Fixed-Topology UCR State Encoding for High-Dimensional Quantum Reinforcement Learning

Junchen Han(韩钧琛)[1], Fengtao Xiang(项凤涛)[1,*] Hao Shi(石皓)[1,†] and Ming Zhang(张明)[1]

[1]College of Intelligence Science and Technology, National University of Defense Technology, Changsha, 410073, China

August 28, 2026

Abstract

In the era of noisy intermediate-scale quantum (NISQ) computing, the stable integration of high-dimensional classical states into quantum neural networks is a critical challenge for quantum reinforcement learning. Existing encoding schemes either rely heavily on classical neural networks, which can overshadow the training contributions of the quantum component, or use standard quantum input methods such as amplitude encoding, which suffer from limited trainability, noise sensitivity, and high implementation costs. To address this challenge, this paper repurposes uniformly controlled rotations (UCR) from static state preparation and static data encoding into a state-encoding interface for quantum neural networks, elevating the process of integrating classical states into quantum neural networks to an independent methodological layer. Across Gymnasium continuous-control tasks, our UCR encoding achieves the best peak return, final return, and evaluation-curve AUC compared to representative angle encoding and amplitude encoding, with the AUC improving by +5.9 to +53.5 return units over angle encoding. While our UCR encoding is not the method with the fewest gates, it eliminates the classical neural networks on which angle encoding relies during dimensionality reduction, provides a fixed, sample-independent topology, and supports explicit resource accounting. On BipedalWalker-v3, UCR encoding reduces the actor parameter count of angle encoding from 1003 to 229 and increases the final return from 220.4 to 264.9, which is higher than the final return of amplitude encoding. A systematic evaluation of gate-level noise shows that the absolute performance advantage of UCR encoding in complex continuous control tasks persists in the low-noise regime, but this advantage window narrows as the input dimension and quantum circuit depth increase. For high-dimensional classical state inputs, UCR encoding achieves a more stable performance–resource trade-off than typical quantum encoding methods. Motivated by the problem of inputting classical states into quantum neural networks, our approach may be further explored in broader machine learning scenarios and higher-dimensional applications.

Keywords: quantum reinforcement learning, uniformly controlled rotations, quantum state encoding, high-dimensional classical states

PACS: 03.67.Ac, 03.67.Lx, 07.05.Mh

1. Introduction

In the era of noisy intermediate-scale quantum (NISQ) computing, constrained by the number of qubits and the depth of quantum circuits, quantum reinforcement learning is gradually expanding from proof-of-concept applications to more

*Corresponding author. E-mail: xiangfengtao@nudt.edu.cn

†Corresponding author. E-mail: shihao_hill@163.com

complex simulated control and decision-making problems.[1–5] Recent work further shows that parameterized quantum circuits (PQCs) can be integrated into policy-gradient frameworks such as PPO and deployed on cloud-based quantum processors, highlighting the importance of hardware-aware quantum circuits design.[6,7] As task complexity and state dimensions increase, a fundamental question begins to emerge: **How exactly should high-dimensional classical states be integrated into quantum neural networks?**

Regarding high-dimensional classical inputs, existing quantum machine learning methods can be broadly categorized into two types. The first category follows a "classical compression first, quantum learning later" approach, wherein state compression is first achieved through modules such as classical linear layers, pre-encoding networks, or data re-uploading, followed by the application of lighter-weight quantum embedding techniques like angle encoding.[8–10] While these methods are relatively easy to implement, they rely heavily on classical neural networks, and the actual contribution of the quantum component is difficult to quantify (as shown in Fig. 1). The second category consists of quantum input methods represented by amplitude encoding, which offers logarithmic compression capabilities.[9,11] Research indicates that amplitude encoding may lead to issues such as encoded-state concentration, reduced class separability, and optimization barriers; in classification tasks, the concentration of encoded state means may even result in a loss barrier, creating a lower bound for the loss function that is difficult to overcome via optimization algorithms.[12] Furthermore, noise and experimental studies indicate that amplitude-based inputs using Möttönen state preparation degrade rapidly with increasing circuit depth and under various noise models.[13,14] Even when using faster approximate embedding methods, additional offline clustering, symbol optimization, or specific device assumptions are typically required.[15]

From a broader perspective of quantum scientific computing, research on low-depth state preparation, fast approximate block encoding, real-valued encoded state computation, principal component analysis, QSP/QSVT, and related matrix algorithms has repeatedly shown that discussing only abstract complexity while ignoring actual circuit cost, the number of two-qubit gates, and noise vulnerability can easily lead to an overestimation of a method's potential; its time and space costs, as well as two-qubit gate errors, often truly dominate system performance.[16–23] Related NISQ work has used hybrid decomposition to keep inference within limited quantum resources and communication-aware circuit partitioning to reduce inter-QPU transmission costs.[24,25] Furthermore, work on quantum dimensionality reduction, qudit learning, and financial risk analysis also demonstrates from various perspectives that: When the dimensionality of classical data increases or data processing tasks become more complex, the cost of interfacing classical data with quantum circuits becomes a critical bottleneck; therefore, the classical-to-quantum data interface should not be simply overlooked.[9,10,26] These studies collectively reinforce the conclusion that the training performance of quantum neural networks depends largely on the input interface method. However, the primary focus of existing quantum reinforcement learning literature remains on whether quantum neural networks themselves can serve as function approximators, rather than elevating "how high-dimensional states are fed into quantum networks" to a distinct research topic. In fact, the validity of many results relies heavily on classical preprocessing, low-dimensional representations, or task-specific compression strategies applied to the quantum circuit front-end.[2,6,8,17,27] Although the state input interface is crucial in practice, it is rarely systematically treated as an independent methodological layer in the existing literature.

Therefore, our starting point is that the quantum input of high-dimensional classical states cannot be judged solely by compression ratio; trainability, stability, and quantum state preparation costs must also be considered simultaneously. We treat **how high-dimensional classical states enter quantum policy networks** as a methodological problem worthy of separate study.

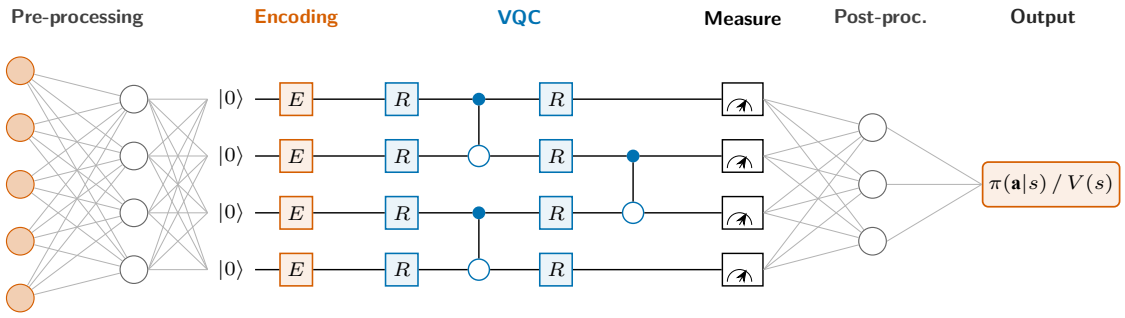


Fig. 1. General structure of a hybrid quantum–classical neural network. The front-end classical network reduces the dimension of the observation to the number of qubits; $|0\rangle^{\otimes Q}$ undergoes evolution via the encoding operator U_E and the variational quantum circuit $U_{\mathrm{VQC}}(\boldsymbol{\theta})$, and a Pauli-$Z$ measurement yields the expectation-value vector, which is then mapped by the back-end classical network to a policy or value output. The variational quantum circuit is expanded into a two-layer structure consisting of rotation gates R and controlled entanglement gates; measurement symbols follow standard quantum circuit notation.

Table 1. Overview of UCR-related methods and the proposed QRL state interface

Method	**Main Objective**	**Role of UCR or controlled rotations**	**Features**
Basic UCR method	General multi-qubit gate synthesis and quantum state transformations	Circuit-synthesis primitive	State interfaces not studied in machine learning
FRQI	Quantum image representation	Encoding pixel values by applying address-controlled rotations to color qubits	Focuses on the representation of image information carried by quantum circuits
QPIXL	Efficient image state synthesis and compression	Uses linear-scale UCR circuits and compression	Focuses on quantum circuit-based image information encoding and compression
QCrank	Vectorized classical data loading	Loads addressed real-valued data entries via reused controlled rotations	Parallel loading and depth optimization for NISQ hardware
Traditional amplitude encoding	Preparing normalized vector states	UCR can serve as an internal primitive in the recursive state preparation process	A mapping from data to quantum states, where normalized features directly become computational-basis amplitudes
Block encoding (e.g., FABLE)	Embedding the scaled matrix A/α into a subblock of a larger unitary operator	UCR can be used to implement matrix-query operators, but additional operations and ancilla-space projection are still required to form a complete block encoding	A data-to-operator mapping
This work	High-dimensional state inputs in QRL	A fixed-template UCR front end in the PPO policy	Focuses on policy learning, differentiability, resource overhead, and statistical performance

UCR first appeared in the context of multi-qubit gate synthesis and quantum state transformation problems; its structure is closely related to Gray-code ordering and Walsh-Hadamard-type linear relations.[14,28] Previous research has applied UCR or related address-controlled rotations to quantum state synthesis, image representation, circuit compression, and vectorized data loading.[29,30] Notably, Balewski et al. further advanced the parallel implementation of UCR-type data loading on real NISQ hardware and achieved good experimental results through hardware-aware parallel scheduling.[31] The main objectives of these works and their relationship to this paper are summarized in Table 1. This paper does not claim to propose a new UCR decomposition, nor does it claim to propose a new general-purpose state preparation algorithm.

For this work, the most significant insight from the UCR approach lies not in the fact that "it is an existing encoding

method," but rather that it provides a mature syntax for structured data input. Unlike existing work on UCR, FRQI, QPIXL, and QCrank—which primarily focuses on data writing, image representation, vectorized encoding, or quantum state preparation, addressing the question of "how to write known data into quantum states"— This work focuses on reconstructing UCR as a differentiable, fixed-topology, fairly comparable, and resource-analyzable state encoding module under the unified PPO protocol, and explicitly elevates the quantum input for high-dimensional classical states to an independent methodological layer within quantum reinforcement learning.

The contributions of this work are primarily reflected in three aspects. First, we reconstruct UCR from a static state preparation and static data encoding into a state encoding module within a hybrid quantum-classical PPO framework. Second, without omitting any classical input data, we established the differentiability and fixed-topology properties of UCR, and, through a dual-path UCR implementation, balanced simulation-based training efficiency, enabling efficient simulation-based training, with gate-level circuits for controlled noise evaluation and hardware deployment resource analysis. Third, we construct a unified comparison protocol for classical-state input interfaces to quantum circuits, focusing on performance, sample efficiency, training stability, resource complexity, and noise behavior. It is important to note that in quantum reinforcement learning, this state interface is not a mere preprocessing detail but should be regarded as a methodological layer of equal importance to variational quantum circuits.

The remainder of this paper is organized to develop and evaluate the proposed input-interface method. Section 2 reviews the Markov decision process in classical reinforcement learning and the essence of existing UCR implementations. Section 3 introduces the proposed UCR encoding and its dual-path implementation, along with a complexity comparison of encoding schemes. Section 4 evaluates the method on Gymnasium control tasks and input-dimensional scaling experiments. Section 5 concludes the paper.

2. Preliminaries

2.1. Markov Decision Process

We consider the standard Markov decision process

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma), \tag{1}$$

where the state $\boldsymbol{x}_t \in \mathcal{S} \subseteq \mathbb{R}^d$ is a high-dimensional classical observation, the action $\boldsymbol{a}_t \in \mathcal{A}$ is a discrete or continuous variable, P denotes the state transition kernel, r denotes the immediate reward, and $\gamma \in (0, 1)$ is the discount factor. This work adopts the standard actor-critic framework and objective settings of Proximal Policy Optimization (PPO).

2.2. UCR Decomposition of Controlled Rotations

The uniformly controlled rotations (UCR) decomposition represents the UCR encoding acting on $k + 1$ qubits (k control qubits + 1 target qubit) as an equivalent circuit alternating between single-qubit rotation gates and CNOT gates. This decomposition relies on two classical constructions: Gray codes and the Walsh–Hadamard transform. This section first presents these two constructs, followed by the UCR decomposition.

Gray Code. Let $N = 2^k$ ($k \geq 1$). The *standard reflected Gray code* is defined as

$$G(i) = i \oplus (i \gg 1), \tag{2}$$

where $\oplus$ denotes the bitwise XOR operation, and $\gg$ denotes the right shift operation. Let $\mathrm{ctz}(\cdot)$ denote the count-of-trailing-zeros function.

Walsh–Hadamard Transform. Let 2×2 Hadamard matrix

$$\boldsymbol{H} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \tag{3}$$

then the $N \times N$ Hadamard matrix $\boldsymbol{H}_N = \boldsymbol{H}^{\otimes k}$ is formed by taking the Kronecker product of $\boldsymbol{H}$ k times. By the properties of the Kronecker product, we know that $\boldsymbol{H}_N$ is symmetric ($\boldsymbol{H}_N = \boldsymbol{H}_N^\top$), and

$$\boldsymbol{H}_N^2 = N\,\boldsymbol{I}_N, \qquad \boldsymbol{H}_N^{-1} = \frac{1}{N}\boldsymbol{H}_N. \tag{4}$$

Define the *Scaled Fast Walsh–Hadamard Transform* (SFWHT) and the *Gray permutation matrix*

$$\mathrm{SFWHT}(\boldsymbol{a}) = \frac{1}{N}\boldsymbol{H}_N\,\boldsymbol{a}, \qquad [\boldsymbol{P}_{\text{gray}}]_{ij} = \delta_{j,G(i)}, \tag{5}$$

where $\delta_{j,G(i)}$ is the Kronecker delta function. $\boldsymbol{P}_{\text{gray}}$ is an orthogonal permutation matrix ($\boldsymbol{P}_{\text{gray}}^\top \boldsymbol{P}_{\text{gray}} = \boldsymbol{I}_N$), whose role is to reorder the natural order into Gray code order. Based on this, we define the *fixed transformation matrix*

$$\boldsymbol{M} = \boldsymbol{P}_{\text{gray}}\,\frac{1}{N}\,\boldsymbol{H}_N \in \mathbb{R}^{N\times N}. \tag{6}$$

$\boldsymbol{M}$ depends only on N and is independent of the input data, statistics, and trainable parameters; its i-th row is equal to the $G(i)$-th row of $(1/N)\boldsymbol{H}_N$.

UCR decomposition. Based on the above construction, given a valid data-angle vector $\boldsymbol{\alpha} \in \mathbb{R}^N$, the UCR decomposition first maps $\boldsymbol{\alpha}$ via the fixed matrix defined in Eq. (6) to a *gate angle* vector

$$\boldsymbol{g} = \boldsymbol{M}\,\boldsymbol{\alpha}, \tag{7}$$

which is then fed into the circuit, followed by N alternating rotation–CNOT gate pairs:

$$\underbrace{H^{\otimes k}}_{\text{superposition}} \quad R(g_0)\,\mathrm{CNOT}(q_{c_0}, q_k)\; R(g_1)\,\mathrm{CNOT}(q_{c_1}, q_k)\;\cdots\; R(g_{N-1})\,\mathrm{CNOT}(q_{c_{N-1}}, q_k), \tag{8}$$

where $R \in \{R_Y, R_Z\}$ is a single-qubit rotation gate, and q_k is the target qubit. The control qubit indices are given by the Gray code difference:

$$c_i = \begin{cases} k - \mathrm{ctz}\big(G(i) \oplus G(i+1)\big) - 1, & 0 \le i < N-1, \\ 0, & i = N-1. \end{cases} \tag{9}$$

This satisfies $c_i \in \{0, 1, \ldots, k-1\}$, meaning that c_i always points to a specific qubit in the control register.

