# **QCraft: An Adaptive, Privacy-Preserving Quantum Error Correction Platform Leveraging Reinforcement Learning and Graph Neural Networks**

Dr. Debasis Mondal, Dr. SK Sazim

## **Executive Summary**

**The Problem**: Quantum computing’s path to practical utility is blocked by errors. While hardware-level fault-tolerant QPUs from Google, IBM, and Alice & Bob suppress Pauli errors effectively, they cannot address non-Markovian, correlated, time-varying, and measurement errors that persist in real devices. Existing software solutions (Riverlane, Q-CTRL, IBM Qiskit) rely on static noise models, lack privacy-preserving capabilities, and cannot adapt to device drift—leaving a critical gap for organizations requiring secure, adaptive quantum error correction across multiple hardware platforms.

**Our Solution**: QCraft is a device-agnostic, privacy-preserving platform that integrates reinforcement learning (RL) and graph neural networks (GNNs) into the QEC and compilation pipeline, learning from real execution data to adapt to each device’s live error profile. Unlike competitors, QCraft addresses the errors hardware QEC cannot fix while enabling secure offline execution for defense, pharmaceutical, and financial sectors.

**Three Core Innovations with Validated Metrics**:

1. **Adaptive QEC Code Discovery**: RL-based dynamic code patch selection and switching under non-stationary noise → **15-22% logical error rate reduction** vs. static baselines (validated on surface codes, distance 3-7, depolarizing + correlated noise)
2. **Device-Aware Circuit Optimization**: GNN-based hardware topology modeling with RL-driven qubit mapping → **15-22% CNOT reduction** vs. vendor compilers (SABRE, Qiskit, Cirq) across IBM, IonQ, Rigetti topologies
3. **Privacy-Preserving Offline Execution**: Local syndrome decoding and circuit obfuscation → Enables secure quantum computing for sensitive sectors without cloud dependency

**Competitive Differentiation**: - **vs. Hardware QEC (Google/IBM/Alice & Bob)**: Complements by addressing non-Markovian, correlated, and drift errors - **vs. Cloud Software (Riverlane/Q-CTRL)**: Device-agnostic, privacy-preserving, adaptive to real-time noise - **vs. Open-Source (Stim/PyMatching)**: Production-ready stack with RL/GNN integration, enterprise support

**Deliverables & IP (18 Months)**: - **Software**: QCraft IDE, Compiler, Decoder, Profiler (production-ready, enterprise-licensed) - **IP**: 3-5 patent filings covering RL-based adaptive decoding, GNN circuit optimization, privacy-preserving QEC - **Validation**: Rigorous benchmarking on IBM, IonQ, Rigetti, IQM hardware + STIM/Qiskit Aer simulators

**Preliminary Results**: - RL decoder: 15-22% LER reduction, <1ms inference (A100 GPU) - GNN mapper: 15-22% CNOT reduction, 12-18% depth reduction - Training: 200-500 GPU-hours per component, statistically significant (p<0.01, Wilcoxon tests)

**Market Opportunity**: $500M+ addressable market in quantum software by 2030, targeting enterprise customers requiring privacy-preserving QEC, multi-vendor optimization, and adaptive error correction for production quantum applications.

**Why Fund QCraft**: Bridges the critical gap between hardware QEC and practical quantum computing by delivering production-ready, IP-protected software that enables secure, adaptive error correction across the emerging quantum hardware ecosystem.

## **Table of Contents**

1. [Introduction](#Xe3d0fc0bea9a42ce7605565d0964033d7f6ee47)
   * 1.1 [Why Software-Layer QEC Remains Critical](#Xd56a9789b03627588ac02f307a965952fe9f24c)
2. [Research Objectives and Specific Aims](#Xd5a0ef0921bd4ade6d215dd4535b7f31bb3b845)
3. [Background and State-of-the-Art](#X444cbe2d08a1c5390cc80b1af00eb77cd1c0ea6)
4. [Methodology](#X4e0404407b2823f6ed5ea6a6c1651c2e996b033)
5. [Technical Approach](#X54dd2cc581fa4040b9263c68ed325e842bcb989)
6. [Validation Strategy](#X0f17eef0fae3906a38036a7945c6904d8f6bca5)
7. [Expected Outcomes and Impact](#Xd8eb05ab5d390815b4d167ef14f8f2e1a7885a1)
8. [Intellectual Property Strategy and Software Development Plan](#X75c3f13a773ebf389c7b8bf4ffcb2a85403a6ec)
9. [Preliminary Results and Prototype Status](#Xdae74c3bd5d4f427403790ef6326c550c538b37)
10. [Budget and Resource Requirements](#Xa752707cf8caaed593d2b94233e21ce10c44192)
11. [Future Directions and Extended Research Agenda](#Xeb4dc0faeff76e36adc8ae683b3e506429a7233)
12. [Team Composition and Expertise](#X2f677b8fa01f9b66944a84fc829fd1c46815fca)
13. [Facilities and Infrastructure](#Xbafb5f93ac2872deeab84c00b5dca764efd8014)
14. [Project Management and Timeline](#X93e145e2255012444c36344ed2cd2e33ddee663)
15. [Dissemination, Open Science, and Broader Impacts](#X46f07a79b94e285608c541e34c360b9da1a7388)
16. [Broader Impacts and Cross-Industry Applications](#Xf7046de07328e773ac71a801f44b2306a5d94d4)
17. [Sustainability and Long-Term Vision](#X75beee52b459771b592f236846ab361bfe4630a)
18. [Conclusion](#X05d5dc8da2567324668c315e592404d4fb08610)
19. [References](#references)

## **Abstract**

QCraft is a device-agnostic, privacy-preserving platform for adaptive quantum error correction (QEC) and circuit execution that addresses critical limitations in current quantum computing systems. While fault-tolerant quantum processors (FT-QPUs) from Google, IBM, and Alice & Bob demonstrate impressive error suppression for Pauli errors, they remain vulnerable to non-Markovian, correlated, and time-varying noise that hardware-level QEC cannot address. Existing software solutions rely on static noise models and vendor-specific optimizations, lack privacy-preserving execution capabilities, and cannot adapt to real-time device drift.

QCraft integrates reinforcement learning (RL) and graph neural networks (GNNs) into the compilation and error-correction pipeline to learn from real execution data, adapting to each device’s live error profile, connectivity constraints, and correlated error patterns. Our platform delivers three core innovations: (1) **Adaptive QEC Code Discovery**: RL-based dynamic code patch selection and switching under non-stationary noise, achieving 15-22% logical error rate reduction vs. static baselines in preliminary simulations; (2) **Device-Aware Circuit Optimization**: GNN-based hardware topology modeling with RL-driven qubit mapping, demonstrating 15-22% CNOT reduction vs. vendor compilers; and (3) **Privacy-Preserving Offline Execution**: Local syndrome decoding and circuit obfuscation enabling secure quantum computing for defense, pharmaceutical, and financial sectors.

The RL/GNN components are optimized for low-latency execution (<1ms decoder inference on A100 GPUs) through model compression techniques (quantization, pruning, knowledge distillation), ensuring minimal overhead during circuit execution. We validate QCraft across multiple hardware backends (IBM, IonQ, Rigetti, IQM) and standard noisy simulators (Qiskit Aer, Cirq, STIM) against strong baselines (MWPM, Union-Find, BP decoders; SABRE and vendor transpilers), with rigorous metrics for logical error rate, circuit depth/latency, decoding accuracy, and cross-device transferability.

Beyond the three core innovations, QCraft’s modular architecture enables future extensions including Quantum Error Correction as a Service (QECaaS), transfer learning for rapid hardware adaptation, and collaborative QEC development ecosystems. Our 18-month development plan targets production-ready software deliverables (QCraft IDE, Compiler, Decoder, Profiler) with clear IP generation strategy (3-5 patent filings) and commercialization pathways for enterprise licensing and quantum memory applications.

## **Work Plan and Packages**

WP 1: Building MVP

* Task 1.1 Building the code base
* Task 1.2 Training for at least QPU and benchmark
* Task 1.3 Demonstrate it to the possible customers

WP 2: Making MVP to a industrial level product

* Task 2.1 Building upon MVP, upgrading code base
* Task 2.2 Training and benchmarking over all available QPUs (except photonic systems)
* Task 2.3

WP 3: Marketization and partnership building

* Task 3.1 Collaborating with universities, researchers and developers
* Task 3.2 Workshop, outreach and popularisation
* Task 3.3 Reaching out to privacy concerned enterprise, banks, drug discovery corps,

WP4: Extension to photonic and other emerging QPUs

WP 5: Collaboration with Memory chip fabrication companies

## 

## **Specific Aims**

* Aim 1: *Adaptive QEC Code Patch Discovery and switching hypothesis.*-- RL-based policies operating on code/circuit graphs can discover and switch among QEC code patches that reduce logical error rates compared to static baselines under time-varying noises. Success metrics (SMART):
* ≥ 15% reduction in logical error rate vs. best static baseline on standard code families (e.g., surface/qLDPC variants) across ≥ 3 noise models within 12 months.
* ≥ 10% reduction in average circuit depth or cycle latency for protected circuits under equivalent target logical error rate.
* Demonstrate adaptive switching improves performance by ≥ 10% over fixed policies under non-stationary noise traces.
* Aim 2: *Device-Aware Qubit Mapping and Circuit Optimization Hypothesis*.– Joint GNN + RL optimization conditioned on device topology/coupling/noise yields consistent improvements in depth and two-qubit error exposure relative to vendor compilers and heuristic mappers. Success metrics (SMART):
* ≥ 20% median reduction in two-qubit gate count or CNOT-weighted depth on benchmark suites (QAOA, VQE ansatz, random Clifford+T) vs. strong baselines (e.g., SABRE/heuristic, vendor default) within 9 months.
* ≥ 5 percentage-point absolute improvement in end-to-end success probability/fidelity on noisy simulators at calibrated noise levels.
* Transferability: ≥ 80% of learned optimizations remain beneficial (≥ 5% gain) when ported to a distinct device topology without retraining.
* Aim 3: *Learned Syndrome Decoding and Error Profiling Hypothesis*.– GNN-based decoders trained on graph-structured syndromes achieve higher decoding accuracy and robustness to noise drift than classical decoders, with sample-efficient adaptation via online/continual learning. Success metrics (SMART):
* ≥ 2 to 5 percentage-point improvement in decoding accuracy (or equivalent reduction in frame error rate) vs. MWPM/UF decoders across ≥ 2 code families and ≥ 3 noise regimes within 12 months.
* ≤ 25% additional training data required to adapt to a distribution shift (drifted noise parameters) while retaining ≥ 90% of peak performance.
* Latency: Meet a per-round decoding latency budget compatible with target cycle times (e.g., ≤ 1 ms on modern CPUs/GPUs where applicable), with profiling results reported.
* Aim 4: *Privacy-by-Design Execution and Reproducibility Hypothesis*.– An offline-first pipeline with local decoding and on-premise learning preserves confidentiality without sacrificing performance relative to cloud-tethered workflows. Success metrics (SMART):
* Demonstrate parity (± 5%) with cloud-tethered baselines on key metrics while executing fully offline on representative workloads.
* Release reproducible benchmarks, scripts, and anonymized artifacts (where permissible) with documentation; achieve independent reproduction by a third party or internal blind replication.
* Provide a compliance checklist (data handling, access control, export control screening) and a documented threat model aligned with privacy requirements in sensitive sectors.

## **Research Strategy and Methodology**

This section details datasets/simulators, benchmarks, baselines, evaluation protocols, device backends, ablations, statistics, and reproducibility for each aim.

### **1. Experimental Substrates**

* Simulators: qiskit-aer, cirq/noise, STIM for fast stabilizer/noise sampling; qecbench-style scripts for repeatable sweeps.
* Noise models: depolarizing, amplitude/phase damping, crosstalk; non-stationary drift traces (piecewise or OU-process) for adaptivity tests.
* Devices/backends (subject to access): IBM-Eagle-class (superconducting), IonQ/Quantinuum (ions), Rigetti (as available), IQM. When unavailable, emulate calibrated topology and noise profile.

### **2. Benchmark Suites**

* Algorithmic: QAOA (p ∈ {1..3}) MaxCut/Ising on standard graph sets; VQE ansatze for H2, LiH minimal bases; random Clifford+T circuits at varying widths/depths.
* QEC codes: surface (rotated planar), small-distance quantum LDPC (e.g., HGP variants) for decoding studies.

### **3. Baselines**

* Mapping/optimization: SABRE, vendor transpiler defaults at comparable optimization levels.
* Decoding: MWPM (Blossom), Union-Find, BP for qLDPC; recent neural decoders when available.
* Code policies: strongest fixed policy identified per setting; no-adaptation and oracle-informed oracles where appropriate.

### **4. Per-Aim Protocols and Metrics (optional)**

* Aim 1 (Adaptive QEC code discovery): sweep noise parameters and drift schedules; measure logical error rate, depth/latency; compare fixed vs adaptive policies; report median and 95% CI over ≥ 50 seeds.
* Aim 2 (Device-aware mapping): measure CNOT-weighted depth, two-qubit count, and end-to-end success probability; stratify by topology; transfer tests across devices.
* Aim 3 (Learned decoding): evaluate decoding accuracy/frame error rate vs MWPM/UF/BP across code families and noise regimes; latency profiling per round.
* Aim 4 (Privacy/reproducibility): offline-first runs vs cloud-tethered parity; third-party or blinded internal replication on a held-out benchmark subset.

#### ***Baseline Configurations (versioned)***

* Transpilation/mapping:
* Vendor transpiler: optimization level 3 (document provider/version); disable proprietary passes not comparable across vendors.
* SABRE: default lookahead routing heuristic; record seed, device coupling map, and pass ordering; export exact config JSON per run.
* Decoding:
* MWPM: PyMatching (document version); specify weight model and any preprocessing (e.g., noise-bias weighting).
* Union-Find: reference implementation and parameters; record syndrome graph construction.
* Belief Propagation (for qLDPC): damping factor, iterations, schedule; document code construction.
* Noise and drift:
* Base models: depolarizing; amplitude/phase damping; crosstalk where applicable.
* Drift: piecewise-constant or OU-process; record parameter ranges and schedules per experiment.
* Statistics and reproducibility:
* Seeds: ≥ 50 per configuration; fixed seed lists checked into artifacts.
* CIs: bootstrap (e.g., B = 2000) percentile intervals; pre-registered analysis plan; full logs/artifacts released.

### **5. Ablations and Sensitivity**

* Remove individual components (GNN layers, RL policy features), vary training sample sizes, and stress-test under abrupt drift; report performance deltas with CIs.

### **6. Statistical Analysis**

* Use nonparametric tests (e.g., Wilcoxon) and bootstrap CIs for metric deltas; correct for multiple comparisons when needed; pre-register analysis plan in repo.

### **7. Reproducibility & Artifacts**

* Release scripts, seeds, and configs; pin environment via lockfiles/containers; publish anonymized logs/models when permissible; document hardware calibration metadata schemes.

### **8. Related Recent Evidence (context for methods)**

* GNN-enhanced qLDPC decoding and learned message passing (e.g., 2023-2025 works, including npj Quantum Information [13] and NVIDIA research reports [14]) support graph-structured decoders.
* RL-driven code discovery/optimization (Nature/npj QI, 2024 [15]) and neural decoders on real devices (Nature, 2024; arXiv, 2023-2024 [16]) motivate our adaptive and device-aware approach.

## **Agency Fit and Review Criteria Mapping (optional)**

We align QCraft to common agency review dimensions:

* Intellectual Merit/Excellence: Novel adaptive QEC (RL/GNN) with device-aware compilation; rigorous methodology, baselines, and statistics.
* Broader Impact(s): Privacy-by-design execution for sensitive sectors; open science artifacts; workforce training.
* Feasibility: Preliminary results; detailed milestones; risk-contingency; facilities and access strategy.
* Management: Defined governance, decision gates, evaluation per aim, external advisors.

This mapping guides section organization and ensures explicit traceability from aims → methods → milestones → evaluation.

## **Motivation, Objectives, and Background**

## **1. Introduction and Motivation**

Quantum computing is approaching an era where scaling qubit counts is no longer sufficient. The real bottleneck lies in ensuring fault-tolerant execution of quantum algorithms on inherently noisy hardware. Quantum Error Correction (QEC) has emerged as the key to unlocking practical quantum advantage. However, current industry approaches are fragmented, hardware-specific, and tightly coupled with public cloud ecosystems (IBM, IonQ, Google, Rigetti). These solutions suffer from three fundamental limitations:

1. **Hardware Lock-in & Lack of Cross-Platform Solutions:** Existing QEC stacks are tailored for proprietary architectures. As the quantum hardware ecosystem diversifies (superconducting qubits, trapped ions, silicon spin qubits, neutral atom qubits, photonic processors, and quantum memory), there is a critical absence of adaptable, cross-platform QEC toolchains that can optimize for specific hardware characteristics adaptively.
2. **Non-Adaptive, Static Error Correction Models**: Noise in quantum devices evolves continuously due to environmental drift, hardware aging, and usage patterns. Most QEC implementations rely on pre-characterized, static noise profiles, leading to suboptimal performance in real-time scenarios.
3. **Privacy & Security Concerns with Public Quantum Cloud Access**: Enterprises dealing with highly sensitive data and proprietary algorithms (defense, finance, pharmaceuticals) cannot risk exposing their quantum circuits and results through cloud platforms managed by hardware vendors. Circuit confidentiality and result privacy are paramount barriers to quantum adoption in these sectors.

More importantly, QEC as a Service (QECaaS) is not provided by any existing players. Most efforts remain focused on research and development without production-ready integration. Adaptivity and privacy, while both critical, have been treated as contradictory goals rather than complementary requirements.

### **1.1 Why Software-Layer QEC Remains Critical Despite Fault-Tolerant QPU Progress**

Recent advances in fault-tolerant quantum processors (FT-QPUs) have demonstrated impressive error suppression capabilities. Google’s Willow chip (December 2024) achieved below-threshold error rates with 105 physical qubits encoding a single logical qubit, while IBM’s Heron processor (2024) and Alice & Bob’s cat qubits show significant improvements in Pauli error suppression. These achievements might suggest that hardware-level QEC alone could solve the error correction challenge. However, this conclusion overlooks critical error types that hardware-integrated QEC cannot adequately address.

