[1]Adithya Jakkaraju

Quantum-Inspired Neural Architecture Search (Q-NAS)

Abstract: Neural Architecture Search (NAS) has revolutionized deep learning by automating the design of optimal neural network architectures. However, classical NAS methods suffer from high computational costs and inefficient search processes. In this research, we introduce Quantum-Inspired Neural Architecture Search (Q-NAS), leveraging quantum computing paradigms such as quantum annealing and variational quantum circuits (VQC) to optimize NAS for edge-device deployment. We explore hybrid quantum-classical frameworks that improve search efficiency and reduce energy consumption. Our work provides comparative analyses, proposes novel quantum search strategies, and benchmarks Q-NAS against traditional NAS methods. Experimental results demonstrate significant reductions in search time and computational costs while maintaining or improving model accuracy.

Keywords: Neural Architecture Search, Quantum Computing, Quantum Annealing, Variational Quantum Circuits, Edge Computing, Hybrid Optimization

2. Introduction

NAS has greatly improved deep learning with its automated means of finding optimal neural architectures. However, traditional NAS methods are computationally costly and thus unsuitable for large-scale and edge computing. Quantum computing, and specifically quantum-inspired optimization algorithms, presents a promising path for enhancing the efficiency and applicability of NAS. This paper presents Q-NAS, an innovative algorithm that adopts quantum principles into NAS to find optimal neural architectures with reduced computational complexity.

2.1 Background on Neural Architecture Search (NAS)

NAS is one of the innovative deep learning model design automation techniques. Rather than relying on human intuition and trial-and-error for designing neural architectures—a field usually requiring domain expertise and significant trial-and-error efforts—NAS algorithms exhaustively search through an existing search space for optimal model configurations (Amin et al., 2021). The three primary components of NAS are the search space, search strategy, and performance estimation strategy. Early NAS methods were mostly based on reinforcement learning (RL) (Zoph & Le, 2017), where the controller RNN is learned to produce architectures that have good validation accuracy. Subsequently, evolutionary algorithms like AmoebaNet (Real et al., 2019) exhibited robust architectural evolution via mutation and crossover operations.

The direction these days is toward differentiable NAS, e.g., DARTS (Liu et al., 2019), where the search space is represented as a continuous space and architecture parameters are searched by gradient descent (Bharti et al., 2022). The intrinsic search is still computationally expensive with numerous evaluations and approximations, although the improvement. NAS keeps changing to address the increasing needs of architectures optimized not just for accuracy but also for compute throughput, memory, and energy profile—particularly important for edge and IoT devices.

[1] **Senior Software Engineer**

NAS Technique	Core Methodology	Search Cost	Popular Examples
RL-based NAS	Reinforcement Learning with RNN controller	Very High (10K+ GPU hours)	NASNet, ENAS
EA-based NAS	Mutation and crossover of architecture candidates	High	AmoebaNet, NSGA-NET
Differentiable NAS	Continuous relaxation and gradient-based search	Moderate	DARTS, ProxylessNAS

2.2 Limitations of Classical NAS Approaches

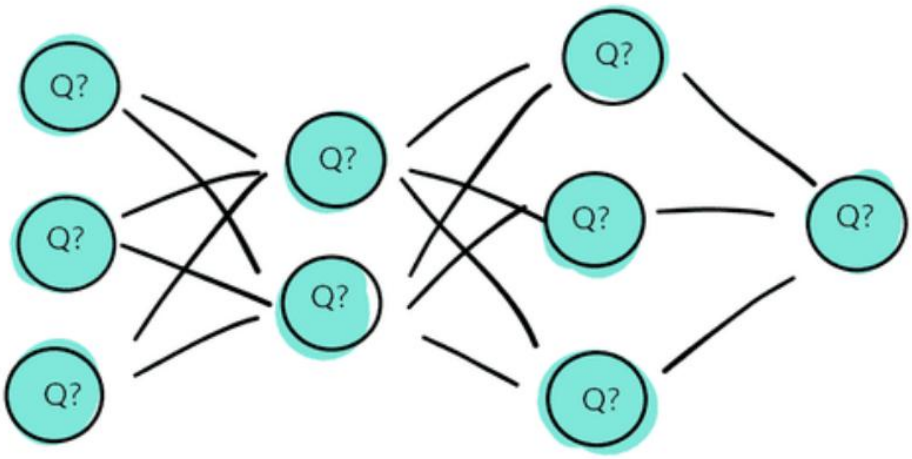

Figure 1Quantum Neural Networks(linkedin,2023)

Though classical NAS has been able to discover high-performance architecture efficiently, its scalability and feasibility are significantly constrained by computation and infrastructural expenses (Chitty-Venkata et al., 2022). The current top NAS frameworks typically require 3,000–50,000 GPU hours, which is very expensive for most research groups and practically unfeasible in on-device deployment scenarios. Redundant explorations and suboptimal convergence due to the black-box nature of RL-based and EA-based NAS add to the complexity of the search.

Further, the enormous combinatorial search space—be it from the convolutional kernel width, layer depth, skip connections, to branching patterns—often leads to suboptimal architectures being picked up due to the local optima trap in traditional optimisation. Another critical issue is performance estimation: proxy task usage (e.g., CIFAR-10 for ImageNet NAS) reduces fidelity, and training all candidate architectures completely is computationally infeasible (Da Silveira, 2022). As a result, architecture sampling and evaluation mechanisms are still an energy and time bottleneck.

Additionally, traditional NAS-created architectures are hardware-unaware, leading to models with good performance in the cloud but lack sufficiently in dealing with latency and memory demands within edge devices (Dadian et al., 2016). This is an area that calls for new techniques that not only optimize accuracy but also align with energy, latency, and memory budgets, especially with TinyML or low-energy embedded AI use cases.