3. Method

3.1. Problem Definition, Notation, and Object of Study

The problem we address is: *How do high-dimensional classical states enter a hybrid quantum-classical PPO policy network?* Building upon the MDP $\mathcal{M}$ defined in Eq. (1), this work does not alter the environment dynamics or the

PPO optimization objective. Instead, while maintaining the actor-critic architecture and the PPO training framework, we investigate how different *state interfaces* affect the representational quality, training stability, and resource consumption of quantum policy networks.

For consistency in subsequent discussion, we abstract the actor's forward computation as

$$\boldsymbol{x} \xrightarrow{\mathcal{N}} \hat{\boldsymbol{x}} \xrightarrow{\mathcal{E}} \boldsymbol{z} \xrightarrow{\mathcal{Q}_\Theta} \boldsymbol{m} \xrightarrow{\mathcal{C}_\Omega} \text{output}, \tag{10}$$

where $\mathcal{N}$ denotes the shared observation normalizer for data preprocessing (see Section 3.3.2 for details), $\mathcal{E}$ represents the subject of this work, denoting various front-end state interfaces, $\mathcal{Q}_\Theta$ denotes the quantum layer and measurement parameterized by Θ, and $\mathcal{C}_\Omega$ denotes the post-quantum classical neural network parameterized by Ω.

We adopt the following notation: lowercase bold italic letters (e.g., $\boldsymbol{x}, \boldsymbol{m}, \boldsymbol{\alpha}$) denote column vectors, uppercase bold italic letters (e.g., $\boldsymbol{M}, \boldsymbol{H}_N$) denote matrices, and the hat symbol $\hat{(\cdot)}$ denotes variables after shared normalization; The quantum state and its $2N$-dimensional coordinate vector in the computational basis are denoted by $|\Psi\rangle$ and $\boldsymbol{v}$, respectively. The sets of trainable parameters for reinforcement learning—$\theta, \phi, \Theta, \Omega$—are denoted by non-bold letters; $|\Psi\rangle$ is used solely for denoting the quantum state. In particular, we define $\theta = (\Theta_a, \Omega_a)$ to denote the complete set of trainable parameters for the actor, and $\phi = (\Theta_c, \Omega_c)$ similarly denotes the set of parameters for the critic.

3.2. Unified Quantum Policy Network Architecture

We address the question of *how high-dimensional classical states are integrated into quantum policy networks* to enable a fair comparison of different front-end approaches. All methods employ the same hybrid quantum-classical PPO actor-critic architecture: shared observation normalization, a unified PPO optimization protocol, uniform quantum layer depth and measurement rules, a unified post-quantum classical neural network form, and consistent handling of discrete and continuous action outputs.

The actor and critic share the same observation normalizer. Given a state $\boldsymbol{x}$, the computational chains for the actor and critic can be written as

$$\boldsymbol{x} \xrightarrow{\mathcal{N}} \hat{\boldsymbol{x}} \xrightarrow{\mathcal{E}_{\text{actor}}} \boldsymbol{z}^{(a)} \xrightarrow{\mathcal{Q}_{\Theta_a}} \boldsymbol{m}^{(a)} \xrightarrow{\mathcal{C}_{\Omega_a}} \pi_\theta(\cdot \mid \boldsymbol{x}), \tag{11}$$

$$\boldsymbol{x} \xrightarrow{\mathcal{N}} \hat{\boldsymbol{x}} \xrightarrow{\mathcal{E}_{\text{critic}}} \boldsymbol{z}^{(c)} \xrightarrow{\mathcal{Q}_{\Theta_c}} \boldsymbol{m}^{(c)} \xrightarrow{\mathcal{C}_{\Omega_c}} V_\phi(\boldsymbol{x}). \tag{12}$$

This setup allows performance differences to be attributed to the input interface itself, corresponding to "how the front-end state encoding is coupled to the quantum circuit."

For the UCR path: $\mathcal{E}$ is instantiated as UCR encoding (directly producing the quantum state $|\Psi\rangle$, i.e., here $\boldsymbol{z} \equiv |\Psi\rangle$); $\mathcal{Q}_\Theta$ is realized by the composition of the variational unitary operator $U_{\text{VQC}}(\Theta)$ and per-qubit Pauli-Z measurements, outputting a readout vector $\boldsymbol{m} \in \mathbb{R}^{n_{\text{qubits}}}$, whose components

$$m_q = \langle\Psi|\, U_{\text{VQC}}^\dagger(\Theta)\, Z_q\, U_{\text{VQC}}(\Theta)\, |\Psi\rangle\,, \quad q = 0, \ldots, n_{\text{qubits}} - 1.$$

Within a unified PPO framework, we compare the UCR encoding of this work with two representative front-end state-encoding interfaces—angle encoding and amplitude encoding—using purely classical PPO as the reference baseline.

3.3. UCR-Based Structured Front-End State Encoding Framework

3.3.1. Qubit Allocation and Notation Conventions

Given an observation dimension $d \geq 2$, we define

$$N = 2^{\lceil \log_2 d \rceil}, \qquad k = \log_2 N, \qquad Q = n_{\text{qubits}} = k + 1. \tag{13}$$

Here, the first k qubits form the *address register* (i.e., the control register in the UCR decomposition in Section 2.2), and the last qubit is the *encoding qubit* (i.e., the target qubit q_k). The number of qubits is uniquely determined by the observation dimension d in UCR encoding.

3.3.2. Shared Normalization and the UCR Effective Data Angle

The encoding pipeline begins with shared online observation normalization. For an input observation $\boldsymbol{x} \in \mathbb{R}^d$, the normalized observation is

$$\hat{x}_j = \text{clip}\left(\frac{x_j - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}},\ -c,\ c \right), \qquad j = 0, \ldots, d-1, \tag{14}$$

where μ_j and σ_j^2 are the mean and variance estimated by the Welford online algorithm, respectively, $c > 0$ is the clipping threshold (default $c = 10$), and $\epsilon = 10^{-8}$ prevents division by zero. Subsequently, $\hat{\boldsymbol{x}} \in \mathbb{R}^d$ is zero-padded to form $\hat{\boldsymbol{x}}_{\text{pad}} \in \mathbb{R}^N$:

$$(\hat{\boldsymbol{x}}_{\text{pad}})_j = \begin{cases} \hat{x}_j, & 0 \leq j < d, \\ 0, & d \leq j < N. \end{cases} \tag{15}$$

Define the *effective data angle* vector for UCR encoding

$$\boldsymbol{\alpha} = \lambda\, \hat{\boldsymbol{x}}_{\text{pad}} \in \mathbb{R}^N, \tag{16}$$

where the scaling factor

$$\lambda = \begin{cases} \pi/c, & \text{full-range mode}, \quad \alpha_i \in [-\pi,\ \pi], \\ \dfrac{\pi}{2c}, & \text{half-range mode}, \quad \alpha_i \in [-\pi/2,\ \pi/2]. \end{cases} \tag{17}$$

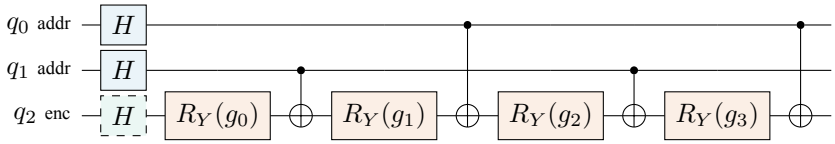
α_j directly represents the target controlled rotation angle corresponding to the address state $|j\rangle$. The default configuration uses full-range ($\lambda = \pi/c$), so that $\alpha_j \in [-\pi, \pi]$ holds for all j.

3.4. Gate-Level Instantiation of UCR Encoding

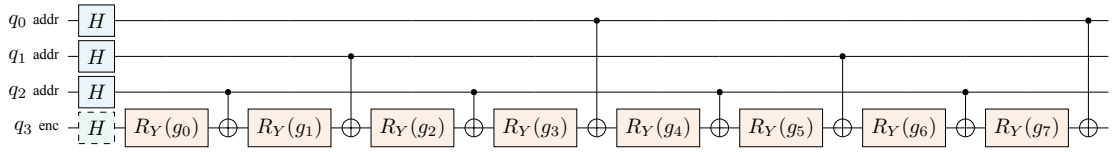
Applying the UCR decomposition given in Section 2.2 (Eq. (7), Eq. (8), Eq. (9)) to the N and $\boldsymbol{\alpha}$ determined by Eq. (13) and Eq. (16) yields a gate-level instance of the UCR encoding. This section first establishes its topological properties, then provides an equivalent representation of the encoder's target quantum state.

Proposition 1 (Fixed gate topology)**.** *In the zero-compression configuration, the control indices c_i in Eq. (9) depend only on N and k, and do not depend on the data value $\boldsymbol{x}$ or any trainable parameters. Therefore, the logic gate topology of the UCR encoding remains fixed throughout the training process.*

Proof. Since adjacent Gray codes differ by only one bit, the XOR of $G(i) \oplus G(i+1)$ is exactly a power of 2, and its value is uniquely determined by i. Applying ctz to this value returns the corresponding bit position, which remains a function of i alone. Under zero compression, all N rotation gates are retained; therefore, the number of gates and their interconnections are entirely determined by (N, k) and are independent of the data. □



(a) Q=3, $\mathbf{c} = [1, 0, 1, 0]$



(b) Q=4, $\mathbf{c} = [2, 1, 2, 0, 2, 1, 2, 0]$

Fig. 2. UCR circuit topology at Q=3 and Q=4. Topology depends only on (N, k) with Q=k+1, N=2^k. Dashed: optional encoding-qubit Hadamard.

Remark 2. *The current training framework supports only zero-compression configurations (*compression $= 0$*). Compressed versions (*compression > 0*) would skip zero-coefficient gates based on data values, making the gate topology dependent on the data; this does not fall within the differentiable training path of this work.*[30]

3.4.1. Encoder Target Quantum State

Taking the configuration (R_Y, no Hadamard initialization on the encoding qubit, as shown in Fig. 2) as an example, the target quantum state prepared by the encoder is

$$|\Psi(\boldsymbol{\alpha})\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle_{\text{addr}} \otimes \left(\cos\frac{\alpha_j}{2}\,|0\rangle + \sin\frac{\alpha_j}{2}\,|1\rangle\right)_{\text{enc}}. \tag{18}$$

This state is mathematically equivalent to the state prepared by the logical gate sequence Eq. (8).

Remark 3 (Relationship with Conventional Amplitude Encoding)**.** *For an observation vector $\hat{\boldsymbol{x}}_{\text{pad}} \in \mathbb{R}^N$, the quantum state prepared by conventional amplitude encoding is*

$$|\Psi_{\text{AE}}(\hat{\boldsymbol{x}}_{\text{pad}})\rangle = \frac{1}{\|\hat{\boldsymbol{x}}_{\text{pad}}\|_2} \sum_{j=0}^{N-1} (\hat{\boldsymbol{x}}_{\text{pad}})_j\,|j\rangle\,. \tag{19}$$

Therefore, the feature components after global normalization directly determine the amplitudes in the computational basis. Global normalization couples all features while naturally losing radial scale information, resulting in a state-dependent and rank-deficient input geometry. In contrast, Eq. (18) maintains uniform weights in the address register and maps each effective data angle α_j to an address-controlled rotation acting on the encoding qubit. The two mappings differ in terms of where the data enter the quantum state, the normalization method, the qubit allocation, and how features are coupled.

Configuration A: R_Y without Hadamard (default). The initial state of the encoding qubit is $|0\rangle$:

$$R_Y(\alpha_i)\,|0\rangle = \cos\frac{\alpha_i}{2}\,|0\rangle + \sin\frac{\alpha_i}{2}\,|1\rangle\,, \tag{20}$$

$$\langle Z\rangle_i \;\equiv\; \langle\psi_i|\,Z\,|\psi_i\rangle = \langle 0|\,R_Y^\dagger(\alpha_i)\,Z\,R_Y(\alpha_i)\,|0\rangle = \cos^2\frac{\alpha_i}{2} - \sin^2\frac{\alpha_i}{2} = \cos(\alpha_i). \tag{21}$$

For the full-range mode ($\alpha_i \in [-\pi, \pi]$), $\langle Z\rangle_i \in [-1, 1]$; for the half-range mode ($\alpha_i \in [-\pi/2, \pi/2]$), $\langle Z\rangle_i \in [0, 1]$.

Configuration B: R_Y with Hadamard. The initial state of the encoding qubit is $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$:

$$R_Y(\alpha_i)\ |+\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} \cos\frac{\alpha_i}{2} - \sin\frac{\alpha_i}{2} \\ \sin\frac{\alpha_i}{2} + \cos\frac{\alpha_i}{2} \end{pmatrix}, \tag{22}$$

$$\langle Z\rangle_i = -\sin(\alpha_i). \tag{23}$$

This configuration offers practical advantages for RL observations: Since shared normalization centers the data around zero, the response function $-\sin(\alpha_i)$ has maximum sensitivity at the data centroid $\left|\frac{\mathrm{d}}{\mathrm{d}\alpha_i}(-\sin\alpha_i)\right|_{\alpha_i=0} = 1$, whereas the default $\cos(\alpha_i)$ has a derivative of zero at the same point.

Configuration C: R_Z (phase encoding). The data is encoded in terms of relative quantum phase rather than amplitude:

$$R_Z(\alpha_i)\ |0\rangle = \mathrm{e}^{-\mathrm{i}\alpha_i/2}\ |0\rangle\,. \tag{24}$$

Since $R_Z(\alpha_i)\,|0\rangle$ differs from $|0\rangle$ only by a global phase, local Z measurements on the encoded qubit always yield $\langle Z_{\text{enc}}\rangle = 1$ (independent of α_i). The encoded information must be extracted by a subsequent variational quantum circuit (VQC) through interference, which converts the phase difference into an amplitude difference, thereby increasing the computational burden on the VQC.

3.5. Default StatePrep-Equivalent Implementation for UCR Encoding During Training

Although logical UCR is implemented by the rotation–CNOT gate sequence defined in Eq. (8), during training in a classical simulator, to improve simulation speed, this gate-level decomposition is not explicitly performed by default. Instead, the state vector $\boldsymbol{v}(\boldsymbol{\alpha})$ is first analytically constructed on the classical side, and then injected into the simulator via a single `StatePrep` call. The default training pipeline is

$$\boldsymbol{x} \xrightarrow{\text{normalization}} \hat{\boldsymbol{x}} \xrightarrow{\text{zero-padding}} \hat{\boldsymbol{x}}_{\text{pad}} \xrightarrow{\times\lambda} \boldsymbol{\alpha} \xrightarrow{\Phi(\cdot)} \boldsymbol{v}(\boldsymbol{\alpha}) \xrightarrow{\texttt{StatePrep}} |\Psi(\boldsymbol{\alpha})\rangle\,, \tag{25}$$

where $\Phi : \mathbb{R}^N \to \mathbb{C}^{2N}$ is the state-vector construction map (see Table 2 for its analytical form). The role of the matrix $\boldsymbol{M}$ is to translate $\boldsymbol{\alpha}$ into the gate angles $\boldsymbol{g}$ required for gate-level decomposition; once the target state $\boldsymbol{v}(\boldsymbol{\alpha})$ is constructed directly, this "gate-parameter interface transformation" is no longer needed, so the StatePrep path does not explicitly compute $\boldsymbol{g} = \boldsymbol{M}\boldsymbol{\alpha}$. Therefore, the main noise-free PPO experiments reported below all use the `StatePrep` path, while the explicit gate-level path is used for numerical equivalence verification, resource accounting, and gate-level noise analysis.

