Using Classical Monte Carlo Sampling and Quantum Amplitude Estimation To Calculate Value At Risk Of Portfolio

Hieu Nguyen, Elaina Wang, Tan Nguyen, Kevin Qu, Riki Choi

Feb 2026

1 Value At Risk (VaR) Problem and Model Assumptions

Definition and Motivation

Value at Risk (VaR) is a widely used risk metric that summarizes potential downside exposure of an investment over a fixed time horizon at a specified confidence level. It is designed to capture tail behavior rather than average performance, since adverse but infrequent events can dominate practical risk considerations. In institutional settings, VaR is used to set trading limits, compare risk across portfolios, and inform capital planning. As summarized by **?**, VaR is widely used by investment and commercial banks to quantify potential financial losses and to evaluate whether capital reserves are sufficient to cover those losses. It helps risk managers understand the probability and extent of potential losses, enabling institutions to assess exposure and guide decisions [**?**].

From a statistical perspective, VaR is a tail quantile of a profit-and-loss distribution. Let V_0 denote the portfolio value at the present time and V_T the value at a future horizon T. Defining the loss random variable

$$L \equiv V_0 - V_T \tag{1}$$

the VaR at confidence level $\alpha \in (0, 1)$ is the α-quantile of L,

$$\mathrm{VaR}_\alpha \equiv \inf \{\ell : \mathbb{P}(L \leq \ell) \geq \alpha\} \tag{2}$$

which is equivalent to the tail bound $\mathbb{P}(L > \mathrm{VaR}_\alpha) \leq 1 - \alpha$. In practice, this quantity is rarely available in closed form for realistic portfolios, because the distribution of L may be induced by many correlated risk factors and nonlinear payoffs. For this reason, practitioners typically rely on three principal computational methodologies, namely the historical method, the variance-covariance (parametric) method, and Monte Carlo simulation, each based on different assumptions about the underlying distribution [**?**].

The computational difficulty in VaR estimation arises from the fact that it is a tail quantity. Estimating a quantile associated with a small tail probability requires either strong modeling assumptions or a large number of samples. Monte Carlo methods are especially attractive because they are model-agnostic once a simulator for the underlying risk factors is specified. However, the sampling burden can become large when the confidence level is high, since accurate resolution of rare events demands many realizations. This sample-complexity bottleneck is central to the motivation of the paper and to our problem objective, because the paper frames risk estimation tasks in terms of estimating probabilities and expectations that can be embedded into amplitude-estimation

primitives.

To isolate the convergence behavior in a controlled setting, we model the one-year log-return of an asset as a Gaussian random variable

$$R \sim \mathcal{N}(\mu, \sigma^2), \tag{3}$$

with mean return $\mu = 0.15$ and volatility $\sigma = 0.20$. In this simplified model, the VaR at confidence level $\alpha = 95\%$ is the smallest threshold v such that the probability of observing a return below v is at least the left-tail probability $\gamma = 1 - \alpha = 5\%$. Using a sign convention that reports VaR as a positive loss in return units, we write

$$\mathrm{VaR}\alpha = -\inf$$

For Gaussian returns, this expression reduces to a closed-form formula

$$\mathrm{VaR}\alpha = -[\mu + z_\gamma, \sigma], \tag{5}$$

where z_γ is the γ th quantile of the standard normal distribution. At the 95% confidence level, $z_{0.05} \approx$ -1.64485 , which yields $\mathrm{VaR}_{0.95}^{\mathrm{theory}} \approx 0.17897$.

Although the Gaussian model is intentionally simplistic, it plays an essential methodological role in our study. Real portfolios often exhibit skewness, heavy tails, volatility clustering, and correlations across positions, none of which are captured by a single normal return distribution. Nevertheless, the normal framework provides an analytic benchmark against which we can directly measure estimator error. This enables a clean comparison of convergence behavior between classical Monte Carlo estimators, whose sampling error typically decreases on the order of $1/\sqrt{N}$, and quantum amplitude-estimation-based routines emphasized in the paper, which aim to achieve improved scaling with respect to the number of queries.

2 Classical Monte Carlo Workflow

In the classical workflow, we begin with an analytic reference value for VaR under the parametric normal model, which corresponds to the variance-covariance method. We assume the one-year log-return R is Gaussian, $R \sim \mathcal{N}(\mu, \sigma^2)$. The VaR at confidence level α is determined by the left-tail probability $\gamma = 1 - \alpha$, meaning we seek the threshold v satisfying $\mathbb{P}(R < v) \geq \gamma$. For a Gaussian random variable, the cumulative distribution function can be written in terms of the standard normal CDF Φ, giving

$$\mathbb{P}(R < v) = \Phi\left(\frac{v - \mu}{\sigma}\right) \geq \gamma.$$

Solving for v yields $v = \mu + \sigma z_\gamma$, where $z_\gamma = \Phi^{-1}(\gamma)$ is the γ-quantile of the standard normal distribution.