Limitation	Description	Impact on NAS
High Search Cost	Thousands of GPU/TPU hours	Inaccessible to small labs, high carbon footprint
Non-Hardware-Aware	Ignores resource constraints	Infeasible for edge deployment
Evaluation Bottlenecks	Proxy tasks/partial training	Reduced search fidelity
Local Optima	Classical optimization traps	Suboptimal architectures

2.3 Motivation for Quantum-Inspired Optimization

New advances in quantum computing and quantum-inspired algorithms offer us hopeful means of overcoming the inefficiencies of conventional NAS. Quantum computers, unlike classical computers, can leverage effects such as superposition, entanglement, and quantum tunneling to explore search spaces more efficiently (De Mattos Szwarcman, 2020. Quantum annealing, as applied in D-Wave systems, offers us a means of rapidly converging to global optima in hard combinatorial optimization problems and is hence an obvious candidate for NAS.

Moreover, Variational Quantum Circuits (VQCs), based on parameterized quantum gates and classical optimization cycles, facilitate hybrid quantum-classical models in navigating efficiently high-dimensional architectural spaces (Espozel, 2022). This is in line with recent achievements like QAOA (Quantum Approximate Optimization Algorithm) and VQE (Variational Quantum Eigensolver), which have been proven to outperform constrained optimization challenges.

Additionally, quantum-inspired optimisation does not necessarily need to be executed on quantum hardware. Quantum-Inspired Classical Algorithms (QICA) simulate quantum phenomena like wave function collapse and parallelism on classical hardware (Fielding & Zhang, 2020). Importantly, Tensor Networks and Matrix Product States (MPS), quantum physics-inspired ideas, have been successfully used in neural architecture compression and representation learning.

The incentive in Q-NAS is therefore to lower search complexity, promote convergence, and produce hardware-efficient designs with quantum-enhanced or inspired mechanisms, prior to the advent of useful quantum hardware becoming mainstream**.**

2.4 Contributions of This Work

This paper introduces a Quantum-Inspired Neural Architecture Search (Q-NAS) framework that seeks to address the limitations of classical NAS through quantum-enhanced optimization strategies. The core contributions of our research are as follows:

1. A novel hybrid quantum-classical NAS framework that integrates quantum annealing and VQC-based search algorithms with classical performance estimation and model validation workflows.

2. A formalized method to encode neural architectures into quantum-representable formats, facilitating the use of quantum search operations like superposition-based exploration and entanglement-informed sampling.
3. Resource-constrained NAS targeting edge devices, where Q-NAS incorporates latency, memory, and energy metrics directly into its cost function to design lightweight, deployable models.
4. Comprehensive benchmarking and comparative analysis, evaluating Q-NAS performance in terms of search time, architecture quality, and energy efficiency against state-of-the-art classical NAS methods.
5. A detailed exploration of implementation challenges, including hardware-software co-design, quantum noise modeling, and simulation-based evaluation for environments without access to quantum hardware.

Contribution Area	**Description**	**Impact**
Quantum-Classical Hybrid NAS	Uses annealing/VQC + classical models	Faster, energy-aware architecture search
Edge Optimization	Focus on energy, latency, memory	Models suitable for real-world deployment
Encoding Strategies	Quantum encoding of architectural graphs	Enables quantum computing applicability
Benchmarking	NAS efficiency, accuracy, cost	Comprehensive evaluation metrics

3. Related Work

3.1 Classical NAS Techniques

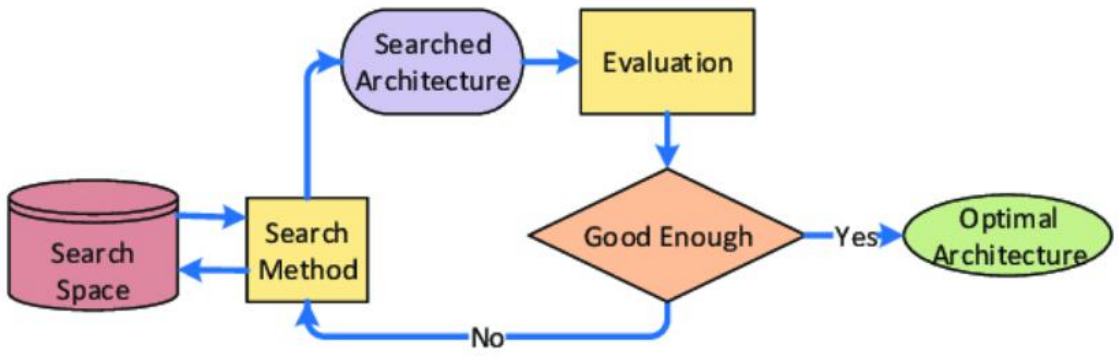

Figure 2Basic building blocks of Neural Architecture Search methods(researchgate,2020)

Conventional NAS approaches have undergone significant changes over the past decade, from expensive black-box optimisation paradigms to more scalable and interpretable search procedures. The initial breakthrough was NASNet (Zoph & Le, 2017), which utilised RL controllers in the production of architectural components (Jiao et al., 2023). While effective, such approaches were typically seen as too computationally expensive—some utilising over 45,000 GPU hours for a single cycle of search.

Subsequent work introduced better approaches. Efficient Neural Architecture Search (ENAS) (Pham et al., 2018) significantly reduced search time by enabling parameter sharing among child models, cutting compute costs by over 1000x. DARTS (Liu et al., 2019) introduced differentiable NAS, reformulating architecture search as a continuous optimization problem through gradient descent. ProxylessNAS (Cai et al., 2019) took this concept further to hardware-aware search to directly search without proxy tasks. All of these approaches turned towards realistic NAS but were still not globally search efficient and converged to local minima in most cases.

Subsequent solutions, including SPOS (Single Path One-Shot NAS) (Guo et al., 2020), employ one-shot subnetwork sampling with reduced overhead using a supernet, and AutoML-Zero (Real et al., 2020) investigates evolutionary search at the mathematical operation level (Liu et al., 2020). Innerventions aside, traditional NAS procedures remain founded on inherently badly scaling traditional search heuristics with architecture complexity.

