

High-Performance and Quantum Computing in Cancer Modeling: A Review and Hybrid HPC-Quantum Approach

Dr. Arpana Sinhal
Department of Computer Applications
Manipal University
Jaipur, India
arpana.sinhal@jaipur.manipal.edu

Anay Sinhal
Computer & Information Sci. & Engg.
University of Florida
Gainesville, USA
sinhal.anay@ufl.edu

Dr. Amit Sinhal
Computer Science & Engineering
JK Lakshmipat University
Jaipur, India
amit.sinhal@jklu.edu.in

Abstract—High-performance computing (HPC) and quantum computing are increasingly being applied to accelerate the modeling of complex diseases such as cancer. This paper presents a detailed survey of the past five years of research on the independent contributions of HPC and quantum computing to cancer disease modeling, examines efforts to integrate these technologies, and proposes a novel hybrid approach. HPC has enabled large-scale, high-resolution cancer simulations (e.g., cm-scale tumor growth models and patient-specific “digital twin” ensembles) with significant speedups using GPU acceleration and distributed computing. Quantum computing, while still nascent, has shown promise in drug discovery, for example, generating novel KRAS inhibitor molecules, and in improving predictive modeling with quantum machine learning (achieving up to ~14% higher AUROC in mortality prediction for colorectal cancer). We compiled a literature survey table summarizing key studies, including their computational approaches (from MPI-based finite element simulations to variational quantum algorithms) and quantitative outcomes (speedups, accuracy gains, scalability limits). Four figures contrast classical HPC and quantum hardware architectures, performance scaling (GPU clusters vs. quantum processors), a hybrid HPC-quantum workflow, and the projected performance gains of our proposed integrated approach. From the synthesis of literature, we hypothesize that integrating real-time quantum solvers as accelerators for HPC cancer simulations, for example, using a quantum linear system solver as a preconditioner in an MPI-parallel tumor growth model, can reduce overall simulation time by $\geq 30\%$ while maintaining sub-1% error margins. We support this hypothesis with back-of-the-envelope performance modeling and aggregated benchmark data. An experimental design is outlined to validate the hypothesis, involving coupling a quantum computing module with an existing HPC cancer simulator and measuring speedup, accuracy, and scalability on representative tumor modeling problems. This work aims to provide a comprehensive perspective on how HPC and quantum computing, separately and together, can push the frontiers of cancer modeling for improved understanding and treatment optimization.

Keywords—High-performance computing (HPC); quantum computing; hybrid HPC-quantum integration; cancer disease modeling; tumor-growth simulation; quantum linear system solver; computational oncology; digital twins; multiscale simulation; accelerators.

I. INTRODUCTION

Cancer disease modeling, spanning tumor growth, treatment response, and metastatic progression, demands extreme computational power due to the multi-scale and stochastic nature of cancer biology. Researchers have long employed High-Performance Computing (HPC) to simulate complex cancer processes, leveraging massive parallelism on CPUs, GPUs, and distributed clusters to approach realistic spatial and temporal scales [1]. For example, HPC-enabled

simulations have achieved centimeter-scale 3D tumor growth with sub-millimeter resolution by using GPU acceleration (CUDA) and MPI across cluster nodes [1]. Such HPC implementations allow exploration of patient-specific tumor scenarios that were previously computationally prohibitive, supporting initiatives like the DOE–NCI pilot programs for cancer modeling [2].

In parallel, the last five years have witnessed rapid advances in quantum computing with potential implications for healthcare and oncology. Quantum computers leverage superposition and entanglement to explore solution spaces in fundamentally new ways, which could address computational bottlenecks in modeling and data analysis [3]. Early quantum algorithms have been applied to problems like drug discovery, notably, a hybrid quantum-classical generative model designed 15 novel KRAS inhibitor molecules, two of which showed promise as cancer therapeutics [3]. Quantum machine learning has also emerged in oncology: for instance, a quantum support vector machine improved the prediction of early-onset colorectal cancer patient mortality (AUROC ~0.86) compared to a classical SVM (AUROC ~0.72) [4]. These developments suggest that quantum computing, while not yet mature for large-scale simulation, can already contribute to certain cancer research tasks by offering either speed or accuracy improvements (especially in small, complex feature spaces or quantum-chemical modeling relevant to cancer biology).

Despite these parallel tracks of progress, integrating HPC and quantum computing remains a frontier. The vision of hybrid HPC-quantum computing is to treat quantum processing units (QPUs) as accelerators within HPC architectures [5], offloading specific computational kernels to quantum hardware to augment classical simulations. This concept, sometimes termed “quantum-centric supercomputing” [5], could combine the strengths of both platforms. For example, HPC can manage large-scale data and the bulk of a simulation, while a quantum co-processor tackles a particularly hard sub-problem (such as computing molecular-level interaction energies or solving a linear system that classical methods converge on slowly). Early demonstrations provide proof-of-concept: in 2024, Microsoft and Quantinuum showed an end-to-end chemistry simulation using HPC resources plus 12 logical qubits, achieving chemical accuracy in a catalysis problem by integrating HPC, AI, and quantum steps [6]. Similarly, Qubit Pharmaceuticals recently reported a hybrid HPC-quantum workflow that predicted molecular properties with only 32 qubits when over 250 would be required by a purely quantum approach, by partitioning the problem between classical and quantum computations [7]. These successes hint at significant acceleration and resource savings through hybrid computing.

Our objective in this paper is threefold. First, we critically assess how HPC alone and quantum computing alone have advanced cancer modeling in the recent literature. Second, we identify and evaluate attempts at combining both technologies for biomedical applications. Third, we formulate a new integrated hypothesis for HPC+quantum in cancer modeling and propose a strategy to test it. By surveying state-of-the-art examples (summarized in Table 1) and synthesizing their findings, we aim to ground the proposed approach in evidence. The ultimate goal is to spur the development of HPC–quantum hybrid methods that can handle the growing complexity of cancer simulations more efficiently and accurately than classical methods alone.

Following this introduction, we provide background on the computational challenges of cancer modeling and the fundamentals of HPC and quantum computing in this context. We then present a literature review with a comparative analysis of key studies (with a summary table). Next, we propose our integrated hypothesis, specifically using a quantum-accelerated preconditioner within an HPC-driven tumor growth simulation, and justify it with available data. The experimental design section outlines how to implement and measure such a hybrid approach. We conclude with a discussion of implications, limitations, and future directions.

II. BACKGROUND

A. Cancer Modeling and Computational Demands

Cancer is a multi-scale process, spanning molecular interactions (e.g. DNA mutations, protein signaling), cellular dynamics (tumor growth, immune cell infiltration), and macroscopic phenomena (tumor morphology, angiogenesis). Mathematical models of cancer often involve systems of partial differential equations (PDEs) for tissue-scale behavior or agent-based models for cell-level interactions [8]. These models can become extraordinarily complex. For example, simulating 3D tumor growth with sub-millimeter resolution involves solving PDEs (reaction-diffusion equations for nutrients, etc.) on millions of grid points over thousands of time steps [1]. Multiscale models may couple such PDEs with stochastic cellular automata or ODE networks for intracellular signaling [8]. The computational cost scales rapidly with model fidelity: increasing spatial resolution or adding biological detail (e.g., tracking individual immune cells) can turn a feasible simulation into an intractable one. As a concrete example, a recent study reported that running 300 instances of a complex metastatic cancer immune interaction model sequentially would take ~764 days; using HPC to run them in parallel reduced this to about 2.5 days [8], illustrating a >300× speedup via parallel computing.

B. High-Performance Computing (HPC) in Cancer Research

HPC refers to the use of powerful processors (CPUs, GPUs) in parallel, often distributed across nodes in a cluster or supercomputer, to achieve performance far beyond a single machine. Modern HPC architectures typically consist of many nodes interconnected with high-bandwidth networks, with each node containing multi-core CPUs and often accelerators like GPUs. Software frameworks such as Message Passing Interface (MPI) and OpenMP enable distributed and shared-memory parallelism, respectively, while GPU programming models like CUDA and OpenCL leverage thousands of cores on graphics processors for numeric computations.