Since VaR is conventionally reported as a positive loss magnitude rather than a negative return threshold, we define the VaR in return units as the negative of this left-tail threshold,

$$\mathrm{VaR}_\alpha^{\mathrm{theory}} = -\left(\mu + \sigma z_\gamma\right).$$

Substituting $\mu = 0.15, \sigma = 0.20$, and $\gamma = 0.05$ with $z_{0.05} \approx -1.64485$ gives

$$\mathrm{VaR}_{0.95}^{\mathrm{theory}} = -[0.15 + (-1.64485) \times 0.20] \approx 0.1790.$$

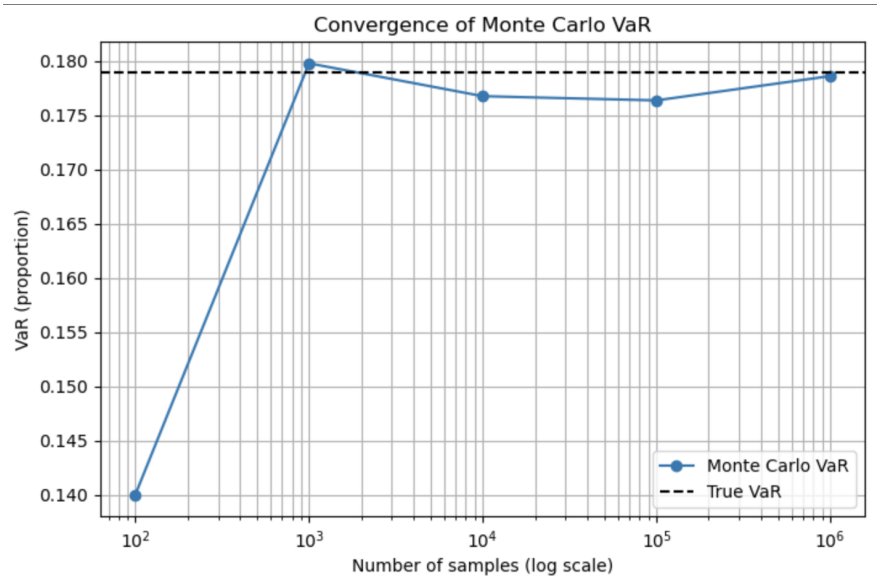


Figure 1: Log-log plot of the estimated VaR value as a function of sample size. Notice that as N becomes large, the estimate stabilizes to the benchmark value.

This closed-form value serves as a ground-truth benchmark throughout the study. It enables us to quantify estimation error directly when we later approximate VaR using finite-sample Monte Carlo simulations and to assess whether the quantum routines reproduce the correct value while exhibiting the improved convergence behavior emphasized by the paper.

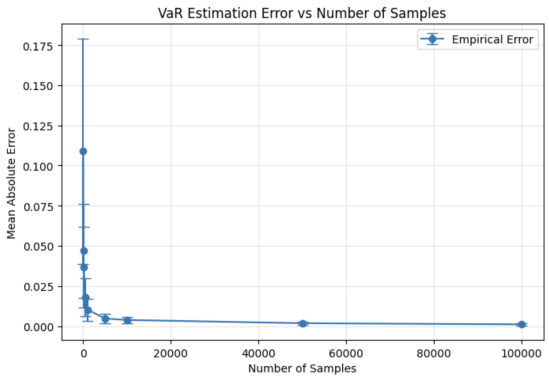
2.1 Monte Carlo VaR Estimate

The Monte Carlo (MC) method approximates the 5% lower quantile of the return distribution by sampling many independent realizations $\{R_i\}_{i=1}^N$ from $\mathcal{N}(\mu, \sigma^2)$, sorting the outcomes and reading off the empirical quantile. Concretely, the estimator for the VaR at confidence level $\alpha = 95\%$ is

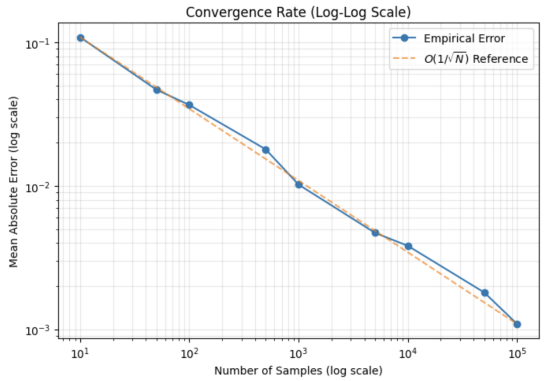
$$\widehat{\mathrm{VaR}}_{0.95}(N) = -\mathrm{quantile}_\gamma\Big(\{R_i\}_{i=1}^N\Big), \tag{6}$$

