and operation commands. Specifically, our rule-based approach first identifies gate definitions using regular expressions that capture gate names, parameters, and qubit arguments; any numerical suffixes specific to certain internal naming conventions (e.g., `_gate_q_`, `unitary_`, `mcx_vchain_`) are stripped to maintain consistency and generality in subsequent analyses. For standard quantum commands, we parse the operation names, optional parameters, and target qubits separately. These components are then tokenized, again removing extraneous numerical suffixes to ensure uniformity. The parsing mechanism also explicitly handles structural delimiters, such as opening and closing braces, crucial for correctly interpreting nested gate definitions and circuit hierarchies.

These custom rules enable scalable and automated preprocessing of QASM descriptions, facilitating efficient symbolic analysis. Figure 7 also presents an example of how a single QASM description is tokenized using the base tokenizer and the quantum-native tokenizer, which pinpoints the brought efficiency. In total, we extend 266 specific vocabularies that contain complete semantics, such as `mcx` indicating a multi-controlled X gate. The rule definitions, together with their corresponding Python implementations, are elaborated in Supplementary Information 6.

Chain-of-Thought training

Chain-of-Thought (CoT) reasoning is an emergent capability in LLMs (see Supplementary Information 2 for introductions of LLM-related techniques), allowing them to solve complex problems through step-by-step

deduction^{9,16}. Formally, given a prompt \mathbf{Q} , CoT augments the output by generating intermediate reasoning steps $\{c_1, c_2, \dots, c_n\}$ before producing the final answer \mathbf{A} :

$$\text{CoT}(\mathbf{Q}) = \{[c_1, c_2, \dots, c_n], \mathbf{A}\}. \quad (5)$$

To improve GroverGPT+'s reasoning in the symbolic analysis of quantum circuits, we adopt explicit CoT training, where intermediate reasoning chains are included in the supervision training targets. Unlike prior work¹⁴ that directly predicts output probabilities, our approach models the full analysis process as a sequence of logical deductions.

We construct two types of CoT training datasets. The first, CoT Data with Oracle-only Input, includes only Oracle QASM code as input, with CoT reasoning sequences as targets; this design encourages GroverGPT+ to reason directly from oracle structure. The second, CoT Data with Full-circuit Input, includes the full Grover circuit QASM code as input, paired with CoT outputs to enhance accuracy and focus on Oracle extraction. These datasets span various qubit ranges to ensure broad generalization. We detail our CoT training technique in Supplementary Information 9.

Data availability

The data supporting the findings of this study are available on GitHub <https://github.com/mchen644/GroverGPT-plus>.

Code availability

The results are reproducible with the code available on GitHub <https://github.com/mchen644/GroverGPT-plus>.

Received: 11 August 2025; Accepted: 29 January 2026;

Published online: 15 February 2026

References

- Shor, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev.* **41**, 303–332 (1999).
- Grover, L. K. A fast quantum mechanical algorithm for database search. In *Proc. of the 28th Annual ACM Symposium on Theory of Computing* 212–219. Gary L. Miller (eds) (Association for Computing Machinery (ACM), 1996).
- Nielsen, M. A. & Chuang, I. L. *Quantum Computation and Quantum Information* (Cambridge University Press, 2010).
- Cross, A. et al. Openqasm 3: A broader and deeper quantum assembly language. *ACM Trans. Quantum Comput.* **3**, 1–50 (2022).
- Pednault, E., Gunnels, J. A., Nannicini, G., Horesh, L. & Wisnieff, R. Leveraging secondary storage to simulate deep 54-qubit Sycamore circuits. arXiv preprint arXiv:1910.09534 (2019).
- Pan, F. & Zhang, P. Simulation of quantum circuits using the big-batch tensor network method. *Phys. Rev. Lett* **128**, 030501 (2022).
- Chundury, S., Li, J., Suh, I.-S. & Mueller, F. Diaq: efficient state-vector quantum simulation. arXiv preprint arXiv:2405.01250 (2024).
- Liu, A. et al. Deepseek-v3 technical report. arXiv preprint arXiv:2412.19437 (2024).
- Wei, J. et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems* **35**, 24824–24837 (2022).
- Bhattacharyya, A. On a measure of divergence between two statistical populations defined by their probability distribution. *Bull. Calcutta Math. Soc.* **35**, 99–110 (1943).
- Matsumoto, K. Reverse test and quantum analogue of classical fidelity and generalized fidelity. arXiv preprint arXiv:1006.0302 (2010).
- Starke, D. S., Basso, M. L. & Maziero, J. Efficient fidelity estimation: alternative derivation and related applications. *Commun. Theor. Phys.* **76**, 095101 (2024).
- Grattafiori, A. et al. The llama 3 herd of models. arXiv preprint arXiv:2407.21783 (2024).
- Wang, H. et al. Grovergpt: a large language model with 8 billion parameters for quantum searching. arXiv preprint arXiv:2501.00135 (2024).
- Touvron, H. et al. Llama: open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023).
- Wei, J. et al. Emergent abilities of large language models. *Trans Mach Learn. Res.* (2022).

Acknowledgements

M.C., J.C., and J.L. are supported in part by the University of Pittsburgh, School of Computing and Information, Department of Computer Science, Pitt Cyber, Pitt Momentum fund, PQI Community Collaboration Awards, John C. Mascaró Faculty Scholar in Sustainability, NSF award 2535915, Thinking Machines Lab and Cisco Research. This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725. P.L. and T.C. are partially supported by the Amazon Research Award, the UNC Accelerating AI Awards, the NAIRR Pilot Award, the OpenAI Researcher Access Award, and the Gemma Academic Program GCP Credit Award.

Author contributions

J.L. and T.C. proposed the study of a large language model in symbolic analysis of Grover's algorithm. M.C. proposed the methods of quantum-native tokenization and adopted Chain-of-Thought technique. M.C. designed and performed the experiments, with support from J.C., P.L., and H.W. under the supervision of J.L. and T.C. M.C., J.C., and P.L. wrote the manuscript, with inputs and contributions from all authors.

Competing interests

J.L. is an associate editor of *npj Quantum Information*, but was not involved in the editorial review of, or the decision to publish this article. All other authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s41534-026-01195-1>.

Correspondence and requests for materials should be addressed to Tianlong Chen or Junyu Liu.

Reprints and permissions information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2026