Comparison with the gate-level path:

$$\underbrace{\hat{\boldsymbol{x}}_{\text{pad}} \xrightarrow{\times\lambda} \boldsymbol{\alpha} \xrightarrow{\boldsymbol{M}\cdot} \boldsymbol{g} \xrightarrow{k+2N \text{ gates}} |\Psi\rangle}_{\text{gate-loop path}} \quad\longleftrightarrow\quad \underbrace{\hat{\boldsymbol{x}}_{\text{pad}} \xrightarrow{\times\lambda} \boldsymbol{\alpha} \xrightarrow{\Phi} \boldsymbol{v} \xrightarrow{\texttt{StatePrep}} |\Psi\rangle}_{\text{state preparation path}}. \tag{26}$$

The two paths share the prefix $\hat{\boldsymbol{x}}_{\text{pad}} \xrightarrow{\times\lambda} \boldsymbol{\alpha}$ and diverge in the middle: The gate-loop path continues with a left multiplication by $\boldsymbol{M}$ to obtain the gate angle, then expands into $k + 2N$ gate simulations, while the state-preparation path directly produces the state vector $\boldsymbol{v}$ via Φ (constructed using trigonometric functions) and injects it into the simulator via a single `StatePrep`. The former retains the matrix multiplication by $\boldsymbol{M}$, while the latter omits it entirely.

Table 2. Analytic forms of the StatePrep state vector $\boldsymbol{v}(\boldsymbol{\alpha})$ ($j = 0, \ldots, N-1$).

Configuration	v_{2j}	v_{2j+1}
R_Y, no H	$\frac{1}{\sqrt{N}} \cos\frac{\alpha_j}{2}$	$\frac{1}{\sqrt{N}} \sin\frac{\alpha_j}{2}$
R_Y, with H	$\frac{1}{\sqrt{N}} \cos\left(\frac{\alpha_j}{2} + \frac{\pi}{4}\right)$	$\frac{1}{\sqrt{N}} \sin\left(\frac{\alpha_j}{2} + \frac{\pi}{4}\right)$
R_Z, no H	$\frac{1}{\sqrt{N}} \mathrm{e}^{-\mathrm{i}\alpha_j/2}$	0
R_Z, with H	$\frac{1}{\sqrt{2N}} \mathrm{e}^{-\mathrm{i}\alpha_j/2}$	$\frac{1}{\sqrt{2N}} \mathrm{e}^{+\mathrm{i}\alpha_j/2}$

Table 2 provides the analytical forms of the state vector $\boldsymbol{v}(\boldsymbol{\alpha}) \in \mathbb{C}^{2N}$ for all four configurations. The state vector uses an interleaved indexing scheme: v_{2j} corresponds to $|j, 0\rangle$, and v_{2j+1} corresponds to $|j, 1\rangle$ ($j = 0, \ldots, N-1$).

The four configurations can be unified into the following compact form. Let the initial state of the encoding qubit be $|\xi\rangle = |0\rangle$ (no Hadamard) or $|\xi\rangle = |+\rangle$ (with Hadamard), and let $R \in \{R_Y, R_Z\}$ be a rotation operator, then

$$|\Psi(\boldsymbol{\alpha})\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle_{\text{addr}} \otimes R(\alpha_j) \, |\xi\rangle_{\text{enc}} \, . \tag{27}$$

Its $2N$-dimensional coordinate vector in the computational basis $\{|j, e\rangle\}$ is given by $\boldsymbol{v}$. Since each component of this vector is an elementary trigonometric or complex exponential function of α_j, it is differentiable with respect to $\boldsymbol{\alpha}$.

3.6. Gradient Properties of UCR Encoding

The scope of the following analysis is: **within the UCR encoding layer, after shared normalization and before VQC**. The Jacobian of the encoding layer has constant scaling or constant norm properties; the end-to-end gradient of the complete policy network must also be multiplied by the Jacobians of VQC and the subsequent classical network, whose behavior is determined by the parameters and structure of these downstream modules. It is important to note that this section analyzes how changes in the input affect the encoded state, rather than the gradients of the PPO loss with respect to the trainable VQC parameters. The latter, namely $\partial\mathcal{L}/\partial\theta_j$, is examined separately in the experimental section (see Section 4.7); therefore, the properties of the Jacobian derived in this section should be understood solely as local conditioning properties of the UCR encoding sublayer, and should not be regarded as sufficient conditions for favorable downstream VQC parameter-gradient trainability or deep-circuit convergence.

3.6.1. Jacobian of the Logical Gate-Level Decomposition

Consider the logical gate-level path $\hat{\boldsymbol{x}}_{\text{pad}} \mapsto \boldsymbol{g}$. From Eq. (7) and Eq. (16), we have $\boldsymbol{g} = \boldsymbol{M}\boldsymbol{\alpha} = \lambda \boldsymbol{M} \hat{\boldsymbol{x}}_{\text{pad}}$, and its Jacobian is

$$\frac{\partial \boldsymbol{g}}{\partial \hat{\boldsymbol{x}}_{\text{pad}}} = \lambda \, \boldsymbol{M}. \tag{28}$$

Since $\boldsymbol{P}_{\text{gray}}$ is an orthogonal matrix and $(1/\sqrt{N})\boldsymbol{H}_N$ is also an orthogonal matrix,

$$\boldsymbol{M}^\top \boldsymbol{M} = \frac{1}{N^2} \boldsymbol{H}_N^\top \underbrace{\boldsymbol{P}_{\text{gray}}^\top \boldsymbol{P}_{\text{gray}}}_{=\boldsymbol{I}_N} \boldsymbol{H}_N = \frac{1}{N^2} \boldsymbol{H}_N^2 = \frac{1}{N} \boldsymbol{I}_N, \tag{29}$$

Based on the PennyLane quantum computing framework.[32]

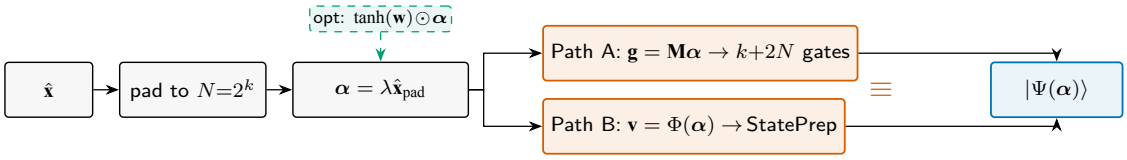


Fig. 3. UCR encoding principle. Classical preprocessing converts $\hat{\mathbf{x}}$ into angles $\boldsymbol{\alpha}$; the gate-template path and the state-preparation path produce numerically equivalent $|\Psi(\boldsymbol{\alpha})\rangle$. Dashed: optional trainable input scaling.

where the second equality uses the symmetry of $\boldsymbol{H}_N$, and the third equality uses Eq. (4). Since all singular values of $\boldsymbol{M}$ are $1/\sqrt{N}$, the singular values of the mapping $\hat{\boldsymbol{x}}_{\text{pad}} \mapsto \boldsymbol{g}$ satisfy

$$\sigma_i\left(\frac{\partial \boldsymbol{g}}{\partial \hat{\boldsymbol{x}}_{\text{pad}}}\right) = \frac{\lambda}{\sqrt{N}}, \qquad i = 1, \ldots, N, \tag{30}$$

where $\sigma_i(\cdot)$ denotes the i-th singular value. This indicates that the logical gate-level encoding submap is a constant scaling in all directions.

3.6.2. Gradient Norm of the StatePrep Path

Under the default training path, the encoding Jacobian is no longer a linear constant matrix because the state vector is composed of trigonometric functions. Taking the default R_Y path without Hadamard initialization on the encoding qubit as an example,

$$\frac{\partial v_{2j}}{\partial \alpha_j} = -\frac{1}{2\sqrt{N}} \sin\frac{\alpha_j}{2}, \qquad \frac{\partial v_{2j+1}}{\partial \alpha_j} = \frac{1}{2\sqrt{N}} \cos\frac{\alpha_j}{2}, \tag{31}$$

and when $l \neq j$, $\partial v_{2j}/\partial \alpha_l = \partial v_{2j+1}/\partial \alpha_l = 0$ (the state vector elements at different addresses are uncoupled). The Euclidean norm of each column of the Jacobian is

$$\left\| \frac{\partial \boldsymbol{v}}{\partial \alpha_j} \right\|_2 = \sqrt{\frac{\sin^2(\alpha_j/2)}{4N} + \frac{\cos^2(\alpha_j/2)}{4N}} = \frac{1}{2\sqrt{N}}. \tag{32}$$

which is independent of the data values α_j. In other words, although the StatePrep path is not a linear mapping, it still avoids data-dependent gradient explosion or vanishing gradients within the UCR encoding sublayer.

3.6.3. Trainable Input Scaling

If trainable input scaling is enabled (disabled by default, as shown in Fig. 3), the following is performed before encoding:

$$\tilde{\boldsymbol{\alpha}} = \tanh(\boldsymbol{w}) \odot \boldsymbol{\alpha}, \tag{33}$$

where $\boldsymbol{w} \in \mathbb{R}^N$ is a learnable parameter, $\odot$ denotes element-wise multiplication, and is initialized as $w_j = 2.0$ (such that $\tanh(2.0) \approx 0.964$, approximating the identity mapping). In this case, the isotropic scaling property described above holds only for the $\hat{\boldsymbol{x}}_{\text{pad}} \mapsto \boldsymbol{\alpha}$ sub-path; the overall Jacobian is additionally affected by $\text{diag}(\tanh(\boldsymbol{w}))$.

3.7. Complexity Analysis and Structural Comparison of Encoding Schemes

We compare the complexity differences among three types of front-end interface encoding schemes. The resource counts are computed for individual front-end encoding blocks (excluding VQC and post-classical readout, and disregarding the actor/critic duplication). Let $d \geq 2$ be the observation dimension, $k = \lceil \log_2 d \rceil$, $N = 2^k$; hereinafter, we denote the number of single-qubit rotation gates as n_R, the number of Hadamard gates as n_H, the number of CNOT gates as n_{CNOT}, the circuit depth as D, the qubit budget as Q, and the number of trainable front-end parameters as n^{θ}_{pre}.

UCR encoding. The logical UCR decomposition consists of k address-register Hadamard gates, an optional $\delta_H \in \{0, 1\}$ Hadamard gate on the encoding qubit, N single-qubit rotation gates, and N CNOT gates. The logical gate counts are

$$n_H = k + \delta_H, \quad n_R = N, \quad n_{\text{CNOT}} = N, \quad n_{\text{total}} = k + \delta_H + 2N. \tag{34}$$

For sequential execution, the depth is

$$D_{\text{logical}} = 1 + \delta_H + 2N, \tag{35}$$

where the first term 1 arises from the fact that the k address Hadamard gates can be executed in parallel on different qubits; D_{logical} grows linearly with N. Combining this with Eq. (13), the qubit budget of UCR and the number of trainable front-end parameters are

$$Q_{\text{UCR}} = k + 1, \qquad n^{\theta}_{\text{pre}} = \begin{cases} 0, & \text{default configuration,} \\ N, & \text{when Eq. (33) is enabled.} \end{cases} \tag{36}$$

The default training path does not explicitly execute the above gate sequence, but instead calls `StatePrep` once after constructing the state vector $\boldsymbol{v}(\boldsymbol{\alpha})$ on the classical side (Eq. (25)). The encoding cost during training consists of two parts: (i) zero-padding, scaling, and trigonometric calculations on the classical side; (ii) the execution overhead of a single `StatePrep` primitive on the simulator. This metric is not directly comparable to the number of gates at the logical gate level. Additionally, the R_Y configuration produces real-valued state vectors, which are compatible with all simulation backends; whereas R_Z generates complex-valued state vectors. Since Lightning's C++ layer only accepts real-valued parameters, this triggers a type compatibility issue, causing the system to automatically fall back to the logical-gate-level UCR path without executing the StatePrep path.

Angle Encoding (Angle-R_Y). After a trainable classical preprocessing network $\mathcal{P}_{\text{pre}} : \mathbb{R}^d \to \mathbb{R}^Q$ outputs a Q-dimensional bounded angle vector, the resulting angles are encoded into Q qubits through R_Y rotations:

$$Q_{\text{Angle}} = Q \text{ (hyperparameters)}, \quad n_R = Q, \quad n_H = 0, \quad n_{\text{CNOT}} = 0, \quad D = 1. \tag{37}$$

The quantum circuit side is the shallowest and sparsest, but the classical cost of the front end increases with d: when implemented using a two-layer $\mathbb{R}^d \to \mathbb{R}^h \to \mathbb{R}^Q$ architecture

$$n^{\theta}_{\text{pre}} = O(dQ + h(d + Q)), \tag{38}$$

where h is the hidden layer width of the preprocessing network; the burden of dimensionality reduction and compression is offloaded to the classical module.

Amplitude Encoding. Directly perform state preparation on the zero-padded and L^2-normalized observation state vector:

$$Q_{\text{Amp}} = k, \quad n^{\theta}_{\text{pre}} = 0, \quad n_R = \Theta(N), \quad n_{\text{CNOT}} = \Theta(N), \quad D = \Theta(N). \tag{39}$$

Here, we use the asymptotic tight bound notation $\Theta(\cdot)$ rather than the upper bound $O(\cdot)$: Möttönen et al.'s state preparation construction provides an $O(N)$-order upper bound for n_R and n_{CNOT} (of the same order as the depth when executed sequentially);[14] Shende–Markov–Bullock and Plesch–Brukner further proved that the lower bound on the number of CNOT operations required for any n qubit state preparation is $\Omega(2^n) = \Omega(2^{Q_{\text{Amp}}}) = \Omega(N)$;[33,34] Since the upper and lower bounds match, the tight bound $\Theta(N)$ on n_R and n_{CNOT} cannot be improved; the depth $D = \Theta(N)$ is

Table 3. Comparison of quantum encoding scheme structures. UCR assumes a zero-compression configuration (Remark 2) and, by default, does not enable trainable scaling (Eq. (33)) or Hadamard initialization on the encoding qubit (Section 3.4.1); under the state preparation path, the number of encoding gates is reduced to a single call to `StatePrep`. h is the hidden layer width of the trainable preprocessing network in angle encoding.

Property	Angle Encoding	Amplitude Encoding	UCR Encoding
Number of qubits Q	Hyperparameters	k	$k+1$
Classical preprocessing	MLP (trainable)	L^2 normalization	SFWHT (fixed)
Trainable preprocessing parameters	$O(dQ + h(d+Q))$	0	0
Encoding capacity	Q features	2^Q amplitudes	2^{Q-1} features
Encoding Jacobian	Nonlinear	Nonlinear	Isotropic constant scaling $\lambda \boldsymbol{M}$
Gradient pathology risk	tanh saturation	Normalized denominator	none in the encoding layer (see Section 3.6)
Number of encoding gates	Q	$\Theta(N)$	$k + 2N$ / 1 (StatePrep)
Circuit topology	Fixed	Fixed	Fixed (Proposition 1)

an estimate of the same order under the sequential execution assumption. The qubit budget reaches the information-theoretic lower bound, but the gate-level expansion involves multi-control rotation decompositions; the asymptotic upper bound is of the same order as UCR, but the hidden constant is significantly larger. In this work, the experiment uses `qml.AmplitudeEmbedding` to invoke the simulator's native preparation to avoid explicit decomposition; in the noisy evaluations, an explicit `MottonenStatePreparation` is used to account for gate-level noise and resource costs.

