**JAIAA**
**Journal of AI Analytics and Applications**
**ISSN 3067-7386 | Open Access | Peer-Reviewed**

**RESEARCH ARTICLE**
**Vol. 1, No. 4 (2023), pp. 52–66**
**DOI: 10.63646/jaiaa.2023.010404**

# Bayesian Feedback Analytics for Quantum-Assisted Target Recovery in Security-Critical Data Spaces

Rizal Munadi[1]; Teuku Yuliar Arif[2]; Npurdianta Sembiring[3, *]

[1] Department of Electrical and Computer Engineering, Universitas Malikussaleh, Lhokseumawe 24351, Indonesia
[2] Department of Informatics, Universitas Syiah Kuala, Banda Aceh 23111, Indonesia
[3] Department of Information Systems, Universitas Samudra, Langsa 24416, Indonesia
* Corresponding author: nurdianta.sembiring@unsam.ac.id

**ARTICLE INFO**
Received
July 18, 2023
Revised
September 21, 2023
Accepted
November 12, 2023
Available Online
December 30, 2023
DOI
10.63646/jaiaa.2023.010404

Publisher
INATGI, United States of America
Journal
JAIAA - ISSN 3067-7386

**Abstract**
Grover's search and its amplitude-amplification generalisation give a quadratic speed-up for locating marked entries in an unstructured data space, yet recovering every marked entry rather than a single one introduces a fragile dependence on the assumed number of targets. Because the optimal number of amplification rounds is fixed by this count, an estimate that drifts from the truth either halts the search early—leaving targets undiscovered—or drives it into an unbounded hunt for entries that do not exist. We frame this difficulty as an online inference problem and present a quantum-classical procedure that treats the unknown target count as a random variable, maintaining a posterior over candidate counts that is revised after each measurement through Bayesian feedback analytics. A lightweight classical layer prunes improbable hypotheses, defers updates until the recent outcome record is informative, and protects neighbourhood diversity so that the true count is not discarded prematurely. Across search spaces from 2^12 to 2^24 the method drives the mean number of unrecovered targets close to zero—for a space of 2^24 entries the static baseline left roughly 60.9 targets unfound on average while our procedure left 0.82—and it does so without inflating cost, requiring about 389,400 oracle calls against 408,000 for the baseline. A vulnerability-enumeration case study on large open-source code bases shows the same pattern, recovering all catalogued defects where the fixed-count baseline missed several percent. We argue that adaptive posterior correction should be a default component of any complete-enumeration quantum search in security-critical settings

**Keywords:** Quantum search; amplitude amplification; bayesian inference; quantum-classical hybrid; target enumeration; vulnerability detection; adaptive estimation

## I. INTRODUCTION

Unstructured search—deciding which entries of a large collection satisfy a stated condition when no index or ordering can be exploited—sits at the heart of many computational workloads. On a classical machine the only general strategy is to test entries one at a time, so the expected effort grows linearly with the size of the collection. The quantum search procedure introduced by Grover, and later generalised as amplitude amplification, changes this picture by preparing all entries in superposition and rotating probability mass toward the marked ones, so that a marked entry is observed after a number of queries that scales only with the square root of the collection size (Brassard et al., 2002; Montanaro, 2015). This quadratic advantage has made the procedure a workhorse subroutine in quantum algorithm design, appearing inside counting, optimisation, simulation, and learning routines (Biamonte et al., 2017; Cerezo et al., 2021).

A subtlety that is easy to overlook is that one run of the procedure returns at most one marked entry, chosen uniformly among those that exist. Many applications, however, do not want a single witness; they want the

 

RESEARCH ARTICLE
Vol. 1, No. 4 (2023), pp. 52–66
DOI: 10.63646/jaiaa.2023.010404

complete set. Enumerating every marked entry yields a far richer description of the problem—the full distribution of solutions, the structure of the feasible region, and the relationships among solutions—which in turn supports better downstream decisions. An astronomer screening a catalogue for bodies that could host life wants every candidate, not one; an auditor scanning a code base for a class of defect wants every occurrence, because a single overlooked instance can keep a system exposed (Li et al., 2022; Lu & Xu, 2019).

Recovering the whole set requires repeated application of the search, and because independent runs may return entries that were already seen, naive repetition is wasteful: collecting all marked entries by chance follows the familiar coupon-collector growth and costs on the order of the target count times its logarithm. A cleaner strategy modifies the marking oracle after each discovery so that found entries are no longer marked; every subsequent run then yields a fresh entry, and the number of runs falls to roughly the target count. This is the structure we adopt as our point of comparison throughout the paper.

All of these complete-enumeration schemes share a hidden assumption: that the number of marked entries is known, or has been estimated accurately, before the search begins. The assumed count is not a cosmetic parameter. It fixes the number of amplification rounds applied in each run, and that round count is what concentrates probability on the marked subspace. Several quantum routines estimate the count in advance—quantum counting via phase estimation, and a family of lighter alternatives that avoid controlled operations (Suzuki et al., 2020; Grinko et al., 2021; Wie, 2019; Venkateswaran & O'Donnell, 2021). Every such estimator, however, returns an approximation with a user-chosen tolerance: a tighter tolerance costs more, so practitioners deliberately accept a margin that keeps estimation cheaper than the search it serves.

When only one witness is sought, a loose count estimate is forgiving, because even a sub-optimal round count still leaves a usable success probability. Complete enumeration is far less forgiving. If the assumed count is smaller than the truth, the search stops once that many entries have been gathered and silently abandons the rest. If the assumed count is larger than the truth, the search keeps querying for entries that no longer exist and may never stop on its own. In both regimes the mismatch also pushes the round count away from its optimum, depressing the per-run success probability and inflating the total query budget. The dependence on an accurate count is therefore the central reliability bottleneck for find-all quantum search. Figure 1 sketches the architecture we propose to address it.

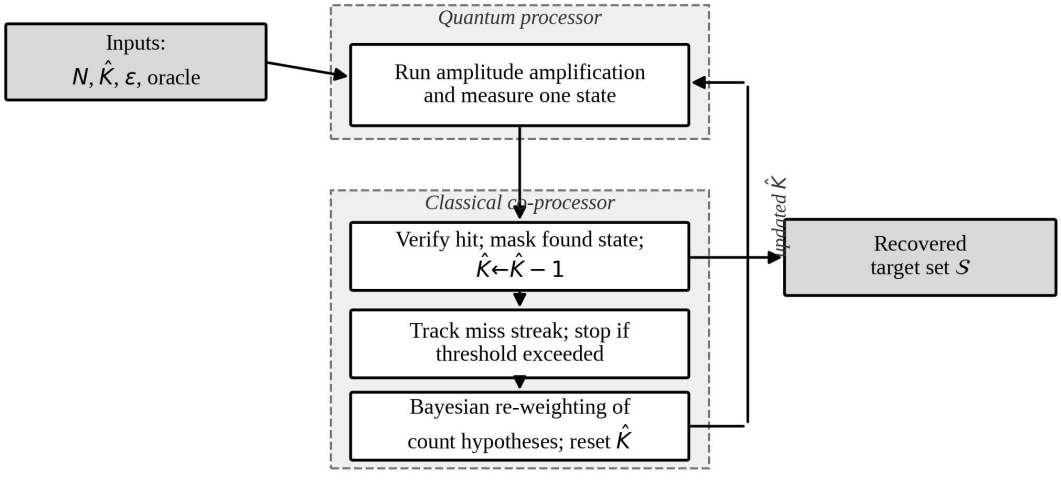


Figure 1. Architecture of the proposed quantum-classical recovery loop with Bayesian feedback on the target count.

