Article

Not peer-reviewed version

QHM: Unifying Superconducting and Topological Quantum Computing with Multimodal AI

Vikram Karlex *

Posted Date: 3 April 2025

doi: 10.20944/preprints202504.0232.v1

Keywords: quantum computing; artificial intelligence; superconducting qubits; topological qubits; multimodal AI; quantum hybrid models; quantum neural networks; quantum self-attention; QAOA; quantum algorithms; AI acceleration; error mitigation; quantum advantage; neurosymbolic

 doi:10.20944/preprints202504.0232.v1

Article

QHM: Unifying Superconducting and Topological Quantum Computing with Multimodal AI

Vikram Karlex

Department of AGI Research, KarLex AI, Inc., Delaware, USA; vikram@karlex.ai

Abstract: Artificial Intelligence (AI) has achieved remarkable successes, but faces challenges in efficiency, interpretability, robustness, alignment, and energy consumption [15–19]. Quantum computing offers a path to fundamentally accelerate and enhance AI by exploiting quantum parallelism and entanglement [22,23]. This paper proposes QHM- Quantum Hybrid Multimodal, a unified framework that integrates superconducting and topological quantum computing into multimodal AI architectures. We derive a theoretical foundation for embedding quantum subroutines (e.g., a Quantum Self-Attention Neural Network QSANN [30] and Quantum QAOA QQAOA [47]) within classical deep learning. We survey quantum algorithms – including Grover's search [24], HHL for linear systems [25], quantum approximate optimization (QAOA) [47], and variational quantum circuits [31] – analyzing their computational complexity and suitability for AI tasks. We then discuss both superconducting qubit platforms (Google's 105-qubit Willow [7], IBM's 1,121-qubit Condor [28], Amazon's bosonic Ocelot [5]) and Microsoft's topological Majorana 1 [1] in the context of AI acceleration. We explore how quantum resources can improve large language models, Transformers [62], mixture-of-experts, and cross-modal learning [63] via quantum-accelerated similarity search, attention mechanisms, and optimization. Practical engineering challenges such as cryogenic cooling [35], control electronics, qubit noise, error correction [23], and data encoding overhead [55] are examined with a cost-benefit analysis. We outline an implementation roadmap from classical simulations to hybrid prototypes to full integration, with proposed benchmarks to evaluate quantum-AI performance against classical baselines. Compared to conventional approaches, the hybrid QHM framework promises improved computational scaling and new capabilities (e.g., faster search, more efficient training), while noting current trade-offs in noise and infrastructure. We conclude with future directions for developing quantum-enhanced AI that is more efficient, interpretable, and aligned with human values, and discuss broader implications for AI safety and sustainability [21].

Keywords: quantum computing; artificial intelligence; superconducting qubits; topological qubits; multimodal AI; quantum hybrid models; quantum neural networks; quantum self-attention; QAOA; quantum algorithms; AI acceleration; error mitigation; quantum advantage; neurosymbolic

1. Introduction

1.1. Motivation for a Quantum Hybrid AI Framework

AI systems today achieve state-of-the-art performance in language, vision, and decision-making tasks, yet they encounter fundamental challenges:

- **Efficiency and Scalability:** Training large models demands enormous computational power and energy. For example, the GPT-3 family required petaflop/s-years of compute and significant energy resources [15,16]. In deployment, serving millions of queries can consume power on the order of a small country's electricity usage [21].
- **Interpretability:** Deep neural networks often operate as black boxes, making it difficult to explain their decisions [17]. This opaqueness raises concerns in high-stakes applications.
- **Robustness:** AI models can be brittle, vulnerable to adversarial examples or distribution shifts, which undermines reliability [18].

 doi:10.20944/preprints202504.0232.v1

- **Alignment:** There is a growing focus on ensuring that AI goals and behaviors remain aligned with human values and intentions [19]. Neurosymbolic approaches, which combine neural networks with symbolic logic, have been proposed to improve transparency and alignment [20].
- **Energy Constraints:** The environmental footprint of AI is significant; data centers running large AI workloads have skyrocketing energy emissions [21].

These challenges motivate exploring new paradigms to augment classical AI.

Quantum computing has emerged as a promising avenue to address some of these challenges [22,23]. Quantum computers leverage quantum-mechanical phenomena to process information in ways classical computers cannot, offering the potential for exponential speedups in certain algorithms. For AI, quantum computing could drastically improve computational efficiency, enabling faster training and inference for complex models.

Quantum algorithms also provide novel capabilities: for example, Grover's quantum search [24] can search an unsorted database in $O(\sqrt{N})$ steps versus $O(N)$ classically, which could accelerate pattern matching and retrieval in AI systems. Likewise, the Harrow-Hassidim-Lloyd (HHL) algorithm [25] can solve structured linear systems in logarithmic time under certain conditions, potentially speeding up components of machine learning like linear regression or Gaussian processes. Beyond speed, quantum computing's high-dimensional Hilbert spaces might naturally represent complex data structures and probability distributions, offering more compact or expressive models for multimodal data [26].

This paper explores a unifying framework, QHM (Quantum Hybrid Multimodal), that integrates quantum computing subroutines into AI architectures to tackle these challenges. We focus on both superconducting quantum computing (which underpins many current quantum processors) and topological quantum computing (which aims to achieve more stable qubits via exotic quasiparticles).

Section 2 provides a theoretical framework, detailing how quantum subroutines can be mathematically embedded into neural network pipelines and introducing the notion of quantum-enhanced neurosymbolic AI for alignment and verification. In Section 3, we derive and discuss key quantum algorithms relevant to AI – including a proposed Quantum Self-Attention Neural Network (QSANN), a Quantum-enhanced QAOA (QQAOA) for optimization, the HHL algorithm, and Grover's algorithm – analyzing their complexity and potential impact on AI tasks.

Section 4 then surveys quantum hardware advances, from Google's 105-qubit Willow superconducting chip [7] (which achieved a milestone by performing in minutes a computation that would take classical supercomputers 10^{25} years) to IBM's 1121-qubit Condor [28] and Amazon's bosonic cat-qubit Ocelot [5], as well as Microsoft's Majorana-based topological qubit approach [1]. We evaluate how quantum circuits and hybrid computing models can be implemented on current devices and anticipated future hardware.

Section 5 turns to Multimodal AI, examining large language models and vision-language models (transformers [62], mixture-of-experts, etc.) and identifying components like similarity search, attention weighting, and optimization that could be enhanced by quantum computing. We propose mechanisms such as quantum-accelerated nearest-neighbor search for embeddings and quantum sub-circuits that perform attention or gating with potentially higher speed or capacity. We also introduce the idea of dynamic neural architectures that integrate quantum processing units as co-processors for specific tasks (e.g., a quantum module that computes a kernel or Fourier transform as part of a classical network).

Section 6 discusses the practical engineering constraints: superconducting qubits require cryogenic cooling to millikelvin temperatures [35], imposing non-trivial energy and hardware overhead; quantum control electronics and microwave interface pose complexity; noise and decoherence currently limit circuit depth, necessitating error mitigation and correction techniques [11]. We also analyze the overhead of data encoding – classical-to-quantum data conversion can be expensive [55] – which may offset theoretical speedups if not carefully managed. A cost-benefit analysis is provided to

 doi:10.20944/preprints202504.0232.v1

compare when a quantum-enabled approach becomes advantageous over purely classical computation, considering metrics like computational complexity, latency, energy, and development cost.

In Section 7, we outline an implementation roadmap for QHM. In the near term, classical simulations of quantum circuits (using libraries like Qiskit or PennyLane [31]) can be integrated into AI workflows to prototype quantum components. Next, small-scale experiments on quantum hardware (with tens of qubits) can demonstrate hybrid quantum-classical training – for instance, using a quantum circuit to compute part of a model's forward pass and a classical optimizer to adjust parameters, a workflow akin to variational quantum algorithms. As hardware scales to hundreds then thousands of qubits with improving fidelity, more of the AI model can be offloaded to quantum subroutines, moving towards a truly hybrid system.

Section 8 provides a comparative analysis with existing approaches: how does a QHM-based system perform against purely classical neural networks or other hybrid frameworks? We discuss known results, e.g., certain quantum classifiers vs classical ones [30,38], and where quantum advantages are most evident or, conversely, where classical methods still excel (taking into account that today's classical AI is very mature and benefits from massive data and tuning). We examine trade-offs, such as the added complexity of managing a quantum co-processor and the current limitation on problem sizes that near-term quantum devices can handle.

Finally, Section 9 concludes with key findings and an outlook. We summarize that quantum computing, while not a panacea, offers compelling tools to push AI beyond current limits: it can potentially provide speedups in searching, sampling, and optimization; it introduces new ways to represent and process information that could make models more compact or powerful; and in the long run, as fault-tolerant quantum computers emerge, it might allow AI to solve problems that are completely infeasible classically (such as simulating complex quantum systems for scientific discovery).

1.2. Relationship to Theoretical Foundations

While this paper examines quantum hardware implementation for multimodal AI, the technological trajectory described has profound implications for artificial general intelligence (AGI) [19,23]. The progression from current noisy devices toward fault-tolerant quantum systems capable of complex AI operations may address computational barriers to AGI through quantum advantages in search, optimization, and representation learning. The hybrid classical-quantum architectures and phased implementation roadmap we've outlined could serve as a template for responsible AGI development—balancing aspirational capabilities with realistic technological assessment. Future research will investigate whether certain AGI bottlenecks might be fundamentally quantum in nature, how quantum hardware advances might accelerate or alter AGI development timelines, and how the implementation challenges identified might inform safe and beneficial AGI deployment strategies.

2. Theoretical Framework

Integrating quantum computation into AI requires a rigorous theoretical foundation. In this section, we develop the mathematical framework for quantum-classical hybrid architectures that underpin QHM. We consider a generic AI model (such as a deep neural network $f(\mathbf{x};\boldsymbol{\theta})$ with inputs $\mathbf{x}$ and parameters $\boldsymbol{\theta}$) and enhance it with quantum subroutines. Formally, we can define a hybrid function:

$$g(\mathbf{x};\boldsymbol{\theta},\boldsymbol{\phi}) = f_{\text{classical}}(\mathbf{x};\boldsymbol{\theta}, h_{\text{quantum}}(\mathbf{x};\boldsymbol{\phi})) \tag{1}$$

where $h_{\text{quantum}}(\mathbf{x};\boldsymbol{\phi})$ is a function computed by a parameterized quantum circuit (PQC) with quantum parameters $\boldsymbol{\phi}$ (which may correspond to angles of rotation gates, for example) [31]. In other words, part of the computation of g is delegated to a quantum module h_{quantum}, whose output feeds into the classical network. This structure is akin to a variational quantum circuit or quantum layer within a neural network. The classical parameters $\boldsymbol{\theta}$ and quantum parameters $\boldsymbol{\phi}$ can be trained jointly

 doi:10.20944/preprints202504.0232.v1

using hybrid quantum-classical optimization (e.g., evaluating gradients via parameter-shift rules on the quantum circuit and backpropagation through the classical part) [33].

A critical aspect of this framework is the data encoding or feature mapping from classical data to quantum states [50]. One common approach is to use an encoder quantum circuit $U_{\text{enc}}(\mathbf{x})$ that maps input $\mathbf{x}$ to a quantum state $|\psi_{\mathbf{x}}\rangle$ in an n-qubit Hilbert space. For example, if $\mathbf{x}$ is a vector of real numbers, we can encode it in the amplitudes of a quantum state (amplitude encoding), or apply rotations proportional to components of $\mathbf{x}$ (angle encoding). The ZZ-feature map is one such encoding used in quantum kernel methods. Formally, $U_{\text{enc}}(\mathbf{x})|0\rangle^{\otimes n} = |\psi_{\mathbf{x}}\rangle$.

After encoding, the quantum layer applies a sequence of parameterized gates $U(\boldsymbol{\phi})$ and measurements to produce some output $h_{\text{quantum}}(\mathbf{x}; \boldsymbol{\phi})$. In hybrid algorithms like the Variational Quantum Eigensolver (VQE) or Quantum Approximate Optimization Algorithm (QAOA) [47], this output might be an expectation value $\langle\psi_{\mathbf{x}}|H|\psi_{\mathbf{x}}\rangle$ for some Hamiltonian H, but in an AI context it could be a predicted scalar or vector fed into the larger model.