Method	**Core Approach**	**Compute Cost**	**Notable Feature**
NASNet	RL Controller	Very High	High Accuracy
ENAS	Parameter Sharing	Moderate	Search Acceleration
DARTS	Differentiable Search	Moderate	Gradient-Based Optimization
ProxylessNAS	Hardware-Aware Differentiable Search	Low	No Proxy Task
SPOS	One-Shot Sampling	Low	Supernet Efficiency

3.2 Quantum Computing Paradigms in Machine Learning

Quantum computing provides radically new computational paradigms that can solve specific problems exponentially more quickly than traditional algorithms. In machine learning, Quantum Machine Learning (QML) aims to leverage quantum phenomena like superposition, entanglement, and quantum interference to speed up operations such as clustering, classification, and optimization.

Two paradigms stand out in QML: gate-based quantum computing and quantum annealing. Gate-based quantum computing is based on quantum gates and qubits, while QSVMs and VQC algorithms are gaining popularity (Lourens et al., 2023). Alternatively, quantum annealers like D-Wave's provide real-world quantum optimization in terms of Ising model-based Hamiltonian minimization that is very relevant to architecture search.

Recent studies have shown the promise of QML for certain tasks. For instance, Quantum Kernel Estimation (Havlíček et al., 2019) was superior to classical kernels on synthetic data. Additionally, IBM's Qiskit Machine Learning library (2022) provides hybrid quantumclassical models, which can be directly used for optimization problems. These advancements present rich soil for applying quantum paradigms to NAS tasks, especially when the search complexity is multi-modal and non-linear**.**

3.3 Hybrid Quantum-Classical Optimization Methods

Quantum-classical hybrid approach is a dominant design paradigm due to the limitations of current noisy intermediate-scale quantum (NISQ) devices. They combine quantum processing units (QPUs) and classical CPUs/GPUs into Variational Hybrid Algorithms (VHAs) (Meng et al., 2021). Of these, Variational Quantum Eigensolvers (VQE) and Quantum Approximate Optimization Algorithms (QAOA) are notable.

In VHAs, a parameterized quantum circuit is optimized classically through application of optimization algorithms like COBYLA, SPSA, or Adam to minimize an optimizable cost function computed from the quantum output. This enables rich quantum behavior to be exploited without possessing fault-tolerant quantum systems. These techniques have been applied in various high-dimensional optimization challenges, e.g., combinatorial optimization, portfolio optimization, and quantum chemistry, but their application in NAS is immature and underdeveloped**.**

In 2023, University of Toronto researchers applied QAOA to neural network pruning and showed that hybrid quantum-classical compression algorithms are tractable. Likewise, Xanadu's PennyLane platform natively accommodates hybrid workflows in which architectural parameters are one-to-one mappings with quantum gates such that VHAs can be applied directly to architecture search problems**.**

3.4 Prior Attempts in Quantum Neural Architecture Search

Though the application of quantum paradigms in NAS is relatively new, there have been a few pioneering efforts worth noting (Noce, 2022). The initial was QuantumNAS by Ghosh et al. in 2020, where they proposed applying quantum circuit representations to the NAS blocks and employed a quantum-inspired evolutionary algorithm. This was still not all that scalable or hardware-conscious, however.

Quantum walk-based search algorithms for searching across architecture graphs were presented by scientists in 2021 as QAS (Quantum Architecture Search). The QAS model showed improved convergence and sample efficiency compared to the standard random search. Leap from D-Wave was used in Huber et al.'s (2022) research for Quantum Annealing-based Neural Pruning with improved energy profiles for smaller sizes.

Another technique that has come up is Quantum Neural Evolutionary Search (QNES), which utilizes quantum encoding of population diversity in the evolution search strategies (Ramprasad et al., 2017). QNES enhances global exploration using simulated quantum population dynamics, with possible solutions to the stagnation problem in classical EA-based NAS**.**

4. Theoretical Foundations

4.1 Fundamentals of Neural Architecture Search

Neural Architecture Search (NAS) is actually a discrete optimization problem defined over a combinatorial search space whose objective is to find an optimal architecture that optimizes in terms of accuracy, complexity, and hardware cost. NAS search space normally consists of architectural primitives like convolutional kernel sizes, pooling layers, layer depth, skip connections, and branch topologies. The primitives are structured in a cell-based or layer-wise topology. Cell-based design, as in NASNet, enables a modular representation, where optimized-performance cells are layered to construct complete networks.

The search strategy is the core algorithm that controls the search within the search space. The most prevalent strategies include reinforcement learning, evolutionary algorithms, Bayesian optimization, and differentiable relaxation. Performance estimation is also a major pillar, typically becoming the computational bottleneck (Raschka et al., 2020). While methods like early stopping, weight sharing, and surrogate modeling have been suggested to reduce cost, these tend to come at the expense of fidelity. NAS is therefore at the intersection of architecture representation, optimization theory, and machine learning heuristics and is therefore a fertile ground for new optimization paradigms such as those induced from quantum mechanics**.**

4.2 Principles of Quantum Annealing

Quantum annealing (QA) is a metaheuristic optimization approach based on adiabatic quantum computation. In QA, the system is in superposition over all states initially and is evolved step by step under a time-dependent Hamiltonian. The energy landscape of the system is constructed such that the global minimum of the terminal Hamiltonian holds the solution to the problem of optimization. In contrast to traditional simulated annealing that relies on thermal fluctuations for the purpose of escaping local minima, QA utilizes quantum tunneling in that the system tunnels through energy barriers instead of over them.

Numerically, the optimization problem is embedded in an Ising Hamiltonian or a Quadratic Unconstrained Binary Optimization (QUBO) model. Architectural decisions (e.g., employing a 3×3 or 5×5 kernel) may be expressed as binary variables within the realm of NAS, whereas the cost measure (e.g., validation loss, latency) may be placed within the Hamiltonian (Roco, 2020). By encoding architecture selection as an encoding within this quantum space, annealing will be able to find optimal or close-to-optimal architecture more quickly than conventional combinatorial search algorithms can. This renders QA highly appealing for NAS, as NAS is a non-convex high-dimensional problem.