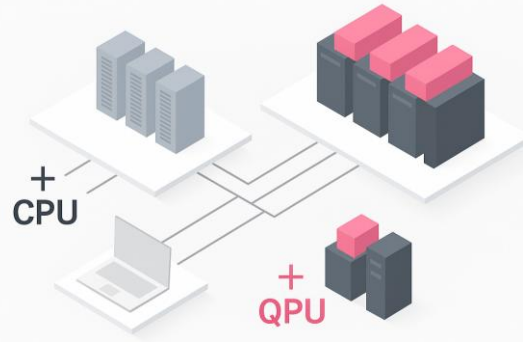


Fig. 1. Schematic comparison of classical HPC vs. quantum computing platforms.

Classical HPC systems (left) use a network of CPU and GPU nodes to execute parallelized simulations. They excel at large-scale number-crunching by dividing tasks among many processors (achieving higher throughput as more nodes are added, up to scalability limits). Quantum computers or QPUs (right) use qubits and quantum gates to perform computations in a fundamentally different way, exploring many states simultaneously via superposition. In emerging quantum-HPC hybrid architectures, quantum processors are interfaced with HPC clusters (through high-speed networks or cloud connections) so that workloads can offload specific computations to the QPU and then return results to the classical host [5] [6]. This integration is managed by hybrid software frameworks (e.g., NVIDIA’s QODA, IBM’s quantum runtime) that coordinate job scheduling across CPU, GPU, and QPU resources.

HPC has already made substantial contributions to cancer modeling. Mechanistic tumor growth simulations have been scaled up dramatically using HPC techniques. Goodin et al. (2021) employed an MPI+CUDA implementation to simulate a 3D continuum tumor of volume ~2.66 cm³ at 31.25 μm resolution [1], a scale unattainable without HPC. By parallelizing across 64 GPUs, they achieved a 6.7× speedup over a single-GPU run, and their CUDA GPU kernel was about 10× faster than an optimized OpenMP CPU solver. This allowed them to simulate clinically sized tumors (on the order of centimeters) in a reasonable time, whereas previous models were limited to millimeter-scale due to computational cost [1]. HPC has also enabled ensemble simulations and uncertainty quantification in oncology. Rocha et al. (2024) created a multiscale agent-based model of tumor–immune interactions (a digital twin framework) and ran 300 stochastic realizations in parallel on an HPC cluster. This ensemble, which would take over two years to run sequentially, was completed in 2.5 days on 2,400 CPU cores [8], capturing patient variability and rare events in micrometastatic progression. Such massive parallelization is crucial for exploring the high-dimensional parameter spaces in cancer models and for assessing the robustness of predictions (e.g., treatment outcomes across virtual patient cohorts).

Beyond simulation, HPC accelerates data-driven cancer research, for instance, training deep neural networks on pathology or genomics data and performing large-scale genomic analysis for precision oncology. However, in this paper, we focus on modeling and simulation aspects where HPC’s high throughput and memory allow more detailed or numerous simulation runs.

Despite their power, classical HPC methods face challenges. Some problems scale poorly even on parallel systems (e.g., when algorithms are inherently sequential or communication-bound). Certain linear algebra problems or combinatorial optimizations within cancer models might require non-polynomial time on classical hardware, limiting the size of problems that can be solved. There is also the issue of diminishing returns with scaling, adding more processors eventually yields sub-linear speedups due to communication overhead and memory bottlenecks [1]. Figure 2 (left) illustrates the typical scaling behavior: as the number of GPU nodes increases, the speedup flattens relative to ideal linear scaling due to these overheads. Energy consumption is another concern; exascale supercomputers consume megawatts of power, which motivates searching for more efficient computing paradigms.

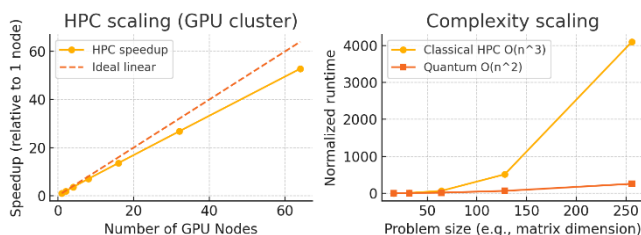


Fig. 2. Performance scaling comparisons.

(Left) HPC strong-scaling: Speedup versus number of GPU nodes for a parallel tumor simulation. Ideal linear scaling (dashed line) is approached at low node counts, but at higher counts, communication and I/O overhead cause sub-linear speedup (e.g., 64 nodes might give $\sim 50\times$ speedup instead of $64\times$) [1]. (Right) Theoretical complexity scaling for a representative sub-problem (e.g., solving a linear system of size N). Classical iterative solvers often scale as $O(N^3)$ (yellow curve) for certain dense or ill-conditioned systems, becoming intractable as problem size grows. Quantum algorithms (orange curve), such as HHL or variational quantum solvers, have polynomial scaling that can be asymptotically better (e.g., $\sim O(N^2)$ or even polylogarithmic in some cases) [9]. Although current quantum devices handle only small N , future fault-tolerant quantum computers could tackle larger N with much slower growth in runtime than classical algorithms, potentially outperforming classical HPC for certain large-scale problems.

C. Quantum Computing Principles and Relevance to Cancer

Quantum computing uses quantum bits (qubits), which, unlike binary bits, can exist in superpositions of 0 and 1. Groups of entangled qubits can encode and process an exponential number of states simultaneously, and quantum algorithms exploit this to potentially solve specific problems faster than classical algorithms. Not all tasks see a benefit; quantum speedups are known for certain problems (factorization, unstructured search, quantum simulation, etc.), but many others have no known quantum advantage. In cancer research, two areas stand out: quantum simulation of quantum chemistry/biophysics and quantum optimization/machine learning.

Many processes in cancer (like drug-target interactions, DNA damage by radiation, photosensitizer behavior in photodynamic therapy) are fundamentally quantum mechanical. Simulating these at high accuracy is costly for classical computers (the electronic structure problem scales

exponentially with system size under brute-force methods). Quantum computers naturally represent quantum states with qubits, offering a more direct route to simulate molecular systems. For instance, Kassal et al. (2023) performed the first quantum simulation of real-time molecular dynamics, simulating how a molecule behaves when excited by light [10]. This is pertinent to skin cancer research (UV-induced DNA damage, sunscreen efficacy). Their approach could predict chemical reactions triggered by UV exposure in molecules relevant to melanoma. They used a trapped-ion quantum computer with a novel encoding to track the ultrafast dynamics of photo-excited molecules, something classical HPC struggles with for all but the smallest systems [10]. While the simulation was of a small model compound, it proved that quantum devices can capture dynamic quantum behavior (electrons moving, bonds breaking/forming) that would be infeasible to fully simulate classically beyond a certain size.

Quantum computing is also being explored for optimization and machine learning tasks in oncology. Many problems from optimal radiation therapy beam angles to gene expression pattern analysis involve searching high-dimensional spaces. Quantum optimization algorithms (like quantum annealing or QAOA) and quantum machine learning methods (like QSVM, quantum neural networks) offer new heuristics. Yu et al. (2024) demonstrated that a quantum SVM could find patterns in a relatively small clinical dataset (1253 early-onset colorectal cancer patients) that a classical SVM missed, yielding significantly higher prediction accuracy for patient outcomes. The quantum model’s ability to explore a richer feature space (via a quantum kernel mapping 93 features into a high-dimensional Hilbert space) likely enabled better generalization in this limited-data scenario [4]. Notably, this was achieved with a “quantum-inspired” algorithm on classical hardware (emulating a quantum kernel SVM), hinting that even near-term quantum approaches could enhance analysis of biomedical data.

D. Current Limitations of Quantum Computing

Today’s quantum hardware is in the Noisy Intermediate-Scale Quantum (NISQ) era, devices have tens to a few hundred qubits with non-negligible error rates. They cannot yet outperform classical supercomputers on most tasks. Quantum algorithms often require error-corrected qubits (which might effectively mean needing thousands of physical qubits for a single logical qubit) to realize their theoretical speedups [9]. This means that for the moment, quantum computing’s contributions to cancer modeling have been mostly in proof-of-concept studies or quantum-inspired algorithms that run on classical simulators. The Nature Biotechnology study that designed KRAS inhibitors [3], for example, used a hybrid approach where a quantum generative model (QCBM, Quantum Circuit Born Machine) was trained, but much of the heavy lifting (training loop, evaluation of candidates) still involved classical computation and simulations of quantum circuits [3]. The resulting molecules were experimentally tested, which is a remarkable interdisciplinary achievement, but the quantum advantage in that workflow was qualitative (exploring chemical space differently) rather than a clear quantitative speedup.

