AutoFid: Adaptive and Noise-Aware Fidelity Measurement for Quantum Programs via Circuit Graph Analysis

Tingting Li[1,2], Ziming Zhao[1,*], Jianwei Yin[1,*]
[1]Zhejiang University, Hangzhou, China
[2]Shanghai Qi Zhi Institute, Shanghai, China
{litt2020, zhaoziming}@zju.edu.cn, zjuyjw@cs.zju.edu.cn

***Abstract*—Quantum computers in the Noisy Intermediate-Scale Quantum (NISQ) era face significant challenges due to inherent noise and limited qubit coherence. Accurate fidelity evaluation of quantum states necessitates multiple repeated measurements to obtain statistical results. But determining the optimal number of measurements remains an open problem due to the dynamic, device-dependent nature of quantum noise. Existing methods either assume prior knowledge of the noise model or inherently employ a fixed measurement strategy, which limits their applicability in practical deployment scenarios. This paper presents AutoFid, an adaptive and noise-aware fidelity measurement framework that automatically determines the number of required tests based on circuit structure and hardware feedback. AutoFid models quantum circuits as Directed Acyclic Graphs and estimates structural complexity via random walks, enabling estimation of measurement effort. It further incorporates transpilation-aware features such as gate fidelity, depth inflation, and crosstalk to refine iteration budgets. During runtime, AutoFid dynamically samples fidelity results and employs an early stopping strategy based on confidence intervals to reduce redundant measurements while preserving accuracy guarantees. We evaluate AutoFid on 18 quantum benchmarks executed on real IBMQ hardware platforms. Experimental results show that AutoFid reduces measurement costs by more than 50% compared to both fixed-shot and learning-based baselines, while consistently maintaining fidelity bias below 0.01. Additional analysis using classical software testing metrics and ablation studies demonstrate its effectiveness, robustness, and adaptability across a wide range of quantum workloads.**

***Index Terms*—Adaptive fidelity estimation, Confidence intervals, Mixing time, Quantum software testing.**

I. Introduction

Quantum computing, leveraging quantum-mechanical principles such as superposition and entanglement, holds the potential to transform fields including optimization [1], [2], cryptography [3], [4], simulation [5], [6], [7], and material science [8]. Applications to date span quantum simulations [9], quantum network [3], [10], machine learning [11], and cryptographic protocol analysis [12]. However, the field remains largely experimental, constrained by hardware challenges such as decoherence, gate errors, and scalability issues [9]. We are currently in the Noisy Intermediate-Scale Quantum (NISQ) era, where devices are limited by quantum noise [13], [14], restricted qubit counts, and the absence of full quantum error correction [15], making accurate computation and state evaluation particularly difficult in practice.

To enable reliable use of noisy quantum hardware, it is essential to characterize the performance of quantum circuits precisely. In quantum computing, *fidelity* is a widely used metric that quantifies how closely the outcome of a noisy quantum program execution matches its ideal, error-free behavior. A high fidelity indicates that the computation closely follows the intended quantum evolution, while low fidelity reflects stronger noise impact and unreliability. This metric underpins correctness evaluation, guides optimization, and supports cross-hardware comparison. State-of-the-art protocols such as randomized benchmarking (RB) [16] and cross-entropy benchmarking (XEB) [17] estimate fidelity by repeatedly executing predefined quantum circuits and analyzing outcome statistics [18]. However, these methods often require expertly chosen repetition counts empirically: too few runs yield inaccurate fidelity estimates, while too many waste scarce quantum resources. Prediction-based approaches relying on prior noise characterization [19], [20] are also impractical in practice, since real devices exhibit complex, unknown, and time-varying noise behaviors.

Consequently, there is a pressing need for systematic methods to determine the number of measurements in a resource-efficient yet accurate manner. Rather than relying on ad-hoc or fixed-shot strategies, we propose a data-driven noise-aware method to enable *adaptive fidelity measurement*. We conceptualize fidelity estimation as a *test planning problem* in the software engineering sense: allocating limited testing budgets, balancing accuracy against latency/costs, and adapting to time-varying hardware conditions. This perspective reveals three central challenges. (i) *Evolving noise environments.* Quantum noise is stochastic and time-varying [21], [22]. A fixed measurement budget that is effective at one time may become unreliable later due to noise drift under hardware calibrations. (ii) *Circuit-hardware interplay.* Fidelity is not only circuit-dependent but also shaped by device characteristics. Compilation, qubit mapping, and error mitigation steps [23] alter how circuits interact with hardware noise, yet we lack a general complexity indicator that reflects this interaction. (iii) *Accuracy-efficiency balance.*

[*] Corresponding Authors.

2643-1572/25/$31.00 ©2025 IEEE
DOI 10.1109/ASE63991.2025.00216

Different scenarios place conflicting demands: validation of fault-tolerant schemes requires highly precise fidelity estimates, while routine calibration or rapid benchmarking prioritizes speed. Strategies should therefore adapt measurement depth to the use case.

To address these challenges, we propose an adaptive fidelity measurement framework for quantum circuits, named AutoFid. Our approach systematically characterizes quantum circuits through a unified representation and complexity-driven analysis, enabling an optimized and statistically grounded benchmarking process. By modeling each circuit as a Directed Acyclic Graph (DAG) and leveraging a random-walk-based complexity estimator [24], [25], [26], AutoFid quantifies circuit structural difficulty and dynamically allocates measurement resources accordingly. The random-walk outcome is further combined with the backend-aware information and used to guide adaptive shot allocation, ensuring that circuits with higher topological or backend-induced complexity receive proportionally larger measurement budgets. Moreover, AutoFid introduces a *confidence-bound-aware measurement pipeline* that rigorously controls statistical uncertainty during measurement. The process operates over rolling index windows and continuously monitors the dispersion and convergence of fidelity estimates. Each window is validated for stability via exponentially weighted residual checks and bounded deviation criteria, while a configurable confidence level $(1-\alpha)$ governs the stopping condition.

Specifically, when the half-width of the confidence interval (halfCI) falls below a user-defined precision threshold δ and the effective sample size exceeds a guard minimum, the loop terminates early and reports a certified fidelity estimate with bounded error probability. This confidence-driven control is integrated with adaptive batch scheduling: batch sizes grow geometrically when far from the target, and switch to fine-grained linear increments as the confidence bound approaches δ, preventing overshooting. When instability is detected, a multiplicative backoff and window reset are triggered, ensuring robustness under backend fluctuations. Furthermore, an online shot forecasting module estimates the additional evidence required to achieve the desired $(\delta, 1-\alpha)$ precision, enabling transparent test budgeting and reproducibility.

By combining structural complexity estimation, backend-aware information enhancement, and confidence-controlled termination, AutoFid achieves both high measurement efficiency and statistically verifiable fidelity guarantees even against non-stationary noise drift[1].

In summary, this paper makes three key contributions.

- We formalize fidelity estimation for quantum programs as an adaptive testing problem, emphasizing efficiency, stability, and cross-hardware robustness.
- We present AutoFid, a novel framework that combines random walk-based DAG analysis with transpilation-aware complexity modeling and confidence-bound convergence guarantees, enabling accurate and resource-efficient fidelity measurement.
- We evaluate AutoFid on 18 diverse quantum benchmarks across IBM quantum devices, demonstrating comparable accuracy with >50% fewer measurements, robustness under noise drift, and negligible runtime overhead. We also conduct a series of experiments in terms of effectiveness, redundancy, and stability evaluations. The code is available on the online repository[2].

II. Background and Related Work

A. Quantum vs. Classical Computing

The fundamental distinction between quantum and classical computing lies in how they represent and manipulate information. Classical computers operate on bits, which can only take the values 0 or 1, and apply deterministic logic gates to process them in a sequential or parallel fashion. Quantum computers, by contrast, use qubits that can exist in a linear combination of both 0 and 1, known as a superposition. A qubit's state can be expressed as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where the squared amplitudes $|\alpha|^2$ and $|\beta|^2$ define measurement probabilities [9]. In addition, qubits can exhibit *entanglement*, creating correlations that have no classical counterpart, and *interference*, which allows quantum algorithms to amplify correct outcomes while canceling erroneous ones. By harnessing these principles, quantum algorithms can evaluate many computational paths simultaneously, offering the potential for dramatic speedups on tasks such as quantum network [27], optimization [28], [29], [30], and quantum simulation [11], [12].

The Noisy Intermediate-Scale Quantum (NISQ) era denotes near-term quantum processors that possess a moderate number of qubits but lack scalable fault-tolerant error correction. Such devices exhibit non-negligible gate and readout errors, limited coherence times and connectivity, and substantial calibration and control overheads. [13], [14], [15]. While NISQ devices already enable early progress in simulation, optimization, and cryptography [9], [31], they exhibit highly variable behavior due to device-specific and time-varying noise. To obtain statistically meaningful results, quantum programs must be executed repeatedly (so-called *shots*), much like running multiple test cases to reduce variance in software evaluation. However, excessive measurement shots consume valuable runtime and compute resources on quantum devices, raising the question of how to allocate measurement budgets under competing constraints such as task execution queueing delays, hardware calibration overheads, and measurement accuracy requirements. [32], [33].

B. Fidelity Estimation as Performance Testing

Fidelity estimation [34] is central to validating quantum hardware performance, as it measures how closely noisy executions align with ideal theoretical calculation outcomes [35], [36], [37]. Canonical benchmarking protocols form the foundation of this effort. For example, randomized benchmarking (RB) [16] applies sequences of randomly chosen Clifford gates,

[1] This noise-aware scheme adapts to runtime measurement behavior in real time, allowing statistically certified termination under dynamically varying quantum noise.