where $\gamma = 1 - \alpha = 0.05$. The negative sign converts from return to loss. As $N \to \infty$, the estimator converges almost surely to the true VaR by the law of large numbers. In practice, the finite sample error behaves like $O(N^{-1/2})$ because the empirical CDF converges to the true CDF at the parametric $1/\sqrt{N}$ rate. In Figure (1), we illustrate the convergence behavior of the classical Monte Carlo estimator for VaR under our Gaussian return model. The plot shows the estimated VaR as a function of the number of simulated samples N, with the horizontal dashed line indicating the analytic "true" VaR used as a benchmark. At small sample sizes, the Monte Carlo VaR estimate exhibits substantial deviation from the true value. This is an expected feature of quantile estimation in the tail of a distribution. Since VaR at the 95% confidence level is determined by the 5% left tail, a simulation with only 10^2 draws contains, on average, only about five samples in the relevant region. In that regime, the empirical 5% quantile is determined by a handful of extreme observations, so the estimate can shift noticeably depending on whether the simulation happens to include unusually large negative returns or not.

As N increases, the estimate rapidly approaches the dashed reference value and then begins to



(a) Classical absolute mean VaR error on the y-axis vs. number of samples on x-axis



(b) subfigure (a) plotted in log-log scaling.

Figure 2: Classical MC estimation of VaR

stabilize. By $N \sim 10^3$, the Monte Carlo estimate is already close to the benchmark, and for $N \geq 10^4$ the curve fluctuates only mildly around the true VaR. These remaining oscillations are not numerical artifacts, but rather finite-sample effects that persist even at large N. Quantile estimators converge more slowly than mean estimators in the sense that they remain sensitive to local sampling noise near the tail threshold, and small changes in the ordering of extreme samples can shift the empirical quantile. The key takeaway from Figure (1) is therefore twofold. First, classical Monte Carlo does converge reliably to the correct VaR, confirming the validity of the simulation workflow. Second, the convergence is governed by the scarcity of tail events, meaning that achieving high precision in a tail-risk statistic requires a large sample budget. This motivates the later comparison to quantum amplitude-estimation-based routines, which are designed to estimate tail probabilities with improved scaling in the number of oracle queries.

2.2 Convergence Study: Error Scaling

A convergence or error-scaling study is meant to answer a specific quantitative question that is often more important than any single numerical estimate. It asks how quickly an estimator becomes reliable as we increase computational effort. For Monte Carlo methods, the computational effort is the number of independent samples N. Each sample provides one noisy piece of information about the distribution of returns, and the VaR estimator aggregates these samples into an estimate of a tail quantile. Because this aggregation is ultimately driven by sampling randomness, there is an intrinsic tradeoff between accuracy and cost. Error scaling makes that tradeoff explicit by measuring how the typical estimation error decays as a function of N. This is especially relevant for VaR, since it is determined by rare tail events, and rareevent estimation is often the most computationally expensive part of a risk pipeline.

In figure (2), we plot the result of this for the classical VaR estimation. The left panel of the figure displays the mean absolute error of the Monte Carlo VaR estimator as a function of N, together with error bars representing trial-to-trial variability across repeated experiments. The key qualitative feature is the large uncertainty at small sample sizes. When N is small, the 5% tail contains only a handful of points, so the empirical quantile is highly sensitive to which extreme observations happen to appear in the sample. This produces both a large average error and a wide spread across trials. As N increases, the estimator begins to "lock onto" the true VaR, and both

the mean error and the variability decrease substantially. By the time N reaches the regime of 10^4 to 10^5 samples, the error bars are small and the mean absolute error is correspondingly low, indicating that the empirical 5% quantile has concentrated near the analytic benchmark.

The right panel presents the same errors on a log-log scale and overlays a reference curve proportional to $1/\sqrt{N}$, which is the characteristic scaling of classical Monte Carlo sampling. The empirical curve is approximately parallel to this reference line over several decades in N, confirming that the VaR estimator exhibits the expected Monte Carlo convergence rate. This observation is directly tied to the core computational challenge motivating the broader project. If error decreases only like $N^{-1/2}$, then achieving a modest improvement in precision requires a disproportionately large increase in samples. In risk estimation, where VaR must often be computed repeatedly across portfolios, time horizons, and stress scenarios, this scaling becomes a practical bottleneck. Establishing the classical $1/\sqrt{N}$ baseline is therefore essential, because it sets the benchmark against which quantum amplitude-estimation-based routines are evaluated, and it clarifies why a faster scaling in the number of oracle queries would represent a meaningful computational advantage for tail-risk metrics such as VaR.