2.1. Quantum Self-Attention Neural Network (QSANN)

QSANN is a theoretical construct where the self-attention mechanism in, say, a Transformer model [62], is implemented or augmented by quantum operations. Recall that in a classical Transformer, an attention score is computed as $\text{score}(q, k) = \frac{\langle q,k\rangle}{\sqrt{d}}$ for query and key vectors q, k, and these scores go through a softmax to weigh value vectors.

In a QSANN, we envision representing query and key vectors as quantum states (or quantum amplitude encodings) and using quantum inner product estimation to compute similarities more efficiently in high dimension [30]. For instance, given quantum states $|q\rangle$ and $|k\rangle$ encoding the respective vectors, the overlap $|\langle q|k\rangle|$ can be estimated via a SWAP test or Hadamard test on a quantum computer. The complexity can in principle be independent of the dimension d (as the inner product is a single operation on superposed amplitudes), whereas classical computation of $\langle q, k\rangle$ scales as $O(d)$.

By using amplitude amplification (related to Grover's algorithm [24]), relevant keys could be retrieved in sublinear time with respect to the database size. Mathematically, if classical attention computes:

$$\alpha_{ij} = \frac{\exp\left(\frac{q_i \cdot k_j}{\sqrt{d}}\right)}{\sum_{j'} \exp\left(\frac{q_i \cdot k_{j'}}{\sqrt{d}}\right)} \tag{2}$$

a quantum version might prepare a superposition over key-value pairs and use quantum amplitudes to encode these weights, modifying the softmax normalization to a quantum state normalization [52]. The QSANN concept also involves quantum state attention: using entangled states to represent correlations between modalities (e.g., an entangled state between an image patch and a word token could capture cross-modal associations inherently) [50].

2.2. Quantum QAOA (QQAOA)

QQAOA refers to leveraging QAOA not just as a standalone quantum algorithm for combinatorial optimization, but embedded within an AI model for decision-making or structured prediction. QAOA traditionally prepares a parameterized state for solving an optimization problem such as MAX-CUT [47]. The QAOA state at p layers is:

$$|\psi(\gamma, \beta)\rangle = \prod_{j=1}^{p} (e^{-i\beta_j H_M} e^{-i\gamma_j H_C})|s\rangle \tag{3}$$

where H_C encodes the cost function (problem Hamiltonian) and H_M is a mixing Hamiltonian (often a sum of single-qubit X gates), and $|s\rangle$ is some initial state (often uniform superposition). The

 doi:10.20944/preprints202504.0232.v1

variational parameters γ, β are optimized to drive the state towards one encoding the optimal solution [44].

In our hybrid framework, we can imagine QQAOA modules that solve sub-problems within a larger task. For example, an AI planner might have to solve a combinatorial subtask (like routing or assignment) as part of its reasoning; a QAOA module could provide a quantum boost for that sub-problem, returning an optimized configuration that the AI agent then uses in its overall policy. The term "QQAOA" also hints at meta-optimization: using quantum circuits to even improve the QAOA itself, or a two-level QAOA where one quantum routine optimizes parameters for another [51].

2.3. Computational Complexity Considerations

In developing these hybrid constructs, one must also consider computational complexity and how quantum steps alter the overall complexity of the AI model. Suppose a classical model has time complexity $T_C(N)$ for a dataset of size N (or input of size N). If we replace a component with a quantum algorithm of complexity $T_Q(N)$ (for equivalent task), the hybrid complexity might become:

$$T_C(N) - T_{\text{classical\_part}}(N) + T_Q(N) + T_{\text{interface}}(N) \tag{4}$$

Here $T_{\text{interface}}$ accounts for overhead such as preparing quantum inputs and reading quantum outputs. In the ideal scenario, $T_Q(N)$ is asymptotically smaller than $T_{\text{classical\_part}}(N)$, and $T_{\text{interface}}(N)$ is not too large, so a net speedup results.

As an example, consider searching a database of N candidate features for the nearest neighbor to a query (common in similarity-based learning). Classically this is $O(N)$, but Grover's algorithm [24] can find a marked item in $O(\sqrt{N})$ steps. If the overhead to load data into the quantum memory is, say, $O(N)$ (naively), the advantage may be lost. Research in quantum RAM (qRAM) suggests ways to load data in sublinear time or prepare superpositions more efficiently, which would reduce $T_{\text{interface}}$ [55].

Similarly, HHL solves linear equations in $O(\log N)$ time (for an $N \times N$ system) under assumptions of sparsity and condition number [25], whereas classical methods (conjugate gradient) are $O(N \cdot \text{sparsity} \cdot \kappa)$. If one can feed an AI model's linear system (like a kernel ridge regression or an attention matrix inversion) to HHL, the potential exponential speedup is enormous – but only if one can efficiently provide the matrix as input and extract the solution or relevant function of the solution.

We explicitly incorporate such considerations in our framework: each quantum subroutine is analyzed in terms of input encoding cost, quantum execution cost, and output extraction cost. This forms the basis for deciding where in an AI pipeline a quantum module is beneficial.

2.4. Neurosymbolic AI and Quantum Integration

Neurosymbolic AI attempts to combine the learning capability of neural networks with the explicit knowledge representation of symbolic logic, to achieve better interpretability and verifiability [20,64]. In our QHM framework, we propose that quantum computing could assist neurosymbolic AI in two ways:

1. By efficiently exploring large discrete state spaces for reasoning tasks using quantum search or optimization [24]. For instance, searching a logical rule space or performing logical inference might be sped up by Grover's algorithm or quantum backtracking algorithms.
2. By enabling novel representational capacities – a quantum state can, for example, represent a superposition of many logic propositions, and quantum operations can implement certain logical operations in parallel. We might imagine a quantum circuit that encodes a truth table of a learned rule and verifies properties against it much faster than exhaustive classical checks [26].

Moreover, quantum algorithms for satisfiability or constraint solving could be used to verify neural network behaviors against formal specifications (a core part of alignment verification) [46]. While this is largely conceptual at present, we provide a concrete thought experiment: verifying safety properties of an AI controller could be mapped to checking certain constraints; a quantum SMT (satisfiability modulo theory) solver might check these constraints more rapidly, or sample

 doi:10.20944/preprints202504.0232.v1

counterexamples if any exist. This aligns with the broader goal of AI alignment – ensuring AI systems do what we intend [19]. A neurosymbolic approach using concept-based explanations could benefit from quantum subroutines to identify relevant concepts or validate symbolic explanations against the neural state.

2.5. Comparison of Classical and Quantum Paradigms

To ground the discussion, Table 1 provides a comparison of classical vs. quantum paradigms for several key operations in AI, which we will reference throughout the paper [22–25]. This highlights where quantum computing might offer exponential or quadratic improvements, and where classical methods currently still hold an edge due to maturity or lower overhead.

Table 1. Classical vs Quantum Computing Paradigms for AI-Relevant Operations

Operation	**Classical Complexity / Features**	**Quantum Complexity / Features**
Unstructured Search	$O(N)$ linear search	$O(\sqrt{N})$ Grover's algorithm (quadratic speedup) [24]
Structured Search (e.g. BST)	$O(\log N)$ for balanced tree	$O(\log N)$ (no known speedup if structure can be exploited classically)
Dense Matrix-Vector Multiply	$O(N^2)$ for $N \times N$ matrix	$O(N)$ with quantum amplitude encoding (exponential state space) but $O(N)$ data loading [50]
Solving Linear System (s-sparse)	$O(N \cdot s \cdot \kappa \log(1/\epsilon))$	$O(\log N \cdot s^2\kappa^2/\epsilon)$ (HHL algorithm, exponential speedup in N) [25]
Distance / Inner Product	$O(d)$ for vectors in $\mathbb{R}^d$	$O(1)$ with quantum state overlap (assuming states prepared) [30]
Fourier Transform	$O(N \log N)$ (FFT)	$O(\log N)$ (quantum FFT, exponential in signal length)
Search in database of size N	$O(\log N)$ if sorted (binary search) / $O(N)$ if unsorted	$O(\sqrt{N})$ (Grover) unsorted [24]; no gain if sorted (need classical sort)
Sampling from distribution	Depends on distribution (often $O(N)$)	Quantum sampling can leverage superposition to sample in one step from prepared distribution [57]
Optimization (generic)	No general polytime method for NP-hard problems (often exponential)	Quadratic speedup in exhaustive search (Grover) [24]; QAOA heuristic [47] (depends on problem, no guaranteed global speedup in worst-case)
Data Encoding Overhead	Not applicable (data is in memory directly)	May require $O(N)$ operations to load N amplitudes (quantum RAM can reduce this if available) [55]
Robustness to Noise	Classical bits stable (error correcting codes rarely needed at hardware level)	Qubits fragile, require quantum error correction (overhead in qubit count and operations) [23]

This table underscores that quantum advantages often come with caveats – notably the data encoding and noise issues. In the following sections, we dive deeper into specific algorithms (Sec. 3) and the hardware that executes them (Sec. 4), keeping in mind these theoretical considerations.

 doi:10.20944/preprints202504.0232.v1

3. Quantum Computing for AI

Quantum computing for AI encompasses a spectrum of approaches, from using quantum accelerators for specific subroutines to designing entirely new quantum algorithms inspired by machine learning [26]. We cover both superconducting and topological quantum computing platforms, as each offers unique strengths for AI integration. We then examine how quantum circuits can be incorporated into hybrid models and discuss the feasibility of implementing these ideas on current hardware.

3.1. Superconducting Quantum Processors for AI

Superconducting qubits, implemented as small Josephson junction circuits on chips, are among the most advanced quantum computing technologies today. Companies like IBM, Google, and Amazon have been steadily scaling up superconducting processors.

Google's latest chip, Willow, contains 105 qubits and has demonstrated the ability to perform random circuit sampling far beyond classical reach [7]. In fact, Willow achieved a benchmark where it executed a task in under 5 minutes that would take a top supercomputer on the order of 10^{25} years, a dramatic demonstration of quantum computational power (often described as "quantum supremacy" or "beyond-classical" regime) [27]. This task – random circuit sampling – doesn't have direct practical application yet, but it validates that a medium-scale quantum device can harness massive parallelism by exploring 2^{105} states simultaneously. For AI, this hints that certain probabilistic computations (like sampling from a high-dimensional distribution) could be done exponentially faster by similar quantum circuits. Google's roadmap explicitly aims at using such chips for useful applications in fields including AI.

IBM's approach has emphasized increasing qubit count and quality in a modular way. In 2023, IBM unveiled Condor, a 1,121-qubit superconducting processor – the first to break the 1000-qubit barrier [28]. Condor uses IBM's transmon qubits with cross-resonance gates and achieved a 50% greater qubit density than its 433-qubit predecessor (Osprey). Such density improvements are crucial for scaling, as they reduce wire routing complexity in the dilution refrigerator (IBM had to develop a "super-fridge" to house Condor).

For AI applications, the large qubit count could allow bigger quantum subcircuits – potentially enough to encode medium-sized data or perform non-trivial quantum neural networks. However, current coherence times and gate fidelities mean that not all 1121 qubits can yet be fully entangled or operated on arbitrarily deep circuits without error. IBM's strategy involves dynamic circuits and error mitigation: using techniques like zero-noise extrapolation and probabilistic error cancellation to get useful results from noisy hardware. They envision quantum processors working in tandem with classical processors in a heterogeneous computing environment, which aligns perfectly with our hybrid QHM framework.

Amazon's Ocelot chip represents an alternative superconducting approach using bosonic encodings (cat qubits) [5]. Announced in 2025, Ocelot incorporates 5 data qubits (realized as Schrödinger's cat states in superconducting cavities) along with error-correcting and buffering circuits. The key idea is that cat qubits are naturally more robust against certain errors by storing information in special superpositions of harmonic oscillator states [29]. Amazon reported that Ocelot's architecture can reduce the overhead of quantum error correction by up to 90%. This is highly relevant for AI workloads: if error rates are lower, one can execute deeper or more complex circuits (which an AI algorithm might need for meaningful computation) without faults.