[2] https://github.com/Secbrain/AutoFid

measuring how the survival probability decays with sequence length, which yields an average error rate insensitive to state-preparation and measurement (SPAM) errors. Cross-entropy benchmarking (XEB) [17] is widely used in quantum experiments, compares the measured output distribution of random circuits with the ideal theoretical distribution, providing a direct measure of circuit fidelity. Related statistical analyses [18] formalize how repeated measurement outcomes converge to reliable fidelity estimates.

Beyond these protocols, recent research seeks to lower the cost of fidelity estimation. Learning-based predictors leverage machine learning to infer fidelity without exhaustive measurements. For example, Zhang *et al.* [38] proposed direct fidelity estimation from reduced data, while Yu *et al.* [39] and Liu *et al.* [40] developed statistical and reliability-based models. Wang *et al.* [41] and Tan *et al.* [19] advanced hardware-aware predictors that adapt to real-device data. Variational schemes [42], [43], [44] treat fidelity as an optimization objective, adjusting circuits or measurement settings to minimize resource use. Finally, classical-shadow techniques [37] employ randomized measurements and compressed sensing to approximate fidelities of many observables simultaneously, substantially reducing measurement overhead. Together, these efforts illustrate a trajectory from brute-force repetition toward more adaptive, predictive, and resource-efficient fidelity testing.

C. *Indicators of Circuit-Hardware Complexity*

Determining how many measurements are required for a given quantum circuit depends on its effective "difficulty" after it is transpiled and mapped to a specific hardware backend [23]. Transpilation involves gate decomposition, qubit mapping, and routing, all of which alter noise exposure and execution cost. Traditional metrics include circuit depth and two-qubit gate counts, which are correlated with accumulated error due to gate imperfections. Others leverage the coupling-graph structure, capturing the hardware's qubit connectivity and constraints. Some sophisticated graph-theoretic approaches introduce features such as random-walk characteristics. For instance, Tong *et al.* [25] and Li *et al.* [24] studied random walks to capture structural properties of graphs, while Craswell *et al.* [26] applied random-walk-based similarity measures in ranking contexts, which inspires analogous use for circuit connectivity analysis. On the hardware side, backend-aware features summarize device state, such as calibration data, error rates, and dynamic queue latency. These provide a real-time view of hardware variability. Recent works integrate these circuit features with learning-based regressors [41], [19], enabling predictive models that estimate the number of shots required to achieve a target confidence.

Our framework treats all these indicators as backend-aware signals rather than prescribing a single formula. Random-walk features, hardware-aware signals, and adaptive measurements can each be employed depending on available resources and robustness requirements, giving practitioners flexibility in balancing accuracy, cost, and adaptability to noise drift.

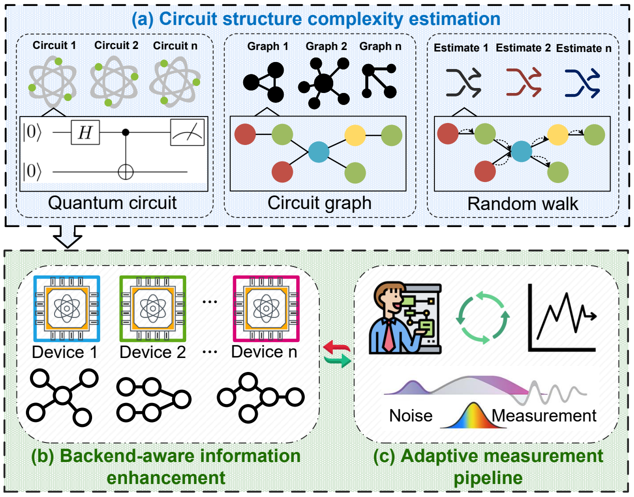


Fig. 1: The overview workflow of AutoFid.

D. *Motivations*

The above landscape motivates an adaptive approach guided by three core design principles. First, fidelity estimation must account for noise drift and prioritize stability in stopping criteria. Since quantum hardware noise is heterogeneous and non-stationary [45], estimators should monitor stability online and only accept measurements when estimates remain consistent over a stable window, rather than relying solely on a fixed-precision threshold. Second, measurement budgets should be scheduled hierarchically across circuits and devices. Instead of fixing repetitions per circuit, the system allocates measurements dynamically across benchmarks, backends, and time windows to maximize overall effectiveness under a global budget. Third, batching strategies should be latency-aware and informed by circuit-hardware complexity. Batch sizes should reflect end-to-end runtime considerations, including transpilation, queueing, and readout delays, and incorporate difficulty indicators from circuit complexity, hardware-aware, or learning-based metrics, with random-walk features used optionally.

These principles reframe fidelity estimation as a software engineering problem of test planning under uncertainty, involving adapting to noise drift, allocating scarce measurements where they are impactful, and coupling decision policies to practical runtime constraints. We propose an adaptive fidelity measurement framework for quantum circuits that combines DAG-based structural analysis, random-walk complexity estimation, backend information enhancement, and confidence-bound aware adaptive batching to allocate measurement resources while providing statistically certified fidelity guarantees dynamically.

III. Methodology

Figure 1 presents the overall workflow of AutoFid. Given an input quantum circuit, AutoFid first builds its circuit graph representation and applies random walks to estimate circuit complexity in Figure 1 (a). To further account for backend-specific noise, AutoFid incorporates transpilation-aware information enhancement in Figure 1 (b), which adjusts planning

Algorithm 1 Adaptive Measurement Pipeline

```
Input: Quantum circuit $Q$, backend $b$, target half-width $\delta$, confidence $1-\alpha$, window size $w$
Output: Estimate $\bar{F}_W$ with certified half-width $\leq \delta$
Build DAG $\mathcal{G}$ from $Q$ and transpile to $M$ on backend $b$
Compute structural and hardware descriptors $\rightarrow$ difficulty score $s_{\text{final}}$ (§ III-C)
Set initial batch size $P_0$ based on $s_{\text{final}}$
Initialize window $W \leftarrow \emptyset$, iteration counter $t \leftarrow 0$
while true do
    Run $P_t$ shots to obtain the batch estimate $\hat{F}_t$
    Update rolling window $W$ (keep last $w$ batches)
    Stability check (§ III-D). If residuals or deviations exceed threshold, reset $W$, reduce $P_{t+1}$, and continue
    Precision check (§ III-D). Compute mean $\bar{F}_W$, effective size $N_{\text{eff}}$, and half-width halfCI
    if halfCI $\leq \delta$ and $N_{\text{eff}}$ sufficient then
        return $(\bar{F}_W, \text{halfCI})$
    end if
    Adapt next batch size $P_{t+1}$, increase when far from $\delta$, taper growth near target
    $t \leftarrow t+1$
end while
```

based on device mappings and statistics. This complexity and backend-aware signals drive an adaptive measurement planner in Figure 1 (c), which dynamically allocates shots and applies an early-stopping rule. Together, these components form a noise-aware and resource-efficient fidelity estimation pipeline. By applying this principle to quantum circuits [25], we can achieve the desired accuracy level with a small number of measurement iterations. The algorithm is outlined in Algorithm 1. It iteratively updates measurement batches based on observed stability and precision, enabling AutoFid to balance accuracy and resource efficiency under varying circuit and hardware conditions.

A. Circuit Graph Representation

We represent the quantum circuits as a Directed Acyclic Graph (DAG) to facilitate subsequent modeling and analysis. In Figure 2, the Bernstein-Vazirani quantum circuit is converted into a DAG representation [46] to analyze the circuit's topological connections. Specifically, due to the existence of multi-qubit gates, it belongs to a Multi-Directed Graph (MultiDiGraph) from a more detailed perspective. In the MultiDiGraph, each quantum gate is represented as a node, while directed edges encode the execution dependencies between gates. Node attributes store information such as gate type, acting qubits, and parameters. For multi-qubit gates, the in-degree and out-degree reflect the number of involved qubits (*e.g.,* the *CX* gate in BV_3 has both in-degree and out-degree equal to 2). In this way, qubit flows are modeled as multiple directed edges connecting input and output ports, making the physical connectivity of the circuit more transparent [47]. Mathematically, we model each quantum circuit into a DAG $\mathcal{G} = (V, E)$, where $|V| = n$ nodes and the adjacency matrix $A \in \{0,1\}^{n\times n}$, where $a_{ij} = 1$ indicates a directed edge from node i (a gate operation) to node j. Unlike undirected graphs, A is not symmetric, as the quantum circuit enforces a strict temporal and dependency order, *i.e.,* once a gate is applied, the state cannot revert to the prior qubit configuration. This ensures the graph retains directionality and faithfully reflects the causal structure of the circuit. The degree matrix is defined as $D = \text{diag}(d_1, \ldots, d_n)$ with $d_i = \sum_{j=1}^{n} a_{ij}$ representing the out-degree of node i.

