QuRAFT: Enhancing Quantum Algorithm Design by Visual Linking between Mathematical Concepts and Quantum Circuits

Zhen Wen, Jieyi Chen, Yao Lu, Siwei Tan, Jianwei Yin, Minfeng Zhu, and Wei Chen

Abstract—The emergence of quantum computers heralds a new frontier in computational power, empowering quantum algorithms to address challenges that defy classical computation. However, the design of quantum algorithms is challenging as it largely requires the manual efforts of quantum experts to transit mathematical expressions to quantum circuit diagrams. To ease this process, particularly for prototyping, educational, and modular design workflows, we propose to bridge the textual and visual contexts between mathematics and quantum circuits through visual linking and transitions. We contribute a design space for quantum algorithm design, focusing on the textual and visual elements, interactions, and design patterns throughout the quantum algorithm design process. Informed by the design space, we introduce QuRAFT, a visual interface that facilitates a seamless transition from abstract mathematical expressions to concrete quantum circuits. QuRAFT incorporates a suite of eight integrated visual and interaction designs tailored to support users in the formulation, implementation, and validation process of the quantum algorithm design. Through two detailed case studies and a user evaluation, this paper demonstrates the effectiveness of QuRAFT. Feedback from quantum computing experts highlights the practical utility of QuRAFT in algorithm design and provides valuable implications for future advancements in visualization and interaction design within the quantum computing domain. A free copy of this paper and all supplemental materials are available at https://osf.io/xvzgh/.

Index Terms—Quantum computing, quantum circuit, visual linking, design space, interaction design, design tool

1 Introduction

Quantum algorithms take advantage of quantum mechanics (e.g., superposition, entanglement, and quantum interference) to achieve unprecedented computational capabilities [1]. For example, Shor's algorithm [2] exploits these principles to efficiently factor large integers. As the development of actual quantum computers progresses [3], various domains such as computational chemistry [4], machine learning [5], and financial modeling [6] stand to gain significantly from advancements in quantum algorithms. However, the design of quantum algorithms remains challenging due to the conceptual gap between theoretical quantum mechanics and practical computation models [7]. This gap manifests in the disconnect between mathematical formulations and quantum circuit implementations (Fig. 1-A, B), making the translation from theory to practice arduous, error-prone, and heavily reliant on expert intuition.

While existing quantum development tools provide valuable functionality, they follow two dominant paradigms that present challenges for seamless theory-to-practice translation. *Programming tools* like Qiskit [8] and Cirq [9] provide precise control over quantum circuits at the code level, yet demand cognitive reconstruction—forcing users to mentally map symbolic equations into syntactic structures. Meanwhile, *visual tools* like Quirk [10] and IBM Composer [11] offer intuitive circuit manipulation through graphical interfaces, but their tile-based metaphors reduce quantum operations to isolated gate placements, failing to capture the underlying mathematical continuity that governs quantum circuits. The significant modality differences between textual mathematics and visual quantum circuits force users into a high-cognitive-load loop of manual translation and mental context-switching, which not only stifles creativity but also increases the likelihood of conceptual errors. This challenge is particularly acute in foundational stages of the scientific and educational workflow. While designing novel quantum algorithms requires high-level creative thinking, a critical bottleneck emerges when translating well-defined mathematical constructs into circuits. Students learning quantum computing must decipher how mathematical concepts like unitary matrices or the Quantum Fourier Transform manifest as gates. Researchers face similar challenges when reproducing paper results or integrating known modules into larger algorithms. These ubiquitous scenarios demand tools that streamline the translation from mathematical notation to circuit components, fostering a more direct path from theory to implementation.

To address these challenges, we propose to integrate visualization and interactive techniques as key strategies in navigating the complexities of quantum algorithm design. Previous studies [12], [13] have demonstrated the effectiveness of visual and interactive connections on enhancing user experience and efficiency in creative authoring scenarios. This work attempts to explore the potential of these advancements within the realm of quantum algorithm

Zhen Wen, Jieyi Chen, Yao Lu, and Wei Chen are with the State Key Lab of CAD&CG, Zhejiang University. Email: {wenzhen, chenjieyi_juraws, 3200104705, chenvis}@zju.edu.cn.
Siwei Tan and Jianwei Yin are with the Advanced Computing and Emerging Service Lab, Zhejiang University. E-mail: siweitan@zju.edu.cn, zjuyjw@cs.zju.edu.cn.
Minfeng Zhu is with Zhejiang University. E-mail: minfeng_zhu@zju.edu.cn.
Wei Chen is the corresponding author.
Manuscript received xx.xx.20xx; revised xx.xx.20xx.

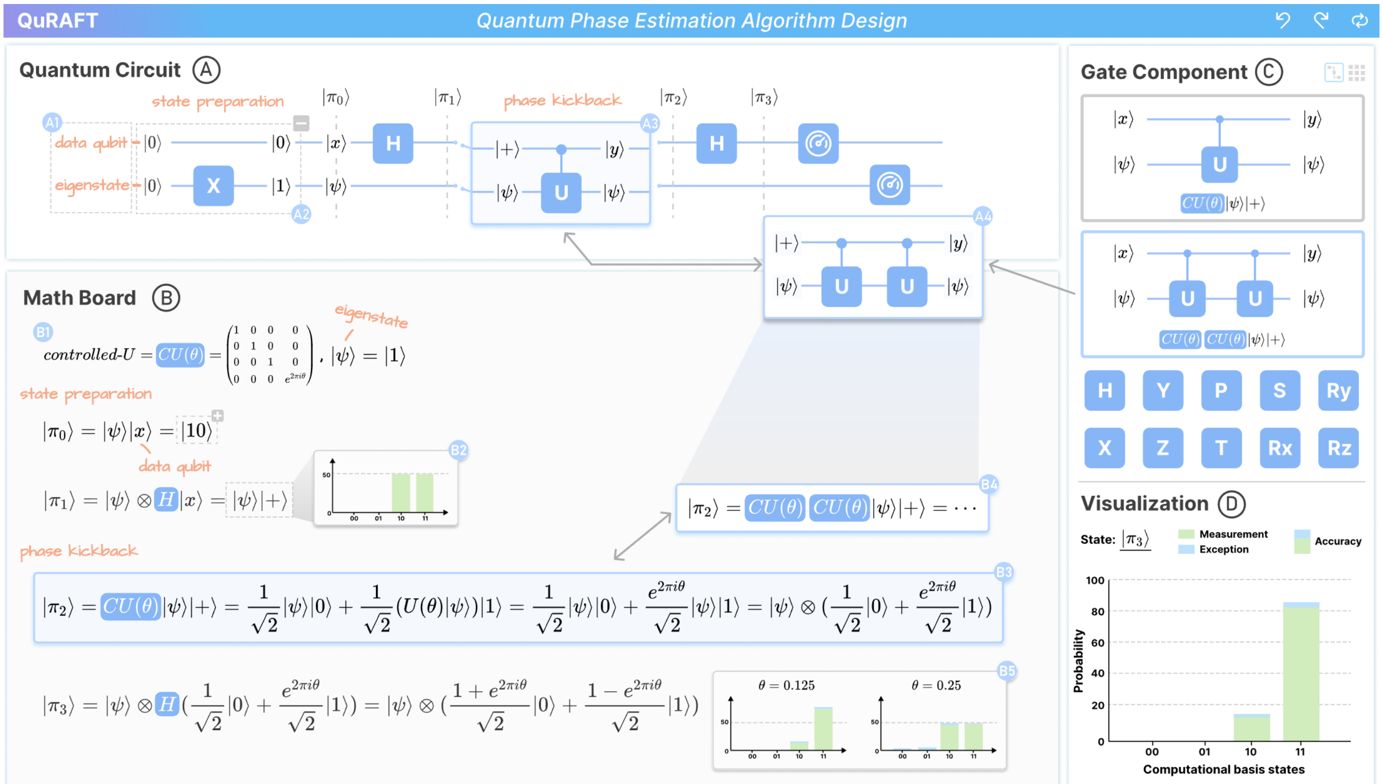


Fig. 1. The QuRAFT interface integrates eight visual and interactive design techniques we have proposed for quantum algorithms.The Quantum Circuit View (A) displays the circuit implementation of the algorithm, and the Math Board (B) presents the corresponding mathematical formulation of the algorithm. The Gate Component (C) and Visualization (D) View provide useful tools for algorithm construction and verification.

design. Working closely with domain experts, we contribute a structured design space that categorizes the components of quantum algorithm design into four dimensions: Representation Type, Quantum Entity, Interaction Complexity, and Action. Through this framework, we identify prevalent design patterns that offer valuable insights into the design of QuRAFT (pronounced similarly to "Craft"), an interface that incorporates eight design techniques within a user-friendly visual and interactive environment. QuRAFT is developed involving user-centered design principles and iterative development, aimed at delivering an efficient and intuitive tool for those researching quantum algorithms. Our research method is grounded in close collaboration with domain experts, ensuring that our proposed design techniques address real-world needs and enhance the quantum algorithm design process. Two case studies exemplify the real-world application of our techniques in quantum algorithm design scenarios. Furthermore, two user evaluations conducted with 16 experts demonstrate the effectiveness of our design approach, highlighting its streamlining of translation from math concepts to circuits and reduction of cognitive load.

In conclusion, the contributions of this paper are summarized as follows: (1) a design space that categorizes the quantum algorithm design process, covering both textual and visual representation spaces, (2) the QuRAFT prototype, incorporating the design of eight visual and interactive techniques that bridge the textual and visual contexts in the quantum algorithm design process, and (3) two case studies and two user evaluations of our designs, providing implications for future quantum development tool design.

2 Related Work

In this section, we discuss the related work in quantum algorithm design, quantum algorithm visualizations, and the techniques for bridging textual and visual representations.

2.1 Quantum Algorithm Design

The recent advancement of quantum computers has marked a significant milestone for quantum algorithms [3], transitioning their design from purely theoretical constructs to an integration of theoretical foundations and practical implementation [14], [15]. This evolution has sparked a wave of innovative design techniques and computational tools tailored for this new era of quantum algorithm design.

