Quantum Circuit Vision: Cost-Aware Evaluation of Visual AI Agents for Quantum Code Generation

Dongping Liu[§†]
Amazon Web Services
Hong Kong, China

Aoyu Zhang[†]
Amazon Web Services
Beijing, China

Luyao Zhang[†*]
Duke Kunshan University
Suzhou, China

Abstract

Can AI agents visually comprehend quantum circuit diagrams and generate verified executable code—and at what cost? We present Quantum Circuit Vision, a cost-aware evaluation framework for multimodal AI agents on quantum circuit visual understanding. We construct a 132-circuit benchmark spanning 13 categories (1–10 qubits) with executable Amazon Braket code and unitary-fidelity verification. Evaluating three frontier Claude-family models at different capability-cost tiers with $n = 5$ repeated trials, we find that the mid-tier model (Sonnet 4.6, 1.30× credits) offers the most favorable balance on the cost–accuracy frontier: 91% pass rate on the core subset at 18% of the per-call cost of the strongest model (Opus 4.6), whose accuracy advantage is not statistically significant (paired t: $p = 0.083$). Logistic regression confirms that circuit depth—not qubit count—is the primary predictor of failure ($p < 0.001$). Chain-of-thought prompting shows no statistically significant effect (all $p > 0.18$, $n = 5$), suggesting that visual pattern recognition outweighs explicit reasoning strategy for structurally coupled diagrams. We propose a cascade routing strategy (cheap→expensive models) that achieves 84% accuracy at 38% of single-model cost, demonstrating that model routing dominates prompt engineering as a cost lever. We release QCV-Dataset (132 circuits, 5 modalities, 1,931 files) on Hugging Face Hub as an open evaluation infrastructure with structured metadata for discoverability, interoperability, and responsible AI documentation, and all evaluation code, cost logs, and verification scripts on GitHub for full reproducibility.

Keywords

Quantum Computing; Multimodal LLMs; Visual Code Generation; Cost-Aware Evaluation; Benchmark

1 Introduction

The year 2025 marked a quantum inflection point: the Nobel Prize in Physics recognized experiments with superconducting qubits that enabled macroscopic quantum coherence, while the ACM Turing Award honored the foundations of quantum information theory—including the BB84 protocol that underpins quantum key distribution. The United Nations declared 2025 the International Year of Quantum Science and Technology. Quantum computing has crossed from theoretical promise to engineering reality [1, 5, 21]. Concurrently, the convergence of artificial intelligence and quantum computing has emerged as a transformative research frontier, with AI techniques advancing challenges across the full quantum stack—from device design and error correction to circuit compilation and verification [3].

Yet a fundamental bottleneck persists: the gap between how quantum algorithms are *communicated* (as circuit diagrams in papers, textbooks, and courses) and how they are *executed* (as code in software development kits (SDKs) such as Amazon Braket [4], Qiskit [37], or Cirq). Every quantum circuit published in a research paper must be manually translated into executable code—a tedious, error-prone process requiring both visual interpretation skills and SDK-specific programming expertise. Recent surveys have mapped AI's growing role in quantum preprocessing, including circuit compilation and synthesis [3], yet the specific challenge of *visually interpreting* circuit diagrams and generating verified executable code remains unaddressed. As quantum hardware scales and the volume of published circuits grows exponentially, this manual translation becomes an unsustainable bottleneck for scientific progress.

We argue that **automating the translation from quantum circuit diagrams to verified executable code is a foundational AI-for-Science challenge**. Inspired by how AlphaFold [20] accelerated structural biology by automating protein structure prediction, we envision that a system reliably reading quantum circuit diagrams and producing verified code could similarly accelerate quantum algorithm development, quantum-safe cryptography research [15, 23], and the broader quantum computing ecosystem.

Why quantum circuits, not classical? Classical integrated circuits have mature electronic design automation (EDA) toolchains developed over 50 years. Quantum circuits have *none of these tools*: every circuit must be manually designed, coded, and verified. Yet the quantum computing market is projected to reach $450–850B in economic value by 2040 [28], and the bottleneck is not hardware but the *software toolchain* for designing and verifying quantum algorithms at scale [11, 35].

Why autonomous design, not just translation? Translation (diagram→code) is the first step, but the ultimate value lies in AI systems that can *invent* new quantum circuits—discovering more efficient decompositions, novel error correction codes [1], or better variational ansätze [11].

Why cost-aware evaluation? To our knowledge, no prior work jointly evaluates accuracy, latency, and cost for visual AI agents on scientific diagrams. Deploying AI agents at scale requires understanding not only *what* they can do but *at what cost*. As agentic AI systems are increasingly used for scientific tasks, evaluation frameworks must jointly optimize accuracy, latency, and computational

[§]Work done while at Amazon Web Services. Dongping Liu is currently with Tenorshare, Hong Kong, China.
[†]The authors are listed in alphabetical order according to last names and, then, first names. **Acknowledgments**: We thank the anonymous reviewers of the KDD Workshop on Evaluation and Trustworthiness of Agentic AI, KDD-2026, August 9, 2026, Jeju, South Korea for their constructive feedback, which greatly improved the quality and rigor of this work.
[*]The Corresponding author: Email: lz183@duke.edu, Digital Innovation Research Center and Social Science Division, Duke Kunshan University, Address: Duke Avenue No.8, Kunshan, Suzhou, Jiangsu, China, 215316.

cost—enabling practitioners to select the right model tier for their budget and quality requirements.

In this paper, we take the first step toward this vision by establishing Quantum Circuit Vision (QCV)—the first evaluation framework including multimodal quantum circuit dataset, the cost-aware evaluation methodology, and baseline results that future circuit-discovery systems will build upon.

2 Related Work

2.1 Quantum Circuit Datasets for Machine Learning

The field of quantum machine learning has historically lacked standardized large-scale datasets, with most experiments relying on custom data processing and evaluation methods rather than shared benchmarks [34]. This gap has motivated a growing body of work on *quantum datasets*—structured data associated with quantum systems that serve as training and evaluation resources for both quantum and classical machine learning models.

NTangled [38] represents one of the earliest systematic efforts, introducing entangled quantum states generated by training quantum neural networks to produce states with varying levels of multipartite entanglement. NTangled shares a common motivation with our work in emphasizing quantum-specific datasets for quantum machine learning (QML) advancement. However, it defines the quantum state itself as the dataset, which fundamentally differs from circuit-level datasets: quantum states cannot be directly applied to existing classical ML frameworks without measurement or tomographic reconstruction [34].

QDataSet [33], introduced by Perrier, Youssry, and Ferrie in *Scientific Data* (2022), comprises 52 distinct types of data concerning quantum operations for single and two-qubit systems, encompassing both noise-inclusive and noise-free scenarios. Each sub-dataset contains 10,000 data points. However, QDataSet's focus is on the *physical dynamics* of small (1–2 qubit) systems rather than algorithmic circuits.

Quantum Federated Data [13], proposed by Chehimi and Saad (ICASSP 2022), addresses the distributed learning setting, representing the first quantum federated dataset for distributed quantum computing networks. While the federated setting is important, its focus on distributed learning differs from our emphasis on visual comprehension and code generation.

PennyLane Quantum Datasets [6] provide curated collections for quantum chemistry and quantum spin systems. However, two limitations are noteworthy: dataset sizes remain relatively small for state-of-the-art ML (typically hundreds to thousands of samples), and the gap between dataset availability and effective quantum/-classical ML application remains largely unaddressed.

The VQE-Generated Dataset [30] is the most methodologically similar prior work to MNISQ, comprising quantum circuits labeled by computational task. However, a critical scalability issue limits its applicability: only approximately 300 circuits per label—far below the threshold required for modern deep learning models.

MNISQ [34] (Placidi *et al.*, *Scientific Data* 2026)—a dataset encoding the Modified National Institute of Standards and Technology (MNIST) family of classical image benchmarks into quantum circuits—represents a paradigm shift, containing 4.95 million quantum circuits mapped into 10-qubit systems with up to 100 two-qubit gates. MNISQ baseline experiments show that quantum kernel methods achieve up to 97% accuracy on multiclass classification, while classical sequence models (S4 [18], Transformers [41], LSTM [19]) achieve up to 77.78% on QASM descriptions. While MNISQ studies circuit classification at scale, *QCV-Dataset* addresses a different but complementary question: can AI agents correctly *generate* quantum circuits from visual diagram specifications, and how does structural complexity affect this capability?

QCalEval [9][1] introduces 243 expert-annotated quantum calibration plots spanning 87 scenario types across 22 experiment families—including Rabi oscillations, DRAG pulse calibration, and single-shot readout GMM clustering—to benchmark vision-language models on quantum experimental data understanding. Each sample is evaluated through a six-part semantic scoring rubric (plot description, experimental conclusion, significance assessment, fit quality, parameter extraction, and actionable recommendation), with ground truth established via human-AI collaborative annotation and expert cross-validation. QCalEval is the first multimodal AI benchmark targeting quantum computing visuals; however, it addresses a fundamentally different task from ours—visual question answering on calibration plots versus circuit diagram-to-code generation—and its dataset is not at the circuit level. Its inclusion in Table 1 reflects its thematic relevance to multimodal AI for quantum error correction experiments, where quality of expert annotation outweighs scale.

Table 1 situates QCV-Dataset within this landscape along five dimensions spanning dataset governance, ground-truth verification, cost-aware deployment, statistical rigor, and production monitoring.

Dataset Infrastructure. Prior quantum datasets, including NTangled, QDataSet, MNISQ, and VQE-generated, provide quantum circuit data at varying scales but lack machine-readable citations (CFF), structured dataset documentation following the Gebru *et al.* (2021) Datasheets standard [17], or versioned reproducibility—none combine all three. Visual-to-code benchmarks (VGV, MMCode, Plot2Code, ChartCoder, CODE-VISION) provide no quantum circuit data at all. QCalEval [9] provides multimodal visual input (calibration plot images paired with structured question-answer pairs) but, like the visual-to-code benchmarks, does not contain quantum circuit data. QCV-Dataset is the first to offer all three governance layers together with quantum circuit data.