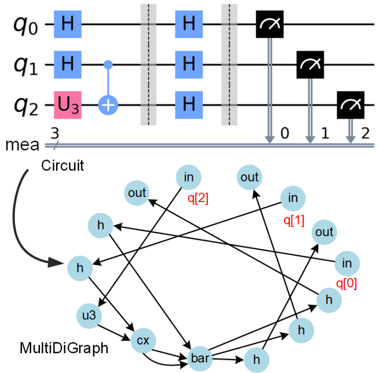


Fig. 2: The DAG conversion of the quantum circuit.

$$A = [a_{ij}], \quad a_{ij} = \begin{cases} 1 & \text{if there is an edge } i \rightarrow j, \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

$$D = \text{diag}(d_1, \ldots, d_n), \quad d_i = \sum_{j=1}^{n} a_{ij}. \tag{2}$$

These matrices form the basis for downstream random walk analysis. The random walk transition matrix is constructed as $Tran = D^{-1}A$, where each row of $Tran$ sums to one, representing the probability distribution of moving from node i to its directed successors in a single step.

B. Circuit Structure Complexity Estimation

To capture the inherent structural complexity of a quantum circuit, we employ a lightweight random walk algorithm over its DAG representation. Random walks are fundamental processes in the study of Markov chains and have applications in various fields, including physics, computer science, and network theory [48]. In graph theory and Markov chain analysis, a random walk is a stochastic process that transitions between nodes based on defined probabilities. A key property of such processes is the *mixing time*, which measures the number of steps required for the walk to approach a stationary distribution, where the probability of being at any given node becomes stable and no longer depends on the starting point. In the context of quantum circuits, this mixing time reflects how information or dependencies propagate through the gate structure. A smaller mixing time typically corresponds to circuits with shallow depth or weak inter-gate dependencies, while larger mixing times suggest more intricate entanglement patterns, broader gate fan-in/fan-out, or deeper computational paths. Therefore,

Algorithm 2 Mixing Time Estimation

```
Input: Adjacency matrix $A$, degree matrix $D$, tolerance $\epsilon$, max steps $S_{\max}$
Output: Estimated mixing time $t_{\text{mix}}$
$Tran = D^{-1}A$                                ▷ Transition matrix
$\pi_0 \leftarrow$ Uniform distribution over nodes
for $s = 1$ to $S_{\max}$ do                     ▷ Iteration steps $S$
    $\pi_s \leftarrow \pi_{s-1} Tran$
    $\Delta_s \leftarrow \max_v \|\pi_s(v) - \pi_{s-1}(v)\|_{\text{TV}}$    ▷ $\|\cdot\|_{\text{TV}}$ means Total Variation Distance, TVD.
    if $\Delta_s < \epsilon$ then
        return $t_{\text{mix}} \leftarrow s$
    end if
end for
return $t_{\text{mix}} \leftarrow S_{\max}$              ▷ Fallback if not converged
```

we treat the estimated mixing time t_{mix} as a compact indicator of circuit complexity.

As shown in Algorithm 2, which presents a method for estimating the mixing time of a Markov chain represented by a graph. Given the adjacency matrix A and degree matrix D, the algorithm first constructs the transition probability matrix $Tran = D^{-1}A$, which defines the random walk dynamics over the graph. The algorithm then initializes a uniform distribution for the random walk's starting positions across all nodes, then iteratively updates it via the transition matrix until convergence or a maximum iteration limit is reached. Specifically, starting from an initial uniform distribution π_0 over all nodes, the algorithm iteratively updates the state distribution by multiplying with $Tran$. At each step s, it computes the total variation distance Δ_s between consecutive distributions π_s and π_{s-1}. When this distance falls below a predefined tolerance ϵ, the algorithm terminates, returning the current step s as the estimated mixing time t_{mix}. If the maximum number of steps $S_{\max}$ is reached without convergence, $S_{\max}$ is returned as an upper bound estimate. This procedure provides a computationally efficient approach to assessing the convergence speed of stochastic processes modeled by graphs, which is critical for applications such as network analysis, probabilistic verification, and software reliability assessment.

C. Backend-Aware Information Enhancement

In addition to natively estimating circuit-intrinsic complexity, we incorporate *backend-aware* signals that arise from compilation and mapping onto a concrete quantum device. In other words, § III-B summarizes structural properties of the logical DAG, while this section enhances the characterization with hardware-specific transformations so that subsequent decisions reflect the *physical* execution context of the transpiled program.

Calibration and Mapping Signals. For a quantum backend b at a specific time, we collect a calibration snapshot

$$\mathbf{c}_b = \big[\{\epsilon_i^{(1\text{q})}\},\ \{\epsilon_{ij}^{(2\text{q})}\},\ \{r_i\},\ \{T_1(i)\},\ \{T_2(i)\},\ \ldots\big], \quad (3)$$

where $\epsilon_i^{(1\text{q})}$ and $\epsilon_{ij}^{(2\text{q})}$ are backend-reported 1-qubit/2-qubit gate error rates, r_i are readout errors, and T_1/T_2 coherence. Given the transpilation mapping logical → physical deployable quantum circuits and the post-transpilation circuit M, we derive three canonical descriptors:

(i) *Gate Fidelity Score (GFS).* An aggregate reliability indicator over the transpiled gate set $\mathcal{G}(M)$:

$$\text{GFS}(M, b) \;=\; \exp\Big(\sum_{g \in \mathcal{G}(M)} w_g \, \log(1 - \epsilon_g)\Big), \quad (4)$$

where ϵ_g is the error rate for gate g on its bound qubits and w_g are optional importance weights.

(ii) *Depth Change Ratio (DCR).* A measure of depth inflation induced by mapping and deployment optimizations is as

$$\text{DCR} \;=\; \frac{\text{depth}_{\text{post}}(M)}{\text{depth}_{\text{pre}}(\mathcal{G})}, \quad (5)$$

where $\text{depth}_{\text{pre}}(\mathcal{G})$ denotes the logical circuit depth computed from the original (pre-transpilation) gate dependency DAG, and $\text{depth}_{\text{post}}(M)$ denotes the physical circuit M depth after mapping and scheduling on the target quantum backend b. Hence, DCR captures the relative depth overhead caused by hardware-specific compilation, routing, and timing constraints, quantifying how much the physical realization deviates from the ideal logical structure.

(iii) *Mapping-Induced Crosstalk (MIC).* A routing sensitivity indicator that increases with simultaneous two-qubit activity on nearby couplers. A practical proxy aggregates neighborhood-adjusted two-qubit density:

$$\text{MIC} \;=\; \frac{1}{|\mathcal{S}|} \sum_{(i,j) \in \mathcal{S}} \underbrace{\rho_{2\text{q}}(i,j)}_{\text{local 2Q utilization}} \cdot \underbrace{\chi(i,j)}_{\text{coupler proximity}}, \quad (6)$$

where $\mathcal{S}$ are active edges after mapping, χ penalizes operations on adjacent or overlapping neighborhoods, and $\rho_{2\text{q}}(i,j)$ denotes the local utilization of the two-qubit gate on coupler (i,j), computed as the fraction of total scheduling steps in which a two-qubit gate is executed on that coupler. Formally, it is given by summing an indicator function over all time steps and normalizing by the total number of steps. Optionally, this measure can be adjusted to account for neighboring coupler activity, capturing potential crosstalk effects that arise when adjacent qubits perform simultaneous two-qubit operations.

Information Enhancement and Adjustment. We aggregate key backend indicators into a compact feature vector

$$\phi_{\text{hw}}(M, b) = \big[\text{GFS},\ \text{DCR},\ \text{MIC},\ \rho_{2\text{q}},\ \bar{\epsilon}_{1\text{q}},\ \bar{\epsilon}_{2\text{q}}\big], \quad (7)$$

where $\rho_{2\text{q}}$ measures the local two-qubit gate utilization, and $\bar{\epsilon}_{1\text{q}}$, $\bar{\epsilon}_{2\text{q}}$ are the mean single-qubit and two-qubit gate error rates obtained from backend calibration. This vector is mapped through a monotone aggregation function $\psi(\cdot)$, which combines the indicators into a scalar backend factor $C_{\text{adj}} \in [0, 1]$ while preserving monotonicity. This factor adjusts both the overall difficulty score and the structure-aware batching policy:

$$\begin{aligned}
\underbrace{s_{\text{final}}}_{\text{difficulty}} &= \text{norm}\big(\mu\, f_{\text{struct}}(\mathcal{G}) + \upsilon\, \psi(\phi_{\text{hw}})\big), \\
\underbrace{P}_{\text{batch size}} &= \text{clip}\Big(\kappa \cdot g\big(s_{\text{final}}, \text{DCR}, \rho_{2\text{q}}\big),\ P_{\min}, P_{\max}\Big).
\end{aligned} \quad (8)$$

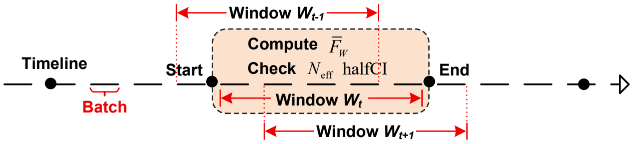


Fig. 3: Intuitive explanation of the rolling window.

where $\mu, \upsilon \geq 0$ are weighting coefficients of the structural complexity $f_{\text{struct}}(\cdot)$ and the hardware-related factors. Meanwhile, $\kappa > 0$ is a scaling coefficient that adjusts the overall magnitude of the batch size P, and $g(\cdot)$ is a monotonic function of the estimated difficulty and transpilation stressors.

Intuitively, C_{adj} down-weights overly optimistic structural difficulty estimates when the mapped circuit exhibits high error rates (low GFS, high $\bar{\epsilon}_{1\text{q}}, \bar{\epsilon}_{2\text{q}}$), excessive depth (high DCR), or strong crosstalk sensitivity (high MIC).