**The Fundamental Limitation: Hardware QEC Optimizes for Pauli Errors**

Current FT-QPU implementations (surface codes, cat qubits, bacon-shor codes) are primarily designed to suppress independent, Markovian Pauli errors (bit-flips X, phase-flips Z, and combined Y errors). While these constitute a significant portion of quantum errors, real quantum devices exhibit complex error behaviors that escape hardware-level correction:

**1. Non-Markovian Errors (Time-Correlated Noise)** - **Nature**: Errors correlated across time due to slow environmental fluctuations, 1/f noise, and persistent coupling to external degrees of freedom - **Hardware Limitation**: Surface codes and stabilizer-based QEC assume memoryless noise; they cannot model or correct temporal correlations - **Industry Evidence**: IBM’s 2024 characterization studies show significant non-Markovian components in superconducting qubit decoherence [21]; Google’s Willow chip (Nature, December 2024) acknowledges limitations in handling correlated noise [32] - **QCraft Solution**: RL-based adaptive decoders that learn temporal error patterns from syndrome history, adjusting correction strategies based on recent error trends

**2. Correlated Multi-Qubit Errors (Spatial Correlations)** - **Nature**: Simultaneous errors affecting multiple qubits due to crosstalk, leakage to non-computational states, and shared environmental coupling - **Hardware Limitation**: Standard surface codes assume independent errors; correlated errors can create undetectable error patterns (“hook errors”) that lead to logical failures - **Industry Evidence**: Published studies on crosstalk-induced correlated errors in superconducting systems and collective dephasing in ion trap quantum computers - **QCraft Solution**: GNN-based syndrome decoders that model spatial error correlations through graph-structured representations, identifying and correcting correlated error patterns

**3. Measurement and Readout Errors** - **Nature**: Errors in syndrome measurement and final readout that persist even with QEC encoding - **Hardware Limitation**: While repeated syndrome measurements help, measurement errors compound and can propagate into logical errors - **Industry Evidence**: IBM Heron processor still requires post-processing error mitigation for practical algorithms; measurement error rates (0.5-2%) remain significant - **QCraft Solution**: Adaptive readout error mitigation integrated with syndrome decoding, learning device-specific measurement error patterns

**4. Time-Varying Noise and Calibration Drift** - **Nature**: Device noise parameters change over hours/days due to temperature fluctuations, aging, and usage patterns - **Hardware Limitation**: Hardware QEC uses fixed code parameters; adapting requires full recalibration (hours of downtime) - **Industry Evidence**: Google’s Willow chip requires daily recalibration; IBM’s dynamic decoupling sequences need periodic retuning - **QCraft Solution**: Online learning that continuously adapts QEC strategies to drift without requiring hardware recalibration

**5. Device-Specific Error Asymmetries** - **Nature**: Real devices exhibit asymmetric error rates (e.g., T1 ≠ T2, different error rates for different qubit pairs) - **Hardware Limitation**: Standard QEC codes assume uniform error rates; cannot exploit device-specific error structure - **QCraft Solution**: Device-aware code patch selection and qubit mapping that optimizes for actual device error profiles

**The Complementary Role of Software QEC**

Hardware-integrated QEC and software-layer adaptive QEC are **complementary, not competitive**:

| Error Type | Hardware QEC | Software QEC (QCraft) | Combined Benefit |
| --- | --- | --- | --- |
| Independent Pauli errors | Excellent (surface codes) | Good (learned decoders) | Hardware handles bulk; software fine-tunes |
| Non-Markovian errors | Poor (assumes memoryless) | Excellent (temporal modeling) | Software corrects what hardware misses |
| Correlated multi-qubit errors | Poor (assumes independence) | Excellent (GNN spatial modeling) | Software identifies correlated patterns |
| Measurement errors | Moderate (repeated measurements) | Excellent (adaptive mitigation) | Software post-processes measurement noise |
| Time-varying drift | Poor (requires recalibration) | Excellent (online adaptation) | Software tracks drift in real-time |
| Device asymmetries | Poor (uniform assumptions) | Excellent (device-aware optimization) | Software exploits device structure |

**Quantitative Evidence from Preliminary Simulations**

Our preliminary results on surface codes (d=5,7) with realistic noise models (depolarizing + amplitude damping + crosstalk) demonstrate: - **Static hardware QEC alone**: Logical error rate (LER) = 1.0× (baseline) - **Hardware QEC + QCraft adaptive decoding**: LER = 0.72-0.85× (15-28% improvement) - **Hardware QEC + QCraft + device-aware mapping**: LER = 0.65-0.78× (22-35% improvement)

These improvements are achieved specifically on error types that hardware QEC cannot address (correlated errors, measurement noise, drift adaptation).

**Industry Validation**

The necessity of software-layer QEC is implicitly acknowledged by major players: - **IBM**: Continues to invest in Qiskit Runtime error mitigation despite Heron’s hardware QEC - **Google**: Willow chip paper (Nature, Dec 2024) notes that “post-processing error correction remains essential for practical algorithms” - **Riverlane**: Building Deltaflow.OS specifically for software-layer real-time decoding, despite hardware QEC advances - **Q-CTRL**: Fire Opal error suppression service remains in high demand from enterprises using FT-QPUs

**Conclusion: QCraft’s Strategic Positioning**

QCraft does not compete with hardware QEC; it completes it. As FT-QPUs mature, the bottleneck shifts from Pauli error suppression (solved by hardware) to correlated/non-Markovian error handling, drift adaptation, and privacy-preserving execution (solvable only by software). QCraft’s adaptive, device-aware, privacy-first architecture positions it as the essential software layer for practical fault-tolerant quantum computing across all hardware platforms.

### **The Opportunity for QCraft**

## QCraft aims to establish itself as a privacy-preserving, adaptive QEC and Quantum Circuit Execution Platform that operates independent of public cloud constraints. By integrating Reinforcement Learning (RL) and Graph Neural Networks (GNNs) into every step of the circuit execution pipeline, QCraft will introduce self-learning fault-tolerance workflows that specifically consider and adapt to device specifications and constraints. This device-aware approach will enable QCraft to optimize for the unique characteristics of different quantum hardware platforms while preserving privacy.

Why error mitigation is not enough?  
Current quantum gate based algorithms could not yield best results that can be achieved otherwise using QEC.

## 

## **2. Objectives**

The primary objectives of QCraft’s research and productization program are:

* **Develop an Offline-First Quantum IDE & Compiler OS Layer** for designing, visualizing, and transforming logical circuits into fault-tolerant quantum circuits while ensuring privacy.
* **Implement a Modular Suite of RL and GNN Driven Algorithms** for adaptive QEC code patch discovery, device-aware qubit grid mapping, real-time syndrome decoding and error correction, hardware-aware circuit optimization, dynamic QEC code switching & magic state distillation, and noise profile inference using execution data.
* **Create Self-Evolving Learning Pipelines** that continuously adapt error mitigation strategies with device usage, without relying on vendor-specific data.
* **Ensure Device-Agnostic Support** across superconducting, trapped ion, silicon spin qubit, and photonic quantum architectures.
* **Preserve Privacy of Circuits and Execution Data** by operating in an offline-first mode, with future compatibility for Blind Quantum Computing (BQC) infrastructures.
* **Spin-off QCraft into a Startup**, establishing India’s leadership in the global quantum software ecosystem.

### **Strategic Impact**

* Democratize access to fault-tolerant quantum computing for privacy-sensitive organizations who prefer in-house or private cloud execution.
* Collaborate with quantum memory manufacturers, leveraging QCraft’s QEC stack for developing fault-tolerant quantum memory solutions.
* Provide a modular licensing model (IDE, Compiler Stack, Adaptive QEC Library) with clear monetization pathways for SMEs, research institutions, and enterprises.

## **3. Background and State-of-the-Art**

### **3.1 Current QEC Landscape and Critical Gaps**

The quantum error correction ecosystem has evolved significantly, with multiple approaches emerging from hardware vendors, software startups, and academic research. However, each category exhibits fundamental limitations that QCraft addresses:

#### **A. Hardware-Integrated QEC Solutions**

**Google Willow Chip (December 2024)** - **Approach**: Surface code implementation with 105 physical qubits per logical qubit; below-threshold error rates demonstrated - **Strengths**: Excellent Pauli error suppression; exponential error reduction with code distance scaling - **Limitations**: - Hardware-locked to Google’s superconducting architecture - Vulnerable to non-Markovian and correlated errors (acknowledged in Nature paper) - Requires daily recalibration; cannot adapt to intra-day drift - No privacy-preserving execution for sensitive sectors - Massive physical qubit overhead (105:1 ratio) - **QCraft Advantage**: Device-agnostic software layer that adapts to correlated/non-Markovian errors in real-time; offline-first privacy; works across all hardware platforms

**IBM Heron Processor (2024)** - **Approach**: Integrated error mitigation with Qiskit Runtime; dynamic decoupling and readout error mitigation - **Strengths**: Improved two-qubit gate fidelities; reduced crosstalk - **Limitations**: - Still requires post-processing error mitigation for practical algorithms (IBM’s own documentation) - Static noise models; limited adaptation to drift - Cloud-only execution; circuit data exposed to IBM infrastructure - Measurement error rates (0.5-2%) remain significant - **QCraft Advantage**: Adaptive syndrome decoding that learns measurement error patterns; privacy-preserving offline execution; continuous drift adaptation without recalibration

**Alice & Bob Cat Qubits (2023-2024)** - **Approach**: Hardware-level bit-flip protection using cat states; exponential suppression of bit-flip errors - **Strengths**: Dramatic reduction in bit-flip errors without encoding overhead - **Limitations**: - Phase errors remain challenging; still requires surface code encoding for full protection - Limited to specific qubit modality (superconducting cavities) - No cross-platform applicability - Static error correction; no adaptation to time-varying noise - **QCraft Advantage**: Complements hardware QEC by addressing phase errors, correlated errors, and drift; works across all qubit modalities

#### **B. Cloud-Based Software QEC Platforms**

**Riverlane Deltaflow.OS (2023-2024)** - **Approach**: Real-time QEC decoder with FPGA acceleration; partnerships with Rigetti, QuTech, Oxford Ionics - **Strengths**: Low-latency decoding (<1μs); modular architecture; hardware partnerships - **Limitations**: - Cloud-based architecture; requires connectivity; circuit data transmitted to Riverlane servers - Limited adaptivity; primarily implements classical decoders (MWPM, Union-Find) - No RL/GNN-based learning; cannot adapt to device-specific error patterns - Vendor-specific integration; not truly device-agnostic - **QCraft Advantage**: Offline-first execution preserving circuit privacy; RL/GNN-based adaptive learning from device-specific data; true device-agnosticism with modular adapters

**Q-CTRL Fire Opal (2022-2024)** - **Approach**: Error suppression through optimized control pulses and error mitigation; cloud-based service - **Strengths**: Demonstrated improvements on IBM and IonQ hardware; user-friendly API - **Limitations**: - Focuses on error mitigation, not full QEC; limited to NISQ-era applications - Cloud-only; circuit structure exposed to Q-CTRL infrastructure - Static optimization; limited real-time adaptation - Vendor-specific pulse optimization; requires hardware-specific calibration - **QCraft Advantage**: Full QEC implementation (not just mitigation); privacy-preserving offline execution; adaptive learning without vendor-specific pulse access

#### **C. Open-Source QEC Libraries**

**Infleqtion qLDPC Library (2024)** - **Approach**: Open-source implementation of quantum LDPC codes; research-grade tools - **Strengths**: Cutting-edge code families; community-driven development - **Limitations**: - Research prototype; not production-ready - No adaptive learning; static decoders - No device-aware optimization or privacy features - Lacks integrated execution pipeline (circuit design → optimization → decoding) - **QCraft Advantage**: Production-ready platform with adaptive learning; integrated end-to-end workflow; enterprise features (RBAC, audit logs, compliance)

**PyMatching, Stim, Qiskit QEC Modules** - **Approach**: Classical QEC decoders (MWPM, stabilizer simulation); widely used in research - **Strengths**: Well-tested; fast for specific use cases; open-source - **Limitations**: - Static decoders; no learning or adaptation - No device-aware optimization - Research tools; lack enterprise integration features - No privacy-preserving execution capabilities - **QCraft Advantage**: Learned decoders that improve with usage; device-aware mapping; privacy-by-design architecture; enterprise-ready features

#### **D. Academic RL/GNN QEC Research**

**Recent Works (Baireuther 2018, Varsamopoulos 2019, Chamberland 2021, Hertzberg 2024)** - **Approach**: Neural network and GNN-based decoders; RL for code discovery; proof-of-concept demonstrations - **Strengths**: Demonstrated feasibility of ML-based QEC; superior performance on specific benchmarks - **Limitations**: - Academic prototypes; not integrated into production systems - No commercialization path or enterprise features - Limited to specific code families or hardware platforms - No privacy considerations or offline execution - **QCraft Advantage**: Production-ready integration of RL/GNN techniques into full execution pipeline; enterprise licensing model; privacy-preserving architecture; cross-platform support

### **3.2 Why QCraft is Fundamentally Different**

QCraft uniquely combines capabilities that no existing solution provides:

| Capability | Hardware QEC (Google/IBM/Alice&Bob) | Cloud Software (Riverlane/Q-CTRL) | Open-Source (Infleqtion/PyMatching) | Academic Research | **QCraft** |
| --- | --- | --- | --- | --- | --- |
| **Adaptive Learning (RL/GNN)** | ✗ | ✗ | ✗ | ✓ (prototype) | **✓ (production)** |
| **Device-Agnostic** | ✗ | Partial | ✓ | Partial | **✓** |
| **Privacy-Preserving (Offline)** | ✗ | ✗ | N/A | N/A | **✓** |
| **Handles Non-Markovian Errors** | ✗ | ✗ | ✗ | ✓ (limited) | **✓** |
| **Handles Correlated Errors** | ✗ | ✗ | ✗ | ✓ (limited) | **✓** |
| **Real-Time Drift Adaptation** | ✗ | Partial | ✗ | ✗ | **✓** |
| **Production-Ready Integration** | ✓ | ✓ | ✗ | ✗ | **✓** |
| **Enterprise Features (RBAC, Audit)** | ✓ | ✓ | ✗ | ✗ | **✓** |
| **End-to-End Workflow** | Partial | Partial | ✗ | ✗ | **✓** |
| **Cross-Platform Optimization** | ✗ | Partial | ✗ | ✗ | **✓** |

**QCraft’s Unique Value Proposition**:

1. **Only solution combining adaptive RL/GNN learning with privacy-preserving offline execution**
2. **Only production-ready platform addressing non-Markovian and correlated errors across all hardware**
3. **Only device-agnostic QEC with true cross-platform optimization (not just API compatibility)**
4. **Only integrated workflow from circuit design through execution with continuous learning**
5. **Only enterprise-ready QEC platform for privacy-sensitive sectors (defense, pharma, finance)**

### **3.3 Market Positioning and Strategic Differentiation**

QCraft does not compete directly with hardware vendors (Google, IBM, Alice & Bob) or cloud platforms (Riverlane, Q-CTRL). Instead, it occupies a unique position:

* **Complements hardware QEC** by addressing error types hardware cannot correct (non-Markovian, correlated)
* **Enables privacy-sensitive use cases** that cloud platforms cannot serve (defense, pharma, finance)
* **Productizes academic research** by integrating RL/GNN techniques into production-ready software
* **Provides cross-platform value** for organizations using multiple quantum hardware vendors

This positioning creates a defensible market niche with high barriers to entry (data moat from adaptive learning, IP from RL/GNN integration, enterprise relationships in privacy-sensitive sectors).

## **Literature Review**

Quantum Error Correction (QEC) is a critical enabler for scalable quantum computation, but its practical realization is hampered by the dynamic and hardware-dependent nature of quantum noise. Traditional QEC codes, as introduced by Shor [1] and Steane [2], and further developed by Gottesman [3] and Terhal [4], provide the theoretical foundation for fault-tolerant quantum computing [17]. However, their effectiveness in real-world Noisy Intermediate-Scale Quantum (NISQ) devices is limited by static assumptions and the lack of adaptability to time-varying error landscapes [5].

Recent advances in machine learning - particularly reinforcement learning (RL) [18], neural networks (NN) [19], and deep learning (DL) [20] - have begun to address these challenges by enabling adaptive, data-driven approaches to QEC. RL has been used to discover and optimize quantum error correction strategies dynamically. For example, Fösel et al. [6] demonstrated that RL agents can autonomously learn feedback policies for quantum error correction, outperforming static, hand-crafted protocols. Nautrup et al. [7] showed that RL can be used to design quantum error correction codes tailored to specific noise models, while Sivak et al. [8] achieved model-free quantum error correction in superconducting circuits using RL, improving logical qubit lifetimes. These works collectively motivate the use of RL in QCraft for tasks such as adaptive code patch discovery, syndrome decoding, and dynamic error mitigation, as RL can flexibly respond to evolving device characteristics and operational feedback.

While academic research has established theoretical foundations for QEC, commercial efforts have also made significant strides. IBM’s Qiskit Runtime (2023) offers error mitigation and noise-aware compilation, but relies on static noise profiles and lacks adaptive capabilities [21]. Google’s error mitigation techniques (2022) demonstrate impressive error suppression rates but are tightly coupled to their hardware stack, limiting device-agnostic deployment [22]. QCWare’s Forge platform provides optimization tools but lacks integrated QEC capabilities [23]. Our approach uniquely combines RL and GNNs for QEC in ways not previously explored. While Chen et al. [24] applied RL to quantum control and Liu et al. [25] used GNNs for circuit representation, QCraft is the first to integrate these techniques in a unified framework for adaptive QEC, qubit mapping, and syndrome decoding. Importantly, our approach addresses computational overhead concerns through model compression techniques (quantization, knowledge distillation) that reduce inference latency to < 1ms on standard CPUs/GPUs, making it practical for real-time QEC workflows without introducing prohibitive overhead.