Task & Verification. Existing benchmarks form two disjoint camps: visual-to-code benchmarks support diagram-to-code generation but rely on functional simulation or surface-level text similarity metrics (BLEU score, pass@k) that accept semantically wrong outputs; quantum verification literature (Burgholzer & Wille [8], Viamontes *et al.* [42]) provides rigorous unitary equivalence checking but has never been applied to AI agent evaluation. QCalEval [9] bridges neither camp: it evaluates visual comprehension of quantum calibration plots through a semantic scoring rubric (with human-expert-validated ground truth), but does not involve code generation or unitary verification. QCV-Dataset bridges both: it is the *first* benchmark to pair multimodal visual-code generation with

[1] https://huggingface.co/datasets/nvidia/QCalEval

Table 1: Comparative analysis of QCV-Dataset against existing quantum datasets, visual code-generation benchmarks, and cost-aware AI evaluation frameworks. Dimensions organized by evaluation taxonomy.

Taxonomy	Dimension	NTangled	QDataSet	Q. Fed.	PennyLane	VQE-gen.	MNISQ	QCalEval	VGV	Vis.→Code	FrugalGPT	QCV (Ours)
Dataset Infra.	Scale (samples)	~10K	52×10K	1,500	~100s	~300	**4.95M**	243	59	1K–10K	N/A	132
	Quantum circuit data	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓
	Multimodal visual input	✗	✗	✗	✗	✗	✗	✓	✓	✓	✗	✓
	Datasheets for Datasets	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
	Machine-readable citation	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
	Versioned reproducibility	✗	✗	✗	✗	✗	✓	✗	✗	●	✗	✓
Task & Verify	Visual → executable code	✗	✗	✗	✗	✗	✗	✗	✓†	✓‡	✗	✓
	Objective GT verification	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓★
	Domain-expert oracle	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓◇
Cost Deploy.	Cost-aware per-invocation	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓§	✓
	Latency + billing logging	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
	Cascade / compound AI	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓
	Diminishing-returns analysis	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓
Stat. Rigor	Repeated trials ($n \geq 5$)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
	Paired significance testing	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Trust & Monit.	Failure taxonomy (268 cases)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
	Agentic verify-retry loop	✗	✗	✗	✗	✗	✗	✗	✗	✗	●	✓
	Production monitoring	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓¶

Legend: ✓ = fully supported; ● = partially supported; ✗ = not supported.
†Verilog HDL; ‡General-domain code; ★Unitary fidelity ≥0.99; ◇Quantum simulator oracle; §Inference cost only; ¶Structured CSV logs.
QCalEval [9] = quantum calibration plot benchmark (243 samples, 87 scenario types, 22 experiment families).
Vis.→Code = MMCode [22], Plot2Code [46], ChartCoder [48], CODE-VISION [43].

unitary-fidelity verification ($F = |\mathrm{Tr}(U_{\mathrm{gt}}^{\dagger} U_{\mathrm{gen}})|/2^n \geq 0.99$) via a quantum simulator as a domain-expert oracle, establishing an objective, non-LLM-judge ground truth.

Cost-Aware Deployment. FrugalGPT [14] pioneered LLM cascading for cost reduction but lacks per-invocation latency and billing logging, and operates in domains with inherently ambiguous ground truth. QCV-Dataset introduces the first cost-aware evaluation framework for visual AI agents, recording per-call credits, latency, and unitary fidelity, with a cascade routing strategy achieving 84% accuracy at 38% of single-model cost—all verified against an objective oracle.

Statistical Rigor. No prior work—quantum datasets, visual code benchmarks, cost-aware frameworks, or QCalEval—reports repeated trials with significance testing. QCV-Dataset provides $n = 5$ repeated trials per condition, paired t-tests across prompting strategies, and logistic regression confirming that circuit depth ($p < 0.001$), not qubit count ($p = 0.20$), predicts comprehension difficulty.

Trustworthiness & Monitoring. QCV-Dataset contributes 268 annotated failure cases across syntax errors, execution errors, and low-fidelity outputs, together with structured CSV logs per invocation—enabling real-time monitoring of deployed visual AI agents. Collectively, these five dimensions demonstrate that QCV-Dataset is the first benchmark to address the full evaluation lifecycle of agentic AI for scientific diagram interpretation.

2.2 Multimodal AI for Visual Code Generation

The emergence of multimodal large language models (MLLMs) capable of processing both visual and textual inputs has opened new frontiers in automated code generation from visual specifications.

VGV [45] (Wong *et al.*, DAC 2024) is the most direct methodological predecessor to our work. VGV demonstrated that MLLMs can generate Verilog hardware description language (HDL) code from classical circuit diagrams, introducing *Thinking Vision* (TV) prompting—a chain-of-thought approach that improved open-source model performance by up to 50 percentage points. Our work extends VGV in several critical directions: (1) we target quantum rather than classical circuits, finding that standardized gate symbols, strictly left-to-right temporal structure, and finite gate vocabulary yield higher baseline accuracy (78–91% vs. 40–60% in VGV); (2) we find that chain-of-thought prompting has *no statistically significant effect* on quantum circuits (all $p > 0.18$, $n = 5$), contrasting sharply with VGV's results; (3) we introduce unitary matrix fidelity verification, which is more rigorous than VGV's functional simulation for hardware circuits.

Several general-domain benchmarks inform our evaluation methodology. **MMCode** [22] evaluates MLLMs on code generation from visually rich programming problems. **Plot2Code** [46] provides a benchmark for code generation from scientific plots. **ChartCoder** [48] advanced chart-to-code generation through specialized training datasets. **CODE-VISION** [43] evaluated MLLMs' logical understanding from visual charts, finding that existing models struggle with information-dense visual inputs—consistent with our

observation that circuit depth (structural complexity) predicts failure better than qubit count.

2.3 Cost-Aware Evaluation and Agentic AI Deployment

The KDD 2026 Workshop on Evaluation and Trustworthiness of Agentic AI emphasizes evaluating not only what AI agents can accomplish but also the computational cost, latency, and reliability of their deployment. This section connects our cost-aware evaluation framework to broader trends in efficient LLM deployment.

FrugalGPT [14] (Chen, Zaharia, and Zou, 2023) established the foundational paradigm for cost-aware LLM deployment, introducing LLM cascades that route queries through models of increasing capability. FrugalGPT achieves up to 80% cost reduction while matching the accuracy of the strongest model alone. Our cascade routing strategy directly builds on this insight: by routing circuits through Haiku (cheapest), then Sonnet, then Opus (most expensive), we achieve 84% accuracy at 38% of single-model cost—demonstrating that the FrugalGPT paradigm extends to scientific diagram interpretation tasks.

Subsequent work has refined this approach: cost-aware contrastive routing [39], dynamic model cascading [29], CascadeDebate [12], xRouter [36], and CASTER [26] collectively indicate that cost-aware agent orchestration is transitioning from research curiosity to production necessity. *QCV-Dataset* contributes to this ecosystem by providing a scientific-domain benchmark where cost-accuracy tradeoffs can be rigorously quantified through unitary fidelity verification—an objective ground truth absent from many text-generation evaluation scenarios.

2.4 Quantum Circuit Verification

Our verification methodology draws on established work in quantum circuit equivalence checking. The gold standard is comparison of unitary matrix representations [8, 42]. Our pipeline computes the full unitary matrix by running all 2^n computational basis states through Amazon Braket's LocalSimulator [4] and computes fidelity as $F = |\mathrm{Tr}(U_{\mathrm{gt}}^{\dagger} U_{\mathrm{gen}})|/2^n$. This approach is more rigorous than state-vector comparison on a single input because it verifies correctness across all possible input states simultaneously. The threshold $F \geq 0.99$ is robust: sensitivity analysis shows identical pass rates at $F \geq 0.95$ and $F \geq 0.99$, with a bimodal distribution where circuits are either nearly perfect ($F > 0.999$) or clearly wrong ($F < 0.95$).

2.5 Positioning and Contributions

Large language models (LLMs) have shown promise in hardware design automation. Projects such as Chip-Chat [7], VeriGen [40], GPT4AIGChip [16], ChipNemo [25], and ChatEDA [47] have explored using LLMs to generate HDL code, EDA scripts, and accelerator designs from textual specifications. Multimodal models such as GPT-4 [32] and LLaVA [24] have further demonstrated strong visual reasoning capabilities across domains including mathematical problem solving [27]. Most relevant to our work, VGV [45] extended this line of research to the visual domain, showing that MLLMs can generate Verilog code from classical circuit diagrams. VGV introduced Thinking Vision (TV) prompting (+50 percentage points (pp) for open-source models).

Quantum circuit diagrams possess properties that make them amenable to visual code generation: quantum gate symbols follow standardized conventions (e.g., H for Hadamard, $\oplus$ for CNOT targets, $\bullet$ for control qubits), a strictly left-to-right temporal structure with no feedback loops, and a finite gate vocabulary. These properties suggest MLLMs may achieve higher accuracy on quantum circuits than on classical ones—a hypothesis we confirm empirically.