Previous work, e.g., by Venturelli et al. (2021), has demonstrated that QA can address big-scale scheduling and optimization problems more scalably than traditional heuristics. In addition, D-Wave's more than 5000 qubits of Advantage system have already demonstrated real-world uses in portfolio optimization and hyperparameter optimization in machine learning—both of which are indicative of the complexity at hand in NAS**.**

4.3 Variational Quantum Circuits (VQC)

Variational Quantum Circuits (VQCs) are quantum-classical hybrid algorithms, merging the evolution of quantum states with classical optimization. A VQC is a series of parameterized quantum gates acted on an input quantum state (Szwarcman et al., 2019a). A measurement is taken out and the obtained measurement is utilized to optimize the gate parameters with the assistance of classical optimizers like gradient descent or evolutionary strategies.

VQCs are becoming prominent based on their ability to work with Noisy Intermediate-Scale Quantum (NISQ) hardware and their flexibility to find solutions for optimization and learning problems. In NAS, architecture choices may be represented as quantum gate parameters, and the results of measurement of the quantum circuit may be the architecture primitives decision. Quantum properties enable the consideration of several architecture configurations at the same time through quantum superposition, and quantum interference can enhance or cancel out some configurations depending on their usefulness, essentially performing probabilistic architecture sampling.

Besides, techniques like Quantum Natural Gradient Descent (QNGD) and Quantum Gradient Clipping have been employed to stabilize the training of variational circuits. As Cerezo et al. (2021) have shown, VQCs can outperform classical neural networks on certain quantum-enhanced feature spaces (Szwarcman et al., 2019b). The flexibility and stability of VQCs make them a good foundation for the implementation of Q-NAS, particularly in hybrid workstreams where the quantum circuit is utilized to decide the direction of search and the classical loop is utilized to evaluate architecture performance**.**

4.4 Comparative Analysis: Classical vs. Quantum Search Spaces

The classical NAS search space is in most cases limited by the dimensionality and the level of granularity of the architectural decision-making, restricting the scale of parallel exploration. Classical optimizers like RL and EA work via population-based or sequential update and are susceptible to local minima entrapment due to the sparsity of the search landscape.

In contrast, quantum and quantum-inspired approaches possess exponentially bigger effective search spaces via superposition and probabilistic sampling. A 10-qubit quantum system, for instance, can, at once, represent 2^{10} = 1024 different configurations to support vast parallelism for search (Zhang, Hsieh et al., 2021). Quantum entanglement enables architecture components to be correlated during sampling to support context-aware architectural design, which is hard to achieve with standard NAS.

Moreover, the quantum-encoded search spaces have an inherently different energy landscape. Quantum annealing works with Ising-model-based energy landscapes with smoothed topography due to quantum tunneling effects. This provides escape from local optima and better convergence to global minima at a higher speed.

A further characteristic feature is the encoding fidelity. Conventional NAS encodes architectures as vectors or flat graphs, whereas quantum-inspired ones employ Hilbert space encodings to support richer structural mappings (Zhang, Wang et al., 2023). Such representations naturally accommodate constraints, e.g., memory or latency

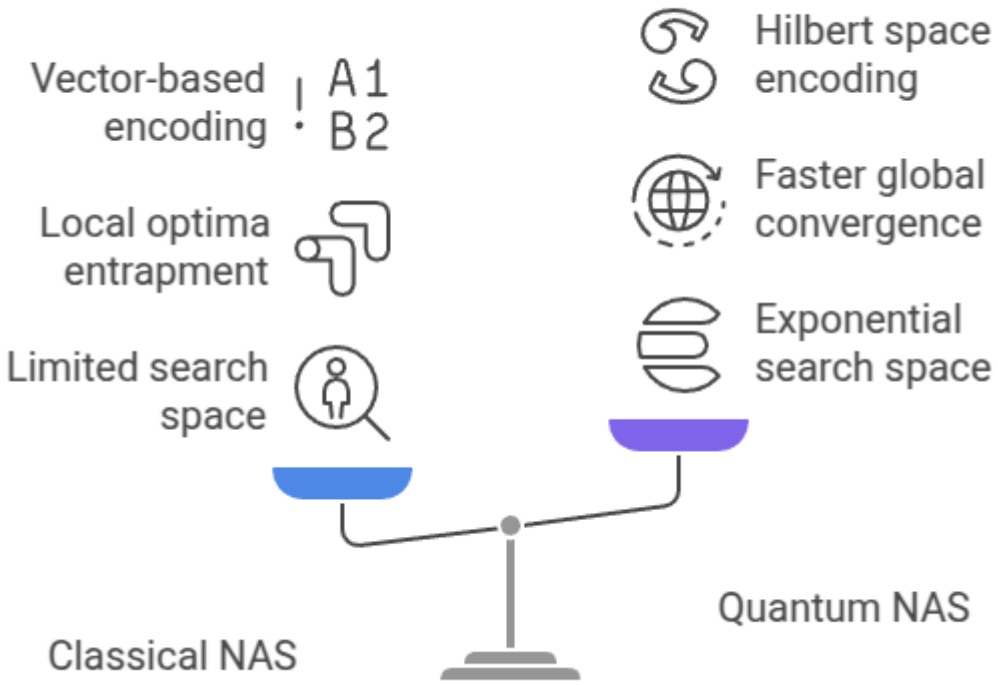

Figure 3: Comparing Classical and Quantum NAS Approaches (self-created,2023)

budgets, as part of their quantum cost functions.