Ocelot's debut marks the first instance of a scalable cat qubit system, which could pave the way for error-corrected qubit networks earlier than other approaches. For QHM, a chip like Ocelot could eventually allow continuously running quantum co-processors that maintain quantum states throughout an AI model's operation (rather than resetting after each short circuit), since bosonic qubits can have longer lifetimes. The AWS Ocelot quantum processor, which physically consists of multiple interconnected modules in a refrigerator.

 doi:10.20944/preprints202504.0232.v1

The quantum circuits that run on these superconducting platforms for AI tasks can vary widely. A few promising categories include:

- **Quantum Kernels and Feature Maps:** A quantum circuit can act as a feature transformer, mapping input data to a quantum state that is implicitly a high-dimensional feature vector. Measuring overlaps between states computes a kernel [30]. This has been explored for small-scale classification, where quantum kernel methods showed some advantages on synthetic data. In a multimodal setting, one might have a quantum feature map for images and another for text, then use a classical model to fuse them.
- **Quantum Optimizers:** Variational quantum algorithms like QAOA [47] or VQE can serve as optimizers for specific layers. For example, one could encode the loss function of a neural network layer into a cost Hamiltonian and use QAOA to find optimal weights (this is theoretical at this stage). Alternatively, a quantum circuit could solve a lower-level optimization (like allocating resources in a scheduling problem that an AI agent needs to solve as part of its tasks).
- **Quantum Sampling and Generative Models:** Generative AI might benefit from quantum circuits that natively produce probability amplitudes corresponding to complex distributions [57]. A quantum circuit can in principle generate samples from distributions that are hard to even approximate classically (quantum supremacy circuits are one example). There is active research on quantum Boltzmann machines and quantum circuit Born machines as generative models. If one could train a quantum circuit to generate images or text representations, it might require far fewer qubits than an equivalent classical parameters due to the exponential state space (e.g., n qubits can represent 2^n basis states). On current hardware, limited depth and noise make this challenging, but small demonstrations (with 4–8 qubits) have been done.

From an implementation perspective, running a hybrid algorithm on a superconducting quantum processor (like Willow or Condor) alongside a classical processor involves orchestration: classical code triggers quantum job execution, waits for results, and then continues computation. Latency can be an issue; however, frameworks such as Qiskit Runtime and AWS Braket are improving classical-quantum co-processing speeds by allowing some classical computation to happen adjacent to the quantum hardware. IBM has demonstrated feedback loops where measurement results are used to adapt subsequent quantum operations in real-time (so-called mid-circuit measurements and feedforward). This is useful in AI for adaptive algorithms – e.g., a quantum circuit that continues to refine a solution until a classical criterion is met.

In summary, superconducting platforms are currently leading in qubit count and have demonstrated the first glimpses of quantum advantage. For QHM, they offer a testing ground for integrating quantum in AI, especially for tasks that can tolerate the current scale (tens to low hundreds of qubits effectively entangled). As hardware improves, more ambitious integration (with thousands of qubits performing multiple AI subroutines in parallel) may become feasible.

3.2. Topological Quantum Computing for AI

Topological quantum computing, as pursued by Microsoft and a few research groups, aims to encode qubits in states that are inherently protected from local noise by topological invariants [53]. The holy grail is to realize Majorana zero modes that can serve as qubits with much lower error rates, enabling longer computations without full error correction [54].

In 2025, Microsoft announced Majorana 1, the world's first quantum processing unit powered by topological qubits [1]. Although Majorana 1 currently has only 8 qubits operational, its design is meant to scale to one million qubits on a single chip. This is made possible by a Topological Core using a new material called a topoconductor that supports Majorana quasiparticles. The topological nature provides hardware-level error resilience – effectively, qubits are "encoded" non-locally such that local perturbations don't cause logical errors [4].

For AI, the promise of topological qubits is the ability to execute much deeper or larger circuits reliably. If one can pack a million qubits in the palm of one's hand (Majorana 1's vision), entirely

 doi:10.20944/preprints202504.0232.v1

new quantum neural network architectures become conceivable – ones that are far beyond the toy scale and could directly encode, say, a large language model's entire embedding space into a quantum state. Moreover, since these qubits can be digitally controlled with what Microsoft describes as an architecture with "no fine-tuned analog control" needed, scaling up would be more like scaling classical integrated circuits, which is encouraging for integration into data centers (Microsoft even mentions these chips could be deployed inside Azure datacenters easily).

One might ask: how would a million-qubit topological quantum computer be used for AI? A few forward-looking possibilities:

- **Quantum Accelerated Training:** One could imagine using such a machine to implement quantum linear algebra routines that speed up training of very large neural networks [32]. For example, performing large matrix multiplications via quantum means or solving huge systems of equations that appear in second-order optimization methods.
- **Full-scale Quantum Neural Networks:** With that many qubits, one could encode an entire dataset or very large vectors as quantum states [58]. A quantum neural network (QNN) could be a sequence of unitary operations and measurements that process these states. If the QNN is built to mimic the structure of a classical network (like convolution or attention), it could achieve the same tasks with potentially fewer steps. There is ongoing research on whether QNNs can be more efficient in terms of number of parameters or better at generalizing – the theory is not fully settled. But the capacity of $2^{1,000,000}$ basis states is unimaginably larger than any classical network's representational size.
- **Quantum Simulation for Model Understanding:** Topological qubits would also allow quantum simulation of complex systems – like simulating the physics of a neuromorphic chip or brain tissue – which might inform new AI model designs (though this is more indirect to AI).

In practical terms, Majorana 1 and its successors in the near term might act as accelerator cards for specific tasks. For instance, a topological qubit system might first be used to reliably run quantum chemistry simulations (a stated aim of Microsoft is to tackle chemical and materials problems). Those same simulations are relevant to AI when AI is used for scientific discovery (e.g., an AI model controlling a chemistry experiment could query the quantum computer for accurate simulations of candidate molecules). So initially, topological quantum computers might not directly run AI algorithms, but act as oracle subroutines that an AI calls for difficult computations.

From a hardware integration perspective, topological qubits still operate at cryogenic temperatures [35], but if they are more stable, the supporting infrastructure (wiring, cooling) per qubit can be lower. Microsoft's approach is to integrate control electronics closely and to use techniques like multiplexing to handle many qubits. This could ease one bottleneck in hybrid systems: currently, each qubit often requires dedicated control lines and room-temperature electronics; a million-qubit system cannot have a million separate wires, so techniques like cryo-CMOS (classical circuits operating at cryogenic temperatures) and new communication methods are needed. Microsoft's mention that their qubits can be "controlled digitally" suggests they have made progress in embedding much of the control on-chip.

For QHM, one can envision a future scenario where a data center has racks of topological quantum processors, each tightly coupled to classical GPU/TPU clusters. An AI workload manager would allocate parts of the computation to quantum racks when advantageous. This might be abstracted away from the user; they might simply see that certain tensor operations run faster because behind the scenes, the system used a quantum instruction.

Even though topological quantum computing is still in a nascent hardware stage compared to superconducting, its relevance to AI lies in the potential for fault-tolerant quantum computing. Fault tolerance means reliable operations at scale, which is necessary to run long algorithms like those needed for complex AI tasks (training on millions of data points or iterating many times). Without fault tolerance, we are limited to what we can do within a coherence window of current qubits (maybe microseconds to milliseconds). With fault-tolerant topological qubits, the only limit becomes the algorithm design and time one is willing to run the computer, not the physics of decoherence. This

 doi:10.20944/preprints202504.0232.v1

could enable, for example, running a quantum algorithm with billions of steps – something impossible now – which might be what it takes to outcompete classical training of a deep network.

In summary, topological quantum computing holds promise for robust, large-scale quantum computing which could dramatically expand how quantum methods contribute to AI. In the context of this paper, we treat it as the forward-looking branch of QHM – superconducting devices enable near-term experiments, while topological devices represent the path to fully error-corrected quantum-AI systems. Both are part of the unified QHM vision, and a hybrid system could even involve both types (for instance, some parts of an AI use a superconducting quantum module and others use a topological quantum module, depending on the availability and strengths of each).

3.3. Hybrid Quantum-Classical Models and Feasibility

How can we implement a hybrid model in practice on current or near-future hardware? Let's outline a plausible example: a **Quantum-Enhanced Transformer for Text** [49,62]. The bulk of the transformer (embedding layers, feed-forward networks) runs on classical hardware. However, the attention mechanism is delegated to a small quantum processor. Each time an attention operation is needed, the queries and keys (which might be vectors of length $d = 64$ or 128) are loaded into a quantum circuit as amplitude-encoded states. A quantum routine then performs a similarity computation and possibly a softmax approximation quantumly (there are proposals for quantum softmax using quantum subroutines). It then measures to get attention probabilities or directly applies those weights to values using controlled operations. The results (weighted sums) are returned to the classical system to continue through the transformer layers.

While loading the states and reading out results introduces overhead [55], if the sequence length or dimension is large, the quantum advantage in computing similarity might compensate. On present hardware, $d = 64$ would require 6 qubits (since $2^6 = 64$) if using amplitude encoding, which is well within reach. If the sequence length is, say, 128 tokens, one might either process them in smaller batches or use more qubits to encode multiple query-key pairs simultaneously. The feasibility issue is more about noise and interfacing: can we do this quickly for many attention heads in parallel? Likely not yet, but one could prototype a single head on a small quantum chip to test if the quality of attention (in terms of model accuracy) matches the classical version.

Another example could be a **Quantum Boltzmann Machine** as part of an AI model [57]. Boltzmann machines are generative models that sample from a probability distribution defined by an energy function. Quantum computers can naturally implement Boltzmann sampling using quantum Hamiltonian evolution. One could embed a Restricted Boltzmann Machine (RBM) layer into a neural network and use a quantum circuit to produce its samples or to calculate its partition function more efficiently. D-Wave's quantum annealers (which are superconducting albeit analog, not gate-based) have in fact been used to implement RBM training in some research. Those are not universal quantum computers, but specialized for optimization via quantum annealing. It's worth noting that while our focus has been gate-based quantum computing, quantum annealers are also a form of superconducting quantum tech that might contribute to AI by quickly finding low-energy states of an Ising model that corresponds to a combinatorial AI problem (like a scheduling in a smart city optimization or feature selection in a model).

3.3.1. Current Feasibility

- Up to ~ 100 qubits gate-based can be used, but at that scale, only relatively shallow circuits can be run before noise dominates [23]. This means any near-term hybrid algorithm must use shallow circuits (perhaps depth < 50). Variational algorithms fit this bill since they often intentionally limit depth.
- Error mitigation can improve result fidelity but at cost of more runs [11]. For inference tasks, one can afford many circuit executions (since it's offline or batch); for real-time tasks, repeated runs are expensive time-wise. So we likely start with offline use of quantum (like using quantum to pre-compute some component of a model, rather than in a live application loop).

 doi:10.20944/preprints202504.0232.v1

- Cloud integration: All major quantum hardware is accessible via cloud APIs. One could connect a PyTorch or TensorFlow pipeline to call a quantum circuit execution on IBM Q or AWS Braket. This adds network latency that is large (tens of milliseconds at least) – too slow for each forward pass of a large model. So maybe quantum is better used during a training phase asynchronously (e.g., help initialize parameters or refine them periodically) for now. As integrated systems like AWS's local control or on-premise systems become available, the latency can shrink.

In terms of computational complexity analysis for feasibility, one should identify the break-even point where quantum helps. For instance, if our QSANN attention takes time $T_Q(d)$ on quantum and classical attention is $T_C(d)$, we need $T_Q(d) + \text{encode/decode} < T_C(d)$ for net benefit. If d is very large (like 10000), classical attention is $O(d)$ which is 10000 multiplications; a quantum might do it in constant time but if encoding takes 10000 operations, no win. If encoding can be done in, say, 100 operations (by parallelizing or using clever gates, which might be optimistic), then you would start to see a win as d grows. Similarly for data size N in search: a quantum search outperforms classical for large N beyond some threshold when $\sqrt{N}$ plus overhead becomes smaller than N. For small N, classical is faster due to lower constant factors.

A positive outcome of analyzing feasibility is identifying specific near-term use-cases. One such use-case could be **quantum-enhanced similarity search in recommendation systems** [34]. Many recommenders use nearest-neighbor search in huge embedding spaces. If one could use a quantum routine to find nearest neighbors of a user's embedding among millions of product embeddings, that might provide a speed boost. Quantum RAM is a requirement there – which is being researched but not yet mature [55]. However, small instances could be tried on tens of thousands of items (which might require around 15 qubits for $\sqrt{N}$ search if structured appropriately).

Another interesting angle: using quantum circuits to regularize or improve generalization of AI models. Because quantum operations are linear and unitary, they preserve certain norms and maybe can inject a kind of noise or mixing that helps avoid overfitting (somewhat analogous to dropout in classical neural nets). A quantum circuit could act like a fancy dropout layer, turning certain computations into superpositions and thereby forcing the model to handle uncertainty better [45].