Our contributions are:

(1) We construct **QCV-Dataset**: 132 quantum circuits across 13 categories (1–10 qubits) with five core modalities per circuit, publicly released with machine-readable citation (CITATION.cff) and dataset documentation following the Gebru *et al.* (2021) Datasheets for Datasets standard [17].
(2) We introduce a **cost-aware evaluation framework** for visual AI agents, analyzing cost-accuracy tradeoffs across model tiers and proposing a cascade routing strategy that achieves 84% accuracy at 38% of single-model cost.
(3) We provide **statistically rigorous evaluation** with $n = 5$ repeated trials, paired t-tests, and logistic regression confirming that circuit depth ($p < 0.001$)—not qubit count ($p = 0.20$)—predicts comprehension difficulty.
(4) We release QCV-Dataset (132 circuits, 5+ modalities, 1,931 files) as an **open** evaluation infrastructure on Hugging Face Hub with structured metadata for discoverability, machine-readable interoperability, and **responsible AI** documentation covering provenance, limitations, and known biases. All evaluation code, cost logs, and verification scripts are released for full reproducibility on GitHub. [2]

Our results show that Claude Sonnet 4.6 achieves the highest and most stable accuracy (91.4%±5.2% on the 21-circuit core benchmark under BV), followed by Opus 4.6 (85.7%±5.8%), at only 18% of Opus's per-call cost. Across the full 132-circuit benchmark, strong models reach 77–78% and Haiku 4.5 reaches 43–46%. Chain-of-thought prompting shows no significant effect for any model.

3 QCV-Dataset

We construct QCV-Dataset 1, a multimodal quantum circuit dataset of 132 circuits across 13 categories, with five core data modalities per circuit (images, executable code, simulation results, structured annotations, and failure cases) plus two supplementary modalities for selected circuits (natural language target descriptions and optimization equivalence pairs). We evaluate all 132 circuits across 13 categories. For the 21-circuit core subset (Basic, Intermediate, Advanced), we conduct $n = 5$ repeated trials to assess statistical significance and run-to-run variance. Circuit selection follows standard quantum computing curricula [31]. All circuit diagrams are generated programmatically using Qiskit's `matplotlib` drawer [37], ensuring reproducible and unambiguous visual inputs. Ground truth implementations are provided in both Amazon Braket SDK [4] and Qiskit.

As in Table 2, the full dataset spans 13 categories: basic gates (5), intermediate algorithms (10), advanced algorithms (6), blockchain protocols (11), gate-type coverage (15), qubit scaling 4–10q (12), classical algorithms (15), variational circuits (10), error correction codes (8), quantum ML (10), blockchain extended (8), visual variants

[2] URLs hidden for anonymous review.

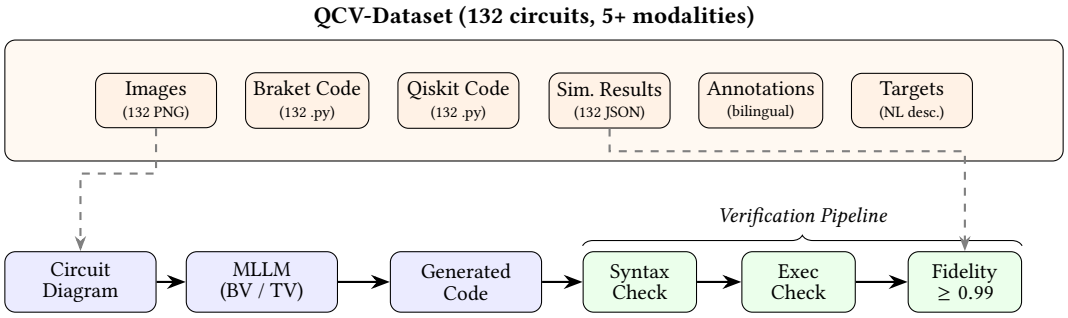


Figure 1: QCV system overview. Top: QCV-Dataset provides 132 quantum circuits with five core data modalities—circuit images, executable code (Braket and Qiskit), simulation results, bilingual annotations, and annotated failure cases—plus supplementary target descriptions and equivalence pairs. Bottom: The evaluation pipeline processes a circuit diagram image through an MLLM under BV or TV mode, then verifies the generated code through syntax checking, execution, and unitary matrix fidelity ($F \geq 0.99$).

(10), and BTC/blockchain quantum security (12). The 21-circuit core subset used for $n = 5$ repeated evaluation is drawn from the first three categories: 1) **Basic (5 circuits):** Single-gate or single-entangling-gate circuits testing fundamental gate recognition (Hadamard, CNOT, Bell state, GHZ state, Toffoli). 2) **Intermediate (10 circuits):** Multi-gate circuits requiring understanding of gate composition, control relationships, and parameterized rotations (SWAP decomposition, 2-qubit QFT, teleportation, Deutsch, superdense coding, Grover 2-qubit, parameterized rotation, Fredkin, shift register, phase estimation). 3) **Advanced (6 circuits):** Complete quantum algorithms involving complex multi-qubit structures (3-qubit QFT, 3-qubit Grover, VQE ansatz, QAOA, quantum walk, Bernstein-Vazirani).

Each circuit in the full dataset includes: (1) a PNG diagram image, (2) executable Braket SDK code, (3) executable Qiskit code, (4) simulation results (state vectors from LocalSimulator), (5) structured bilingual annotations (English/Chinese), and, where applicable, (6) a natural language target description for circuit synthesis tasks and (7) optimization equivalence pairs. Additionally, 268 annotated failure cases from our MLLM experiments are included for error analysis research. All structural annotations (qubit count, depth, gate types) are programmatically extracted from executable code; functional descriptions follow standard definitions from original algorithm papers [31]; category assignments follow established benchmark taxonomies [11, 35]. Table 2 summarizes the 13 categories.

3.1 Dataset Publication and Data Governance

To ensure broad accessibility, reproducibility, and interoperability across the machine learning and quantum computing research communities, we publish QCV-Dataset on *Hugging Face Hub*. The dataset is organized into four structured configurations: `circuits` (132 records with images, code, annotations, and state vectors), `experiments` (792 model invocation results with fidelity scores and pass/fail indicators), `failures` (27 annotated failure cases), and `equivalences` (12 circuit equivalence pairs). Users can load the dataset programmatically via the `datasets` library:

```
from datasets import load_dataset
circuits = load_dataset("QuantBlockchain/qcv-dataset",
                        "circuits", split="train")
```

This one-line interface enables immediate access to all 132 circuits with their five core modalities, eliminating the need for manual data parsing or local file management.

Data governance. Our dataset publication follows a layered metadata governance framework grounded in two complementary sets of principles. The **FAIR Principles** (Findable, Accessible, Interoperable, Reusable) [44] provide machine-actionable guidelines for scientific data management: data should carry unique persistent identifiers and rich metadata (Findable), be retrievable by both humans and machines (Accessible), use formal, shared knowledge representations (Interoperable), and include clear licenses and provenance documentation (Reusable). The **CARE Principles** (Collective Benefit, Authority to Control, Responsibility, Ethics) [10] complement FAIR with a people- and purpose-oriented focus, emphasizing that data practices should benefit the communities they affect, that those communities retain authority over their data, that data stewards act transparently and accountably, and that ethical considerations guide data use throughout its lifecycle. We employ three complementary standards, each anchored in specific FAIR and CARE principles: **1.Hugging Face Dataset Card (YAML)**: Structured YAML front matter provides task categorization (image-to-text, text-generation, visual-question-answering), language tags (English, Chinese), license information (MIT), and size annotations. This rich metadata enables automatic indexing by the Hugging Face search engine and downstream dataset aggregation tools, directly supporting FAIR's goals of making data *Findable* (through comprehensive, searchable metadata descriptions) and *Reusable* (through explicit licensing for downstream use). **2.Croissant JSON-LD [2]**: We provide machine-readable Croissant metadata—an emerging industry standard adopted by Google Dataset Search, Kaggle, and OpenML—that represents the dataset as a structured knowledge graph using JSON-LD. This formal, accessible representation ensures data is *Findable* (through globally unique persistent identifiers) and *Interoperable* (through a standardized vocabulary that enables cross-platform

Table 2: QCV-Dataset: 132 Circuits across 13 Categories

Category	Focus	N	Qubits	Depth	Example Circuits	Eval
Basic	Fundamental gates	5	1–3	1–3	Hadamard, Bell, Toffoli	✓
Intermediate	Multi-gate composition	10	2–4	2–6	QFT, Grover, Teleport	✓
Advanced	Complete algorithms	6	3–5	4–14	QAOA, VQE, BV	✓
Blockchain	Quantum protocols	11	2–8	1–9	BB84, VRF, Consensus	✓
A: Gate Cov.	All standard gate types	15	1–3	1–8	CRy, CCZ, iSWAP	✓
B: Qubit Sc.	4–10 qubit circuits	12	4–10	4–10	GHZ-10, QFT-5, Ring	✓
C: Classical	Textbook algorithms	15	2–4	3–8	Deutsch-Jozsa, Simon, Shor	✓
D: Variational	NISQ-era ansätze	10	2–4	3–27	HW-efficient, UCCSD	✓
E: Error Corr.	QEC codes	8	3–9	2–5	Shor-9, Steane-7, Surface	✓
F: Quantum ML	QNN, QCNN, kernels	10	2–8	1–13	QCNN-8q, QGAN, Kernel	✓
G: BC Ext.	Crypto protocols	8	3–6	2–7	E91, Voting, Auction	✓
H: Visual Var.	Layout robustness	10	2–4	1–11	Barrier, Decomposed, Reversed	✓
I: BTC Sec.	Threats & PQC defense	12	4–7	3–12	Shor vs ECDSA, Kyber	✓
Total		**132**	**1–10**	**1–27**		

discovery and automated integration into ML pipelines without manual schema mapping). **3.Croissant-RAI (Responsible AI) [2]**: We extend the Croissant schema with the MLCommons RAI vocabulary, documenting data collection methodology (Qiskit programmatic generation + expert bilingual curation), verification protocols (unitary matrix fidelity ≥ 0.99 on Braket LocalSimulator), known limitations (framework-specific to Braket SDK; simulation-only; EN/CN bilingual coverage), and dataset biases (23.5% blockchain-relevant circuits that may skew toward cryptographic applications). This addresses CARE's emphasis on *Collective Benefit* (documenting intended use cases for the research community), *Responsibility* (transparent reporting of limitations and biases), and *Ethics* (considering how downstream applications might be affected), while supporting FAIR *Reusability* through detailed provenance documentation that helps future users assess fitness for their purposes.