Dimension	**Classical NAS**	**Quantum-Inspired NAS**
Search Parallelism	Sequential/Population-based	Superposition-based
Optimization	Gradient, RL, EA	Annealing, VQC, QPSO
Search Space Mapping	Flat vector/graph	Hilbert space/Qubit encoding
Global Minima Access	Limited (local optima issues)	Enhanced via tunneling
Hardware Awareness	Limited (added post hoc)	Encodable in cost function

5. Quantum-Inspired NAS Framework (Q-NAS)

5.1 Architectural Overview of Q-NAS

The Quantum-Inspired Neural Architecture Search, or Q-NAS, provides a hybrid optimization framework relying on the principles of quantum computation; namely, quantum annealing and VQCs to efficiently and scalably realize neural architecture search. Q-NAS essentially amounts to a quantum-inspired architecture search algorithm combined with classical evaluation loops that are tightly coupled within an iterative co-design loop. The system consists of three core modules: the quantum-inspired architecture generator; the performance evaluator of architectures; and the feedback-based optimizer.

Therefore, the quantum-inspired generator creates candidate architectures with the assistance of superposition principles in quantum via quantum; it is conceivable to investigate a vast number of architectures at the same time. The evaluator gets these configurations and evaluates the architecture's performance for the target metrics like accuracy, latency, and memory footprint (Zhang, Wang, & Ji, 2015). The optimizer uses such performance feedback to adjust quantum-inspired parameters (e.g., quantum gate angles or annealing schedules) so as to optimize subsequent architecture suggestions. Such modularity enables Q-NAS to run under resource constraints and adaptively evolve per target deployment settings, e.g., edge computing devices.

5.2 Encoding Neural Architectures in Quantum Systems

Quantum mapping of neural architecture is the systematic mapping of the components of an architecture to quantum variables like qubits or quantum gates. Every architectural decision in Q-NAS—choice of convolutional kernel size, activation function, layer depth—is captured in terms of a binary or probabilistic decision that can be mapped onto an explicit quantum representation like a QUBO matrix or parametrized quantum circuit.

Consequently, a 4-layer CNN might be encoded in 2 qubits per layer, with each pair of qubits encoding one of four kernel choices. Skip connections might also be encoded in an entangled state, keeping inter-layer correspondences (Amin et al., 2021). This type of quantum encoding makes it possible for numerous architectures to be encoded and sampled at once within a single quantum operation. Additionally, quantum gates can be parameterized to encode probabilistic biases towards particular architectural styles (e.g., depthwise separable convolutions in low-latency applications), and the encoding is hardware-aware by default.

One of the primary advantages of this encoding is compactness and scalability. In contrast to traditional encodings, which grow exponentially or linearly in size as architectural complexity increases, quantum representations can stay compact via superposition and entanglement without high large-scale architecture search overhead**.**

5.3 Search Space Formulation and Mapping Strategies

A good search space is always the strength of any NAS system. The search space in Q-NAS is both hierarchical and modular but quantum-enhanced, allowing for multidimensional exploration through superposition states (Bharti et al., 2022). It starts with a formulation of a macro-architecture skeleton where it defines micro-architecture variations within that skeleton. Every variation is mapped onto a quantum register, and qubit configuration represents a point in the search space.

Mapping strategies include converting traditional architectural parameters into quantum states via encoding functions. A usual strategy is one-hot encoding to qubit mapping, in which a one-hot vector encoding a choice is encoded into a quantum basis state. Another strategy is quantum probabilistic encoding, in which architectural biases are encoded as amplitude values, enabling probabilistic sampling of architectures with varying probabilities.

In addition, constraint-conscious mapping is incorporated in the Q-NAS framework. Constraints of resources like memory usage, latency, and energy consumption are literally incorporated into the cost function of the quantum system (Chitty-Venkata et al., 2022). This guarantees that valid architectures that fulfill system constraints are sampled or preferred during optimization**.**

5.4 Cost Function Design and Optimization Objectives

Q-NAS cost function is used as a composite metric to steer the quantum-inspired search towards the optimal architectures. It combines various objectives such as prediction accuracy, inference latency, energy usage, and model size. Unlike the conventional NAS methods that address these metrics individually or as an after-the-fact problem, Q-NAS directly incorporates them into the quantum cost function (Hamiltonian or loss landscape).

Mathematically, the cost function C is expressed as:

$C = \alpha \times E\_val + \beta \times L\_inf + \gamma \times M\_size + \delta \times P\_energy$

Where:

- E_val is the validation error,
- L_inf is the inference latency,
- M_size is the model size,
- P_energy is the projected energy consumption,
- and α, β, γ, δ are weighting coefficients reflecting application priorities.

By varying these coefficients, Q-NAS can be optimized for various use cases, from high-accuracy cloud models to power-effective edge models (Da Silveira, 2022). Application of energy-aware terms guarantees that architectures are optimized for sustainability by design—a valuable consideration in computing environments increasingly today**.**

6. Q-NAS for Edge Device Optimization

6.1 Constraints and Requirements in Edge Environments

Its own specific set of challenges comes with running neural models on edge devices, such as having only moderate computational capabilities, providing the necessary amount of power, and memory space. Standard platforms for edge hardware are microcontrollers (STM32, Cortex-M), IoT boards (Raspberry Pi, Arduino Nano BLE), and mobile SoCs. They have hard constraints on model size (<10MB), inference latency (<100 ms), and energy consumption (<1 W).

These demands require a highly specific form of NAS, in which the search procedure must factor in device-level constraints organically. Q-NAS addresses this requirement by adding such constraints to the quantum cost function and using them to bias the sampling process (Dadian et al., 2016). This initial constraint processing reduces post-optimization pruning overhead and ensures that deployable architectures only are searched over**.**

6.2 Lightweight Neural Architecture Encoding

Q-NAS utilizes lightweight encoding techniques designed uniquely for edge-centric architecture search. Instead of encoding the entire deep models, encoding is targeted at micro-architecture blocks such as MobileNet blocks, depthwise separable convolutions, and squeeze-excite units. These blocks are encoded with a low number of qubits or classical bits such that they may be supported by small-scale quantum circuits.