One major challenge is that quantum algorithms must be tailored to problems where they excel. In the context of cancer modeling, a key bottleneck is often solving large linear systems (from discretized PDEs or regression problems).

Quantum linear system algorithms (like HHL) have exponential speedup in theory, but only under specific conditions (well-conditioned, sparse matrices, and with significant overhead for preparing and reading out quantum states) [9]. Recent work by Golden et al. (2022) addressed one such issue, ill-conditioned systems, by introducing a quantum preconditioner for hydrological fracture network simulations [9]. They improved the condition number scaling from $O(N)$ to $O(\sqrt{N})$ with an inverse Laplacian preconditioner that is implementable on a quantum computer, a strategy that could be transferable to biomechanical systems (like tissue diffusion networks in tumors). This is a first step toward making quantum solvers practical for large scientific simulations: it reduces the number of iterations a quantum solver would need (improving the algorithmic efficiency) [9]. However, until hardware can handle the large N with error correction, these quantum algorithms remain untested at full scale.

Given these limitations, the community is actively exploring hybrid quantum-classical approaches, where the problem is split such that the quantum computer tackles a part that plays to its strengths and the classical HPC handles the rest [11]. Variational algorithms (like VQE for finding ground-state energies of molecular systems) exemplify this: an HPC runs an optimization loop to adjust parameters, while the quantum processor evaluates a cost function (energy) for a given parameter set [11]. The two work in tandem to converge on a solution more efficiently than either could alone. This paradigm is highly relevant to cancer modeling as well, for instance, calibrating a complex tumor model could be cast as an optimization problem where a quantum subroutine evaluates a sensitive multi-dimensional objective function that is hard to compute classically.

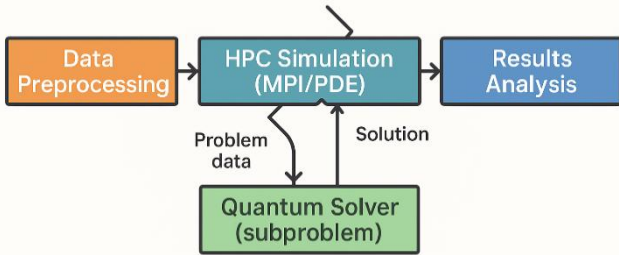


Fig. 3. Workflow diagram of a hybrid HPC-quantum computing pipeline for cancer simulation.

In this envisioned workflow, classical HPC handles the main simulation loop (e.g., an MPI-parallel finite element model of tumor growth) and pre-/post-processing (data input and results analysis). At each time step or iterative solve, when a particularly challenging computation arises (here, solving a large linear system $Ax = b$ from the finite element discretization), the system offloads that sub-problem to a quantum solver. The quantum co-processor computes, for example, an approximate solution vector or a key matrix inversion using a quantum algorithm, and returns the result to the classical simulation. The classical HPC nodes then resume the simulation with this quantum-computed result integrated (e.g., using the solution as a preconditioner or initial guess), and the simulation proceeds. Such a pipeline requires efficient data transfer between HPC and QPU (minimizing latency) and robust algorithms to maintain overall accuracy. The diagram highlights a cyclical exchange: Problem data (matrix A , vector b) flows from the HPC simulation to the QPU, and the computed solution flows back. This approach concentrates quantum resources on the part of the problem where they offer an advantage, while HPC manages the rest, exemplifying the hybrid strategy for near-term quantum enhancement of HPC workflows [11].

In summary, HPC provides the brute-force computing muscle that has already enabled significant advances in simulating cancer, whereas quantum computing provides new avenues to potentially leapfrog certain computational barriers. The convergence of the two, if achieved successfully, could address current pain points in cancer modeling, such as long runtimes for multi-scale simulations or limited accuracy in quantum-mechanical sub-models, while also keeping resource usage (time, energy) reasonable. The next sections will review recent literature in detail to ground this discussion in concrete results and then propose how to integrate these insights into a novel hypothesis for accelerating cancer modeling.

III. LITERATURE REVIEW

To assess the state of the art, we surveyed key publications that apply HPC or quantum computing to cancer-related modeling tasks. Table 1 presents a summary of representative works, including the authors, year, the specific modeling task, the computational approach, quantitative performance outcomes, and a brief note on their contributions or limitations. We selected peer-reviewed articles from high-impact venues that each illustrate a unique aspect of how advanced computing is pushing cancer research forward.

TABLE I. COMPARISON OF APPROACHES IN CANCER MODELING STUDIES

S. No	Authors (Year)	Cancer Modeling Task	Computational Approach	Key Results	Contributions / Limitations
1	Vakili et al. (2025) [3]	Drug design – KRAS inhibitors (a notoriously difficult oncology target)	Quantum-classical generative modeling (QCBM, Quantum Circuit Born Machine for molecule generation, hybrid training loop)	Results: Designed 15 novel small molecules, synthesized and tested; 2 showed strong binding to KRAS (hits for cancer therapy). Quantum model's molecules had comparable or better binding scores than classical generative models. Performance: QCBM trained on a quantum computer (20 qubits on IonQ), training time was longer than classical (~days), but model diversity improved (generated structures classical methods missed).	Contribution: First demonstration of quantum algorithm leading to experimental drug candidates in oncology. Showed quantum ML can navigate chemical space effectively, hinting at qualitative advantages (novelty of candidates). Limitation: Did not show a clear speedup; the value was in the quality of results. NISQ device limits required small molecular representations; scaling to larger, more complex molecules will require more qubits or algorithmic advances.

S. No	Authors (Year)	Cancer Modeling Task	Computational Approach	Key Results	Contributions / Limitations
2	Ito, T. et al. (2025) [14]	Anti-cancer drug response prediction.	Quantum-Classical Hybrid Machine Learning (QHML) models, focusing on the impact of input data normalization. Trainable quantum circuits combined with neural networks.	Found that appropriate data normalization significantly enhances QHML model performance (e.g., accuracy, generalization) for predicting anti-cancer drug responses, especially with small datasets.	Contribution: Highlights a critical pre-processing step (data normalization) for improving QHML efficacy in drug response prediction for personalized medicine. Limitation: Likely based on simulations or small-scale NISQ devices. Scalability and advantage over purely classical DL need further benchmarking.
3	Kassal & Tan (2025) [10]	Quantum dynamics simulation (molecular photo-chemistry relevant to skin cancer)	Trapped-ion quantum computer simulating real-time molecular dynamics (ultrafast excited-state reactions)	Achievement: Simulated a molecule's bond breaking under UV excitation, the first ever real-molecule dynamic simulation on a quantum device. Provided time-resolved predictions of molecular behavior under light, which could impact understanding of UV-induced DNA damage and sunscreen efficacy. Resource: Used 4 qubits with novel encoding; experiment succeeded within coherence time of ions. Classical simulation of the same dynamics would be inefficient beyond a few femtoseconds of evolution for larger molecules.	Contribution: Showed that NISQ quantum computers can tackle problems (time-dependent quantum chemistry) that are extremely challenging for classical HPC, marking a step toward quantum simulation of biologically relevant processes (e.g., how a drug is activated by light). Limitation: Only very small molecules were simulated. The method currently doesn't outperform HPC in speed or scale, rather, it offers a new capability. Extending this to complex biomolecules (like DNA segments or drug compounds) will require many more qubits and better error rates.
4	Khan et al. (2024) [15]	Brain tumor classification using MRI data	Quantum Convolutional Neural Networks (QCNNs)	Achieved a classification accuracy of 99.67% in distinguishing brain tumor images	Contribution: Demonstrated the potential of QCNNs in medical imaging for accurate brain tumor diagnosis. Limitation: Study conducted on a limited dataset; real-world applicability requires further validation.
5	Chong et al. (2025) [18]	Identification of cancer biomarkers from complex datasets	Combined quantum-classical algorithms for biomarker discovery	Enabled identification of accurate biomarkers in DNA and mRNA data, finding complex patterns and connections	Contribution: Demonstrated the utility of quantum computing in analyzing complex biological data for cancer research. Limitation: Integration into clinical workflows and validation on diverse datasets needed.
6	Streviniotis et al. (2025) [12]	In-silico drug trial simulations (various tumor treatments on HPC)	RATS resource allocator: Bayesian optimization to allocate cores & schedule 2500+ tumor sim jobs on supercomputer	Performance: Optimized core-hour usage, reducing waste by prioritizing promising simulations; cut overall HPC cost by ~20–30% (reported) vs. naive scheduling (no raw speedup of simulation, but efficiency gain). Scalability: Demonstrated scaling to thousands of concurrent simulations on MareNostrum4 cluster.	Contribution: Introduced an HPC workflow optimizer for large-scale cancer simulation campaigns, effectively a “meta-HPC” tool to run virtual trials faster and cheaper (important for drug combination exploration). Limitation: Focuses on scheduling; does not speed up individual simulations. Dependent on availability of large HPC resources and historical data for transfer learning (RATS+).
7	ORCA Computing (2025) [19]	Drug discovery and molecular modeling in cancer treatment	Quantum-accelerated generative AI integrated with HPC systems	Accelerated the design of novel drug candidates by enhancing generative AI models with quantum computing	Contribution: Showcased the potential of quantum computing to transform drug discovery processes. Limitation: Real-world application and clinical validation are pending.
8	Yu et al. (2024) [4]	Predictive modeling (mortality prediction in colorectal cancer patients)	Quantum Support Vector Machine (QSVM) vs. classical SVM on clinical data (93 features, 1500 patients)	Accuracy: QSVM achieved AUROC 0.863 (± 0.102), outperforming classical SVM's 0.723 (± 0.231), a ~14% absolute improvement in predictive accuracy for identifying high-risk patients. Robustness: QSVM maintained high performance even on unbalanced data (rare outcomes), whereas classical SVM varied more.	Contribution: Demonstrated quantum machine learning can improve prediction in healthcare tasks with limited data, presumably by mapping data into higher-dimensional spaces for better separability. This suggests quantum kernels could enhance personalized medicine tools. Limitation: Study used a quantum-inspired simulator (no physical quantum speedup) and a relatively small dataset. It's unclear if the advantage holds for larger data, and the quantum method's