UCR encoding occupies a unique position (Table 3): it offers data capacity comparable to amplitude encoding (2^{Q-1} vs. 2^Q features) without requiring trainable preprocessing parameters; In terms of depth, angle encoding is always of depth 1; UCR and amplitude encoding are both of the order of $\Theta(N)$, but UCR's topology is data-independent (Proposition 1), whereas amplitude encoding hides a larger constant due to multi-control decomposition; In the encoding Jacobian, angle encoding is affected by tanh saturation and the amplitude is affected by the normalization denominator, both of which are nonlinear; UCR encoding exhibits isotropic constant scaling with a singular value of $\lambda/\sqrt{N}$ at the logic gate level (Eq. (30)); under the state preparation path, the Jacobian column norm is always $1/(2\sqrt{N})$ (Eq. (32)), both of which are independent of data values, ensuring that the gradient flow through the encoding layer has good conditionality; On the classical side, UCR performs only a fixed linear transformation of $O(N \log N)$ complexity, while on the quantum side, the topology is fixed and analytically equivalent to a single `StatePrep` operation (Eq. (26), Section 3.5). In summary, UCR achieves a more balanced trade-off between complexity and expressiveness.

4. Experiments

4.1. Experimental Objectives

Research Hypothesis: under a unified PPO backend, a unified training budget, and unified observation normalization, can a fixed topology UCR frontend achieve a more stable performance-resource trade-off among typical quantum encoding methods?

The experiments in this paper uniformly adopt the typical reinforcement learning algorithm Proximal Policy Optimization (PPO); the actor–critic formulations in Section 3.2 describe the implementation flexibility of the framework, while all experiments reported in this paper consistently use a quantum actor and a classical critic. The Classical MLP

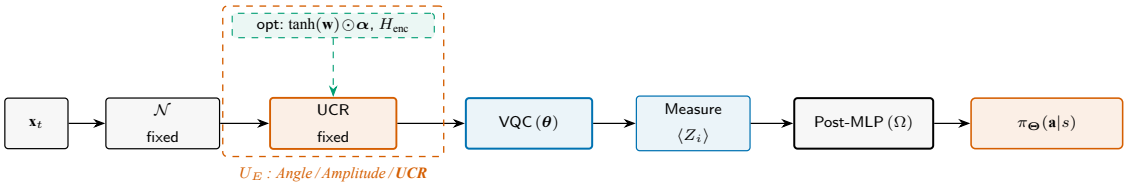


Fig. 4. Quantum policy network architecture. The normalized observation sequentially passes through the encoding interface (orange dashed box), the variational quantum circuit, the Pauli-Z measurement, and the classical readout, ultimately outputting the policy $\pi_{\boldsymbol{\Theta}}(\mathbf{a}|s)$. The encoding interface U_E can be implemented using Angle, Amplitude, or UCR; this paper adopts the UCR encoding scheme. The trainable parameters are $\boldsymbol{\theta}$ and Ω. Thick borders indicate trainable blocks, and the green dashed box denotes the optional UCR module.

	Classical map	Quantum encoding	Circuit cost
Angle-R_Y	trainable MLP + $\tanh\pi$ $d \to Q$	parallel $R_Y(\alpha_i)$ on Q qubits	Q rotations 0 CNOT
Amplitude	pad to 2^Q, L_2-normalize no trainable params	Möttonen state preparation global cascade	$\sim 2^Q$ rotations $\sim 2^Q - 1$ CNOT
UCR (ours)	fixed $\mathbf{g} = \mathbf{M}\boldsymbol{\alpha}$ $\mathbf{M}$ deterministic	fixed UCR cascade address-controlled R_Y	k H, N R_Y, N CNOT topology fixed

Fig. 5. Three encoding interfaces under a unified classical-to-quantum pipeline. Angle-R_Y places representation capacity in a trainable classical front-end; amplitude encoding removes the front-end classical neural network; UCR uses a fixed transform $\mathbf{g} = \mathbf{M}\boldsymbol{\alpha}$ and a topology determined only by (N, k), where $Q = k + 1$ and $N = 2^k$.

serves primarily as a reference upper bound and a calibration item for the training protocol in this paper; it is not intended to prove that hybrid quantum-classical PPO is generally superior to classical PPO. This paper maintains consistency in the PPO framework, training budget, evaluation protocol, actor/critic readout method, and random seeds within the same environment. It solely compares the performance, stability, and resource efficiency of three typical encoding interfaces—angle encoding, amplitude encoding, and UCR encoding—under the same PPO framework. (See Fig. 4)

Angle-R_Y first reduces the dimension of normalized observations to Q dimensions via a classical preprocessing network, then compresses them using $\tanh(\cdot)\,\pi$ before feeding them into `AngleEmbedding`. Amplitude encoding zero-pads and L_2-normalizes the normalized observations before feeding them into amplitude embedding. UCR, on the other hand, starts from the normalized observations, generates rotation angles via zero-padding and a fixed Walsh–Hadamard–Gray linear transformation, and writes them into a fixed topology uniquely determined by the input dimension. Changes in the sample only alter the rotation angles, without changing the gate connection relationships (see Fig. 5). Unless otherwise stated, all main UCR experiments initialize the encoding qubit with a Hadamard gate.

For our main experiments, we select five Gymnasium environments: CartPole-v1, Acrobot-v1, MountainCarContinuous-v0, LunarLanderContinuous-v3, and BipedalWalker-v3, covering discrete and continuous actions, low- to high-dimensional observations, and varying levels of dynamical difficulty. Additionally, Ant-v5 and Humanoid-v5 are included only for high-dimensional feasibility checks in the appendix and are not included in the main-text claims. For discrete actions, we use a categorical policy, while for continuous actions, we employ a Jacobian-corrected tanh-Gaussian policy. For continuous experiments, the entropy regularization uniformly uses a pre-squash Gaussian entropy proxy rather than the exact entropy corresponding to strictly bounded distributions. Observations are normalized using Welford online normal-

Table 4. Main environment training and configuration. Steps: total number of training steps; $Batch$: PPO batch size; N_{env}: number of parallel environments; η_π, η_V: initial learning rates for the actor and critic, respectively (in units of 10^{-4}); $c_{\mathcal{H}}$: initial value of the entropy coefficient. UCR and Amplitude formats are $(Q, L_q;\ \texttt{post})$, Angle-R_Y is $(Q, L_q;\ \texttt{pre}\,|\,\texttt{post})$; The comma-separated widths a, b denote the widths of the two MLP layers. "—" indicates that the module is not included.

Env	Steps	$Batch/N_{\text{env}}$	η_π/η_V	$c_{\mathcal{H}}$	h	UCR	Amp	Angle-R_Y
CartPole-v1	4×10^5	$256/16$	$2.5/2.5$	0	64	$3, 2;\ 4$	$2, 2;\ 4$	$3, 2;\ 8\,\vert\,4$
Acrobot-v1	4×10^5	$256/16$	$5/6$	0.02	64	$4, 2;\ 8$	$3, 2;\ 8$	$4, 2;\ 12\,\vert\,8$
MountainCarC-v0	4×10^5	$256/16$	$3/3$	0.01	64	$2, 2;\ 2$	$1, 2;\ 2$	$2, 2;$ —$\,\vert\,2$
LunarLanderC-v3	1.5×10^6	$64/8$	$3/3$	0.01	64	$4, 2;\ 8$	$3, 2;\ 8$	$4, 2;\ 16\,\vert\,8$
BipedalWalker-v3	1×10^7	$4096/32$	$5/4$	0.001	128	$6, 3;\ 8, 8$	$5, 3;\ 8, 8$	$6, 3;\ 16, 16\,\vert\,8, 8$

Shared hyperparameters: GAE $\lambda = 0.95$, PPO clip $\epsilon = 0.2$, discount factor $\gamma = 0.99$, value loss coefficient $c_v = 0.5$, gradient norm clipping 0.5. Both the learning rate and the entropy coefficient decay linearly to 10% of their initial values.

ization. The normalizer shares a frozen snapshot during rollouts and updates, and is updated only after the entire rollout is complete to prevent normalization drift from affecting the PPO ratio. The variational quantum layers following each front-end encoder uniformly employ ring entanglement, R_Y rotation layers, and Pauli-Z measurements; the classical simulation backend uses PennyLane `default.qubit`. The main experiments uniformly use 10 training seeds. Deterministic evaluations are performed periodically; for each seed, 10 episodes are run using an independent evaluation seed, and the results are averaged. Throughout the paper, training seeds are used as the statistical unit. The main environment training and structural settings are listed in Table 4.

This paper selects appropriate evaluation metrics based on different experimental objectives. "Best Eval Return" indicates the maximum performance that can be achieved during training and is suitable for comparing main tasks and analyzing performance ceilings in structural ablation studies. "Final Eval Return" indicates how much performance is retained at the end of training and is suitable for comparing the usability of final policies in main tasks. AUC comprehensively reflects both the learning speed and the duration of high returns, making it suitable for comparing sample efficiency under the same training budget. IQM is used to robustly aggregate return performance toward the end of training, reducing the impact of a few outlier high scores or training collapse results,[35] and is suitable for experiments involving input dimension expansion. Therefore, Best Eval Return, Final Eval Return, AUC, and IQM focus on peak performance, final checkpoint performance, sample efficiency, and robust statistical performance, respectively. The metrics used in the subsequent VQC gradient diagnostics and gate-level noise experiments are described in the corresponding subsections.

Gate-level noise protocol and statistics. The main experiments use the analytical StatePrep equivalent path, while the noise evaluation uses explicit gate-level circuits. After circuit expansion, a noise channel with parameter p_1 is applied after each single-qubit gate, and a noise channel with parameter

$$p_2 = 10p_1. \tag{40}$$

Noise covers all quantum gates in the encoding front end, VQC, and final quantum rotations. The tested noise strengths are

$$p_1 \in \left\{0, 10^{-4}, 3\times10^{-4}, 10^{-3}, 3\times10^{-3}, 10^{-2}\right\}. \tag{41}$$

Measurement noise is turned off, and noisy circuits are evaluated using density matrix evolution and analytical expectation values. Amplitude encoding uses explicit state preparation decomposition in noisy evaluations to fully account for its state preparation gates and corresponding noise costs. All noise results in the main text use the final checkpoint.

Table 5. Dual-path consistency check: Differences between the differentiable StatePrep path and the explicit gate template path in terms of VQC measurement expectation values, policy distribution parameters, and value predictions.

Cfg	Env	N	$\|\Delta\langle Z\rangle\|_\infty$	$\|\Delta\mu\|_\infty$	Speedup	Verdict
V1	CartPole-v1 (RY)	4	$\sim 3.06\times10^{-7}$	$\sim 7.63\times10^{-6}$	$\sim 1.50\times$	PASS
V2	CartPole-v1 (RY,Hadamard-off)	4	$\sim 3.87\times10^{-7}$	$\sim 4.59\times10^{-6}$	$\sim 1.39\times$	PASS
V3	LunarLanderContinuous-v3 (RY)	8	$\sim 2.68\times10^{-7}$	$\sim 6.11\times10^{-6}$	$\sim 1.72\times$	PASS
V4	BipedalWalker-v3 (RY)	32	$\sim 2.83\times10^{-7}$	$\sim 4.47\times10^{-6}$	$\sim 3.26\times$	PASS
V5	CartPole-v1 (RZ)	4	$\sim 4.17\times10^{-7}$	$\sim 3.70\times10^{-6}$	$\sim 1.53\times$	PASS

To directly compare UCR and Angle-R_Y, we define the difference in noisy returns for the same seed

$$\Delta R_{\mathrm{U-A}} = R^{\mathrm{UCR}}_{\mathrm{noisy}} - R^{\mathrm{Angle}}_{\mathrm{noisy}}. \tag{42}$$

A positive value indicates that UCR achieves a higher actual noisy return, while a negative value indicates that Angle-R_Y achieves a higher one. To describe the extent to which an encoding retains its performance relative to its own noise-free performance, we define

$$S_{\mathrm{norm}} = \frac{R_{\mathrm{noisy}} - R_{\mathrm{random}}}{R_{\mathrm{clean}} - R_{\mathrm{random}}}. \tag{43}$$

Here, $S_{\mathrm{norm}} = 1$ indicates that the performance of the noise-free strategy is preserved relative to the random strategy, while $S_{\mathrm{norm}} = 0$ indicates a degradation to the level of the random strategy. When $|R_{\mathrm{clean}} - R_{\mathrm{random}}| < 10$, the normalization denominator is too small, and that seed is excluded from the retention summary; the number of valid seeds is denoted by n_{valid}. S_{norm} is used solely to measure the degree to which a given encoding retains its own noise-free performance, and is not used by itself for absolute performance ranking across encodings.

Statistical inference is based on independent training seeds, not evaluation episodes. The main noise scan and multi-channel experiments use 10 seeds per grid cell, while the dimensionality experiments use 5 seeds per grid cell. Inter-encoder comparisons use paired differences within the same seed, and $B = 20000$ percentile bootstrap samples are used to compute pointwise 95% confidence intervals; these intervals are used to describe differences at predefined noise points.

4.2. Dual-Path Consistency Verification

UCR distinguishes between two implementation paths: the StatePrep path and the logic gate template path, with the latter used for gate-level resource accounting and noise evaluation. The former supports efficient classical computer simulation training, while the latter supports gate count, depth, and hardware interpretability. The Methods section has already demonstrated that, under classical input data with zero compression (compression $= 0$), UCR's differentiable StatePrep path is logically equivalent to the explicit gate template path.

To rule out various risks at the implementation level, this section supplements the analysis with a minimal numerical sanity check, without repeating the full RL training for the gate template path. We compare three levels: L1 VQC measurement expectations $\langle Z_i\rangle$, L2 policy distribution parameters, and L3 single-forward execution time. Five sets of configurations cover H_{enc} on/off, R_Y/R_Z, discrete/continuous actions, and $N \in \{4, 8, 32\}$. The pass threshold is set to $\tau = \max(10N\varepsilon_{32}, 10^{-6})$. The formal PASS/FAIL verdict is determined by L1; L2 and L3 serve as downstream consistency and computational-efficiency diagnostics, respectively.

Table 5 compares the two paths of UCR; L1 and L2 exhibit negligible numerical discrepancies, whereas L3 reports execution efficiency. $\|\Delta\langle Z\rangle\|_\infty$ remains stable between 2.7×10^{-7} and 4.2×10^{-7}, while $\|\Delta\mu\|_\infty$ does not exceed

7.6×10^{-6}; these errors do not accumulate significantly as N increases. Additionally, the complex-valued state vector produced in the R_Z case passed consistency checks, indicating that dtype handling did not introduce additional bias.