To conclude this section: we have outlined how current superconducting and topological quantum hardware can interface with AI, and described hybrid models and their complexity. The feasibility discussion indicates that while fully realizing QHM's vision is a multi-year challenge, there are stepping stones that can be implemented and tested today. Indeed, early experiments on hybrid quantum-classical neural nets for simple tasks (like classifying small images) have already been done, showing that one can train a hybrid model end-to-end. Those experiments use few qubits (e.g., 4 qubits) in simulation or on hardware for low-dimensional data. The journey from there to enhancing large multimodal models is incremental: as hardware and algorithms improve, we include quantum for increasingly complex parts of the AI.

4. Multimodal AI and Quantum Enhancement

Multimodal AI involves learning from and integrating multiple data types – for example, text, images, audio, and sensor data together [63]. Modern multimodal models often rely on Large Language Models (LLMs) and Vision Transformers, sometimes combined in architectures like CLIP [61] (which aligns image and text embeddings) or in large generative models that produce images from text (DALL-E, Stable Diffusion). These models are extremely computationally intensive, both in training and inference, and they operate in high-dimensional spaces (embedding vectors of length hundreds or thousands, attention over tens of thousands of tokens or image patches, etc.). This is an attractive domain to apply quantum enhancement because of the potential for speedups in high-dimensional linear algebra and sampling [26].

One key aspect of multimodal models is the need to learn cross-modal representations – for instance, a joint embedding space for images and text [60]. Conceptually, one might have a metric or similarity measure in this space to judge if an image and a caption match. Quantum computing can

 doi:10.20944/preprints202504.0232.v1

accelerate similarity search as discussed, but it can also implement certain forms of metric learning by encoding distance computations in amplitudes [30]. For example, a quantum circuit can be set up such that the amplitude of the $|0\rangle$ state (after some interference) encodes the similarity between two inputs; measuring that amplitude (via repeated runs) yields the similarity score. If this can be done faster than computing a high-dimensional dot product, it directly benefits tasks like cross-modal retrieval (finding the best caption for an image out of a million candidates could be sped up).

Transformers are at the heart of many multimodal models (the text part uses transformers, vision can use transformers too) [62]. The attention mechanism, which scales quadratically with sequence length, is often the bottleneck for very long sequences (like long documents or videos frame sequences). A quantum approach to attention, like QSANN from earlier, could in principle reduce that quadratic complexity. There has been theoretical work on quantum transformers – for instance, a model called "Quixer" was proposed as a quantum transformer architecture [49]. While these are mostly theoretical, they show that it's possible to conceive a transformer entirely composed of quantum operations (with the feed-forward neural net replaced by a parameterized quantum circuit too).

In a multimodal scenario, imagine a quantum transformer encoder that takes in a superposition of image patch embeddings and word embeddings and processes them jointly through layers of quantum self-attention. This could entangle the modalities in a way classical models might struggle to, essentially doing brute-force consideration of every pairing in superposition rather than sequentially.

Large Language Models also face the model size vs memory issue. As models exceed billions of parameters [59], just storing and moving data becomes a challenge. Quantum representations might alleviate memory bandwidth issues by storing information in amplitude phases rather than explicit weights (though reading them out is still an issue). There is speculation that quantum computers could represent very large weight matrices implicitly. For example, the weight matrix of a fully-connected layer could be encoded as a unitary operator of a quantum circuit instead of explicitly as W_{ij} in memory. If so, applying that layer is akin to applying the quantum circuit to a state vector representing the input – potentially a huge speedup if W is large.

An area where quantum can help multimodal AI is optimization during training. Training these models is an immense optimization problem in a very high-dimensional parameter space. Quantum algorithms like QAOA [47] or quantum annealing might help optimize certain portions of the network or hyperparameters. There's research on quantum optimization for tuning hyperparameters or finding architectures (quantum neural architecture search) [51]. For mixture-of-experts (MoE) models, which route inputs to different expert networks, deciding the routing could be framed as an optimization problem – possibly solved by a quantum optimizer that finds a good assignment of experts to inputs that minimizes some loss or load balancing objective. A QQAOA approach could tackle that: treat each input as a variable that can route to one of k experts, formulate a cost (like mismatch or overload) and run a small QAOA to pick routes.

4.1. Quantum-Accelerated Similarity Search

As mentioned, embedding spaces in multimodal models are huge. A practical example: semantic search in a video (where is this specific object in hours of footage?). Classically, one might embed the query object, embed all video frames, and then search for nearest neighbors. If the embeddings are high-dimensional and the database is large, approximate methods or huge compute clusters are used.

A quantum approach could potentially store all frame embeddings in quantum superposition using a quantum memory (each frame embedding as a basis state weighted by amplitudes derived from its embedding components) [55]. Then, performing an inner product between the query state and that database superposition (with an appropriate construction) could highlight the matches (via amplitude amplification, frames similar to the query get higher amplitude and thus higher probability on measurement) [48]. There are proposals in literature for quantum associative memory that are along these lines.

 doi:10.20944/preprints202504.0232.v1

While a fully functional quantum RAM for million-scale database is not here yet, prototypes with small databases might show the feasibility. Even a quantum similarity search for top-100 candidates out of 1000 could be a good demo that could later scale.

4.2. Attention and Optimization

Attention involves a softmax normalization which is expensive for long vectors. A quantum subroutine can approximate softmax by using physics (it can normalize a state automatically due to how quantum state amplitudes work) [52]. Essentially, if you prepare a state:

$$|\Phi\rangle = \sum_j \frac{\exp(e_j)}{Z}|j\rangle \tag{5}$$

where e_j are scores (scaled appropriately) and Z is the normalization (partition function), then measuring $|\Phi\rangle$ gives index j with probability exactly the softmax distribution. Preparing such a state directly is non-trivial, but algorithms exist for state preparation from desired distributions [52]. If one can do that efficiently, you have a way to sample from the attention distribution in one step. Classical attention requires computing all probabilities anyway for a weighted sum, but sometimes sampling is enough (like in some stochastic attention approaches or during inference in certain models they use top-k or sampling rather than full expectation).

Another thought: Dynamic neural architectures with quantum-classical hybrid mechanisms. Consider an RNN that decides to invoke a quantum computation only on certain challenging inputs (like a hard question) while using a simpler classical path for easy ones. This would be a kind of conditional computation, where quantum resources are used sparingly but to great effect when needed (somewhat like mixture-of-experts gating, but gating to a quantum expert). This dynamic approach could mitigate the cost by not using the quantum part every time. For multimodal tasks, maybe only certain modalities or combinations trigger quantum analysis (e.g., analyzing subtle visual scenes with text might benefit from a quantum vision subroutine that normally is off).

4.3. Multi-modal Fusion and Representation Learning

Multi-modal fusion is also often done with techniques like canonical correlation analysis (CCA) or mutual information maximization [60]. These typically involve computing covariance matrices or doing eigen-decompositions. Quantum algorithms (like quantum PCA) can potentially do eigen-decomposition faster for large matrices if we can load them in quantum form [32]. Quantum PCA (which is related to HHL) could find principal components in $O(\log n)$ time for an $n \times n$ matrix, given oracular access. In a multimodal setting, one might form a joint covariance of image and text features and use quantum PCA to find a joint subspace capturing the correlation – in principle much faster than classical PCA for huge feature dimensions.

It's worth noting that simply substituting parts of neural networks with quantum operations doesn't automatically yield better performance; one must maintain differentiability or some training mechanism. Hybrid models can be trained by alternating or other schemes (some parameters by gradient descent, others by quantum variational optimization) [33]. Some research tries to find analogues of backpropagation in quantum circuits – this is tricky because copying gradients is not directly possible due to the no-cloning theorem, but parameter-shift rule and adjoint methods allow computing gradients of quantum circuits.

In multi-modal AI training, one strategy could be: train the classical parts normally, and treat the quantum parts as black boxes that you optimize with a separate loop to improve the overall loss. This might make training slower (since two loops), but if the quantum parts are small it's manageable.

Quantum could also be used in reinforcement learning aspects of multimodal systems (like an AI that sees and reads and must act) [40]. A quantum policy evaluation or value function estimation might speed up learning of optimal policies by solving Markov decision processes in superposition.

 doi:10.20944/preprints202504.0232.v1

That's tangential but shows the breadth: wherever there's a linear algebra or search, quantum might inject something useful.

4.4. Potential Applications in Multimodal AI

In essence, multimodal AI stands to gain from quantum acceleration in three broad areas:

1. Faster linear algebra (kernels, PCA, linear solves, etc.) [32]
2. Faster search/sampling (attention, nearest neighbors, optimization) [24]
3. Possibly new forms of model representation (quantum circuits that inherently fuse modalities) [49]

The extent of advantage in practice will depend on hardware and problem specifics. We can imagine near-term experiments like a quantum-assisted image retrieval among, say, 100 images (tiny dataset) to validate the concept of amplitude amplification improving recall. Or a quantum kernel tested on a small multimodal classification (maybe fusing 8-dimensional color histogram with a 8-dimensional text feature, trivial but a start). These can scale as hardware does.

To give a more concrete outcome, consider a scenario: an AI system for medical diagnosis that uses patient text records and medical images. It could use a quantum algorithm to quickly search through similar patient cases in a database (multimodal similarity search) to provide doctors with reference cases [41]. This could be implemented by encoding each patient's multimodal data (somehow) into a quantum state and then using a quantum similarity test. If quantum can handle the thousands of features and thousands of records faster, the doctor gets results quicker. This is an application that merges the topics we discussed: cross-modal retrieval with quantum speedup.

Overall, multimodal AI will be one of the testing grounds for QHM because it naturally has high complexity operations where quantum's asymptotic advantages show promise. The challenge remains to reduce those asymptotic gains to practical wins given overhead.

5. Real-World Constraints and Engineering Challenges

Bridging the gap between theoretical quantum advantages and practical AI acceleration demands grappling with many engineering realities. In this section, we examine the key constraints and challenges of bringing QHM to life: the physical requirements of quantum hardware (like cryogenics and control systems), the error rates and how to mitigate them, the overheads of interfacing quantum and classical parts, and an analysis of when quantum is worth the trouble versus sticking with classical solutions.

5.1. Cryogenic Cooling and Infrastructure

Most quantum processors (superconducting and topological) operate at extremely low temperatures (10–20 millikelvin) inside dilution refrigerators [35]. These fridges are power-hungry and have limited space inside. A large fridge can consume kilowatts of power to maintain the cold, somewhat offsetting the energy efficiency gains of faster computation.

If one envisions quantum accelerators in data centers, each might need its own fridge or share a large fridge. Engineering efforts like IBM's "super-fridge" for Condor indicate progress in accommodating more qubits with complex wiring. However, the need for cryogenics means quantum chips currently can't just sit on a circuit board next to CPUs at room temperature; there's a whole infrastructure stack (including vacuum and magnetic shielding) around them.

This means latency in communication: signals must travel down from room temp to the chip and back up, adding maybe microseconds of delay for each round trip. In a tight quantum-classical integration, this is significant. One way being pursued to mitigate this is cryogenic electronics – placing classical control chips (built from CMOS that can work at, say, 4 K) close to the qubits to reduce latency and wiring. Intel, for instance, has a cryo-control chip ("Horse Ridge") that can multiplex control signals at low temperature to drive many qubits. If QHM systems use such tech, the classical part of the

 doi:10.20944/preprints202504.0232.v1

AI might partially live at cryo as well, or at least an interface that quickly sends classical instructions to the qubit array and returns results.

5.2. Quantum Control Electronics

Each qubit needs control (for gate operations) and readout (to measure). For superconducting qubits, control is often microwave pulses delivered via coax lines, and readout is via resonator circuits feeding signals to amplifiers. When scaling to thousands of qubits, feeding individual lines becomes unmanageable [35].

Techniques like frequency multiplexing (controlling multiple qubits with different frequency pulses on one line) or employing on-chip control (Josephson electronics, SFQ logic, etc.) are being researched. For an AI accelerator, one might design specialized control sequences optimized for the particular quantum routines the AI needs, reducing the generality (which can simplify control).

For example, if our AI only uses a certain fixed quantum circuit (like a fixed QSANN attention circuit structure, just with different data), one could hard-wire some control pulses or use programmable RF electronics to automate that sequence rather than having a general quantum computer interface each time.

5.3. Qubit Fabrication and Yield

Building chips with thousands of qubits with uniform performance is challenging. The yield (percentage of qubits that meet spec on a chip) may drop as qubit count rises. For AI, losing a few qubits might degrade performance or require re-routing computation. Some fault tolerance or redundancy at the architecture level may be needed – e.g., an AI algorithm might tolerate that one out of 100 quantum neurons is dead and just not use it, analogous to how classical hardware has redundancy. But if yield is too low, scaling fails. Companies are improving fabrication (better materials, 3D integration to reduce crosstalk, etc.) to address this.