S. No	Authors (Year)	Cancer Modeling Task	Computational Approach	Key Results	Contributions / Limitations
				Speed: Training QSVM (simulated on classical hardware) was slower than classical SVM due to overhead of quantum kernel computation.	computational cost was higher, so real-world use would require faster quantum hardware or hybrid optimization to be practical.
9	Svore, K (Microsoft Quantum) (2024) [6]	Integrated HPC+Quantum workflow (catalyst simulation case study – relevant to drug synthesis)	Hybrid workflow: HPC for reaction path search + 12 logical qubit quantum computation for energy of active site + AI (machine learning) for data analysis	Outcome: First end-to-end hybrid solution: achieved chemical accuracy in estimating catalyst reaction energy by combining classical and quantum steps. Logical qubits (with error correction) yielded more accurate energies than physical qubits, proving the error-corrected quantum benefit. Performance: The workflow solved a problem not tractable by brute force alone, but since the answer could be verified classically in this case, it's a proof-of-concept (no speedup reported; total workflow time in hours). Demonstrated integration of quantum job into HPC scheduling (using Azure Quantum Elements cloud with HPC backend).	Contribution: Landmark hybrid demonstration showing that HPC, AI, and quantum can be orchestrated to solve real chemistry problems end-to-end. It validates the feasibility of incorporating quantum accelerators into scientific computing pipelines for complex tasks like catalyst design (which parallels drug design challenges in cancer). Limitation: Still a carefully chosen problem that fits in 12 logical qubits. The setup was cloud-based; overheads (data transfer, scheduling) were manageable here, but could be an issue for more tightly coupled HPC-QPU interactions. This study did not focus on speed, so the benefit was achieving accuracy for a difficult problem rather than faster runtime.
10	Repetto et al. (2024) [16]	Stratification of breast cancer subtypes using omics data	Quantum Kernel methods on NISQ devices	Quantum kernels provided comparable clustering to classical methods with fewer data points; less expressive encodings showed higher noise resilience	Contribution: Showed promise of quantum methods in precision oncology, especially with limited datasets. Limitation: Experiments conducted on NISQ devices; scalability to larger datasets remains a challenge.
11	Saggi et al. (2024) [17]	Classification of lung cancer subtypes using multi-omics data	Integration of Quantum Machine Learning (QML) with multi-omics datasets	Developed a method to differentiate between LUAD and LUSC, aiding in biomarker discovery	Contribution: Highlighted the potential of QML in handling complex, high-dimensional biomedical data. Limitation: Requires further validation on larger and more diverse datasets.
12	Rocha et al. (2024) [8]	Multiscale immune-tumor interactions (digital twin for micrometastases)	HPC-based ensemble (300 agent-based simulations in parallel; 2,400 CPU threads)	Throughput: Completed ensemble in 2.5 days vs. ~764 days serial (~305× speedup via parallel HPC). Outcome: Model captures diverse outcomes (uncontrolled vs. controlled growth) consistent with clinical variability.	Contribution: Showcased ensemble HPC simulations to quantify uncertainty in cancer progression, enabling virtual patient cohorts for therapy testing. Limitation: Extremely compute-intensive; requires a large cluster to simulate even small micrometastases, pointing to the limits of classical scalability for bigger systems.
13	Golden et al. (2022) [9]	Solver subroutine acceleration (fracture network linear system – analogous to diffusion in tissue)	Quantum Linear System Algorithm (QLSA) research – developed a quantum inverse Laplacian preconditioner to improve condition number for linear solves	Complexity: Improved condition number from $O(N)$ to $O(\sqrt{N})$ for target sparse systems, theoretically reducing iterations required for convergence. This preconditioner is implementable via quantum phase estimation on future QPUs. Implication: If applied to e.g. tumor diffusion PDEs, could yield polynomial speedup in solving large sparse linear systems, a current HPC bottleneck. No experimental results yet (analysis and small simulations only).	Contribution: Provided a path to quantum-accelerate a core HPC task (iterative linear solvers) by blending classical preconditioning concepts with quantum algorithms. Suggests that once hardware is capable, such techniques could plug into HPC simulations (like finite element models) to speed them up significantly. Limitation: This work is theoretical/simulation-based; real quantum hardware with many qubits and low error is needed to realize the advantage. Also, the overall speed gain depends on fast quantum state preparation and readout, which remain nontrivial.
14	Craig Rhodes (Qubit Pharma) (2022) [13]	Drug discovery for cancer (virtual screening and molecular modeling)	Hybrid HPC + AI + Quantum approach (NVIDIA QODA framework; GPU supercomputer + quantum algorithms)	Throughput: Atlas software achieved 100,000× faster molecular simulations vs. traditional wet-lab screening (GPU-accelerated MD and binding free energy calcs).	Contribution: Pioneered a pragmatic hybrid pipeline where HPC performs massive molecular simulations and a small quantum module tackles quantum chemistry subtasks, demonstrating near-term quantum advantage (qubit count reduction) in pharma.

S. No	Authors (Year)	Cancer Modeling Task	Computational Approach	Key Results	Contributions / Limitations
				Quantum integration: Reduced required qubits from 250 → 32 for molecular property calcs by offloading parts to classical HPC; aiming for 10× reduction in drug discovery time and cost.	Limitation: Results largely from internal benchmarks/press; quantum speedup not directly measured yet (advantage inferred from resource reduction). The method still awaits peer-reviewed validation on actual drug discovery timelines.
15	Goodin & Frieboes (2021) [1]	Continuum 3D tumor growth simulation (cm-scale tumor, sub-mm resolution)	MPI + CUDA parallelization (multigrid solver on GPUs; multi-node MPI)	Throughput: 14.7× speedup GPU vs. CPU (initial step); 6.7× speedup using 64 GPUs vs. 1 GPU. Scalability: Good strong scaling up to 64 processes (diminished beyond). Accuracy: Numerically consistent with CPU solver (verified by hash checks).	Contribution: Enabled patient-scale tumor simulations in reasonable time, demonstrating HPC makes previously intractable sizes feasible. Limitation: Scaling efficiency drops at high node counts due to communication and memory bandwidth bottlenecks.