The design of a quantum algorithm typically involves the creative combination of existing algorithmic paradigms, such as quantum Fourier transform [16] and variational quantum eigenvalue solver [17]. Quantum researchers constantly explore new paradigms leveraging the power of mathematics [18], [19] and machine learning [20], [21]. One of the key challenges in algorithm design is determining how to effectively combine existing paradigms to construct effective quantum circuit implementations. Kang et al. [22] address this challenge by introducing the concept of modular synthesis, which offers a structured approach to constructing quantum circuits from modular components. Maslov et al. [23] utilize a template-based method to simplify quantum circuits. KetGPT [24] presents an innovative extension to existing quantum algorithms through the use of a generative model. While these methodologies primarily target the optimal algorithmic performance, there remains a notable gap in research concerning the enhancement of the quantum algorithm design experience. This gap highlights the need for innovative approaches that improve the workflow of designing quantum algorithms, making it more user-friendly and accessible for designers navigating the intricacies of quantum computing.

The development and simulation of quantum algorithms are supported by a growing ecosystem of computational tools and programming languages, such as Qiskit [8], Cirq [9], and Pennylane [25]. These tools offer powerful, low-level control for constructing, simulating, and executing circuits on quantum hardware. Their primary goal is to optimize performance and lower the barrier to programmatic circuit construction. However, these tools enforce a fragmented workflow that forces users into a high-cost loop of context-switching between the math, codes, and circuits. Our work seeks to enhance the user experience from a new perspective, focusing on easing the transition from mathematical formulations to quantum circuit representations.

2.2 Quantum Algorithm Visualizations

The mechanism of quantum algorithms is intricate and non-intuitive, posing considerable challenges for researchers in terms of understanding and development [26]. To mitigate these challenges, visualization techniques have been increasingly adopted [27], for instance by offering graphical representations of quantum states that make abstract concepts more tangible. Bloch Sphere [28] is a 3D geometric representation widely used for presenting quantum states. Following, several designs [29]–[32] are proposed to extend the expression and capability of Bloch Sphere. Ruan et al. [33] further integrate the characteristics of quantum entanglement and superposition into their visual design. To interpret the procedure of quantum algorithms, researchers visualize the evolution of quantum states coupled with quantum circuits [34], [35], revealing how quantum states change and interact throughout the computation process. Additionally, Wen et al. [36] augment the understanding of quantum circuits by embedding semantic information within circuit diagrams. VIOLET [37] introduces a visual analytics approach that explains the intricate components of quantum neural networks, including the input encoder, ansatz training, and output measurement.

Given that the formulation of quantum algorithms heavily relies on mathematical constructs, visualizing mathematical elements can significantly enhance the comprehension of such algorithms. Head et al. [38] conduct an extensive design study on practical augmentations of math formulas, providing guidance on how to create visual and interactive mathematical formulas to improve readability and comprehension. Many tools have been developed to facilitate the sketching and design processes in technical fields through math augmentation [39]–[41], enhancing the ease and efficiency of working with complex mathematical concepts. However, there are few studies on the application of these advancements specifically for quantum computing. We aim to integrate interactive math augmentations with quantum circuit visualizations, supporting researchers in quantum algorithm design with greater clarity and efficiency.

2.3 Bridging Textual and Visual Representations

The connection between text and visual representations plays a crucial role in enhancing the interpretation and communication of complex information [13]. This interplay has been extensively explored by communities of visualization and human-computer interaction, focusing on the methodologies for visualizing and leveraging this connection [42].

Various approaches integrate text with visual elements to clarify key information. Kong et al. [43] extract references between text and charts using crowdsourcing to highlight visual marks related to text. Badam et al. [44] design contextual visualizations connecting text with tables for improved document readability. Lai et al. [45] introduce automatic alignment of text and visualizations with animated annotations. Sporthesia [46] enhances sports video viewing by analyzing narrative comments and augmenting identified entities directly onto video. Recent research further leverages interactive linking to augment user engagement. Studies [47], [48] show that interactive linking between text and visualizations significantly increases user engagement in storytelling. Kori [49] introduces a mixed-initiative approach for interactive synthesis of text and charts within data documents, enabling a more engaging exploration of data narratives. VizFlow [50] leverages text-chart connections to facilitate the authoring of data-driven articles, while CrossData [12] enhances the process of creating data documents by exploiting the relationships between text and data. Charagraph [51] proposes a method for the interactive generation of charts, allowing for real-time annotations in documents rich with data. These advancements highlight the role of interactive linking in not only clarifying complex information but also in engaging users more deeply with the content creation and exploration.

In quantum computing, quantum circuits visually represent complex math formulations, yet there is a lack of clear guidelines for bridging these textual and visual representations. We present a design study that addresses this gap, aiming to enhance the quantum algorithm design process by improving user comprehension and engagement.

3 Methods

The goal of this work is to bridge the gap between the textual and visual contexts in the quantum algorithm design process through visualization and interaction techniques. We worked with two experienced quantum researchers for design studies. They are both currently working in a university quantum computing lab and has extensive experience in the optimization and design of quantum algorithms. We primarily characterized the problem through a comprehensive literature review and iterative discussions with the domain expert. Consequently, we contributed a design space that delineates four critical design dimensions and identifies several essential design patterns (Sect. 4).

Informed by the above studies, we crafted a series of visual and interactive techniques for quantum algorithm design and iteratively developed a prototype, in collaboration with the quantum experts and visualization experts. The quantum experts outlined four design goals for supporting their practical workflow, which informed the subsequent technique development (Sect. 5). They also provided

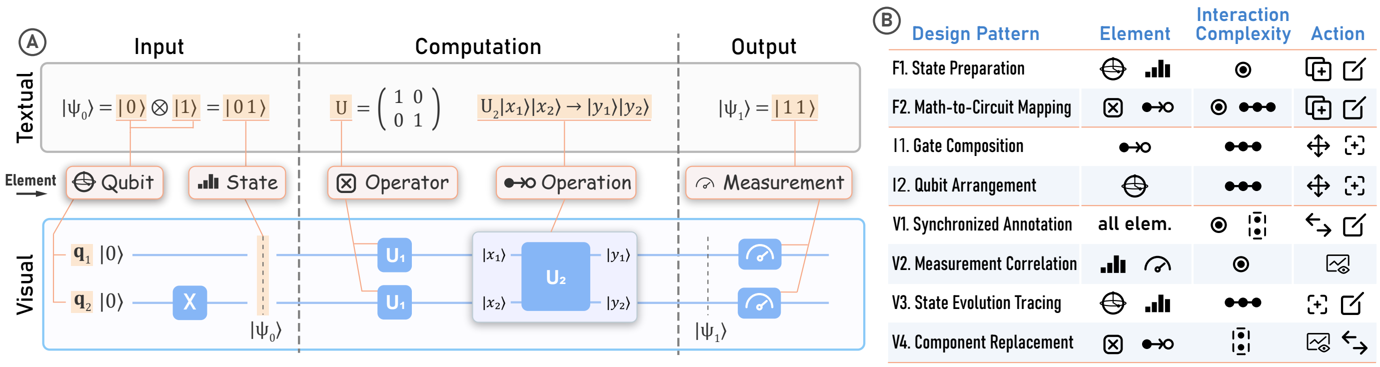


Fig. 2. The illustration of our design space. (A) We characterize the fundamental elements of quantum algorithm design in both textual and visual representation space. (B) A list of design patterns and their relations to our design dimensions, which also informs the techniques proposed in Sect. 5.

feedback on the visual design and user experience of our developed techniques throughout the process. Over three months, weekly online meetings covered everything from initial design concepts to development stages, with each session lasting 30 to 60 minutes. Finally, we evaluated the prototype system with 10 other quantum experts from the quantum lab, refining our prototype based on their feedback and discussing significant design implications in Sect. 8.

4 General Concepts and Design Space

In this section, we introduce key concepts of quantum algorithms and present a characterization of the quantum algorithm design process.

4.1 Concepts of Quantum Algorithms

A quantum algorithm is characterized by its components of *input*, *computation*, and *output*, similar to the structure of classical algorithms. However, the units of information, operations, and executions in quantum algorithms diverge from those employed in classical computational algorithms. Consequently, the workflow of quantum algorithm design is distinct from that of classical algorithm design.

Framework. The three fundamental components of quantum algorithms are significantly different from those in classical computing [7]:

- *Input - Quantum States.* In classical algorithms, inputs are typically binary data represented by bits, which can be either 0 or 1. In contrast, the inputs to a quantum algorithm are quantum states, which are represented by qubits. Qubits differ from classical bits in that they can exist in a superposition of states, meaning a qubit can represent both 0 and 1 simultaneously, denoted as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where α and β are complex probability amplitudes. This property allows quantum algorithms to process a vast amount of possibilities concurrently.
- *Computation - Quantum Operations.* Quantum computation involves the manipulation of quantum states using quantum gates, which are the building blocks of quantum circuits. These gates operate on qubits in ways that cannot be replicated by classical logic gates. Quantum gates operate on the principles of unitarity, ensuring that operations are reversible and preserve the total probability of the quantum state. Quantum operations can entangle qubits, creating a correlation between them that is a key resource for quantum algorithms, providing an advantage over classical algorithms in certain tasks.
- *Output - Measurements.* The output of a quantum algorithm is derived from measurements made on the quantum state after computation. Unlike classical deterministic outputs, quantum measurements are probabilistic, owing to the collapse of the quantum state upon observation. The result of a measurement is typically a set of classical bits, which are obtained by measuring the qubits' states. Due to the probabilistic nature of quantum measurements, quantum algorithms often require repeated runs to gather statistical data and deduce the correct output.

Workflow. The intricacies of these components necessitate a different approach to designing quantum algorithms compared to classical algorithms. Drawing from literature reviews [14], [15], [26] and consultations with domain experts, we have identified three core stages in the design process of quantum algorithms: (1) *Formulation*, which involves the initial ideation and mathematical formulation aimed at solving a specific quantum problem; (2) *Implementation*, where this formulation is transformed into a quantum circuit using a sequence of quantum gates; and (3) *Validation*, where the algorithm undergoes extensive testing in iterative cycles to verify that its performance adheres to theoretical models and achieves the desired benchmarks. It is notable that our work deliberately excludes the examination of these algorithms in the context of hardware testing and debugging, focusing solely on the theoretical and simulation-based aspects of quantum algorithm development.

4.2 Fundamental Design Elements

The design of quantum algorithms is rooted in several key elements, which can be systematically categorized with **Representation Type** and **Quantum Entity**.