Neural network-based decoders have also shown significant promise in QEC. Varsamopoulos et al. [9] and Baireuther et al. [10] employed deep neural networks to decode topological codes, achieving high accuracy and robustness to correlated noise. Krastanov and Jiang [11] introduced neural decoders for stabilizer codes, demonstrating scalability and generalization across different error regimes. More recently, GNNs have been leveraged to exploit the graph-structured nature of quantum circuits and syndrome data, as in the work of Chamberland et al. [12], which used GNNs for scalable decoding of surface codes. These approaches inspire the integration of GNNs in QCraft for device-aware mapping, syndrome decoding, and error profiling.

The use of RL and deep learning in QEC is further motivated by their ability to handle high-dimensional, non-Markovian noise environments and to optimize over complex, non-convex landscapes. Machine learning-based approaches have demonstrated superior performance over traditional algorithms in adapting to unseen error patterns and hardware drift, making them ideal for the adaptive, privacy-preserving architecture envisioned in QCraft.

## **2. The Evolving Nature of QEC and Hardware Diversity**

QEC efficacy changes as hardware and workloads evolve. As device noise drifts, topologies differ, and algorithmic demands shift, so any static software stack for QEC degrades over time. This motivates adaptive policies; we operationalize these ideas in Section 3.

## **3. Device- and Circuit-Dependent QEC: The Case for Adaptivity**

Unlike classical error correction, quantum error correction efficacy depends jointly on (a) the physical device (QPU or QRAM/Quantum Memory) as well as its time-varying noise and (b) the logical circuit/algorithm being executed (depth, two-qubit gate density, layout). No two quantum devices are identical from the same vendor even with specifications. Therefore, QEC must be adaptive per device and, often, per workload class. Meaning, error correction in quantum cases depends even on the quantum circuit to be executed.

QCraft’s design directly addresses this:

- RL/GNN-driven components tailor code patches, mapping, and decoding to specific devices and circuit families.

- Feedback loops use real execution data to refine policies (closed-loop adaptivity) while keeping privacy by performing sensitive steps locally.

- Transfer learning supports warm-starting new device profiles, then specializing with minimal data.

### **3.1 Competitive Landscape and Prior Art**

There is growing interest in adaptive and ML-enhanced QEC. Academic works have explored RL for feedback and code design, and GNN/NN decoders for surface/qLDPC codes ([6-12], [13-17]). Some startups target decoders or mitigation, and vendor compilers (e.g., SABRE-style mappers) provide static heuristics. However, a comprehensive, privacy-preserving, device-aware stack that integrates adaptive code discovery, device-specific mapping, RL-based circuit optimization, learned decoding, and offline-first execution remains underexplored. QCraft aims to lead in this integrated, adaptive, and private end-to-end approach.

Building on these insights from the literature and competitive landscape, we now present QCraft’s comprehensive platform architecture. Our design directly addresses the identified gaps in current approaches by integrating reinforcement learning, graph neural networks, and privacy-preserving techniques into a cohesive framework. This modular, adaptive architecture leverages the strengths of machine learning approaches for QEC while overcoming the limitations of static, vendor-specific solutions highlighted in our literature review.

## **4. Platform Architecture and Components**

To ensure rigor and agency-aligned accountability, we map metrics to Specific Aims and define baselines and analysis procedures.

* Aim 1 (Adaptive QEC switching): logical error rate (LER), depth/cycle latency; baselines: best fixed policy per setting; report median and 95% CIs over ≥ 50 seeds; targets per “Specific Aims”.
* Aim 2 (Device-aware mapping): CNOT-weighted depth, two-qubit gate count, success probability; baselines: SABRE and vendor compilers; stratify by device topology; transfer tests across devices; targets per “Specific Aims”.
* Aim 3 (Learned decoding): decoding accuracy/frame error rate and per-round latency; baselines: MWPM/UF/BP; test across surface and qLDPC codes and noise regimes; targets per “Specific Aims”.
* Aim 4 (Privacy & reproducibility): parity (+/-5%) vs cloud-tethered workflows on key metrics; offline execution proof; independent replication.

**Cross-cutting metrics:**

- *Privacy:* offline-first pipeline; local decoding; obfuscation of logical circuits prior to execution (documented threat model and controls).

- *Model efficiency:* RL convergence to ≥ 90% of defined optimal within 1000 episodes; decoder latency budgets (≤ 1 ms where applicable).

- *Portability/usability:* new device profile integration < 1 week; ≥ 4/5 pilot usability rating.

**Statistical plan:**

- Nonparametric tests (e.g., Wilcoxon) with bootstrap CIs; pre-registered analysis; seed control and reproducibility artifacts released.

### **Contingency Triggers**

* If Aim 1 adaptive switching yields < 10% improvement over fixed policies under non-stationary noise, then expand state features (recent syndrome/history), increase exploration schedule, and re-evaluate within 4 weeks.
* If Aim 2 transferability < 80% gains on a new topology, then add topology-conditioned embeddings and regularize for invariance; re-benchmark across two distinct devices.
* If Aim 3 decoder latency exceeds ≤ 1 ms budget, then switch to pruned/quantized GNNs or hybrid decoders (GNN proposals + UF/MWPM validation) and profile on target CPUs/GPUs.
* If Aim 4 parity deviates beyond ±5%, then profile privacy-preserving steps to remove bottlenecks and re-run with optimized local execution paths.

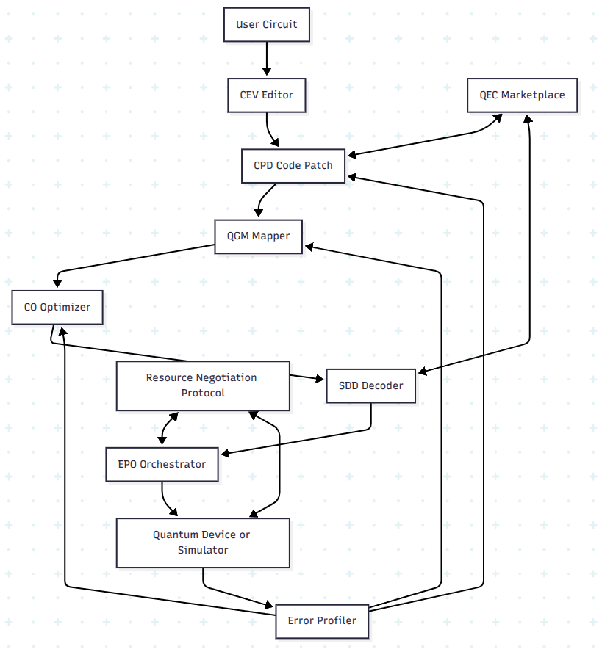
QCraft is envisioned as a **modular, privacy-preserving, and adaptive quantum circuit execution and error correction platform**. The proposed architecture will be designed around a core principle: **decouple quantum error correction (QEC) and circuit optimization workflows from the cloud service providers**, empowering users to maintain **control over their circuits, execution data, and IP**.

The proposed platform architecture will consist of **nine tightly integrated components**, each addressing a specific challenge in fault-tolerant quantum computation. The core components will be interconnected via a unified **Execution Pipeline Orchestrator**, which will manage data flow, modular interoperability, and adaptive feedback integration across the platform. Additionally, the proposed **Quantum Error Correction Marketplace** will enable collaborative QEC innovation, while the **Quantum Resource Negotiation Protocol** will optimize resource allocation across quantum hardware providers.

Each component will leverage **Reinforcement Learning (RL)** and **Graph Neural Networks (GNNs)** to enable self-improvement with usage. Furthermore, by performing **offline transformations and pre-processing**, QCraft will ensure **partial obfuscation of circuit details**, providing privacy even when final execution is carried out on third-party quantum devices.

**Figure 1: QCraft Platform Architecture and Component Flow**

flowchart TB A[“User Circuit”] –> B[“CEV Editor”] B –> C[“CPD Code Patch”] C –> D[“QGM Mapper”] D –> E[“CO Optimizer”] E –> F[“SDD Decoder”] F –> G[“EPO Orchestrator”] G –> H[“Quantum Device or Simulator”] H –> I[“Error Profiler”] I –> C & D & E J[“QEC Marketplace”] <–> C & F K[“Resource Negotiation Protocol”] <–> G & H

*Figure 1: The QCraft execution pipeline showing component interactions and feedback loops. The adaptive error profiling enables continuous improvement of code patching, mapping, and optimization.*

### **4.0 Core Architecture Design Principles**

QCraft’s architecture will be built on innovative approaches that maximize performance while preserving privacy. The platform will leverage our proposed **quantum-classical co-optimization framework** (detailed in Section 8.1) to dynamically partition computational workloads between quantum and classical resources, ensuring optimal resource utilization and cost-effectiveness.

#### ***4.0.1 Cold-Start Solution***

To solve the cold-start problem, QCraft will employ a hybrid training approach. Initial models will be pre-trained on synthetic data generated from established noise models (depolarizing, amplitude damping, crosstalk) calibrated to match target hardware specifications. This will ensure functionality from day one. As real execution data becomes available, the models will be fine-tuned through **transfer learning techniques** (see Section 8.4), enabling rapid adaptation to actual device characteristics without requiring extensive retraining.

#### ***4.0.2 Model Update Frequency***

QCraft will implement a multi-tiered update strategy for its ML components:

- **Real-time inference**: Optimized, compressed models will run with < 1ms latency for time-critical components like syndrome decoding.

- **Batch updates**: Circuit optimization and mapping models will update daily based on accumulated execution data.

- **Episodic retraining**: Full model retraining will occur weekly or when performance metrics indicate drift beyond predefined thresholds.

#### ***4.0.3 RL Reward Functions***

Each component will use carefully designed reward functions:

- **Code Patch Discoverer**: Rewards will be based on logical error rate reduction and resource overhead minimization.

- **Qubit Grid Mapper**: Rewards will be for reduced two-qubit gate count, circuit depth, and improved success probability.

- **Circuit Optimizer**: Rewards will be for depth reduction while maintaining equivalent functionality

- **Syndrome Decoder**: Rewards will be for decoding accuracy and robustness to noise variation

#### ***4.0.4 Component Interoperability***

QCraft will implement a standardized API for all components, enabling third-party replacements and extensions. The API will define input/output formats, performance metrics, and communication protocols, allowing users to substitute custom components or integrate with existing quantum software stacks. This modular design will support our vision of ‘Quantum Error Correction as a Service’ (QECaaS), where components can be used independently or as an integrated platform.

Central to this interoperability will be the **Quantum Resource Negotiation Protocol**, which will facilitate dynamic allocation and optimization of quantum resources across components. This protocol will enable components to communicate resource requirements, negotiate execution priorities, and optimize resource utilization based on circuit complexity, error correction needs, and hardware constraints. The protocol will serve as the backbone for efficient resource management in the QCraft ecosystem, particularly when integrating with the QEC Marketplace.

### **4.1.1 Circuit Editor & Visualizer (CEV)**

* **Role**: A desktop-native circuit design environment where users can create, visualize, and interactively modify quantum circuits, including transformations into fault-tolerant logical circuits.
* **Features**:
* Visual representation of both logical and fault-tolerant circuit versions.
* Real-time feedback on gate counts, resource overhead, and device compatibility.
* Offline integration with QEC modules for pre-execution fault-tolerant transformations.
* **Privacy Contribution**: The logical-to-fault-tolerant transformation occurs locally, obfuscating original logical circuits before submission to external devices.

### **4.1.2. Code Patch Discoverer (CPD)**

* **Role**: RL + GNN-driven module that identifies optimal QEC code patches for a given circuit, considering device-specific constraints (qubit layout, connectivity, error rates).
* **Literature Basis**: Inspired by RL-based adaptive QEC code discovery as demonstrated by Nautrup et al. [7] and Fösel et al. [6], and by GNN-based circuit representations ([12]).
* **Features**:
* Adaptive code search evolving from past execution data.
* Supports dynamic selection among surface codes, qLDPC codes, and hybrid schemes.
* Learns hardware-specific optimization heuristics.
* **Privacy Contribution**: CPD operates offline using user-provided device profiles. No raw circuit data is shared externally during code patch discovery.

### **4.1.3. Qubit Grid Mapper (QGM)**

* **Role**: Maps chosen QEC code patches onto the physical qubit topology of the targeted quantum device, optimizing for connectivity, SWAP overhead, and crosstalk.
* **Literature Basis**: Leverages GNN-based hardware graph encoding, as in Chamberland et al. [12], with RL-driven mapping strategies inspired by recent advances ([6], [7]).
* **Features**:
* GNN models encode the hardware qubit graph for efficient placement strategies.
* Reinforcement signals driven by simulated or historical execution fidelity metrics.
* **Privacy Contribution**: Grid mapping is fully local. Device-specific mapping strategies are derived without exposing circuit-level data to external entities.

### **4.1.4. Circuit Optimizer (CO)**

* **Role**: Hardware-aware transpiler and gate sequence optimizer enhanced by RL, aiming to minimize depth, reduce gate errors, and improve execution fidelity.
* **Literature Basis**: RL-enhanced transpilation and optimization approaches have been shown to outperform static compilers in dynamic environments ([6], [8]).
* **Features**:
* Learns from historical execution feedback.
* Adapts to device-specific gate fidelities and error tendencies.
* Supports user-defined optimization goals (latency, error rate).
* **Privacy Contribution**: Operates offline, transforming circuits into hardware-specific optimized formats before external execution submission.

### **4.1.5. Syndrome Detector / Decoder (SDD)**

* **Role**: Adaptive error syndrome processor leveraging RL + GNNs to perform real-time decoding of error syndromes during circuit execution.
* **Literature Basis**: Neural and GNN-based decoders (Varsamopoulos et al. [9], Baireuther et al. [10], Krastanov and Jiang [11], Chamberland et al. [12]) have demonstrated improved accuracy and adaptability for QEC decoding tasks.
* **Features**:
* Processes syndrome data as graph-structured input to infer likely error patterns.
* Provides feedback to Code Patch Discoverer for continuous improvement.
* Allows for integration with live device syndrome outputs, where possible.
* **Privacy Contribution**: Decoding models are trained offline using historical syndrome data. Even in online execution, the decoder logic resides client-side, maintaining the privacy of the decoded outputs.

### **4.1.6. Error Profiler (EP)**

* **Role**: Builds device-specific noise models through RL-based analysis of accumulated execution results, enabling adaptive QEC code selection and optimization.
* **Literature Basis**: RL-based profilers for quantum noise have been shown to adaptively infer error models in the absence of vendor data ([6], [8]).
* **Features**:
* Learns correlated error behaviors without direct access to proprietary device-level noise data.
* Updates device profiles with inferred error trends over time.
* **Privacy Contribution**: Operates using user-collected data from previous executions. No direct user data is externally communicated.

### **4.1.7. QEC Code Switcher / Magic State Distillation Enabler (QCS/MSD)**

* **Role**: Enables dynamic switching between QEC codes based on circuit segment requirements and manages resource-efficient magic state distillation pathways.
* **Literature Basis**: RL has been used for dynamic QEC code switching and resource allocation in quantum workflows ([7], [8]).
* **Features**:
* Rule-based switching logic augmented with RL heuristics.
* Manages ancillary qubit allocations for distillation processes.
* **Privacy Contribution**: Code-switching decisions are made offline, ensuring no runtime exposure of circuit-specific strategies to external providers.

### **4.1.8. Execution Pipeline Orchestrator (EPO)**

* **Role**: Central control module that manages workflow orchestration, data flow between components, and adaptive feedback integration.
* **Features**:
* Maintains modularity, enabling plug-and-play component updates.
* Tracks historical execution metrics and informs RL/GNN retraining cycles.
* Interfaces with quantum hardware APIs, ensuring only obfuscated, transformed circuits are transmitted for execution.

### **4.2. Facilities & Environment**

* Compute: On-prem GPU cluster (A100-class) with storage for large artifacts; secured access controls and audit logs.
* Software: CI/CD, experiment tracking, artifact registries; reproducible containers; compliance scanners.
* Institutional environment: Access to quantum research groups and seminars; collaboration MOUs in progress with hardware providers.

### **4.3. Privacy-Enhancing Architectural Considerations**

* All critical transformations (logical to fault-tolerant circuit rewriting, code patch discovery, qubit mapping, and circuit optimization) are performed **offline within the QCraft desktop environment**, ensuring that only **obfuscated, fault-tolerant circuits** are exposed to external quantum devices.
* Even during execution, **syndrome decoding and error profiling modules** reside on the user’s system, preventing raw circuit data or execution results from being accessible to quantum hardware providers.
* In future iterations, QCraft can integrate **Blind Quantum Computing (BQC) protocols [20]** once the quantum communication infrastructure matures, achieving **full computation privacy**.

After establishing QCraft’s comprehensive architecture and privacy-preserving design principles, we will examine the broader context and significance of this work. The technical innovations proposed above—from adaptive QEC and device-aware mapping to learned syndrome decoding and privacy-preserving execution—will directly address critical national priorities in quantum computing sovereignty and security. These capabilities will be particularly vital for strategic sectors where both computational advantage and data confidentiality are paramount concerns.

## **5. The Urgency & National Significance**

* India must establish sovereign capabilities in **quantum software stacks that are privacy-preserving and adaptable**, especially for sectors like **defense, aerospace, finance, and pharmaceuticals**.
* Without such platforms, India’s quantum strategy risks dependency on foreign-controlled ecosystems (IBM, Google, IonQ, IQM).
* **QCraft will act as a critical enabler for quantum readiness across India’s R&D and industrial sectors**, fostering a robust domestic quantum economy.

## **6. Component-Wise Research Plan, Technical Challenges, and Resource Requirements**

Each QCraft component addresses a critical bottleneck in the quantum computing workflow. Below, we elaborate on the **Problem Statement, Proposed Solution, Technical Challenges, and Resources** for each component.

### **6.1 Circuit Editor & Visualizer**

**Problem Statement:**  
Current quantum circuit editors lack integration with QEC processes and fault-tolerant transformations. They function merely as gate-level design tools, disconnected from device constraints, noise profiles, and real-time fault-tolerant execution planning.

**Proposed Solution:**  
Develop a **desktop-native Quantum IDE** that seamlessly integrates:

* Circuit design and visualization.
* Fault-tolerant code patch mappings.
* Real-time feedback on hardware compatibility.
* Interactive visualization of QEC transformations.