Our response is to stop treating the count as a fixed input and instead treat it as a quantity to be learned while the search runs. We keep a probability distribution over candidate counts and revise it after each measurement using the observed pattern of hits and misses, a scheme we call Bayesian feedback analytics. As

evidence accumulates the distribution sharpens around the true count, the working estimate that drives the amplification rounds is reset to the most probable candidate, and the search self-corrects toward complete recovery. The quantum device performs only the amplification and measurement; all inference is classical and cheap, in the spirit of established hybrid quantum-classical designs (Endo et al., 2021; McClean et al., 2016).

This paper makes four contributions. First, we cast complete-enumeration quantum search as an online Bayesian estimation problem and derive the per-measurement likelihood that links an observed outcome to each candidate count. Second, we design a classical control layer that keeps the inference affordable through three mechanisms: pruning low-probability candidates, deferring updates until the recent outcome record actually carries information, and preserving spatial diversity in the candidate set so that the true count and its neighbours are not pruned together. Third, we show analytically that the added inference does not change the asymptotic query complexity, which remains governed by the search itself. Fourth, we evaluate the method in simulation across a wide range of problem sizes and in a vulnerability-enumeration case study, demonstrating that it drives unrecovered targets close to zero while modestly reducing total queries. The remainder of the paper is organised as follows. Section II reviews the necessary background. Section III develops the method. Section IV analyses its cost. Section V reports the empirical study, and Section VI the security case study. Section VII surveys related work and Section VIII concludes.

**Table I. Symbols and quantities used throughout the paper.**

| Symbol | Meaning |
|---|---|
| $N$ | Number of entries in the search space. |
| $n$ | Qubits needed to index the space, $n = \lceil \log_2 N \rceil$. |
| $K$ | True number of marked entries; we assume $K \le \sqrt{N}$. |
| $\hat{K}$ | Working estimate of the marked-entry count. |
| $\varepsilon$ | Tolerance of the initial count estimate, typically of order $\sqrt{N}$. |
| $j$ | Amplification rounds in one run, $j = \lfloor \pi/4 \times \sqrt{(N/\hat{K})} \rfloor$. |
| $\mathscr{H}$ | Hypothesis set: candidate counts with posterior weights. |
| $\rho$ | Pruning fraction applied to $\mathscr{H}$ after each update. |

Table I collects the notation. We write $N$ for the size of the space and $K$ for the unknown number of marked entries, and we assume $K$ does not exceed the square root of $N$; once the marked fraction grows beyond that point, the quadratic edge of quantum search erodes and a classical scan becomes competitive (Zhang & Korepin, 2020). The working estimate is $\hat{K}$, and the inference machinery maintains a hypothesis set $\mathscr{H}$ whose members are candidate values of $K$ paired with posterior weights.

## II. AMPLITUDE AMPLIFICATION AND COMPLETE ENUMERATION

### *A. The amplification primitive*

Consider a space of $N$ entries indexed by $n$ qubits, of which $K$ are marked by an oracle that flips the phase of marked basis states. The procedure begins from the uniform superposition produced by a layer of Hadamard gates, then repeatedly applies an operator that reflects the state about the marked subspace and then about the initial superposition. Each application rotates the state vector by a fixed angle within the two-dimensional plane spanned by the marked and unmarked components, so the probability of measuring a marked entry rises sinusoidally with the number of rounds. After $j$ rounds that probability is well approximated by $\sin^2((2j+1)\theta)$, where the rotation angle satisfies $\sin^2\theta = K/N$. The probability is maximised near $j = \lfloor \pi/4 \times \sqrt{(N/K)} \rfloor$, which is the canonical round count and the source of the quadratic speed-up (Brassard et al., 2002; Roy et al., 2022).

Two features of this expression matter for our purposes. The success probability is periodic in $j$, so overshooting the optimum is as damaging as undershooting it; pushing too far actually drives probability back

out of the marked subspace. And the optimal j depends on K through a square root, so the round count cannot be chosen without a value for K. When K is replaced by an estimate $\hat{K}$, the realised success probability is $\sin^2((2\lfloor \pi/4\times\sqrt{(N/\hat{K})}\rfloor +1)\theta)$ with the true θ, which degrades smoothly as $\hat{K}$ departs from K. Deterministic and fixed-point variants reshape this dependence but do not remove the need for a count when the goal is to enumerate an unknown number of entries (Toyama et al., 2013; Gilliam et al., 2021).

### *B. From one witness to the whole set*

To gather every marked entry, the search is run repeatedly. If the oracle is left unchanged, repeated runs sample marked entries with replacement, and the number of runs needed to see all of them follows coupon-collector behaviour, scaling as the target count times its logarithm. The standard improvement removes each discovered entry from the marked set by editing the oracle, so that runs sample without replacement and the run count drops to approximately the number of marked entries. Decrementing the working estimate by one after each discovery keeps the round count near its moving optimum as the remaining count shrinks. This without-replacement scheme is the baseline against which we measure, and it is the most efficient prior strategy that still assumes a fixed initial count.

### *C. Estimating the count in advance*

Because the round count needs a value for K, a complete-enumeration pipeline normally begins with a counting step. Quantum counting applies phase estimation to the amplification operator and reads the rotation angle, from which K follows; its accuracy is set by the number of evaluation qubits and therefore by the query budget allocated to it (Wie, 2019). Lighter estimators replace the expensive controlled operations with maximum-likelihood post-processing over circuits of differing depths, or with iterative refinement that uses only the amplification operator (Suzuki et al., 2020; Grinko et al., 2021; Giurgica-Tiron et al., 2022; Rall, 2021). All of them deliver an estimate with a tolerance the user selects by trading accuracy for cost. The practical convention is to set the tolerance proportional to the square root of N, which keeps counting cheaper than the enumeration that follows; driving the tolerance toward zero would make counting dominate and erase the quantum advantage. The residual error this convention leaves is exactly the error our method is designed to absorb.

## III. ADAPTIVE RECOVERY WITH BAYESIAN FEEDBACK

We now develop the proposed procedure. The guiding idea is to maintain a posterior distribution over the unknown count and to let the search and the inference inform each other: the search produces hits and misses, those outcomes update the posterior, and the updated posterior resets the working estimate that shapes the next search runs. We describe the inference model first, then the control logic that keeps it cheap and stable.

### *A. A posterior over candidate counts*

Before any run we seed a hypothesis set with the integer candidates that lie within the estimator's tolerance of the initial estimate, that is, every value from $\hat{K}$ minus ε to $\hat{K}$ plus ε. With no reason to favour any candidate we assign them equal prior weight, a uniform prior that reflects genuine ignorance within the tolerance band. As runs proceed, each candidate's weight is the posterior probability that it is the true remaining count given the outcomes observed so far. Over time the weight mass migrates toward the candidate that best explains the record, and the working estimate is reset to whichever candidate currently carries the most weight.