Dimension I: Representation Type. Quantum algorithms are presented through diverse representation types. Typically, the algorithms are precisely formulated using mathematical symbols. However, in practice, these theoretical constructs need to be transformed into the more tangible form of quantum circuits, which offer a more intuitive grasp of their operational sequences. In light of their modalities, we categorize the representations into two types:

- *Textual Representations* involve the use of mathematical notation and algorithmic language to describe quantum states, operations, and the behavior of quantum systems, which are vital for conveying the precise, formal details of quantum algorithms.
- *Visual Representations* refer to quantum circuit diagrams which provide an intuitive visual narrative of how quantum information is processed within the algorithm. They also serve as a blueprint for implementation on actual quantum computing hardware, outlining the operational sequence required for execution.

Dimension II: Quantum Entity. The basic elements of quantum algorithms encompass various quantum entities, including qubits, unitary transformations, measurements, etc. [14], each with a specific role in the computational process. The design of quantum algorithms involves the construction of a quantum system utilizing these entities. Some entities bear different names in mathematical and quantum circuit contexts. With consistency in mind, we uniformly categorize them as follows:

- *Qubit* is the quantum analogue of the classical bit and serves as the basic unit of quantum information, which is textually denoted as $|x\rangle$. Sometimes, it may represent one component within the basis states, articulated as $|x_i\rangle$ within the larger quantum state $|x_1 x_2 \cdots x_n\rangle$. In the quantum circuit, a qubit is visualized as a horizontal line that extends throughout the circuit, denoting its presence and involvement in quantum operations and transformations.
- *State* encapsulates the information and condition of single or multiple qubits. Textually, a quantum state is represented using a state vector, denoted as $|\psi\rangle$. A state of multiple qubtis is composed using tensor product as $|\psi\rangle = |x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle = |x_1 x_2 \cdots x_n\rangle$. On this basis, any state of a quantum system can be expressed as a linear combination of basis states, such as $|\psi\rangle = \sum_i c_i |i\rangle$, where the coefficients c_i represents complex probability amplitudes, and $|i\rangle$ signifies the basis states of the system. For instance, a two qubit system has four basis states denoted $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$. Visually, the quantum states are intimately related to qubits. The state of a quantum system at a specific time step is represented as one or a collection of qubits. These states undergo transformation after the application of operations on the qubits, reflecting the dynamic nature of quantum information processing.
- *Operators* are mathematical entities that modify quantum states and are strictly required to be unitary matrices, following quantum mechanics constraints. In quantum circuits, these operators are represented as quantum gates placed on qubits. A single unitary matrix can be represented by either an individual gate or a combination of gates, which are typically visualized as square blocks in the circuit diagrams. Additionally, a gate can be connected to other qubits to indicate control mechanisms. For instance, a controlled-NOT gate flips the target qubit's state only if the control qubit is in the state $|1\rangle$.
- *Operation* describes the evolution of a quantum state from its input to its output, facilitated by a sequence of operators. It is mathematically represented by the application of unitary operators to quantum states. For a single operator, this is expressed as $U|\psi\rangle = |\psi'\rangle$, where U is a unitary matrix, $|\psi\rangle$ is the input state, and $|\psi'\rangle$ is the output state. For a sequence of operations, the evolution is described by the composition of unitary operators: $U_n U_{n-1} \cdots U_2 U_1 |\psi\rangle = |\psi'\rangle$, where the operators are applied from right to left in chronological order.
- *Measurement* indicates the end of quantum operations and the transition from quantum states to classical information. In theoretical formulations, researchers usually compute expected values or probability distribution for quantum states to predict the outcomes of measurements. In practice, measurement gates are positioned at the end of circuits to observe the outcomes of executed operations, which are then compared with theoretical expectations.

Figure 2-A illustrates the intersection of Representation Type and Quantum Entity within the design space, offering concrete examples of how each quantum entity is depicted across both textual and visual representations. It serves as a visual guide that connects the abstract math descriptions with their corresponding circuit representations, demonstrating how each entity is manifested in the two representation types within the quantum algorithm design space.

4.3 Interactive Design Process

The creation of quantum algorithms is not only a theoretical exercise but also an interactive process that requires dynamic engagement with the design elements. To encapsulate this interactivity, we introduce two design dimensions, **Interaction Complexity** and **Action**, which outline the levels of interaction complexity and the range of actions that designers can employ within the design space.

Dimension III: Interaction Complexity. This dimension illustrates the levels of complexity encountered when designers interact with elements within the design environment. It highlights the varying degrees of engagement required, from simple to complex as follows:

- *Atomic Interaction* focuses on interactions with individual quantum entities, such as a particular qubit or a single quantum gate, emphasizing direct and straightforward engagement with single elements. This level is characterized by actions performed on isolated design elements, requiring attention to detail and precision.
- *Collective Interaction* involves handling multiple quantum entities as a unified group, where the primary concern is the management and coordination of actions across the collective. While relationships between elements are not the central focus, the designer must navigate the complexity of dealing with several elements simultaneously.
- *Relational Interaction* represents the highest level of interaction complexity, where designers must consider and manage the intricate relationships and dependencies between multiple quantum entities. This necessitates a deep understanding of the dynamic interplay among these elements, as well as the ability to predict and respond to the cascading effects of actions within this interconnected system, such as a group of dependent operations.

Dimension IV: Action. This dimension captures the range of operations or manipulations that designers can perform within the interactive design process. These actions are fundamental to shaping and refining quantum algorithms, offering a toolkit for creativity and exploration.

- *Generating*: The creation of new elements in design space, such as adding new qubits or constructing gate sequences, laying the foundation for algorithm design.
- *Editing*: Modifying existing elements, allowing for refinement and optimization of algorithm components, such as setting the state of a qubit or changing the parameters of a quantum gate.
- *Moving*: Reorganizing elements within the design environment, which can influence the algorithm's structure and functionality.
- *Focusing*: Narrowing attention to specific elements or areas, enabling detailed examination and adjustment, often to simplify complexity or to enhance precision.
- *Comparing*: Evaluating elements or configurations against others, essential for decision-making and selection of optimal solutions.
- *External Visualization*: Employing visual aids or representations outside the immediate design environment to enhance understanding or convey complex information effectively, providing alternative insights or perspectives on the quantum state.

4.4 Cross-Modal Design Patterns

The interplay between textual and visual representations is crucial in the design process of quantum algorithms. We identify some notable design patterns that involve both textual and visual elements from the workflow of domain experts. Figure 2-B provides an overview of the design patterns and their associated parts in our design space.

Formulation (F1–F2). In the Formulation phase, design patterns focus on laying the groundwork for the algorithm. This involves defining the initial quantum states and establishing a clear correspondence between the mathematical underpinnings of quantum operations and their visual representations in circuit diagrams. The patterns in this phase ensure that the foundational elements of the algorithm are accurately and effectively represented in both modalities.

Implementation (I1–I2). During the Implementation phase, the emphasis shifts to the construction of the quantum circuit. The design patterns applied in this phase address the organization and arrangement of quantum gates and qubits within the circuit. The goal is to reflect the algorithm logic in a manner that is both visually coherent and theoretically sound, facilitating the practical realization of the algorithm.

Validation (V1–V4). The Validation phase is where the design patterns aid in the examination and confirmation of the algorithm's correctness. This phase involves a thorough cross-checking of the consistency across textual and visual representations, as well as an analysis of measurement outcomes and state evolution. The patterns employed here are crucial for ensuring that the algorithm behaves as expected and is ready for deployment or further research.

5 QuRAFT

Informed by the design space, we design QuRAFT, a proof-of-concept prototype that enhances quantum algorithm design process with a series of delicately designed visual and interactive techniques.

5.1 Design Goals

We distill a set of design goals to inform the development of QuRAFT, derived from prior design studies and feedback from quantum computing experts reflecting on their experiences in quantum algorithm design. These goals are continuously refined throughout our iterative development process, incorporating ongoing expert feedback.

G1. Transform Mathematical Concepts into Circuit Routines. Feedback from quantum computing experts indicates that the design of a quantum algorithm typically starts with a mathematical formulation, followed by the translation of this formulation into an equivalent quantum circuit. However, our design space reveals the complexity and diversity of elements in a quantum algorithm's formulation. Manually converting these math concepts into circuit routines is tedious and prone to errors. Therefore, QuRAFT aims to automate the detection and transformation of mathematical constructs into quantum circuit components, thereby streamlining the design process and reducing errors.

G2. Compose Circuit Components Flexibly and Logically. The implementation of quantum algorithms involves the careful composition of various components in a manner that faithfully reflects their logical relationships defined in math formulation. However, existing tools for creating quantum circuits and methods for automatic circuit synthesis often do not offer explicit rationales for the composition of circuit components. Consequently, QuRAFT is developed with the goal of extracting and leveraging the latent relationships among these components, thus provides a flexible and intuitive interface that allows users to logically and effortlessly compose and rearrange circuit components.

G3. Review and Refine Algorithm Details. Upon the implementation of quantum circuits, designers need to verify the correctness of algorithms. However, the validation usually entails frequent context-switching between mathematical formulations, quantum circuits, and execution results. Thus, QuRAFT attempts to introduce capabilities that facilitate the simultaneous review and refinement of algorithm details, integrating these various contexts seamlessly.

G4. Maintain Consistency Between the Formulation and Implementation. As a process involving two closely related forms of content, the design of quantum algorithms must ensure that the initial mathematical formulation and the final quantum circuit implementation are consistently aligned. However, despite their close relationship, the iterative nature of design often leads to misalignments between these two critical content. As such, QuRAFT should maintain dynamic association between them throughout the entire quantum algorithm design process.

In the following three sections, we introduce eight design techniques in QuRAFT to address these design goals and how they support the algorithm design process. Specifically, the formulation and implementation process is supported by State Preparation, Math-to-Circuit Mapping, Gate Composition, and Qubit Arrangement techniques (Fig. 3), while the validation process is facilitated by Synchronized Annotation, Measurement Correlation, State Evolution Tracing, and Component Replacement techniques (Fig. 4).