D. Adaptive Measurement Pipeline

We present here the end-to-end pipeline for accurate and efficient fidelity measurement, as shown in Figure 1. Given a logical circuit, we construct its dependency DAG $\mathcal{G}$ and obtain a transpiled (physical) circuit M via a mapping policy logical→physical, which specifies how logical qubits and gates are embedded onto the target hardware topology. Structural descriptors from the logical graph $\mathcal{G}$ (topological features) are combined with backend-aware descriptors extracted from the mapped circuit M (*e.g., GFS, DCR, MIC*) to yield a normalized difficulty score $s_{\text{final}} \in [0, 1]$.

Rolling Window and Confidence-Bound Check. Batches are executed sequentially, yielding per-batch fidelity estimates $\hat{F}_t$ and within-batch dispersions. As shown in Figure 3, a rolling index window $W_t = \{t - w + 1, \ldots, t\}$ is maintained and deemed *stable* if (i) the Exponentially Weighted Moving Average (EWMA) [49] of residuals remains within a scale-aware band, and (ii) the maximum deviation inside the window is bounded by a multiple of the window standard deviation. On a stable window, the window mean $\bar{F}_W$, standard deviation s_W, and effective sample size N_{eff} are computed (default set $N_{\text{eff}} = |W_t|$, with weighted variants when batch sizes differ). For the confidence level $(1 - \alpha)$, the two-sided half-width is

$$\text{halfCI} \;=\; c_\alpha \, \frac{s_W}{\sqrt{N_{\text{eff}}}}, \qquad c_\alpha \in \left\{ z_{\alpha/2},\; t^{(|W_t|-1)}_{1-\alpha/2},\; c^{\text{boot}}_\alpha \right\}. \quad (9)$$

The window passes the precision check if $\text{halfCI} \leq \delta$ and $N_{\text{eff}} \geq N^{\text{guard}}_{\text{min}}$ (*i.e.,* minimum guard window size).

Batch Adaptation and Termination. If the confidence-bound check is not yet satisfied, the next batch size is adapted as

$$P_{t+1} \;=\; \text{clip}\Big(\eta_t \cdot g\left(s_{\text{final}}, \text{DCR}, \rho_{2\text{q}}\right),\; P_{\min},\; P_{\max}\Big), \quad (10)$$

where the adaptation rate η_t grows geometrically when far from the target but transitions to linear increments as halfCI approaches the confidence half-width threshold δ to avoid overshooting. If the stability check fails, a multiplicative backoff is applied $P_{t+1} \leftarrow \max\{P_{\min}, \lfloor \rho P_t \rfloor\}$ with the scale factor $\rho \in (0, 1)$, followed by window reset, as shown in Figure 4.

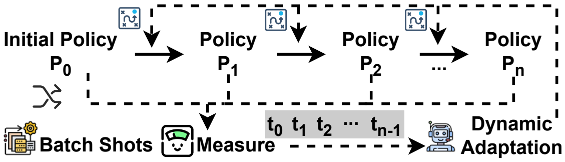


Fig. 4: Dynamical pipeline of adaptive iteration policy.

The loop terminates when the window is stable, the confidence-bound condition holds, and $N_{\text{eff}} \geq N^{\text{guard}}_{\text{min}}$, and the reported estimate is $\bar{F}_W$ with a certified $(1 - \alpha)$ interval of half-width at most δ. To provide a rigorous, distribution-free confidence guarantee, we apply Hoeffding's inequality [50] to the total effective sample size within the stable window. Specifically, each batch j in the window W_t contributes a weight $\mathcal{P}_j$ proportional to its effective number of shots, and the cumulative measure

$$\mathcal{P}_W \;=\; \sum_{j \in W_t} \mathcal{P}_j, \quad (11)$$

represents the total effective sample size within the window. Then, using Hoeffding's inequality on this aggregate quantity

$$\mathbb{P}\left(|\hat{F} - F| > \delta\right) \;\leq\; 2\exp\!\left(-2\,\mathcal{P}_W\,\delta^2\right), \quad (12)$$

where $\hat{F}$ denotes the empirical fidelity aggregated over stable window, and F is the true fidelity, which equivalently requires

$$\mathcal{P}_W \;\geq\; \frac{1}{2\delta^2}\,\log\!\left(\frac{2}{\alpha}\right), \quad (13)$$

for the window to provide a valid $(1-\alpha)$ confidence guarantee.

Shot Forecasting and Budgeting. As $\text{halfCI} \propto 1/\sqrt{N_{\text{eff}}}$, the additional evidence required to meet $(\delta, 1 - \alpha)$ is forecast online as

$$N^{\text{add}}_{\text{min}} \;\approx\; \max\left\{0,\; \left(\frac{c_\alpha\, s_W}{\delta}\right)^2 - N_{\text{eff}}\right\}. \quad (14)$$

This value is then used to limit the next batch size P_{t+1} and is reported for transparent resource budgeting. By localizing inference to stable windows, sequential-peeking effects are mitigated without altering the controller. The resulting pipeline is reproducible, robust to backend drift, adaptive to mapping-induced difficulty, and provides explicit, window-local confidence guarantees at a user-selected confidence level.

IV. Experimental Setup

This section presents the experimental design and infrastructure to evaluate our proposed approach AutoFid. We first define the key research questions. Then, we describe the experimental environment, benchmarks, and implementation settings.

A. Research Questions (RQs)

We design a series of experiments to answer the following research questions.

- **RQ1: Measurement Efficiency across Various Circuit.** How effectively does AutoFid estimate the required number of fidelity measurements across diverse circuits and varying fidelity-bias thresholds?

TABLE I: Benchmarks used in our experiments.

Abbreviation	**Benchmark**
BV	Bernstein-Vazirani algorithm
Clifford	Random Clifford circuit
Ising	Linear Ising model
QAOA	Quantum Approximate Optimization Algorithm
VQE	Variational Quantum Eigensolver
QFT	Quantum Fourier Transform
QKNN	Quantum Kernel method for Nearest Neighbors
QNN	Quantum Neural Network
QPE	Quantum Phase Estimation
QSVM	Quantum Support Vector Machine
QuGAN	Quantum Generative Adversarial Network
RB	Randomized Benchmarking
Amplitude	Amplitude Estimation
Shor	Shor's algorithm for integer factorization
Simon	Simon's algorithm for hidden bit extraction
SU2	SU(2) quantum circuits
VQC	Variational Quantum Circuit
XEB	Cross-Entropy Benchmarking

- **RQ2: Fidelity-Budget Trade-Off.** How does the fidelity bias evolve with increasing measurement budgets, and what Pareto behavior emerges between accuracy and resource consumption?
- **RQ3: Comparative Evaluation and Stability.** How does AutoFid compare with existing ML-based estimators (QuCT, QuEst) under standardized testing dimensions such as stability, bias consistency, and test redundancy?
- **RQ4: Component Contribution and Overhead.** What is the contribution of the design component (mixing-time estimation, backend-aware adaptation, and early-stopping control), and how much overhead does each introduce?
- **RQ5: Scalability and Hardware Generalization.** How does AutoFid scale with circuit size and qubit count, and can it generalize from simulated backends to real trapped-ion hardware under different quantum algorithms?

B. Experimental Platform and Settings

Our experiments are mainly conducted on *IBM Quantum hardware platforms* [51] using the *Qiskit* software development kit [46]. We mainly employ three superconducting backends, *i.e.,* `Sherbrooke`, `Kyiv`, and `Brisbane`, each providing up to 127 physical qubits. The baseline fidelity estimation uses a default of `shots=10000`, with a measurement interval of `step=20` for iteration-level comparisons.

Unless otherwise stated, the confidence level is set to $\alpha = 0.05$ and the default fidelity-bias threshold to $\delta = 0.01$. For sensitivity and ablation studies, we vary $\delta \in \{0.01, 0.02, 0.03\}$ and measurement budgets $P_{sum} \in \{500, 1000, \ldots, 10000\}$ to examine robustness under different accuracy-budget trade-offs.

C. Benchmark Circuits

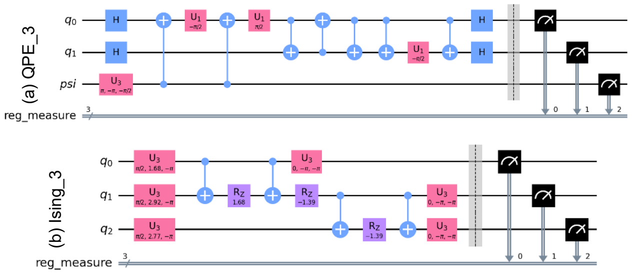


Fig. 5: Some instances of quantum circuits, *e.g.,* QPE and Ising.

We evaluate AutoFid across 18 representative quantum algorithms widely adopted in quantum software verification, optimization, and machine-learning workloads [52], [53], [54], [55]. The selected benchmarks, summarized in Table I, cover a diverse set of algorithmic paradigms:

- **Classical Algorithms (7)**: BV, Simon, QPE, QFT, Shor, Clifford, Ising;
- **Variational Algorithms (5)**: QAOA, VQE, VQC, SU2;
- **Machine Learning Models (4)**: QKNN, QNN, QSVM, QuGAN;
- **Benchmarking Circuits (3)**: RB, XEB, Amplitude.