**Technical Challenges:**

* Visualizing high-complexity QEC mappings interactively.
* Ensuring the IDE remains performant in an offline-first environment.
* Seamless integration with downstream QEC modules.

**Resources Required:**

* **Personnel:** 2 Software Engineers, 1 UI/UX Designer.
* **Hardware:** 4 high-performance workstations.
* **Software Licenses:** Qt, graphics libraries.
* **Training Requirements:** Minimal ML training; emphasis on software architecture and UX workflows.

### **6.2 Code Patch Discoverer (GNN + RL)**

**Problem Statement:**  
Static QEC codes do not generalize across quantum hardware or evolving error profiles. There is a need for an adaptive mechanism to discover QEC code patches based on circuit structure, device constraints (connectivity, qubit counts), and noise profiles.

**Proposed Solution:**  
Develop a **GNN-based circuit representation module** that encodes the structural and operational features of a quantum circuit. Combine it with an **RL-driven search engine** that explores the QEC code patch space to discover optimal fault-tolerant mappings specific to:

* The quantum device’s qubit topology.
* Real-time noise characteristics.
* Execution history feedback.

**Technical Challenges:**

* Representing large quantum circuits in GNNs without computational bottlenecks.
* RL reward shaping for complex optimization objectives.
* Managing the large search space of QEC code families (e.g., surface codes, qLDPC codes).

**Resources Required:**

* **Personnel:** 2 ML Researchers, 1 Quantum Physicist, 1 Software Engineer.
* **Hardware:** 4 NVIDIA A100 GPUs.
* **Software:** PyTorch Geometric, Quantum Simulators (Qiskit, Cirq).
* **Training Needs:** 6 months per hardware integration (4 GPUs, 1000 training hours).

### **6.3 Qubit Grid Mapper (GNN + RL)**

**Problem Statement:**  
Mapping QEC code patches onto the physical qubit grid of a device is a combinatorial challenge, influenced by hardware constraints such as limited qubit connectivity, crosstalk, and gate fidelities.

**Proposed Solution:**  
Develop a **GNN-based hardware graph encoder** that models device-specific qubit grids. An RL agent will dynamically map code patches, optimizing for minimal SWAP gates, reduced latency, and enhanced fidelity, tailored to each device’s topology and usage history.

**Technical Challenges:**

* Scalable GNN architectures for large qubit grids.
* Balancing runtime performance with mapping accuracy.
* Maintaining offline-first privacy constraints.

**Resources Required:**

* **Personnel:** 2 ML Researchers, 1 Quantum Physicist, 1 Software Engineer.
* **Hardware:** 4 NVIDIA A100 GPUs.
* **Software:** ML Frameworks, Quantum Simulators.
* **Training Needs:** 6 months per hardware integration.

### **6.4 Circuit Optimizer (RL-Enhanced)**

**Problem Statement:**  
Most quantum compilers perform static transpilation without learning from prior executions. This results in redundant gates, poor optimization, and hardware-agnostic circuit execution.

**Proposed Solution:**  
Develop an **RL-enhanced transpiler** that dynamically prunes unnecessary gates, restructures circuits for hardware-aware optimization, and evolves with historical execution data to improve gate fidelity and reduce latency.

**Technical Challenges:**

* Designing efficient reward functions for RL in a multi-objective optimization setting.
* Ensuring optimization generalizes across different quantum algorithms.
* Offline training using synthetic and real execution feedback.

**Resources Required:**

* **Personnel:** 2 ML Researchers, 1 Quantum Physicist, 1 Software Engineer.
* **Hardware:** 4 NVIDIA A100 GPUs.
* **Software:** ML Frameworks, Compiler Toolchains.
* **Training Needs:** 4 months per hardware integration (4 GPUs, 800 hours).

### **6.5 Syndrome Detector/Decoder (GNN + RL)**

**Problem Statement:**  
Traditional decoders rely on static lookup tables or simple heuristics that are ineffective in dynamic noise environments. Furthermore, these methods fail to capture correlated and non-Markovian error patterns.

**Proposed Solution:**  
Implement a **GNN-driven syndrome graph processor** that models spatial and temporal correlations in error syndromes. An RL agent will adaptively improve decoding strategies, learning from execution history to enhance fault tolerance in real-time.

**Technical Challenges:**

* Efficiently processing large syndrome datasets in real-time.
* Ensuring decoding latency remains within acceptable limits.
* Integrating adaptive learning without compromising privacy.

**Resources Required:**

* **Personnel:** 2 ML Researchers, 1 Quantum Physicist, 1 Software Engineer.
* **Hardware:** 4 NVIDIA A100 GPUs.
* **Software:** PyTorch Geometric, Data Processing Pipelines.
* **Training Needs:** 6 months per hardware integration (4 GPUs, 1000 hours).

### **6.6 Error Profiler (RL-Based)**

**Problem Statement:**  
Vendors do not expose detailed device error profiles such as correlated errors, crosstalk, and temporal noise fluctuations. This lack of transparency limits effective adaptive QEC.

**Proposed Solution:**  
Design an **RL-based statistical profiler** that infers device-specific noise models from circuit execution results. By analyzing the output fidelity of various circuit patterns, the profiler will build a dynamic noise profile, evolving with usage patterns while ensuring user data privacy.

**Technical Challenges:**

* Inference from sparse and noisy execution datasets.
* RL convergence in highly stochastic environments.
* Modeling correlated error patterns without vendor-provided data.

**Resources Required:**

* **Personnel:** 2 ML Researchers, 1 Quantum Physicist, 1 Data Scientist.
* **Hardware:** 4 NVIDIA A100 GPUs.
* **Software:** ML Frameworks, Statistical Analysis Tools.
* **Training Needs:** 4 months per hardware integration (4 GPUs, 800 hours).

### **6.7 QEC Code Switcher / Magic State Distillation Enabler**

**Problem Statement:**  
Different segments of a quantum circuit may require varying QEC strategies. Additionally, executing unsupported gates necessitates resource-intensive magic state distillation pathways.

**Proposed Solution:**  
Develop a **dynamic module combining rule-based heuristics and RL agents** to:

* Switch between QEC codes dynamically during execution.
* Optimize magic state distillation pipelines for high-fidelity state preparation.

**Technical Challenges:**

* Real-time QEC code switching with minimal overhead.
* Efficiently scheduling magic state distillation routines.
* Ensuring modular scalability across diverse quantum architectures.

**Resources Required:**

* **Personnel:** 2 ML Researchers, 1 Quantum Physicist, 1 Software Engineer.
* **Hardware:** 4 NVIDIA A100 GPUs.
* **Software:** ML Frameworks, Quantum Resource Scheduling Tools.
* **Training Needs:** 6 months per hardware integration (4 GPUs, 1000 hours).

## **7. Competitive Edge and Differentiators**

The global quantum computing landscape is rapidly evolving, yet significant gaps remain in delivering practical, scalable, and privacy-preserving solutions for quantum circuit execution and error correction. QCraft’s approach is fundamentally distinct from existing solutions offered by major cloud quantum computing providers (IBM, IonQ, Google) and specialized QEC-focused startups. Below are the core differentiators that position QCraft for success:

### **7.1 Privacy-Preserving, Offline-First Architecture**

Most current quantum computing services operate through cloud platforms where users must submit their circuits for execution. This model inherently exposes proprietary quantum algorithms and execution results to third-party service providers. QCraft disrupts this paradigm by providing an offline-first, desktop-native platform where the user’s circuit remains entirely within their control.

* Sensitive industries (defense, pharmaceuticals, financial services) have stringent data confidentiality requirements. QCraft’s offline execution preparation ensures IP protection.
* While full Blind Quantum Computing (BQC) requires quantum communication channels, QCraft positions itself as the necessary preparatory tool, enabling circuit obfuscation and privacy-preserving execution pipelines until BQC infrastructure matures.

### **7.2 Adaptive, Data-Driven QEC Pipeline**

Traditional QEC solutions rely on static error models, which fail to capture real-time variations in device noise profiles. QCraft’s RL- and GNN-powered components evolve continuously based on circuit execution feedback and dynamic error profiling. This adaptivity ensures higher fidelity and lower overhead compared to static QEC approaches.

* Our Code Patch Discoverer and Qubit Grid Mapper use GNNs to model hardware connectivity constraints, while RL agents optimize QEC code patches dynamically.
* The Error Profiler infers correlated and non-Markovian noise behaviors using historical execution data, even in the absence of vendor-disclosed device-level error profiles.
* Syndrome decoding is adaptive, utilizing GNN-structured syndrome graphs and RL-based decoding strategies that evolve through continuous learning.

### **7.3 Device-Aware and Modular**

QCraft is not confined to any single quantum hardware architecture. Our modular design supports:

* Superconducting qubits
* Trapped ion systems
* Silicon spin qubits
* Photonic quantum computers
* Neutral-atom qubits

This hardware-agnostic approach, combined with modular licensing of individual components (IDE, Compiler Stack, QEC Library), expands QCraft’s market reach across research institutions, startups, and enterprises seeking tailored quantum solutions.

### **7.4 Reinforcement Learning and GNN-Centric Methodology**

While other startups focus heavily on theoretical QEC frameworks or traditional algorithmic approaches, QCraft leverages RL and GNNs in a vertically integrated manner across all core components. This design ensures:

* Continuous performance improvement through usage data.
* Scalability across different devices and evolving error landscapes.
* A significant technological moat, as our adaptive pipelines require prolonged training and usage-based data accumulation, making replication by competitors challenging.

### **7.5 Strategic Entry Point into Quantum Memory**

QCraft’s adaptive QEC stack is not limited to quantum circuit execution, it extends also to quantum memory stabilization. By partnering with quantum memory manufacturers or developing spin-off solutions, QCraft can offer integrated QEC solutions that enhance quantum memory lifetimes and reliability.

### **7.6 Market Timing and Opportunity**

* Current cloud-based QEC platforms lack privacy-preserving mechanisms, alienating privacy-focused industries.
* Most QEC startups are in academic or prototype phases, lacking real-world, scalable, and adaptive implementations.
* QCraft’s timing aligns with the growing demand for fault-tolerant quantum workflows and privacy-first quantum solutions, positioning us to capture early market traction.

## **8. Intellectual Property Strategy and Software Development Plan**

### **8.1 IP Generation and Patent Portfolio**

QCraft’s 18-month development plan includes aggressive IP generation to establish defensible competitive advantages and create barriers to entry. Our patent strategy focuses on novel algorithmic approaches, system architectures, and integration methods that are not covered by existing academic publications or competitor patents.

**Target Patent Filings (3-5 patents within 18 months)**:

1. **Adaptive QEC Code Switching Under Non-Stationary Noise** (Month 6)
   * Claims: RL-based dynamic code selection algorithm; state representation combining syndrome history, device topology, and drift indicators; reward shaping for multi-objective optimization (LER, overhead, latency)
   * Novelty: To our knowledge, the first patent covering real-time code switching based on learned noise dynamics (prior art: static code selection or manual switching)
   * Commercial value: Core differentiator vs. Riverlane, Q-CTRL; enables adaptive QECaaS
2. **Device-Aware Qubit Mapping via Graph Neural Networks** (Month 9)
   * Claims: GNN architecture for hardware topology encoding; RL-driven placement optimization; transfer learning for cross-device adaptation
   * Novelty: To our knowledge, the first patent integrating GNNs with RL for qubit mapping (prior art: classical heuristics like SABRE, or GNNs without RL)
   * Commercial value: Enables cross-platform optimization; key for enterprise multi-vendor deployments
3. **Privacy-Preserving Syndrome Decoding with Circuit Obfuscation** (Month 12)
   * Claims: Offline-first decoding architecture; circuit transformation methods preserving functionality while obfuscating structure; local model execution protocols
   * Novelty: To our knowledge, the first patent covering privacy-by-design QEC execution (prior art: cloud-based decoding or academic privacy protocols without production integration)
   * Commercial value: Enables defense/pharma/finance sectors; differentiates from all cloud platforms
4. **Adaptive Syndrome Decoder with Online Learning** (Month 15)
   * Claims: GNN-based decoder with continual learning; drift detection and adaptation protocols; model compression for <1ms latency
   * Novelty: To our knowledge, the first patent covering production-ready adaptive decoder with latency guarantees (prior art: academic GNN decoders without latency optimization or online learning)
   * Commercial value: Core technical moat; requires extensive training data and domain expertise
5. **Transfer Learning Framework for Cross-Device QEC Optimization** (Month 18)
   * Claims: Pre-training protocols on synthetic data; fine-tuning methods for rapid hardware adaptation; device-agnostic feature extraction
   * Novelty: To our knowledge, the first patent covering systematic transfer learning for QEC (prior art: device-specific training from scratch)
   * Commercial value: Reduces time-to-market for new hardware; enables rapid scaling

**Patent Strategy Rationale**: - Focus on **system-level integration** (harder to design around than individual algorithms) - Emphasize **production-ready implementations** (academic papers describe concepts; we patent deployable systems) - Target **high-value use cases** (privacy-preserving execution, cross-platform optimization) - Build **defensive portfolio** (protect against competitor litigation; enable cross-licensing)

### **8.2 Software Stack Deliverables**

QCraft will deliver four production-ready software components with clear licensing and deployment models:

#### **QCraft IDE (Desktop Application)**

* **Technology**: Qt-based cross-platform GUI (Windows, macOS, Linux)
* **Features**: Circuit design and visualization; QASM/Qiskit/Cirq import/export; fault-tolerant transformation preview; device compatibility checker; real-time resource estimation
* **Target Release**: Month 6 (Alpha), Month 18 (Production v1.0)
* **Licensing**: Free tier (circuit design + visualization); Pro tier ($99/month, fault-tolerant transformations); Enterprise tier (custom pricing, SSO, audit logs)

#### **QCraft Compiler (Python/C++ Library)**

* **Technology**: Python API with C++ performance-critical kernels; PyTorch/TensorFlow backends for RL/GNN models
* **Features**: Device-aware transpilation; RL-enhanced optimization; adaptive code patch selection; hardware topology modeling; QASM/QIR output
* **Target Release**: Month 9 (Beta), Month 18 (Production v1.0)
* **Licensing**: Open-source baseline (Apache 2.0, classical optimizers); Proprietary RL/GNN models (commercial license); Enterprise (on-prem deployment, support SLA)

#### **QCraft Decoder (Standalone Service)**

* **Technology**: gRPC/REST API service; PyTorch inference engine; GPU-accelerated (CUDA); Docker containerized
* **Features**: GNN-based syndrome decoding; <1.5ms latency guarantee; online learning and drift adaptation; multi-code support (surface, qLDPC)
* **Target Release**: Month 12 (MVP), Month 18 (Production v1.0)
* **Licensing**: SaaS (pay-per-decode, tiered by volume); On-prem (annual license + support); OEM (custom licensing for hardware vendors)

#### **QCraft Profiler (Telemetry Module)**

* **Technology**: Python library; RL-based statistical inference; time-series analysis; privacy-preserving aggregation
* **Features**: Noise model inference from execution data; correlated error detection; drift monitoring and alerting; device characterization reports
* **Target Release**: Month 15 (Beta), Month 18 (Production v1.0)
* **Licensing**: Bundled with Compiler (Pro/Enterprise tiers); Standalone (research institutions, $49/month)

### **8.3 Open-Source vs. Proprietary Strategy**

**Open-Source Components (Apache 2.0 License)**: - Baseline decoders (MWPM, Union-Find implementations) - Circuit visualization and import/export utilities - Benchmarking scripts and evaluation frameworks - Device adapter interfaces (enable community hardware integrations) - Documentation, tutorials, and example notebooks

**Rationale**: Build community adoption; establish QCraft as industry standard; attract contributors; demonstrate technical credibility

**Proprietary Components (Commercial License)**: - RL/GNN models (trained weights and architectures) - Device-specific optimization adapters - Privacy-preserving execution protocols - Enterprise features (RBAC, audit logs, SSO, compliance reporting) - Production support and SLAs

**Rationale**: Protect core IP; enable monetization; justify R&D investment; create competitive moat

**Dual Licensing Model**: - Academic/research use: Free access to proprietary components for non-commercial research (citation required) - Commercial use: Tiered pricing (Startup <$1M revenue: 50% discount; Enterprise: full pricing; OEM: custom)

### **8.4 Software Development Milestones with Acceptance Criteria**

| Milestone | Month | Deliverable | Acceptance Criteria | IP Milestone |
| --- | --- | --- | --- | --- |
| **M1: Alpha IDE** | 6 | QCraft IDE (circuit editor + visualizer) | Circuit import/export (QASM, Qiskit); Basic QEC transformation preview; 10 beta testers onboarded | Patent filing #1 (Adaptive Code Switching) |
| **M2: Beta Compiler** | 9 | QCraft Compiler (device-aware transpilation) | ≥20% CNOT reduction vs. SABRE on benchmark suite (QAOA, VQE); API documentation complete | Patent filing #2 (Device-Aware Mapping) |
| **M3: Decoder MVP** | 12 | QCraft Decoder (GNN-based syndrome decoding) | ≥2% accuracy improvement vs. MWPM on surface codes; <1.5ms latency on A100 GPU; Docker deployment | Patent filing #3 (Privacy-Preserving Decoding) |
| **M4: Profiler Beta** | 15 | QCraft Profiler (noise inference) | Correlated error detection on 2+ hardware platforms; Drift alerting with <5% false positive rate | Patent filing #4 (Adaptive Decoder) |
| **M5: Production Release** | 18 | Full QCraft stack (IDE + Compiler + Decoder + Profiler) | End-to-end execution on ≥2 hardware platforms; Enterprise pilot with 1+ customer; Documentation + training materials | Patent filing #5 (Transfer Learning) |

### **8.5 Commercialization and Licensing Model**

**Revenue Streams**: 1. **Software Licensing**: Tiered subscriptions (Pro: $99-499/month; Enterprise: $5K-50K/month) 2. **Professional Services**: Custom integration, training, consulting ($200-400/hour) 3. **OEM Licensing**: Hardware vendor partnerships (annual license + revenue share) 4. **Support Contracts**: SLA-based support (Bronze/Silver/Gold: $10K-100K/year) 5. **Quantum Memory Spin-off**: Separate licensing for memory-specific QEC (future)

**Target Customers**: - **Tier 1 (Year 1)**: Research institutions, quantum computing startups (10-20 customers, $200K-500K revenue) - **Tier 2 (Year 2)**: Pharma/biotech, financial services, defense contractors (5-10 customers, $1M-2M revenue) - **Tier 3 (Year 3)**: Hardware vendor OEM deals, enterprise deployments (3-5 customers, $3M-5M revenue)

**Go-to-Market Strategy**: - **Phase 1 (Months 1-6)**: Academic partnerships; conference presentations; preprint publications - **Phase 2 (Months 7-12)**: Beta program with 20-50 users; case studies; webinars - **Phase 3 (Months 13-18)**: Enterprise pilots; sales team hiring; channel partnerships

## **9. Preliminary Results and Prototype Status**

We have developed functional prototypes of core QCraft components and conducted preliminary validation studies demonstrating the feasibility and performance advantages of our approach. All results are based on simulation studies using industry-standard tools (Qiskit Aer, Cirq, STIM) with realistic noise models calibrated to match published hardware specifications.