3 Quantum Amplitude Estimation (QAE)

Quantum Amplitude Estimation (QAE) is a quantum algorithm designed to estimate an unknown probability by encoding that probability into the amplitude of a quantum state and then extracting it using coherent quantum interference. In our VaR setting, the probability we care about is a tail probability of the return distribution. Given a proposed threshold v, the relevant event is that the return falls below that threshold, $R < v$. A convenient way to express this is through an indicator function,

$$\chi_v(R) \equiv 1\{R < v\},$$

so that the tail probability is the expectation of this indicator,

$$\mathbb{P}(R < v) = \mathbb{E}\left[\chi_v(R)\right].$$

Classical Monte Carlo estimates $\mathbb{E}\left[\chi_v(R)\right]$ by sampling $R_1, \ldots, R_M$ and averaging,

$$\widehat{p}_M(v) = \frac{1}{M}\sum_{i=1}^{M} \chi_v\left(R_i\right),$$

which yields the familiar sampling error scaling error $= O(1/\sqrt{M})$. This slow convergence is a central computational bottleneck for VaR, because VaR depends on rare tail events. When $\alpha = 95\%$, the tail probability is $\gamma = 1 - \alpha = 0.05$, meaning only about $0.05M$ samples directly "inform" the relevant region of the distribution.

QAE targets the same mathematical object, the tail probability $p(v) = \mathbb{P}(R < v)$, but it does so by constructing a quantum circuit whose measurement statistics encode $p(v)$ as an amplitude. The algorithm begins by defining a state preparation unitary $\mathcal{A}$ acting on n "data" qubits and one "flag" (ancilla) qubit. The goal is to prepare a state of the form

$$\mathcal{A}|0\rangle^{\otimes n}|0\rangle = \sqrt{1-a}\,|\psi_0\rangle\,|0\rangle + \sqrt{a}\,|\psi_1\rangle\,|1\rangle,$$

where the final qubit is the flag qubit and

$$a = \mathbb{P}(\text{flag} = 1) = \mathbb{P}(R < v).$$

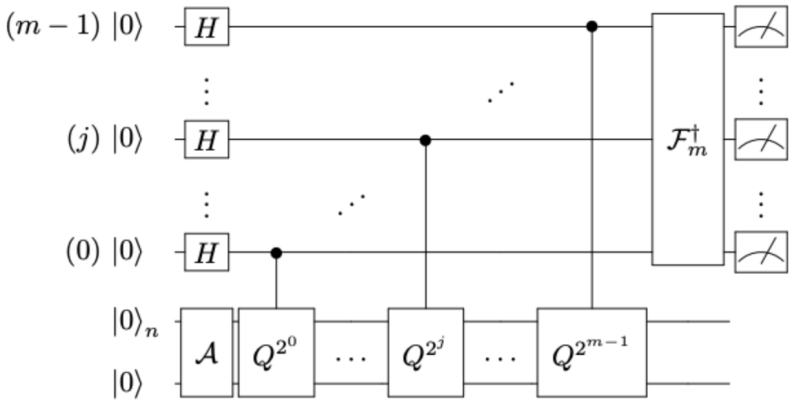


Figure 3: The quantum circuit to implement QAE

In other words, if we were to apply $\mathcal{A}$ and measure only the flag qubit, it would return " 1 " with probability a . The core problem is therefore reduced to estimating a given access to $\mathcal{A}$ and certain related reflections.

The canonical QAE algorithm constructs a Grover-like operator $\mathcal{Q}$ from $\mathcal{A}$ and reflections, typically written as

$$\mathcal{Q} = -\mathcal{A}S_0\mathcal{A}^\dagger S_\chi,$$

where S_0 is a reflection about the all-zero input state and S_χ is a reflection that flips phase depending on whether the flag qubit marks "success." The essential fact is that Q acts as a rotation in the twodimensional subspace spanned by the "good" and "bad" components of the state. One can parameterize the amplitude by an angle θ defined through

$$a = \sin^2(\theta), \quad \theta \in [0, \pi/2].$$

Under this parameterization, repeated applications of Q amplify and phase-rotate the good component in a way that depends on θ. The purpose of QAE is to estimate θ (and hence a) efficiently.