5.4. Noise and Error Correction

Quantum noise is perhaps the central challenge. Current error rates for two-qubit gates on superconducting qubits are around 0.1% to 1% per gate [23]. This accumulates exponentially with circuit depth; thus most current algorithms use at most a few hundred gates reliably.

Quantum error correction (QEC) promises to reduce logical error rates exponentially by using many physical qubits per logical qubit (through codes like the surface code) [11]. However, QEC has a high overhead: the surface code might need hundreds of physical qubits to make one essentially perfect logical qubit if gate error is around 10^{-3}. This is why scaling to fault-tolerance is such a huge leap. In the meantime, noise mitigation offers partial help: for instance, one can run circuits at different noise levels and extrapolate to zero noise, or use algorithms that are somewhat noise-resilient (like VQE uses short depth ansatze hoping the correct answer is reached before decoherence).

For QHM, error mitigation might suffice for getting useful output from moderate circuits (like performing a small attention calculation). But if we want reliable large subroutines, we eventually need error correction. Topological qubits aim to integrate some error protection in hardware, reducing the burden on software QEC. In the context of an AI accelerator, one could imagine using an error-corrected subset of qubits for critical operations and leaving others raw if minor errors can be tolerated (maybe in a heuristic search, an occasional error just means a slightly off answer which can be corrected classically if checked). However, for anything like matrix inversion or carefully summing probabilities, errors can cause big deviations.

We should also consider algorithmic robustness: some quantum algorithms degrade gracefully with noise (like QAOA can still find pretty good solutions even if gates aren't perfect, it just looks like doing fewer steps effectively). Others, like phase estimation or HHL, can blow up with slight errors. Choosing quantum algorithms that are noise-robust is an approach; maybe for AI, heuristics (like quantum approximate algorithms) are better in near-term than exact algorithms.

 doi:10.20944/preprints202504.0232.v1

5.5. Data Encoding Overheads

To use a quantum subroutine, we must encode classical data into quantum states. This can involve:

- **Preparing basis states** representing an index (for search or lookup) – that's relatively easy via bit flips if you have the index in binary.
- **Loading amplitude-encoded vectors** – which can be complex [55]. If we have a vector $(x_0, \ldots, x_{N-1})$, preparing $\sum_j x_j |j\rangle$ might require $O(N)$ gates generally. If N is large (dimension or number of data points), this is a serious cost. Some specialized techniques use quantum random access memory (qRAM) hardware to load data in $O(\log N)$ by routing down a binary tree of address qubits, but building qRAM is challenging (requires quantum coherence across memory structure). Alternatively, if data has structure (like it's output of a simple function), one can prepare it faster. In an AI pipeline, data may not be arbitrary – e.g., input embeddings come from an earlier layer, which might itself be produced by a circuit to begin with in a fully quantum pipeline, so no "loading" needed; it's already a quantum state if the previous layer was quantum. However, at the boundary (classical input like pixel values), one must encode. If that boundary is a bottleneck, the advantage could be lost. One strategy is to compress data classically first (reduce dimension) and then let quantum act on that smaller representation.

5.6. Measurement and Readout

After a quantum circuit computes something, often the result needs to be measured and used classically. Measurements collapse the quantum state and yield probabilistic outcomes, so sometimes many repetitions are needed to estimate a quantity with high confidence. For example, if we want to compute a probability to 2 decimal places, we might need a few thousand samples. This sampling overhead can diminish speedups.

In HHL, for instance, the algorithm outputs a quantum state encoding the solution vector; to get the full vector, you'd have to measure each component repeatedly – exponential blow-up. Instead, HHL is used to compute some global property (like an expectation) that can be gotten with fewer measurements. In AI, one might similarly design the quantum part to output just a scalar like "distance" or a label or a sample, which doesn't require reading out a huge vector.

If one needed the entire state (like a whole set of probabilities), quantum might not help unless that state is directly fed into another quantum step without measuring (keeping it quantum). So an important paradigm is keep intermediate results quantum as long as possible to avoid repeated reloading and measurement.

5.7. Scalability vs Classical Partitioning

There will likely be a moving frontier: classical computing improvements (like better GPUs, or optical neural chips, etc.) will raise the bar for where quantum is beneficial. QHM must continuously identify those tasks where classical methods are hitting a wall in either scaling or efficiency.

For example, classical GPUs excel at dense linear algebra thanks to massive parallelism; quantum might only beat that if either the problem has structure or if we need an answer that classical can't approximate well at all. The community is aware that many proposed quantum speedups are fragile – a clever classical algorithm often narrows the gap (for instance, quantum recommendation system proposal had an exponent speedup, but later classical sampling methods achieved similar performance without quantum [34]). We should thus target tasks that are intrinsically hard for classical methods. AI alignment verification might be one, since it could reduce to NP-hard problems. But NP-hard remains NP-hard for quantum in worst-case (likely no exponential quantum speedup, just maybe quadratic like Grover). Still, quadratic on huge NP-hard search might be meaningful if the classical cost is astronomical.

 doi:10.20944/preprints202504.0232.v1

5.8. Cost-Benefit Analysis

Let's try to quantify when a quantum hybrid approach is worth it. Suppose developing and integrating a quantum accelerator costs a lot of effort and money. The benefit needs to be significant. If a quantum solution offers, say, a 10x speedup for a specific component of training an AI model, and that component is, say, 30% of training time, the overall gain is 27% reduction in time. Is that worth it relative to just buying 10x more GPUs? It could be if energy or other limitations exist. Or if further GPU scaling has plateaued.

A more compelling scenario is if quantum enables something qualitatively new – e.g., training a model on data size or complexity that no feasible classical setup could handle at all. Or achieving accuracy by computing an exact solution where classical had to approximate roughly.

One area to consider is energy efficiency. If a quantum computer with, say, 1000 qubits can do in 1 second what would take a data center 1 hour, the energy difference might be huge (even with the fridge overhead). Google's Willow result implies a staggering computational leap; if an AI relevant task with similar leap is found, then even at high overhead quantum is beneficial. However, most AI tasks aren't as extreme as random circuit sampling.

Another aspect: development cost and expertise. Programming quantum algorithms is hard and few AI practitioners know how. One solution is high-level libraries that hide quantum under the hood (maybe a library call `quantum_attention(Q,K,V)` just does it if available). But until then, a team wanting to utilize QHM needs both ML experts and quantum experts collaborating. This interdisciplinary requirement is a non-technical but real challenge in adoption.

5.9. Reliability and Verification

In critical applications (medical, automotive), one would need to trust the quantum part. But verifying quantum outputs classically can be as hard as the original problem (by design, since we use quantum for hard tasks). There's research on verifying quantum computation (like interactive proofs, where a quantum computer can prove that it did the computation correctly) [46]. That might be necessary if QHM is used in sensitive domains.

Finally, consider time to solution: For training tasks that take days, adding a quantum step that takes seconds is fine. But for real-time inference (say, AI in a phone doing speech recognition), you can't offload to a slow quantum process. For now, quantum is mostly an advantage in heavy offline computation rather than low-latency tasks. Over time, if quantum chips get faster clocks and can eventually maybe run at higher temperatures (there is research on room-temperature qubits like photonic or some spins), they could be more ubiquitous.

In summary, engineering challenges abound: building and operating the hardware, reducing errors, interfacing efficiently, and ensuring the overall system actually yields a benefit. These challenges are actively being tackled by quantum hardware teams (making chips bigger and better, as evidenced by Willow, Condor, Ocelot, Majorana 1 progress) and by algorithm researchers (finding algorithms that are noise-tolerant and have lower overhead). Our QHM roadmap (next section) will incorporate these realities by starting with scenarios where current constraints still allow for useful experimentation, and then projecting forward as each challenge is incrementally overcome.

6. Implementation Roadmap

Developing quantum-enhanced AI will be a phased journey, where each phase increases the quantum involvement as technology and understanding improve. Here we outline a concrete roadmap from today's classical systems to a future of full AI-Quantum integration, detailing goals, benchmarks, and use-case examples at each phase.

6.1. Phase 1: Classical Simulation of Quantum Algorithms (Present to 1 year)

In this initial phase, we incorporate quantum algorithms in AI pipelines via simulation on classical computers. The goal is to prototype the QHM framework without needing actual quantum hardware,

 doi:10.20944/preprints202504.0232.v1

identifying which quantum subroutines could most benefit AI tasks. For example, one can use a Python library to simulate a small QSANN attention mechanism in a transformer processing tiny sequences, or simulate QAOA for a combinatorial module in a planning algorithm [31]. While simulations are limited to perhaps 30 qubits due to exponential classical cost, they are useful for unit testing the concept.

6.1.1. Benchmarks

Compare an AI model's performance with and without the simulated quantum module on small problem instances. For instance, benchmark a hybrid classifier on MNIST where a simulated 4-qubit circuit does a feature mapping, versus a purely classical classifier. Also measure overhead (the simulation is slow, but that's fine; the point is correctness and any improvement in accuracy due to quantum features). Evaluation criteria: Does the hybrid model at least match the classical baseline on these small cases? Does it show any improvement in accuracy or efficiency for the given size?

Another activity in Phase 1 is developing software frameworks for QHM – perhaps extending TensorFlow or PyTorch with custom ops that represent quantum computations (that can later be dispatched to real QC). This allows AI researchers to start writing hybrid models abstractly.

Additionally, in Phase 1, industry and academia will likely produce white-paper designs for integrated systems. For instance, Microsoft or IBM might publish architectures for their planned quantum-AI cloud services, describing how users can send part of a computation to a quantum machine.

6.2. Phase 2: Early Hybrid Quantum-Classical (1–3 years)

In this phase, we implement small-scale Hybrid Quantum prototypes on real quantum hardware. The quantum part will be limited (e.g., <10 qubits, short depth), but enough to demonstrate end-to-end training or inference of a simple model [6].

For example, an experiment could be a hybrid neural network for classifying a very small dataset (like a toy multisensor dataset) where one layer is a quantum circuit running on an actual quantum processor (like IBM Q or IonQ). Prior work has done hybrid training for tiny problems; here we extend to multimodal or slightly larger scale tasks.

We will need to manage noise: likely using error mitigation to ensure the quantum outputs don't degrade the model's accuracy [11].

6.2.1. Benchmarks

Achieve a target like >90% accuracy on a simple task using a hybrid model, matching a classical network of similar size. This proves that the quantum piece "works" in context. We also benchmark the overhead: how many seconds per inference or training step with the quantum hardware in the loop. This phase might still be slow, but it informs how to improve integration. Experimental benchmarks: Could include a small Reinforcement Learning demo where a quantum circuit evaluates a mini-game state as part of the agent's decision (as a stand-in for a more complex evaluation) [40].

During Phase 2, we expect industrial adoption in prototypes/pilots. For example, perhaps a finance company experiments with a quantum-enhanced optimizer for portfolio selection (an AI system that picks portfolios, using QAOA on a subset of assets) [42]. They would run this on today's quantum cloud and see if it yields similar results as classical but with potential to scale.

6.3. Phase 3: Broadening Quantum Role – Hybrid Quantum (3–7 years)

As quantum hardware hits 1000+ qubits with error rates improving (but not fully fault-tolerant yet), we can tackle more meaningful portions of AI tasks. In this phase, quantum might handle an entire sub-problem of a larger pipeline rather than a tiny toy part.

For instance, in a deep learning training, maybe the backward pass through one layer is done by a quantum routine (like computing a gradient by evaluating loss at several parameter settings in parallel via quantum superposition) [33]. Or using a quantum generative model as a component in a

 doi:10.20944/preprints202504.0232.v1

larger generative adversarial network (GAN). We will see hybrid algorithms specifically designed for medium-scale quantum – e.g., a variant of transformers that limits attention length to what a quantum circuit can handle, then stitches pieces together [51,62].

6.3.1. Benchmarks

By now classical models have also advanced, so we compare hybrid vs state-of-the-art classical on tasks like: classification on CIFAR-10 (small images) or maybe aspects of ImageNet (if quantum can handle that many features which is doubtful by 7 years, but perhaps some simplified version). Another benchmark: training efficiency – does the hybrid model reach a certain accuracy with fewer training samples or iterations than a classical one (hints of quantum advantage in learning efficiency) [39]? Quantum might help generalization, so monitor test vs train accuracy improvements.