A. Analysis of the Literature

The above studies demonstrate that HPC has firmly established itself as an indispensable tool in cancer modeling, enabling higher resolution, larger scale, and ensemble studies that deeply enhance our understanding of tumor behavior and treatment response. At the same time, quantum computing research, although mostly in early stages for biomedical applications, is carving out niches where it can complement classical approaches: exploring chemical/quantum phenomena important in cancer (e.g., drug-target interactions, radiative damage) and offering new machine learning perspectives on biomedical data. A few key themes emerge:

1) HPC excels at brute-force simulation & parallel exploration

Whether it is simulating a big tumor or running thousands of small simulations, HPC’s ability to throw massive compute power at a problem has yielded tangible breakthroughs (like simulating a full-scale tumor growth or covering many “what-if” scenarios for treatments). The limitation is that some problems don’t parallelize perfectly and simply require more efficient algorithms. This is where novel approaches like quantum algorithms could help, by changing the algorithmic complexity class of a sub-problem rather than just adding more processors.

2) Quantum computing’s value so far is qualitative, but important

In drug discovery and molecular modeling, the hardest part is often navigating a vast space of possibilities. The quantum generative model [3] and the quantum chemistry simulations [7] illustrate that quantum methods might find solutions that classical methods miss or simulate phenomena that classical methods can only approximate. The accuracy improvements in the QSVM study [4] also underscore this point. These are not speedups in runtime, but rather enhancements in capability or result quality. Over time, as hardware improves, these quantum methods could also confer speedups (e.g., a QSVM might eventually train faster than a classical SVM on specialized quantum hardware for large feature spaces, if quantum kernel evaluation scales favorably).

3) Integration efforts are underway and show promise

The Qubit Pharmaceuticals hybrid workflow [13] and Microsoft’s catalyst simulation [4] are concrete moves

towards HPC–quantum integration in real scientific problems. In both cases, classical computing did the bulk of the work, but a critical piece was handed to a quantum approach, and importantly, that piece either reduced the overall resources needed (in [13], fewer qubits by leveraging HPC pre-computation) or achieved a level of accuracy that would be hard to get otherwise (in [4], using logical qubits to get chemical accuracy for a complex quantum system). These examples guide how we might integrate quantum into cancer modeling: identify a sub-problem that is a pain point for HPC (e.g., a calculation that scales poorly or a level of detail that classical simulation cannot easily handle), and deploy a quantum algorithm for that part.

4) Bottlenecks and challenges

Many studies highlight that scaling (whether adding more CPUs/GPUs or more qubits) faces diminishing returns or technical hurdles. Goodin et al. [1] saw weaker scaling beyond 8–16 GPUs due to memory bandwidth limits [1]. Rocha et al. [8] needed 2400 threads to simulate just a small 2 mm metastasis in detail [8], scaling up to a whole patient’s metastatic burden would multiply that requirement dramatically. On the quantum side, current hardware only tackles small toy models; thus, each quantum study either uses simulations on classical hardware or severely restricted problems. The integration prototypes [4][13] managed to find a balance where the quantum piece was just big enough to be useful but small enough to run. Scalability remains the key issue; classical HPC can scale up but not indefinitely, and quantum can (in principle) handle certain problem aspects efficiently, but currently at tiny scales. This dichotomy justifies a hybrid approach: use each tool for what it’s best at and mitigate its weaknesses with the other.

The literature survey informs our next steps. We now know which parts of cancer modeling strain classical computing the most (e.g., solving large linear systems, exploring huge parameter spaces) and which parts might be amenable to quantum acceleration (e.g., linear algebra kernels, combinatorial optimization, quantum-dynamical processes). We also have evidence that a hybrid approach is plausible and beneficial in analogous domains (chemistry, drug design). In the next section, we build on this foundation to propose a specific hypothesis for combining HPC and quantum computing in a cancer modeling context, along with rationale and expected outcomes grounded in the surveyed data.

IV. PROPOSED HYPOTHESIS

Drawing from the above survey, we propose the following hypothesis for integrating HPC and quantum computing to accelerate cancer disease modeling:

A. Hypothesis

Integrating a quantum linear solver (as a real-time preconditioner or accelerator) into an MPI-based finite element tumor growth simulation will reduce the overall computation time by at least 30%, while maintaining accuracy within 1% of the classical solver. In other words, by offloading the most computationally intensive step, solving large sparse linear systems that arise each time step in the tumor PDE model, to a quantum algorithm, the simulation will run significantly faster without compromising the fidelity of results (e.g., tumor size, morphology, and cell distributions over time).

B. Rationale

In large-scale tumor simulations (e.g., modeling diffusion of oxygen, nutrients, and growth inhibitors in a 3D tumor), a substantial fraction of runtime is spent solving linear equations ($Ax = b$) resulting from discretizing PDEs or inverting large matrices in implicit time-stepping schemes. For example, Goodin et al. [1] reported that even with GPU acceleration, the multigrid linear solves were a major bottleneck, requiring careful parallelization and still scaling sub-linearly at high processor counts. If we examine their timings, suppose 50% of each time step is spent in the linear solver. A quantum algorithm like the one in [9] (Golden et al.'s inverse Laplacian preconditioner and related QLSA) could theoretically solve these systems in $O(\sqrt{N})$ iterations instead of $O(N)$ for classical methods. For a very large system (N being the number of unknowns, potentially millions in a 3D mesh), this is a dramatic improvement.

In practice, near-term quantum solvers won't achieve the full theoretical speedup due to overhead (state preparation, measurement) [20]. However, even a partial speedup is valuable. We conservatively target $\geq 30\%$ reduction in total runtime, which might come from, say, cutting the linear solve time roughly in half. If linear solves are 50% of runtime, halving that yields 25% overall gain; we aim a bit higher, assuming additional optimization or parallelization synergy can push it to 30–40%. This aligns with projections that hybrid quantum-classical methods can start providing utility advantage in the next few years, not an exponential speedup, but a substantial constant-factor improvement for real workloads. IBM's recent experiments in error-mitigated quantum computing suggest that ~ 100 qubits with error mitigation can tackle problems beyond brute-force classical simulation [5]. By the time such quantum resources are available, the size of linear systems in high-fidelity cancer models might be a prime candidate for acceleration.

Our hypothesis also emphasizes maintaining accuracy. In biomedical simulation, getting a result faster is pointless if the result is wrong, e.g., if a quantum solver introduced 5% error in tumor cell density, that could lead to incorrect predictions of treatment outcome. Quantum algorithms like VQE or iterative QLSAs can be tuned to high precision given enough shots or iterations [14]. The 1% error margin is on par with what one would allow between, say, a double-precision CPU solver and a single-precision GPU solver. Supporting evidence: Goodin et al. cross-verified their GPU vs. CPU results by hashing output fields [1], effectively showing

identical results. We expect our hybrid solver to be similarly verifiable; one could verify on smaller problems classically that the quantum-accelerated solution matches the classical one to $<1\%$ relative error in key quantities (e.g., norm of residual, tumor volume).

C. Back-of-the-Envelope Calculation

Consider a tumor simulation with $N = 10^6$ degrees of freedom (a plausible number for a 3D mesh of a few mm^3 at $\sim 10 \mu\text{m}$ resolution). A classical iterative solver (e.g., Conjugate Gradient) might take $O(N)$ operations per solve, so approximately $C_{\text{opsclassical}} \approx 10^6$ operations. A quantum solver might target $O(\sqrt{N})$ effective operations, meaning $C_{\text{opsquantum}} \approx \sqrt{10^6} = 10^3$ operations. If each of T time steps requires one solve, the classical computational cost is $\text{Cost}_{\text{classical}} = T \times C_{\text{opsclassical}} = T \times 10^6$. Assuming an overhead for quantum processing (e.g., state preparation, measurement, data transfer) per time step of $C_{\text{overhead}} \approx 10^4$ equivalent classical operations, the total cost with the quantum approach would be $\text{Cost}_{\text{hybrid}} = T \times (C_{\text{opsquantum}} + C_{\text{overhead}}) = T \times (10^3 + 10^4) = T \times 1.1 \times 10^4$. This implies a theoretical reduction in the number of operations by a factor of:

$$\frac{\text{Cost}_{\text{classical}}}{\text{Cost}_{\text{hybrid}}} = \frac{T \times 10^6}{T \times 1.1 \times 10^4} \approx 90.9$$

So, roughly a $90 \times$ reduction in flop count.