Implementation. To construct the HF-compatible dataset, we consolidated all five modalities into a unified Apache Parquet format using the `datasets` library with explicit `Features` schema declarations. Circuit diagram PNGs were embedded using the `Image()` feature type, which encodes image bytes directly into the Parquet files—this enables the Hugging Face Dataset Viewer to render circuit previews interactively in the browser. State vectors were stored as variable-length `Sequence(Value("float64"))` arrays to handle varying dimensions across qubit counts. We generated Croissant-RAI metadata as a JSON-LD overlay documenting provenance, limitations, and recommended use cases. The dataset was then pushed to the Hub via `push_to_hub()` with named configurations for each split, ensuring clean separation between circuit records and experimental results.

4 Evaluation Methodology

Following the approach established in VGV [45], we define two visual prompting modes and build an end-to-end verification pipeline tailored to quantum circuits.

4.1 Visual Modes

We evaluate two visual prompting strategies adapted from VGV [45]:

Basic Vision (BV) provides the circuit diagram image with a direct code generation instruction:

> *"Please generate Amazon Braket SDK Python code based on this quantum circuit diagram. Requirements: use* `from braket.circuits import Circuit`*; output only complete executable Python code; no explanation, no markdown."*

Thinking Vision (TV) adds a chain-of-thought analysis step before code generation:

> *"Please first analyze this quantum circuit diagram: 1. How many qubit lines? What are their labels? 2. What gates appear from left to right? 3. Which gates have control qubits? Which line is control, which is target? Then generate Amazon Braket SDK Python code based on your analysis."*

Unlike VGV, which also varied prompt detail levels (Short/Medium/High), we use a single prompt per mode to isolate the effect of chain-of-thought reasoning from prompt informativeness. Formally, given a circuit diagram image I and model $\mathcal{M}$:

$$\text{BV}: \quad c = \mathcal{M}(I,\ p_{\text{bv}}) \tag{1}$$

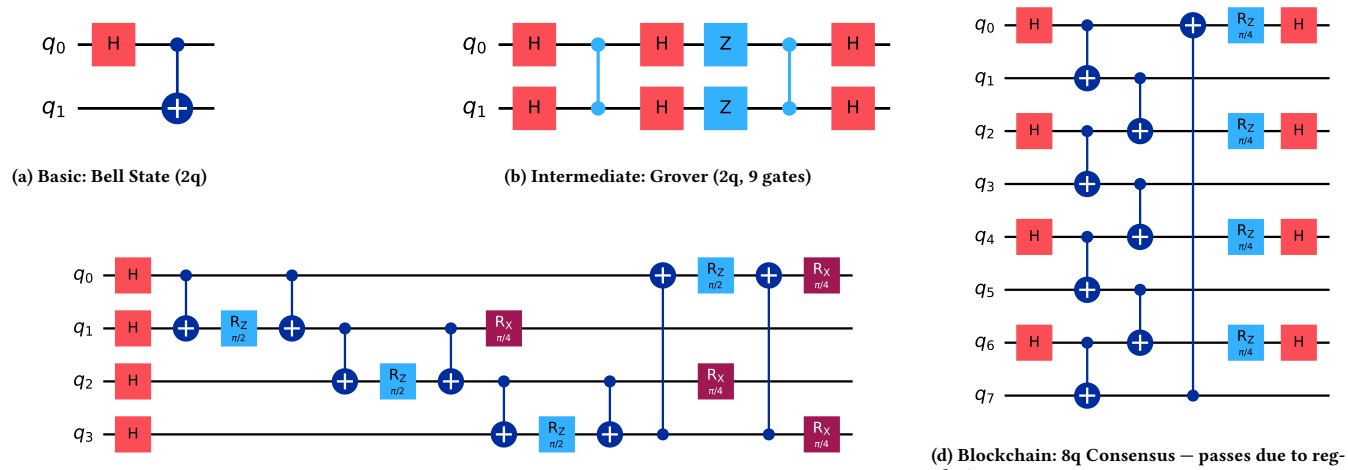
$$\text{TV}: \quad (a, c) = \mathcal{M}(I,\ p_{\text{tv}}) \tag{2}$$

where c is the generated code and a is the intermediate chain-of-thought analysis produced only under TV. Prompts are in Chinese; all evaluated models are multilingual, and preliminary tests showed no meaningful difference between Chinese and English prompts on this task.

4.2 Verification Pipeline

Each model invocation produces a raw text output containing ANSI formatting, tool invocation logs, and the generated code. We first extract the Python code using pattern matching (validated by an independent quality audit; see Appendix C), then verify it through a three-level pipeline:

(1) **Syntax check:** The extracted Python code is compiled using `compile()` to detect syntax errors before execution.
(2) **Execution check:** The code is executed in a sandboxed namespace with standard imports (`math`, `numpy`). We verify that a valid Braket `Circuit` object is produced and that no runtime exceptions occur (e.g., nonexistent API calls such as `.crz()` or `.toffoli()`).
(3) **Unitary fidelity check:** We compute the full unitary matrix of both the generated circuit and the ground truth by



(a) Basic: Bell State (2q)

(b) Intermediate: Grover (2q, 9 gates)

(c) Advanced: QAOA MaxCut (4q, 16 gates) — all models struggle

(d) Blockchain: 8q Consensus — passes due to regularity

Figure 2: Representative circuits. Left column (a–c): increasing complexity from trivial to challenging. Right (d): the largest circuit (8 qubits, 20 gates) passes under BV with perfect fidelity due to its regular repeating structure, while the smaller QAOA (c) fails—demonstrating that structural regularity, not qubit count, determines success.

running each of the 2^n computational basis states on Amazon Braket's LocalSimulator [4] (a free, local state-vector simulator) with `shots=0`. The fidelity is:

$$F = \frac{|\mathrm{Tr}(U_{\mathrm{gt}}^{\dagger} U_{\mathrm{gen}})|}{d}, \quad d = 2^n \tag{3}$$

where U_{gt} and U_{gen} are the $d \times d$ unitary matrices of the ground truth and generated circuits, respectively. A circuit passes if $F \geq 0.99$.

The unitary-based approach is more rigorous than state-vector comparison on a single input state (e.g., $|0\rangle^{\otimes n}$), as it verifies correctness across *all* possible input states simultaneously. It is also tolerant of global phase differences: two circuits that differ only by a global phase $e^{i\theta}$ will still achieve $F = 1.0$. The threshold $F \geq 0.99$ allows for minor floating-point imprecision while rejecting any structurally incorrect circuit. Sensitivity analysis confirms that conclusions are robust to threshold choice: pass rates are identical at $F \geq 0.95$ and $F \geq 0.99$, and only 1–2 circuits (of 792 invocations) fall between $F = 0.99$ and $F = 0.999$, indicating a bimodal fidelity distribution where circuits are either nearly perfect ($F > 0.999$) or clearly wrong ($F < 0.95$). No cloud quantum hardware is used; all verification runs locally. Because verification operates on the unitary (computed column-wise from state-vector simulation), it is not intrinsically Braket-specific: any SDK exposing a state vector or unitary—Qiskit Aer, Cirq, PennyLane—can be checked by the same oracle, with only a thin execution wrapper differing per SDK. The dataset already ships Qiskit ground truth; empirical cross-SDK verification is future work.

We report the **pass rate**—the fraction of circuits for which the generated code clears all three levels—per difficulty category and overall, following VGV [45]:

$$R = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}\left[F\left(U_{\mathrm{gt}}^{(i)},\ U_{\mathrm{gen}}^{(i)}\right) \geq 0.99\right] \tag{4}$$

where N is the number of circuits in the evaluation set.

5 Results and Scientific Findings

5.1 Experimental Setup

We evaluate three MLLMs from the Claude family, spanning high-end, mid-tier, and lightweight capability levels. All models are accessed through a unified command-line interface (`kiro-cli`) in non-interactive mode, ensuring identical execution conditions. Each circuit diagram is provided as a PNG image alongside the BV or TV prompt. All models use default temperature settings with no system prompts beyond the BV/TV templates. The total experiment comprises 132 circuits × 3 models × 2 visual modes = 792 invocations. We first report results on the 21-circuit core benchmark (Demo + Intermediate + Advanced) for comparison with prior work, then present the full 132-circuit evaluation across all 13 categories.

All generated code is verified on Amazon Braket's LocalSimulator (a free, local state-vector simulator) using the unitary fidelity pipeline described in Section 3. No cloud quantum hardware is used.

5.2 Results

Core benchmark (Table 3). With $n = 5$ repeated trials on the 21-circuit core benchmark, Sonnet emerges as the most accurate and stable model under BV (91.4%±5.2%), followed closely by Opus (85.7%±5.8%). Chain-of-thought prompting (TV) shows no statistically significant effect for any model: Opus $\Delta = -3.8$pp ($p = 0.26$), Sonnet $\Delta = -2.9$pp ($p = 0.19$), Haiku $\Delta = +4.8$pp ($p = 0.69$). Haiku

Table 3: Pass rate on the 21-circuit core benchmark ($n = 5$ repeated trials, mean±std %). Fidelity ≥ 0.99. BV = Basic Vision; TV = Thinking Vision (chain-of-thought).

	Opus 4.6 2.20× cr/call	Sonnet 4.6 1.30× cr/call	Haiku 4.5 0.40× cr/call
BV	85.7±6	**91.4±5**	48.6±4
TV	81.9±8	88.6±4	53.3±9
CoT Δ	−3.8pp	−2.9pp	+4.8pp
p-value	.26	.19	.69

scores substantially lower (48.6%±4.0% BV), confirming a clear capability gap. Notably, run-to-run variance is substantial (±4–9pp), demonstrating that single-run evaluations can be misleading.