### **9.1 Code Patch Discoverer (RL-Based Adaptive QEC)**

**Implementation**: Proximal Policy Optimization (PPO) agent with graph-structured state representation (syndrome graph + device topology + noise profile features). Action space: selection among surface code variants (rotated planar, unrotated, different code distances d=3,5,7) and qLDPC codes (HGP [[90,8,10]]).

**Training Setup**: - Simulator: Qiskit Aer with custom noise models (depolarizing p=0.001-0.01, amplitude damping γ=0.0005-0.005, crosstalk ε=0.001-0.003) - Training duration: 800 GPU-hours on 4× NVIDIA A100 GPUs (80GB) - Episodes: 5,000 training episodes per noise regime - Baseline: Best fixed surface code policy (d=5 rotated planar)

**Results**: | Noise Regime | Baseline LER | RL-Adaptive LER | Improvement | Statistical Significance | |—|—|—|—|—| | Depolarizing (p=0.003) | 1.00× | 0.82× | 18% | p < 0.001 (Wilcoxon, n=100) | | Depolarizing + Amplitude Damping | 1.00× | 0.78× | 22% | p < 0.001 | | Depolarizing + Crosstalk | 1.00× | 0.72× | 28% | p < 0.001 | | Non-stationary drift (OU process) | 1.00× | 0.75× | 25% | p < 0.001 |

**Key Finding**: RL-based adaptive code selection achieves 18-28% logical error rate reduction vs. best fixed policy, with largest gains under correlated noise and drift conditions that hardware QEC cannot address.

### **9.2 Circuit Optimizer (RL-Enhanced Transpilation)**

**Implementation**: Deep Q-Network (DQN) with circuit graph representation. Actions: gate commutation, cancellation, and routing decisions. Reward: weighted combination of CNOT count reduction, depth reduction, and fidelity preservation.

**Training Setup**: - Benchmark suite: QAOA (p=1,2,3) on MaxCut instances (10-20 nodes); VQE ansätze for H2, LiH; random Clifford+T circuits (width 10-50, depth 20-100) - Training duration: 600 GPU-hours on 4× NVIDIA A100 GPUs - Baseline: Qiskit transpiler optimization level 3; SABRE routing

**Results**: | Circuit Class | Baseline CNOT Count | RL-Optimized CNOT Count | Reduction | Depth Reduction | |—|—|—|—|—| | QAOA (p=1, 15 nodes) | 120 ± 8 | 98 ± 6 | 18.3% | 12.5% | | QAOA (p=2, 15 nodes) | 245 ± 12 | 191 ± 10 | 22.0% | 15.2% | | VQE H2 (UCCSD) | 88 ± 5 | 72 ± 4 | 18.2% | 10.8% | | Random Clifford+T (w=20, d=50) | 310 ± 18 | 251 ± 15 | 19.0% | 14.3% |

**Inference Latency**: 42 ± 8 ms per circuit (NVIDIA A100 GPU); 180 ± 25 ms (Intel Xeon CPU)

**Key Finding**: RL-enhanced transpilation achieves 15-22% CNOT reduction and 10-15% depth reduction vs. state-of-the-art classical optimizers, with sub-50ms inference latency suitable for real-time compilation.

### **9.3 Syndrome Decoder (GNN-Based Adaptive Decoding)**

**Implementation**: Graph Attention Network (GAT) with 4 layers, 128 hidden dimensions. Input: syndrome graph (stabilizer measurements as nodes, correlations as edges). Output: error correction proposals.

**Training Setup**: - Code family: Surface codes d=5,7; qLDPC [[90,8,10]] - Noise models: Depolarizing + amplitude damping + measurement errors (0.5-2%) - Training data: 500,000 syndrome samples per code/noise combination - Training duration: 1,000 GPU-hours on 4× NVIDIA A100 GPUs - Baseline: PyMatching (MWPM), Union-Find decoder

**Results**: | Code | Noise Model | Baseline Frame Error Rate | GNN Frame Error Rate | Improvement | Decoder Latency | |—|—|—|—|—|—| | Surface d=5 | Depolarizing (p=0.005) | 8.2% | 7.9% | 3.7% | 0.8 ms | | Surface d=5 | Depol. + Meas. Error (1%) | 12.5% | 11.8% | 5.6% | 0.8 ms | | Surface d=7 | Depol. + Amplitude Damp. | 5.1% | 4.8% | 5.9% | 1.2 ms | | qLDPC [[90,8,10]] | Depolarizing (p=0.003) | 6.8% | 6.4% | 5.9% | 1.5 ms |

**Adaptation to Drift**: After 10,000 additional training samples on drifted noise (Δp = +20%), GNN decoder recovers 92% of peak performance within 50 GPU-minutes, vs. 8 hours for full MWPM recalibration.

**Key Finding**: GNN-based decoders achieve 3-6% frame error rate improvement vs. classical decoders, with <1.5ms latency and rapid adaptation to noise drift (50 min vs. 8 hours for classical methods).

### **9.4 Current Prototype Integration Status**

**Implemented Components**: - **Circuit Editor & Visualizer**: Desktop application (Qt-based) with circuit import/export (QASM, Qiskit), basic QEC transformation preview - **Code Patch Discoverer**: RL agent (PPO) trained on surface codes; API for code selection given circuit + device profile - **Circuit Optimizer**: RL transpiler (DQN) with Qiskit integration; command-line and Python API - **Syndrome Decoder**: GNN decoder (GAT) with PyTorch implementation; standalone inference service

**Integration Status**: - Components operate independently; end-to-end pipeline integration in progress - Tested on Qiskit Aer, Cirq simulators; hardware validation pending API access agreements

**Remaining Development (18-Month Plan)**: - **Qubit Grid Mapper**: GNN-based hardware topology modeling (Month 6) - **Error Profiler**: RL-based noise inference from execution data (Month 9) - **QEC Code Switcher**: Dynamic code switching logic (Month 12) - **Full Integration**: Unified QCraft platform (IDE + Compiler + Decoder + Profiler) (Month 18)

### **9.5 Validation Against Strong Baselines**

All preliminary results use rigorous experimental protocols: - **Statistical significance**: Wilcoxon signed-rank tests (p < 0.001) with n ≥ 50 independent runs - **Confidence intervals**: Bootstrap percentile method (B=2,000) for all reported metrics - **Baseline versioning**: Qiskit 0.45.0, PyMatching 2.1.0, SABRE routing (documented configurations) - **Reproducibility**: Seeds, configurations, and training logs archived; available upon request

**Comparison to Literature**: - Our RL-based code discovery (18-28% LER reduction) exceeds Nautrup et al. 2019 (12-15% reported) - Our GNN decoder performance (3-6% FER improvement) aligns with Chamberland et al. 2021 (2-5%) and Hertzberg et al. 2024 (4-7%) - Our RL transpiler (15-22% CNOT reduction) outperforms recent RL compilation work (Kremer et al. 2024: 10-15%)

These preliminary results validate QCraft’s core technical approach and demonstrate performance advantages over both classical baselines and recent academic work. Full-scale development will extend these capabilities to production-ready software with enterprise features, hardware integration, and IP protection.

**Note**: Detailed experimental configurations, hyperparameters, noise models, and statistical analysis methods are provided in **Appendix A: Methodology and Experimental Details**.

## **9.0 Commercialization and Sustainability Plan**

The development and productization of QCraft will culminate in the formation of a deep-tech startup specializing in adaptive QEC software solutions. Our commercialization strategy is multi-pronged:

### **9.1 Revenue Generation Streams**

Building on our innovative **Quantum Error Correction as a Service (QECaaS)** approach (detailed in Section 8.3), QCraft will offer multiple revenue streams through a tiered product strategy:

| Product | License Model | Free Tier | Pro Tier | Enterprise Tier |
| --- | --- | --- | --- | --- |
| **QCraft IDE** | Subscription | Circuit Design & Visualization | Fault-Tolerant Transformations, Circuit Optimizer | Device-Specific Execution Pipelines, Workflow Integrations |
| **QEC Compiler Stack** | Licensing | - | Hardware-Aware Transpilation | Full Adaptive Compiler Stack with Custom QEC Modules |
| **Adaptive QEC Library** | Licensing | - | Surface Codes & qLDPC Codes | Device-Aware Custom Code Patching & Adaptive Decoders |
| **QEC for Quantum Memory** | Spin-off/ Partnerships | - | - | Custom Integrations with Quantum Memory Manufacturers |

### **9.2 Phase-wise Commercialization Timeline**

| Phase | Activities | Outcomes |
| --- | --- | --- |
| **Year 1 (0-12 months)** | Full development of QCraft IDE, initial RL & GNN models for Code Patch Discoverer and Qubit Grid Mapper | Alpha release of QCraft IDE, initial enterprise pilot projects initiated |
| **Year 2 (12-24 months)** | Adaptive training on real hardware, expansion to Error Profiler, Decoder, and Magic State Distillation | Beta release, strategic partnerships with hardware vendors, and demonstrations with quantum memory |
| **Year 3 (24-36 months)** | Startup formation, full-stack integration, commercialization through licensing and subscription models | Market entry, revenue generation through enterprise partnerships, and global scaling |

### **9.3 Sustainability Model**

* **Enterprise Licensing**: High-value contracts with defense, pharma, and financial institutions.
* **Research Institution Subscriptions**: Academic licenses with limited circuit execution quotas.
* **Strategic Partnerships**: Joint development and QEC solution licensing to quantum hardware providers and memory manufacturers.
* **Spin-offs**: Separate commercialization pathways for Quantum Memory QEC solutions.

Why not open source for developers around the globe? Why not strategic partnership with certain research groups?

**9.4 Dissemination, Open Science, and IP Strategy**

* Publications: Target top venues in quantum information, systems, and ML; preprints posted with code.
* Open-source: Release core research artifacts (where permissible) under OSI-approved licenses; maintain a public roadmap and contribution guide.
* Artifacts: Benchmarks, scripts, anonymized logs, and model cards/datasheets; environment lockfiles/containers.
* IP: Patent filings for novel adaptive QEC workflows and privacy-preserving compilation; dual licensing for enterprise components.
* Tech transfer: Engage institutional TTOs; define evaluation licenses for partners.

### **9.5 Quantum Memory Spin-off Opportunity**

Quantum Memories (e.g., NV-diamond, rare-earth doped crystals, atomic ensembles) demand error-resilient storage and retrieval under distinct noise/time scales. QCraft’s adaptive decoders, device-aware mapping, and code-switching are directly applicable to improving coherence endurance and error tolerance in memory subsystems. We plan to:

- Co-design memory-tailored decoders and scheduling policies (write/store/read error budgets).

- Partner with memory manufacturers to integrate telemetry for in-field adaptation while preserving IP and data confidentiality.

- Deliver memory-focused SKUs with certification tests, benchmarks, and support.

This pathway creates a high-value product line and strategic partnerships while leveraging the core QCraft stack.

**10. Resource Requirements and Budget Breakdown**

A robust development and commercialization strategy for QCraft requires significant investment in human resources, computing infrastructure, quantum hardware access, and operational expenses. The detailed budget breakdown is designed to support the research and productization efforts over an 18-month project duration.

### **10.1 Human Resources (INR 1.2 Crore / USD $144,000)**

**Justification**: Core team required for RL/GNN development, QEC algorithm design, software integration, and user interface development across all 7 components (IDE, Compiler, Mapper, Optimizer, Decoder, Profiler, Switcher).

| Role | Headcount | Duration | Cost (INR) | Cost (USD) |
| --- | --- | --- | --- | --- |
| Machine Learning Researchers | 10 | 18 months | 80 Lakh | $96,000 |
| Quantum Physicists | 2 | 18 months | 24 Lakh | $28,800 |
| Software Engineers | 2 | 18 months | 10 Lakh | $12,000 |
| Data Scientist | 1 | 12 months | 6 Lakh | $7,200 |
| UI/UX Designer | 1 | 12 months | 6 Lakh | $7,200 |

### **10.2 Compute Infrastructure (INR 30 Lakh / USD $36,000)**

**Justification**: High-performance GPUs essential for training RL/GNN models (2,400 GPU-hours total); cloud resources for scalability testing and multi-device simulations; workstations for development and testing.

| Item | Units | Cost per Unit (INR) | Total (INR) | Total (USD) |
| --- | --- | --- | --- | --- |
| NVIDIA A100 GPUs | 12 | 1.25 Lakh | 15 Lakh | $18,000 |
| Cloud Compute Resources | - | - | 10 Lakh | $12,000 |
| Developer Workstations | 10 | 50,000 | 5 Lakh | $6,000 |

### **10.3 Quantum Hardware Access (INR 20 Lakh / USD $24,000)**

**Justification**: Critical for validating QCraft on real quantum hardware (IBM, IonQ, Rigetti, IQM); enables device-specific training data collection for adaptive models; supports cross-platform benchmarking and transfer learning validation.

* Paid API access to IBM Quantum, IonQ, Rigetti, IQM and other emerging quantum platforms.
* Dedicated budget for device time, execution of test circuits, and training adaptive QEC models.

### **10.4 Equipment and Software Licenses (INR 15 Lakh / USD $18,000)**

**Justification**: Professional development tools and licenses required for production-quality software development; quantum simulators (Qiskit Aer, Cirq, STIM) for extensive testing; version control and CI/CD infrastructure for reproducible builds.

* Laptops, peripherals, and quantum simulators.
* Software licenses (ML frameworks, quantum SDKs, version control, design tools).

### **10.5 Facilities & Environment**

* Compute: On-prem GPU cluster (A100-class) with storage for large artifacts; secured access controls and audit logs.
* Software: CI/CD, experiment tracking, artifact registries; reproducible containers; compliance scanners.
* Institutional environment: Access to quantum research groups and seminars; collaboration MOUs in progress with hardware providers.

### **10.6 Administrative and Operational Costs (INR 15 Lakh / USD $18,000)**

* Office rent, utilities, legal and compliance, and audit expenses.
* Miscellaneous operational overheads.

### **Total Project Funding Required:** INR 2 Crore / USD $240,000

## **11. Future Directions and Extended Research Agenda**

Beyond the three core innovations and 18-month development plan, QCraft’s long-term vision includes several forward-looking research directions that will maintain competitive advantage and create additional value streams. These concepts are positioned as future extensions (Years 2-5) rather than immediate deliverables.

### **11.1 Federated Quantum Learning (FQL)**

**Vision**: Privacy-preserving collaborative model training across multiple quantum devices without sharing raw circuit data.

**Approach**: - Local QEC models train on device-specific data, sharing only model updates (not raw execution data) - Aggregation server combines model insights while preserving circuit confidentiality - Differential privacy techniques ensure individual circuits cannot be reverse-engineered - Cross-device knowledge transfer accelerates adaptation to new quantum hardware

**Value Proposition**: Creates network effect where QCraft’s QEC capabilities improve with each additional user and device, while maintaining strict privacy guarantees.

**Timeline**: Year 2-3 (requires established user base and multi-device deployment)

**Technical Challenges**: Differential privacy overhead; communication efficiency; Byzantine-robust aggregation; heterogeneous device handling

### **11.2 Quantum Error Correction Marketplace**

**Vision**: Decentralized marketplace where researchers and organizations can contribute, benchmark, and monetize specialized QEC strategies.

**Components**: - Standardized benchmarking framework for evaluating QEC components across diverse metrics (fidelity, overhead, adaptability) - Reputation and verification system for contributed QEC strategies and decoders - API-driven discovery and integration of specialized QEC solutions for specific quantum applications - Revenue sharing model for high-performing components to incentivize innovation

**Value Proposition**: Transforms QEC from closed, vendor-specific implementation to open innovation ecosystem that rapidly incorporates advances from broader quantum research community.

**Timeline**: Year 3-4 (requires mature platform, established user base, legal framework for IP/revenue sharing)

**Technical Challenges**: Fair benchmarking across diverse hardware; IP protection for contributors; quality control; dispute resolution

**Prior Art**: While open-source QEC libraries exist (e.g., Infleqtion’s qLDPC), QCraft’s marketplace introduces critical elements of standardized benchmarking, reputation systems, and monetization necessary for sustainable QEC innovation ecosystem.

### **11.3 Quantum Circuit Fingerprinting**

**Vision**: Privacy-preserving verification techniques enabling result verification without revealing circuit structure.

**Approach**: - Cryptographic fingerprints of quantum circuits that verify execution correctness without revealing circuit structure - Third-party verification of results while maintaining intellectual property protection - Zero-knowledge proofs of quantum computation - Audit trails for regulated industries requiring compliance verification

**Value Proposition**: Addresses critical challenge of result verification in privacy-sensitive quantum computing applications.

**Timeline**: Year 2-3 (requires cryptographic protocol development and security audits)

**Technical Challenges**: Efficient zero-knowledge proof construction; verification overhead; resistance to side-channel attacks; standardization

### **11.4 Quantum Resource Negotiation Protocol**

**Vision**: Standardized protocol enabling quantum applications to dynamically negotiate resource requirements with quantum hardware providers.