In figure (3), we include a diagram of the canonical QAE circuit. It uses an ancilla register of m qubits initialized in superposition by Hadamard gates. Those ancillas control successive powers of the Grover operator, namely $\mathcal{Q}^{2^0}, \mathcal{Q}^{2^1}, \ldots, \mathcal{Q}^{2^{m-1}}$. Schematically, the controlled-application block prepares a phase-encoded state, after which the inverse quantum Fourier transform $F_m^\dagger$ is applied to the ancilla register. Measuring the ancillas yields a bit string that is postprocessed into an estimate $\hat{\theta}$, and then the amplitude estimate is obtained via

$$\hat{a} = \sin^2(\hat{\theta}).$$

This is where the quadratic improvement originates. Instead of learning a by independent sampling of a Bernoulli random variable, QAE uses coherent applications of Q to encode θ into a global phase and then extracts it with phase estimation. In the ideal noiseless setting, to achieve additive error ε in estimating a, classical Monte Carlo typically requires $M = O\left(1/\varepsilon^2\right)$ samples, while QAE achieves $O(1/\varepsilon)$ scaling in the number of applications of the query operator (often called "oracle queries" or "quantum samples"). This contrast is the scaling statement emphasized in the reference we cite and is the core theoretical motivation for applying amplitude estimation to risk metrics.

In our implementation, we follow the paper's broader framing by treating VaR estimation as a probability estimation problem. For each candidate threshold v, we aim to estimate the tail probability $a = \mathbb{P}(R < v)$ and then search for the value of v that satisfies $a \approx \gamma$, where $\gamma = 1 - \alpha$. The quantum component of that workflow is realized through Classiq's QAE or IQAE routines, which provide a high-level interface for constructing $\mathcal{A}$, forming the corresponding Grover operator $\mathcal{Q}$, and running an amplitude estimation procedure. The canonical QAE circuit shown in the figure explains the underlying mechanism, while our use of Classiq allows us to focus on the modeling choices that matter for VaR, namely how we discretize the return distribution, how we define the threshold event $R < v$, and how we translate the resulting amplitude estimate into a tail probability estimate within the VaR computation pipeline.

4 Iterative Quantum Amplitude Estimation (IQAE)

Iterative Quantum Amplitude Estimation (IQAE) can be viewed as a practical, NISQ-oriented reorganization of the amplitude-estimation objective. The quantity of interest is the tail probability

$$a(v) \equiv \mathbb{P}(R < v),$$

which is the cumulative distribution function evaluated at a threshold v. IQAE assumes access to a statepreparation routine that produces a quantum state whose flag qubit is measured as "1" with probability $a(v)$. The algorithm then refines an estimate of $a(v)$ by repeatedly running comparatively low-depth Grover-style circuits and using classical post-processing to maintain a shrinking confidence interval. At iteration k, IQAE tracks an interval $\left[a_k^{\text{low}}, a_k^{\text{high}}\right]$ that is guaranteed, with a prescribed confidence level, to contain the true amplitude $a(v)$. The circuit depth is controlled through an integer m_k that determines how many Grover applications are used in that round, while the statistical uncertainty is controlled through the number of shots. In effect, IQAE alternates between (i) choosing a measurement setting that can maximally reduce the remaining ambiguity in $a(v)$ and (ii) collecting enough measurement outcomes to tighten the interval.

In the IQAE VaR-estimation notebook, we embed IQAE inside a bisection search to recover the VaR threshold. The VaR condition at confidence level $\alpha = 0.95$ corresponds to the left-tail probability $\gamma = 1 - \alpha = 0.05$. On a discretized grid of return thresholds $\{r_j\}$, we search for the smallest index j such that

$$\mathbb{P}\left(R < r_j\right) \geq \gamma.$$

At each bisection step, IQAE estimates $a\left(r_j\right) = \mathbb{P}\left(R < r_j\right)$ and returns a confidence interval $[\, a^{\text{low}}, a^{\text{high}} \,]$. If $a^{\text{high}} < \gamma$, then the threshold is too far into the left tail and the search moves to larger j. If $a^{\text{low}} > \gamma$, then the threshold is not far enough into the tail and the search moves to smaller j. When the interval straddles γ, the algorithm tightens the estimation tolerance and re-estimates at the same index until the comparison becomes decisive. In our run, the tolerance was reduced from $\varepsilon = 0.05$ down to $\varepsilon_{\min} = 0.002$, and the procedure converged after six bisection iterations to the index $j = 38$ on a $2^7 = 128$ point grid. The corresponding threshold was $r_{38} = -0.175$, giving the VaR estimate

$$\widehat{\text{VaR}}_{0.95}^{\text{IQAE}} = -r_{38} = 0.175.$$

When compared to the analytic Gaussian benchmark $\text{VaR}_{0.95}^{\text{theory}} \approx 0.179$, the absolute discrepancy is approximately 0.004 . This residual error is consistent with the fact that we are working with a