The link between an outcome and a candidate is the per-run success probability. Suppose the search has been run with round count j and a candidate asserts that r marked entries remain. Under that candidate the chance the run returns a fresh marked entry is $p = \sin^2((2j+1) \times \arcsin\sqrt{(r/N)})$. A run that succeeds contributes a factor p to that candidate's likelihood; a run that fails contributes a factor $(1-p)$. Multiplying these factors across the recent outcome record gives the likelihood of the record under each candidate, and Bayes' rule turns prior weights and likelihoods into posterior weights after normalisation. This is the same likelihood structure that underlies maximum-likelihood and Bayesian amplitude estimation, repurposed here to infer a count rather

RESEARCH ARTICLE
Vol. 1, No. 4 (2023), pp. 52–66
DOI: 10.63646/jaiaa.2023.010404

than an amplitude (Wiebe & Granade, 2016; Tanaka et al., 2021; Yamamoto et al., 2024).

Two bookkeeping rules keep the model coherent. Because a candidate represents the number of entries still to be found, its asserted value is decremented by one whenever a genuine discovery occurs, so the whole hypothesis set tracks the shrinking remaining count in step with the search. And any candidate whose asserted value would fall below zero is removed outright, since a negative remaining count is impossible; a candidate that claimed ten entries remain is untenable once eleven have already been found.

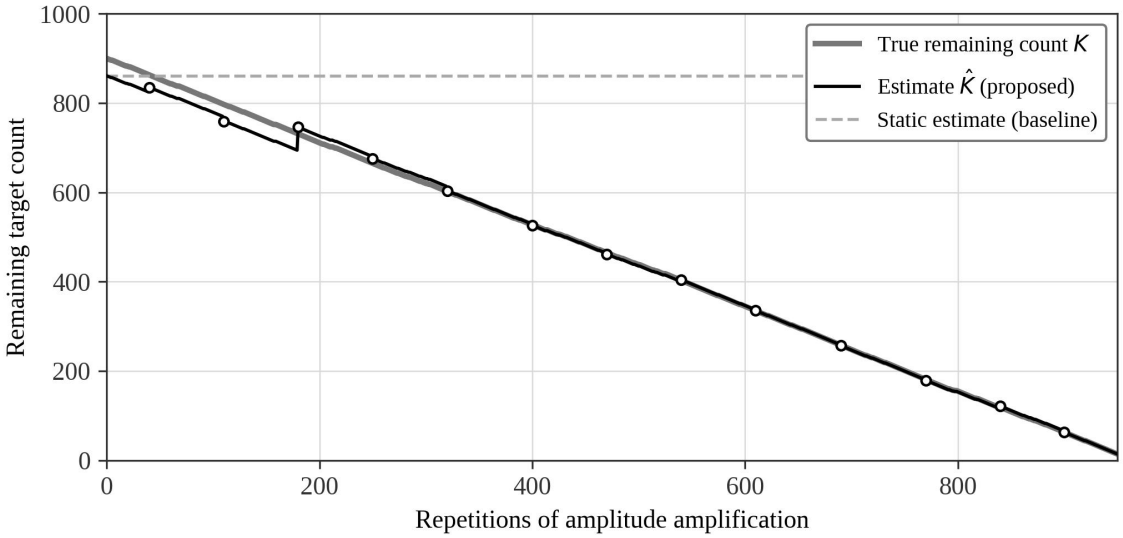


Figure 2. A representative run: the working estimate (solid) tracks the true remaining count (grey) while a static estimate (dashed) does not.

Figure 2 illustrates a single representative run for a space of about four million entries seeded with an initial under-estimate. Early on, when the record is short and the inherent randomness of measurement dominates, the working estimate fluctuates and a few unlucky outcomes can nudge it the wrong way. As the record lengthens the posterior sharpens and the estimate locks onto the true remaining count, following it down to zero. A static estimate, by contrast, holds its initial value and would terminate the search the moment that many entries had been collected, abandoning the remainder.

### *B. When to update, and what to prune*

Updating the posterior after every single run would be both costly and counter-productive, so the control layer governs the cadence and the size of the hypothesis set. Updates are batched: the procedure waits a fixed interval of runs and only then revises the posterior using the recent outcome record, which is itself capped at a bounded length so that memory and update cost stay constant. A short, recent record is sufficient to estimate the candidate weights accurately, and bounding it prevents stale outcomes from diluting the signal.

A second guard decides whether an update is worth doing at all. If the recent record contains nothing but successes, the working estimate is almost certainly close to correct—an unbroken run of fresh discoveries is exactly what a good estimate produces—and revising it would only burn computation. The procedure therefore skips the update unless the recent record contains at least one failure. Failures are the informative events: they signal that the round count is no longer concentrating probability where targets actually are, which is the symptom of a count that has drifted. This guard is especially valuable early, when the relative error of the estimate is small even if the absolute estimate is wrong, so the success probability stays high and early updates would be redundant. As both the remaining count and the estimate shrink, the relative error grows, failures become more frequent, and updates fire exactly when they are needed.

To stop the hypothesis set from growing without bound—which would eventually make each update cost scale with N and dissolve the quantum advantage—the procedure prunes a fixed fraction of candidates after

RESEARCH ARTICLE
Vol. 1, No. 4 (2023), pp. 52–66
DOI: 10.63646/jaiaa.2023.010404

each update. Pruning is the most delicate step. Removing the globally least-probable candidates is tempting but dangerous, because probability tends to vary smoothly across neighbouring counts, so the globally weakest candidates often form a contiguous block. In the early, noisy phase that block can momentarily include the true count and its neighbours, and deleting them as a group would orphan the search far from the truth with no path back. We instead prune stratified: the candidate list is divided into consecutive blocks whose length is the reciprocal of the pruning fraction, and exactly one candidate—the weakest in its block—is removed from each block. This removes the intended fraction overall while guaranteeing that survivors remain spread across the whole range of counts, preserving the diversity that lets the posterior recover if an early estimate was unlucky.

Termination uses the same failure signal in a coarser form. The procedure tracks consecutive failures, and if that streak exceeds a preset threshold it concludes that no further targets remain and stops. The threshold is generous enough that an unlucky burst of misses during normal operation does not trigger a false stop, yet small relative to the query budget so that a genuinely exhausted search ends promptly rather than querying indefinitely for entries that no longer exist.

**Table II. Control parameters of the adaptive layer and their roles.**

| Parameter | Role | Typical setting | Effect of mis-setting |
|---|---|---|---|
| Update interval | Runs between posterior revisions | 20 runs | Too small wastes effort; too large slows correction |
| Record length | Recent outcomes kept for the likelihood | 20 outcomes | Too short is noisy; too long admits stale evidence |
| Failure guard | Require ≥1 recent failure to update | enabled | Disabling causes redundant early updates |
| Pruning fraction ρ | Share of hypotheses removed per update | 0.05 | Too high may drop the truth; too low inflates cost |
| Stop threshold | Consecutive misses before halting | 50 misses | Too low ends early; too high wastes queries |

Table II lists the five control parameters together with their roles and the consequences of poor choices. The settings shown are the values used in our experiments; they were selected once on small instances and held fixed across every problem size, which is itself evidence that the method is not delicate with respect to tuning. The failure guard and stratified pruning are the two ideas that most distinguish the design, and the case-study results in Section VI quantify what they buy.