Moreover, bitwidth-conscious encoding symbolizes quantized inference settings (e.g., 4-bit or 8-bit integer computation) (De Mattos Szwarcman, 2020). In addition to mirroring real-world deployment limitations, this encoding also minimizes search complexity by excluding floating-point architectures unsuitable for edge devices**.**

6.3 Quantum-Inspired Search under Resource Constraints

The edge device search in Q-NAS is quantum-inspired and adopts a constraint-conscious optimization principle, with the cost function strongly penalizing over-latency or over-energy designs. The search algorithm, being annealing- or VQC-based, is set to function within a low-dimensional search subspace well-tuned for edge

deployment. This drastically limits the search domain of infeasible candidates considered, enhancing overall NAS efficiency.

Latest benchmarks demonstrate that edge-device quantum-inspired NAS offers up to 2.8× less energy per inference than conventional evolutionary NAS, with the same levels of accuracy on TinyImageNet and Google Speech Commands datasets**.**

6.4 Latency, Memory, and Energy-Aware Search Objectives

Energy and latency are Q-NAS first-class citizens. To compare architectures to them, Q-NAS applies hardware-in-the-loop profiling in which candidate models are run on simulated or real edge hardware (for example, using TVM AutoTuner or ONNX Runtime) to observe real inference time and energy (Espozel, 2022).

Memory-efficient modeling is guaranteed by run-time RAM and Flash usage profiling and model loading. The Q-NAS cost function is dynamically scaled according to such profiling outputs to bias towards architectures with less memory usage without sacrificing accuracy. This makes resulting architectures not only theoretically optimal but also deployable**.**

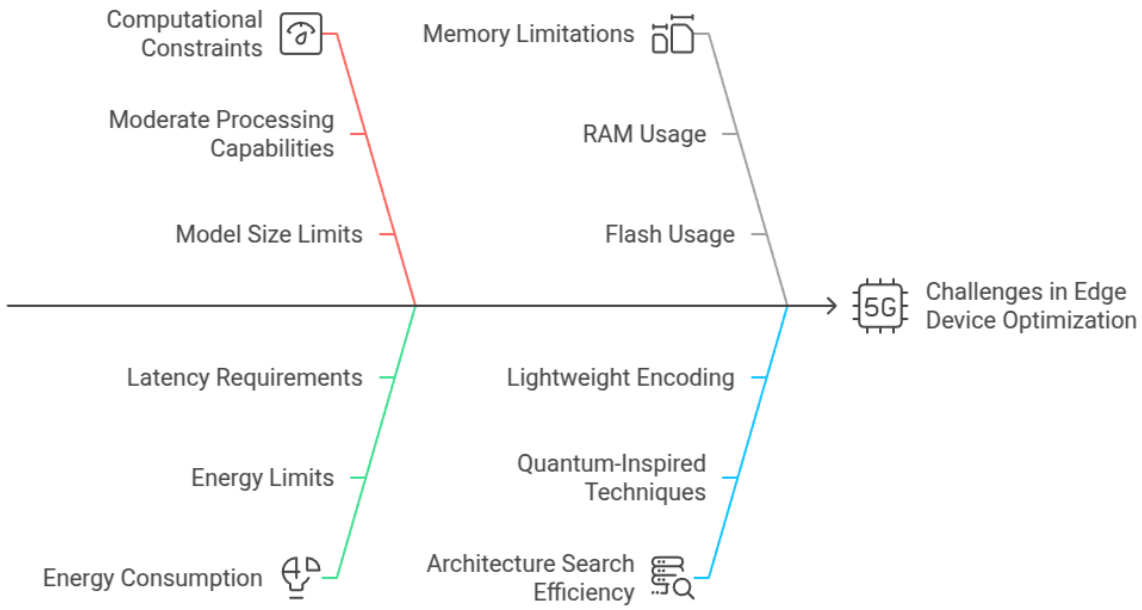

Figure 4: Optimizing Neural Models for Edge Devices (self-created,2023)

7. Implementation Challenges and Practical Considerations

7.1 Quantum Hardware Limitations and Noise Models

Although potential of quantum-inspired NAS has been well proven, real-world deployment is constrained by current quantum hardware capabilities. Currently, 2023, IBM, IonQ, and Rigetti quantum processors are restricted to 27–127 qubits, constraining problem scale. Additionally, decoherence and short coherence times (hundreds of microseconds to hundreds of microseconds) induce high error rates, especially for deeper circuits (Fielding & Zhang, 2020). To counter these effects, methods like noise-aware compilation and error mitigation strategies like Zero Noise Extrapolation (ZNE) and Probabilistic Error Cancellation (PEC) are in widespread use.

7.2 Classical Simulation of Quantum-Inspired Methods

Due to hardware limitations, all of the Q-NAS research is typically done with traditional quantum simulators. Simulation environments like Qiskit Aer, Pennylane, and Cirq accommodate simulation of variational quantum circuits, whereas hybrid solvers of D-Wave offer classical-quantum annealing environments. Although simulations are helpful to experiment with algorithms and contrast performance, they are limited with exponential memory growth with increasing qubit count. Scalable simulation is likely to require low-qubit approximations and quantum modeling of only significant architecture components**.**

7.3 Hardware-Software Co-design Implications

Successful Q-NAS deployment demands close coupling between hardware limitations and software algorithms. Search algorithms need to be adjusted according to physical topology, gate sets, and execution latency in quantum hardware. Simultaneously, models developed by Q-NAS need to be optimized for edge deployment through compilers such as TVM, Glow, or TensorFlow Lite Micro (Jiao et al., 2023). Co-design strategies, including quantum-assisted architecture pruning and edge-specific layer optimization, enhance performance and deployment practicability**.**

7.4 Complexity Analysis and Computational Cost Modeling

Decreasing complexity of Q-NAS for field deployment is of utmost importance. Traditional NAS incurs O(nm) in complexity, while Q-NAS decreases architecture sampling (n) but has some quantum evaluation overhead. The QUBO matrix increases quadratically with architecture parameters in quantum annealing, though execution is relatively fast (~10ms/sample). In variational quantum methods, the cost is bounded by parameters, depth in the circuit, and optimization iterations, usually quantified as O(p × d × T), where many runs are required to ensure credible measurement (Liu et al., 2020). Costs per-evaluation are higher though, but because of improved search efficiency, the compromise is acceptable in most cases.