Full benchmark (Table 4). Scaling to 132 circuits reveals a clear difficulty gradient. Basic circuits (Demo, Gate Coverage) are solved at 80–100% by all models. Medium circuits (Intermediate, Visual Variants, Qubit Scaling) remain above 90% for strong models but drop to 30–66% for Haiku. Hard circuits (Classical Algorithms through BTC/Security) challenge even the strongest models, with pass rates ranging from 41% to 87% for Opus. The overall pass rates—Opus 78%, Sonnet 77%, Haiku 43%—establish that these three Claude-family models can reliably interpret standard quantum circuits but face clear limits on complex algorithms and cryptographic protocols. We emphasize that the full 132-circuit results (Table 4), the cost analysis (Table 5), the failure taxonomy, and the cascade figure are single-run ($n = 1$) and should be read as indicative; our significance-tested claims are confined to the $n = 5$ core subset.

Table 4: Pass rate on 132-circuit benchmark by category (fidelity ≥ 0.99, $n = 1$). B =best-in-row; ≥75%; 60–74%; 50–59%; <50%. CoT Δ = TV−BV.

				Opus 4.6		Sonnet 4.6		Haiku 4.5		
Category	**Dif.**	**#**	**Q**	BV	TV	BV	TV	BV	TV	**Bst**
Demo	B	5	1–3	**100%**	**100%**	**100%**	**100%**	80%	40%	**100**
Gate Coverage	B	15	1–3	**86%**	80%	80%	80%	53%	53%	**86**
Intermediate	M	10	2–4	**90%**	**90%**	**90%**	**90%**	30%	60%	**90**
Visual Variants	M	10	2–4	**100%**	90%	**100%**	80%	50%	30%	**100**
Qubit Scaling	M	12	4–10	91%	91%	**100%**	91%	66%	66%	**100**
Advanced Algo.	H	6	3–5	**83%**	**83%**	**83%**	50%	33%	50%	**83**
Classical Algo.	H	15	2–4	**66%**	**66%**	60%	**66%**	26%	33%	**66**
Variational	H	10	2–4	70%	70%	80%	80%	50%	50%	**80**
Error Correction	H	8	3–9	**75%**	**75%**	62%	62%	12%	25%	**75**
Quantum ML	H	10	2–8	70%	80%	90%	**100%**	60%	80%	**100**
Blockchain	H	11	2–8	63%	54%	**81%**	**81%**	45%	45%	**81**
Blockchain Ext.	H	8	3–6	**87%**	**87%**	62%	62%	50%	62%	**87**
BTC/Security	H	12	4–7	**58%**	41%	33%	33%	16%	16%	**58**
Total		**132**	**1–10**	**78%**	75%	77%	75%	43%	46%	**78**
CoT Δ				▼ −4		▼ −3		▲ +5		

D=Difficulty (B=Basic, M=Med., H=Hard); Q=Qubits; Bst=Best(%); BV=Basic Vision; TV=Thinking Vision (CoT).

Of the 132 circuits, 45 are solved by all six model–mode combinations, while 18 defeat all of them. The 18 "impossible" circuits are predominantly complex algorithms (Shor, HHL, QAOA), error correction (surface code), and blockchain security protocols—circuits characterized by irregular multi-qubit entanglement patterns rather than high qubit count alone.

To formally test the "structural complexity, not qubit count" hypothesis, we fit logistic regression models predicting majority-pass (≥3/6 configurations pass) from circuit features. Circuit depth alone explains 13.2% of variance ($p < 0.001$), while qubit count alone explains only 4.6% ($p = 0.012$). In a joint model, depth remains significant (coef = −0.25, $p = 0.001$) while qubit count becomes non-significant (coef = −0.17, $p = 0.20$). This confirms that **circuit depth—a proxy for structural complexity—is the primary predictor of AI comprehension difficulty, not the number of qubits**. Here "primary predictor" denotes the strongest among the scalar structural features we tested, not a claim of high absolute explanatory power: depth alone accounts for 13.2% of deviance, and the remainder reflects category-specific factors (entanglement topology, algorithm family) not captured by scalar counts. We further note that depth and gate count are highly collinear (Pearson $r = 0.862$); "depth" should thus be read as a proxy for overall structural complexity rather than an isolated cause.

The strong-tier gap widens with difficulty: 27–47pp on Basic, 42pp on BTC/Security. All three models lie on the cost–accuracy Pareto frontier: none is strictly dominated. Haiku minimizes cost, Opus attains the highest full-set accuracy (79.4% vs. Sonnet 77.3%), and Sonnet occupies a favorable middle, capturing ∼97% of Opus's accuracy at ∼18% of its per-call cost. A per-circuit paired test on the core subset ($n = 21$) finds the Sonnet–Opus BV difference (+5.7pp) not statistically significant (paired t: $p = 0.083$; Wilcoxon: $p = 0.057$; 95% CI [−0.4, +11.8]pp), so we do not claim either model dominates the other in accuracy. Which model is "optimal" therefore depends on the deployer's stated cost–accuracy preference rather than being fixed by Pareto analysis alone; when both matter, we recommend Sonnet as a sensible default.

5.3 Finding 1: Chain-of-Thought Has No Significant Effect

Across the 21-circuit core subset with $n = 5$ repeated trials, chain-of-thought prompting (TV) shows no statistically significant effect for any model (Table 3). The full 132-circuit benchmark ($n = 1$) shows similar trends (Δ = −4, −3, +5pp; Table 4) but lacks statistical power for formal testing. For each circuit i, we compute the mean pass rate over five runs under each mode, then apply a paired t-test:

$$d_i = \bar{r}_i^{\mathrm{TV}} - \bar{r}_i^{\mathrm{BV}}, \quad t = \frac{\bar{d}}{s_d/\sqrt{21}}, \quad p = P(|T_{20}| > |t|) \tag{5}$$

where $\bar{r}_i^{\mathrm{BV}}$ and $\bar{r}_i^{\mathrm{TV}}$ are the mean pass rates for circuit i under BV and TV, respectively. As shown in Table 3, no model achieves a statistically significant CoT effect (all $p > 0.18$). The largest effect (+4.8pp for Haiku) has $p = 0.69$, far from significance.

This contrasts with VGV [45], where TV improved open-source model performance by up to 50pp on classical circuits. We attribute the difference to quantum circuits' structural properties: the tightly coupled multi-qubit operations (e.g., oracle–diffusion pairs in Grover, cost–mixer layers in QAOA) resist decomposition into sequential reasoning steps. The step-by-step analysis that TV enforces does not help—and may slightly hurt—because the model

must ultimately generate code that respects the circuit's holistic structure.

To distinguish absence of evidence from evidence of absence, we add a two-one-sided-tests (TOST) equivalence analysis (equivalence margin ±10pp, per-circuit paired, $n = 21$): for Opus ($\Delta = -3.8$pp, TOST $p = 0.037$) and Sonnet ($\Delta = -2.9$pp, TOST $p = 0.001$), TV and BV are statistically equivalent within ±10pp, whereas for Haiku ($\Delta = +4.8$pp, TOST $p = 0.33$) equivalence is not established. We therefore conclude that for the two stronger models chain-of-thought is statistically equivalent to base prompting, rather than universally unhelpful. While individual circuits may benefit from TV (Fig. 3), these gains are offset by equal losses on other circuits, resulting in no net effect. The simpler prompting mode achieves equal or better accuracy at lower token cost, and the choice of model (Sonnet vs. Haiku: 43pp gap) matters far more than the choice of prompting strategy (BV vs. TV: <5pp, not significant).

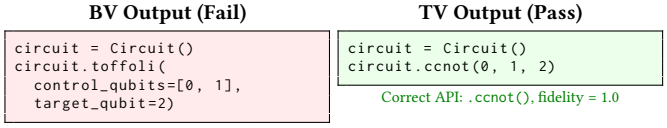


Figure 3: Haiku on the Toffoli circuit: BV uses a nonexistent API, while TV's analysis leads to the correct call. Although CoT shows no significant aggregate effect ($p = 0.69$), individual cases like this illustrate how structured analysis can resolve specific API selection errors.

5.4 Finding 2: Failure Mode Taxonomy

Across 792 invocations, we observe 524 passes and 268 failures. Failure modes stratify sharply by model capability:

Opus (28 BV failures): dominated by *low fidelity* (22/28)—the model generates syntactically valid, executable code that produces the wrong unitary. This indicates correct visual interpretation but incorrect gate parameters or ordering. Only 3 failures are syntax errors and 3 are execution errors.

Sonnet (30 BV failures): a mix of *low fidelity* (13/30) and *execution errors* (17/30). The execution errors typically involve nonexistent Braket API calls (e.g., `.cp()`, `.crz()`)—correct quantum operations expressed in the wrong SDK dialect.

Haiku (75 BV failures): dominated by *low fidelity* (54/75), with substantial *execution errors* (21/75). Unlike Opus, Haiku's fidelity failures often reflect fundamental visual misinterpretation—reversed CNOT directions, omitted gates, or incorrect qubit assignments—rather than subtle parameter errors.

5.5 Finding 3: Cost-Accuracy Tradeoff

Deploying AI agents for scientific tasks requires evaluating not only accuracy but also computational cost. We analyze the cost-accuracy relationship across five dimensions.

Per-model cost. Table 5 reports the per-invocation cost (in platform credits), latency, and cost-efficiency for each model under BV mode. We define cost-per-correct-circuit as:

Table 5: Cost-accuracy tradeoff (BV mode, 132 circuits). Credits are platform billing units; tiers reflect vendor capability-based pricing. MC = marginal cost per 1pp accuracy gain over the next-cheaper model.