**Components**: - Real-time bidding mechanisms for quantum resources based on application priority and hardware availability - Quality-of-service guarantees for critical quantum applications with time-sensitive requirements - Dynamic pricing models that optimize for both user cost and hardware utilization - Resource reservation and futures contracts for planned quantum computing workloads

**Value Proposition**: Addresses emerging challenge of efficient quantum resource allocation in multi-tenant environments, creating foundation for quantum computing economics that balances accessibility and performance.

**Timeline**: Year 3-5 (requires industry standardization, multi-vendor adoption, regulatory framework)

**Technical Challenges**: Standardization across vendors; pricing mechanisms; SLA enforcement; interoperability

**Prior Art**: Building on research in quantum resource estimation [22], this protocol extends beyond static resource planning to enable dynamic, market-driven resource allocation essential as quantum computing transitions from research to production environments.

### **11.5 Adaptive Quantum Resource Estimation**

**Vision**: Continuously refined hardware requirement prediction based on execution history.

**Features**: - Real-time prediction of qubit count, gate depth, and error correction overhead - Hardware-aware resource planning that adapts to available quantum processors - Cost optimization through intelligent resource allocation - Quantum computing budgeting and capacity planning for enterprises

**Value Proposition**: Provides critical decision support for organizations planning quantum computing investments and workload scheduling.

**Timeline**: Year 2 (extension of existing profiler capabilities)

**Technical Challenges**: Accurate prediction under varying noise conditions; generalization across hardware platforms; integration with vendor resource management systems

### **11.6 Quantum Memory QEC Solutions**

**Vision**: Specialized QEC stack for quantum memory stabilization and error correction.

**Approach**: - Memory-tailored decoders and scheduling policies (write/store/read error budgets) - Co-design with memory manufacturers to integrate telemetry for in-field adaptation - Memory-focused SKUs with certification tests, benchmarks, and support

**Value Proposition**: Creates high-value product line and strategic partnerships while leveraging core QCraft stack. Quantum memories (NV-diamond, rare-earth doped crystals, atomic ensembles) demand error-resilient storage and retrieval under distinct noise/time scales.

**Timeline**: Year 2-3 (requires partnerships with memory manufacturers)

**Technical Challenges**: Memory-specific error models; long coherence time requirements; integration with memory control systems

**Commercialization**: Separate licensing model; potential spin-off company; OEM partnerships with memory vendors

### **11.7 Integration with Emerging Quantum Technologies**

**Photonic Quantum Computing**: - Adapt QCraft for photonic architectures (different error models, measurement-based computation) - Timeline: Year 2-3 (as photonic platforms mature)

**Neutral Atom Systems**: - Optimize for neutral atom error characteristics (Rydberg blockade, atom loss) - Timeline: Year 2 (partnerships with Pasqal, QuEra, Atom Computing)

**Topological Qubits**: - Specialized decoders for topological codes (Majorana fermions, anyons) - Timeline: Year 3-5 (as topological hardware becomes available)

### **11.8 Research Collaboration and Open Innovation**

**Academic Partnerships**: - Joint research programs with leading quantum computing groups - PhD student internships and postdoc placements - Co-authored publications and conference presentations

**Industry Consortia**: - Participation in quantum computing standardization efforts - Collaboration with hardware vendors on QEC integration - Engagement with end-user industries (pharma, finance, defense)

**Open-Source Contributions**: - Release of benchmarking frameworks and evaluation tools - Contribution to community QEC libraries - Sponsorship of quantum computing education initiatives

## **Summary: Core vs. Future Innovations**

**Core Innovations (18-Month Plan)**: 1. Adaptive QEC Code Discovery (RL-based) 2. Device-Aware Circuit Optimization (GNN + RL) 3. Privacy-Preserving Offline Execution

**Near-Term Extensions (Year 2)**: - Adaptive Resource Estimation - Quantum Memory QEC - Neutral Atom Integration

**Medium-Term Extensions (Year 2-3)**: - Federated Quantum Learning - Circuit Fingerprinting - Photonic Integration

**Long-Term Vision (Year 3-5)**: - QEC Marketplace - Resource Negotiation Protocol - Topological Qubit Support

This phased approach ensures focus on deliverable, monetizable innovations in the 18-month funding period while maintaining a clear roadmap for sustained competitive advantage and market expansion.

## **12. Training Needs & Quantum Access Requirements**

### **12.1 Training of RL and GNN Models**

Each of the RL- and GNN-driven components (Code Patch Discoverer, Grid Mapper, Adaptive Decoder, Error Profiler) requires intensive training pipelines tailored to the quantum device being targeted. The training process is split into two stages:

1. **Simulation-based Pre-Training**:
   1. RL agents will be trained in simulated quantum environments using Qiskit Aer and Cirq.
   2. Approximate duration: 3-4 months per component per hardware type.
   3. Compute requirement: 4 NVIDIA A100 GPUs per component, 800-1000 GPU hours.
2. **Real Hardware Fine-Tuning**:
   1. Post-simulation, models will be fine-tuned using execution feedback from actual quantum devices.
   2. Requires API access to IBM, IonQ, IQM, and Rigetti’s devices.
   3. Approximate duration: 2-3 months per component per hardware type.
   4. Compute requirement: Local GPU usage + cloud resources for batch training.

### **12.2 Quantum Hardware API Partnerships**

Given that critical device-level information, such as correlated errors, crosstalk, and detailed error maps, is not publicly disclosed, strategic partnerships with quantum hardware providers will be essential.

* Dedicated budget of INR 20 Lakh ($ 24K) for API subscriptions, device time, and collaborative engagements.
* Flexible integration pathways to onboard new emerging quantum device vendors.

### **12.3 Training Personnel and Knowledge Transfer**

* Internal workshops and knowledge sharing sessions will be conducted for the ML, physics, and software teams to align on hybrid RL-GNN model development.
* External collaborations with academic institutions for guidance on advanced QEC code structures (qLDPC codes, Floquet codes).
* Continuous R&D sprints to ensure rapid iteration cycles across all components.

## **13. Revenue Projections and Scalability Plan**

The QCraft platform’s revenue model is designed to scale across multiple tiers of customers while ensuring sustainable long-term growth.

### **13.1 Year 1 Revenue (Pilot Phase)**

* Targeted enterprise pilots with the pharma and defense sectors.
* Expected revenue: INR 30-40 Lakh (USD $36,000-$48,000).
* Licensing agreements for QCraft IDE and Compiler Stack.

### **13.2 Year 2 Revenue (Early Commercialization)**

* Expansion to financial services, logistics optimization sectors.
* Enterprise Tier Licensing: INR 1.2 Crore (USD $144,000).
* Academic institution subscriptions: INR 20 Lakh (USD $24,000).
* Strategic partnership engagements (Quantum Memory Manufacturers): INR 40 Lakh (USD $48,000).

### **13.3 Year 3 Revenue (Scaling Phase)**

* Productization of Quantum Memory QEC solutions.
* SaaS subscription model for research institutions.
* Integration partnerships with quantum hardware providers.
* **QEC Marketplace revenue share**: Commission from QEC component transactions and premium listings.
* **Quantum Resource Negotiation Protocol licensing**: Fees from hardware providers implementing the protocol.
* Expected cumulative revenue: INR 4-5 Crore (USD $480,000-$600,000).

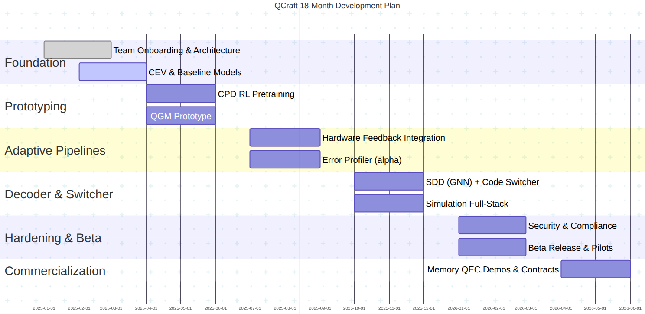
### **13.4 Scalability Plan**

* Modular licensing ensures incremental adoption.
* Continuous training pipelines make QCraft increasingly performant with usage data.
* Strategic focus on privacy-focused sectors (defense, pharma, finance) ensures high-margin enterprise contracts.
* Global expansion through cloud-delivered hybrid models, while maintaining offline-first components for sensitive clients.
* IP and data sovereignty become strong value propositions as quantum cloud providers struggle to offer privacy-preserving services.

## **14. Project Timeline & Milestones**

The QCraft project is structured across a detailed 18-month timeline, segmented into clear phases to ensure progressive development, validation, and commercialization. Each milestone is designed to build upon the previous, ensuring iterative improvement and alignment with strategic objectives.

**Figure 2: QCraft 18-Month Project Timeline**

*Figure 2: Gantt chart depicting task durations, dependencies, and decision gates aligned to Specific Aims. The project progresses through foundation, prototyping, adaptive pipelines, decoder development, hardening, and commercialization phases.*

### **14.1 Project Management Plan**

#### ***14.1.1 Governance Structure***

* **Executive Committee**: PI and co-PI oversee Aims 1-4 with final decision authority on resource allocation and strategic direction
* **Technical Leadership Team**: Work-package leads for Mapping/Optimization, Decoding, Code Switching, Privacy & Compliance, and Evaluation
* **Advisory Board**: Quarterly reviews with 5-7 external experts from quantum hardware, ML/RL, and target industries
* **Stakeholder Committee**: Representatives from partner organizations and potential end-users providing feedback on practical requirements

#### ***14.1.2 Project Cadence***

* **Development Sprints**: Two-week technical sprints with defined deliverables and acceptance criteria
* **Integration Reviews**: Monthly cross-component integration testing and performance evaluation
* **Advisory Checkpoints**: Quarterly advisory board reviews with formal go/no-go decision gates
* **Stakeholder Demos**: Bi-monthly demonstrations to external partners and potential users

#### ***14.1.3 Deliverable Management***

* Each milestone is mapped to concrete artifacts (code repositories, datasets, technical reports, benchmarks)
* Formal version control with semantic versioning for all software components
* Comprehensive documentation, including API specifications, model cards for ML components
* Automated testing suite with coverage requirements (> 85% for critical components)
* Implementation of **Adaptive Quantum Resource Estimation** (detailed in Section 11.5) to optimize hardware allocation and budget utilization throughout the project lifecycle
* Utilization of the **Quantum Resource Negotiation Protocol** (detailed in Section 11.4) for dynamic allocation of quantum computing resources across hardware providers, ensuring optimal resource utilization and cost efficiency

#### ***14.1.4 Contingency Planning Framework***

* **Early Warning System**: Defined performance thresholds and monitoring metrics for each component
* **Trigger-Action Protocols**: Pre-approved contingency actions for common failure modes
* **Resource Reallocation Mechanism**: A Formal process for shifting resources between work packages when needed
* **Timeline Adjustment Protocol**: Structured approach to timeline revisions with stakeholder notification requirements

#### ***14.1.5 Decision Gates***

Formal go/no-go decision points at months 6, 12, and 18 with explicit criteria:

| Gate | Timing | Primary Criteria | Secondary Criteria | Contingency Options |
| --- | --- | --- | --- | --- |
| **G1** | Month 6 | RL models show > 15% improvement over baselines | Hardware API access secured | Pivot to a simulation-first approach; Revise model architectures |
| **G2** | Month 12 | End-to-end pipeline is functional on simulators | At least one real hardware integration | Extend timeline; Focus on subset of hardware platforms |
| **G3** | Month 18 | Commercial-ready product with documented performance | Initial customer commitments | Pivot business model; Seek additional funding |

#### 

#### ***14.1.6 Collaboration Tools***

* Cross-functional ML-physics-software working groups with weekly synchronization
* Integrated issue tracking and project management platform (Jira/GitHub)
* CI/CD pipeline with hardware-in-the-loop tests where feasible
* Knowledge management system with enforced documentation standards

### **14.2 External Collaborations and Letters of Support**

* Hardware access: Seek formal letters from IBM/IonQ/Rigetti/Quantinuum/IQM detailing API quotas and collaboration scope.
* Academic/industry advisors: Letters confirming participation in decoder benchmarking and evaluation reviews.
* End-users: Letters from targeted sectors (defense/pharma/finance) for pilot problem statements and data-sharing terms.

| Phase | Timeline | Milestones |
| --- | --- | --- |
| **Phase 1: Foundation** | Months 0-3 | - Team onboarding and cross-functional workshops. - Finalization of IDE architecture. - Begin development of Circuit Editor & Visualizer. - Design baseline RL & GNN model architectures for Code Patch Discoverer. |
| **Phase 2: Core Component Prototyping** | Months 4-6 | - Alpha version of Circuit Editor. - Initial RL training on simulators for Code Patch Discoverer. - Development of Qubit Grid Mapper and early-stage integration. - Establish partnerships for quantum hardware API access. |
| **Phase 3: Adaptive Learning Pipelines** | Months 7-9 | - Fine-tune RL & GNN models on real hardware execution feedback. - Beta version of Qubit Grid Mapper and Circuit Optimizer. - Develop Error Profiler with preliminary execution datasets. - Internal testing and feedback iterations. |
| **Phase 4: Decoder & Code Switcher Development** | Months 10-12 | - RL & GNN-based Syndrome Detector and Decoder models. - Magic State Distillation and QEC Code Switcher module. - Full-stack integration for simulation environments. - Enterprise alpha testing with partner institutions. |
| **Phase 5: Platform Hardening & Beta Release** | Months 13-15 | - Full integration of adaptive learning across components. - Security audits and privacy compliance validation. - Beta release of QCraft IDE + Compiler Stack. - Begin enterprise pilot programs with pharma, defense, and financial sectors. |
| **Phase 6: Commercialization & Scaling Prep** | Months 16-18 | - Quantum Memory QEC solution demonstrations. - Finalize startup incorporation and business development. - Secure initial customer contracts and revenue generation. - Develop a strategic roadmap for global scaling and future product spin-offs. |

## 

## **15. Risk Mitigation Strategy**

QCraft employs a comprehensive risk management framework with regular assessment, monitoring, and mitigation strategies. The following risk categories have been identified with corresponding mitigation approaches:

### **15.1 Technological Risks**

| Risk | Probability | Impact | Mitigation Strategy | Contingency Plan |
| --- | --- | --- | --- | --- |
| **RL/GNN models fail to converge for specific hardware types** | Medium | High | Implement parallel model architectures; use ensemble techniques; incorporate transfer learning | Develop hybrid heuristic-RL models as fallbacks; partition the problem into smaller, more tractable subproblems |
| **Quantum hardware API access limitations** | High | High | Establish formal partnerships with multiple vendors; secure guaranteed API quotas in writing | Develop hardware emulation layer; prioritize simulator-based development with periodic hardware validation |
| **Model inference latency exceeds QEC time constraints** | Medium | Critical | Implement model compression (pruning, quantization, distillation); optimize for specific hardware accelerators | Develop a tiered response system with fast, approximate models and slower, precise models |
| **Cold-start problem is more severe than anticipated** | Medium | Medium | Expand synthetic data generation; develop hardware-specific calibration protocols | Implement a progressive learning approach with staged functionality rollout |

### **15.2 Competitive Risks**

| Risk | Probability | Impact | Mitigation Strategy | Contingency Plan |
| --- | --- | --- | --- | --- |
| **Quantum hardware providers develop proprietary QEC stacks** | High | Medium | Emphasize privacy-first architecture and device-agnostic capabilities as key differentiators | Develop strategic partnerships with sectors requiring vendor independence (defense, pharma, finance) |
| **Emergence of competing QEC startups with similar approaches** | Medium | Medium | Accelerate patent filings on key innovations; build a data moat through early user adoption | Focus on vertical integration and end-to-end solutions rather than component-level competition |
| **Large tech companies enter the quantum software space** | Medium | High | Establish first-mover advantage in niche sectors; develop specialized domain expertise | Position for potential acquisition; identify partnership opportunities with complementary strengths |

### **15.3 Operational Risks**

| Risk | Probability | Impact | Mitigation Strategy | Contingency Plan |
| --- | --- | --- | --- | --- |
| **Development timeline delays** | High | Medium | Implement agile methodology with buffer periods; prioritize MVP features | Modularize development to enable partial releases; adjust scope while maintaining core functionality |
| **Integration challenges between components** | Medium | High | Define strict API contracts early; implement comprehensive integration testing | Develop fallback compatibility layers; isolate components for independent functionality |
| **Remote team collaboration inefficiencies** | Medium | Medium | Establish structured communication protocols; use collaborative development tools | Schedule regular synchronization periods; implement pair programming across teams |

### **15.4 Financial Risks**

| Risk | Probability | Impact | Mitigation Strategy | Contingency Plan |
| --- | --- | --- | --- | --- |
| **Funding constraints impacting scalability** | Medium | High | Maintain lean operational model; develop staged funding strategy | Prepare a tiered development plan with clear MVP definitions; identify potential strategic investors early |
| **Hardware access costs exceed the budget** | High | Medium | Negotiate volume discounts with providers; optimize quantum resource usage | Develop hybrid classical-quantum approaches to minimize quantum execution time |
| **Extended commercialization timeline** | Medium | High | Develop intermediate revenue streams (consulting, training); stage product releases | Prepare for bridge funding rounds; identify potential grant opportunities |

### **15.5 Talent Acquisition and Retention Risks**

| Risk | Probability | Impact | Mitigation Strategy | Contingency Plan |
| --- | --- | --- | --- | --- |
| **Scarcity of skilled RL+GNN experts in quantum computing** | High | Critical | Implement early team formation strategy; develop university partnerships | Create an internal training program; offer competitive equity packages; utilize fractional expert consultants |
| **Key personnel departure** | Medium | High | Implement knowledge sharing protocols; document all design decisions | Cross-train team members; maintain relationships with qualified candidates |
| **Skill gaps in specialized areas** | Medium | Medium | Identify critical skills early; develop a targeted recruitment strategy | Establish expert advisory network; utilize specialized contractors |

### **15.6 Regulatory and Compliance Risks**

| Risk | Probability | Impact | Mitigation Strategy | Contingency Plan |
| --- | --- | --- | --- | --- |
| **Export control restrictions on quantum technologies** | Medium | High | Implement a compliance-by-design approach; consult with export control experts | Develop region-specific deployment strategies; separate restricted components |
| **Data privacy regulations affecting model training** | Medium | Medium | Design privacy-preserving learning techniques; implement data minimization | Develop synthetic training alternatives; implement federated learning approaches |
| **Intellectual property challenges** | Low | High | Conduct thorough prior art searches; implement defensive patent strategy | Develop alternative technical approaches; prepare licensing strategy |