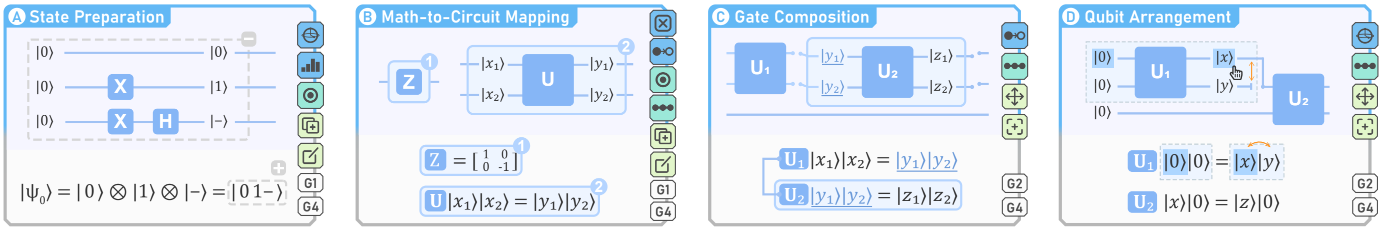


Fig. 3. Design techniques for formulation and implementation. (A) State Preparation simplifies the initialization of quantum states. (B) Math-to-Circuit Mapping translates mathematical expressions of quantum operations directly into visual circuit representations. (C) Gate Composition enables the intuitive and interactive assembly of quantum gates. (D) Qubit Arrangement offers flexible manipulation and organization of qubits in the circuit.

5.2 Math Detection and Transformation

To incorporate G1, we introduce a mechanism that detects and maps mathematical constructs to their circuit equivalents. Our system employs a rule-based parser that uses regular expressions to recognize common quantum notations in LaTeX strings, such as state definitions, unitary matrices, and state transformations. Upon detection, these expressions are translated into structured circuit components. For instance, a matrix definition is encapsulated as a custom gate. To ensure correctness, the system automatically verifies the unitarity of any user-defined matrix, ensuring that only valid quantum operations are created. The following two techniques build upon this core functionality.

State Preparation. At the beginning of a quantum algorithm, there is always a specification for the initial state of the quantum system serving as the algorithm's input. QuRAFT detects equations that specify variable values, and treats them as potential specifications for the initial state of a quantum algorithm, such as $|\psi\rangle = |001\rangle$. The system allows users to input these values textually and automatically incorporates the requisite circuit components that generate these values into the quantum circuit. Furthermore, users are afforded the flexibility to interact with these values—clicking on them toggles the expansion or collapse of detailed component information, or to edit them directly within both the textual and visual representations, as illustrated in Fig. 3-A.

Math-to-Circuit Mapping. The core computation of quantum algorithms is formed by unitary operators and their composite operations. QuRAFT detects two primary types of mathematical formulations that correspond to these computational elements.

- *Unitary Matrix Definition.* A unitary matrix with dimensions $2^n \times 2^n$ precisely specifies the type of operator that acts on n qubits. For example, Fig. 3-B1 depicts the Z-gate, which is a representation for an operator acting on a single qubit. In QuRAFT, each matrix definition equation is transformed into an isolated (atomic) gate within the quantum circuit, where the specific matrix values are abstracted away from the visual representation. These matrices are typically fundamental operators within a quantum algorithm, serving as the building blocks for more complex quantum operations.
- *Quantum State Transition.* This formulation encapsulates the transition of a quantum state resulting from one or a combination of unitary operators. QuRAFT visually maps three essential parts of this formulation onto circuit representations: unitary operators, input state, output state. First, we convert each unitary operator into a distinct gate and visually arrange these gates in accordance with their combination in the formula. The input and output states are then concatenated to the respective ends of this visual sequence, illustrating the flow of the quantum state through the operation. Figure 3-B2 illustrates our concurrent design for both textual and visual representations. Specifically, quantum gates are visualized as blue rectangles, consistent with the unitary operators highlighted in the formula, while input and output states are annotated on their corresponding qubits. User selection on a formula highlights its quantum circuit counterpart simultaneously. Additionally, any user edits to the formula automatically update this mapping. This design ensures coherence and clarity in the depiction of quantum state transitions.

5.3 Quantum Circuit Implementation

We introduce two techniques that allow users to flexibly integrate previously generated components into a complete quantum circuit (G2).

Gate Composition. To implement accurate quantum circuits, experts are eager to compose the circuit components with reliable supports of mathematical formulations. Our system infers the logic relationships between the components and provides visual cues to facilitate gate composition. The key idea of our design is to utilize the logical relationship inherent in the sequence of quantum operations, as described by textual formulas. For instance, the sequential formulas $U_1|x\rangle = |y\rangle$ and $U_2|y\rangle = |z\rangle$ illustrate a state transition from $|x\rangle$ to $|y\rangle$ and then to $|z\rangle$. These formulas suggest a sequential application of U_1 followed by U_2, exemplifying the logical flow and composition of quantum gates within a circuit. To support users in accurately assembling circuit components, QuRAFT maps the input state of user-selected component to its preceding outputs, providing highlighted indicators for easy identification. Users are thus enabled to efficiently arrange quantum gates by simply dragging them into position. Additionally, by creating visual connections to the corresponding formulas, we ensure that the logical integrity is preserved as gates are assembled, fostering a coherent and consistent circuit design process (see Fig. 3-C).

Qubit Arrangement. The quantum algorithms are typically formulated without the layout considerations of their practical implementations. Consequently, experts often need to rearrange qubits to reduce cross lines and optimize circuit layout for efficiency and readability. QuRAFT introduces

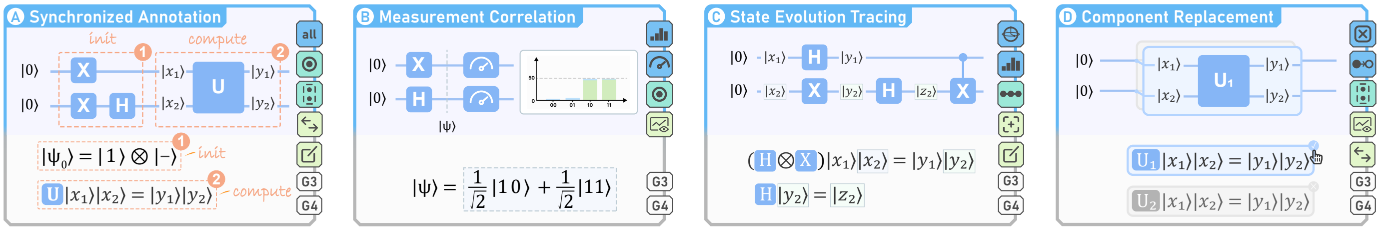


Fig. 4. Design techniques for Validation. (A) Synchronized Annotation allows for annotating on math formulas and quantum circuits simultaneously. (B) Measurement Correlation compares expected outcomes with actual measurements. (C) State Evolution Tracing visualizes the progression of quantum states through the algorithm. (D) Component Replacement facilitates the easy substitution of algorithm components.

interactive features that enable users to manually adjust the positioning of qubits to enhance the circuit layout. Users have the flexibility to select and rearrange the sequence of qubits (see Fig. 3-D). This arrangement is consistently represented across both the textual and visual modalities, ensuring coherence in the design process.

5.4 Interactive Review and Modification

We design four techniques for interactive review and modification on implemented quantum circuits and formulations simultaneously (G3).

Synchronized Annotation. While examining mathematical formulations or quantum circuits, users often need to create notes or sketches on their essential parts. QuRAFT enables users to annotate both textual and visual representations simultaneously (Fig. 4-A). This technique extends the similar concept in software debugging tools to the unique cross-modal context of quantum algorithm design. It establishes a persistent link between abstract mathematical entities and their concrete graphical counterparts, implementing a form of visual linking across linked views. As a result, an annotation created on a formula is automatically propagated to the corresponding gate, and vice versa, ensuring conceptual consistency and reducing the need for manual cross-referencing between the two views. QuRAFT establishes connections between these entities based on the transformation relationships involved in prior formulation and implementation processes. Consequently, users are empowered to highlight annotations in both contexts, thereby verifying the consistency of mathematics and circuits.

Measurement Correlation. To verify the accuracy of quantum algorithms, experts need to compare the actual measurement outcomes from the circuit with their theoretical expectations. An external visualization often acts as a bridge, connecting the expected mathematical outcomes with the observed probability distributions. QuRAFT runs the circuit on a local Qiskit simulator [52], then converts both the expected outcomes, as articulated in the mathematical formulations, and the actual measurements from the circuit into comparative histogram visualizations. When a specific state is selected within the formulation, QuRAFT locates its corresponding position in the circuit for measurement. The outcomes are then depicted in Visualization View (Fig. 1-D). If there is a predefined mathematical expectation for that state, it is concurrently visualized in the same view, facilitating a direct comparison between theoretical and practical outcomes (Fig. 4-B).

In developing the Measurement Correlation technique, our iterative design process evolved through several stages to enhance the visualization of comparing actual measurements against theoretical predictions. Our initial design displayed only the actual measurement distributions (Fig. 5-A), but experts pointed out the difficulty in comparing these directly with mathematical values. To improve, we introduced juxtapose double bars for theoretical and actual outcomes (Fig. 5-B). Although this improved comparison, it resulted in visual clutter due to an exponential increase in bars with the addition of qubits. Continuing to refine, we experimented with overlapped bars (Fig. 5-C) to conserve space, yet this approach led to confusion due to a third color emerging from the overlap of two distributions, complicating the comparison task. The final iteration introduced each bar encoded with a similarity metric (Fig. 5-D), offering an immediate, quantifiable measure of how closely the actual measurements align with expectations, thus finalizing a design that balances clarity, detail, and analytic level.

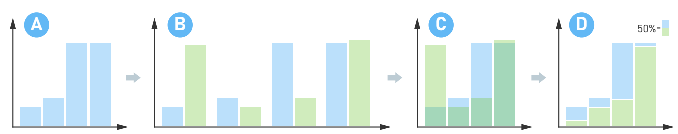


Fig. 5. Iterative visual design for measurement outcomes from (A) to (D).

State Evolution Tracing. Understanding the progression of qubit states throughout an algorithm is essential for experts to fully comprehend the algorithm's workings. Unlike classical programming, where variable values are static until modified, quantum states evolve interdependently in textual formulas and quantum circuits, creating dynamic consistency challenges. To address this, we trace the evolution of quantum states both visually and textually (Fig. 4-C), ensuring their accurate progression is easily verified and understood. QuRAFT implements a brushing and linking interaction across two linked views. When a user selects a qubit state in the quantum circuit or its mathematical formulation, the entire evolution of that state is highlighted, providing a clear depiction of its transformations. Following expert suggestions, we also incorporate functionality that allows users to modify the transformations within its evolution, enhancing the system's interactivity and user control.