Each circuit is evaluated under qubit sizes $\{4, 6, 8, 10\}$ and across all three hardware backends (`Sherbrooke`, `Kyiv`, and `Brisbane`) to ensure coverage and reproducibility. Figure 5 illustrates several example circuits. This collection provides a balanced coverage of computational patterns, from structured quantum algorithms to variational and data-driven workloads, enabling a comprehensive evaluation of AutoFid across both algorithmic and hardware dimensions.

V. Evaluation Results

We present the results of our experiments, organized by the research questions defined in § IV-A.

A. *RQ1: Measurement Efficiency across Various Circuits*

To evaluate the measurement efficiency of AutoFid across different quantum algorithms, we investigate the estimated number of measurement iterations required to achieve varying fidelity bias thresholds. The fidelity bias is defined as the deviation between the measured fidelity under a given sampling budget and the ground-truth fidelity obtained with 10000 shots. Table II reports the estimated measurement counts for 18 representative quantum algorithms under fidelity bias thresholds ranging from 0.01 to 0.03.

As shown in Table II, the required number of measurements increases substantially as the fidelity bias threshold becomes stricter. For instance, in the Bernstein-Vazirani algorithm (BV) with 4 qubits, only 76 measurement iterations are sufficient to reach a fidelity bias of 0.03, whereas achieving a bias below 0.01 demands up to 617 measurements. This exponential growth in measurement cost is more pronounced in structurally complex circuits. For example, when operating with 4 qubits, the Quantum Kernel method for Nearest Neighbors (QKNN), Quantum Neural Network (QNN), and SU(2) variational circuits require 1,249, 1,820, and 1,447 measurements, respectively, to reach a fidelity bias below 0.01. These findings indicate that measurement efficiency

TABLE II: Required iterations for different circuit families under various fidelity-bias constraints.

									Fidelity bias < 0.01									
Qubits	BV	Clifford	Ising	QAOA	VQE	QFT	QKNN	QNN	QPE	QSVM	QuGAN	RB	Ampl.	Shor	Simon	SU2	VQC	XEB
4	617	706	877	5642	560	952	1249	1820	180	1219	683	1182	1088	707	60	1447	1369	267
6	837	2462	2634	6716	2006	3538	2994	3274	994	1857	3253	2049	4531	2683	132	2916	3277	2612
8	917	6367	5558	8238	3783	5389	5889	8440	1651	7138	5196	6451	6699	6777	260	6420	4821	6600
10	993	9381	8963	9822	9043	9267	7550	9668	2042	8658	9476	7890	8700	9206	1424	9224	9591	8420

									Fidelity bias < 0.02									
Qubits	BV	Clifford	Ising	QAOA	VQE	QFT	QKNN	QNN	QPE	QSVM	QuGAN	RB	Ampl.	Shor	Simon	SU2	VQC	XEB
4	154	422	398	1246	278	469	764	904	178	347	454	884	950	467	60	1002	901	261
6	305	1364	1522	2195	1475	1751	1611	2429	498	1238	1753	1519	1371	1757	130	2142	3213	1541
8	804	4836	3799	3213	2958	3975	3293	4785	813	4852	3744	4209	4530	4670	129	4436	3529	4001
10	903	8509	7840	7723	6887	8094	5520	7915	988	6845	7980	6604	7630	8105	279	8077	7813	5867

									Fidelity bias < 0.03									
Qubits	BV	Clifford	Ising	QAOA	VQE	QFT	QKNN	QNN	QPE	QSVM	QuGAN	RB	Ampl.	Shor	Simon	SU2	VQC	XEB
4	76	278	314	621	272	232	460	451	176	345	223	576	537	466	59	141	892	262
6	202	1087	976	1079	1465	1308	786	1592	493	915	701	1004	1021	864	64	1585	2401	1527
8	712	3335	2997	3172	2193	3291	2534	4682	804	3482	2771	3482	3433	3994	127	3579	3532	3153
10	797	7011	6960	7664	5908	7141	4070	7730	984	6206	7383	5474	6549	7226	92	7255	7842	4660

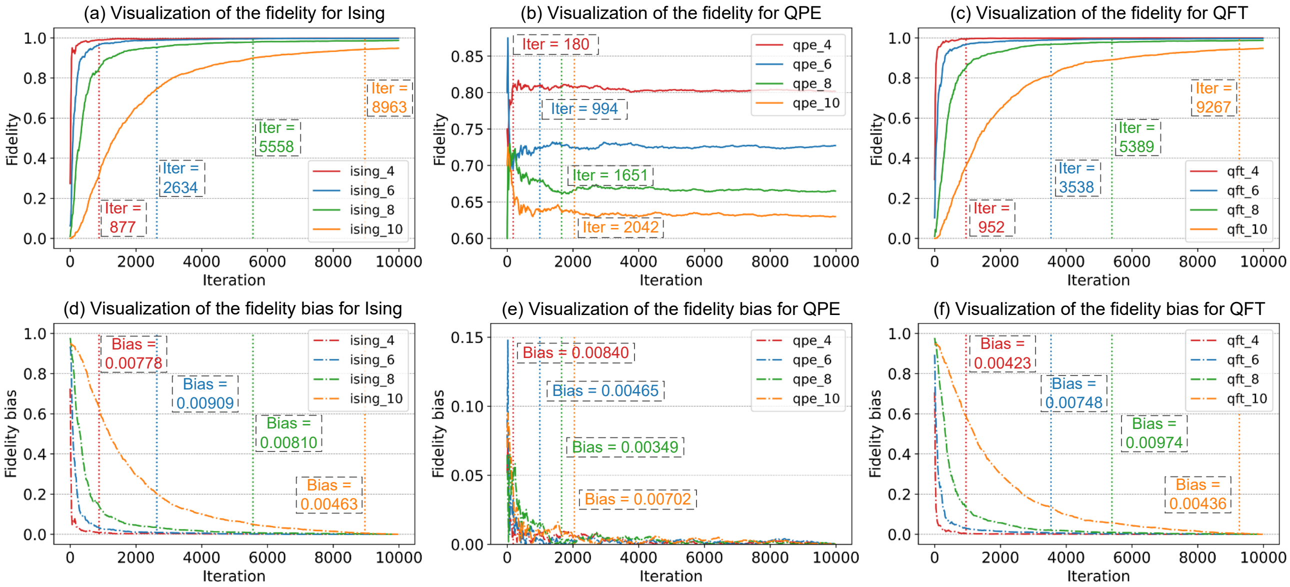


Fig. 6: Visualization of the circuit measurement process.

varies dramatically across circuit structures and algorithmic paradigms. A fixed measurement budget, often determined heuristically or by expert experience, fails to capture such heterogeneity. Instead, adaptive measurement strategies are essential to balance accuracy and resource efficiency, particularly when evaluating quantum software systems that integrate heterogeneous quantum algorithms.

B. RQ2: Fidelity-Budget Trade-Off

1) Fidelity Convergence with Iterative Measurements: To investigate how measurement cost scales with circuit size and fidelity precision, we examine the trade-off between the measurement budget and achieved fidelity across three representative quantum algorithms, *i.e.,* Linear Ising Model (Ising), Quantum Phase Estimation (QPE), and Quantum Fourier Transform (QFT) under qubit counts of $\{4, 6, 8, 10\}$. Figure 6 illustrates both (a∼c) the evolution of measured fidelity as the number of measurement iterations increases and (d∼f) the corresponding fidelity bias convergence curves. The vertical dashed lines indicate the iteration counts or bias thresholds automatically determined by our AutoFid framework.

(i) *Measurement iteration dynamics.* In Figure 6 (a∼c), larger quantum circuits require a greater number of measurement iterations to achieve stable fidelity estimates. For each qubit configuration, the measured fidelity initially rapidly changes with iteration count, followed by a short oscillation phase, and eventually converges to a steady value. In the case of the Ising circuit in Figure 6 (a), the estimated convergence points occur at approximately 877, 2,634, 5,558, and 8,963 iterations for 4, 6, 8, and 10 qubits, respectively. Notably, the measured fidelity values remain stable beyond these automatically detected points, validating the effectiveness of AutoFid in estimating convergence boundaries with minimal overhead.

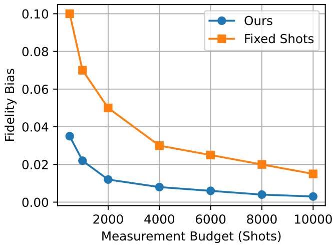


Fig. 7: Test budget *vs* fidelity bias.

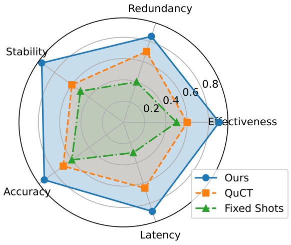


Fig. 8: Five testing metrics.

(ii) *Fidelity bias convergence.* Figure 6 (d∼f) further analyze the evolution of fidelity bias as a function of the measurement budget. Similar to the previous observation, increasing the number of qubits leads to a slower convergence rate, requiring more measurements to achieve a given bias threshold. In the 4-qubit Ising circuit, for instance, a fidelity bias below 0.00778 is achieved after only 877 iterations, whereas the 10-qubit configuration requires 8,963 iterations to reduce the bias to 0.00463. These results demonstrate that AutoFid effectively captures the fidelity-budget trade-off across diverse circuit scales, achieving comparable accuracy with significantly fewer repeated measurements. This property highlights the resource efficiency of our framework, particularly in scenarios where quantum measurement time or shot budgets are constrained.