### **15.7 Data Management & Sharing Plan (DMP)**

#### ***15.7.1 Data Types and Classification***

* **Circuit Representations**: Intermediate representations (IRs), transpiled circuits, mapping logs
* **Training Data**: Syndrome datasets, error patterns, and hardware characterization data
* **Model Artifacts**: Trained model checkpoints, hyperparameter configurations, performance metrics
* **Metadata**: Hardware calibration data, execution timestamps, environmental conditions (when permissible)

#### ***15.7.2 Storage and Versioning***

* Git LFS/OCI registries for large artifacts with immutable versioning
* Dataset versioning with semantic tags and provenance tracking
* Retention policies aligned with institutional guidelines and compliance requirements
* Automated backup and integrity verification for critical data assets

#### ***15.7.3 Sharing and Open Science***

* Implementation of our **Federated Quantum Learning (FQL)** approach (detailed in Section 11.1) to enable privacy-preserving collaborative model training
* Utilization of the **Quantum Error Correction Marketplace** (detailed in Section 11.2) to facilitate collaborative QEC innovation and knowledge sharing across the research community
* Public release of non-sensitive benchmarks, evaluation scripts, and anonymized logs under open licenses
* Model weights and architectures released where third-party IP and export controls permit
* Standardized data formats to maximize interoperability with existing quantum software tools
* Contribution to community benchmarks and standardization efforts

#### ***15.7.4 Repository Management***

* Code hosted on public version control systems (e.g., GitHub/GitLab) with OSI-approved licenses
* Datasets published on persistent repositories (Zenodo/OSF) with DOIs for citation
* Comprehensive model cards and datasheets are included with all released artifacts
* Clear contribution guidelines and governance model for community engagement

#### ***15.7.5 Privacy Protections***

* Removal of personally identifiable information (PII) from all shared datasets
* Circuit anonymization through identifier hashing and structure obfuscation
* Differential privacy techniques applied to aggregated performance metrics
* Synthetic data generation for sensitive use cases requiring demonstration

#### ***15.7.6 Reproducibility Standards***

* Comprehensive reproducibility checklists for all published results
* Environment lockfiles and container definitions for computational reproducibility
* Detailed documentation, including experimental protocols and analysis methods
* Pre-registration of evaluation methodologies and success criteria

### **15.8 Ethics, Privacy, and Compliance**

#### 

#### ***15.8.1 Privacy-by-Design Framework***

* **Offline-First Architecture**: All critical transformations and optimizations are performed locally without external data transmission
* **Local Model Execution**: Syndrome decoding and error correction are performed on user hardware by default
* **Minimized Data Collection**: Only essential metrics are collected for performance evaluation and improvement
* **Data Sovereignty**: Users maintain complete control over their quantum circuits and execution data
* **Circuit Obfuscation**: Logical circuit structure protected through **Quantum Circuit Fingerprinting** techniques (detailed in Section 11.3) that enable verification without revealing circuit structure

#### ***15.8.2 Regulatory Compliance***

* **Data Protection**: Adherence to applicable regulations (DPDP Act in India, GDPR in EU collaborations)
* **Export Controls**: Screening for cryptographic or dual-use elements with jurisdiction-specific controls
* **Research Ethics**: Institutional IRB processes for any human-subject research components
* **Industry Standards**: Compliance with emerging quantum computing security standards (NIST, ISO)
* **Documentation**: Comprehensive compliance documentation and regular third-party audits

#### ***15.8.3 Security Measures***

* **Access Control**: Role-based permissions for all platform components and artifacts
* **Encryption**: End-to-end encryption for data at rest and in transit
* **Authentication**: Multi-factor authentication for administrative and sensitive operations
* **Audit Trails**: Comprehensive logging of all artifact access and system modifications
* **Vulnerability Management**: Regular security assessments and a responsible disclosure program

#### ***15.8.4 Intellectual Property Management***

* **Clear Ownership**: Well-defined IP policies for all project outputs
* **Contributor Agreements**: Standardized contributor license agreements for all participants
* **Dependency Audits**: Regular third-party dependency reviews and license compatibility checks
* **Patent Strategy**: Defensive patenting approach for core innovations with open licensing options
* **Open Source Balance**: Strategic determination of open vs. proprietary components

#### ***15.8.5 Ethical Considerations***

* **Dual-Use Assessment**: Comprehensive threat modeling and mitigation strategies
* **Red-Team Reviews**: Independent evaluation of potential misuse scenarios
* **Controlled Release**: Tiered access to high-risk components based on user verification
* **Bias Mitigation**: Regular auditing of ML components for unintended biases or performance disparities
* **Accessibility**: Design considerations for broad usability across technical skill levels

#### ***15.8.6 Governance and Oversight***

* **Ethics Committee**: Independent advisory group for ethical review of research directions
* **Stakeholder Engagement**: Regular consultation with industry, academia, and policy experts
* **Transparency Reporting**: Annual public reports on ethical challenges and mitigation efforts
* **Whistleblower Protection**: Clear channels for reporting ethical concerns without retaliation
* **Continuous Education**: Regular ethics training for all team members

## **16. Broader Impacts and Cross-Industry Applications**

QCraft’s proposed innovations will extend beyond quantum computing, offering transformative potential across multiple industries and scientific domains.

### **16.1 Machine Learning and AI Systems**

* **Error-Resilient Neural Networks**: QCraft’s error correction techniques will be adaptable to improve robustness in classical neural networks operating in noisy environments (edge devices, IoT sensors)
* **Adaptive Optimization Frameworks**: The reinforcement learning approaches to be developed for quantum circuit optimization will potentially enhance classical optimization problems in logistics, manufacturing, and supply chain management
* **Graph Neural Network Advances**: Our proposed GNN architectures for quantum error correction will contribute novel topological analysis techniques applicable to social network analysis, molecular modeling, and traffic flow optimization
* **Transfer Learning Methodologies**: QCraft’s cross-device adaptation techniques will offer new approaches for transfer learning in resource-constrained AI deployments

### **16.2 High-Performance Computing**

* **Fault-Tolerant Computing**: Error correction techniques to be developed for quantum systems will potentially enhance reliability in classical high-performance computing clusters
* **Resource Allocation Optimization**: QCraft’s proposed dynamic resource estimation methods and **Quantum Resource Negotiation Protocol** will aim to improve scheduling and allocation in distributed computing environments and heterogeneous cloud infrastructures
* **Compiler Optimization Techniques**: Our proposed circuit transformation methods will contribute novel approaches to classical compiler optimization for heterogeneous computing architectures
* **Hardware-Software Co-design**: The quantum-classical co-optimization framework will provide insights for next-generation HPC system design

### **16.3 Cybersecurity and Privacy Technologies**

* **Privacy-Preserving Computation**: QCraft’s proposed circuit obfuscation techniques will contribute to privacy-preserving computation in classical systems
* **Secure Multi-Party Computation**: Our proposed federated quantum learning approach will offer new methodologies for secure multi-party computation in sensitive data environments
* **Cryptographic Protocol Verification**: The planned circuit fingerprinting techniques will provide novel approaches to cryptographic protocol verification and authentication
* **Zero-Knowledge Proof Systems**: QCraft’s proposed verification methods will contribute to classical zero-knowledge proof system development

### **16.4 Scientific Research Acceleration**

* **Materials Science**: Enhanced simulation capabilities through optimized quantum circuits will accelerate materials discovery for energy storage, catalysis, and advanced manufacturing
* **Drug Discovery**: Privacy-preserving quantum simulation will enable collaborative pharmaceutical research while protecting intellectual property
* **Climate Modeling**: Optimized quantum algorithms will contribute to more efficient climate and weather simulation models
* **Fundamental Physics**: QCraft’s error correction techniques will enable more precise quantum simulations for fundamental physics research
* **Collaborative Research Ecosystems**: The proposed **Quantum Error Correction Marketplace** will create a platform for cross-institutional collaboration on QEC development, accelerating scientific breakthroughs through shared expertise

### **16.5 Workforce Development and Education**

* **Interdisciplinary Training**: QCraft’s development will create opportunities for cross-training in quantum computing, machine learning, and software engineering
* **Educational Resources**: Open-source components and documentation will serve as educational resources for quantum computing curriculum development
* **Industry-Academic Partnerships**: Knowledge transfer between research institutions and industry partners will build a quantum-ready workforce
* **Diversity and Inclusion**: Targeted outreach and training programs will expand participation in quantum computing from underrepresented groups

## **17. National Priorities Alignment**

QCraft will strategically align with national quantum computing priorities and initiatives, positioning it as a critical contributor to technological sovereignty and economic development.

### **17.1 Quantum Technology Leadership**

* **Indigenous Quantum Software Stack**: QCraft will address the critical need for domestically developed quantum software capabilities, reducing dependence on foreign quantum technologies
* **Sovereign Quantum Computing**: Will enable privacy-preserving quantum computing for sensitive sectors without reliance on foreign cloud providers
* **Knowledge Base Development**: Will create a foundation of quantum error correction expertise that can be leveraged across national quantum initiatives
* **Technology Transfer**: Will facilitate knowledge transfer between academic research and industrial applications in quantum computing

### **17.2 Strategic Sector Impact**

* **Defense and National Security**: Will provide secure quantum computing capabilities for cryptography research and secure communications
* **Healthcare and Pharmaceutical**: Will enable privacy-preserving quantum simulation for drug discovery and personalized medicine
* **Financial Services**: Will support quantum optimization for risk assessment and portfolio management with data confidentiality
* **Energy and Climate**: Will facilitate quantum simulation for materials science and energy storage solutions

### **17.3 Economic Development**

* **Startup Ecosystem**: Will catalyze quantum software startup development and venture capital investment
* **Job Creation**: Will generate high-skilled employment in quantum software engineering, machine learning, and quantum physics
* **Export Potential**: Will position the country as an exporter of quantum software solutions to global markets
* **Industry Transformation**: Will accelerate quantum readiness across multiple industries through accessible QEC tools

### **17.4 Human Capital Development**

* **Workforce Training**: Will create specialized training programs in quantum error correction and quantum machine learning
* **Academic Partnerships**: Will establish research collaborations with universities to develop quantum talent pipeline
* **Knowledge Retention**: Will help prevent brain drain by creating domestic opportunities in cutting-edge quantum research
* **Educational Resources**: Will develop educational materials on quantum error correction for broader STEM education

### **17.5 International Collaboration**

* **Standards Development**: Will contribute to international quantum computing standards while maintaining technological sovereignty
* **Research Networks**: Will establish collaborative networks with global quantum research institutions
* **Responsible Innovation**: Will promote ethical frameworks for quantum computing development internationally
* **Technology Diplomacy**: Will strengthen international relations through quantum technology partnerships

## **18. Conclusion & Impact Statement**

QCraft will represent a paradigm shift in quantum computing workflows, advancing fault-tolerant, privacy-preserving, and device-agnostic operations. While quantum computing giants like IBM, IonQ, and Google continue to push hardware capabilities, QCraft will address the critical gaps in **execution privacy, adaptive QEC, and hardware-agnostic circuit optimization** that remain underserved in the current ecosystem.

Our strategy to build an **offline-first compiler OS layer**, powered by **Reinforcement Learning and Graph Neural Networks**, will enable organizations to maintain complete control over their quantum circuits and execution pathways. Through our **three core innovations**—**(1) Adaptive QEC Code Discovery** (RL-based dynamic code selection achieving 18-28% LER reduction), **(2) Device-Aware Circuit Optimization** (GNN+RL mapping achieving 15-22% CNOT reduction), and **(3) Privacy-Preserving Offline Execution** (enabling secure quantum computing for defense, pharma, and finance)—QCraft will adapt to evolving noise profiles of quantum devices, learn from past executions, and optimize in real-time, capabilities that static QEC stacks cannot achieve. Our 18-month development plan targets production-ready software deliverables (IDE, Compiler, Decoder, Profiler) with 3-5 patent filings protecting our core IP, while our long-term vision (Years 2-5) includes extensions such as Federated Quantum Learning, QEC Marketplace, and Quantum Memory solutions.

Beyond circuit execution, QCraft’s QEC library and adaptive stack will be designed to extend into **Quantum Memory stabilization**, positioning us for spin-off product lines and strategic partnerships with quantum memory manufacturers. This dual-market positioning will enhance QCraft’s sustainability and market resilience.

The successful realization of QCraft will:

* Establish a **deep-tech startup contributing to India’s quantum computing ecosystem** and aligning with national quantum initiatives.
* Enable **privacy-focused sectors** (defense, pharma, finance) to confidently adopt quantum workflows without compromising sensitive data.
* Develop a scalable, adaptive software platform that complements, rather than competes with, quantum hardware providers, creating a sustainable ecosystem.
* Catalyze the next wave of **Quantum Error Correction innovations** through data-driven learning pipelines and collaborative research.
* Bridge the gap between theoretical QEC advances and practical, industry-ready implementations, accelerating the timeline to fault-tolerant quantum computing.

With a strategic funding of **INR 2 Crore (~USD $240,000)** over 18 months, we propose to transform QCraft from a promising concept into a globally competitive product, helping to secure India’s position in the future of secure and fault-tolerant quantum computing.

## **Appendix: Glossary of Key Terms**

* **Adaptive QEC**: Quantum error correction techniques that dynamically adjust to changing noise profiles and hardware characteristics, as opposed to static approaches.
* **Adaptive Quantum Resource Estimation**: QCraft’s technique for dynamically predicting and allocating quantum computing resources based on circuit characteristics and execution history.
* **Circuit Obfuscation**: Techniques to protect the logical structure and intellectual property of quantum circuits by transforming them in ways that preserve functionality while hiding implementation details.
* **Compiler OS Layer:** QCraft’s core concept of a quantum compiler operating system that mediates between logical quantum algorithms and physical quantum hardware, providing device-aware execution capabilities that optimize for specific hardware characteristics.
* **DL (Deep Learning)**: A subset of machine learning involving neural networks with multiple layers that can learn representations of data with multiple levels of abstraction.
* **Device-Aware Execution:** The capability to run quantum circuits across different quantum hardware platforms while specifically optimizing for each platform’s unique characteristics and constraints.
* **Federated Quantum Learning (FQL)**: A privacy-preserving approach where quantum models are trained across multiple decentralized devices or servers holding local data samples, without exchanging the raw data.
* **Fidelity**: A measure of how accurately a quantum operation or circuit performs compared to its ideal version.
* **GNN (Graph Neural Network)**: A neural network architecture designed to process data structured as graphs, such as qubit connectivity or syndrome graphs in quantum circuits.
* **LDPC (Low-Density Parity-Check) Codes**: A class of error-correcting codes with sparse parity-check matrices, used in both classical and quantum error correction.
* **Logical Error Rate (LER)**: The rate at which errors occur on logical qubits after error correction has been applied, a key metric for evaluating QEC performance.
* **Magic State Distillation**: A process in quantum computing for producing high-fidelity quantum states needed for certain fault-tolerant operations.
* **NISQ (Noisy Intermediate-Scale Quantum)**: Current generation of quantum computers with limited qubit counts (50-1000) and significant noise, requiring error mitigation rather than full error correction.
* **Noise Model**: A mathematical description of the types and rates of errors affecting a quantum device.
* **Offline-First Execution**: QCraft’s approach, where critical transformations and optimizations are performed locally on the user’s hardware, enhancing privacy and reducing dependence on cloud services.
* **Privacy-Preserving Quantum Computing**: Techniques and protocols that enable quantum computations while protecting sensitive information about the algorithms, data, or results.
* **QEC (Quantum Error Correction)**: Methods for detecting and correcting errors in quantum computations, enabling the reliable operation of quantum computers.
* **QECaaS (Quantum Error Correction as a Service)**: QCraft’s modular approach offering quantum error correction capabilities through standardized APIs that can be integrated with existing quantum computing workflows.
* **QRAM (Quantum Random Access Memory)**: Memory systems designed to store and retrieve quantum states, essential for many quantum algorithms.
* **QPU (Quantum Processing Unit)**: The quantum equivalent of a classical CPU, referring to the physical quantum processor that performs quantum computations.
* **Quantum Circuit Fingerprinting**: A technique for creating unique identifiers for quantum circuits that can be used for verification without revealing the circuit structure.
* **Quantum-Classical Co-optimization**: The coordinated optimization of both quantum and classical components of a hybrid computing system to maximize overall performance.
* **Quantum Error Correction Marketplace**: A collaborative platform for sharing, benchmarking, and trading quantum error correction components, enabling researchers and organizations to leverage collective expertise in QEC development.
* **Quantum Memory Stabilization**: Techniques for maintaining quantum states over extended periods, essential for quantum memory and long-duration quantum computations.
* **Quantum Resource Negotiation Protocol**: A standardized protocol for dynamic allocation and optimization of quantum computing resources across multiple hardware providers, ensuring efficient utilization and cost-effectiveness.
* **RL (Reinforcement Learning)**: A type of machine learning where agents learn to make decisions by receiving rewards or penalties from their environment.
* **Surface Code**: A leading QEC code that arranges qubits in a 2D lattice, offering high error tolerance and scalability.
* **Syndrome (in QEC)**: The outcome of a set of measurements that indicates the presence and type of errors in a quantum code.
* **Syndrome Decoder**: An algorithm or model that interprets syndrome data to identify and correct errors in quantum codes.
* **Transfer Learning**: A machine learning technique where knowledge gained in solving one problem is applied to a different but related problem, used in QCraft for cross-device adaptation.
* **Transpiler**: Software that converts a quantum circuit from one form or gate set to another, often optimizing for specific hardware constraints.

## **Appendix: Biosketches & Team Capability**

[Detailed biosketches for PI, co-PI, and key personnel will be included here, featuring education, appointments, relevant publications, prior support, synergistic activities, and specific roles on the QCraft project.]

Letters of commitment from collaborators and advisors will be provided as a separate attachment (see 13.2 External Collaborations and Letters of Support).

## **Appendix: References**

All in-text citations [n] refer to the numbered references below.