Component Replacement. In the iterative design of quantum algorithms, experts often experiment with various composition of quantum operations to observe their effects. To support this exploratory process, we develop a technique that enables users to seamlessly switch between different

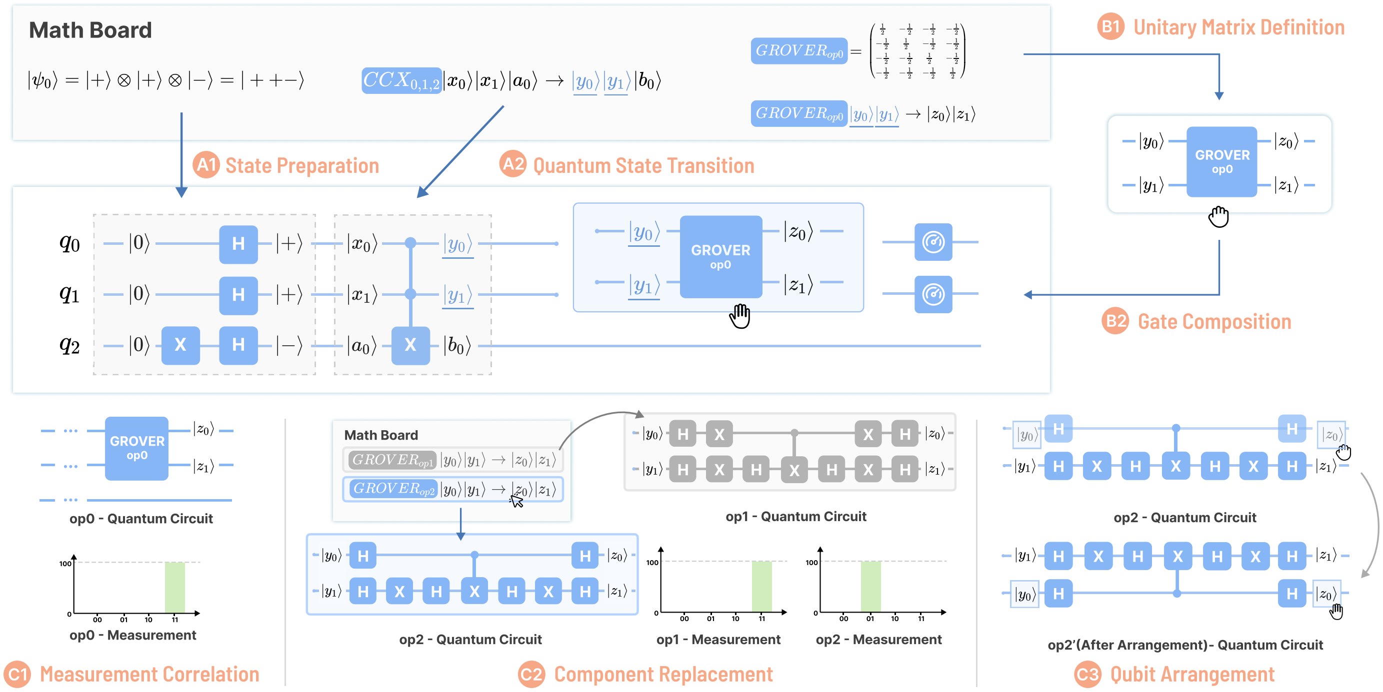


Fig. 6. The use case of using QuRAFT to create Grover's Algorithm. The mathematical formulation is translated into quantum circuit (A1, A2, B1). Users can interactively compose the circuit component (B2), verify the accuracy of outcomes(C1), and explore creative design (C2, C3).

component configurations, maintaining coherence between mathematical formulations and circuit implementations. Users have the option to propose alternative formulations and select which one to activate at any given time. When assembling the corresponding circuit components, they can freely activate alternative configurations (Fig. 4-D). This feature ensures that changes to the configuration are contextually integrated and can be smoothly executed, enhancing the flexibility and depth of algorithm design exploration.

6 Use Cases

To demonstrate how QuRAFT supports the quantum algorithm design process in practice, we present two use cases in collaboration with domain experts. The cases illustrate two fundamental and recurring workflows: (1) implementing a well-known algorithm from its mathematical specification, a common task for researchers reproducing prior work, and (2) iteratively exploring and modifying a canonical algorithm, a key part of creative design and optimization.

6.1 Case 1: Grover's Algorithm

We invited two experts to implement Grover's algorithm [53] using QuRAFT, a quantum search algorithm known for its ability to solve the unstructured search problem faster than any classical algorithm. This case illustrates how experts develop a quantum algorithm using QuRAFT, highlighting the efficiency of our designed techniques in facilitating quantum algorithm creation process.

As the experts were quite familiar with Grover's algorithm, they quickly formulated the algorithm in mathematics by typing in LaTex. Firstly, they defined the initial state $|{+}{+}{-}\rangle$ of the quantum system. QuRAFT identified the state as the input of quantum circuit, automatically generating a state preparation component comprised of H-gates and X-gates (F1, Fig. 6-A1). Next, they described a CCX operation applied to three qubits in the math formula, which QuRAFT adeptly translated into an equivalent circuit component (F2, Fig. 6-A2). Lastly, the experts typed the matrix to define a Grover Operator, then QuRAFT encapsulated the matrix definition as a single quantum gate (F2, Fig. 6-B1).

After the formulation, the experts used QuRAFT to construct the circuit by dragging and positioning generated components. QuRAFT maintained the circuit's consistency with the formulation by highlighting connections between components (I1, Fig. 6-B2). Following circuit composition, the experts predicted specific outcomes from the Grover Operator, expecting a definitive search result of "11". QuRAFT offered the visualization that allowed them to compare the actual measurement outcomes with their expected probability (V2, Fig. 6-C1), confirming the algorithm's correctness.

Eager to delve deeper into the Grover Operator's design, the experts experimented with composing two new Grover Operators using classical quantum gates instead of the matrix definition. They replaced the new designed operators for previous one (V4, Fig. 6-C2). According to the switching between different formulations, they found an equivalent design that yielded the same result "11" and another design producing "01". On this basis, the experts came up with a new idea that they could create another operator that results in "10" by rearranging the qubit sequence to flip the component (I2, Fig. 6-C3). They rearranged the qubits directly within the circuit, with QuRAFT synchronizing the formulation accordingly. Finally, this exploration enabled the experts to implement the algorithm in a manner consistent with the math formulation while also expanded design possibilities, showcasing QuRAFT's usability and support for creative exploration in algorithm design.

6.2 Case 2: Quantum Phase Estimation

In this case, experts were invited to reproduce the quantum phase estimation, a critical technique in quantum computing for estimating the eigenvalue of an eigenvector of a unitary matrix U, as detailed in the IBM Quantum textbook [54]. This case shows how our features enhance understanding and iterations on an existing quantum algorithm.

Initially, the experts typed a unitary operator as $controlled\text{-}U$ in its matrix representation (Fig. 1-B1), identifying it as the target to be estimated. QuRAFT then transformed this definition into a CU-gate representation within the quantum circuit (F2). To construct the quantum system, the experts registered two qubits. The first, denoted as the data qubit $|x\rangle$, is designed to hold the data from subsequent computations. The second, represented as the quantum state eigenvector $|\psi\rangle$, corresponds to the relevant eigenvector of U, initialized as $|1\rangle$. To note the duty of these qubits, the experts added annotations next to their notations. The QuRAFT seamlessly synchronized these annotations on the circuit diagram, ensuring that the roles of the qubits were clearly visible and understandable within the visual representation of the circuit (V1, Fig. 1-A1). Consequently, a two-qubit system was initialized in the state $|\psi\rangle|x\rangle = |1\rangle|0\rangle$ (F1, Fig. 1-A2). The experts clicked on the mathematical representation of initial state to examine its preparation components in the quantum circuit. They confirmed that an X-gate was appropriately applied to produce the state $|1\rangle$.

Following this setup, the experts formulated a transformation of the state to $|\psi\rangle|+\rangle$ by applying a Hadamard operation on $|x\rangle$, as shown in the math formulation (Fig. 1-B2). QuRAFT translated this formulation into a H-gate operation targeting $|x\rangle$ (F2). The experts found the outcomes correctly fit their expectation, then proceeded to compose the new generated component into the circuit. They dragged the component from the Component Panel (Fig. 1-C) to the Circuit Panel (Fig. 1-A). Connection highlights ensured that the input of this component $|x\rangle$ was connected appropriately with the output of the state preparation (I1). Meanwhile, the measurement visualization demonstrated the current output of the circuit was consistent with experts' expectation written in math (V2, Fig. 1-B2). Subsequently, the experts applied the CU operation and calculated excepted results (F2, Fig. 1-B3). This operation was also transformed and positioned within the circuit (I1, Fig. 1-A3). They performed a measurement check against the expected outcomes. Continuing, applying another Hadamard operation brought the system to its final state. Through this process, the experts observed the phase kickback effect introduced by the CU operation and annotated this phenomenon on the CU-gate (V1, Fig. 1-A3). Lastly, they varied the parameter θ in the $controlled\text{-}U$ matrix to validate the accuracy of the algorithm through measurement outcomes with different circuit configurations (V2, Fig. 1-B5).

To enhance the precision of the phase estimation, the experts introduced a doubling phase method by modifying the single CU operation to a double CU operation in the mathematical formulation (Fig. 1-B4), which accordingly doubled the application of CU-gate in the circuit (F2, Fig. 1-A4). This increased the effect of phase kickback to 2θ, thereby improving estimation precision. This phenomenon, $CU(\theta)CU(\theta) = CU(2\theta)$, was verified by experts using the combination of Component Replacement and Measurement Correlation features.

For applying this enhancement to the algorithm, the experts expanded the quantum system by adding the number of data qubits, previously denoted as $|x\rangle$. When the experts selected the first $|x\rangle$ in formulas for editing, QuRAFT traced the appearance of this qubit in both math formulas and quantum circuits with highlights (V3), aiding the experts in expanding $|x\rangle$ to two qubits, now denoted as $|a_0a_1\rangle$. Consequently, they updated the operators previously affecting $|x\rangle$, e.g., applied new Hadamard operations on the new added qubit. They also defined a new phase kickback component comprising more CU operations, followed by the application of an inverse QFT operation to convert the phase state into the evaluation state. Finally, they inputted various θ values and verified the success of algorithm enhancement.