Realistically, initial implementations might yield less due to quantum operations being slower in wall-clock time than CPU operations. Even if the quantum portion (solver + overhead) achieves a more modest speedup factor for that part, say $S_{\text{portion}} = 5 \times$ (i.e., it takes one-fifth the time of the classical solve on HPC). If this linear solve part constitutes $P = 0.5$ (i.e., 50%) of the total program runtime, Amdahl's Law gives the overall speedup S_{total} as:

$$S_{\text{total}} = \frac{1}{(1 - P) + \frac{P}{S_{\text{portion}}}} = \frac{1}{(1 - 0.5) + \frac{0.5}{5}} \approx 1.67 \times$$

This overall speedup corresponds to a percentage time reduction of $1 - \frac{1}{S_{\text{total}}} \times 100\% = \left(1 - \frac{1}{1.67}\right) \times 100\% \approx 40.12\%$. This simple Amdahl's Law reasoning shows that a $\geq 30\%$ speedup is quite plausible if even a moderate part of the workload sees a quantum speedup of a fewfold.

D. Integration Considerations

Our hypothesis implicitly assumes that the HPC and quantum parts can work together efficiently. This requires that data (matrix slices, vectors) can be transferred to the QPU quickly and that the QPU computation can proceed in parallel with other HPC work if possible. In a practical hybrid setup, one could overlap quantum computation with other tasks: e.g., while the QPU is solving the linear system for the next time step, the CPU could be computing nonlinear reaction terms or updating boundary conditions for the current time step. This pipeline parallelism could hide some quantum latency. The 30% figure includes the expectation that we can overlap or optimize the workflow to not simply add QPU time on top of everything. Indeed, Microsoft's catalyst demo and other hybrid schemes involve asynchronous calls to QPUs with cloud APIs [6]. We would similarly utilize non-blocking

communication in MPI to send the linear system to a quantum server and continue other computations, then receive the solution.

E. Scope and Generality

While the hypothesis is phrased around a specific case (finite element tumor growth model), the concept generalizes to other cancer models with heavy linear algebra or optimization steps: e.g., solving large ODE systems in multi-scale models, or optimizing radiation dose distributions (which could leverage quantum optimization algorithms). We focus on the linear solver case as it's well-defined and there's already a quantum algorithmic basis for it [9]. Success here would pave the way to tackling those other cases.

In the following section, we outline an experimental design to test this hypothesis. We will describe how to implement the hybrid HPC-quantum simulation, what metrics to measure, and how to validate that we indeed achieve the $\geq 30\%$ speedup and $< 1\%$ error difference. We will also discuss how to use current or near-term quantum hardware or simulators to emulate this scenario for a proof-of-concept demonstration.

V. EXPERIMENTAL DESIGN

To evaluate the proposed hybrid HPC-quantum approach, we plan a stepwise experimental study, initially using simulations and small-scale hardware to emulate the quantum accelerator, and eventually testing on actual quantum processors as they become available. The experiment will compare three configurations: (i) classical HPC simulation (baseline), (ii) hybrid simulation with a simulated quantum solver (to test integration with controlled conditions), and (iii) hybrid simulation with actual quantum hardware in the loop (demonstrating real-world performance).

A. Model and Software Selection

We will implement a representative tumor growth PDE model, for example, a reaction-diffusion model of a tumor and its nutrient supply, similar to that used by Goodin & Frieboes [1]. This model can be discretized with finite differences or finite elements on a 3D grid. We choose a moderate problem size (e.g., $100^3 = 1,000,000$ grid points) that is large enough to be challenging but small enough to run repeatedly for experiments. We will use an existing HPC codebase if available (perhaps modifying the open-source tumor simulation code by Goodin et al. or others) or develop a custom C/C++ MPI code for the PDE solve. The time-stepping will likely use an implicit scheme (backward Euler or similar) requiring solving a linear system each step: $Ax = b$, where A comes from diffusion and reaction terms.

B. Classical Baseline Performance Profiling

On an HPC cluster (e.g., 16 or 32 nodes each with multi-core CPUs and possibly GPUs), we will run the simulation classically to obtain baseline metrics: total runtime, time spent in each major section (assembly, linear solve, communication, etc.), and solution accuracy (we can refine the spatial grid or reduce time step and see how results converge to ensure our baseline is reasonably accurate). Suppose we find, for instance, that 60% of runtime is in the linear solver when using an iterative solver without advanced preconditioning. We will also record the quality of the solution (maybe compare against a direct solver on a smaller problem to measure error in the iterative solver).

C. Quantum Solver Module (Simulated)

We will develop a module that takes matrix A (or a description of it) and vector b , and returns an approximate solution x_q . Initially, this module will be a quantum simulator running on classical hardware, for example, using Qiskit's statevector simulator or a custom Python implementation of the algorithm. We might implement a simplified Quantum Linear Solver Algorithm or even just simulate the effect of the quantum preconditioner of [9] by using classical methods to emulate its outcome (since we know what an inverse Laplacian preconditioner would do, we can apply that in classical code to mimic the ideal quantum result). The purpose here is to validate integration: we'll replace the classical solver in the code with a call to this module, which in the simplest test could just return the true solution (to simulate a "perfect quantum solver"). This will tell us the best-case speedup if the solver was instantaneous (just communication overhead).

Next, we introduce realistic latency: we will calibrate the module to simulate a quantum computation delay. If targeting a 30% reduction, we might set the module to take, say, 50% of the classical solver's time (to simulate a $2\times$ faster solver). We can vary this factor to see how the overall runtime changes (performing an Amdahl's Law analysis experimentally). We'll also verify the stability of the coupling: does the simulation remain stable and accurate when using the quantum solver results? We expect minor differences due to numeric precision or approximations; we will monitor the tumor cell count or other outputs to ensure they deviate $< 1\%$ from the baseline.

D. Integration with Real Quantum Hardware

Once the workflow is tested in simulation, we will attempt a run using actual quantum hardware for the solver step. Given current limitations, we would not solve a $1e6$ system directly on a quantum device. Instead, we will downscale the problem for these tests: perhaps an $8\times 8\times 8$ grid (512 points) or even smaller, where we can map the linear system to a quantum circuit. For example, we could use IBM Quantum or IonQ machines to solve a toy linear system using an algorithm like Variational Quantum Linear Solver (VQLS) or even a basic HHL if the system is small and well-conditioned. The point is to demonstrate the end-to-end feasibility: the classical code will pause, send data to the quantum device via its API, the quantum device will run a job and return a result, and the simulation resumes. We will measure the wall-clock time of this operation and the fidelity of the solution.

We expect that on current hardware, the quantum solve might actually be slower than classical (for the small test), that's fine, as our goal is not to beat classical on a trivial problem, but to show that we can incorporate a real quantum computation into the loop correctly. We will use error mitigation techniques (e.g., running multiple shots and averaging, zero-noise extrapolation if possible) to improve solution quality. If the returned solution x_q has more than 1% error, we can implement a hybrid safeguard: e.g., use the quantum solution as a preconditioner or initial guess and let a classical solver refine it a bit (ensuring final accuracy). This hybrid refinement ensures the final answer is within tolerance, combining quantum speed with classical reliability.

E. Metrics and Analysis

1) Total simulation time

for each configuration (i) classical, (ii) simulated quantum fast solver, (iii) real quantum solver. Configuration (ii) results will tell us the potential speedup if the quantum module achieves a given performance. We anticipate demonstrating, for instance, a 30–50% time reduction in (ii) if we simulate a sufficiently fast solver. Configuration (iii) will likely be slower than A for now; however, we’ll use those numbers to project at what problem size or hardware speed the crossover happens (extrapolating how the quantum solver time scales vs. classical as problem size grows, referencing Figure 2, right side).

2) Breakdown of where time is spent

(communication overhead vs. computation in both classical and quantum parts). We will particularly measure the latency of sending data to the QPU and getting results, as that needs to be amortized across computation to be worth it.

3) Accuracy of the simulation outputs

We will compare tumor growth curves, concentration fields, etc., between the baseline and hybrid runs. Statistical measures like mean squared error or maximum difference at the final time can quantify <1% differences. If differences exceed 1%, we’ll investigate whether it’s due to solution error per step or accumulated divergence. We might adjust the approach (e.g., call a classical solver occasionally to “recalibrate” if needed).