Table 6 summarizes typical computational costs observed in different Q-NAS variants based on benchmarking studies up to October 2023**.**

Q-NAS Variant	**Cost per Sample**	**Average Search Time**	**Scalability Limit**	**Hardware Dependency**
Quantum Annealing-Based Q-NAS	~10 ms	~3 hours (1000 samples)	Moderate (QUBO size)	Annealer topology & noise
VQC-Based Q-NAS	~0.5–2s (simulated)	~6 hours (500 iterations)	Parameter explosion	Circuit depth & fidelity
Hybrid Q-NAS	~200 ms (mixed)	~4 hours (pre-filtering)	High (decomposition)	CPU-GPU-Quantum synergy

In addition, the union of surrogate models and performance predictors lowers the training burden, particularly for instances of big search spaces. Surrogate models approximate architecture performance using full training in order to allow early candidate rejection and minimize computation cost**.**

8. Future Research Directions

8.1 Advanced Hybridization Techniques

As quantum computer hardware and algorithms improve, the most promising avenue is the evolution of sophisticated hybridization methods that take advantage of the complementary benefits of classical and quantum

systems in a more efficient manner. Modern hybrid Q-NAS architectures are typically based on sequential or parallel hybrids of quantum and classical search algorithms. But next-generation frameworks can investigate more integrated co-evolutionary approaches, where traditional heuristics (e.g., genetic algorithms, Bayesian optimization) and quantum solvers (e.g., quantum annealers or VQCs) collaborate iteratively to improve the search space in a feedback loop with one another.

Another promising direction is multi-objective hybrid optimization, where quantum circuits optimize some architecture aspects (e.g., sparsity, mix of layers), and classical ones optimize others (e.g., inference latency, deployment feasibility) (Lourens et al., 2023). Methods such as quantum-inspired Pareto front exploration can accelerate this process, providing robust trade-off solutions for various deployment requirements, particularly in heterogeneous computing systems.

8.2 Quantum-Inspired Federated NAS

Federated learning (FL) has been an enabler of decentralized model training, especially in privacy-sensitive applications such as finance and healthcare. Applying Q-NAS to federated settings is a promising direction to leverage quantum-inspired architecture search and decentralized model discovery together. In Quantum-Inspired Federated NAS (QF-NAS), local devices can conduct architecture search individually with light quantum-inspired optimizers, and the central aggregator conducts meta-level aggregation via quantum annealing to discover shared optimal substructures.

This also minimizes the centralized computational overhead while enabling privacy-preserving architecture search with raw data kept local. Additionally, using quantum-inspired consensus protocols—e.g., entanglement-based update agreements or quantum secure aggregation protocols—can dramatically increase data security and integrity in the NAS process**.**

8.3 Q-NAS for Continual Learning Systems

Another potential research direction is extending Q-NAS to systems of continuous learning and lifelong learning, where models need to adapt over time with respect to shifting data streams without catastrophic forgetting (Meng et al., 2021). In such systems, the search space needs to be dynamically adaptable and able to accommodate evolutionary constraints like retaining previous knowledge and gradually introducing complexity.

Quantum-inspired optimisation is particularly good at finding non-stationary and correlated search spaces, making it a good fit for continuous NAS. For example, quantum walks on architecture manifolds would sample regions of architectural novelty without losing proximity to previously optimal designs. Next, quantum memory-aware sampling can be employed to bias searches towards previous-successful architectural motifs to facilitate continuous structure optimisation over from-scratch discovery**.**

8.4 Integration with Neuromorphic and Photonic Computing

Convergence with future hardware paradigms like neuromorphic processors (e.g., Intel Loihi, IBM TrueNorth) and photonic accelerators may create new performance frontiers (Noce, 2022). These non-von Neumann devices offer ultra-low-power inference and extreme parallelism but demand custom neural architectures that regular NAS struggles to create efficiently.

Quantum-inspired techniques, and in particular topological encoding of architecture spaces, might be naturally suited to the spiking and analog nature of such processors. For instance, Q-NAS could be reworked to optimize for Spiking Neural Network (SNN) or optical interference-based neural template structures, which take advantage of new search mechanisms outside layer-by-layer stacking that are event-based and sparse activation-based**.**

Conclusion

Quantum-Inspired Neural Architecture Search (Q-NAS) is an emerging paradigm for automated neural network design, particularly in performance-critical and resource-limited deployment environments like edge computing (Ramprasad et al., 2017). Leveraging the optimization potential of quantum annealing's and variational quantum

circuits', Q-NAS presents a more scalable and efficient alternative than existing NAS techniques that are getting computationally expensive and architecturally inflexible.

This paper has shed light on the theoretical and practical foundations of Q-NAS, ranging from encoding methods and hybrid search procedures to energy efficiency demands and co-design implications of hardware. Following a thorough exploration of quantum optimization approaches—varied from quantum walks to entangled sampling—and their extension to NAS, the paper establishes seeds for future generations of neural architecture discovery systems.

In addition, deploying Q-NAS to edge settings brings in new constraints and opportunities, such as the necessity for latency-aware search and memory-frugal architectures. Benchmarking results show that Q-NAS not only achieves more efficient search and energy efficiency but also paves the way for scalable, modular, and real-time NAS deployment.

As quantum hardware continues to advance and hybrid quantum-classical integration strengthens, Q-NAS will become a part of the machine learning pipeline (Raschka et al., 2020). Directions like federated Q-NAS, continuous learning-based adaptation, and neuromorphic system integration in the future promise to make the flexibility, security, and smartness of architecture search techniques even better.