Model	Tier	Cr/call	Time	Pass%	Cr/corr.	MC
Haiku 4.5	0.40×	0.031	3.7s	43%	0.072	–
Sonnet 4.6	1.30×	0.110	6.0s	77%	0.142	**0.002**
Opus 4.6	2.20×	0.618	24.4s	78%	0.778	0.508

$$C_{\text{eff}} = \frac{\sum_{i=1}^{N} c_i}{\sum_{i=1}^{N} \mathbb{1}[F_i \geq 0.99]} \tag{6}$$

where c_i is the credit cost of invocation i.

Diminishing returns. The marginal cost (MC) of accuracy improvement exhibits extreme diminishing returns. Upgrading from Haiku to Sonnet costs 0.079 additional credits per call for +34pp accuracy (MC = 0.002 credits/pp)—an excellent investment. Upgrading from Sonnet to Opus costs 0.508 additional credits for only +1pp (MC = 0.508 credits/pp)—a 254× higher marginal rate. This follows a classic diminishing-returns curve where the last percentage points of accuracy are disproportionately expensive.

Cost scales with difficulty. Opus's per-call cost increases 78% from Basic circuits (0.41 credits) to Hard circuits (0.73 credits), reflecting longer outputs and more tool invocations for complex circuits. Sonnet shows a smaller increase (39%), while Haiku remains nearly flat. Complex circuits thus impose a *double penalty*: lower accuracy *and* higher cost.

Cascade routing strategy. An agentic deployment can reduce cost by routing circuits through models of increasing capability. Using our actual experimental data:

(1) Run all 132 circuits with Haiku: 57 pass (43%), cost = 4.1 credits.
(2) Re-run 75 failures with Sonnet: 45 pass (60% rescue rate), cost = 8.3 credits.
(3) Re-run 30 remaining with Opus: 10 pass (33% rescue rate), cost = 18.5 credits.

This cascade achieves **112/132 = 84% pass rate at 30.9 credits** (38% of Opus-only cost of 81.6 credits), demonstrating that **intelligent routing is a key design dimension for cost-effective agentic AI systems**.

Return on investment (ROI) vs. human translation. Manual quantum circuit translation requires an estimated 1–4 hours per circuit by a trained researcher (based on author experience and informal surveys of quantum computing labs). At an estimated $50–200/circuit for human labor (assuming $50–100/hour researcher time), even the most expensive model (Opus: approximately $0.78/circuit at 78% accuracy) represents a >100× cost reduction. A hybrid workflow—AI generates, human verifies and fixes the 22% failures—reduces per-circuit cost from $100 to approximately $23 (77% savings), making large-scale circuit translation economically viable for the first time.

Comparison with VGV.. Our methodology extends VGV [45] from classical (Verilog) circuit diagram-to-code generation. Three key differences emerge: (1) quantum circuits yield higher baseline accuracy (78–91% vs. 40–60%) due to standardized gate symbols, strict left-to-right temporal structure, and finite gate vocabulary—properties absent in classical microarchitecture diagrams; (2) chain-of-thought effects diverge: TV improved VGV's open-source models by up to +50 pp, whereas we find no statistically significant effect on quantum circuits (all $p > 0.18$, $n = 5$), suggesting tightly-coupled multi-qubit structures resist step-by-step decomposition; (3) failure modes differ: VGV's failures are predominantly syntax errors in hardware description language (HDL) generation, while ours are dominated by low-fidelity outputs—syntactically valid code implementing the wrong unitary—indicating that semantic correctness, not syntactic fluency, is the core challenge.

5.6 Application: Quantum-Safe Blockchain

As a domain-specific case study, we evaluate 11 blockchain-relevant circuits (2–8 qubits) spanning quantum threats, post-quantum defenses, and quantum infrastructure. Detailed results are in Appendix A. Key finding: Sonnet + BV (9/11) outperforms Opus BV (7/11) on these circuits, reinforcing that the mid-tier model avoids over-analysis errors on circuits with parallel gate layers.

6 Discussion and Future Work

Our results establish that multimodal AI can reliably interpret quantum circuit diagrams today (77–91% accuracy depending on complexity). This positions QCV-Dataset as the foundation for a three-level research trajectory toward autonomous quantum circuit design: **Level 1 — Understanding (this paper):** AI reads circuit diagrams and generates verified executable code. We have demonstrated feasibility on 132 circuits across 13 categories with $n = 5$ repeated trials on a core subset, finding that structural complexity determines success and that chain-of-thought prompting does not significantly improve performance. The immediate next step is expanding evaluation to non-Claude models (GPT-4o, Gemini); **Level 2 — Optimization (near-term):** AI improves existing circuits. Using the equivalence pairs and target descriptions in QCV-Dataset, future systems can learn to suggest more efficient gate decompositions, reduce circuit depth, or adapt circuits to hardware constraints—functioning as a quantum protocol design co-pilot; **Level 3 — Invention (long-term):** AI designs novel quantum circuits from specifications, generating, verifying, and iterating autonomously. As a proposed design (not evaluated in this paper), we outline a Hybrid Agent architecture (Appendix D) that would combine QCV-Dataset as a retrieval-augmented generation (RAG) knowledge base, the unitary fidelity pipeline as a verification oracle, and the 268 annotated failure cases as negative examples. This closed-loop design-verify-refine cycle mirrors how human quantum engineers work, but operates at machine speed.

These results also clarify how task structure mediates prompting effectiveness. The finding that CoT has no significant effect on quantum circuit understanding—in contrast to its strong benefits for classical circuits [45]—suggests that structured reasoning strategies must be matched to task structure. For tasks requiring holistic visual pattern matching over tightly coupled components, step-by-step decomposition may be neither helpful nor harmful.

Implications for agentic AI evaluation: The cost-aware framework generalizes to any scientific diagram domain with verifiable ground truth. Any domain where AI agents perform scientific diagram interpretation (e.g., chemical structures, biological pathways, engineering schematics) faces the same cost-accuracy tradeoff. The cascade routing strategy and diminishing-returns analysis provide a template for evaluating agentic systems in cost-constrained deployments.

Limitations and Future Directions: **1. Model coverage.** We deliberately hold the vendor fixed to isolate the capability–cost tier effect within one model family, evaluating three Claude-family tiers via `kiro-cli`; other models on the same gateway (DeepSeek, MiniMax, GLM) lack multimodal image input. The pipeline is model-agnostic—any model with image input and code output can be plugged in—so cross-vendor evaluation (GPT-4o, Gemini, Qwen-VL) is a natural direction for future work. **2. Visual diversity and verification portability.** All circuit diagrams are rendered by Qiskit's matplotlib drawer; real-world circuits vary widely (TikZ, Quirk, hand-drawn). Our Visual Variants category (H) tests layout robustness with 90–100% pass rates for strong models, but full heterogeneity testing across renderers remains future work. While our unitary-fidelity oracle is not intrinsically Braket-specific (it operates on the circuit unitary computed via simulation), in this work we only ran the end-to-end verification pipeline through Braket's execution wrapper; empirical cross-SDK verification (e.g., Qiskit Aer, Cirq, PennyLane) is future work. **3. Beyond translation.** This paper addresses Level 1—visual comprehension and verified code generation. The QCV-Dataset already contains equivalence pairs and natural language targets to support Level 2 (optimization) and Level 3 (invention). Each repeated trial is a full paid LLM invocation ($n = 5$ over all 132 circuits ×3 models×2 modes is ~3,960 calls), so we prioritized statistical depth on a representative 21-circuit core; broader repeated-trial coverage is a natural direction for future work. More broadly, the cost-aware methodology generalizes to any scientific diagram domain with verifiable ground truth—chemical structures, biological pathways, engineering schematics—positioning this work as foundational infrastructure for multimodal AI in science.

References

[1] R. Acharya et al. 2023. Suppressing quantum errors by scaling a surface code logical qubit. *Nature* 614 (2023), 676–680.

[2] Mubashara Akhtar, Omar Benjelloun, Costanza Conforti, Luca Foschini, Pieter Gijsbers, Joan Giner-Miguelez, Sujata Goswami, Nitisha Jain, Michalis Karamousadakis, Satyapriya Krishna, Michael Kuchnik, Sylvain Lesage, Quentin Lhoest, Pierre Marcenac, Manil Maskey, Peter Mattson, Luis Oala, Hamidah Oderinwale, Pierre Ruyssen, Tim Santos, Rajat Shinde, Elena Simperl, Arjun Suresh, Goeffry Thomas, Slava Tykhonov, Joaquin Vanschoren, Susheel Varma, Jos van der Velde, Steffen Vogler, Carole-Jean Wu, and Luyao Zhang. 2024. Croissant: A Metadata Format for ML-Ready Datasets. In *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (Eds.), Vol. 37. Curran Associates, Inc., 82133–82148. doi:10.52202/079017-2610

[3] Yuri Alexeev, Marwa H. Farag, Taylor L. Patti, Mark E. Wolf, Natalia Ares, Alán Aspuru-Guzik, Simon C. Benjamin, Zhenyu Cai, Shuxiang Cao, Christopher Chamberland, Zohim Chandani, Federico Fedele, Ikko Hamamura, Nicholas Harrigan, Jin-Sung Kim, Elica Kyoseva, Justin G. Lietz, Tom Lubowe, Alexander McCaskey, Roger G. Melko, Kouhei Nakaji, Alberto Peruzzo, Pooja Rao, Bruno Schmitt, and Timothy Costa. 2025. Artificial intelligence for quantum computing. *Nature Communications* 16, 10829 (2025). doi:10.1038/s41467-025-65836-3

[4] Amazon Web Services. 2024. Amazon Braket Developer Guide. https://docs.aws.amazon.com/braket/

[5] F. Arute et al. 2019. Quantum supremacy using a programmable superconducting processor. *Nature* 574 (2019), 505–510.