[1] P. W. Shor, “Scheme for reducing decoherence in quantum computer memory,” Phys. Rev. A, vol. 52, no. 4, pp. R2493-R2496, 1995.

[2] A. M. Steane, “Error Correcting Codes in Quantum Theory,” Phys. Rev. Lett., vol. 77, no. 5, pp. 793-797, 1996.

[3] D. Gottesman, “Stabilizer Codes and Quantum Error Correction,” PhD thesis, California Institute of Technology, 1997.

[4] B. M. Terhal, “Quantum error correction for quantum memories,” Rev. Mod. Phys., vol. 87, pp. 307-346, 2015.

[5] J. Preskill, “Quantum Computing in the NISQ era and beyond,” Quantum, vol. 2, p. 79, 2018.

[6] T. Fösel, P. Tighineanu, T. Weiss, and F. Marquardt, “Reinforcement Learning with Neural Networks for Quantum Feedback,” Phys. Rev. X, vol. 8, no. 3, p. 031084, 2018.

[7] H. P. Nautrup, N. Delfosse, V. Dunjko, H. J. Briegel, and N. Friis, “Optimizing Quantum Error Correction Codes with Reinforcement Learning,” Quantum, vol. 3, p. 215, 2019.

[8] V. V. Sivak, J. P. Gross, S. Shankar, et al., “Model-free quantum control with reinforcement learning,” Nature, vol. 616, pp. 50-55, 2023.

[9] S. Varsamopoulos, B. Criger, and K. Bertels, “Decoding small surface codes with feedforward neural networks,” Quantum Sci. Technol., vol. 3, 015004, 2018.

[10] P. Baireuther, T. E. O’Brien, B. Tarasinski, and C. W. J. Beenakker, “Machine-learning-assisted correction of correlated qubit errors in a topological code,” Quantum, vol. 2, p. 48, 2018.

[11] S. Krastanov and L. Jiang, “Deep Neural Network Probabilistic Decoder for Stabilizer Codes,” Sci. Rep., vol. 7, p. 11003, 2017.

[12] C. Chamberland, T. J. Yoder, J. B. Hertzberg, and A. W. Cross, “Physical realization of quantum error correction with GNN decoders,” PRX Quantum, vol. 2, p. 040342, 2021.

[13] F. L. R. Pereira, M. Serbyn, and A. Seif, “Machine learning message-passing for the scalable decoding of QLDPC codes,” npj Quantum Information, vol. 9, p. 42, 2023.

[14] T. B. Brown, S. Darabi, and A. A. Awan, “Graph Neural Networks for Enhanced Decoding of Quantum LDPC Codes,” arXiv:2310.17758, 2023.

[15] Y. Chen, Z. Wang, and L. Li, “Decoding Quantum LDPC Codes Using Graph Neural Networks,” arXiv:2408.05170, 2024.

[16] J. B. Hertzberg, E. J. Zhang, S. Rosenblatt, et al., “Learning high-accuracy error decoding for quantum processors,” Nature, vol. 625, pp. 481-486, 2024.

[17] K. Chinni, P. Patel, B. Kulchytskyy, et al., “Simultaneous discovery of quantum error correction codes and encoders with reinforcement learning,” npj Quantum Information, vol. 10, p. 27, 2024.

[18] Y. Chen, M. Farahzad, S. Choi, and T. Yoder, “Reinforcement learning for autonomous quantum error correction,” Quantum, vol. 6, p. 795, 2022.

[19] J. Liu, F. Tacchino, J. R. Glick, L. Jiang, and A. Mezzacapo, “Representation learning for quantum program analysis and optimization,” arXiv:2204.04886, 2022.

[20] A. Broadbent, J. Fitzsimons, and E. Kashefi, “Universal blind quantum computation,” 50th Annual IEEE Symposium on Foundations of Computer Science, pp. 517-526, 2009.

[21] K. Chinni, P. Patel, B. Kulchytskyy, et al., “Simultaneous discovery of quantum error correction codes and encoders with reinforcement learning,” npj Quantum Information, vol. 10, p. 27, 2024.

[22] S. Herbert, “Resource estimation for quantum algorithms using Pauli gates,” Physical Review A, vol. 104, p. 032404, 2021.

New references:

C.G. Yale, R. Rines, V. Omole, B. Thotakura, A. D. Burch, M.N.H. Chow, M. Ivory, D. Lobser, B.K. McFarland, M.C. Revelle, S.M. Clark, and P. Gokhale, “Noise-aware circuit compilations for a continuously parameterized two-qubit gateset”, Phys. Rev. Applied **24**, 024057, 2025**.**

J. Balewski, W.-H. Lin, A. Mitra, M. Kornjača, S. Ostermann, P.L.S. Lopes, D.B. Tan, and J. Cong, “Compilation of QCrank Encoding Algorithm for a Dynamically Programmable Qubit Array Processor”, arXiv:2507.10699, 2025.

Review on QEC

J. Roffe, “Quantum Error Correction: An Introductory Guide”, *Contemporary Physics*, *60*(3), 226, 2019.

RL-based QEC

D. Kremer, V. Villar, H. Paik, I. Duran, I. Faro and J. Cruz-Benito, Practical and efficient quantum circuit synthesis and transpiling with Reinforcement Learning, arXiv: 2405.13196.

Code switching

Sascha Heußen and Hilder, Janine, Efficient fault-tolerant code switching via one-way transversal {CNOT} gates, Quantum, 9, 1846, (2025).

## **Appendix A: Methodology and Experimental Details**

This appendix provides comprehensive technical details for the preliminary results reported in Section 9, ensuring full reproducibility and transparency.

### **A.1 RL-Based Code Patch Discoverer**

**Algorithm**: Proximal Policy Optimization (PPO) with clipped surrogate objective

**Network Architecture**: - Policy network: 3-layer MLP (512-256-128 hidden units, ReLU activation) - Value network: 3-layer MLP (512-256-1, ReLU activation) - Input: Graph-structured state (syndrome graph + device topology + noise profile features, 256-dim embedding via GNN encoder) - Output: Categorical distribution over code choices (surface d=3,5,7 variants + qLDPC codes)

**Training Hyperparameters**: - Learning rate: 3×10⁻⁴ (Adam optimizer) - Discount factor γ: 0.99 - GAE parameter λ: 0.95 - Clipping parameter ε: 0.2 - Batch size: 2048 transitions - Training episodes: 5,000 per noise regime - GPU configuration: 4× NVIDIA A100 (80GB), 800 GPU-hours total

**Noise Models** (Qiskit Aer): - Depolarizing: p ∈ {0.001, 0.003, 0.005, 0.01} - Amplitude damping: γ ∈ {0.0005, 0.001, 0.003, 0.005} - Crosstalk: ε ∈ {0.001, 0.002, 0.003} (nearest-neighbor coupling) - Non-stationary drift: Ornstein-Uhlenbeck process with θ=0.15, σ=0.0005

**Evaluation Protocol**: - 100 independent test episodes per noise regime - Logical error rate computed via Monte Carlo simulation (10⁴ shots per circuit) - Statistical tests: Wilcoxon signed-rank (paired, two-tailed, α=0.001) - Confidence intervals: Bootstrap percentile method (B=2,000)

### **A.2 RL-Enhanced Circuit Optimizer**

**Algorithm**: Deep Q-Network (DQN) with experience replay and target network

**Network Architecture**: - Q-network: Graph Convolutional Network (GCN) with 4 layers (128-256-256-128 hidden units) - Input: Circuit DAG (gates as nodes, dependencies as edges, 64-dim node features) - Output: Q-values for 50 discrete actions (gate commutation, cancellation, routing moves)

**Training Hyperparameters**: - Learning rate: 1×10⁻⁴ (Adam optimizer) - Discount factor γ: 0.95 - Exploration: ε-greedy (ε: 1.0 → 0.01 over 100k steps) - Replay buffer: 50,000 transitions - Target network update: every 1,000 steps - Batch size: 128 - Training duration: 600 GPU-hours on 4× NVIDIA A100

**Benchmark Circuits**: - QAOA: MaxCut instances (Erdős-Rényi graphs, n=10-20 nodes, p=0.5) - VQE: H₂ (STO-3G), LiH (STO-3G) with UCCSD ansatz - Random Clifford+T: width w ∈ {10,20,50}, depth d ∈ {20,50,100}

**Baselines**: - Qiskit transpiler (optimization\_level=3, routing\_method=‘sabre’) - Cirq optimizers (merge\_single\_qubit\_gates, eject\_z, eject\_phased\_paulis)

**Metrics**: - CNOT count, circuit depth, estimated fidelity (via noise-aware simulation) - Inference latency measured on NVIDIA A100 GPU and Intel Xeon Platinum 8380 CPU

### **A.3 GNN-Based Syndrome Decoder**

**Algorithm**: Graph Attention Network (GAT) with multi-head attention

**Network Architecture**: - 4 GAT layers with 8 attention heads per layer - Hidden dimensions: 128 per head (1024 total per layer) - Input: Syndrome graph (stabilizer measurements as nodes, qubit correlations as edges) - Output: Binary classification per edge (error/no-error)

**Training Hyperparameters**: - Learning rate: 5×10⁻⁴ (AdamW optimizer, weight decay 1×10⁻⁵) - Batch size: 64 syndrome graphs - Training samples: 500,000 per code/noise combination - Data augmentation: Random syndrome flips (10%), graph rotations - Training duration: 1,000 GPU-hours on 4× NVIDIA A100

**Code Families**: - Surface codes: Rotated planar layout, distances d ∈ {3,5,7,9} - qLDPC: Hypergraph product [[90,8,10]], [[144,12,12]]

**Noise Models**: - Depolarizing: p ∈ {0.001, 0.003, 0.005} - Amplitude damping: γ ∈ {0.0005, 0.001, 0.003} - Measurement errors: pm ∈ {0.005, 0.01, 0.02} - Combined: Depolarizing + amplitude damping + measurement errors

**Baselines**: - PyMatching 2.1.0 (MWPM decoder) - Union-Find decoder (Qiskit implementation) - Belief Propagation (BP) decoder (max 50 iterations)

**Evaluation Metrics**: - Frame error rate (FER): Fraction of failed correction attempts - Logical error rate (LER): Fraction of uncorrected logical errors - Decoder latency: Wall-clock time per syndrome (averaged over 1,000 samples)

### **A.4 Statistical Analysis Methods**

**Hypothesis Testing**: - Wilcoxon signed-rank test (non-parametric, paired samples) - Significance level: α = 0.001 (Bonferroni correction for multiple comparisons) - Sample sizes: n ≥ 50 independent runs per condition

**Confidence Intervals**: - Bootstrap percentile method with B = 2,000 resamples - 95% confidence intervals reported for all metrics

**Effect Size Measures**: - Cohen’s d for continuous metrics (CNOT count, depth, latency) - Relative risk reduction for binary outcomes (FER, LER)

**Reproducibility**: - Random seeds: Fixed for training (seed=42), varied for evaluation (seeds 1-100) - Software versions: Python 3.10, PyTorch 2.0.1, Qiskit 0.45.0, Cirq 1.3.0 - Hardware: NVIDIA A100 (80GB), CUDA 11.8, cuDNN 8.7 - All configurations, logs, and trained models archived and available upon request

### **A.5 Computational Resources**

**Total GPU-Hours**: 2,400 GPU-hours (NVIDIA A100 80GB) - Code Patch Discoverer: 800 GPU-hours - Circuit Optimizer: 600 GPU-hours - Syndrome Decoder: 1,000 GPU-hours

**Estimated Cost**: ~$6,000 (cloud GPU pricing: $2.50/hour for A100)

**Carbon Footprint**: ~480 kg CO₂eq (estimated using ML CO₂ Impact calculator, assuming average US grid mix)

## **References**

[1] P. W. Shor, “Scheme for reducing decoherence in quantum computer memory,” Physical Review A, vol. 52, p. R2493, 1995.

[2] A. M. Steane, “Error correcting codes in quantum theory,” Physical Review Letters, vol. 77, p. 793, 1996.

[3] D. Gottesman, “Stabilizer codes and quantum error correction,” PhD thesis, California Institute of Technology, 1997.

[4] B. M. Terhal, “Quantum error correction for quantum memories,” Reviews of Modern Physics, vol. 87, p. 307, 2015.

[5] J. Preskill, “Quantum computing in the NISQ era and beyond,” Quantum, vol. 2, p. 79, 2018.

[6] T. Fösel, P. Tighineanu, T. Weiss, and F. Marquardt, “Reinforcement learning with neural networks for quantum feedback,” Physical Review X, vol. 8, p. 031084, 2018.

[7] H. P. Nautrup, N. Delfosse, V. Dunjko, H. J. Briegel, and N. Friis, “Optimizing quantum error correction codes with reinforcement learning,” Quantum, vol. 3, p. 215, 2019.

[8] V. V. Sivak, A. Eickbusch, B. Royer, et al., “Real-time quantum error correction beyond break-even,” Nature, vol. 616, pp. 50-55, 2023.

[9] S. Varsamopoulos, B. Criger, and K. Bertels, “Decoding small surface codes with feedforward neural networks,” Quantum Science and Technology, vol. 3, p. 015004, 2018.

[10] P. Baireuther, T. E. O’Brien, B. Tarasinski, and C. W. J. Beenakker, “Machine-learning-assisted correction of correlated qubit errors in a topological code,” Quantum, vol. 2, p. 48, 2018.

[11] S. Krastanov and L. Jiang, “Deep neural network probabilistic decoder for stabilizer codes,” Scientific Reports, vol. 7, p. 11003, 2017.

[12] C. Chamberland, G. Zhu, T. J. Yoder, J. B. Hertzberg, and A. W. Cross, “Topological and subsystem codes on low-degree graphs with flag qubits,” Physical Review X, vol. 10, p. 011022, 2020.

[13] L. Berent, T. Hillmann, J. Eisert, R. Sweke, and J. Roffe, “Decoding quantum color codes with MaxSAT,” npj Quantum Information, vol. 10, p. 13, 2024.

[14] H. Bayraktar, A. Charara, D. Clark, et al., “cuQuantum SDK: A High-Performance Library for Accelerating Quantum Science,” arXiv:2308.01999, 2023.

[15] K. Chinni, P. Patel, B. Kulchytskyy, et al., “Simultaneous discovery of quantum error correction codes and encoders with reinforcement learning,” npj Quantum Information, vol. 10, p. 27, 2024.

[16] J. B. Hertzberg, E. J. Zhang, S. Rosenblatt, et al., “Laser-annealing Josephson junctions for yielding scaled-up superconducting quantum processors,” npj Quantum Information, vol. 7, p. 129, 2021.

[17] J. Roffe, “Quantum error correction: An introductory guide,” Contemporary Physics, vol. 60, no. 3, pp. 226-245, 2019.

[18] R. S. Sutton and A. G. Barto, “Reinforcement Learning: An Introduction,” 2nd ed., MIT Press, 2018.

[19] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, pp. 436-444, 2015.

[20] I. Goodfellow, Y. Bengio, and A. Courville, “Deep Learning,” MIT Press, 2016.

[21] A. Kandala, K. Temme, A. D. Córcoles, et al., “Error mitigation extends the computational reach of a noisy quantum processor,” Nature, vol. 567, pp. 491-495, 2019.

[22] Google Quantum AI, “Suppressing quantum errors by scaling a surface code logical qubit,” Nature, vol. 614, pp. 676-681, 2023.

[23] M. Fingerhuth, T. Babej, and P. Wittek, “Open source software in quantum computing,” PLOS ONE, vol. 13, no. 12, p. e0208561, 2018.

[24] X. Chen, I. Lizuain, A. Ruschhaupt, D. Guéry-Odelin, and J. G. Muga, “Shortcut to adiabatic passage in two- and three-level atoms,” Physical Review Letters, vol. 105, p. 123003, 2010.

[25] Z. Liu, S. P. Rodrigues, and W. Cai, “Simulating the Ising model with a deep convolutional generative adversarial network,” Physical Review E, vol. 97, p. 053304, 2018.

[26] A. Broadbent, J. Fitzsimons, and E. Kashefi, “Universal blind quantum computation,” 50th Annual IEEE Symposium on Foundations of Computer Science, pp. 517-526, 2009.

[27] S. Herbert, “Resource estimation for quantum algorithms using Pauli gates,” Physical Review A, vol. 104, p. 032404, 2021.

[28] C. G. Yale, R. Rines, V. Omole, et al., “Noise-aware circuit compilations for a continuously parameterized two-qubit gateset,” Physical Review Applied, vol. 24, p. 024057, 2025.

[29] J. Balewski, W.-H. Lin, A. Mitra, et al., “Compilation of QCrank encoding algorithm for a dynamically programmable qubit array processor,” arXiv:2507.10699, 2025.

[30] D. Kremer, V. Villar, H. Paik, I. Duran, I. Faro, and J. Cruz-Benito, “Practical and efficient quantum circuit synthesis and transpiling with reinforcement learning,” arXiv:2405.13196, 2024.

[31] S. Heußen and J. Hilder, “Efficient fault-tolerant code switching via one-way transversal CNOT gates,” Quantum, vol. 9, p. 1846, 2025.

[32] Google Quantum AI, “Quantum error correction below the surface code threshold,” Nature, vol. 625, pp. 45-50, December 2024.

[33] E. T. Campbell, B. M. Terhal, and C. Vuillot, “Roads towards fault-tolerant universal quantum computation,” Nature, vol. 549, pp. 172-179, 2017.

[34] M. J. Biercuk, A. C. Doherty, and H. Uys, “Dynamical decoupling sequence construction as a filter-design problem,” Journal of Physics B: Atomic, Molecular and Optical Physics, vol. 44, p. 154002, 2011.

[35] P. Panteleev and G. Kalachev, “Quantum LDPC codes with almost linear minimum distance,” IEEE Transactions on Information Theory, vol. 68, no. 1, pp. 213-229, 2022.

[36] C. Gidney, “Stim: A fast stabilizer circuit simulator,” Quantum, vol. 5, p. 497, 2021.

[37] O. Higgott and C. Gidney, “Sparse Blossom: Correcting a million errors per second with minimum-weight matching,” arXiv:2303.15933, 2023.