In terms of speed, the speedup ratio of the StatePrep path relative to the explicit gate template path increases as N grows. At $N = 4$, it is approximately 1.4–1.5 times faster; at $N = 8$, it reaches 1.7 times faster; and at $N = 32$, it reaches 3.3 times faster. Therefore, using StatePrep in simulation training is a reasonable choice in terms of efficiency; At the same time, the gate template path can still be used for gate-level resource accounting and subsequent gate-level noise evaluation. Based on the above, the dual-path implementation using UCR encoding is entirely suitable for the comparative experiments in the remainder of this paper.

4.3. Performance Comparison of Main Tasks

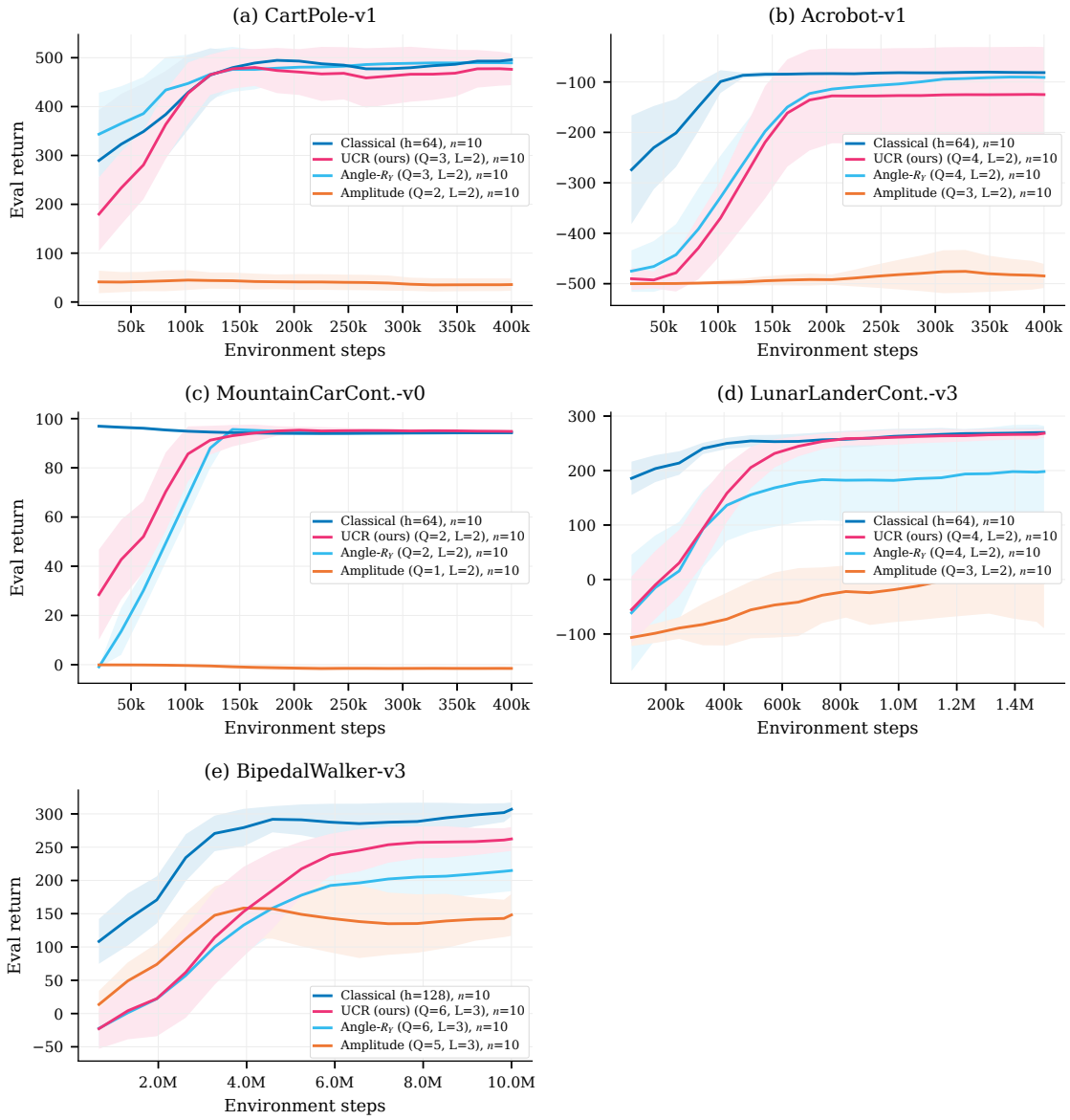


Fig. 6. Learning curves on the five main Gymnasium tasks: (a) CartPole-v1, (b) Acrobot-v1, (c) MountainCarContinuous-v0, (d) LunarLanderContinuous-v3, and (e) BipedalWalker-v3. The horizontal axis represents the number of environment steps, the vertical axis represents the mean evaluation return, and the shaded regions indicate the 95% confidence intervals across 10 seeds.

The results for the five main tasks (see Fig. 6 and Table 6, Table 7, Table 8, and Table 9) show that UCR exhibits

Table 6. Main comparison on 5 standard RL benchmarks (obs_dim $\in [2, 24]$, 10 seeds each). **Best Eval Return**; mean with 95% bootstrap CI.

Environment	Classical	**UCR** (ours)	Angle-R_Y	Amplitude
CartPole	500.0 ± 0.0	500.0 ± 0.0	500.0 ± 0.0	92.7 ± 22.3
Acrobot	-74.5 ± 1.4	-118.7 ± 64.3	-81.7 ± 5.9	-457.0 ± 36.0
MountainCarC	98.1 ± 0.3	97.4 ± 0.8	97.1 ± 0.3	-0.0 ± 0.0
LunarLanderC	276.8 ± 6.3	277.8 ± 5.5	232.5 ± 49.0	65.7 ± 56.0
BipedalWalker	317.0 ± 2.2	282.5 ± 9.8	238.2 ± 17.5	228.6 ± 20.0

95% CI by 10000-resample percentile bootstrap;**bold**: best among quantum encodings (Classical shown for reference only, not a competitor).

Table 7. Main comparison on 5 standard RL benchmarks (obs_dim $\in [2, 24]$, 10 seeds each). **Final Eval Return**; mean with 95% bootstrap CI.

Environment	Classical	**UCR** (ours)	Angle-R_Y	Amplitude
CartPole	494.7 ± 7.9	480.7 ± 22.4	487.9 ± 14.7	37.2 ± 9.7
Acrobot	-82.4 ± 2.9	-126.5 ± 64.2	-88.7 ± 4.9	-487.1 ± 15.3
MountainCarC	94.3 ± 0.2	94.6 ± 0.4	94.5 ± 0.6	-1.5 ± 0.6
LunarLanderC	270.2 ± 7.3	267.5 ± 5.0	204.2 ± 63.4	-4.9 ± 86.7
BipedalWalker	299.8 ± 11.5	264.9 ± 7.1	220.4 ± 18.9	143.1 ± 16.0

95% CI by 10000-resample percentile bootstrap; **bold**: best among quantum encodings (Classical shown for reference only, not a competitor).

Table 8. Main comparison on 5 standard RL benchmarks (obs_dim $\in [2, 24]$, 10 seeds each). **Eval AUC** (sample efficiency, step-normalized trapezoidal integral of eval-return curve); mean with 95% bootstrap CI.

Environment	Classical	**UCR** (ours)	Angle-R_Y	Amplitude
CartPole	454.2 ± 18.3	430.8 ± 19.9	464.2 ± 14.7	39.9 ± 7.3
Acrobot	-107.5 ± 9.4	-220.6 ± 53.3	-194.2 ± 24.0	-489.0 ± 11.2
MountainCarC	94.5 ± 0.2	85.8 ± 2.1	79.9 ± 1.8	-1.1 ± 0.5
LunarLanderC	251.9 ± 5.9	201.4 ± 14.2	147.9 ± 49.2	-36.6 ± 31.8
BipedalWalker	260.5 ± 8.8	171.5 ± 17.3	140.6 ± 19.6	125.4 ± 18.1

95% CI by 10000-resample percentile bootstrap; **bold**: best among quantum encodings (Classical shown for reference only, not a competitor).

the most stable overall performance among the quantum encodings; however, this conclusion must be interpreted on a task-by-task basis.

In CartPole-v1, Classical, UCR, and Angle-R_Y all quickly reach near-perfect performance. Since the task itself is relatively simple, it is difficult to further distinguish their encoding capabilities. In Acrobot-v1, Angle-R_Y outperforms UCR, indicating that simple angle encoding remains a strong baseline in low-dimensional discrete, underactuated systems. Therefore, this paper does not consider low-dimensional discrete tasks as primary evidence of UCR's advantages.

It is continuous control tasks that truly drive the performance gap. In MountainCarContinuous-v0, UCR ranks in the top tier among quantum encodings for Best, Final, and AUC, while Amplitude largely fails to develop an effective policy. In LunarLanderContinuous-v3, UCR achieves a Best Eval Return of 277.8 ± 5.5 and a Final Eval Return of 267.5 ± 5.0, significantly higher than Angle-R_Y and Amplitude, and close to the Classical reference results. BipedalWalker-v3 further amplifies this difference: Classical remains the strongest reference, but within the quantum encoding group, UCR leads in Best, Final, and AUC, and the differences relative to Angle-R_Y and Amplitude are statistically supported.

Additionally, to test the robustness of the main findings with respect to the learning rate, we increased the learning

Table 9. Stability of the main comparison across 10 seeds. We report the seed-level standard deviation and the fraction of seeds reaching the pre-specified task threshold, with Wilson 95% confidence intervals.

Environment	Method	Seed SD	$k/10$	Success rate (Wilson 95% CI)
CartPole	Classical	0.0	10/10	100% [72, 100]%
CartPole	UCR (ours)	0.0	10/10	100% [72, 100]%
CartPole	Angle-R_Y	0.0	10/10	100% [72, 100]%
CartPole	Amplitude	38.1	0/10	0% [0, 28]%
Acrobot	Classical	2.4	10/10	100% [72, 100]%
Acrobot	UCR (ours)	134.0	9/10	90% [60, 98]%
Acrobot	Angle-R_Y	10.5	9/10	90% [60, 98]%
Acrobot	Amplitude	62.5	0/10	0% [0, 28]%
MountainCarC	Classical	0.5	10/10	100% [72, 100]%
MountainCarC	UCR (ours)	1.4	10/10	100% [72, 100]%
MountainCarC	Angle-R_Y	0.6	10/10	100% [72, 100]%
MountainCarC	Amplitude	0.0	0/10	0% [0, 28]%
LunarLanderC	Classical	10.9	10/10	100% [72, 100]%
LunarLanderC	UCR (ours)	9.5	10/10	100% [72, 100]%
LunarLanderC	Angle-R_Y	97.4	9/10	90% [60, 98]%
LunarLanderC	Amplitude	94.1	1/10	10% [2, 40]%
BipedalWalker	Classical	3.8	10/10	100% [72, 100]%
BipedalWalker	UCR (ours)	16.9	1/10	10% [2, 40]%
BipedalWalker	Angle-R_Y	29.6	0/10	0% [0, 28]%
BipedalWalker	Amplitude	34.1	0/10	0% [0, 28]%

Seed SD is the sample standard deviation across 10 independent seeds (ddof $= 1$).Success is computed from each seed's best evaluation return using pre-specified task thresholds. Wilson 95% CIs are reported for binomial success rates.

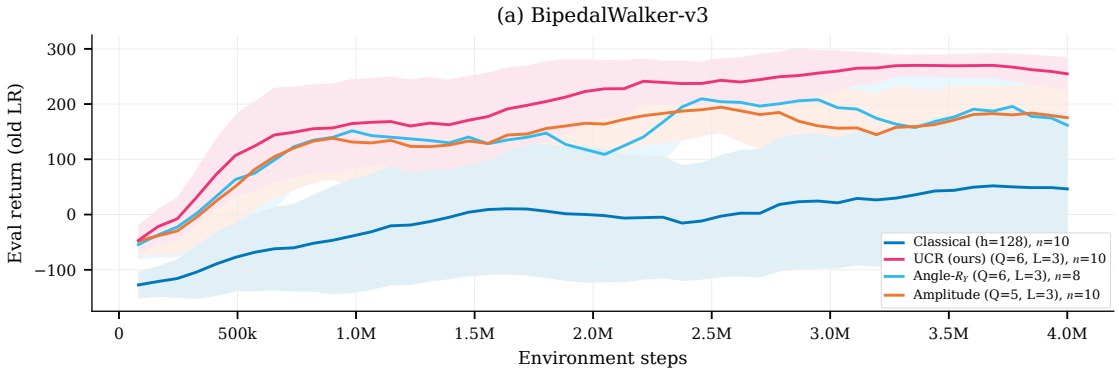


Fig. 7. Learning curves of BipedalWalker-v3 under an aggressive learning rate. The network architecture for all methods remains consistent with the main experiment, and the learning rate is increased by an order of magnitude compared to the main setup.

rate by an order of magnitude while keeping all other settings and the network architecture unchanged. The results in Fig. 7 indicate that the UCR encoding exhibits the strongest robustness among the quantum encodings, rapidly entering the high-return region and continuing to rise steadily. In contrast, Angle-R_Y and Amplitude exhibit lower plateaus and more pronounced fluctuations in the later stages, while the classical network is clearly under-trained under this capacity-learning rate combination, exhibiting extremely high variance.

These results demonstrate that UCR's advantage is not accidental and that it offers more stable performance compared to other encodings, particularly in terms of sustained gains across continuous actions, higher-dimensional states, and

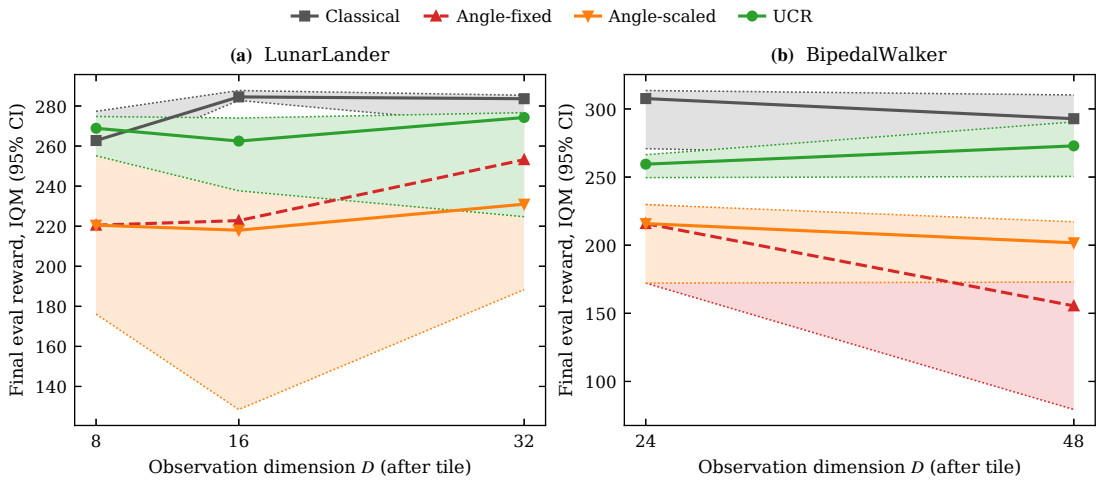


Fig. 8. Final-return IQM under dimensional scaling on (a) LunarLanderContinuous-v3 and (b) BipedalWalker-v3. Shading indicates the 95% bootstrap confidence interval. UCR maintains approximately dimension-invariant performance across two continuous-control tasks, while Angle-fixed shows clear degradation at the high-dimensional end of BipedalWalker-v3.

longer training processes. More precisely, UCR does not prove that quantum PPO has surpassed classical PPO; rather, it demonstrates that under a unified PPO training protocol, fixed-topology UCR provides superior front-end expressive power compared to commonly used Angle-R_Y and Amplitude encoders, thereby aiding the parameter update training of the downstream variational quantum layer.