7 User Evaluation

To evaluate the effectiveness of our proposed design techniques and their impact on the quantum algorithm design process, we conducted a qualitative and a quantitative evaluation. QuRAFT was developed as a web-based interface, functioned as a technology probe to facilitate participants utilizing our techniques in quantum algorithm design tasks.

7.1 Qualitative Evaluation

This evaluation was designed to evaluate the effectiveness of QuRAFT's formulation, implementation, and validation techniques in supporting quantum algorithm design.

7.1.1 Methodology

We conducted the qualitative evaluation through semi-structured interviews and observational sessions with quantum computing experts.

Participants. We recruited 10 participants (P1-P10, age: 21-27) from university quantum computing labs, including 2 undergraduates (P1-P2) and 8 graduates (P3-P10), one of whom also serves as a quantum computing teacher (P6). While this shared background ensured their familiarity with the core concepts, we acknowledge that it also represents a limitation, as their perspectives may not fully capture those of industry practitioners. Future work should aim to include a more diverse participant pool. These participants represented a breadth of specializations within quantum computing, such as quantum finance, superconducting circuits, quantum algorithm optimization, etc. None of the participants were the same as the experts who helped design the system. Four of them had research experience over two years. All participants were proficient in Qiskit, alongside other tools like Pennylane, QuTiP, and Wolfram, reflecting a well-rounded toolset in their daily work, which ranged from learning to academic research. Six participants had a strong foundation in the algorithms explored during the evaluation. The experiments were conducted online and lasted 40 to 60 minutes, with each participant equally receiving approximately US$15 as an incentive for their participation.

Task. It was divided into two parts. The first part, *Reproduction*, required participants to use QuRAFT to implement a well-known quantum algorithm. This part aimed to evaluate the system's ability to facilitate creating an algorithm accurately, which involves the formulation and implementation process. The second part, *Creative Design*, challenged participants to re-design a part of the reproduced algorithm. This part was intended to assess QuRAFT's support for the creative and innovative aspects of algorithm development, focusing on how well it enables users to conceptualize, visualize, and iterate on new ideas of quantum algorithms. Participants were recommended to sufficiently use our proposed validation techniques to verify their work.

Procedure. The evaluation began with a 5-minute introduction to outline the study purpose, then followed by a 15-minute tutorial, providing a comprehensive walkthrough of QuRAFT's features, using a specific use case. Before starting the task, participants were given sufficient time to freely navigate the system and pose any questions regarding the algorithm they would be working on, ensuring a level of comfort and familiarity with the user interface. Participants then independently completed the task over a 20-minute period, encouraged to explore QuRAFT's features while having access to support for any inquiries. After task completion, participants were invited to a 15-minute semi-structured interview. We elicited their evaluation on each feature and the usability of QuRAFT using 7-Point Likert questionnaires, gathered in-depth insights into how these features might influence their work in practical scenarios, and collected suggestions for potential improvements.

7.1.2 Results

In summary, the evaluation results highlight the effectiveness of QuRAFT's formulation, implementation, and validation techniques in enhancing the quantum algorithm design process, as well as the system's overall usability and user experience (Fig. 7). We provide a detailed analysis below.

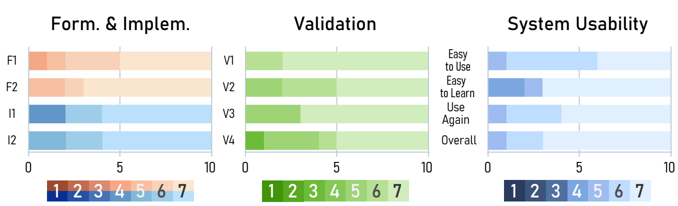


Fig. 7. The results of qualitative evaluation, including the ratings of QuRAFT's features and system usability.

Formulation Techniques. Participants found the formulation techniques provided by QuRAFT to be highly effective, with an average rating of 6.35 ($\sigma = 0.933$). The State Preparation feature (F1) was praised for its "efficiency in setting up initial states," which users found essential for starting the design process on the right foot (P1). The Math-to-Circuit Mapping (F2) was acclaimed for its "innovative approach to translating complex mathematical expressions directly into circuit components," streamlining the algorithm design process (P6). Participants acknowledged the usability of these features in reducing effort and time spent on algorithm formulation, especially for creating from scratch or reproducing existing one following mathematical descriptions.

Implementation Techniques. The implementation techniques received positive feedback, with an average rating of 6.3 ($\sigma = 1.031$). The Gate Composition feature (I1) was commended as it provided "an intuitive and reliable guidance on how to accurately compose quantum gates" (P5), which was benefited from the consistent visual linking between mathematical expressions and circuit components. Participants expressed that composing quantum gates was most error-prone in traditional quantum programming environments, and QuRAFT's feature significantly reduced the likelihood of mistakes. Meanwhile, the Qubit Arrangement feature (I2) was appreciated for allowing "flexible and intuitive organization of qubits," enhancing the overall design clarity and effectiveness (P8). Furthermore, one participant noted that visually building the circuit is preferable to typically writing code for sketching out ideas, as it provides a more intuitive and interactive experience (P3).

Validation Techniques. Validation techniques were highly valued by participants, with an average score of 6.35 ($\sigma = 0.9753$). Synchronized Annotation (V1) and State Evolution Tracing (V3) were recognized for their roles in "enhancing understanding of the roles of algorithm components and qubits," (P10). The cross-view highlighting that comes with user interaction was also seen as a significant advantage, as it reduces the cognitive load when the circuit scales up (P2). Measurement Correlation (V2) and Component Replacement (V4) were particularly noted for providing "clear, real-time insights into algorithm behavior and potential issues," aiding in swift adjustments (P7).

System Usability. In terms of system usability, QuRAFT was rated favorably across four metrics: easy to learn (6.3), easy to use (6.2), likelihood to use again (6.5), and overall satisfaction (6.6). Participants appreciated the system's user-friendly interface, comprehensive feature set, and the overall efficiency it brought to the quantum algorithm design process. Participants also commented that QuRAFT provided a novel and effective paradigm for quantum algorithm design, "creating quantum circuits tightly coupled with mathematical expressions was very useful, but it had remained unrealized in other tools" (P4). The participant who was also a teacher (P6) indicated that QuRAFT's user-friendly and visual interface is more accessible for students to learn quantum computing algorithms, compared to static textbooks or traditional programming environments.

Suggestions. Participants also provided crucial critiques that highlight areas for improvement. A frequently mentioned shortcoming was the lack of an undo/redo feature, which participants considered essential for an efficient workflow. Some participants expressed concerns about the system's scalability, noting that while the interface was effective for the given tasks, it could become visually cluttered with more complex circuits. Furthermore, two participants suggested that a more flexible input method for formulas, such as pen-based sketching, would better support rapid, exploratory ideation compared to the rigidity of LaTeX.

7.2 Quantitative Evaluation

This evaluation was designed to evaluate the overall impact of QuRAFT on the efficiency and accuracy of quantum algorithm design using quantitative metrics.

7.2.1 Methodology

We followed a within-subject design to compare the performance of participants using QuRAFT and without QuRAFT in completing a quantum algorithm design task.

Experimental Conditions. The evaluation was conducted under two conditions for comparison:

- Baseline condition: Participants wrote code using Qiskit in Jupyter Notebook to complete the task. IBM Quantum Composer was also available for participants to use if they needed a visual interface.
- QuRAFT condition: Participants used QuRAFT to complete the task, leveraging its formulation, implementation, and validation techniques.

Participants. We additionally recruited six participants for the qualitative evaluation. All of them are quantum computing researchers from university labs, with a minimum of one-year experience in quantum algorithm design or development. All participants were proficient in Qiskit and familiar with basic quantum algorithms.

Task. The task required participants to implement the same quantum algorithm (3-qubit quantum phase estimation) under both conditions. The task involved three stages: *Formulation*, *Implementation*, and *Validation*. First, participants were given a description of the algorithm, mathematical expressions, and pseudocode describing the input, output, and procedure of the algorithm. They were asked to formulate the algorithm by mapping each mathematical expression to quantum circuits. Next, participants implemented the algorithm using provided tools. Finally, participants validated their implementation by tracing the state evolution and correlating the measurement results with expected outcomes.

Procedure. The evaluation involved two sessions for each participant, one for each condition. We counterbalanced the order of the two conditions to mitigate potential learning effects. The two sessions were separated by several days to minimize the impact of memory on the results. At the beginning of each session, participants were given a 5-minute introduction to the study purpose and the task they would be performing. Participants were then given a 10-minute tutorial on how to use QuRAFT if they were in the QuRAFT condition. Before starting the task, participants were given sufficient time to familiarize themselves with the experimental setup and ask any questions. They then independently completed the task with access to support as needed. After completing the task, participants participated in a 5-minute post-study interview to provide feedback about their experience in each condition and the challenges they encountered during the task. The evaluation lasted totally 40-60 minutes for each participant. It was conducted online, and recorded with participants' consent. Each participant equally received $20 after completing the evaluation.

Metrics. We reviewed the recorded sessions after the evaluation to collect two quantitative metrics: task completion time and the number of errors made during the task.

7.2.2 Results

Table 1 summarizes the quantitative comparison between the Baseline condition (using Qiskit with IBM Quantum Composer) and the QuRAFT system. Overall, QuRAFT yielded substantially lower task completion times and reduced number of errors error across all participants.

Task Completion Time. From a timing perspective, participants in the Baseline condition required an average of 13.01 minutes (SD = 3.32) to complete the assigned tasks. In contrast, participants using QuRAFT consistently demonstrated shorter completion times, ranging between 0.83 and 2.03 minutes, with a group average of 1.16 minutes (SD = 0.45). Each participant experienced significant reduction in task duration. These findings suggest that QuRAFT's interactive features and automated processes considerably diminish the manual overhead in developing quantum algorithms.

Error Analysis. Error analysis further supports the effectiveness of the QuRAFT system. In the Baseline condition, participants recorded an average of 1.17 errors (SD = 0.75), with a total of 7 errors observed. These errors comprised incorrect gate placement (1/7), erroneous qubit assignment (3/7), and circuit structure mistakes during the insertion of measurement code in Jupyter Notebook (3/7). In contrast, only a single error was recorded with QuRAFT, which resulted from a misunderstanding regarding the order of mathematical expressions. This near-elimination of errors highlights how features such as visual circuit composition and context-aware validation can help avoid mistakes commonly observed in standard quantum programming tools.