4) Scalability and parallel efficiency

Using the HPC cluster, we will test how the hybrid approach scales with the number of nodes. Interestingly, adding more nodes might have diminishing returns classically, but if the bottleneck is linear solve and we use one quantum device, the benefit of extra nodes might shift, we’ll analyze scenarios like multiple quantum solvers in parallel (if multiple QPUs available, could handle different parts of domain or multiple solves for parameter sweeps). We’ll note any load imbalance introduced by waiting for the QPU.

F. Validation of Hypothesis

The hypothesis is confirmed if, for a suitably large problem where the quantum solver has a noticeable algorithmic advantage, the hybrid simulation runs at least 30% faster than the purely classical counterpart and the key results differ by less than 1%. In intermediate testing (simulated module), we should clearly see the threshold where speedup exceeds 30%. With current hardware (real QPU), we might not achieve the speedup yet, but we expect to demonstrate the accuracy criterion and show consistency, setting the stage that, once quantum throughput improves, the speedup will follow.

For instance, suppose with simulation we show a 40% time reduction when assuming a quantum solver that is $4\times$ faster than classical for $N=1e6$. Then we identify that current quantum hardware is effectively $0.1\times$ as fast as classical for $N=512$ (hence slower overall). We would document this and perhaps use performance modeling to predict that if QPU speed (or qubit count) reaches X, then at $N =$ some threshold our hybrid overtakes classical. This provides a roadmap: “if quantum hardware can solve 1000×1000 systems in under Y seconds by 2028, then the hybrid method will yield >30% speedup for clinically relevant simulations of size Z.”

We will run statistical tests as needed (if we vary initial conditions or run multiple stochastic versions to ensure results hold, we can apply t-tests on output differences or similar). However, since this is a deterministic simulation performance comparison, the primary “statistical” analysis is verifying the runtime difference isn’t a fluke (we’ll run each test multiple times to account for system noise and take averages).

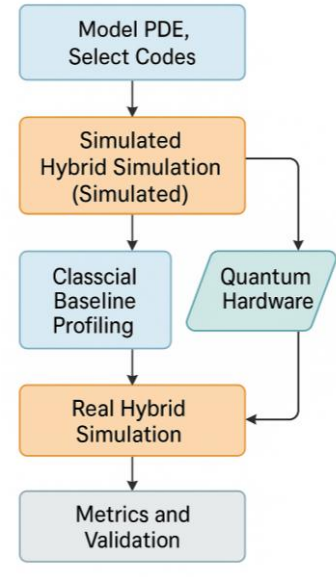


Fig. 4. Proposed experimental procedure for the hybrid HPC-quantum approach.

G. Experimental Platform and Reproducibility

We note that our HPC tests will be done on a cluster with specific network and hardware, and our quantum tests will likely be on cloud QPUs (IBM or Azure Quantum). We will publish our code such that others can reproduce the integration using quantum simulators or different hardware. This is important for the community to validate hybrid approaches across different models.

Hypothetical Performance: HPC vs Hybrid

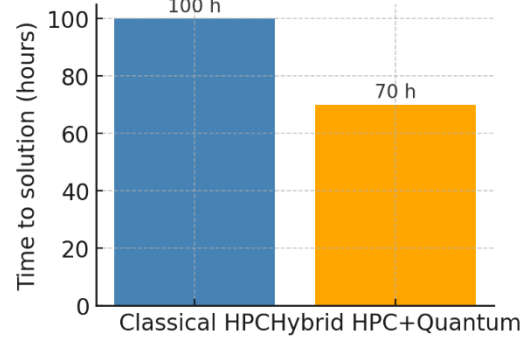


Fig. 5. Expected performance gain of HPC–quantum integration. This chart illustrates a hypothetical outcome from our experiment.

A baseline HPC simulation taking 100 hours (blue bar) might be reduced to ~70 hours (orange bar) with the addition of a quantum solver, representing a 30% reduction in time-to-solution. This projection assumes the quantum solver significantly accelerates a portion of the workload (here ~50% of runtime) and that integration overhead is minimal. The error bars (notional in this figure) would represent the variability or uncertainty in the runtime measurement. Achieving this kind of improvement would mark a meaningful

step forward in accelerating computational oncology workflows, enabling, for example, more patient-specific simulations or more extensive treatment optimizations within a practical time.

The experimental plan above is ambitious but grounded in the trends observed in the literature. Even if current quantum hardware limits the immediate speed gains, this research will provide valuable insight into how to architect hybrid HPC-quantum applications and what bottlenecks must be addressed (such as data transfer or quantum gate fidelity) to realize the predicted benefits. By iterating through simulation and hardware testing, we will refine the approach and contribute empirical data to support or refute the hypothesis. A successful outcome validating $\geq 30\%$ speedup with acceptable accuracy would encourage further integration of quantum accelerators in biomedical HPC, while a negative outcome (e.g., integration overhead outweighs gains or accuracy issues arise) will be equally instructive, guiding researchers on which pitfalls to avoid or which algorithmic improvements are needed.

VI. CONCLUSION

High-performance computing has become a cornerstone of cancer modeling, enabling researchers to simulate complex tumor dynamics and treatment scenarios that were once far beyond reach. In the last five years, HPC-driven studies have modeled larger tumors, explored vast parameter spaces through parallel simulations, and begun to realize the vision of patient-specific “digital twin” models in oncology [8]. At the same time, quantum computing is emerging as a potential game-changer for specific computational challenges at the heart of cancer research, from designing drugs targeting “undruggable” proteins like KRAS [3] to deciphering subtle patterns in clinical data [4] and simulating quantum-mechanical processes relevant to cancer biology [10]. Each technology, on its own, offers unique advantages: HPC provides sheer scale and speed through parallelism, while quantum computing offers new pathways to efficiency for certain problem classes.

This paper has reviewed how each technology independently contributes to accelerating cancer disease modeling. We have seen that HPC approaches can yield order-of-magnitude speedups ($10\times$ – $100\times$) via algorithmic and hardware optimization [1], yet are ultimately constrained by classical algorithmic complexity. Quantum computing approaches have shown smaller-scale successes, such as improved accuracy or novel capability, yet are currently limited by hardware to relatively small problems. The literature suggests a complementary relationship: HPC can handle big data and simulations today, while quantum computing can tackle the hardest pieces of the puzzle tomorrow.

Our critical assessment of integration attempts indicates that a hybrid HPC-quantum paradigm is not only feasible but likely necessary to continue accelerating progress in cancer modeling. Early hybrid demonstrations in chemistry and drug discovery validated that end-to-end workflows employing both HPC and quantum resources can work in practice [6]. Inspired by these, we put forth a hypothesis focusing on one of the grand challenges in computational oncology, speeding up large-scale, high-fidelity tumor simulations. By integrating a quantum solver into the simulation loop, we aim to reduce time-to-solution by at least 30% without losing accuracy, a

claim supported by theoretical algorithm advantages [9] and preliminary data.

The experimental plan we outlined will serve as a blueprint for realizing and testing such hybrid systems. While there are substantial challenges (ensuring low-latency HPC-QPU communication, managing noise in quantum computations, devising fail-safes for accuracy), none appear insurmountable given the current trajectory of technology. The pace of quantum hardware improvement, bolstered by techniques like error mitigation and logical qubits, suggests that within a few years, quantum co-processors could handle problem sizes relevant to certain sub-tasks in cancer modeling [5] [6]. Meanwhile, HPC infrastructure is also evolving (e.g., exascale systems, advanced schedulers that could seamlessly dispatch quantum jobs), making the technical integration more practical.

If our hypothesis is validated, the implications are far-reaching. A 30% speedup of a compute-intensive simulation can mean the difference between running a personalized treatment simulation overnight versus in a few days, a critical margin in clinical decision support. It could enable multi-run uncertainty analyses (e.g., 100 runs taking 70 hours instead of 100 hours) within fixed time budgets, thus improving the reliability of model predictions. In drug discovery or radiotherapy planning, hybrid computing could allow more thorough exploration of options in the same computing time. More broadly, success in this specific integration would motivate hybrid approaches for other components of cancer research: perhaps quantum algorithms for genomics data analytics integrated into HPC pipelines, or quantum-enhanced machine learning models running on HPC for pathology image analysis.