4.4. IQM Main Analysis Under Dimensional Scaling

Experimental Focus: In this section, we increase only the observation dimension while keeping the dynamical task environment fixed, to investigate whether a fixed-topology UCR can mitigate the degradation of high-dimensional quantum state encoding.

We compare four observation encodings—Classical, Angle-fixed, Angle-scaled, and UCR encoding—on LunarLanderContinuous-v3 ($D \in \{8, 16, 32\}$) and BipedalWalker-v3 ($D \in \{24, 48\}$). Angle-fixed maintains a fixed number of qubits, consistent with the non-dimension-expanded baseline, to expose the information bottleneck when a classical MLP compresses a D-dimensional state into Q-dimensional quantum angles. Angle-scaled adjusts the number of qubits to follow the automatic configuration of UCR. Each group uses 5 seeds, with the primary metric being the interquartile mean (IQM) of returns over the final 100 evaluation episodes.

Fig. 8 shows that UCR encoding is the only quantum encoding that is approximately dimension-invariant across both tasks; it uses a fixed SFWHT structure to map the input to controlled rotation angles. Although UCR's fixed topology reduces task adaptability to some extent, it lowers the risk of front-end learning failure under high-dimensional inputs. On LunarLanderContinuous-v3, IQM ranges from 263 to 274, while on BipedalWalker-v3, it increases from 259 ($D = 24$) to 273 ($D = 48$). The performance of Angle-fixed on BipedalWalker-v3 degrades significantly as D increases, dropping from 216 to 156. Angle-scaled shows a smaller decline, dropping from 216 to 202 on BipedalWalker-v3. Classical consistently maintains the highest absolute reward: its IQM reaches 285 ($D = 16$) on LunarLanderContinuous-v3 and 308 ($D = 24$) on BipedalWalker-v3. It should be noted that, due to the constant column-norm property of its Jacobian, our UCR encoding is expected to exhibit good conditioning robustness when scaled to high dimensions. In contrast, Angle-fixed enters the saturation region of tanh more frequently in high dimensions, causing the front-end gradients to vanish,

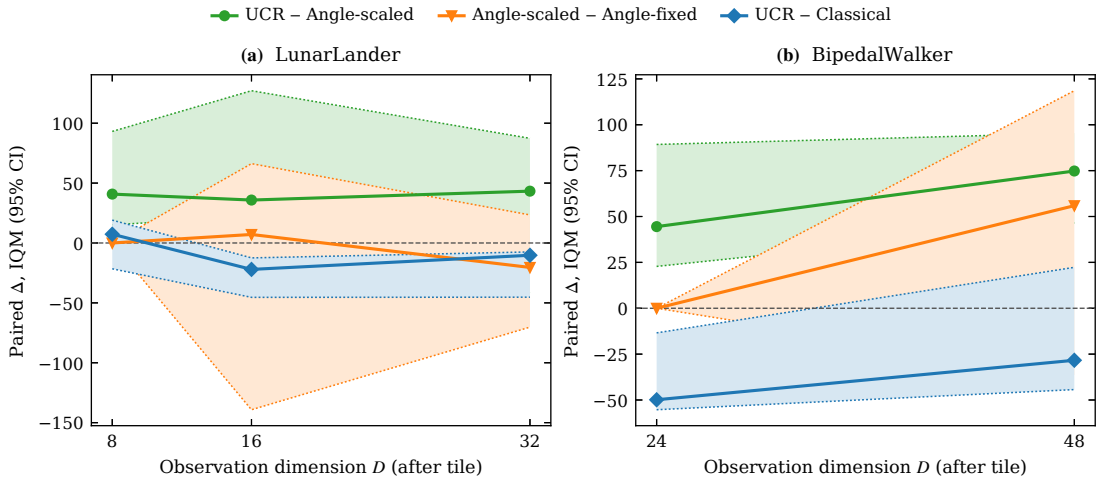


Fig. 9. IQM differences from same-seed paired bootstrap on (a) LunarLanderContinuous-v3 and (b) BipedalWalker-v3. UCR's advantage over Angle-scaled is significant on both dimensions of BipedalWalker-v3 and on $D = 8, 16$ of LunarLanderContinuous-v3.

which leads to collapse on the high-dimensional BipedalWalker-v3 task. This result demonstrates that the value of UCR lies not in surpassing full classical PPO, but in enhancing the stability of the quantum front-end under dimensional scaling.

Comparisons between methods are based on same-seed paired differences in IQM and their bootstrap confidence intervals. This approach was chosen because dimensionality experiments require determining whether performance differences occur consistently across input dimensions D, rather than merely comparing the heights of two mean curves. Same-seed pairing helps offset some of the variability caused by shared random conditions, thereby allowing for a more direct assessment of whether UCR's advantage over Angle-scaled remains consistent across different dimensions.

Fig. 9 further illustrates the paired differences. UCR–Angle-scaled is significantly positive for all D values in BipedalWalker-v3 (+44.5, +74.8); it is significant for D=8/16 in LunarLanderContinuous-v3 (+40.8, +35.8), but no longer significant for D=32 (CI=[-42, $+87$]). For Angle-scaled–Angle-fixed, the CI spans 0 for all D values, failing to reach significance. This indicates that increasing the qubit budget alone is insufficient to consistently improve Angle encoding; at least with the statistical power of 5 seeds, no strong conclusions can be drawn. In contrast, the difference between UCR encoding and Angle-scaled is more consistent, suggesting that the performance difference resulting from fixed topology encoding is the primary factor. Furthermore, UCR performs worse than Classical on LunarLanderContinuous-v3 and BipedalWalker-v3; this section does not support the claim that "the quantum front-end outperforms classical MLPs." It does support the conclusion that, under the same quantum resource accounting, UCR is more resistant to performance degradation caused by the expansion of the observation dimension than Angle-class encodings.

Therefore, the value of this section within the full paper is clear: it provides a direct test of whether "fixed-topology UCR can mitigate the degradation of high-dimensional quantum state encoding." The results show that UCR maintains a more stable IQM under dimensional scaling and significantly outperforms Angle-scaled under a fair quantum budget.

4.5. UCR Structural Ablation

Ablation experiments (Table 10) examine the impact of adding a Hadamard gate on the encoding-qubit circuit, the angular domain upper bound, and trainable input scaling on UCR. The results indicate that the effectiveness of UCR does not depend on any single enhancement.

In Acrobot-v1, UCR still achieves optimal or near-optimal performance even after removing the Hadamard gate from

Table 10. UCR ablation on Acrobot-v1 and LunarLanderContinuous-v3 (10 seeds each). Factors: encoding-layer Hadamard (H/noH) × fixed angle range (π / $\pi/2$), plus one trainable-input-scale variant (`noH_pi_in`).

Environment	UCR (H, π)	UCR (H, $\pi/2$)	UCR (noH, π)	UCR (noH, $\pi/2$)	UCR (noH, π, inp.scl.)
Acrobot	-118.7 ± 95.9	-80.6 ± 8.3	-76.4 ± 1.3	-77.3 ± 2.4	-76.2 ± 1.5
LunarLanderC	277.8 ± 6.8	247.8 ± 39.0	240.8 ± 54.7	220.1 ± 70.5	244.3 ± 55.0

95% CI half-width $= t_{0.975,n-1} \cdot$ SE.

the encoding-qubit circuit. This suggests that the underlying architecture of UCR is already trainable, and Hadamard is not a necessary condition for the model to "go from unable to learn to able to learn." In LunarLanderContinuous-v3, the configuration with Hadamard and an angular domain of π performs better, indicating that in medium-dimensional continuous control, Hadamard initialization may improve feature dispersion and quantum expressivity, thereby raising the performance ceiling.

Contracting the angular domain to $\pi/2$ does not yield consistent benefits. While it may mitigate early optimization fluctuations in certain low-dimensional tasks, it may also compress the effective expression range in continuous control. Input scaling similarly acts as an enhancement rather than a core condition for UCR to hold.

Therefore, the ablation results support a more robust conclusion: the foundational architecture of UCR is already a trainable front-end encoder; Hadamard, angular domain bounds, and input scaling are enhancements tailored to different task scenarios, rather than determinants of the method's validity.

Table 11. Details of circuit and parameter counts under different environments and encoding schemes. The number of parameters for the encoding front-end gates (n_H, n_{rot}, n_{CNOT}) does not include VQC; n_{policy} denotes the Actor parameters, and n_{total} denotes the total number of Actor+Critic parameters; the number of UCR gates and parameters is uniquely determined by the fixed topology (Proposition 1).

Environment	Method	Q	n_H	n_{rot}	n_{CNOT}	n_{front}	n_{VQC}	n_{pre}	n_{post}	n_{policy}	n_{total}
CartPole	Classical	—	—	—	—	—	—	0	0	4,610	9,155
	UCR (ours)	3	2	4	4	10	21	0	26	48	4,593
	Angle-R_Y	3	0	3	0	3	21	67	26	115	4,660
	Amplitude	2	0	4	3	7	14	0	22	37	4,582
Acrobot	Classical	—	—	—	—	—	—	0	0	4,803	9,476
	UCR (ours)	4	3	8	8	19	28	0	67	96	4,769
	Angle-R_Y	4	0	4	0	4	28	136	67	232	4,905
	Amplitude	3	0	8	7	15	21	0	59	81	4,754
MountainCarC	Classical	—	—	—	—	—	—	0	0	4,418	8,835
	UCR (ours)	2	1	2	2	5	14	0	9	25	4,442
	Angle-R_Y	2	0	2	0	2	14	6	9	31	4,448
	Amplitude	1	0	2	3	5	7	0	7	16	4,433
LunarLanderC	Classical	—	—	—	—	—	—	0	0	4,868	9,669
	UCR (ours)	4	3	8	8	19	28	0	58	89	4,890
	Angle-R_Y	4	0	4	0	4	28	212	58	301	5,102
	Amplitude	3	0	8	7	15	21	0	50	74	4,875
BipedalWalker	Classical	—	—	—	—	—	—	0	0	20,232	40,073
	UCR (ours)	6	5	32	32	69	60	0	164	229	20,070
	Angle-R_Y	6	0	6	0	6	60	774	164	1,003	20,844
	Amplitude	5	0	32	31	63	50	0	156	211	20,052
Ant	Classical	—	—	—	—	—	—	0	0	94,992	188,177
	UCR (ours)	8	7	128	128	263	104	0	552	665	93,850
Humanoid	Classical	—	—	—	—	—	—	0	0	159,522	314,915
	UCR (ours)	10	9	512	512	1,033	220	0	2,357	2,595	157,988

4.6. Joint Analysis of Resources and Performance

Resource statistics (Table 11) show that UCR is not the method with the fewest gates. Compared to Angle-R_Y, UCR introduces more rotation gates and CNOT gates. For example, in BipedalWalker-v3, the Angle-R_Y encoding frontend contains only 6 rotation gates and no CNOT gates, whereas UCR contains 32 rotation gates and 32 CNOT gates. Angle-R_Y delegates part of its representational capacity to a classical preprocessing network, resulting in a lighter quantum frontend. UCR, on the other hand, explicitly encodes structural information into a fixed quantum topology, avoiding the need for additional classical preprocessing. Amplitude uses fewer qubits or fewer gates in some tasks. For example, its encoding frontend in BipedalWalker-v3 is smaller than that of UCR. However, lower resource usage does not necessarily translate to better performance. Combined with the main experimental results, it is evident that Amplitude degrades more easily on tasks such as MountainCarContinuous-v0, LunarLanderContinuous-v3, and Acrobot-v1. This indicates that comparing only the number of qubits or gates is insufficient to evaluate the relative merits of methods.

Therefore, the combined resource–performance results do not support the simplistic claim that "UCR is the most resource-saving method." A more accurate conclusion is that UCR utilizes resources more efficiently and is more interpretable. It sacrifices some gate-level simplicity in exchange for a fixed topology, a gate structure independent of samples, the ability to calculate resource requirements prior to training, and more robust performance gains in continuous control tasks. Under NISQ constraints, this trade-off is more important than simply pursuing the fewest qubits or the fewest gates.

4.7. VQC Parameter-Gradient Diagnostics

The Jacobian analysis presented above only explains how the input changes the quantum state after UCR encoding; it does not address whether the gradient of the loss function with respect to the trainable parameters gradually approaches zero during VQC training. The concern to be examined is that UCR input states may exhibit strong entanglement, causing the gradients of the VQC parameters to decrease rapidly as the number of qubits increases, thereby increasing the risk of a barren plateau.[36] To address this, we repeatedly initialize VQCs with the same circuit architecture using different random parameter initializations for the same batch of input observations, and directly examined whether the loss gradients for each parameter retain appreciable variation under different initializations; intuitively, if these gradients all approach zero and exhibit almost no variation as the system grows, the optimizer will struggle to find effective update directions.

This section does not use task return as the primary metric, but instead distinguishes between three levels: input state structure, gradient concentration, and numerical vanishing. $V_{\text{med}}^{\text{common}}$ is the primary gradient statistic in the main text, used to measure whether VQC parameter gradients become increasingly concentrated across different random initializations; $S_{\text{bal}}/S_{\text{ref}}$ and K_{med} describe, respectively, the entanglement of input states and the similarity among different input states; G_{RMS} and $\rho_{<10^{-6}}$ further examine the overall gradient magnitude and the proportion of gradients approaching numerically zero (see Appendix A2 for details). By combining these metrics, we can avoid equating a decrease in gradient variance directly with the disappearance of all gradients, and we can also avoid interpreting low entanglement directly as a better representation of the input.