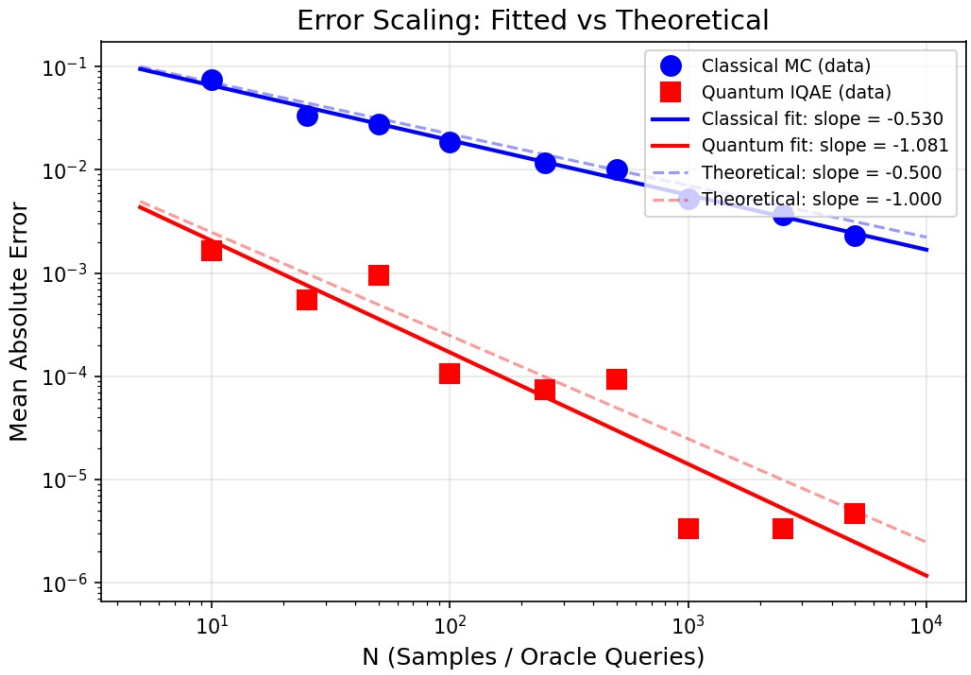


Figure 4: Error scaling comparison between classical Monte Carlo and quantum IQAE for estimating $\Pr(R < r_{38})$. Each point represents the absolute error of the estimator. The quantum errors decay roughly as $O(N^{-1})$, while the classical errors decay as $O(N^{-1/2})$. Data are adapted from our experiments and the error scaling notebook.

finite grid and a finite probability resolution. Even if amplitude estimation were exact, discretization and truncation of the distribution support introduce a baseline approximation error because the true 5th percentile generally falls between grid points.

The scaling-error code and the associated plot quantify the computational implication of this approach. Rather than focusing only on the final VaR threshold, we fix the target index $j = 38$ and compare how accurately classical Monte Carlo and IQAE estimate the same tail probability $a\left(r_{38}\right) = \mathbb{P}\left(R < r_{38}\right)$ as a function of computational budget. The classical baseline computes the empirical CDF at r_{38} using N independent samples, which produces an error that typically scales like

$$|\widehat{a}_N - a| = O\left(N^{-1/2}\right).$$

The IQAE routine estimates the same probability using a budget that is naturally measured in oracle queries, meaning the total number of Grover-style applications aggregated across iterations. In the ideal regime, amplitude-estimation-based methods target scaling of the form up to logarithmic and constant factors that depend on confidence parameters and shot noise.

The fitted log-log plot in figure (4) supports this theoretical picture in our implementation. The classical Monte Carlo data (blue) is well-described by a slope near -0.53 , which aligns closely with the theoretical $-1/2$ exponent. The IQAE data (red) is well-described by a slope near -1.081 ,

Method	Fitted exponent β	Theoretical exponent	R^2 of fit
Classical MC	-0.5304	-0.5	0.9896
Quantum IQAE	-1.0813	-1.0	0.8867

Table 1: Scaling exponents from log–log regression of error vs sample size. A more negative exponent implies faster convergence. The quantum estimator exhibits a slope close to -1, twice as steep as the classical Monte Carlo.

which is close to the ideal -1 exponent expected from amplitude estimation. Interpreted operationally, this difference is the central message for the VaR problem. Under classical sampling, reducing the typical error by a factor of two requires roughly a fourfold increase in samples because $N^{-1/2}$ scaling implies $(2\varepsilon)^{-2}/\varepsilon^{-2} = 4$. Under IQAE-like scaling, a comparable factor-of-two error reduction can be obtained with roughly a twofold increase in queries, since N^{-1} scaling implies $(2\varepsilon)^{-1}/\varepsilon^{-1} = 2$. Since VaR estimation is fundamentally a tail-probability computation repeated across many thresholds, portfolios, and scenarios, the observed improvement in scaling directly addresses the core computational bottleneck that motivated the project and the comparison emphasized in the paper.