## IV. COST AND RESOURCE ANALYSIS

A method that recovers more targets is only useful if it does not pay for that recovery with a worse asymptotic cost. We therefore account for the additional work the adaptive layer introduces and show that it stays below the cost of the search it supports. We measure cost in oracle calls, since those dominate runtime, and we treat the best case (an accurate estimate, no failures) and the worst case (a poor estimate that triggers updates at every interval) separately.

### *A. Query complexity*

The baseline without-replacement search applies the amplification operator a number of times that starts near $\sqrt{(N/K)}$ per run and grows as the remaining count falls; summing over all discoveries and approximating the sum by an integral gives a total proportional to $\sqrt{(NK)}$ oracle calls (Brassard et al., 2002). Our procedure inherits this dominant term unchanged, because the quantum work per run is identical. In the best case the failure guard is never satisfied, no updates run, and the total stays at $\sqrt{(NK)}$. In the worst case updates fire at every interval. Each update processes the surviving hypotheses, and because stratified pruning keeps roughly a fixed multiple of the tolerance many candidates alive—the geometric series of retained fractions sums to a constant—the per-update work is proportional to the tolerance, with each candidate costing a constant amount

because the outcome record is bounded. Summed over the updates performed during the search, the inference contributes a term proportional to the tolerance, which under the conventional choice of tolerance is of order √N. Since the marked count is at most √N, this inference term is dominated by the √(NK) search term, so the overall complexity is unchanged.

It is worth noting where this leaves the preliminary counting step. With the conventional tolerance of order √N, quantum counting itself costs on the order of √N oracle calls, again dominated by the search. Tightening the tolerance toward one would force counting toward a cost of order N, larger than the search and asymptotically no better than a classical scan. The conventional tolerance is thus the sweet spot, and it is precisely the regime in which a loose estimate must be corrected during the search—the regime our method targets (Suzuki et al., 2020; Venkateswaran & O'Donnell, 2021).

### *B. Memory*

Both the baseline and the proposed method store the set of recovered entries, which costs memory proportional to the number of marked entries. The proposed method additionally stores the hypothesis set, which holds at most a number of candidates proportional to the tolerance together with their weights. Under the conventional tolerance both contributions are of order √N, so the adaptive layer does not raise the asymptotic memory footprint. On the quantum side the two methods are identical, using the same index register of $\lceil \log_2 N \rceil$ qubits and the same amplification circuitry, so quantum memory remains logarithmic in N (Bharti et al., 2022).

A practical numerical safeguard deserves mention. Posterior weights can become extremely small as evidence accumulates, risking floating-point underflow. We store and update log-weights instead of weights, turning the products in the likelihood into sums and the normalisation into a stable log-sum-exp; this preserves accuracy without affecting the asymptotic analysis and mirrors standard practice in Bayesian estimation pipelines (Wiebe & Granade, 2016).

**Table III. Cost of each component (oracle calls unless noted); f(N) is the oracle cost.**

| Component | Best case | Worst case | Notes |
|---|---|---|---|
| Amplification (search) | √(NK) × f(N) | √(NK) × f(N) | Dominant term in both cases |
| Outcome logging | K | K | Constant work per run |
| Posterior update | constant | ε | Skipped unless a failure is present |
| Hypothesis pruning | constant | ε | Stratified, one per block |
| Dominant total | √(NK) × f(N) | √(NK) × f(N) | Matches the baseline |

Table III summarises the accounting. The shaded final row is the headline: the dominant term is the search itself in both the best and worst cases, so the adaptive layer is asymptotically free. Under the standard assumptions that the oracle costs at least logarithmically in N, that the marked count does not exceed √N, and that the estimator tolerance is of order √N, the search term governs throughout.

## V. EMPIRICAL EVALUATION

We evaluate the method against the without-replacement baseline on two questions: how many targets each method fails to recover, and how many oracle calls each consumes. Coverage is the primary metric, since complete recovery is the goal; query count measures speed, since oracle calls dominate runtime. Figure 3 shows the evaluation pipeline.

**JAIAA** **RESEARCH ARTICLE**
**Journal of AI Analytics and Applications** **Vol. 1, No. 4 (2023), pp. 52–66**
**ISSN 3067-7386 | Open Access | Peer-Reviewed** **DOI: 10.63646/jaiaa.2023.010404**

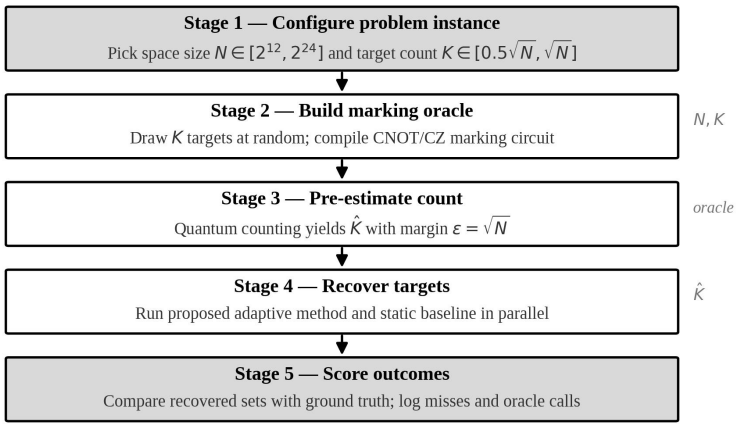


Figure 3. Five-stage evaluation pipeline used to compare the adaptive method against the static baseline.

### *A. Setup*

Each trial fixes a space size N, drawn from 2^12 up to 2^24 to probe scaling, and a target count K drawn from one half of √N up to √N, stepping by one. For each pair we sample K marked entries at random and build a phase-marking oracle from CNOT and controlled-Z gates. We then run quantum counting on that oracle to obtain the initial estimate, setting the tolerance to √N to balance reliability against cost, exactly as a real deployment would. Both methods receive only the estimate, the oracle, and the space; neither is given the true target set or the true count. To prevent the over-estimate regime from looping forever, both methods stop after fifty consecutive failures. We then compare each method's recovered set against the ground truth and record misses and oracle calls, repeating every configuration one hundred times and reporting averages.

Smaller instances were executed on a standard circuit simulator to validate correctness; for the larger sizes, full circuit simulation is prohibitively slow, so we used a purpose-built simulator that reproduces the measurement statistics of the amplification procedure and was cross-checked against the circuit simulator on the small instances where both are feasible. This mirrors the methodology used in other large-scale studies of amplitude-based routines, where exact state-vector simulation becomes infeasible well before the interesting regime (Stamatopoulos et al., 2020; Gacon et al., 2020).

### *B. Coverage*

Figure 4 reports the mean number of unrecovered targets as the space grows. The static baseline degrades steadily: as N increases the fixed initial estimate is, on average, further from the truth, and the gap translates directly into abandoned targets, reaching roughly sixty unrecovered targets at the largest size and frequently losing between five and twenty percent of the marked set. The adaptive method stays essentially flat near zero across the entire range, never averaging more than about one unrecovered target. The shaded region between the curves is the coverage that adaptive correction recovers.