Overall, the results confirm that the adaptive measurement scheme in AutoFid provides an efficient mechanism to balance fidelity precision and measurement cost, achieving convergence-aware fidelity estimation without over-sampling.

2) Pareto Analysis under Fixed Measurement Budgets: We further conduct a Pareto analysis to quantify how AutoFid performs when the measurement budget is fixed in advance. Specifically, we set the measurement budget $P_{sum} \in \{500, 1000, \ldots, 10000\}$ and compare the resulting fidelity bias achieved by AutoFid against a baseline fixed-shot strategy. As illustrated in Figure 7, AutoFid consistently achieves lower fidelity bias across all budget levels. When measurement resources are severely constrained (*e.g.,* $P_{sum} = 1000$), AutoFid attains a fidelity bias of approximately 0.025, whereas the fixed-shot baseline produces a significantly higher bias of around 0.07. As the budget increases to $P_{sum} = 4000$, the bias achieved by AutoFid reduces to about 0.01, while the baseline remains above 0.03. Even at the largest budget ($P_{sum} = 10000$), AutoFid maintains a clear advantage, achieving a fidelity bias near 0.005 compared to the baseline's 0.017. These results reveal a strong Pareto dominance of AutoFid in the fidelity-cost space. AutoFid consistently forms a lower Pareto frontier, providing more accurate fidelity estimation under equal or smaller resource constraints.

This advantage is particularly evident in low-resource settings, demonstrating the practicality of AutoFid for fidelity evaluation on constrained quantum hardware where measurement shots are costly and time-limited.

C. RQ3: Comparative Evaluation and Stability

This research question investigates how our proposed framework, AutoFid, performs against state-of-the-art ML-based fidelity prediction approaches: QuCT [19] and QuEst [41] in terms of both testing effectiveness and runtime stability. Unlike QuCT and QuEst, which directly regress circuit fidelity from handcrafted or learned circuit features, AutoFid explicitly estimates the minimal number of measurement shots required to achieve a target fidelity accuracy. This design provides stronger interpretability and resource awareness, which are critical in practical quantum testing pipelines.

TABLE III: Comparison of measurement efficiency across benchmark circuits under fidelity bias < 0.02.

Circuit	**Measurement Shots (↓)**		
	Ours	**QuCT**	**QuEst**
QAOA	**1246**	1524	2149
QKNN	**764**	1339	1986
Amplitude	**950**	1674	2178
SU2	**1002**	1789	2237

1) Multi-Dimensional Evaluation Metrics: To provide a holistic comparison, we evaluate all three methods across five key testing dimensions, *i.e., effectiveness*, *redundancy*, *stability*, *accuracy*, and *latency*. These dimensions capture both the software testing efficiency and operational robustness of quantum measurement processes. Figure 8 illustrates the radar chart comparing the three approaches.

AutoFid consistently achieves the target fidelity level with 40-60% fewer measurement iterations than baseline methods, as indicated by its superior coverage on the "Effectiveness" axis. This efficiency arises from its adaptive convergence detection mechanism, which dynamically adjusts measurement budgets. Compared with fixed-shot sampling strategies, AutoFid effectively eliminates unnecessary measurement repetitions. The ratio of actual shots to the theoretical minimum decreases from 5.1× (Fixed Shots) and 2.4× (QuCT) to just 1.2× under AutoFid, representing substantially leaner test execution. When executed across multiple IBM Quantum backends (`Sherbrooke`, `Kyiv`, and `Brisbane`) and time intervals, AutoFid exhibits high measurement consistency with a standard deviation below 0.002, significantly outperforming competing methods. This stability is especially valuable given the temporal and backend-dependent noise fluctuations inherent in NISQ hardware. AutoFid achieves higher fidelity estimation accuracy while maintaining lower latency. Its adaptive early-stopping mechanism, guided by circuit structural features (*e.g.,* gate connectivity and qubit depth), allows convergence to be detected efficiently, avoiding excessive quantum executions.

Overall, these results demonstrate that AutoFid achieves a well-balanced trade-off across all key testing dimensions, highlighting the benefit of our structure-aware fidelity estimation in practical quantum testing.

2) Measurement Efficiency under Fidelity Constraints: We further compare AutoFid, QuCT, and QuEst on 4-qubit circuits under a target fidelity bias threshold of < 0.02. Table III reports the required number of measurement shots to meet this accuracy constraint across representative circuits, including the QAOA, QKNN, Amplitude, and SU2 quantum circuits. Across all tested algorithms, AutoFid achieves the target fidelity bias

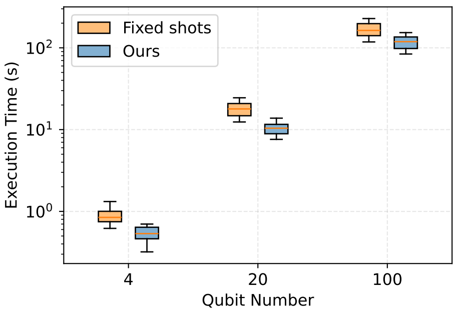
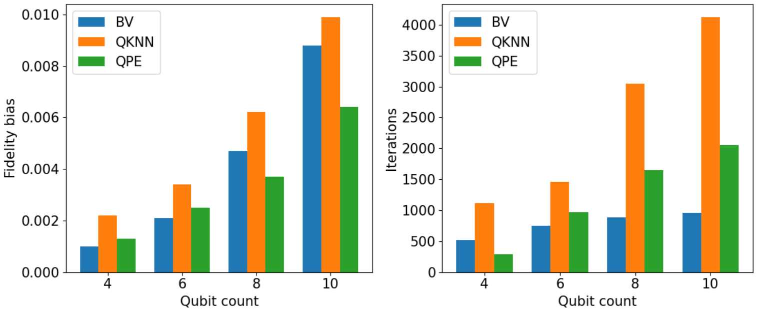


Fig. 9: Overhead evaluation.



(a) Fidelity bias

(b) Iterations

Fig. 10: Fidelity measurement on IonQ.

TABLE IV: Ablation study on the proposed model components.

Variant	Fidelity Bias (↓)	Test Redundancy (↓)
Full	0.004	1.1
w/o t_{mix}	0.011	2.0
w/o backend	0.009	1.8
Fixed-P	0.006	1.6

TABLE V: Overhead breakdown.

Steps	Time (s)		
	$4q$	$20q$	$100q$
Transpilation	0.14	1.62	22.8
Random walk	2.3×10^{-3}	8.1×10^{-3}	9.6×10^{-2}
Extract features	3.4×10^{-3}	1.1×10^{-2}	7.1×10^{-2}
Execution	0.41	8.6	93.2

with the lowest measurement cost, requiring $18\%{\sim}44\%$ fewer shots than QuCT and up to 62% fewer than QuEst, while satisfying fidelity requirements. The improvement stems from AutoFid's dynamic allocation of measurement shots based on circuit-level structural dependencies and backend-specific noise calibration. In contrast, QuCT relies on gate-level statistical modeling, and QuEst uses static graph embeddings that lack adaptive convergence monitoring, both leading to suboptimal measurement utilization.

D. RQ4: Component Contribution and Overhead

1) Ablation Study on Components: To evaluate the contribution of each component in AutoFid, we conduct an ablation study by disabling individual modules. We consider three variants, *i.e.,* replacing mixing time with node count (w/o t_{mix}), removing backend-aware adjustment features (w/o backend), and fixing the measurement batch to a constant size without early stopping (Fixed-P). Table IV presents the impact of each variant on two metrics, fidelity bias and test redundancy. (i) In the first variant, denoted as w/o t_{mix}, we replace the structure-aware mixing time estimate with a naive metric based on the DAG node count. As a result, removing t_{mix} causes fidelity bias to increase sharply from 0.004 to over 0.011, leading to an increase of more than 1.5×. Meanwhile, as test redundancy doubles, suggesting over-sampling becomes more frequent.

(ii) In the second variant, w/o backend adjustment, we remove all backend-aware features, including the gate fidelity score (GFS), depth change ratio (DCR), and mapping-induced crosstalk (MIC). Without these device-specific signals (GFS, DCR, MIC), fidelity bias rises to 0.009 and redundancy reaches 1.8×. This indicates that hardware awareness is essential for noise adaptation and efficient stopping. The lack of backend information impairs the model's ability to account for hardware-specific noise effects, leading to a higher variance in fidelity estimates and weaker noise adaptation.

(iii) Finally, in the Fixed-P setting, we disable the early stopping mechanism and instead fix the measurement batch shots. While this simplification removes adaptive control, it introduces substantial inefficiency, resulting in redundant measurements being reached 1.6×, without corresponding improvements in fidelity. Overall, these results validate that both structural estimation and backend sensitivity are necessary for the observed efficiency and accuracy gains of AutoFid.

2) Overhead Breakdown: To assess the runtime overhead introduced by our method, we provide a breakdown analysis. The steps "Random walk" and "Extract features" correspond to the additional cost of our approach. As shown in Table V, these components incur negligible overhead compared with transpilation and execution. We report averaged results with circuits at 4, 20, and 100 qubits. The construction of the DAG is a standard step in the compilation flow of quantum circuits, every circuit submitted to real hardware undergoes this transformation as part of transpilation before execution. AutoFid leverages this existing representation and adds a lightweight random-walk mixing time estimation, whose complexity is $O(|G|^2)$. The breakdown in Table V shows that the dominant cost arises from execution and, to a lesser extent, transpilation, whereas the overhead introduced by our method is negligible. For instance, on 100 qubits, execution requires 93.2 s and transpilation 22.8 s, while the additional costs from random walk and feature extraction are only $9.6{\times}10^{-2}$ s and $7.1{\times}10^{-2}$ s respectively, together less than 0.2% of the total runtime. Similar trends are observed at 4 and 20 qubits.