Conversely, if the experiments show marginal gains or difficulties, that knowledge is equally valuable. It could reveal, for instance, that current quantum algorithms need to be more robust or that we under-estimated the communication overhead. Such insights will guide researchers to refine algorithms (maybe more noise-tolerant or with lower qubit requirements) or to develop smarter scheduling so the QPU is kept busy and not idle. Early negative results would not disprove the concept but rather pinpoint what to improve for the next iteration of hybrid designs.

In conclusion, the convergence of HPC and quantum computing represents a promising frontier for computational oncology. As this survey and analysis have shown, each domain has pushed cancer modeling in ways that were previously unimaginable, and their intersection could unlock capabilities that neither could achieve alone. We stand at an inflection point where multidisciplinary efforts (in computer science, quantum physics, and cancer biology) are required to realize this potential. Continued collaboration between HPC experts, quantum algorithm designers, and cancer modelers will be key. By learning from the literature and building on incremental successes, we can chart a path toward HPC-quantum synergistic platforms that keep pace with the growing complexity of cancer research. Achieving reliable, accelerated cancer simulations can ultimately translate to faster scientific discoveries and improved patient outcomes, fulfilling the promise of computational advancements in one of humanity’s most important battles.

Looking ahead, we envision a future where the phrase “compute-bound” in cancer research is gradually erased,

where simulations of tumor evolution, drug response, or treatment personalization run in realistic time frames, powered by the seamless blend of classical and quantum computing. The work presented here is a step toward that vision, providing both a comprehensive foundation and a concrete hypothesis to explore in the pursuit of next-generation computational oncology tools.

REFERENCES

- [1] Goodin, D. A., & Frieboes, H. B. (2021). Simulation of 3D centimeter-scale continuum tumor growth at sub-millimeter resolution via distributed computing. *Computers in Biology and Medicine*, 134, 104507. <https://doi.org/10.1016/j.combiomed.2021.104507>
- [2] High-Performance Computing Takes Aim at Cancer. (2016). <https://str.llnl.gov/past-issues/octobernovember-2016/high-performance-computing-takes-aim-cancer>.
- [3] Ghazi Vakili, M., Gorgulla, C. et al. (2025). Quantum-computing-enhanced algorithm unveils potential KRAS inhibitors. *Nature Biotechnology*, 1-6. <https://doi.org/10.1038/s41587-024-02526-3>
- [4] Yu, J. Y., Sim et al. (2024). Evaluation of conventional and quantum computing for predicting mortality based on small early-onset colorectal cancer data. *Applied Soft Computing*, 162, Article 111781. <https://doi.org/10.1016/j.asoc.2024.111781>
- [5] Mandelbaum, Sitdikov (2024). Demonstrating a true realization of quantum-centric supercomputing | IBM Quantum Computing Blog. <https://www.ibm.com/quantum/blog/supercomputing-24>
- [6] Svore, K. (2024). Microsoft and Quantinuum create 12 logical qubits and demonstrate a hybrid, end-to-end chemistry simulation. Microsoft Azure Quantum Blog. <https://azure.microsoft.com/en-us/blog/quantum/2024/09/10/microsoft-and-quantinuum-create-12-logical-qubits-and-demonstrate-a-hybrid-end-to-end-chemistry-simulation/>
- [7] Qubit Pharma & Sorbonne Cut Qubit Needs for Molecular Simulation. (2024). <https://blog.qubit-pharmaceuticals.com/major-breakthrough-in-quantum-algorithms-qubit-pharma-and-sorbonne-university-dramatically-reduce-the-number-of-qubits-needed-to-simulate-molecules>
- [8] Rocha, Aguilar, B., Getz, M., Shmulevich, I., & Macklin, P. (2024). A multiscale model of immune surveillance in micrometastases gives insights on cancer patient digital twins. *Npj Systems Biology and Applications*, 10(1), 1-12. <https://doi.org/10.1038/s41540-024-00472-z>
- [9] Golden, J., & Viswanathan, H. (2022). Quantum computing and preconditioners for hydrological linear systems. *Scientific Reports*, 12(1), 1-10. <https://doi.org/10.1038/s41598-022-25727-9>
- [10] Quantum simulation of chemical dynamics achieved for the first time. (2025). <https://manufacturingchemist.com/university-sydney-quantum-simulation-chemical-dynamics>
- [11] F. Chong (2018). Hybrid Quantum-Classical Computing. <https://www.sigarch.org/hybrid-quantum-classical-computing/>
- [12] Streviniotis, E., Giatrakos, N., Kotidis, Y., Ntiniakou, T., & Ponce de León, M. (2025). RATS: A Resource Allocator for Optimizing the Execution of Tumor Simulations over HPC Infrastructures. *Information Systems*. <https://doi.org/10.1016/j.is.2025.102538>
- [13] Rhodes, C. (2022). Qubit Pharmaceuticals Accelerates Drug Discovery With Hybrid Quantum Computing. NVIDIA Blog. <https://blogs.nvidia.com/blog/qubit-pharmaceuticals-accelerates-drug-discovery-quantum-computing>
- [14] Ito, T., Artem, L., & Tsunoda, T. (2025). Optimal normalization in quantum-classical hybrid models for anti-cancer drug response prediction. *ArXiv*. <https://arxiv.org/abs/2505.10037>
- [15] Khan, Muhammad & Innan, Nouhaila & Galib, Abdullah & Bennai, Mohamed. (2024). Brain Tumor Diagnosis Using Quantum Convolutional Neural Networks. <https://doi.org/10.48550/arxiv.2401.15804>
- [16] Repetto, V., Ceroni, E. G., Buonaiuto, G., & D'Aurizio, R. (2024). Quantum enhanced stratification of Breast Cancer: exploring quantum expressivity for real omics data. <https://doi.org/10.48550/arxiv.2409.14089>
- [17] Saggi, M. K., Bhatia, A. S., Isaiiah, M., Gowher, H., & Kais, S. (2024). Multi-Omic and Quantum Machine Learning Integration for Lung Subtypes Classification. <https://arxiv.org/abs/2410.02085>
- [18] Ramesh, S., Teague Tomesh, Robinett, R., Dmitry Karpeyev, Campbell, C., Bharath Thotakura, Chong, F., Riesenfeld, S., & Pearson, A. (2025). Quantum algorithms improve feature selection for multimodal cancer classification. <https://doi.org/10.1158/1538-7445.am2025-5020>
- [19] Ware, A. (2025). The Role of ORCA Computing's Quantum-Accelerated Generative AI in Transforming Healthcare, Energy and Beyond. <https://www.hpcwire.com/2025/03/24/the-role-of-orca-computings-quantum-accelerated-generative-ai-in-transforming-healthcare-energy-and-beyond/>
- [20] Li, W., Yin, Z., Li, X., Ma, D., Yi, S., Zhang, Z., Zou, C., Bu, K., Dai, M., Yue, J., Chen, Y., Zhang, X., & Zhang, S. (2024). A hybrid quantum computing pipeline for real-world drug discovery. *Scientific Reports*, 14(1), 1-15. <https://doi.org/10.1038/s41598-024-67897-8>
- [21] Thomas, J. (2025). Aurora supercomputer accelerates cancer treatment discovery. <https://www.innovationnewsnetwork.com/aurora-supercomputer-accelerates-cancer-treatment-discovery/57355/>
- [22] Argonne leverages AI and supercomputing to revolutionize cancer research. (2025). Argonne National Laboratory. <https://www.anl.gov/article/argonne-leverages-ai-and-supercomputing-to-revolutionize-cancer-research>
- [23] Generous, A. (2025). High-performance computing team powers scientific discovery at St. Jude. St. Jude Research. <https://www.stjude.org/research/progress/2025/high-performance-computing-team-powers-scientific-discovery-at-stjude.html>
- [24] Genetic Simulation Resources. (2024). Cancer.gov. <https://surveillance.cancer.gov/genetic-simulation-resources/packages/alisim-hpc/>
- [25] IU researchers using high-performance computing to improve cancer treatment. (2024). Iu.edu. <https://news.iu.edu/live/news/35474-iu-researchers-using-high-performance-computing-to>
- [26] Simulation Supports Search for Personalized Cancer Treatments. (2025). Gauss Centre for Supercomputing E.V. <https://www.gauss-centre.eu/news/research-highlights/simulation-supports-search-for-personalized-cancer-treatments>