**JAIAA** **RESEARCH ARTICLE**
**Journal of AI Analytics and Applications** **Vol. 1, No. 4 (2023), pp. 52–66**
**ISSN 3067-7386 | Open Access | Peer-Reviewed** **DOI: 10.63646/jaiaa.2023.010404**

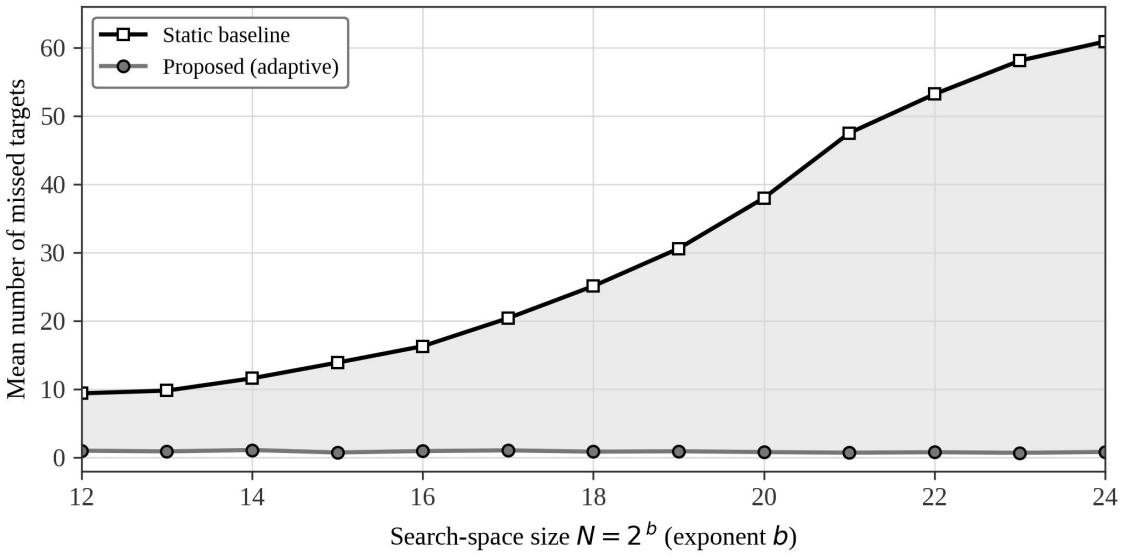


Figure 4. Mean unrecovered targets versus search-space size for the static baseline and the adaptive method.

The mechanism behind the gap is the one Figure 2 illustrated at the level of a single run. Because the adaptive method continually resets its working estimate toward the most probable count, it keeps the round count near optimal as the remaining count falls, which keeps the per-run success probability high and prevents both premature termination and aimless over-searching. The baseline, holding a single estimate fixed, drifts off the optimum as the remaining count changes and pays for it in missed targets. For a concrete point, at a space of 2^15 entries with one hundred and ten marked entries the baseline left about 15.3 targets unfound on average—near fourteen percent—while the adaptive method left about 0.74, well under one percent.

### *C. Query cost*

Figure 5 reports mean oracle calls on a logarithmic scale. Both methods grow with N, as expected, since larger spaces hold more targets to recover. The adaptive method consistently uses slightly fewer calls than the baseline across the whole range. This may seem counter-intuitive given that the adaptive layer performs extra work, but the inference cost is dominated by the search, and the improvement in per-run success probability more than repays it: keeping the round count near optimal means fewer wasted runs and fewer redundant queries. At the largest size the baseline averaged about 408,000 oracle calls and the adaptive method about 389,400, a reduction of roughly four to five percent alongside the large gain in coverage.

**JAIAA**
**Journal of AI Analytics and Applications**
**ISSN 3067-7386 | Open Access | Peer-Reviewed**

**RESEARCH ARTICLE**
**Vol. 1, No. 4 (2023), pp. 52–66**
**DOI: 10.63646/jaiaa.2023.010404**

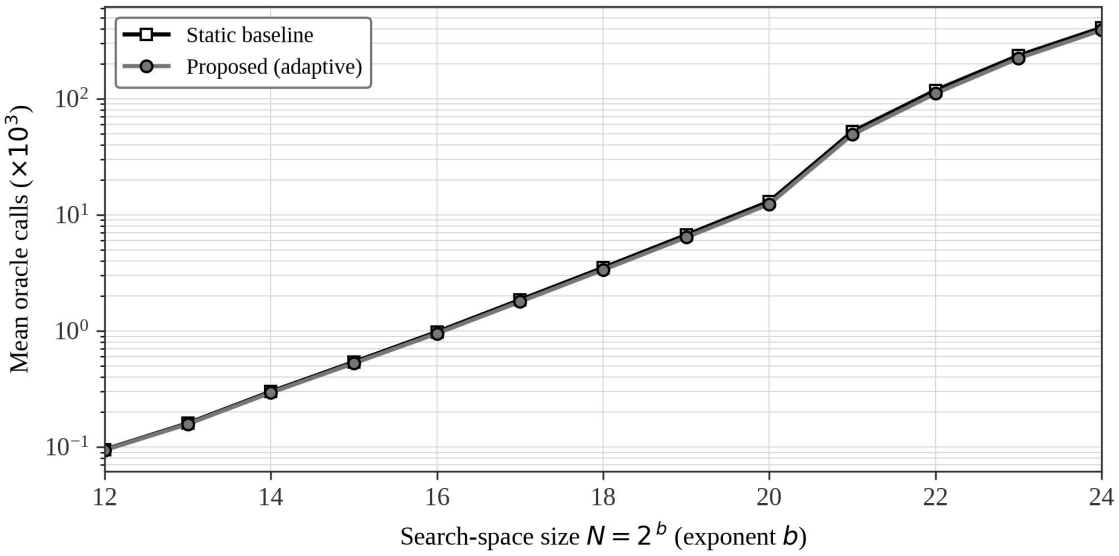


Figure 5. Mean oracle calls versus search-space size (log scale) for the static baseline and the adaptive method.

Taken together, the two figures show that adaptive correction is not a trade of speed for coverage. The method improves coverage dramatically—from losing a meaningful fraction of targets to losing essentially none—while at the same time shaving a few percent off the query budget. The reliability gain is therefore free, or better than free, in the regime studied.

## VI. CASE STUDY: ENUMERATING SOFTWARE VULNERABILITIES

To show how the simulated results translate to a security-critical workload, we model the task of finding every vulnerable location in a large code base. We treat each line of code as an entry in the search space and each catalogued vulnerability as a marked entry, so that complete enumeration corresponds to recovering every defect. This abstraction lets us ask, for realistic sizes, how many vulnerabilities a quantum-assisted scan would miss under the static baseline versus the adaptive method. The framing follows a growing literature that pairs learning-based defect localisation with exhaustive enumeration of candidate sites (Chakraborty et al., 2022; Fu & Tantithamthavorn, 2022).

We draw counts from public vulnerability records for two widely deployed open-source projects of very different scales—a large operating-system kernel on the order of tens of millions of lines with several thousand catalogued defects, and a security library on the order of hundreds of thousands of lines with a few hundred defects. We grade defects by severity following standard scoring practice, run both methods one hundred times per project under the matching space and target sizes using the Section V procedure, and compare recovered sets against the catalogue.