TABLE 1
Quantitative comparison between the Baseline (Qiskit with IBM Quantum Composer) and QuRAFT conditions.

Participant	Task Time (min)		Error Count	
	Baseline	QuRAFT	Baseline	QuRAFT
P1	16.91	**1.01**	2	**0**
P2	12.70	**2.03**	1	**1**
P3	12.33	**0.9**	1	**0**
P4	8.88	**0.9**	1	**0**
P5	16.88	**0.83**	2	**0**
P6	10.32	**1.2**	0	**0**
Mean (SD)	13.01 (3.32)	**1.16 (0.45)**	1.17 (0.75)	**0.17 (0.41)**

8 Discussion

In this section, we discuss design implications, limitations, and future work of our study.

8.1 Design Implications

This design study reveals implications for future research on quantum algorithm design, informed by our evaluation results and domain expert feedback.

Augmenting quantum circuit with math significantly enhances user comprehension. Most participants (8/10) commended our system for its effectiveness in comprehending algorithms. Feedback from domain experts indicates a general preference for reading visual quantum circuit rather than math formulas to learn algorithms in their daily work. By visually linking abstract mathematical concepts to their practical applications in quantum circuits, users gain a deeper and more intuitive grasp of the underlying processes, underscoring the value of mathematical augmentation in quantum computing education and design.

Concurrent interaction across math and quantum circuit boosts user engagement and efficiency. Participants found the interactive math formulas engaging and enjoyable to use. Our techniques are commented "novel" by experts as their frequently used tools often only provide interactions on either circuit or math (P6). Our design space that encompasses both mathematical and circuit modalities, opens up a broader canvas for interaction design. Participants further appreciated how our system "reduces effort cost in switching between formulation, quantum circuit, and programming code" (P3). This seamless integration allows users to directly manipulate and observe the effects of changes in mathematical formulations on quantum circuits in real-time, thereby streamlining the design process and significantly reducing the time and effort needed to develop and refine quantum algorithms.

Levels of interaction complexity informs design approach. An observation from user evaluation revealed participants usually have more requirements on complex interactions. For basic, atomic interactions, straightforward solutions such as simple highlight are adequate and provide sufficient guidance, such as the State Preparation. However, as interactions become more collective and relational, encompassing multiple components or complex relationships between elements, there is a clear need for more sophisticated tools. This gradation in interaction complexity underscores the necessity for a tiered approach to tool design, ensuring that user needs are met at every level of engagement.

Potential to integrate high-level action techniques. During the evaluation, we observed participants frequently employed specific techniques in combination to accomplish particular tasks. For instance, Qubit Arrangement was often paired with State Evolution Tracing. Some participants to suggest that the integration of techniques to facilitate high-level actions could enhance design efficiency and reduce the learning curve. Our work lays the foundation for this advancement, suggesting that future tools could benefit from merging multiple actions into cohesive operations.

Code is another essential representation for large-scale quantum algorithm design. In our post-study interviews, quantum experts expressed a clear shift in tool preferences based on circuit scale. For small, simple circuits, experts favor visual interfaces and visualization tools that enable rapid prototyping and immediate validation of design ideas. However, as the complexity increases, these intuitive displays become cluttered and hard to interpret, leading experts to rely more on programming tools. Techniques such as looping and code encapsulation allow for efficient extension and validation of large-scale circuits, making code an essential modal of representation alongside math formulas and circuit diagrams. Our work has presented an effective design paradigm by integrating math and circuit representation for small-scale quantum algorithms. With the integration of code representations, it could be extended to support the development of larger-scale quantum algorithms.

8.2 Limitations and Future Work

Our investigation through QuRAFT uncovers several areas ripe for future exploration and development, with several challenges and opportunities ahead.

Scalability. The evaluation demonstrated QuRAFT's effectiveness for small-scale and toy problems, which is significant for learning quantum basis and early-stage development of scalable algorithms. As the algorithms and circuits grow in complexity and size, incorporating a greater number of formulas and quantum gates, the clarity of current visual designs may be affected. To address this, future iterations could explore the integration of abstraction and simplification techniques to reduce the cognitive load on users and minimize visual clutter [36], potentially adapting our approach for more advanced quantum computing tasks.

Input Modalities. Currently, QuRAFT relies on LaTeX for the input of mathematical formulas, a method that, while precise, may not be the most intuitive for all users. Feedback from experts indicates a potential enhancement in usability through the adoption of pen input for entering formulas, making the interface more accessible and aligning with natural writing habits. Additionally, the integration of sketch-based interactions for mathematical expressions [41], such as sliding over values to adjust them, could further streamline the design process, offering a more dynamic and user-friendly approach to modifying algorithm parameters directly within the system.

Intelligent Reasoning and Generation. Our concept of mapping mathematical formulations to quantum circuit representations has been well appreciated by experts. Our current methodology relies on visual mappings between mathematical concepts and quantum circuits, guided by predefined rules. Moving forward, there is significant potential to adopt machine learning techniques to enable more sophisticated mappings. Furthermore, integrating mathematical reasoning and error-checking functionalities could also streamline the design process and reducing the likelihood of errors in the final quantum algorithm.

Noisy Environment. While our current work emphasizes the theoretical and logical aspects of quantum algorithm design, the practical application of these algorithms on real quantum computers introduces the challenge of noise [3]. To address this, future iterations of our approach could incorporate features for noise visualization [55] and error mitigation strategies [56], [57], equipping users with tools to adapt their algorithms for practical applicability and enhanced robustness on actual quantum hardware.

9 Conclusion

This study explored the design space of bridging mathematical concepts and quantum circuits in quantum algorithm design process. We characterize the design process across its textual and visual representation spaces. QuRAFT is developed to facilitate the transition between mathematical concepts and practical quantum circuit, with a series of delicately designed techniques. Our case studies and user evaluation have demonstrated the benefits of integrating math and circuit in quantum algorithm design, which also provide valuable design implications for future work.

Acknowledgments

This research was supported by the Fundamental Research Funds for the Central Universities (No. 226-2024-00228), the National Natural Science Foundation of China (No. 62132017), "Pioneer" and "Leading Goose" R&D Program of Zhejiang (No. 2024C01167), Zhejiang Provincial Natural Science Foundation of China (No. LD24F020011), and Zhejiang Pioneer (Jianbing) Project (No. 2023C01036).

The authors acknowledge the use of AI tools for language polishing and proofreading of this manuscript.

References

[1] A. Montanaro, "Quantum algorithms: an overview," *npj Quantum Inf.*, vol. 2, no. 1, pp. 1–8, Jan. 2016.

[2] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM J. Comput.*, vol. 26, no. 5, pp. 1484–1509, Oct. 1997.

[3] J. Preskill, "Quantum computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, Aug. 2018.

[4] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan, "Quantum computational chemistry," *Reviews of Modern Physics*, vol. 92, no. 1, p. 015003, Mar. 2020.

[5] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195–202, Sep. 2017.

[6] D. J. Egger, C. Gambella, J. Marecek, S. McFaddin, M. Mevissen, R. Raymond, A. Simonetto, S. Woerner, and E. Yndurain, "Quantum computing for finance: State-of-the-art and future prospects," *IEEE Transactions on Quantum Engineering*, vol. 1, pp. 1–24, Oct. 2020.

[7] P. W. Shor, "Introduction to quantum algorithms," in *Proc. Sympos. Appl. Math.*, vol. 58, 2002, pp. 143–160.

[8] A. tA v, M. S. ANIS, Abby-Mitchell, H. Abraham, AduOffei, R. Agarwal, G. Agliardi, M. Aharoni, V. Ajith, I. Y. Akhalwaya, G. Aleksandrowicz *et al.*, "Qiskit: An open-source framework for quantum computing," 2021.

[9] C. Developers, "Cirq," Zenodo, December 2022.

[10] C. Gidney, "Quirk: Quantum circuit simulator," https://algassert.com/quirk, Jan. 2024.

[11] IBM Quantum, "IBM quantum composer," https://quantum.ibm.com/composer, Jan. 2024.

[12] Z. Chen and H. Xia, "Crossdata: Leveraging text-data connections for authoring data documents," in *Proc. CHI*. New York: ACM, 2022, pp. 1–15.

[13] R. E. Mayer, "Multimedia learning," in *Psychology of learning and motivation*. Elsevier, 2002, vol. 41, pp. 85–139.

[14] A. A. Adedoyin, J. J. Ambrosiano, P. M. Anisimov, W. R. Casper, G. Chennupati, C. J. Coffrin, H. N. Djidjev, D. O. Gunter, S. Karra, N. W. Lemons *et al.*, "Quantum algorithm implementations for beginners," *ACM Transactions on Quantum Computing*, vol. 3, no. LA-UR-20-22353, Jul. 2022.

[15] C. Shao, Y. Li, and H. Li, "Quantum algorithm design: techniques and applications," *Journal of Systems Science and Complexity*, vol. 32, no. 1, pp. 375–452, Feb. 2019.

[16] L. Ruiz-Perez and J. C. Garcia-Escartin, "Quantum arithmetic with the quantum fourier transform," *Quantum Information Processing*, vol. 16, pp. 1–14, Jun. 2017.

[17] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G. H. Booth *et al.*, "The variational quantum eigensolver: a review of methods and best practices," *Physics Reports*, vol. 986, pp. 1–128, Aug. 2022.

[18] R. J. Lipton and K. W. Regan, *Quantum algorithms via linear algebra: a primer*. Cambridge: MIT Press, 2014.

[19] A. Daskin and S. Kais, "Decomposition of unitary matrices for finding quantum circuits: application to molecular hamiltonians," *The Journal of chemical physics*, vol. 134, no. 14, Apr. 2011.

[20] E. C. Behrman, J. E. Steck, P. Kumar, and K. Walsh, "Quantum algorithm design using dynamic learning," *arXiv preprint arXiv:0808.1558*, vol. abs/0808.1558, Aug. 2008.

[21] J. Bang, J. Ryu, S. Yoo, M. Pawłowski, and J. Lee, "A strategy for quantum algorithm design assisted by machine learning," *New Journal of Physics*, vol. 16, no. 7, p. 073017, Jul. 2014.

[22] C. G. Kang and H. Oh, "Modular component-based quantum circuit synthesis," *ACM on Programming Languages*, vol. 7, no. OOPSLA1, Apr. 2023.