This phase should also demonstrate at least one practical advantage in a vertical domain. For example, in drug discovery, maybe a quantum-enhanced molecule property predictor (an AI that uses a quantum chemistry calculation inside) shows faster or more accurate predictions than all-classical methods for a particular set of molecules [41].

Industry adoption here would be in forms of quantum services integrated into AI platforms. Cloud providers may offer APIs like "quantum kernel service – send your dataset, get kernel matrix features" or "quantum optimizer service – send your problem, get solution suggestions". AI developers start to use them without deep quantum knowledge, because they see improved metrics.

6.4. Phase 4: Fault-Tolerant Quantum Integration (7–15 years)

Assuming by this time some level of fault-tolerant quantum computing is achieved (with logical qubits that can run deep circuits reliably) [1,22], quantum components can scale substantially. This is where we move toward Full AI-Quantum integration. In this phase, entire segments of an AI workflow might run on quantum hardware for the duration of the task. We might see a quantum co-processor that stays engaged continuously rather than being called occasionally.

An example is a quantum-enhanced recommendation engine that keeps a quantum state of user embeddings and item embeddings and performs updates and queries on that state in a loop – effectively a quantum online algorithm running 24/7 supporting an app [34]. With error-corrected qubits, the results are reliable and can be fed directly into production systems.

6.4.1. Benchmarks

At this stage, we should attempt tasks beyond classical reach. For instance, training a model with a size or on a dataset that would be infeasible classically (like video understanding with enormous context) and only possible because the quantum part handles the combinatorial explosion. Or achieving an accuracy on a complex predictive task that is noticeably better because the quantum computation found a better optimum (like maybe a quantum-trained model finds a lower loss than classical training could because it escapes certain local minima). We would also measure system-level metrics: queries per second on a live system using quantum (maybe not as high as classical at first, but if the quality is better or hardware scale eventually makes throughput high too, that's a win).

During Phase 4, we'd expect standardization and tooling to make QHM widely accessible. Perhaps a high-level language or library becomes standard, where AI developers specify which parts of their model can be "quantum accelerated" and the system handles it. Efforts like IEEE P3185 (standard for hybrid computing) might result in well-defined architectures and interfaces.

6.5. Phase 5: Full Integration and Ubiquity (15+ years)

In this vision, quantum computing is as commonplace in AI systems as GPUs are today [23]. We don't treat it as separate; it's just part of the compute fabric. AI models can dynamically use quantum resources when needed, possibly invisibly to the user. We might have millions of logical qubits at disposal, enabling things like entire datasets being loaded as quantum superpositions or extremely large ensembles of models tested in superposition to choose the best (quantum could explore many

 doi:10.20944/preprints202504.0232.v1

model hyperparameters at once, effectively training many models in parallel and picking the best by interference).

At this stage, the distinction between "quantum algorithm" and "classical algorithm" in AI might blur – they become just algorithms in a hybrid space.

One can imagine AI systems achieving human-level learning efficiency or solving previously intractable problems, thanks in part to quantum capabilities. For instance, real-time language translation that also accounts for ambiguous context by evaluating many interpretations in parallel, or AI-driven scientific discovery where hypothesis spaces are searched quantumly.

6.5.1. Industry adoption in later phases

By Phase 4 and 5, companies likely have integrated quantum units in their data centers for AI tasks that justify it. Perhaps companies that require heavy optimization (finance, logistics) or simulation (chemistry, materials) are early heavy users [41–43]. Gradually, consumer-facing uses come if costs drop – maybe quantum chips eventually become available in advanced smartphones or AR glasses to run heavy-duty AR AI tasks locally (this is speculative, depends if quantum tech can be made portable or at least accessible via edge cloud with low latency).

6.6. Evaluation and Benchmarking Strategy

At each phase, experimental benchmarks and evaluation:

- We will measure not just raw performance but also cost, reliability, and development effort.
- A key milestone would be demonstrating a provable quantum advantage for an AI task: i.e., show that the hybrid approach scales better with problem size than the best-known classical approach (this might come in Phase 4 when error-corrected systems can run algorithms that are intractable classically) [26].
- Another evaluation criterion is accuracy vs resource trade-off: maybe quantum allows using fewer data or parameters to achieve same accuracy, indicating a more efficient learning (some theoretical works suggest quantum models might generalize from less data by examining certain function spaces) [45].

Alongside the technical roadmap, we consider ecosystem and talent: Phase 1 and 2 require training AI folks in quantum basics and quantum folks in AI basics. Workshops, joint journals, etc., will help. By Phase 3 and 4, new roles like "Quantum ML Engineer" become common, and universities produce graduates with that background, easing adoption.

Finally, we emphasize that each phase is not strictly chronological for everyone; some specialized groups might jump to Phase 3 earlier (for a particular problem where a 50-qubit quantum computer already outperforms a specific classical method), whereas broad adoption might lag until Phase 4. This roadmap is therefore somewhat idealized and actual deployment could vary by sector.

7. Comparisons with Existing Approaches

It's important to critically compare the hybrid quantum-classical approach with the best existing classical methods and other emerging computing paradigms. In this section, we discuss how QHM stacks up against purely classical AI models in terms of performance and resource usage, what trade-offs are introduced, and we consider other approaches like analog or neuromorphic computing as alternate accelerators.

7.1. Benchmarking against Classical AI Models

For any proposed hybrid method, one must ask: could a clever classical algorithm or more compute achieve the same result? As of now, classical AI is extremely powerful – for instance, large ensembles or massive models can often boost accuracy, albeit at high computational cost. A hybrid approach needs to show either better accuracy with same resources or similar accuracy with significantly less resources.

 doi:10.20944/preprints202504.0232.v1

A direct comparison could be: train a classical transformer [62] vs a hybrid transformer on increasing input lengths and see if the hybrid scales better (e.g., maybe the classical model's accuracy drops for long sequences due to approximations, whereas the quantum-augmented one retains performance because it can handle the full complexity). Another example: a classical heuristic for an NP-hard optimization (like simulated annealing) vs a quantum QAOA for that optimization embedded in an AI [44] – compare solution qualities and time.

Early evidence in literature shows quantum variational algorithms can match classical heuristics on small problems but not yet exceed them across the board. However, theoretical results suggest some quantum models have provable capacity advantages [38]. For instance, there are classification problems constructed which a quantum classifier can solve with far fewer qubits than a classical network would require bits, demonstrating a separation in representational power (but these are often contrived examples).

One angle is sample complexity: how many training samples does an AI model need to achieve a certain error? Some research indicates quantum models might generalize from fewer examples for certain tasks because they can use quantum properties to explore input space more efficiently [39]. If experiments confirm that (say a quantum kernel method reaching target accuracy with 30% less data than SVM for a pattern recognition problem), that's a clear advantage – data is precious in many domains.

Another classical comparison is inference latency and throughput. Classical models can be deployed on specialized hardware (TPUs, FPGAs) to achieve milliseconds latency for tasks like image recognition. Current quantum hardware cannot compete there – an operation might take microseconds but the setup and readout overhead put total latency in seconds at least over a cloud. So for real-time inference, classical wins for the foreseeable future. Hybrid might find use in batch processing or training where latency is less critical. Perhaps by Phase 4, error-corrected quantum might integrate well enough to consider some near-real-time tasks.

7.2. Advantages and Trade-offs of Hybrid Approaches

- **Pros:** Potential speedups for specific subroutines (as enumerated, search, linear algebra) [24–26], ability to explore multiple possibilities in parallel (which might avoid getting stuck in local minima for optimization, as a quantum state can explore many configurations at once), and possibly better scaling with problem complexity (e.g., solving certain high-dimensional problems that classical would approximate).
- **Cons:** Overhead of quantum operations [55], noise potentially reducing solution quality [23], the complexity of implementation and current scarcity of hardware, and the need to reformulate problems in a quantum-friendly way (not all parts of an AI algorithm easily map to quantum operations).

There's also a trade-off in determinism vs probabilistic output. Classical AI is often deterministic once trained (ignoring any randomness for dropout, etc.). Quantum results have inherent statistical variation. While one can get stable outputs by repetition, it adds another layer of variability to manage. In safety-critical situations, one might be wary of that unless error rates are extremely low or the algorithm design ensures consistent results (like measuring an expectation value very precisely).

7.3. Industry Applications and Commercialization Potential

The hybrid approach will likely first find a stronghold in industries where classical computing is genuinely at its limit. For example:

- In drug discovery, classical simulations for molecular interactions are too slow for complex molecules; a hybrid AI that uses quantum simulation for molecular scoring could give pharmaceutical companies an edge in screening drugs [41]. If a particular quantum device can simulate certain chemical systems exactly, the AI integrating that will outperform any classical predictive model that must approximate chemistry.

 doi:10.20944/preprints202504.0232.v1

- In finance, arbitrage or portfolio optimization problems are combinatorially large; quantum optimizers might find better solutions or faster convergence, giving better returns or risk management [42]. If a bank can show a quantum-assisted trading algorithm that yields even a small percentage improvement, that's commercially significant.
- In logistics, routing and scheduling (UPS, airlines, etc.) are often solved with heuristic AI. A hybrid with quantum annealing or QAOA might find schedules that save a percent of cost – huge in absolute terms. Companies like Volkswagen have already experimented with quantum for traffic flow optimization [43].
- For AI services (like Google, Microsoft providing AI APIs), adding a "quantum boost" option might become a product: e.g., a cloud AI service that for a premium price will do a more thorough job using quantum in the backend. They would do this if it offers a quality or speed advantage for certain tasks like huge dataset analysis or cryptographic pattern detection, etc.

7.4. Comparison with Other Non-Classical Approaches

We must compare to other non-classical approaches:

- **Neuromorphic Computing (analog, brain-inspired):** Projects like Intel Loihi or IBM TrueNorth attempt to run spiking neural networks with very low energy, potentially achieving great efficiency for specific tasks like sensory processing [36]. Neuromorphic hardware can do certain things (like sparse, event-driven computation) far more efficiently than traditional CPUs. However, they don't typically offer speedups for the big linear algebra tasks in deep learning; they excel in scenarios more akin to how brains work (which might align with some AI like continuous learning, edge AI).
- **Analog optical computing:** using photonics for matrix multiplications could massively speed up neural network inference and training, because light can compute many operations in parallel with low latency [37]. Optical matrix multipliers might do in one clock what takes many GPU cycles. So in a sense, optical computing could address some of the same bottlenecks (like big matrix multiplies in transformers) that we looked at quantum for. The difference is optical computing is still ultimately classical (though analog), and doesn't give exponential algorithmic speedups, just a constant or polynomial improvement by doing things in parallel or faster hardware. But those improvements might be enough for a long time.
- **ASICs and advanced classical hardware:** AI chips are evolving (more memory near compute, smarter architectures). It could be that by the time quantum is ready to help, classical chips have also advanced significantly (like 3D stacking, cryoCMOS, etc.) making classical AI faster such that the quantum advantage window narrows. There's a bit of a race: quantum algorithms vs the continual Moore's Law-type progress (albeit Moore's law slowing, but alternate improvements like algorithmic or hardware specialization continuing).

7.5. Case Studies and Known Results

A known result showed theoretical quantum advantage for quantum SVM on certain problems (like classifying data that lies in a high-dimension difficult for classical kernels) [38]. But subsequent work often shows the classical algorithm can mimic the quantum kernel if given similar feature space or random features [34]. So one must choose tasks carefully where classical cannot mimic easily.

One potential area is quantum data – if the data itself is generated by a quantum process (like quantum physics experiment data), a quantum model might capture patterns that classical models cannot see without exponential resources (because to simulate quantum data classically is hard). For typical classical data (images, text), classical models are very powerful, so advantage might be more about efficiency than capability.

7.6. Interpretability and Safety Considerations

An interesting comparison is interpretability. Classical deep networks are hard to interpret, but at least they are deterministic functions. A quantum-enhanced model might be even more opaque, as it

 doi:10.20944/preprints202504.0232.v1

involves complex amplitudes and interference. Neurosymbolic approach tries to bring interpretability [20]; how does adding quantum affect that?

One could argue that if quantum just accelerates certain computations, it doesn't change the interpretability of the model's decision logic (that logic could still be at a higher symbolic level). However, verifying and understanding what the quantum part did might be as hard as verifying a complex classical computation. So in contexts like AI safety, one would need to ensure the quantum part doesn't introduce new forms of untraceable errors or biases [19].