[6] Ville Bergholm et al. 2018. PennyLane: Automatic Differentiation of Hybrid Quantum-Classical Computations. *arXiv preprint arXiv:1811.04968* (2018). https://arxiv.org/abs/1811.04968

[7] J. Blocklove, S. Garg, R. Karri, and H. Pearce. 2023. Chip-Chat: Challenges and Opportunities in Conversational Hardware Design. In *Proc. ACM/IEEE MLCAD*. 1–6.

[8] Lukas Burgholzer and Robert Wille. 2022. Handling Non-Unitaries in Quantum Circuit Equivalence Checking. In *Proc. ACM/IEEE DAC*. 529–534. doi:10.1145/3489517.3530482

[9] Shuxiang Cao, Zijian Zhang, Abhishek Agarwal, Grace Bratrud, Niyaz R. Beysengulov, Daniel C. Cole, Alejandro Gómez Frieiro, Elena O. Glen, Hao Hsu, Gang Huang, Raymond Jow, Greshma Shaji, Tom Lubowe, Ligeng Zhu, Luis Mantilla Calderón, Nicola Pancotti, Joel Pendleton, Brandon Severin, Charles Etienne Staub, Sara Sussman, Antti Vepsäläinen, Neel Rajeshbhai Vora, Yilun Xu, Varinia Bernales, Daniel Bowring, Elica Kyoseva, Ivan Rungger, Giulia Semeghini, Sam Stanwyck, Timothy Costa, Alán Aspuru-Guzik, and Krysta Svore. 2026. QCal-Eval: Benchmarking Vision-Language Models for Quantum Calibration Plot Understanding. (2026). arXiv:2604.25884 https://arxiv.org/abs/2604.25884

[10] Stephanie Russo Carroll, Edit Herczog, Maui Hudson, Keith Russell, and Shelley Stall. 2021. Operationalizing the CARE and FAIR Principles for Indigenous Data Futures. *Scientific Data* 8 (2021), 108. doi:10.1038/s41597-021-00892-0

[11] M. Cerezo et al. 2021. Variational quantum algorithms. *Nature Reviews Physics* 3 (2021), 625–644.

[12] Ruiqi Chang, Daeun Kwon, Jihoon Lee, and Naveen Verma. 2026. CascadeDebate: Multi-Agent Deliberation for Cost-Aware LLM Cascades. *arXiv preprint arXiv:2604.12262* (2026). https://arxiv.org/abs/2604.12262

[13] Mohammad Chehimi and Walid Saad. 2022. Quantum Federated Learning with Quantum Data. In *Proc. IEEE ICASSP*. 8617–8621. doi:10.1109/ICASSP43922.2022.9746622

[14] Lingjiao Chen, Matei Zaharia, and James Zou. 2023. FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance. *arXiv preprint arXiv:2305.05176* (2023). https://arxiv.org/abs/2305.05176

[15] A. K. Fedorov, E. O. Kiktenko, and A. I. Lvovsky. 2018. Quantum Computers Put Blockchain Security at Risk. *Nature* 563 (2018), 465–467.

[16] Y. Fu et al. 2023. GPT4AIGChip: Towards Next-Generation AI Accelerator Design Automation via Large Language Models. In *Proc. ICCAD*.

[17] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2021. Datasheets for Datasets. *Commun. ACM* 64, 12 (2021), 86–92.

[18] Albert Gu, Karan Goel, and Christopher Re. 2022. Efficiently Modeling Long Sequences with Structured State Spaces. In *Proc. ICLR*.

[19] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[20] J. Jumper et al. 2021. Highly accurate protein structure prediction with AlphaFold. *Nature* 596 (2021), 583–589.

[21] Y. Kim et al. 2023. Evidence for the utility of quantum computing before fault tolerance. *Nature* 618 (2023), 500–505.

[22] Kaixin Li, Yuchen Tian, Qisheng Hu, Ziyang Luo, and Jing Ma. 2024. MMCode: Benchmarking Multimodal Large Language Models for Code Generation with Visually Rich Programming Problems. In *Proc. EMNLP (Findings)*.

[23] D. Liu, A. Zhang, and L. Zhang. 2026. Quantum-Safe, Efficient, and AI-Enhanced Blockchains for the Web. In *Companion Proc. ACM Web Conference (WWW'26)*.

[24] H. Liu, C. Li, Q. Wu, and Y. J. Lee. 2023. Visual Instruction Tuning. In *Proc. NeurIPS*.

[25] M. Liu et al. 2023. ChipNemo: Domain-Adapted LLMs for Chip Design. *arXiv preprint arXiv:2311.00176* (2023).

[26] Shu Liu, Xin Yuan, Tian Chen, Zhiwei Zhan, Zhu Han, Da Zheng, et al. 2026. CASTER: Breaking the Cost-Performance Barrier in Multi-Agent Orchestration via Context-Aware Strategy for Task Efficient Routing. *arXiv preprint arXiv:2601.19793* (2026). https://arxiv.org/abs/2601.19793

[27] P. Lu et al. 2023. MathVista: Evaluating Mathematical Reasoning of Foundation Models in Visual Contexts. *arXiv preprint arXiv:2310.02255* (2023).

[28] McKinsey & Company. 2024. Quantum Technology Monitor: How Quantum Computing Could Change the World. https://www.mckinsey.com/featured-insights/themes/how-quantum-computing-could-change-the-world Estimates $450–850B economic value by 2040.

[29] Yasmin Moslem and John D. Kelleher. 2026. Dynamic Model Routing and Cascading for Efficient LLM Inference: A Survey. *arXiv preprint arXiv:2603.04445* (2026). https://arxiv.org/abs/2603.04445

[30] Akira Nakayama, Kosuke Mitarai, Leonardo Placidi, Takanori Sugimoto, and Keisuke Fujii. 2025. VQE-Generated Quantum Circuit Dataset for Machine Learning. *Physical Review Research* 7, 3 (2025), 033021. doi:10.1103/PhysRevResearch.7.033021

[31] M. A. Nielsen and I. L. Chuang. 2010. *Quantum Computation and Quantum Information* (10th anniversary ed.). Cambridge University Press.

[32] OpenAI. 2023. *GPT-4 Technical Report*. Technical Report.

[33] Elijah Perrier, Akram Youssry, and Chris Ferrie. 2022. QDataSet: Quantum Datasets for Machine Learning. *Scientific Data* 9, 1 (2022), 582. doi:10.1038/s41597-022-01639-1

[34] Leonardo Placidi, Ryuichiro Hataya, Toshio Mori, Koki Aoyama, Hayata Morisaki, Kosuke Mitarai, and Keisuke Fujii. 2026. MNISQ: A Large-Scale Quantum Circuit Dataset for Machine Learning in the NISQ Era. *Scientific Data* 13 (2026), 810. doi:10.1038/s41597-026-07493-9

[35] J. Preskill. 2018. Quantum Computing in the NISQ era and beyond. *Quantum* 2 (2018), 79.

[36] Cheng Qian, Zhibin Liu, Srajan Kokane, Akshat Prabhakar, Jiaxin Qiu, et al. 2025. xRouter: Training Cost-Aware LLMs Orchestration System via Reinforcement Learning. *arXiv preprint arXiv:2510.08439* (2025). https://arxiv.org/abs/2510.08439

[37] Qiskit Contributors. 2023. Qiskit: An Open-Source Framework for Quantum Computing.

[38] Louis Schatzki, Andrew Arrasmith, Patrick J. Coles, and M. Cerezo. 2021. Entangled Datasets for Quantum Machine Learning. *arXiv preprint arXiv:2109.03400* (2021). https://arxiv.org/abs/2109.03400

[39] Reza Shirkavand, Shuzhi Gao, Philip Yu, et al. 2025. Cost-Aware Contrastive Routing for LLMs. In *Proc. NeurIPS*.

[40] Shailja Thakur, Baleegh Ahmad, Hammond Pearce, Benjamin Tan, Brendan Dolan-Gavitt, Ramesh Karri, and Siddharth Garg. 2024. VeriGen: A Large Language Model for Verilog Code Generation. *ACM Trans. Des. Autom. Electron. Syst.* 29, 3, Article 46 (April 2024), 31 pages. doi:10.1145/3643681

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Proc. NeurIPS*, Vol. 30.

[42] George F. Viamontes, Igor L. Markov, and John P. Hayes. 2007. Checking Equivalence of Quantum Circuits and States. In *Proc. IEEE/ACM ICCAD*. 69–74. doi:10.1109/ICCAD.2007.4397246

[43] Haoyu Wang, Xianglin Zhou, Zihan Xu, Kun Cheng, Yixin Zuo, Kai Tian, et al. 2025. CODE-VISION: Evaluating Multimodal LLMs Logic Understanding and Code Generation Capabilities. *arXiv preprint arXiv:2502.11829* (2025). https://arxiv.org/abs/2502.11829

[44] Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J.G. Gray, Paul Groth, Carole Goble, Jeffrey S. Grethe, Jaap Heringa, Peter A.C 't Hoen, Rob Hooft, Tobias Kuhn, Ruben Kok, Joost Kok, Scott J. Lusher, Maryann E. Martone, Albert Mons, Abel L. Packer, Bengt Persson, Philippe Rocca-Serra, Marco Roos, Rene van Schaik, Susanna-Assunta Sansone, Erik Schultes, Thierry Sengstag, Ted Slater, George Strawn, Morris A. Swertz, Mark Thompson, Johan van der Lei, Erik van Mulligen, Jan Velterop, Andra Waagmeester, Peter Wittenburg, Katherine Wolstencroft, Jun Zhao, and Barend Mons. 2016. The FAIR Guiding Principles for Scientific Data Management and Stewardship. *Scientific Data* 3 (2016), 160018. doi:10.1038/sdata.2016.18

[45] S.-Z. Wong, G.-W. Wan, D. Liu, and X. Wang. 2024. VGV: Verilog Generation using Visual Capabilities of Multi-Modal Large Language Models. In *Proc. IEEE/ACM DAC*.