**Table IV. Mean unrecovered vulnerabilities and oracle calls; one hundred trials per project.**

| Category | Catalogued | Baseline missed | Adaptive missed |
|---|---|---|---|
| Kernel — low severity | 587 | 49.8 | 0 |
| Kernel — medium severity | 1,970 | 164.9 | 0 |
| Kernel — high severity | 970 | 81.8 | 0 |
| Kernel — critical | 83 | 6.6 | 0 |
| Kernel — all defects (K = 3,610) | 3,610 | 303 (8.4%) | 0 |
| Kernel — mean oracle calls | — | 794,300 | 653,800 |

RESEARCH ARTICLE
Vol. 1, No. 4 (2023), pp. 52–66
DOI: 10.63646/jaiaa.2023.010404

| | | | |
|---|---|---|---|
| Library — all defects (K = 311) | 311 | 13 (4.2%) | 0 |
| Library — mean oracle calls | — | 37,700 | 18,700 |

Table IV gives the outcome. On the kernel the static baseline left about three hundred defects unrecovered on average—roughly eight percent of the catalogue—including dozens of high-severity and several critical cases. On the smaller library it missed about four percent. The adaptive method recovered every catalogued defect in every severity class across all trials, missing none. The cost columns echo Section V: the adaptive method also used substantially fewer oracle calls, cutting the kernel estimate from roughly $7.9\times10^5$ to $6.5\times10^5$ and the library estimate from about $3.8\times10^4$ to $1.9\times10^4$, because its better-calibrated round counts succeed more often.

The operational reading is stark. In security enumeration the cost of a miss is not symmetric with the cost of an extra query: a single overlooked high-severity defect can mean remote compromise, credential theft, or prolonged outage, whereas a few thousand extra queries are routine. Reducing misses from hundreds to zero therefore changes the risk profile qualitatively, not just quantitatively (Lu & Xu, 2019; Xu et al., 2021). More broadly, the method suits any setting where every valid match must be found and the match count is unknown in advance—safety analysis that must locate every failure mode, conformational search that must not miss a low-energy state, or database queries with complex predicates whose result size is as hard to predict as the query itself. In each of these the fixed-count assumption is exactly what fails, and adaptive correction is exactly what restores reliable enumeration.

## VII. RELATED WORK

### *A. Enumeration with a known count*

The largest body of prior work assumes the count is known or pre-estimated. The simplest approach runs the search a large fixed number of times and keeps the most frequent outcomes; it usually finds most targets but wildly overspends on runs and offers no optimality guarantee. The coupon-collector approach runs until the desired number of distinct targets appears, paying the logarithmic overhead of duplicates. The without-replacement approach edits the oracle after each discovery to suppress duplicates, reaching near-linear run counts and the $\sqrt{(NK)}$ query bound we use as our baseline (Brassard et al., 2002; Montanaro, 2015). A hybrid line folds partial discovery into the counting step itself, recovering some targets while estimating the count, but its worst case still carries the logarithmic factor. Every one of these methods inherits the fixed-count fragility we address: a wrong count means missed targets, wasted runs, or non-termination.

### *B. Search without a prior count*

A separate line removes the need for a prior count. One family grows the assumed count by trial and error, starting small and enlarging it whenever the search fails, while another adds ancilla machinery to monitor the success probability before measuring, as in fixed-point and oblivious amplitude amplification (Toyama et al., 2013; Gilliam et al., 2021; Roy et al., 2022). These schemes avoid a separate estimator but tend to pay a per-target query cost on the order of $\sqrt{N}$ and still need many repetitions for complete enumeration, giving an overall cost comparable to or worse than estimation-based methods. Our work keeps the efficiency of the estimation route while removing its central weakness, by correcting the estimate online rather than abandoning it.

### *C. Bayesian and amplitude-estimation methods*

Our inference machinery draws directly on the amplitude- and phase-estimation literature. Maximum-likelihood and Bayesian amplitude estimation combine measurements from circuits of varying depth to infer an amplitude, and rejection-filtering and related Bayesian phase estimators maintain and update a posterior over an unknown phase from sequential measurements (Suzuki et al., 2020; Tanaka et al., 2021; Tanaka et al., 2022; Wiebe & Granade, 2016; Paesani et al., 2017; O'Brien et al., 2019). We adapt this posterior-update viewpoint to a discrete quantity—the count of remaining targets—and embed it in a closed loop that feeds the inferred count back into the search. Iterative and low-depth amplitude-estimation variants are complementary: any of

RESEARCH ARTICLE
Vol. 1, No. 4 (2023), pp. 52–66
DOI: 10.63646/jaiaa.2023.010404

them could supply our initial estimate, and our layer would then keep that estimate accurate as the search consumes targets (Grinko et al., 2021; Giurgica-Tiron et al., 2022; Plekhanov et al., 2022; Rall, 2021; Nakaji, 2020; van den Berg, 2021).

### *D. Algorithmic versus hardware error*

It is important to separate the error we address from hardware noise. A large literature mitigates the errors that decoherence and imperfect gates inject into quantum computations, through error mitigation and correction techniques that spend extra quantum resources (Cai et al., 2023; Endo et al., 2021; Bharti et al., 2022). The error we target is different in kind: it arises from the estimation algorithm itself and would persist even on flawless hardware, because a count estimator with finite tolerance simply returns an approximate count. Our remedy is classical post-processing that refines the estimate from observed outcomes, so it composes with, rather than competes against, hardware error handling. The broader programme of hybrid quantum-classical computing—where a classical co-processor steers a quantum subroutine—frames our contribution, and surveys of quantum computing in industry and security situate why complete, reliable enumeration matters in practice (Lu et al., 2023; Ye & Lu, 2022; Lu et al., 2024; Lu & Yang, 2024; Egger et al., 2020). Foundational demonstrations of programmable quantum advantage and shallow-circuit separations indicate that the hardware substrate for such routines is maturing (Arute et al., 2019; Bravyi et al., 2018), while cryptanalytic studies of Grover-based key search underline the security stakes of accurate quantum search at scale (Jaques et al., 2020; Rahman & Paul, 2022; Mandal et al., 2024; Wang et al., 2022). Learning-based feature maps and circuit-learning methods suggest further ways the classical layer could be enriched (Havlíček et al., 2019; Schuld et al., 2019; Mitarai et al., 2018).

## VIII. CONCLUSION

Complete enumeration with quantum search is only as reliable as the count it assumes, and that count is, in practice, an estimate with a deliberately loose tolerance. We have shown that the resulting fragility can be removed without abandoning the estimation route, by treating the count as a latent variable and correcting it online through Bayesian feedback. The classical control layer—batched and guarded updates, stratified pruning that protects diversity, and a failure-streak stopping rule—keeps the inference cheap enough that the asymptotic query and memory costs match the search itself, while a log-domain implementation keeps it numerically robust.

Empirically the method drives unrecovered targets close to zero across four orders of magnitude in space size and does so while modestly lowering the query budget, and a vulnerability-enumeration case study shows the same behaviour on realistic data, recovering every catalogued defect where a fixed-count baseline missed several percent including critical cases. Because the gain in coverage comes without a cost penalty, we argue that adaptive posterior correction belongs in the default toolkit for any complete-enumeration quantum search whose target count is uncertain, especially in security-critical settings where a single missed match is consequential.

Several directions remain open. The update could be made smoother, replacing the jump to the most probable candidate with a damped move that reduces transient swings and improves convergence stability. The posterior could carry a richer model of correlated outcomes, or be coupled to hardware-noise estimates so that algorithmic and physical error are corrected jointly. And the feedback principle could be applied to other quantum subroutines whose performance hinges on a pre-estimated parameter, wherever an online classical estimate can be fed back into the quantum loop.