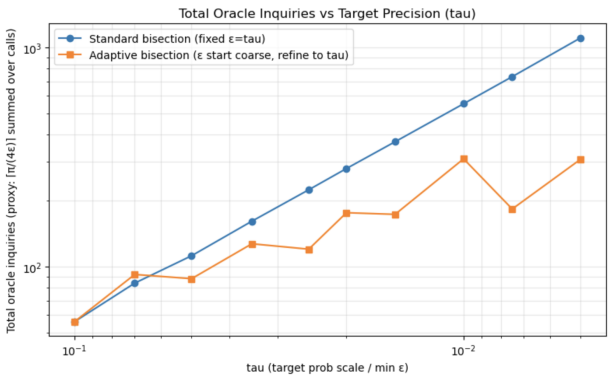
5 Extensions

In the extensions of our work, we focused on two practical questions that arise immediately once the baseline Gaussian VaR benchmark is in place. The first question is algorithmic and concerns how we couple amplitude estimation to the outer bisection search used to locate the VaR threshold. The second question is modeling-driven and concerns what changes when the return distribution is no longer light-tailed. Both extensions are motivated by the same principle. VaR is fundamentally a tail statistic, so both the search strategy and the distributional tail behavior can dominate the overall computational cost and the stability of the resulting estimate.

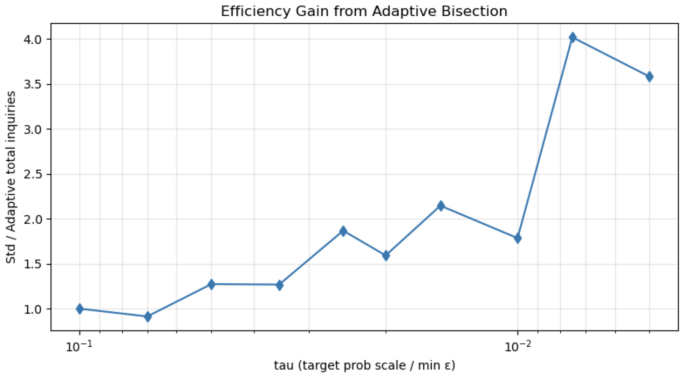
5.1 Adaptive Bisection Search For IQAE

In our baseline IQAE VaR routine, we locate the VaR threshold by bisection over a discretized grid $\{r_j\}$, where at each step we must decide whether the tail probability $a(r_j) = \mathbb{P}(R < r_j)$ lies below or above the target $\gamma = 1 - \alpha$. This decision is made using an IQAE confidence interval $[\, a^{\text{low}}, a^{\text{high}} \,]$. A standard approach is to enforce a fixed additive tolerance ε at every bisection step, meaning that each step pays essentially the same "high-precision" cost even though early bisection steps are far from the final threshold and do not require such fine probability resolution. The adaptive bisection idea in the Adaptive_IQAE notebook formalizes a simple optimization. Early in the bisection, we only need enough accuracy to determine the sign of $a(r_j) - \gamma$. High precision is only necessary once the search has narrowed to a small interval of candidate indices.

Operationally, we parameterize the target probability scale by a "final precision" parameter τ, which plays the role of the minimum tolerated amplitude-estimation error near the decision boundary. In the adaptive scheme, we begin with a coarser tolerance $\varepsilon_{\text{start}}$ and only refine toward τ when the IQAE confidence interval overlaps γ, indicating that we are near the threshold where classification becomes ambiguous. The first adaptive plot, "Total Oracle Inquiries vs Target Precision (τ)," compares the total query budget for two strategies. The standard approach (fixed $\varepsilon = \boldsymbol{\tau}$ throughout) grows steadily and predictably as $\boldsymbol{\tau}$ decreases, which is consistent with the fact that IQAE



(a) Classical absolute mean VaR error on the y-axis vs. number of samples on x-axis



(b) subfigure (a) plotted in log-log scaling.

Figure 5: Classical MC estimation of VaR

query complexity scales roughly like $O(1/\varepsilon)$ for a target additive error ε. The adaptive approach uses substantially fewer total oracle inquiries across most τ values because it avoids spending high precision on bisection steps that are not yet informative. The second adaptive plot, "Efficiency Gain from Adaptive Bisection," reports the ratio of query budgets (standard divided by adaptive). The gains are modest at very coarse precision, then become meaningfully larger as τ becomes smaller. This trend matches the underlying logic. When the final target precision is stringent, a fixed- ε strategy overpays repeatedly, while an adaptive strategy concentrates the expensive, high-precision IQAE calls only near the end of the search where they materially affect the decision.