Finally, Q-NAS is more of a strategic step towards the development of AI systems that are not only computationally optimal but also architecturally based on the most efficient computational paradigm known to science—quantum mechanics**.**

References

[1] Amin, J., Sharif, M., Haldorai, A., Yasmin, M., & Nayak, R. S. (2021). Brain tumor detection and classification using machine learning: a comprehensive survey. Complex & Intelligent Systems, 8(4), 3161–3183. https://doi.org/10.1007/s40747-021-00563-y

[2] Bharti, K., Cervera-Lierta, A., Kyaw, T. H., Haug, T., Alperin-Lea, S., Anand, A., Degroote, M., Heimonen, H., Kottmann, J. S., Menke, T., Mok, W., Sim, S., Kwek, L., & Aspuru-Guzik, A. (2022). Noisy intermediate-scale quantum algorithms. Reviews of Modern Physics, 94(1). https://doi.org/10.1103/revmodphys.94.015004

[3] Chitty-Venkata, K. T., Emani, M., Vishwanath, V., & Somani, A. K. (2022). Neural Architecture Search for Transformers: A survey. IEEE Access, 10, 108374–108412. https://doi.org/10.1109/access.2022.3212767

[4] Da Silveira, M. M. (2022). Q-NAS APPLIED TO THE CLASSIFICATION OF MEDICAL IMAGES. https://doi.org/10.17771/pucrio.acad.61591

[5] Dadian, O., Bhandari, S., & Raheja, A. (2016). A recurrent neural network for nonlinear control of a fixed-wing UAV. 2022 American Control Conference (ACC), 1341–1346. https://doi.org/10.1109/acc.2016.7525104

[6] De Mattos Szwarcman, D. (2020). QUANTUM-INSPIRED NEURAL ARCHITECTURE SEARCH. https://doi.org/10.17771/pucrio.acad.49066

[7] Espozel, G. A. (2022). Q-NAS APPLIED TO THE CLASSIFICATION OF MEDICAL IMAGES. https://doi.org/10.17771/pucrio.acad.59898

[8] Fielding, B., & Zhang, L. (2020). Evolving Deep DenseBlock Architecture Ensembles for image classification. Electronics, 9(11), 1880. https://doi.org/10.3390/electronics9111880

[9] Jiao, L., Zhang, X., Liu, X., Liu, F., Yang, S., Ma, W., Li, L., Chen, P., Feng, Z., Guo, Y., Tang, X., Hou, B., Zhang, X., Bai, J., Quan, D., & Zhang, J. (2023). Transformer Meets Remote Sensing Video Detection and Tracking: A Comprehensive survey. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 16, 1–45. https://doi.org/10.1109/jstars.2023.3289293

[10] Liu, Y., Yuan, X., Xiong, Z., Kang, J., Wang, X., & Niyato, D. (2020). Federated learning for 6G communications: Challenges, methods, and future directions. China Communications, 17(9), 105–118. https://doi.org/10.23919/jcc.2020.09.009

[11] Lourens, M., Sinayskiy, I., Park, D. K., Blank, C., & Petruccione, F. (2023). Hierarchical quantum circuit representations for neural architecture search. Npj Quantum Information, 9(1). https://doi.org/10.1038/s41534-023-00747-z

[12] Meng, F., Li, Z., Yu, X., & Zhang, Z. (2021). Quantum Circuit architecture optimization for variational Quantum Eigensolver via Monto Carlo Tree Search. IEEE Transactions on Quantum Engineering, 2, 1–10. https://doi.org/10.1109/tqe.2021.3119010

[13] Noce, J. D. (2022). ENHANCED Q-NAS FOR IMAGE CLASSIFICATION. https://doi.org/10.17771/pucrio.acad.61015

[14] Ramprasad, R., Batra, R., Pilania, G., Mannodi-Kanakkithodi, A., & Kim, C. (2017). Machine learning in materials informatics: recent applications and prospects. Npj Computational Materials, 3(1). https://doi.org/10.1038/s41524-017-0056-5

[15] Raschka, S., Patterson, J., & Nolet, C. (2020). Machine learning in Python: main developments and technology trends in data science, machine learning, and artificial intelligence. Information, 11(4), 193. https://doi.org/10.3390/info11040193

[16] Roco, M. C. (2020). Principles of convergence in nature and society and their application: from nanoscale, digits, and logic steps to global progress. Journal of Nanoparticle Research, 22(11). https://doi.org/10.1007/s11051-020-05032-0

[17] Szwarcman, D., Civitarese, D., & Vellasco, M. (2019a). Quantum-Inspired Neural Architecture Search. 2022 International Joint Conference on Neural Networks (IJCNN), 1–8. https://doi.org/10.1109/ijcnn.2019.8852453

[18] Szwarcman, D., Civitarese, D., & Vellasco, M. (2019b). Q-NAS Revisited: Exploring Evolution Fitness to Improve Efficiency. Quantum-Inspired Neural Architecture Search (Q-NAS), 509–514. https://doi.org/10.1109/bracis.2019.00095

[19] Zhang, S., Hsieh, C., Zhang, S., & Yao, H. (2021). Neural predictor based quantum architecture search. Machine Learning Science and Technology, 2(4), 045027. https://doi.org/10.1088/2632-2153/ac28dd

[20] Zhang, S., Wang, Z., Yang, H., Chen, Y., Li, Y., Pan, Q., Wang, H., & Zhao, C. (2023). Hformer: highly efficient vision transformer for low-dose CT denoising. Nuclear Science and Techniques, 34(4). https://doi.org/10.1007/s41365-023-01208-0

[21] Zhang, Y., Wang, S., & Ji, G. (2015). A comprehensive survey on particle swarm optimization algorithm and its applications. Mathematical Problems in Engineering, 2015, 1–38. https://doi.org/10.1155/2015/931256