E. RQ5: Scalability and Hardware Generalization

1) Scalability Evaluation: We evaluate a series of circuits at 4, 20, and 100 qubits. For large scales, execution and measurement times are estimated by simulation, and wall-clock runtime is modeled as circuit duration plus measurement and reset overhead on a superconducting backend. Fixed-shot baselines (10k for 4 qubits, 200k for 20 qubits, 400k for 100 qubits) are guided by reported hardware sampling rates [56], [57] and instruction-level timing models [51], targeting fidelity ≈ 0.98. As fidelity estimators scale as $O(1/\sqrt{N})$, these budgets provide a reasonable reference while acknowledging circuit-dependent deviations. As shown in Figure 9, results show that adaptive sampling consistently reduces time-to-result while meeting the fidelity constraint $F \geq 0.98$. For 4 qubits, mean runtime decreases by $\sim$35%; on 20 qubits, by $\sim$32%; and on 100 qubits, by $\sim$27%. Circuit-level gains are most pronounced for deep or noise-sensitive workloads, with improvements exceeding 80% (*e.g.,* QFT and QNN), while lightweight circuits yield smaller savings. Overall, adaptive sampling achieves 27%$\sim$35% overhead reduction across scales, lowering measurement cost without compromising fidelity.

2) Extended Experiments: We further evaluate BV, QKNN, and QPE algorithm circuits on the IonQ trapped-ion platform [58] for $n \in \{4, 6, 8, 10\}$ qubits. Figure 10 reports the fidelity bias and adaptive iteration counts. For BV, the average bias remains below 10^{-2} across all qubit sizes, *e.g.,* {0.0010, 0.0021, 0.0047, 0.0088} for {4, 6, 8, 10} qubits. The corresponding adaptive iterations are {523, 754, 887, 962}. QKNN shows higher complexity and thus larger bias (0.0022∼0.0099) and iteration counts (1115∼4127), while QPE achieves relatively lower bias (0.0013∼0.0064) with moderate iteration overhead (289∼2059). These results highlight two key trends on IonQ hardware: (i) fidelity bias decreases with shot count in the variance-limited regime, then saturates at a hardware-limited floor determined by accumulated two-qubit and SPAM errors, and (ii) the floor increases with qubit number, reflecting the linear growth of error sources. The observed bias levels on IonQ are consistent with its reported native fidelities. More importantly, these results demonstrate that our methodology can be readily applied to other quantum hardware platforms, providing a practical baseline for shot-budget planning and platform-dependent accuracy-cost analysis.

VI. Discussion

A. Deep Insights into Circuit Complexity Estimation

To further justify the use of random-walk mixing time as a compact complexity indicator, we compare it with simpler alternatives, including circuit size, depth, and gate count. As shown in Table VI, replacing mixing time with these metrics consistently yields higher fidelity bias (0.008∼0.010 vs. 0.004) and larger redundancy (1.7∼1.9 vs. 1.1). This result indicates that while circuit size, depth, and gate count provide coarse measures of circuit structure, they do not capture noise-sensitive connectivity patterns as effectively as mixing time. The mixing time, derived from circuit topology via random walks, better correlates with uncertainty amplification under hardware noise, thereby guiding adaptive sampling.

TABLE VI: Comparison of different complexity indicators.

Indicator	Mixing time (ours)	Size	Depth	Gate count
Fidelity Bias	0.004	0.010	0.009	0.008
Test Redundancy	1.1	1.9	1.8	1.7

TABLE VII: Evaluation with non-stationary noise.

Circuit	Temporal	Fidelity Bias		
		IBM_Lima	**IBM_Quito**	**IBM_Manila**
BV	Time 1	0.0040	0.0050	0.0060
	Time 2	0.0061	0.0072	0.0080
	Time 3	0.0082	0.0101	0.0120
QPE	Time 1	0.0050	0.0061	0.0070
	Time 2	0.0073	0.0090	0.0102
	Time 3	0.0100	0.0122	0.0140

B. Considering Non-stationary Noise

To examine the effect of temporal and device drift, we evaluate 4-qubit BV and QPE circuits on three IBM backends (Lima, Quito, Manila) across historical noise at different times [59]. As shown in Table VII, fidelity bias increases both over time and across devices. For instance, BV on IBM_Lima grows from 0.0040 (Time 1) to 0.0082 (Time 3), while IBM_Manila exhibits the largest deviation (0.0120 at Time 3). QPE shows a similar trend, with late-time runs incurring up to 2× higher bias compared to early runs. These results confirm that non-stationary noise, arising from calibration drift and hardware variability, can significantly affect fidelity estimation. Nevertheless, AutoFid's adaptive sampling strategy mitigates this risk by dynamically sensing variance and allocating extra shots, and it remains robust to such fluctuations, avoiding premature stopping compared with fixed-shot baselines.

C. Limitation and Future Work

Currently, AutoFid can better exploit its adaptive advantage when noise remains relatively stationary within each measurement batch. Under extreme temporal drift or strongly correlated errors, its early stopping may become conservative or require additional resampling. Extending the framework with drift-detection heuristics and calibration-aware safeguards is an important direction. Future work will also explore cross-platform validation and integration with advanced error-mitigation techniques. In the long term, we aim to generalize AutoFid to support continuous online adaptation across heterogeneous quantum hardware environments.

VII. Conclusion

Fidelity measurement is crucial for quantum program testing, yet shot allocation remains challenging in the NISQ era due to noise and circuit complexity. We propose AutoFid, an adaptive, noise-aware framework that integrates random walk-based complexity estimation, backend-aware correction, and early stopping to reduce redundant measurements while preserving accuracy. Experiments on IBMQ with 18 benchmarks show that AutoFid lowers testing overhead and keeps fidelity bias within 0.01, outperforming fixed-shot and ML-based baselines under Pareto, and cold-start evaluations.

Acknowledgments

This research is supported by the Shanghai Qi Zhi Institute Innovation Program (No. SQZ202318).

References

[1] Miroslav Urbanek, Benjamin Nachman, and Wibe A de Jong. Error detection on quantum computers improving the accuracy of chemical calculations. *Physical Review A*, 102(2):022427, 2020.

[2] Ramin Ayanzadeh, Ahmad Mousavi, Narges Alavisamani, and Moinuddin Qureshi. Enigma: Privacy-preserving execution of qaoa on untrusted quantum computers. *arXiv preprint arXiv:2311.13546*, 2023.

[3] Tingting Li, Ziming Zhao, and Jianwei Yin. Minerva: Enhancing quantum network performance for high-fidelity multimedia transmission. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 3704–3712, 2024.

[4] Sun Xu, Yangming Zhao, Liusheng Huang, and Chunming Qiao. Routing and photon source provisioning in quantum key distribution networks. In *IEEE INFOCOM 2024-IEEE Conference on Computer Communications*, pages 1411–1420. IEEE, 2024.

[5] Wenhui Ren, Weikang Li, Shibo Xu, Ke Wang, Wenjie Jiang, Feitong Jin, Xuhao Zhu, Jiachen Chen, Zixuan Song, Pengfei Zhang, et al. Experimental quantum adversarial learning with programmable superconducting qubits. *Nature Computational Science*, 2(11):711–717, 2022.

[6] Zitian Zhu, Lei Gao, Zehang Bao, Liang Xiang, Zixuan Song, Shibo Xu, Ke Wang, Jiachen Chen, Feitong Jin, Xuhao Zhu, et al. Quantum highway: Observation of minimal and maximal speed limits for few and many-body states. *arXiv preprint arXiv:2408.11900*, 2024.

[7] Liang Xiang, Jiachen Chen, et al. Enhanced quantum state transfer by circumventing quantum chaotic behavior. *Nature Communications*, 15(1):4918, 2024.

[8] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*, volume 2. Cambridge university press Cambridge, 2001.

[9] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2018.

[10] Tingting Li, Ziming Zhao, and Jianwei Yin. QLSel: Demonstrating efficient high-fidelity link selection for quantum networks in the wild. In *Proceedings of The 30th Annual International Conference on Mobile Computing and Networking*, pages 1766–1768, 2024.

[11] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.

[12] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.

[13] Zhirui Hu et al. Battle against fluctuating quantum noise: Compression-aided framework to enable robust quantum neural network. In *DAC*, 2023.

[14] Zhiding Liang, Zhixin Song, Jinglei Cheng, Zichang He, Ji Liu, Hanrui Wang, Ruiyang Qin, Yiru Wang, Song Han, Xuehai Qian, et al. Hybrid gate-pulse model for variational quantum algorithms. In *2023 60th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2023.

[15] Earl Campbell. A series of fast-paced advances in quantum error correction. *Nature Reviews Physics*, 6(3):160–161, 2024.

[16] Easwar Magesan, Jay M Gambetta, and Joseph Emerson. Characterizing quantum gates via randomized benchmarking. *Physical Review A—Atomic, Molecular, and Optical Physics*, 85(4):042311, 2012.

[17] Sergio Boixo, Sergei V Isakov, Vadim N Smelyanskiy, Ryan Babbush, Nan Ding, Zhang Jiang, Michael J Bremner, John M Martinis, and Hartmut Neven. Characterizing quantum supremacy in near-term devices. *Nature Physics*, 14(6):595–600, 2018.