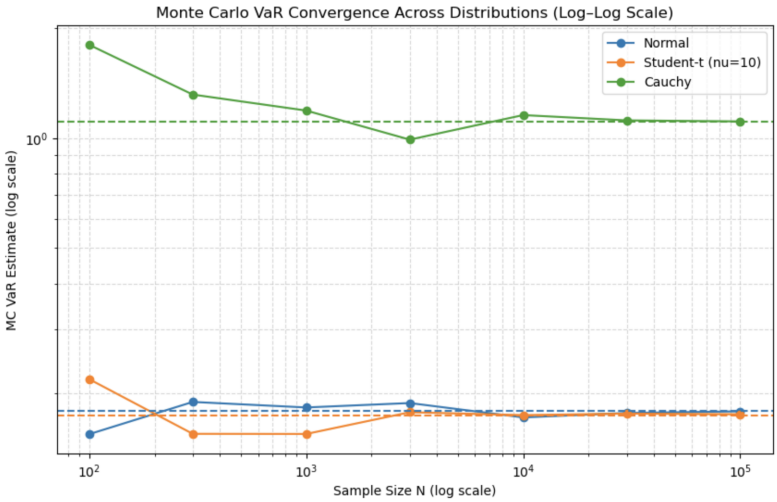
5.2 Heavy-tailed Models

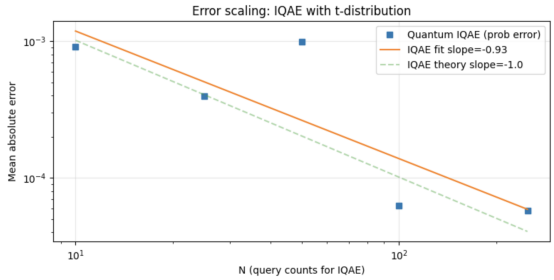
A second extension addresses a modeling limitation of the Gaussian benchmark. Real financial returns can exhibit heavy tails, skewness, and occasional extreme events. To probe how tail behavior affects the stability of VaR estimation, we repeated the classical Monte Carlo workflow under heavier-tailed distributions. Concretely, we considered a Student- t distribution with $\nu = 10$ degrees of freedom, which has finite variance but heavier tails than a Gaussian, and a Cauchy distribution, which is so heavy-tailed that even its mean and variance are not finite. In all cases, the VaR remains a tail quantile. For a given confidence level α and tail probability $\gamma = 1 - \alpha$, the population VaR in return units can still be defined as the negative left-tail quantile,

$$\mathrm{VaR}_\alpha = -q_\gamma(R), \quad q_\gamma(R) = \inf\{v : \mathbb{P}(R < v) \geq \gamma\}.$$

However, the behavior of the finite-sample estimator $\hat{q}_\gamma$ depends strongly on how often extreme outcomes appear and how variable those extremes are.

Figure (6) makes this sensitivity visible. For the normal and Student- $t(\nu = 10)$ cases, the Monte Carlo VaR estimates settle toward their respective reference values as N increases, although the Student- t curve typically exhibits larger finite sample fluctuations at smaller N. This is consistent with the intuition that heavier tails produce more variability in the order statistics that determine the empirical quantile. The Cauchy case is qualitatively different. Because the distribution places substantial probability mass far into the tails, finite samples can contain extremely large outliers that strongly perturb the empirical tail quantile, and the estimator can appear unstable even when N is increased by orders of magnitude. The plot reflects this behavior through the much larger





variability of the Cauchy-based VaR estimates and their weaker visual "concentration" compared to the light-tailed models. This is an important practical lesson. Even before comparing classical and quantum approaches, the underlying tail model can determine whether VaR estimation is numerically stable at a given sample budget.

We then repeated the quantum probability-estimation study under a heavier-tailed model using IQAE. The key object remains the same. For a fixed threshold v, IQAE estimates the tail probability

$$a(v) = \mathbb{P}(R < v),$$

and the VaR threshold is recovered by searching for v such that $a(v) \approx \gamma$. The "Error scaling: IQAE with t-distribution" plot reports the mean absolute error of the IQAE probability estimate as a function of query count N, along with a fitted slope and the theoretical reference slope. The fitted exponent in the figure is approximately -0.93 , which is close to the ideal -1 scaling expected from amplitude-estimation-based routines. The small deviation from -1 is unsurprising in a finite experiment because constant factors, discretization error from representing a continuous distribution on a finite grid, and shot noise can all perturb the observed slope. The main takeaway is that the improved scaling trend persists when we move from Gaussian to a heavier-tailed Student- t model, even though the underlying distribution is more challenging from a tail-estimation standpoint.

Taken together, these extensions strengthen the practical narrative of the report. Adaptive bisection reduces the total query cost by aligning estimation precision with decision difficulty, which matters because IQAE's cost grows as tighter error tolerances are demanded. Heavy-tailed models

expose the fragility of naive tail estimation and show why variance-covariance-style assumptions can be misleading in realistic settings. Finally, the IQAE scaling results under a Student- t distribution suggest that the core quantum advantage being tested in this project is not merely an artifact of the Gaussian benchmark, but is tied more fundamentally to the amplitude-estimation mechanism itself.