For the mth input, the rth random initialization, and the jth VQC parameter, we define

$$g_{rmj} = \frac{\partial C_m(\Theta_r)}{\partial \theta_j}, \tag{44}$$

and

$$V_{mj} = \text{Var}_r\big[g_{rmj}\big], \tag{45}$$

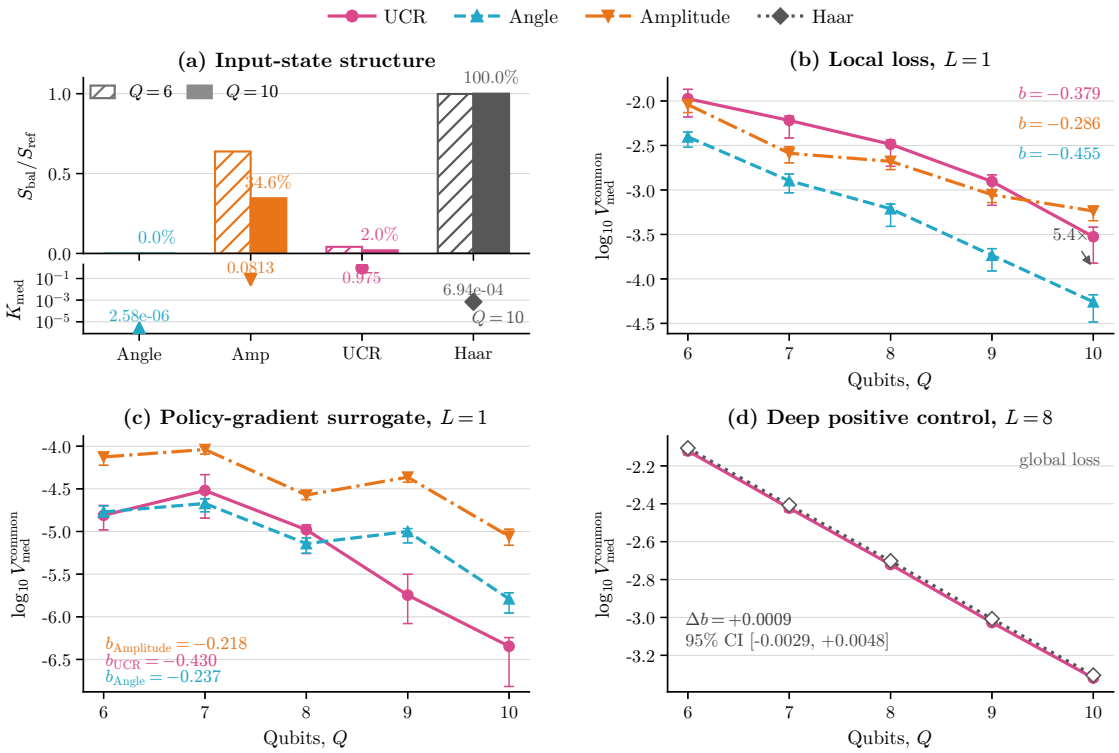


Fig. 10. VQC parameter-gradient diagnostics. The primary statistic is $V_{\mathrm{med}}^{\mathrm{common}}$ (Eq. (46)). (a) Input-state structure. Upper bars show the balanced-bipartition entropy normalised by the Haar/Page reference, $S_{\mathrm{bal}}/S_{\mathrm{ref}}$; hatched and solid bars denote $Q = 6$ and $Q = 10$. Lower markers show the median pairwise fidelity K_{med} at $Q = 10$. The two metrics indicate input-state entanglement and similarity, respectively. (b) Gradient variance versus Q for the local loss at $L = 1$. (c) The corresponding policy-gradient-surrogate result, following a path closer to PPO optimisation. (d) Positive control with a global loss at $L = 8$, comparing UCR with a Haar-random-input reference in which the 32 observations are replaced by Haar-random input states. Slopes are finite-range fits of $\log_{10} V_{\mathrm{med}}^{\mathrm{common}} = a + bQ$ (Eq. (51)), not asymptotic exponents. Error bars in (b)–(d) are 95% bootstrap confidence intervals.

where C_m denotes the diagnostic loss corresponding to that input, and V_{mj} indicates how much variation in the gradient of that parameter is retained across different random initializations. The primary statistic reported in the main text is

$$V_{\mathrm{med}}^{\mathrm{common}} = \underset{m,\, j\in\mathcal{J}_{\mathrm{common}}}{\mathrm{median}} \ \mathrm{Var}_r\left[g_{rmj}\right], \tag{46}$$

where $\mathcal{J}_{\mathrm{common}}$ is the common set of non-structurally-zero parameters retained across the three encoding schemes. Certain parameters have gradients that are always zero for all inputs and initializations due to circuit structure, commutation relations, or loss-induced causal cone constraints. Since these parameters do not affect the current loss, they should not be regarded as evidence of a barren plateau and are therefore uniformly excluded across all three encodings according to the same rules; all small gradients actually caused by the input state are retained.

The experiments compare UCR, Angle, and Amplitude encoding under the same qubit count, VQC architecture, and parameter initialization protocol. For each experimental condition, 32 fixed BipedalWalker-v3 observations, 200 independent VQC random initializations, $Q = 6$–10, and $L = 1$–4. Diagnostics were performed sequentially using local loss, policy gradient surrogate loss, and a global-loss positive control with $L = 8$,[37] corresponding to Fig. 10(b)–(d).

Input State Structure. To examine whether UCR produces highly entangled input states, we calculate the balanced-bipartition entanglement entropy

$$S_{\mathrm{bal}} = -\,\mathrm{Tr}\big(\rho_A \log_2 \rho_A\big), \qquad \rho_A = \mathrm{Tr}_B\big(|\Psi\rangle\langle\Psi|\big). \tag{47}$$

The subsystem dimensions of the balanced bipartition are

$$d_A = 2^{\lfloor Q/2 \rfloor}, \qquad d_B = 2^{\lceil Q/2 \rceil}. \tag{48}$$

The corresponding Haar-random pure-state Page reference entropy is [38]

$$S_{\text{ref}}(Q) = \log_2 d_A - \frac{d_A}{2 d_B \ln 2}. \tag{49}$$

Therefore, the closer $S_{\text{bal}}/S_{\text{ref}}$ is to 100%, the closer the input state is to the high entanglement levels of a typical random state. Fig. 10(a) shows that, for $Q = 10$, the balanced-bipartition entanglement of the UCR in Eq. (47) is only 2.0% of the corresponding reference value in Eq. (49), which is substantially lower than the 34.6% for Amplitude encoding and the 100% for the Haar-random state. Consequently, under the balanced-bipartition entropy metric adopted in this paper, the present experiments do not support the notion that UCR input states exhibit entanglement levels close to those of typical Haar-random states in the ordinary subsystem-bipartition sense.

To further characterize the similarity of quantum states corresponding to different observations, we define the pairwise fidelity

$$K(\boldsymbol{x}, \boldsymbol{x}') = \left| \langle \Psi(\boldsymbol{x}) | \Psi(\boldsymbol{x}') \rangle \right|^2, \tag{50}$$

as well as its median K_{med}. UCR achieves $K_{\text{med}} = 0.975$ when $Q = 10$, indicating that the different input states are relatively close to one another. Therefore, low entanglement and the distinguishability of input states are two distinct issues; this paper does not interpret low entanglement directly as evidence of a better input representation. The differences between the two metrics also indicate that UCR encoding and Amplitude encoding map classical data into structurally distinct regions of Hilbert space.

Local Loss Results. For $L = 1$, the gradient variance for all three encodings decreases as Q increases (Fig. 10(b)). To describe the rate of decrease within a finite test range, we define

$$\log_{10} V_{\text{med}}^{\text{common}} = a + bQ, \tag{51}$$

where b describes the variation only within the interval $Q = 6$–10 and does not represent an asymptotic exponential scaling. The slopes for UCR, Amplitude, and Angle are -0.379, -0.286, and -0.455, respectively. Across all tested Q values, the gradient variance of UCR is higher than that of Angle; at $Q = 10$, it is approximately 5.4 times that of Angle. Therefore, under shallow local loss, no more severe gradient compression was observed for UCR relative to Angle.

Results for Policy-gradient surrogate loss. We retain the main backpropagation paths for policy readout and policy gradients without re-executing a full PPO training run; under this setting, the update signal is closer to that received by the VQC during actual policy optimization than that from the local loss of a single qubit. In Fig. 10(c), the slopes within the finite range for UCR, Angle, and Amplitude are -0.430, -0.237, and -0.218, respectively. This indicates that the variance of the policy gradient for UCR decreases more rapidly as the number of qubits increases, and also suggests that the encoding effect is influenced by both the policy readout and the loss structure; the gradient behavior observed in the local loss cannot be directly extrapolated to the full policy optimization process. Supplementary diagnostics in Appendix A2 further indicate that this decline does not correspond to widespread numerical gradient vanishing; the differences in gradients between encodings primarily occur in shallow circuits and diminish as the VQC deepens.

Deep Positive Control. In the global loss positive control at $L = 8$ (Fig. 10(d)), the gradient variance curves for UCR and Haar random inputs almost coincide, with a slope difference of

$$\Delta b = 0.0009, \qquad 95\%\ \mathrm{CI} = [-0.0029,\ 0.0048]. \tag{52}$$

On the one hand, this result indicates that the current diagnostic workflow can identify typical gradient concentration under deep circuits and global loss functions; on the other hand, it demonstrates that when the circuit is sufficiently deep and the loss is global, circuit depth and loss structure become the primary factors, and UCR encoding cannot prevent the occurrence of a barren plateau.

Overall, within the limited range of system sizes tested in this paper, no additional severe gradient concentration associated with the ordinary entanglement of UCR input states was observed. UCR preserves gradient signals that are stronger than those of Angle under shallow local losses, but its policy-gradient variance still decreases markedly as the number of qubits increases, and gradient concentration also occurs under deep global losses. Therefore, this paper does not claim that UCR eliminates or universally prevents barren plateaus.

The above results should still be interpreted within the finite ranges of qubit number, VQC depth, and loss structure tested in this paper, and cannot be interpreted as a general convergence guarantee for deep UCR-encoded circuits. Recent Lie-algebraic barren-plateau theory indicates that, under the corresponding deep/mixing and algebraic assumptions, even if the dimension of the dynamical Lie algebra (DLA) scales only polynomially, generalized structures such as the $\mathfrak{g}$-purity of the input state or measurement observable relative to the DLA may still lead to loss or gradient concentration.[39] Consequently, the aforementioned balanced-bipartition entropy cannot serve as a substitute for this DLA-relative characterization. This paper has not calculated the DLA scaling of the actual VQC, nor has it computed the $\mathfrak{g}$-purity of UCR-encoded states; therefore, the possibility that UCR input-state structure may adversely affect optimization and convergence in deeper quantum circuits cannot be ruled out.

4.8. Gate-Level Noise

We now evaluate the final-checkpoint policies using the gate-level noise protocol defined in Section 4.1. We report absolute noisy return R_{noisy}, the paired UCR–Angle difference $\Delta R_{\mathrm{U-A}}$ in Eq. (42), and normalized within-encoder retention S_{norm} in Eq. (43).

Fig. 11 first addresses the direct question arising from this: to what extent can the UCR encoding retain its original performance advantages in complex continuous control tasks after introducing explicit gate-level noise. For the six noise intensity levels specified by Eq. (40) and Eq. (41), on the BipedalWalker-v3, the mean paired same-seed return differences between UCR and Angle-R_Y are approximately $+39$, $+42$, and $+46$ at $p_1 = 0$, 10^{-4}, and 3×10^{-4}, respectively, and the corresponding pointwise 95% confidence intervals do not cross zero; when $p_1 = 10^{-3}$, the difference reverses to approximately -96, and the interval again does not cross zero (Fig. 11(c)). LunarLanderContinuous-v3 also demonstrates a low-noise advantage, but uncertainty increases markedly starting at $p_1 = 10^{-3}$; by $p_1 = 10^{-2}$, the UCR–Angle difference is approximately -252. Moreover, the drop in mean return near the critical noise level does not reflect a uniform deterioration across all seeds. For BipedalWalker-v3 at $p_1 = 10^{-3}$, 8/10 UCR seeds remain above the random-policy return, but a few seeds have already clearly failed; the results for Angle-R_Y are more concentrated across seeds and all exceed the random baseline (Fig. 11(d)). The complete scan for the five tasks is shown in Appendix A3.

Therefore, the performance advantage of UCR extends into the low-noise region, but once the task-dependent noise boundary is exceeded, it is overtaken by the shallower Angle-R_Y circuit. At the same time, UCR exhibits significant seed-

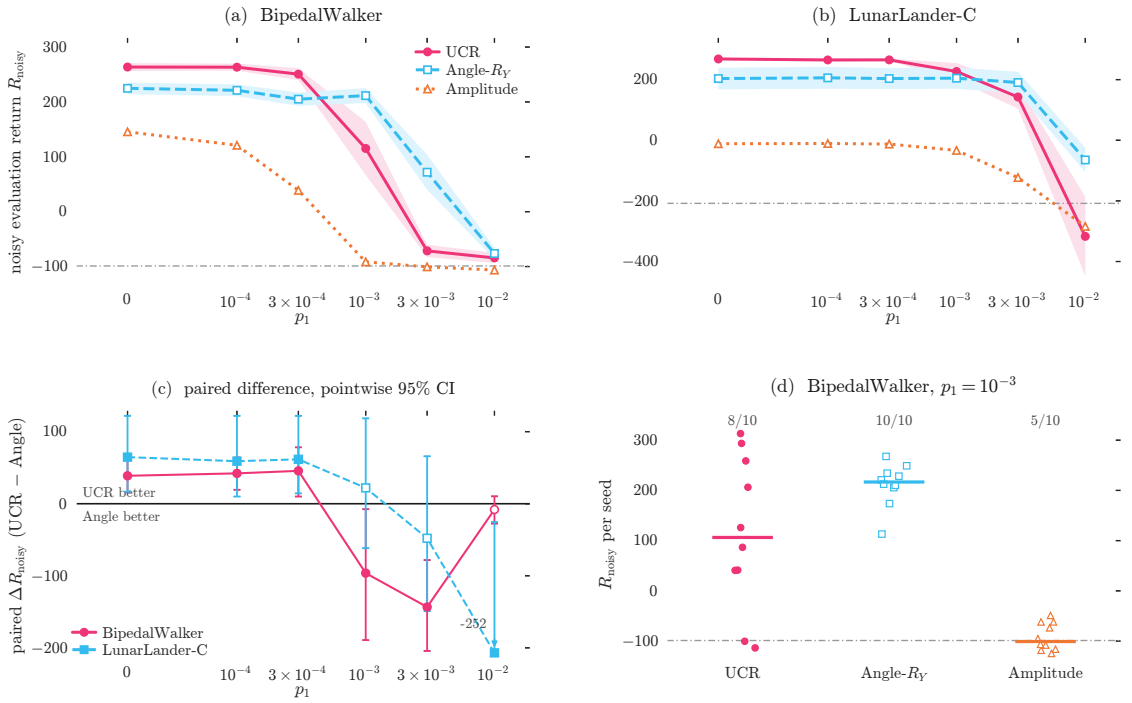


Fig. 11. Low-noise performance window of uniformly controlled rotation (UCR) encoding under gate-level depolarizing noise (Eq. (40) and Eq. (41)). Final-checkpoint policies; 10 independent training seeds per cell. (a) and (b) Mean noisy evaluation return R_{noisy} on BipedalWalker-v3 and LunarLanderContinuous-v3; shaded bands denote seed-level standard errors (SEs) and dash-dotted lines denote random-policy returns. (c) Paired per-seed difference $\Delta R_{\text{U}-\text{A}}$ (Eq. (42)); positive values favour UCR and negative values favour Angle-R_Y. Error bars are pointwise 95% percentile-bootstrap confidence intervals (CIs; $B = 20000$), and filled markers indicate that the interval excludes zero. (d) Seed-level noisy returns on BipedalWalker-v3 at $p_1 = 10^{-3}$; horizontal bars denote medians, and n/N gives the number of seeds above the random-policy return.

level heterogeneity and partial seed-wise collapse near the critical noise level, rather than a uniformly smooth degradation across all seeds.

In the dimensional scaling noise experiments, D denotes the observation dimension fed into the encoding interface, and $N_{2\text{Q}}$ denotes the total number of two-qubit gates in the UCR front end and VQC combined. For the ring entangling structure used in this paper,

$$N_{2\text{Q}} = 2^{\lceil \log_2 D \rceil} + L\left(\lceil \log_2 D \rceil + 1\right), \tag{53}$$

where L is the number of VQC entanglement layers.