[18] Luyan Sun, Andrei Petrenko, Zaki Leghtas, Brian Vlastakis, Gerhard Kirchmair, KM Sliwa, Aniruth Narla, Michael Hatridge, Shyam Shankar, Jacob Blumoff, et al. Tracking photon jumps with repeated quantum non-demolition parity measurements. *Nature*, 511(7510):444–448, 2014.

[19] Siwei Tan, Congliang Lang, Liang Xiang, Shudi Wang, Xinghui Jia, Ziqi Tan, Tingting Li, Jieming Yin, Yongheng Shang, Andre Python, et al. QuCT: A Framework for Analyzing Quantum Circuit by Extracting Contextual and Topological Features. In *Proceedings of the 56th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 494–508, 2023.

[20] Hanrui Wang et al. QuantumNAS: Noise-adaptive search for robust quantum circuits. In *HPCA*, 2022.

[21] Heinz-Peter Breuer and Francesco Petruccione. *The theory of open quantum systems*. Oxford University Press, USA, 2002.

[22] Abhinav Kandala, Kristan Temme, Antonio D Córcoles, Antonio Mezzacapo, Jerry M Chow, and Jay M Gambetta. Error mitigation extends the computational reach of a noisy quantum processor. *Nature*, 567(7749):491–495, 2019.

[23] David C McKay, Thomas Alexander, Luciano Bello, Michael J Biercuk, Lev Bishop, Jiayin Chen, Jerry M Chow, Antonio D Córcoles, Daniel Egger, Stefan Filipp, et al. Qiskit backend specifications for openqasm and openpulse experiments. *arXiv preprint arXiv:1809.03452*, 2018.

[24] Rong-Hua Li, Jeffrey Xu Yu, Lu Qin, Rui Mao, and Tan Jin. On random walk based graph sampling. In *2015 IEEE 31st international conference on data engineering*, pages 927–938. IEEE, 2015.

[25] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. Fast random walk with restart and its applications. In *Sixth international conference on data mining (ICDM'06)*, pages 613–622. IEEE, 2006.

[26] Nick Craswell and Martin Szummer. Random walks on the click graph. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 239–246, 2007.

[27] Tingting Li, Ziming Zhao, and Jianwei Yin. Fortuna: Towards efficient selection of high-fidelity link for quantum network in the wild. In *IEEE INFOCOM 2025-IEEE Conference on Computer Communications*, pages 1–10. IEEE, 2025.

[28] Siwei Tan, Mingqian Yu, Andre Python, Yongheng Shang, Tingting Li, Liqiang Lu, and Jianwei Yin. Hyqsat: A hybrid approach for 3-sat problems by integrating quantum annealer with cdcl. In *2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 731–744. IEEE, 2023.

[29] Tingting Li, Ziming Zhao, and Jianwei Yin. Empowering quantum serverless circuit deployment optimization via graph contrastive learning and learning-to-rank co-designed approaches. *IJCAI 2025*, 2025.

[30] Tingting Li, Ziming Zhao, Liqiang Lu, and Jianwei Yin. Quframe: A novel encoding ensemble framework for quantum neural networks. In *2025 International Conference on Quantum Communications, Networking, and Computing (QCNC)*, pages 583–590. IEEE, 2025.

[31] Kaixuan Huang, Zheng-An Wang, Chao Song, Kai Xu, Hekang Li, Zhen Wang, Qiujiang Guo, Zixuan Song, Zhi-Bo Liu, Dongning Zheng, et al. Quantum generative adversarial networks with multiple superconducting qubits. *npj Quantum Information*, 7(1):165, 2021.

[32] Tingting Li, Ziming Zhao, and Jianwei Yin. Task-driven quantum device fingerprint identification via modeling qnn outcome shift induced by quantum noise. In *Companion Proceedings of the ACM on Web Conference 2024*, pages 557–560, 2024.

[33] Tingting Li and Ziming Zhao. Moirai: Optimizing quantum serverless function orchestration via device allocation and circuit deployment. In *ICWS*, pages 707–717. IEEE, 2024.

[34] Ziming Zhao, Tingting Li, and Zhaoxuan Li. Qufm: Towards efficient quantum link fidelity measurements in quantum networks. In *2025 International Conference on Quantum Communications, Networking, and Computing (QCNC)*, pages 516–520. IEEE, 2025.

[35] András Gilyén and Alexander Poremba. Improved quantum algorithms for fidelity estimation. *arXiv preprint arXiv:2203.15993*, 2022.

[36] Qisheng Wang, Zhicheng Zhang, Kean Chen, Ji Guan, Wang Fang, Junyi Liu, and Mingsheng Ying. Quantum algorithm for fidelity estimation. *IEEE Transactions on Information Theory*, 69(1):273–282, 2022.

[37] Hsin-Yuan Huang, Richard Kueng, and John Preskill. Predicting many properties of a quantum system from very few measurements. *Nature Physics*, 16(10):1050–1057, 2020.

[38] Xiaoqian Zhang, Maolin Luo, Zhaodi Wen, Qin Feng, Shengshi Pang, Weiqi Luo, and Xiaoqi Zhou. Direct fidelity estimation of quantum states using machine learning. *Physical Review Letters*, 127(13):130503, 2021.

[39] Xiao-Dong Yu, Jiangwei Shang, and Otfried Gühne. Statistical methods for quantum state verification and fidelity estimation. *Advanced Quantum Technologies*, 5(5):2100126, 2022.

[40] Ji Liu and Huiyang Zhou. Reliability modeling of nisq-era quantum computers. In *2020 IEEE international symposium on workload characterization (IISWC)*, pages 94–105. IEEE, 2020.

[41] Hanrui Wang, Pengyu Liu, Jinglei Cheng, Zhiding Liang, Jiaqi Gu, Zirui Li, Yongshan Ding, Weiwen Jiang, Yiyu Shi, Xuehai Qian, et al. Quest: Graph transformer for quantum circuit reliability estimation. *arXiv preprint arXiv:2210.16724*, 2022.

[42] Marco Cerezo, Alexander Poremba, Lukasz Cincio, and Patrick J Coles. Variational quantum fidelity estimation. *Quantum*, 4:248, 2020.

[43] Ranyiliu Chen, Zhixin Song, Xuanqiang Zhao, and Xin Wang. Variational quantum algorithms for trace distance and fidelity estimation. *Quantum Science and Technology*, 7(1):015019, 2021.

[44] Kok Chuan Tan and Tyler Volkoff. Variational quantum algorithms to estimate rank, quantum entropies, fidelity, and fisher information via purity minimization. *Physical Review Research*, 3(3):033251, 2021.

[45] Tingting Li, Ziming Zhao, Liqiang Lu, Siwei Tan, and Jianwei Yin. Empowering quantum error traceability with moe for automatic calibration. In *2025 Design, Automation & Test in Europe Conference (DATE)*, pages 1–7. IEEE, 2025.

[46] Gadi Aleksandrowicz et al. Qiskit: An open-source framework for quantum computing. *Accessed on: Mar*, 16, 2019.

[47] Kaitlin N Smith, Michael A Perlin, Pranav Gokhale, Paige Frederick, David Owusu-Antwi, Richard Rines, Victory Omole, and Frederic Chong. Clifford-based circuit cutting for quantum simulation. In *Proceedings of the 50th Annual International Symposium on Computer Architecture*, pages 1–13, 2023.

[48] Martin Haenggi, Jeffrey G Andrews, François Baccelli, Olivier Dousse, and Massimo Franceschetti. Stochastic geometry and random graphs for the analysis and design of wireless networks. *IEEE journal on selected areas in communications*, 27(7):1029–1046, 2009.

[49] J Stuart Hunter. The exponentially weighted moving average. *Journal of quality technology*, 18(4):203–210, 1986.

[50] Jianqing Fan, Bai Jiang, and Qiang Sun. Hoeffding's inequality for general markov chains and its applications to statistical learning. *Journal of Machine Learning Research*, 22(139):1–35, 2021.

[51] IBM. IBMQ Quantum. https://quantum-computing.ibm.com/, 2022.

[52] Hezi Zhang, Anbang Wu, Yuke Wang, Gushu Li, Hassan Shapourian, Alireza Shabani, and Yufei Ding. Oneq: A compilation framework for photonic one-way quantum computation. In *Proceedings of the 50th Annual International Symposium on Computer Architecture*, pages 1–14, 2023.

[53] Gushu Li, Yufei Ding, and Yuan Xie. Tackling the qubit mapping problem for nisq-era quantum devices. In *ASPLOS*, 2019.

[54] Gushu Li, Yufei Ding, and Yuan Xie. Towards efficient superconducting quantum processor architecture design. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 1031–1045, 2020.

[55] Gushu Li, Yufei Ding, and Yuan Xie. Eliminating redundant computation in noisy quantum computing simulation. In *2020 57th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2020.

[56] Alexander Zlokapa, Benjamin Villalonga, Sergio Boixo, and Daniel A Lidar. Boundaries of quantum supremacy via random circuit sampling. *npj Quantum Information*, 9(1):36, 2023.

[57] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

[58] IonQ. IonQ Quantum. https://ionq.com/, 2019.

[59] Tingting Li, Liqiang Lu, Ziming Zhao, Ziqi Tan, Siwei Tan, and Jianwei Yin. QuST: Optimizing quantum neural network against spatial and temporal noise biases. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2024.