## AUTHOR CONTRIBUTIONS

| Author | Contribution |
|---|---|
| Rizal Munadi | Conceptualization, methodology, writing – original draft |
| Teuku Yuliar Arif | Software, formal analysis, validation, visualization |

| | |
|---|---|
| Npurdianta Sembiring | Supervision, writing – review & editing, project administration |

## DECLARATIONS

Conflicts of interest: The authors declare no competing financial interests or personal relationships that could have influenced the work reported in this manuscript.

Data availability: The simulator and analysis scripts that reproduce the reported figures and tables are available from the corresponding author upon reasonable request. No proprietary dataset is redistributed in this manuscript.

Funding: This research received no external funding.

Ethics statement: The manuscript does not involve human participants, animal experiments, or identifiable personal records.

## ABOUT THE AUTHORS

Rizal Munadi is affiliated with the Department of Electrical and Computer Engineering, Universitas Malikussaleh, Indonesia. His research interests include quantum algorithms, network security, and applied computational methods for engineering systems.

Teuku Yuliar Arif is affiliated with the Department of Informatics, Universitas Syiah Kuala, Indonesia. His work focuses on quantum-classical hybrid computing, machine learning, and the simulation of quantum search procedures.

Npurdianta Sembiring is affiliated with the Department of Information Systems, Universitas Samudra, Indonesia. Her research addresses secure information systems, software vulnerability analysis, and the application of emerging computing paradigms to data-intensive problems.

## REFERENCES

Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J. C., Barends, R., et al. (2019). Quantum supremacy using a programmable superconducting processor. Nature, 574(7779), 505–510. https://doi.org/10.1038/s41586-019-1666-5

Bharti, K., Cervera-Lierta, A., Kyaw, T. H., Haug, T., Alperin-Lea, S., Anand, A., et al. (2022). Noisy intermediate-scale quantum (NISQ) algorithms. Reviews of Modern Physics, 94(1), 015004. https://doi.org/10.1103/RevModPhys.94.015004

Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., & Lloyd, S. (2017). Quantum machine learning. Nature, 549(7671), 195–202. https://doi.org/10.1038/nature23474

Brassard, G., Hoyer, P., Mosca, M., & Tapp, A. (2002). Quantum amplitude amplification and estimation. Contemporary Mathematics, 305, 53–74. https://doi.org/10.1090/conm/305/05215

Bravyi, S., Gosset, D., & König, R. (2018). Quantum advantage with shallow circuits. Science, 362(6412), 308–311. https://doi.org/10.1126/science.aar3106

Cai, Z., Babbush, R., Benjamin, S. C., Endo, S., Huggins, W. J., Li, Y., McClean, J. R., & O'Brien, T. E. (2023). Quantum error mitigation. Reviews of Modern Physics, 95(4), 045005. https://doi.org/10.1103/RevModPhys.95.045005

Cerezo, M., Arrasmith, A., Babbush, R., Benjamin, S. C., Endo, S., Fujii, K., McClean, J. R., Mitarai, K., Yuan, X., Cincio, L., & Coles, P. J. (2021). Variational quantum algorithms. Nature Reviews Physics, 3(9), 625–644. https://doi.org/10.1038/s42254-021-00348-9

Chakraborty, S., Krishna, R., Ding, Y., & Ray, B. (2022). Deep learning based vulnerability detection: Are we there yet? IEEE Transactions on Software Engineering, 48(9), 3280–3296. https://doi.org/10.1109/TSE.2021.3087402

Egger, D. J., Gambella, C., Marecek, J., McFaddin, S., Mevissen, M., Raymond, R., Simonetto, A., Woerner, S., & Yndurain, E. (2020). Quantum computing for finance: State-of-the-art and future prospects. IEEE Transactions on Quantum Engineering, 1, 3101724. https://doi.org/10.1109/TQE.2020.3030314

Endo, S., Cai, Z., Benjamin, S. C., & Yuan, X. (2021). Hybrid quantum-classical algorithms and quantum error mitigation. Journal of the Physical Society of Japan, 90(3), 032001. https://doi.org/10.7566/JPSJ.90.032001

Fu, M., & Tantithamthavorn, C. (2022). LineVul: A transformer-based line-level vulnerability prediction. In Proceedings of the 19th

International Conference on Mining Software Repositories (MSR), 608–620. https://doi.org/10.1145/3524842.3528452

Gacon, J., Zoufal, C., & Woerner, S. (2020). Quantum-enhanced simulation-based optimization. In 2020 IEEE International Conference on Quantum Computing and Engineering (QCE), 47–55. https://doi.org/10.1109/QCE49297.2020.00017

Gilliam, A., Woerner, S., & Gonciulea, C. (2021). Grover adaptive search for constrained polynomial binary optimization. Quantum, 5, 428. https://doi.org/10.22331/q-2021-04-08-428

Giurgica-Tiron, T., Kerenidis, I., Labib, F., Prakash, A., & Zeng, W. (2022). Low depth algorithms for quantum amplitude estimation. Quantum, 6, 745. https://doi.org/10.22331/q-2022-06-27-745

Grinko, D., Gacon, J., Zoufal, C., & Woerner, S. (2021). Iterative quantum amplitude estimation. npj Quantum Information, 7(1), 52. https://doi.org/10.1038/s41534-021-00379-1

Havlíček, V., Córcoles, A. D., Temme, K., Harrow, A. W., Kandala, A., Chow, J. M., & Gambetta, J. M. (2019). Supervised learning with quantum-enhanced feature spaces. Nature, 567(7747), 209–212. https://doi.org/10.1038/s41586-019-0980-2

Jaques, S., Naehrig, M., Roetteler, M., & Virdia, F. (2020). Implementing Grover oracles for quantum key search on AES and LowMC. In Advances in Cryptology – EUROCRYPT 2020, 280–310. https://doi.org/10.1007/978-3-030-45724-2_10

Li, Z., Zou, D., Xu, S., Jin, H., Zhu, Y., & Chen, Z. (2022). SySeVR: A framework for using deep learning to detect software vulnerabilities. IEEE Transactions on Dependable and Secure Computing, 19(4), 2244–2258. https://doi.org/10.1109/TDSC.2021.3051525

Lu, W., Lu, Y., Li, J., Sigov, A., Ratkin, L., & Ivanov, L. A. (2024). Quantum machine learning: Classifications, challenges, and solutions. Journal of Industrial Information Integration, 42, 100736. https://doi.org/10.1016/j.jii.2024.100736

Lu, Y., & Xu, L. D. (2019). Internet of Things (IoT) cybersecurity research: A review of current research topics. IEEE Internet of Things Journal, 6(2), 2103–2115. https://doi.org/10.1109/JIOT.2018.2869847

Lu, Y., & Yang, J. (2024). Quantum financing system: A survey on quantum algorithms, potential scenarios and open research issues. Journal of Industrial Information Integration, 41, 100663. https://doi.org/10.1016/j.jii.2024.100663

Lu, Y., Sigov, A. S., Ratkin, L., Ivanov, L. A., & Zuo, M. (2023). Quantum computing and industrial information integration: A review. Journal of Industrial Information Integration, 35, 100511. https://doi.org/10.1016/j.jii.2023.100511