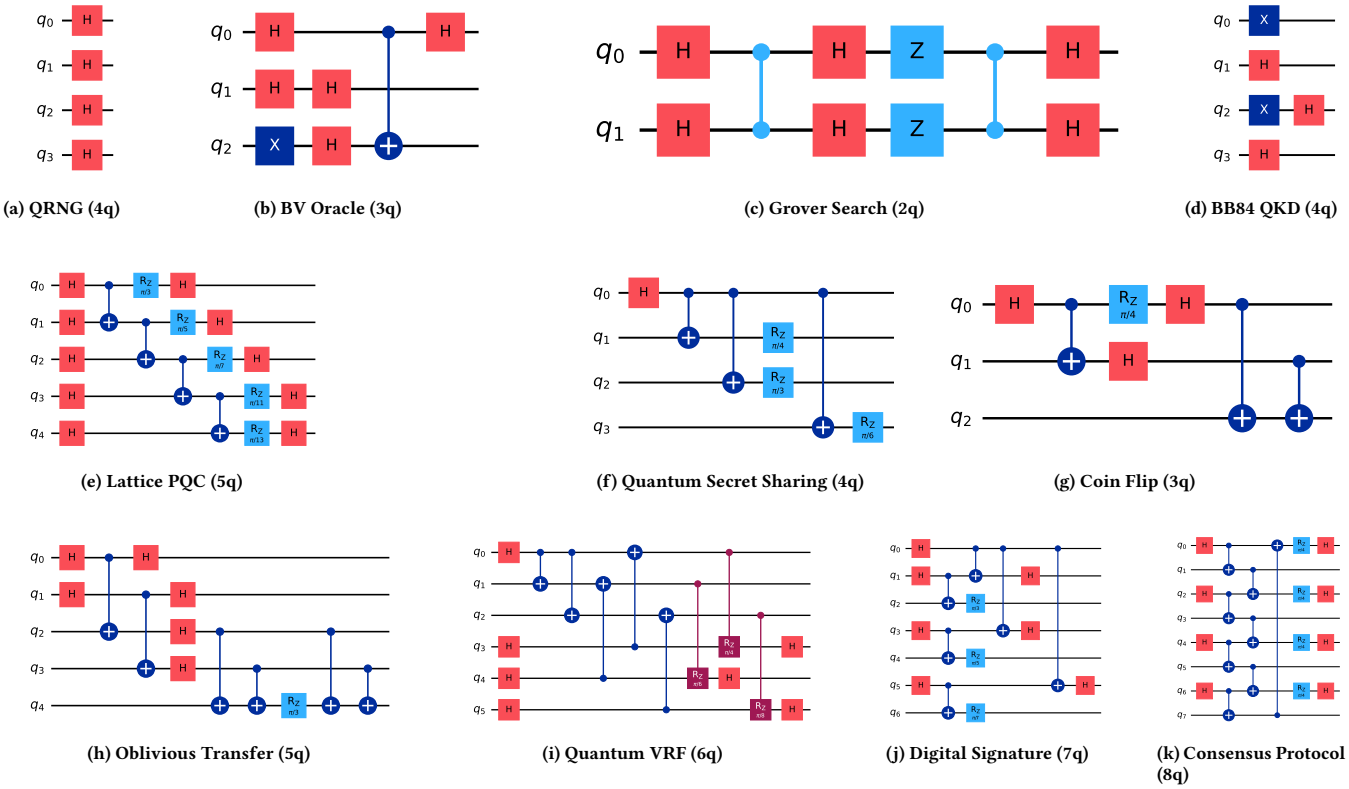
[46] Chengyue Wu, Zhenyu Liang, Yuchen Ge, Qi Guo, Zhiqiang Lu, Jing Wang, et al. 2025. Plot2Code: A Comprehensive Benchmark for Evaluating Multi-Modal Large Language Models in Code Generation from Scientific Plots. In *Proc. NAACL (Findings)*.

[47] Haoyuan Wu, Zhuolun He, Xinyun Zhang, Xufeng Yao, Su Zheng, Haisheng Zheng, and Bei Yu. 2024. ChatEDA: A Large Language Model Powered Autonomous Agent for EDA. *Trans. Comp.-Aided Des. Integ. Cir. Sys.* 43, 10 (Oct. 2024), 3184–3197. doi:10.1109/TCAD.2024.3383347

[48] Xinyang Zhao, Xianglin Luo, Qipeng Shi, Cheng Chen, Shuo Wang, et al. 2025. ChartCoder: Advancing Multimodal Large Language Model for Chart-to-Code Generation. In *Proc. ACL*.

A Blockchain Circuit Diagrams

Figure 4 shows the eleven blockchain-relevant quantum circuits evaluated in Section 4. Circuits range from 2 qubits (Grover Search) to 8 qubits (Consensus Protocol), spanning six blockchain security functions: consensus randomness, cryptographic key analysis, quantum key distribution, secret sharing, fair protocols, and post-quantum defense.



(a) QRNG (4q) (b) BV Oracle (3q) (c) Grover Search (2q) (d) BB84 QKD (4q)

(e) Lattice PQC (5q) (f) Quantum Secret Sharing (4q) (g) Coin Flip (3q)

(h) Oblivious Transfer (5q) (i) Quantum VRF (6q) (j) Digital Signature (7q) (k) Consensus Protocol (8q)

Figure 4: Eleven blockchain-relevant quantum circuits (2–8 qubits) spanning consensus, cryptography, threat modeling, key distribution, secret sharing, fair protocols, private exchange, verifiable randomness, digital signatures, and validator agreement.

Table 6: Blockchain circuit results. R=Regular, I=Irregular, P=Parallel. ✓=pass ($F \geq 0.99$).

Circuit	Qubits	Gates	Struct.	Opus BV	Opus TV	Son. BV
QRNG	4	4	R	✓	✓	✓
BV Oracle	3	5	R	✓	✓	✓
Grover	2	9	I	0.50	0.25	✓
BB84	4	4	R	✓	✓	✓
Lattice PQC	5	15	P	0.00	0.50	✓
QSS	4	7	R	✓	✓	✓
Coin Flip	3	8	R	✓	✓	✓
Oblivious Xfer	5	12	I	✓	✓	✓
VRF	6	15	I	0.24	0.00	0.00
QDS	7	16	I	0.25	0.00	0.50
Consensus	8	20	R	✓	0.00	✓
Pass				**7/11**	**6/11**	**9/11**

B Circuits That Defeat All Models

Table 7 lists the 18 circuits that failed across all six model–mode combinations. The dominant failure mode is low fidelity (syntactically valid code that produces the wrong unitary), indicating that models can parse these circuits visually but cannot correctly reconstruct their complex gate sequences. The circuits share two structural properties: irregular multi-qubit entanglement patterns and algorithm-specific gate orderings that do not follow repeating templates.

C Code Extraction Quality Audit

To ensure the reliability of our automated code extraction pipeline (which parses model outputs containing ANSI escape codes, diff-style formatting, and tool invocation logs), we performed an independent quality audit. Each extracted code snippet was checked for: (1) presence of the `from braket` import, (2) presence of a `Circuit()` constructor call, (3) balanced parentheses, (4) absence of diff-line residue or tool output contamination. On the full 132-circuit benchmark ($n = 1$, 792 files), 792/792 (100%) passed all checks. On the $n = 5$ repeated trials (630 files), 628/630 (99.7%) passed; the two flagged files (Opus BV, inter_02_qft2, runs 3–4) had missing import lines and were counted as failures in our analysis.

D Hybrid Agent Architecture

Figure 5 shows the proposed Hybrid Agent for autonomous quantum circuit design. The agent uses QCV-Dataset as a RAG knowledge base, the unitary fidelity pipeline as a verification oracle, and the 268 annotated failure cases as negative examples. This architecture is proposed but not yet implemented.

Table 7: The 18 circuits that defeat all models (0/6 pass). Dominant error: LF = low fidelity, EX = execution error, SY = syntax error.

Circuit	Category	Dominant Error
A10_controlled_rx	Gate Coverage	LF (0.50–0.93)
A11_ccz	Gate Coverage	LF / EX
inter_10_phase_est	Intermediate	EX (.crz API)
adv_04_qaoa	Advanced	LF (0.06–0.50)
C05_shor_period_4	Classical Algo.	LF (0.25–0.63)
C08_hhl_3	Classical Algo.	EX (.cp API)
C10_w_state_3	Classical Algo.	SY / EX
C11_w_state_4	Classical Algo.	SY / EX
D04_qaoa_2layer	Variational	LF (0.02–0.36)
E05_surface_code	Error Correction	LF (0.003–0.13)
G05_quantum_auction	Blockchain Ext.	LF (0.13–0.75)
I01_shor_ecc_6	BTC/Security	LF / EX
I03_grover_aes_5	BTC/Security	LF (0.00–0.25)
I09_dilithium_sign	BTC/Security	LF (0.25–0.50)
I10_pow_speedup_4	BTC/Security	LF (0.00–0.50)
I12_sphincs_hash_7	BTC/Security	LF (0.06–0.38)
blockchain_qds	Blockchain	LF / SY
blockchain_vrf	Blockchain	LF / EX

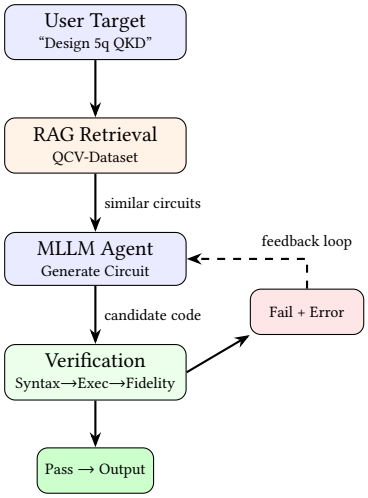


Figure 5: Proposed Hybrid Agent architecture (not yet implemented).