[23] D. Maslov, G. W. Dueck, D. M. Miller, and C. Negrevergne, "Quantum circuit simplification and level compaction," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 3, pp. 436–444, Mar. 2008.

[24] B. Apak, M. Bandic, A. Sarkar, and S. Feld, "KetGPT–dataset augmentation of quantum circuits using transformers," *CoRR*, vol. abs/2402.13352, Feb. 2024.

[25] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, S. Ahmed, V. Ajith, M. S. Alam, G. Alonso-Linaje, B. AkashNarayanan, A. Asadi *et al.*, "PennyLane: Automatic differentiation of hybrid quantum-classical computations," *CoRR*, vol. abs/1811.04968, Nov. 2018.

[26] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*, 10th ed. Cambridge: Cambridge University Press, 2010.

[27] E. W. Bethel, M. G. Amankwah, J. Balewski, R. Van Beeumen, D. Camps, D. Huang, and T. Perciano, "Quantum computing and visualization: A disruptive technological change ahead," *IEEE Computer Graphics and Applications*, vol. 43, no. 6, pp. 101–111, Nov. 2023.

[28] F. T. Arecchi, E. Courtens, R. Gilmore, and H. Thomas, "Atomic coherent states in quantum optics," *Physical Review A*, vol. 6, no. 6, p. 2211, 1972.

[29] J. B. Altepeter, E. R. Jeffrey, M. Medic, and P. Kumar, "Multiple-qubit quantum state visualization," in *Proc. CLEO*. Piscataway: IEEE, 2009, pp. 1–2.

[30] M. Galambos and S. Imre, "Visualizing the effects of measurements and logic gates on multi-qubit systems using fractal representation," *International Journal on Advances in Systems and Measurements*, vol. 5, no. 1, pp. 1–10, Jun. 2012.

[31] V. N. Chernega, O. V. Man'ko, and V. I. Man'ko, "Triangle geometry of the qubit state in the probability representation expressed in terms of the triada of malevich's squares," *Journal of Russian Laser Research*, vol. 38, no. 2, pp. 141–149, Apr. 2017.

[32] J.-B. Lamy, "Dynamic software visualization of quantum algorithms with rainbow boxes," in *Proc. IVAPP*. SciTePress, Feb. 2019.

[33] S. Ruan, R. Yuan, Q. Guan, Y. Lin, Y. Mao, W. Jiang, Z. Wang, W. Xu, and Y. Wang, "Venus: A geometrical representation for quantum state visualization," *Computer Graphics Forum*, vol. 42, no. 3, pp. 247–258, Jun. 2023.

[34] Z. Tao, Y. Pan, A. Chen, and L. Wang, "ShorVis: A comprehensive case study of quantum computing visualization," in *Proc. ICVRV*, IEEE. Piscataway: IEEE, 2017, pp. 360–365.

[35] S. Ruan, Q. Guan, P. Griffin, Y. Mao, and Y. Wang, "Quantumeyes: Towards better interpretability of quantum circuits," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–13, Nov. 2023.

[36] Z. Wen, Y. Liu, S. Tan, J. Chen, M. Zhu, D. Han, J. Yin, M. Xu, and W. Chen, "Quantivine: A visualization approach for large-scale quantum circuit representation and analysis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 30, no. 1, pp. 573—-583, Oct. 2023.

[37] S. Ruan, Z. Liang, Q. Guan, P. Griffin, X. Wen, Y. Lin, and Y. Wang, "Violet: Visual analytics for explainable quantum neural networks," *CoRR*, vol. abs/2312.15276, Dec. 2023.

[38] A. Head, A. Xie, and M. A. Hearst, "Math augmentation: How authors enhance the readability of formulas using novel visual design practices," in *Proc. CHI*. New York: ACM, 2022, pp. 1–18.

[39] J. J. LaViola Jr and R. C. Zeleznik, "MathPad2: a system for the creation and exploration of mathematical sketches," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 432–440, Aug. 2006.

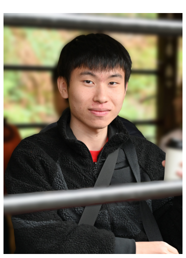
[40] R. Zeleznik, A. Bragdon, F. Adeputra, and H.-S. Ko, "Hands-on math: a page-based multi-touch and pen desktop for technical work and problem solving," in *Proc. UIST*. New York: ACM, 2010, pp. 17–26.

[41] N. Chulpongsatorn, M. S. Lunding, N. Soni, and R. Suzuki, "Augmented math: Authoring ar-based explorable explanations by augmenting static math textbooks," in *Proc. UIST*. New York: ACM, 2023, pp. 1–16.

[42] S. Latif, S. Chen, and F. Beck, "A deeper understanding of visualization-text interplay in geographic data-driven stories," *Computer Graphics Forum*, vol. 40, no. 3, pp. 311–322, Jun. 2021.

[43] N. Kong, M. A. Hearst, and M. Agrawala, "Extracting references between text and charts via crowdsourcing," in *Proc. CHI*. New York: ACM, 2014, pp. 31–40.

[44] S. K. Badam, Z. Liu, and N. Elmqvist, “Elastic documents: Coupling text and tables through contextual visualizations for enhanced document reading,” *IEEE transactions on visualization and computer graphics*, vol. 25, no. 1, pp. 661–671, Aug. 2018.
[45] C. Lai, Z. Lin, R. Jiang, Y. Han, C. Liu, and X. Yuan, “Automatic annotation synchronizing with textual description for visualization,” in *Proc. CHI*. New York: ACM, 2020, pp. 1–13.
[46] Z. Chen, Q. Yang, X. Xie, J. Beyer, H. Xia, Y. Wu, and H. Pfister, “Sporthesia: Augmenting sports videos using natural language,” *IEEE transactions on visualization and computer graphics*, vol. 29, no. 1, pp. 918–928, Oct. 2022.
[47] R. Metoyer, Q. Zhi, B. Janczuk, and W. Scheirer, “Coupling story to visualization: Using textual analysis as a bridge between data and interpretation,” in *Proc. IUI*. New York: ACM, 2018, pp. 503–507.
[48] Q. Zhi, A. Ottley, and R. Metoyer, “Linking and layout: Exploring the integration of text and visualization in storytelling,” in *Proc. CGF*, vol. 38, no. 3. Wiley Online Library, 2019, pp. 675–685.
[49] S. Latif, Z. Zhou, Y. Kim, F. Beck, and N. W. Kim, “Kori: Interactive synthesis of text and charts in data documents,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, pp. 184–194, Jan. 2021.
[50] N. Sultanum, F. Chevalier, Z. Bylinskii, and Z. Liu, “Leveraging text-chart links to support authoring of data-driven articles with VizFlow,” in *Proc. CHI*. New York: ACM, 2021, pp. 1–17.
[51] D. Masson, S. Malacria, G. Casiez, and D. Vogel, “Charagraph: Interactive generation of charts for realtime annotation of data-rich paragraphs,” in *Proc. CHI*. New York: ACM, Apr. 2023, pp. 1–18.
[52] A. Javadi-Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross, B. R. Johnson, and J. M. Gambetta, “Quantum computing with Qiskit,” 2024.
[53] L. K. Grover, “A fast quantum mechanical algorithm for database search,” in *Proc. STOC*. New York: ACM, 1996, pp. 212–219.
[54] IBM Quantum, “IBM Quantum Learning,” https://learning.quantum.ibm.com/, accessed: 2024-03-30.
[55] S. Ruan, Y. Wang, W. Jiang, Y. Mao, and Q. Guan, “VACSEN: A visualization approach for noise awareness in quantum computing,” *IEEE Trans. Vis. Comput. Graph.*, vol. 29, no. 1, pp. 462–472, Jan. 2023.
[56] S. Tannu, P. Das, R. Ayanzadeh, and M. Qureshi, “Hammer: boosting fidelity of noisy quantum circuits by exploiting hamming behavior of erroneous outcomes,” in *Proc. ASPLOS*. New York: ACM, 2022, pp. 529–540.
[57] J. Chen, Z. Wen, L. Zheng, J. Lu, H. Lu, Y. Ren, and W. Chen, “HammingVis: A visual analytics approach for understanding erroneous outcomes of quantum computing in hamming space,” *Graphical Models*, vol. 136, p. 101237, 2024.



Zhen Wen is a Ph.D. candidate at the State Key Lab of CAD & CG, College of Computer Science and Technology, Zhejiang University. He received the B.E. degree in Software Engineering from Zhejiang University of Technology in 2021. His research interests include visual analytics and quantum computing.

Jieyi Chen is a Master student in the State Key Lab of CAD&CG at Zhejiang University, China. She received the B.E. degree in Software Engineering from Zhejiang University of Technology in 2023. Her research interests include visualization and visual analytics.

Yao Lu is a Master's candidate in the Master of Science in Information Networking program at Carnegie Mellon University. He received his B.E. degree in Software Engineering from Zhejiang University in 2024. His research interests include distributed systems, cloud computing, and visual analytics.

Siwei Tan an associate professor at Zhejiang University. His interests include the quantum algorithm and computer architecture. He has published more than 20 papers in top journals and conferences, including MICRO, HPCA, ASPLOS, TCAD, DAC, VIS and TVCG.

Jianwei Yin is currently a full professor in the College of Computer Science, Zhejiang University (ZJU), China. He haspublished more than 100 papers in top international journals and conferences. His current research interests include quantum computing, service computing and business process management. He is also the Associate Editor of the IEEE Transactions on Services Computing.

Minfeng Zhu is a tenure-track assistant professor at the School of Software Technology, Zhejiang University. He received his Ph.D. Degree in Computer Science from the State Key Laboratory of CAD&CG, Zhejiang University in 2020. His research interests include artificial intelligence and visual analysis. For more information, please visit: https://minfengzhu.github.io.

Wei Chen is a professor in the State Key Lab of CAD&CG at Zhejiang University. His current research interests include visualization and visual analytics. He has published more than 80 IEEE/ACM Transactions and IEEE VIS papers. He actively served in many leading conferences and journals, like IEEE PacificVIS steering committee, ChinaVIS steering committee, paper cochairs of IEEE VIS, IEEE PacificVIS, IEEE LDAV and ACM SIGGRAPH Asia VisSym. He is an associate editor of IEEE TVCG, IEEE TBG, ACM TIST, IEEE T-SMC-S, IEEE TIV, IEEE CG&A, FCS, and JOV. More information can be found at: http://www.cad.zju.edu.cn/home/chenwei.