Mandal, S., Anand, R., Rahman, M., Sarkar, S., & Isobe, T. (2024). Implementing Grover's on AES-based AEAD schemes. Scientific Reports, 14, 21105. https://doi.org/10.1038/s41598-024-71725-4

McClean, J. R., Romero, J., Babbush, R., & Aspuru-Guzik, A. (2016). The theory of variational hybrid quantum-classical algorithms. New Journal of Physics, 18(2), 023023. https://doi.org/10.1088/1367-2630/18/2/023023

Mitarai, K., Negoro, M., Kitagawa, M., & Fujii, K. (2018). Quantum circuit learning. Physical Review A, 98(3), 032309. https://doi.org/10.1103/PhysRevA.98.032309

Montanaro, A. (2015). Quantum speedup of Monte Carlo methods. Proceedings of the Royal Society A, 471(2181), 20150301. https://doi.org/10.1098/rspa.2015.0301

Nakaji, K. (2020). Faster amplitude estimation. Quantum Information and Computation, 20(13–14), 1109–1123. https://doi.org/10.26421/QIC20.13-14-2

O'Brien, T. E., Tarasinski, B., & Terhal, B. M. (2019). Quantum phase estimation of multiple eigenvalues for small-scale (noisy) experiments. New Journal of Physics, 21(2), 023022. https://doi.org/10.1088/1367-2630/aafb8e

Paesani, S., Gentile, A. A., Santagati, R., Wang, J., Wiebe, N., Tew, D. P., O'Brien, J. L., & Thompson, M. G. (2017). Experimental Bayesian quantum phase estimation on a silicon photonic chip. Physical Review Letters, 118(10), 100503. https://doi.org/10.1103/PhysRevLett.118.100503

Peruzzo, A., McClean, J., Shadbolt, P., Yung, M.-H., Zhou, X.-Q., Love, P. J., Aspuru-Guzik, A., & O'Brien, J. L. (2014). A variational eigenvalue solver on a photonic quantum processor. Nature Communications, 5, 4213. https://doi.org/10.1038/ncomms5213

Plekhanov, K., Rosenkranz, M., Fiorentini, M., & Lubasch, M. (2022). Variational quantum amplitude estimation. Quantum, 6, 670. https://doi.org/10.22331/q-2022-03-17-670

Preskill, J. (2018). Quantum computing in the NISQ era and beyond. Quantum, 2, 79. https://doi.org/10.22331/q-2018-08-06-79

Rahman, M., & Paul, G. (2022). Grover on KATAN: Quantum resource estimation. IEEE Transactions on Quantum Engineering, 3, 3100409. https://doi.org/10.1109/TQE.2022.3140376

Rall, P. (2021). Faster coherent quantum algorithms for phase, energy, and amplitude estimation. Quantum, 5, 566. https://doi.org/10.22331/q-2021-10-19-566

Roy, T., Jiang, L., & Schuster, D. I. (2022). Deterministic Grover search with a restricted oracle. Physical Review Research, 4(2), L022013. https://doi.org/10.1103/PhysRevResearch.4.L022013

Schuld, M., Bergholm, V., Gogolin, C., Izaac, J., & Killoran, N. (2019). Evaluating analytic gradients on quantum hardware. Physical Review A, 99(3), 032331. https://doi.org/10.1103/PhysRevA.99.032331

Somma, R. D. (2019). Quantum eigenvalue estimation via time series analysis. New Journal of Physics, 21(12), 123025. https://doi.org/10.1088/1367-2630/ab5c60

Stamatopoulos, N., Egger, D. J., Sun, Y., Zoufal, C., Iten, R., Shen, N., & Woerner, S. (2020). Option pricing using quantum computers. Quantum, 4, 291. https://doi.org/10.22331/q-2020-07-06-291

Suzuki, Y., Hisanaga, K., Raymond, R., Tanaka, T., Onodera, T., & Yamamoto, N. (2021). Modified Grover operator for quantum amplitude estimation. New Journal of Physics, 23(8), 083031. https://doi.org/10.1088/1367-2630/ac19da

Suzuki, Y., Uno, S., Raymond, R., Tanaka, T., Onodera, T., & Yamamoto, N. (2020). Amplitude estimation without phase estimation. Quantum Information Processing, 19(2), 75. https://doi.org/10.1007/s11128-019-2565-2

Tanaka, T., Suzuki, Y., Uno, S., Raymond, R., Onodera, T., & Yamamoto, N. (2021). Amplitude estimation via maximum likelihood on noisy quantum computer. Quantum Information Processing, 20(9), 293. https://doi.org/10.1007/s11128-021-03215-9

Tanaka, T., Uno, S., Onodera, T., Yamamoto, N., & Suzuki, Y. (2022). Noisy quantum amplitude estimation without noise estimation. Physical Review A, 105(1), 012411. https://doi.org/10.1103/PhysRevA.105.012411

Toyama, F. M., van Dijk, W., & Nogami, Y. (2013). Quantum search with certainty based on modified Grover algorithms: Optimum choice of parameters. Quantum Information Processing, 12(5), 1897–1914. https://doi.org/10.1007/s11128-012-0498-0

van den Berg, E. (2021). Iterative quantum phase estimation with optimized sample complexity. Physical Review A, 105(2), 022427. https://doi.org/10.1103/PhysRevA.105.022427

Venkateswaran, R., & O'Donnell, R. (2021). Quantum approximate counting with nonadaptive Grover iterations. In 38th International Symposium on Theoretical Aspects of Computer Science (STACS 2021), 59:1–59:12. https://doi.org/10.4230/LIPIcs.STACS.2021.59

Wang, Z.-G., Wei, S.-J., & Long, G.-L. (2022). A quantum circuit design of AES requiring fewer quantum qubits and gate operations. Frontiers of Physics, 17(4), 41501. https://doi.org/10.1007/s11467-021-1141-2

Wie, C.-R. (2019). Simpler quantum counting. Quantum Information and Computation, 19(11–12), 967–983. https://doi.org/10.26421/QIC19.11-12-5

Wiebe, N., & Granade, C. (2016). Efficient Bayesian phase estimation. Physical Review Letters, 117(1), 010503. https://doi.org/10.1103/PhysRevLett.117.010503

Wong, T. G. (2018). Grover search with lackadaisical quantum walks. Journal of Physics A: Mathematical and Theoretical, 48(43), 435304. https://doi.org/10.1088/1751-8113/48/43/435304

Xu, L. D., Lu, Y., & Li, L. (2021). Embedding blockchain technology into IoT for security: A survey. IEEE Internet of Things Journal, 8(13), 10452–10473. https://doi.org/10.1109/JIOT.2021.3060508

Yamamoto, K., Duffield, S., Kikuchi, Y., & Muñoz Ramo, D. (2024). Demonstrating Bayesian quantum phase estimation with quantum error detection. Physical Review Research, 6(1), 013221. https://doi.org/10.1103/PhysRevResearch.6.013221

Ye, Z., & Lu, Y. (2022). Quantum science: A review and current research trends. Journal of Management Analytics, 9(3), 383–402. https://doi.org/10.1080/23270012.2022.2089064

Zhang, K., & Korepin, V. E. (2020). Depth optimization of quantum search algorithms beyond Grover's algorithm. Physical Review A, 101(3), 032346. https://doi.org/10.1103/PhysRevA.101.032346