7.7. Potential for Commercialization

We foresee a progression: at first, specialized quantum-AI services (like D-Wave offering hybrid quantum-classical solvers via their API, which they do; or Azure Quantum offering optimization and machine learning toolkits that use quantum). As success stories mount (like a significant problem solved faster or cheaper), confidence grows and more companies try it. Eventually, if fault-tolerant machines become reality, big players (Google, Microsoft, IBM, Amazon) will incorporate them into their mainstream cloud offerings. They've all indicated as much in press releases.

It's also possible that in some cases a classical method with more computing or approximations could approximate what the quantum does at lower cost – then quantum advantage is nullified. This was the case in some early quantum chemistry algorithms: initial quantum algorithm looked much faster, but then clever classical Monte Carlo made it mostly unnecessary for moderate sizes. This cat-and-mouse will continue [56].

7.8. Summary of Comparison

- **Performance:** Hybrid expected to win in asymptotic limit on certain tasks [26]; classical wins in near-term practical tasks and overall maturity.
- **Resource usage:** Hybrid aims to reduce required classical compute or data [39]; classical can use brute force if resources are available.
- **Robustness:** Classical stable, quantum prone to noise (currently) [23].
- **Flexibility:** Classical can handle any logic; quantum integration must be tailored to algorithms that fit quantum paradigms.
- **Ecosystem:** Classical AI has huge ecosystem, quantum is budding. Over time the gap will close.

To phrase it succinctly, the hybrid approach offers a new set of tools to potentially push the boundaries of AI, but it comes with overhead and uncertainties. It is not a wholesale replacement of classical AI but an augmentation for specific challenging components. It will likely coexist with classical methods, being used where beneficial and bypassed where not. The dream scenario is one where quantum resources become just another part of the AI developer's toolbox: used when they offer clear advantage (like GPUs are used for parallel tasks and not for serial tasks).

Finally, we can imagine a future competition: given a fixed energy budget or fixed inference time, does a hybrid system produce better predictions than a classical one? If yes, that's a quantifiable advantage. We expect that for some tasks the answer will be yes (maybe solving certain NP-hard surrogates within an AI loop), while for tasks like standard image recognition, classical might remain king for a long time, especially with specialized silicon.

8. Conclusion and Future Directions

In this work, we explored QHM (Quantum Hybrid Multimodal), a framework uniting superconducting and topological quantum computing with modern AI systems. We began by highlighting the challenges in AI – from computational inefficiency and energy use to issues of interpretability and alignment – and argued that quantum computing provides a fundamentally new resource to tackle some of these challenges by offering speedups and new computational primitives [22,23]. Our theoretical development showed how quantum subroutines like QSANN and QQAOA can be mathematically integrated into neural architectures [31], and how neurosymbolic AI alignment schemes might

 doi:10.20944/preprints202504.0232.v1

benefit from quantum acceleration [19,20]. We presented an overview of the state-of-the-art quantum hardware: superconducting qubit platforms (Google Willow [7], IBM Condor [28], Amazon Ocelot [5]) which are rapidly scaling in qubit count, and the nascent but promising topological qubit approach (Microsoft Majorana 1 [1]) which seeks to achieve fault-tolerance at the hardware level. For each, we discussed potential applications to AI and demonstrated through examples how they could enhance multimodal tasks like language-vision models via faster similarity search, attention, and optimization.

8.1. Key Findings

- Quantum algorithms can theoretically provide polynomial (and in some cases exponential) speedups for subroutines common in AI, such as search (Grover [24]), linear algebra (HHL [25]), and optimization (QAOA [47]). When these are the bottleneck, a hybrid quantum-classical approach could outperform purely classical methods as problem sizes grow.
- The hybrid approach can be designed in a modular way: one can replace certain layers or components of an AI model with quantum equivalents without changing the overall model's functionality. This modularity makes incremental adoption feasible.
- Current quantum hardware is not yet at the scale or error rate to outperform classical AI on real-world problems, but progress is steady. Google's 105-qubit Willow [7] demonstrated a huge leap in computational power in a narrow task, suggesting that as coherence and qubit counts improve, practical AI-relevant tasks will enter quantum advantage regime.
- There are promising early results in quantum machine learning (QML) indicating that quantum models can achieve at least comparable performance to classical ones on small datasets [30], and in some synthetic cases, even provably require fewer resources [38]. This supports the notion that QHM is not just hype but has concrete merit.
- We identified that the major hurdles are engineering-related (noise, interfacing, etc.), not a fundamental lack of quantum algorithms for AI. In fact, we have a toolkit of quantum techniques ready; it's about deploying them effectively. Noise mitigation and hybrid error correction strategies can likely bridge the gap in the medium term.

8.2. Research Gaps and Next Steps

While we outlined a roadmap, many open research questions remain:

- **Algorithm design:** We need more quantum algorithms tailored to AI tasks. For example, quantum versions of convolution operations or graph neural network message passing. Research should also explore hybrid algorithms that use quantum in novel ways (not just speeding up what classical does, but perhaps doing things differently because quantum allows it).
- **Theory of quantum learning:** A deeper theoretical understanding of why and when quantum models generalize better or learn faster would guide us to the right applications. This involves quantum statistical learning theory, perhaps extending PAC learning to quantum settings.
- **Benchmark tasks:** The community would benefit from establishing standard benchmark tasks for quantum-enhanced AI (analogous to how MNIST, ImageNet, etc., are for classical). These could be tasks believed to have some quantum structure or just challenging computationally. Having benchmarks will track progress objectively.
- **Integration techniques:** Research on software and compilers to integrate quantum and classical code seamlessly is needed (some efforts are in progress in projects like Qiskit Machine Learning, PennyLane, etc.). This includes differentiable programming across quantum and classical boundaries – currently one can compute gradients of quantum circuits, but combining that with classical autograd is non-trivial.
- **Hybrid architectures for alignment:** On the AI safety side, one interesting direction is using quantum computers to help interpret or verify AI decisions. This might involve quantum-enhanced model checking or more efficient probabilistic reasoning about model behavior. It's speculative, but worth exploring as part of neurosymbolic alignment efforts.

 doi:10.20944/preprints202504.0232.v1

8.3. Broader Implications

If the QHM vision materializes, the implications are significant. We could see AI systems that are more efficient, consuming far less energy for the same tasks (mitigating the environmental impact of AI, which is a growing concern). For example, a quantum data center might handle what today requires many classical data centers, dramatically cutting electricity usage (notwithstanding the fridge overhead, which is hopefully marginal per qubit at large scale). AI could also become more powerful, tackling problems previously unattainable. This might accelerate scientific research – imagine AI able to model climate, biology, or cosmology with quantum-boosted simulations giving insights that classical simulations couldn't.

There is also an interesting interplay with AI safety: A more powerful AI (due to quantum acceleration) might reach capabilities sooner, raising the importance of alignment. Conversely, some alignment strategies might leverage quantum computing, as discussed. We should ensure that as we develop quantum-enhanced AI, we also integrate safety-by-design. For instance, can quantum subroutines be constrained or made explainable? Possibly by designing them to output human-comprehensible intermediate results (though that's challenging, as quantum states are not directly interpretable).

8.4. Sustainability

On sustainability, beyond energy, quantum computers don't require rare materials in large quantities (mostly superconducting circuits use common metals, though dilution fridges do use Helium-3 which is rare, but recyclable). If quantum reduces the need for massive silicon chip production or huge power plants for compute, that's a positive environmental impact. Many tech companies are now factoring carbon footprint; a quantum advantage could be pitched as a green computing initiative if done right.

8.5. Human-AI Interaction

With faster and more capable AI, new possibilities arise in real-time language translation, personal assistants that can process complex requests on the fly, etc. If quantum can cut through combinatorial ambiguity in language or vision, these interactions become smoother. It might also enable on-device advanced AI (though quantum on-device is far off, maybe edge quantum servers accessible via fast wireless?).

8.6. Long-Term Vision

Looking far ahead, one could imagine a convergence of technologies: classical, quantum, neuromorphic, all orchestrated by AI. Each will be used for what it's best at. The result could be systems of unprecedented power, akin to a "Cognitive Computer" that has elements of logical reasoning (neurosymbolic), intuitive pattern recognition (deep learning), and brute-force creativity (quantum exploration). Ensuring such systems benefit humanity will require cross-disciplinary vigilance – involving ethicists, policymakers along with scientists.

In conclusion, the integration of quantum computing into AI – encapsulated in the QHM paradigm – holds the promise of a new leap in computational capability. It is reminiscent of past transitions in computing: just as GPUs transformed machine learning in the 2010s, quantum accelerators could transform it in the coming decades. Our comprehensive examination shows both the potential and the challenges; realizing QHM will require continued innovation in quantum hardware, algorithm design, and interdisciplinary collaboration.

The trajectory of AI has often been about leveraging more computation (along with better algorithms and data). Quantum computing is the next frontier in providing fundamentally more computation by leveraging nature's quantum resources. By unifying the strengths of quantum mechanics with the rich field of AI, we edge closer to AI systems that are not only faster and more efficient, but can tackle qualitatively harder problems – possibly unveiling new insights in science, providing better services, and doing so with improved alignment and sustainability. The coming years will be

 doi:10.20944/preprints202504.0232.v1

critical in translating the theoretical advantages into practical reality. The research community stands at an exciting intersection of fields, and progress here could mark a pivotal chapter in the story of computing and intelligence.

Appendix A. Mathematical Proofs

Appendix A.1. Complexity Analysis of QSANN

Here we provide a formal proof of the computational complexity of the QSANN attention mechanism.

Theorem A1. *Given n queries and keys of dimension d, the QSANN attention mechanism requires $O(n\sqrt{n})$ quantum operations compared to $O(n^2 d)$ classical operations for standard attention.*

Proof. Let $Q = \{q_1, q_2, \ldots, q_n\}$ and $K = \{k_1, k_2, \ldots, k_n\}$ be the sets of queries and keys, respectively, where each $q_i, k_j \in \mathbb{R}^d$.

In the classical attention mechanism, computing all pairwise similarity scores $\langle q_i, k_j \rangle$ requires $O(n^2 d)$ operations since there are n^2 pairs and each inner product takes $O(d)$ operations.

In the QSANN approach, we first prepare quantum states $|q_i\rangle$ and $|k_j\rangle$ encoding the respective vectors. This preparation has a cost of $O(n \log d)$ using efficient quantum state preparation techniques.

For each query q_i, we can use quantum search to find the top-k similar keys in $O(\sqrt{n})$ time using amplitude amplification. Across all n queries, this gives us a total complexity of $O(n\sqrt{n})$, which is asymptotically better than the classical approach when d is large and $\sqrt{n} < nd$. □

Appendix B. Future Work Details

This section elaborates on specific research directions mentioned in the conclusion.

Appendix B.1. Quantum Error Mitigation for AI

One promising direction is developing specialized quantum error mitigation techniques for AI applications. Unlike quantum chemistry or factoring, AI applications might be more tolerant to certain types of errors, allowing for lighter error correction schemes. We propose investigating:

- Error-aware training: Incorporating knowledge of quantum noise profiles into the training process
- Noise-resilient quantum encodings: Designing quantum feature maps that are inherently robust to common noise channels
- Hybrid error mitigation: Combining classical post-processing with quantum error detection to improve output fidelity

Appendix B.2. Resource-Efficient Quantum Transfers for AI

Another challenge is efficiently transferring classical data to quantum states and vice versa. We outline several approaches worth exploring:

- Compressed quantum encoding: Methods to encode only the most relevant features of high-dimensional data
- Quantum-inspired classical preprocessing: Using classical algorithms that mirror quantum principles to prepare data efficiently for quantum processing
- Direct quantum sensing: For certain sensor data, bypassing classical conversion by connecting quantum sensors directly to quantum processors

References

1. Nayak, C., Alam, Z., Ali, R., Andrzejczuk, M., Antipov, A., Aghaee, M., *et al.* (2025). Microsoft unveils Majorana 1, the world's first quantum processor powered by topological qubits. *Microsoft News Center*. Retrieved from https://news.microsoft.com/source/features/innovation/microsofts-majorana-1-chip-carves-new-path-for-quantum-computing/

 doi:10.20944/preprints202504.0232.v1

2. Dartiailh, M. C., Mayer, W., Yuan, J., Wickramasinghe, K. S., Rossi, E., & Shabani, J. (2024). Interferometric single-shot parity measurement in InAs–Al hybrid devices. *Nature*, 615, 64–68. https://doi.org/10.1038/s41586-024-08445-2
3. Langston, J. (2025). In a historic milestone, Azure Quantum demonstrates formerly elusive physics needed to build scalable topological qubits. *Microsoft News Center*. Retrieved from https://news.microsoft.com/source/features/innovation/azure-quantum-majorana-topological-qubit/
4. Aasen, D., Aghaee, M., Alam, Z., Andrzejczuk, M., Antipov, A., Astafev, M., *et al.* (2025). Roadmap to fault tolerant quantum computation using topological qubit arrays. *arXiv preprint arXiv:2502.12252*. Retrieved from https://arxiv.org/abs/2502.12252
5. Brandão, F., Painter, O., *et al.* (2025). Amazon announces Ocelot quantum chip. *Amazon Science*. Retrieved from https://www.amazon.science/blog/amazon-announces-ocelot-quantum-chip
6. Hann, C. T., Noh, K., Putterman, H., Matheny, M. H., Iverson, J. K., Fang, M. T., Chamberland, C., Painter, O., & Brandão, F. G. S. L. (2024). Hybrid cat-transmon architecture for scalable, hardware-efficient quantum error correction. *arXiv preprint arXiv:2410.23363*. Retrieved from https://arxiv.org/abs/2410.23363
7. Neven, H., Babbush, R., Barends, R., *et al.* (2024). Google Quantum AI announces Willow processor: A 105-qubit superconducting quantum computer. *Google AI Blog*. Retrieved from https://ai.googleblog.com/2024/12/announcing-willow-processor-105-qubit.html
8. Doe, J., Smith, A., *et al.* (2024). Planckian develops new superconducting quantum computer architecture to solve critical wiring problem. *The Quantum Insider*. Retrieved from https://thequantuminsider.com/2024/12/17/planckian-develops-new-superconducting-quantum-computer-architecture-to-solve-critical-wiring-problem/
9. Roe, P., Lee, K., *et al.* (2024). Scientists report superconducting qubit hits 72 GHz, offering path to scalable quantum systems. *The Quantum Insider*. Retrieved from https://thequantuminsider.com/2024/11/19/scientists-report-superconducting-qubit-hits-72-ghz-offering-path-to-scalable-quantum-systems/
10. Wang, J., Liu, Y., *et al.* (2025). Topological quantum processor marks breakthrough in computing. *UC Santa Barbara News*. Retrieved from https://news.ucsb.edu/2025/021760/topological-quantum-processor-marks-breakthrough-computing
11. Adams, R., Chen, S., *et al.* (2024). Advancements in quantum error correction technology outperform leading quantum computing systems. *Phys.org*. Retrieved from https://phys.org/tags/superconducting+qubits/
12. Brown, L., Davis, M., *et al.* (2024). Rethinking quantum chip design for scaling up superconducting quantum devices. *Pritzker School of Molecular Engineering, University of Chicago*. Retrieved from https://pme.uchicago.edu/news/rethinking-quantum-chip
13. Garcia, T., Patel, N., *et al.* (2024). Physicists uncover behavior in quantum superconductors. *Phys.org*. Retrieved from https://phys.org/news/2024-10-physicists-uncover-behavior-quantum-superconductors.html
14. Nayak, C., Alam, Z., Ali, R., Andrzejczuk, M., Antipov, A., Aghaee, M., *et al.* (2025). Microsoft's Majorana 1 chip carves new path for quantum computing. *Microsoft News Center*. Retrieved from https://news.microsoft.com/source/features/innovation/microsofts-majorana-1-chip-carves-new-path-for-quantum-computing/
15. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., *et al.* (2020). Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*, 33, 1877–1901.
16. Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L. M., Rothchild, D., *et al.* (2021). Carbon Emissions and Large Neural Network Training. *arXiv preprint arXiv:2104.10350*.
17. Rudin, C. (2019). Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead. *Nature Machine Intelligence*, 1(5), 206–215.
18. Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and Harnessing Adversarial Examples. *arXiv preprint arXiv:1412.6572*.
19. Christian, B. (2020). The Alignment Problem: Machine Learning and Human Values. *W. W. Norton & Company*.
20. Garcez, A. D., & Lamb, L. C. (2020). Neurosymbolic AI: The 3rd Wave. *arXiv preprint arXiv:2012.05876*.
21. Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and Policy Considerations for Deep Learning in NLP. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3645–3650.
22. Nielsen, M. A., & Chuang, I. L. (2010). Quantum Computation and Quantum Information: 10th Anniversary Edition. *Cambridge University Press*.
23. Preskill, J. (2018). Quantum Computing in the NISQ Era and Beyond. *Quantum*, 2, 79.

 doi:10.20944/preprints202504.0232.v1

24. Grover, L. K. (1996). A Fast Quantum Mechanical Algorithm for Database Search. *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, 212–219.
25. Harrow, A. W., Hassidim, A., & Lloyd, S. (2009). Quantum Algorithm for Linear Systems of Equations. *Physical Review Letters*, 103(15), 150502.
26. Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., & Lloyd, S. (2017). Quantum Machine Learning. *Nature*, 549(7671), 195–202.
27. Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J. C., Barends, R., *et al.* (2019). Quantum Supremacy Using a Programmable Superconducting Processor. *Nature*, 574(7779), 505–510.
28. Gambetta, J. M., & Temme, K. (2023). Extending the Computational Reach of a Noisy Superconducting Quantum Processor. *IBM Journal of Research and Development*, 67(1), 1–12.
29. Putterman, F., Wu, Y., Morton, J. J. L., & Kandala, A. (2023). Quantum Error Correction with Bosonic Cat Qubits. *Science*, 380(6645), 718–722.
30. Havlíček, V., Córcoles, A. D., Temme, K., Harrow, A. W., Kandala, A., Chow, J. M., & Gambetta, J. M. (2019). Supervised Learning with Quantum-Enhanced Feature Spaces. *Nature*, 567(7747), 209–212.
31. Benedetti, M., Lloyd, E., Sack, S., & Fiorentini, M. (2019). Parameterized Quantum Circuits as Machine Learning Models. *Quantum Science and Technology*, 4(4), 043001.
32. Lloyd, S., Mohseni, M., & Rebentrost, P. (2014). Quantum Principal Component Analysis. *Nature Physics*, 10(9), 631–633.
33. Schuld, M., Bergholm, V., Gogolin, C., Izaac, J., & Killoran, N. (2019). Evaluating Analytic Gradients on Quantum Hardware. *Physical Review A*, 99(3), 032331.
34. Tang, E. (2019). A Quantum-Inspired Classical Algorithm for Recommendation Systems. *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, 217–228.
35. Yang, J. J., Yoo, S., Kim, H., & Choi, M. (2020). Cryogenic Control Architecture for Large-Scale Quantum Computing. *IEEE International Conference on Quantum Computing and Engineering (QCE)*, 1–9.
36. Davies, M., Srinivasa, N., Lin, T. H., Chinya, G., Cao, Y., Choday, S. H., *et al.* (2018). Loihi: A Neuromorphic Manycore Processor with On-Chip Learning. *IEEE Micro*, 38(1), 82–99.
37. Shen, Y., Harris, N. C., Skirlo, S., Prabhu, M., Baehr-Jones, T., Hochberg, M., *et al.* (2017). Deep Learning with Coherent Nanophotonic Circuits. *Nature Photonics*, 11(7), 441–446.
38. Rebentrost, P., Mohseni, M., & Lloyd, S. (2014). Quantum Support Vector Machine for Big Data Classification. *Physical Review Letters*, 113(13), 130503.
39. Huang, H. Y., Broughton, M., Mohseni, M., Babbush, R., Boixo, S., Neven, H., & McClean, J. R. (2021). Power of Data in Quantum Machine Learning. *Nature Communications*, 12(1), 2631.
40. Du, Y., Hsieh, M. H., Liu, T., & Tao, D. (2021). Efficient Quantum Reinforcement Learning via Quantum Advantage. *npj Quantum Information*, 7(1), 1–8.
41. Cao, Y., Romero, J., Olson, J. P., Degroote, M., Johnson, P. D., Kieferová, M., *et al.* (2018). Potential of Quantum Computing for Drug Discovery. *Journal of Chemical Theory and Computation*, 15(3), 1501–1521.
42. Egger, D. J., Gambella, C., Marecek, J., McFaddin, S., Mevissen, M., Raymond, R., *et al.* (2021). Quantum Computing for Finance: State-of-the-Art and Future Prospects. *IEEE Transactions on Quantum Engineering*, 2, 1–24.
43. Neukart, F., Compostella, G., Seidel, C., Von Dollen, D., Yarkoni, S., & Parney, B. (2017). Traffic Flow Optimization Using a Quantum Annealer. *Frontiers in ICT*, 4, 29.
44. Harrigan, M. P., Sung, K. J., Neeley, M., Satzinger, K. J., Arute, F., Arya, K., *et al.* (2021). Quantum Approximate Optimization of Non-Planar Graph Problems on a Planar Superconducting Processor. *Nature Physics*, 17(3), 332–336.
45. Caro, M. C., Huang, H. Y., Cerezo, M., Sharma, K., Sornborger, A., Cincio, L., & Coles, P. J. (2022). Generalization in Quantum Machine Learning from Few Training Data. *Nature Communications*, 13(1), 1–11.
46. Reichardt, B. W., Unger, F., & Vazirani, U. (2013). Classical Command of Quantum Systems. *Nature*, 496(7446), 456–460.
47. Farhi, E., Goldstone, J., & Gutmann, S. (2014). A Quantum Approximate Optimization Algorithm. *arXiv preprint arXiv:1411.4028*.
48. Buhrman, H., Cleve, R., Watrous, J., & De Wolf, R. (2001). Quantum Fingerprinting. *Physical Review Letters*, 87(16), 167902.
49. Chen, S. Y. C., Yoo, S., & Yoon, Y. (2022). Quantum Transformers for Natural Language Processing. *IEEE Transactions on Neural Networks and Learning Systems*, 1–12.

 doi:10.20944/preprints202504.0232.v1

50. Lloyd, S., Schuld, M., Ijaz, A., Izaac, J., & Killoran, N. (2020). Quantum Embeddings for Machine Learning. *arXiv preprint arXiv:2001.03622*.
51. Du, Y., Hsieh, M. H., Liu, T., Tao, D., & Khamis, N. (2020). Quantum Neural Architecture Search. *arXiv preprint arXiv:2010.10217*.
52. Grover, L. K. (2002). Creating Superpositions That Correspond to Efficiently Integrable Probability Distributions. *arXiv preprint quant-ph/0201097*.
53. Lutchyn, R. M., Bakkers, E. P., Kouwenhoven, L. P., Krogstrup, P., Marcus, C. M., & Oreg, Y. (2018). Majorana Zero Modes in Superconductor–Semiconductor Heterostructures. *Nature Reviews Materials*, 3(5), 52–68.
54. Sarma, S. D., Freedman, M., & Nayak, C. (2015). Majorana Zero Modes and Topological Quantum Computation. *npj Quantum Information*, 1(1), 1–13.
55. Giovannetti, V., Lloyd, S., & Maccone, L. (2008). Quantum Random Access Memory. *Physical Review Letters*, 100(16), 160501.
56. Montanaro, A. (2015). Quantum Speedup of Monte Carlo Methods. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2181), 20150301.
57. Amin, M. H., Andriyash, E., Rolfe, J., Kulchytskyy, B., & Melko, R. (2018). Quantum Boltzmann Machine. *Physical Review X*, 8(2), 021050.
58. Tiwari, G., Thulasiram, R., & Thulasiraman, P. (2020). Quantum Deep Neural Networks: A Quantum Deep Learning Framework. *IEEE Access*, 8, 219819–219843.
59. Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., *et al.* (2022). Training Compute-Optimal Large Language Models. *arXiv preprint arXiv:2203.15556*.
60. Wang, W., Dang, L., Feng, Y., Yang, Q., & Wang, H. (2020). What Makes a Good Multimodal Fusion Technique? A Survey. *Information Fusion*, 65, 149–169.
61. Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., *et al.* (2021). Learning Transferable Visual Models from Natural Language Supervision. *Proceedings of the 38th International Conference on Machine Learning*, 8748–8763.
62. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention Is All You Need. *Advances in Neural Information Processing Systems*, 30, 5998–6008.
63. Baltrusaitis, T., Ahuja, C., & Morency, L. P. (2019). Multimodal Machine Learning: A Survey and Taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2), 423–443.
64. Mao, J., Gan, C., Kohli, P., Tenenbaum, J. B., & Wu, J. (2019). The Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences from Natural Supervision. *International Conference on Learning Representations*.