Fig. 12(a) shows that as the input dimension and the number of two-qubit gates increase, the performance crossover point between UCR and Angle-R_Y shifts toward lower noise levels. Under the $p_1 = 10^{-3}$ condition for LunarLanderContinuous-v3, the UCR–Angle difference changes from $+37$ for D8 to -112 for D32; under the $p_1 = 3 \times 10^{-4}$ condition for BipedalWalker-v3, the difference changes from $+43$ for D24 to -74 for D48. It should be noted that for the two configurations of the BipedalWalker-v3 at $p_1 = 10^{-2}$, the paired differences actually revert to -2 and $+4$; at this point, both encodings have degraded, and the paired differences no longer provide discriminative information, so the crossover should be read in the middle noise region. The total number of two-qubit gates for the five configurations is given by Eq. (53), increasing from 16 to 85. Fig. 12(b) plots the normalized retention S_{norm} (Eq. (43)) on the vertical axis, with configurations within the same environment sorted by $N_{2\text{Q}}$. At the two intensity levels of $p_1 = 10^{-3}$ and 3×10^{-3}, S_{norm} decreases monotonically as $N_{2\text{Q}}$ increases; by the time $N_{2\text{Q}} = 85$ (BW D48), the retention rates for both intensity levels have dropped to approximately 0, that is, they return to the level of a random strategy. These results indicate that longer

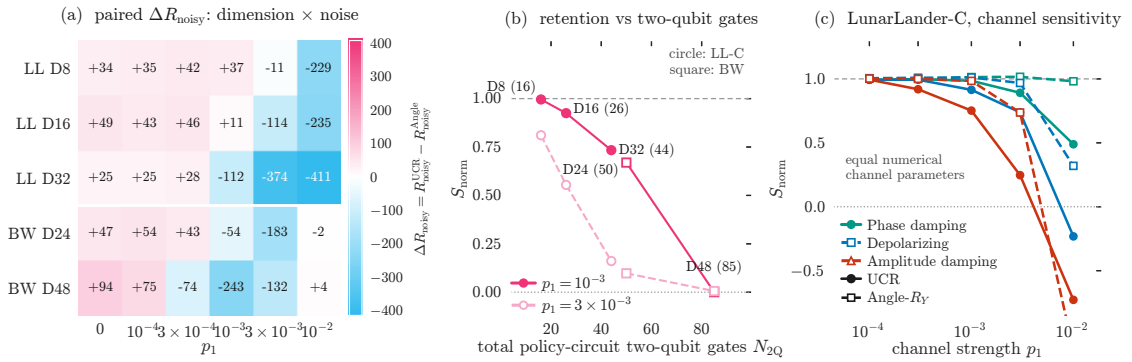


Fig. 12. Dimension and circuit-size dependence of the UCR noise boundary. Protocol and final-checkpoint anchor as in Fig. 11. (a) Paired difference $\Delta R_{\text{U}-\text{A}}$ across five evaluated observation dimensions D. LL and BW denote LunarLanderContinuous-v3 and BipedalWalker-v3; for example, LL D16 is LunarLanderContinuous-v3 at $D = 16$. Each cell reports the seed-mean paired difference. (b) UCR normalized retention S_{norm} versus the total policy-circuit two-qubit gate count N_{2Q} (Eq. (53)). Values in parentheses give N_{2Q}; circles denote LL and squares BW, and lines connect configurations within the same environment only. (c) Channel-sensitivity curves on LunarLanderContinuous-v3 for phase damping, depolarizing, and amplitude damping; solid lines with filled circles denote UCR and dashed lines with open squares Angle-R_Y. Equal numerical channel parameters do not imply equal physical channel severity. Panels (a) and (b) use five training seeds per cell, and panel (c) ten.

explicit gate sequences gradually offset the representational benefits of UCR.

The three types of local noise channels produce different response curves (Fig. 12(c)). The three curves nearly coincide at $p_1 = 10^{-4}$, with differences becoming apparent as noise intensity increases: Combined with Fig. 16 in Appendix A4, this indicates that, at the same numerical noise intensity, the severity of UCR degradation generally follows the order: Amplitude damping $>$ Depolarizing $>$ Phase damping. UCR encoding exhibits relatively stable performance within a certain noise range, but rapid performance degradation occurs once a threshold related to the task and noise channel is exceeded. Based on this, this paper only compares the encoding's sensitivity to different channels and does not rank the severity of harm to real channels.

4.9. Discussion

Progressive effects of each experiment. Section 4.2 verifies numerical- and policy-level equivalence of both UCR paths, excluding inconsistencies between training and deployment resource accounting. Section 4.3 benchmarks five Gymnasium tasks spanning discrete/continuous actions and low- to medium-high-dimensional observations, showing the most stable quantum-encoding performance. Section 4.4 tests whether fixed-topology UCR mitigates encoding degradation as observation dimension increases under fixed dynamics, while Section 4.5 confirms that effectiveness does not depend on encoding-qubit Hadamard, angular range, or trainable input scaling. Section 4.6 shows efficient, interpretable resource use without claiming minimum cost. Section 4.7 evaluates VQC-gradient variance, additional barren-plateau risk, and scope; Section 4.8 identifies a low-noise advantage window narrowing with input dimension, depth, and two-qubit-gate exposure. Ant-v5 and Humanoid-v5 remain in the appendix as exploratory feasibility studies because of insufficient seeds.

Effectiveness and Advantages of UCR Encoding. UCR outperforms both angle encoding and amplitude encoding in terms of Best, Final, and AUC metrics across three continuous control tasks: MountainCarContinuous-v0, LunarLanderContinuous-v3, and BipedalWalker-v3. On low-dimensional discrete tasks, it is only slightly outperformed by Angle-R_Y, which relies on classical neural networks. In dimensionality-scaling experiments, Angle-scaled collapses on high-dimensional tasks, demonstrating that UCR encoding exhibits good conditioning robustness under high-dimensional scaling due to the con-

stant column-norm property of its Jacobian. Furthermore, UCR encoding is the only encoding that remains approximately dimension-invariant on LunarLanderContinuous-v3 and BipedalWalker-v3, and the gap between it and Angle-scaled remains consistently positive, indicating that UCR effectively mitigates encoding degradation in high dimensions. The learning-rate robustness experiment also demonstrate that UCR's advantages are not coincidental.

Fixed Topology Resource Statistics For a given input dimension, the gate count, gate order, and qubit connections in UCR encoding are determined solely by (N, k). Changes in state samples affect only the continuous rotation parameters without altering the circuit topology. For reinforcement learning, the fixed topology keeps the encoding structure constant, so that state changes primarily manifest as variations in continuous parameters within the same circuit framework, thereby reducing structural non-stationarity during training. In terms of compilation, a unified logical topology enables gate decomposition, qubit placement, routing, and scheduling to be performed around a single circuit template. From the perspective of resource estimation, the number of qubits, logical gate count, circuit depth, and two-qubit-gate exposure can be determined by the input dimension prior to training, maintaining a consistent basis for resource accounting across different samples and training seeds—thereby transforming resource consumption from a "random variable" into a "function of dimension." In terms of NISQ deployment, the fixed topology makes circuit load, gate placement, and structural noise exposure more predictable, thereby reducing structural and deployment uncertainties caused by variations in input samples—though it does not provide immunity to noise. As the input dimension, circuit depth, and number of two-qubit gates increase, cumulative gate errors may still offset the representational gains of UCR. At the same time, the identical circuit structure provides reproducible, localizable, and comparable analysis conditions for gradient and noise issues, though it cannot fundamentally eliminate the barren plateau. Based on this fixed structure, this paper further employs a differentiable StatePrep path for classical simulation training, while an explicit gate-level path handles hardware interpretation; the two are equivalent within floating-point precision. Ablation results also indicate that UCR's basic encoding structure is trainable in and of itself, while encoding-qubit Hadamard, angular domain bounds, and trainable input scaling are task-specific enhancements. Thus, the primary value of a fixed topology lies not in directly yielding higher returns, but in enabling high-dimensional quantum state inputs to form a deployable parameterized quantum front end with a consistent training structure, clearly defined resource bounds, and the ability to be stably mapped to gate-level implementations.

Resource–Performance Trade-off. UCR is not the method with the fewest gates, yet it achieves competitive continuous control performance on medium-to-high-dimensional tasks with fewer parameters; Amplitude uses fewer resources but suffers from insufficient learnability and stability; angle encoding relies on a large number of classical neural network parameters, yet its training performance is constrained on slightly more complex tasks. Therefore, a single resource metric is insufficient to judge the quality of a method; joint resource–performance efficiency is the reasonable evaluation criterion. Under NISQ constraints, a trade-off characterized by "slightly higher gate-level cost, fewer parameters, and fixed topology" holds greater deployment value than simply pursuing the fewest qubits or the fewest gates.

Trainability and noise boundary. VQC parameter-gradient diagnostics indicate that, within the limited range of qubits and depth tested in this paper, no additional severe barren-plateau penalty attributable to UCR was observed. However, policy-gradient variance continues to decrease as the number of qubits increases, and deep circuits utilising global cost functions exhibit the expected gradient concentration; furthermore, a polynomial-dimensional dynamical Lie algebra (DLA) alone is insufficient to rule out concentration effects arising from the input-state structure or observable structure.

Consequently, UCR does not eliminate barren plateaus, and these results do not constitute a guarantee of deep-circuit convergence. Likewise, its higher ideal performance does not imply unconditional noise robustness. The gate-level experiments identify a low-noise region in which UCR retains an absolute advantage on complex continuous-control tasks, but this window narrows as input dimension, circuit depth, and two-qubit-gate exposure increase, after which the shallower Angle-R_Y circuit can become preferable.

Limitations and Future Work. First, it should be noted that Classical remains the stronger benchmark for most complex tasks; this paper does not claim that UCR has surpassed classical PPO. Second, the high-dimensional experiments in the appendix only demonstrate that UCR can be extended to Ant-v5 and Humanoid-v5, not that it is already competitive on high-dimensional tasks. The noise experiments are modeled as independent Markovian synthetic noise models and should be interpreted as controlled gate-level stress tests, rather than noise tests on real quantum devices. The barren plateau experiments are limited to finite qubit counts and circuit depths; this paper reports only trends in gradient changes within a limited range. Furthermore, this paper has not characterized the DLA scaling of the actual VQC, nor has it analyzed the DLA-relative $\mathfrak{g}$-purity of UCR-encoded input states or related observables; consequently, deep-circuit convergence in the polynomial-DLA scenario remains to be investigated in future work. Finally, this paper primarily compares three representative quantum encodings and has not yet covered data re-uploading, task-specific encodings, or learnable quantum feature maps. For future work, we plan to explore the following areas: first, introducing real backends or calibrated noise models into noise experiments; second, leveraging the fixed topology and resource-analyzable properties of the UCR encoding to explore its potential value on real quantum computers; third, by combining UCR with other encoding schemes and performing training optimization, to verify whether performance bottlenecks in tasks with higher-dimensional input states can be overcome; fourth, from a theoretical perspective, the UCR encoding studied in this paper may be applicable to machine learning scenarios involving classical-state inputs to quantum circuits in a general sense, and future work can explore applications in more quantum machine learning scenarios; fifth, the UCR encoding described in this paper is implemented using a single encoding qubit; its quantum circuit depth increases linearly with the dimension of the high-dimensional input state and is therefore limited by hardware implementation constraints. By appropriately increasing the number of encoding-qubit circuits—at the cost of additional circuit complexity—the performance bottleneck caused by circuit depth can be alleviated, enabling a further optimization of the resource–performance trade-off.

5. Conclusion

This work reconstructs uniformly controlled rotations (UCR) as a state encoding module for hybrid quantum–classical PPO. The gate topology of UCR is uniquely determined by the observation dimension; sample variations only alter the rotation angle without changing the gate connectivity. We propose dual implementations: a differentiable StatePrep path and an explicit gate-level path. The former is designed for classical simulation training, while the latter supports explicit gate-level resource and noise accounting. The two paths are numerically equivalent (difference $< 10^{-5}$) and show no difference at the policy level.

In continuous control tasks and learning-rate robustness experiments, UCR demonstrates greater stability in peak performance, final performance, and sample efficiency; in low-dimensional discrete tasks, it is second only to angle encoding, which relies on a classical neural network. Our scheme exhibits a prominent comprehensive advantage among quantum encoding methods. Dimensional scaling experiments validate that UCR effectively mitigates the degradation

of high-dimensional quantum state encoding. From the resource perspective, while UCR is not the method with the fewest gates, it offers higher parameter efficiency, a fixed topology, and clear resource interpretability. For medium-to-high-dimensional tasks, it achieves competitive policies with fewer parameters, avoids the low-resource, low-reliability degradation associated with amplitude encoding, and reduces reliance on classical neural networks. Within the range of qubit numbers tested and for shallow VQCs, no additional severe gradient compression attributable to UCR was observed. However, the existing evidence does not constitute a guarantee of deep-circuit convergence, and in deep circuits with global loss functions, UCR cannot be regarded as universally avoiding barren plateaus. Gate-level noise experiments further demonstrate that UCR exhibits a discernible low-noise performance window in complex continuous control tasks, but this window narrows as the input dimension and quantum circuit depth increase.

Overall, the contribution of this paper is not to claim that quantum reinforcement learning comprehensively outperforms classical methods, but rather to provide a clearer, more verifiable path: in hybrid quantum-classical PPO, fixed-topology, differentiable, and resource-interpretable UCR encoding can offer a more reliable performance-resource trade-off compared to commonly used angle and amplitude encodings. This provides a more robust experimental baseline for subsequent research on quantum policy learning for NISQ devices, resource-constrained quantum front-end design, and controlled gate-level evaluation.

Appendix A: Additional Experimental Results

A1. High-Dimensional Feasibility

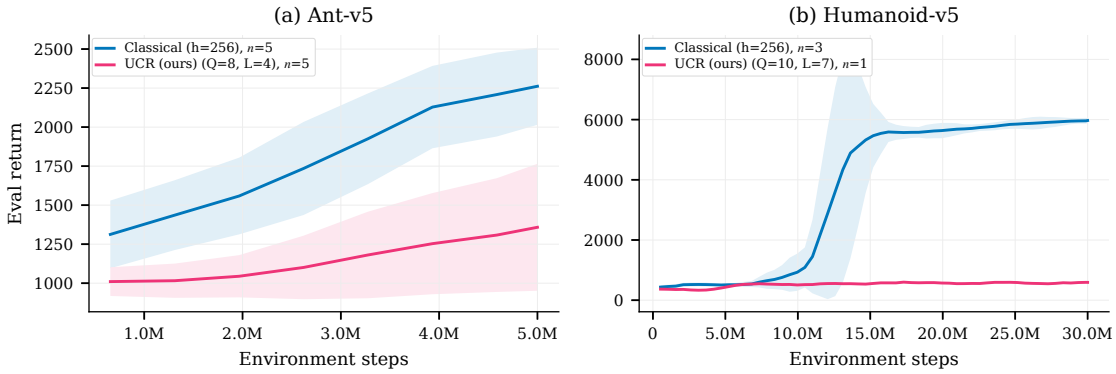


Fig. 13. High-dimensional feasibility curves on (a) Ant-v5 and (b) Humanoid-v5. This experiment is solely for verifying the engineering feasibility of UCR under high-dimensional observations and is not included in the main paper's performance claims.

Table 12. High-dimensional feasibility check on Ant-v5 (obs_dim $= 105$) and Humanoid-v5 (obs_dim $= 348$).

Environment	Method	n	Best eval	Final eval	Eval AUC
Ant	Classical	5	2473.2 ± 87.2	2423.0 ± 150.2	1794.5 ± 86.7
Ant	**UCR** (ours)	5	1445.2 ± 358.3	1398.1 ± 416.0	1143.4 ± 209.9
Humanoid	Classical	3	5976.6 ± 87.2	5967.5 ± 69.6	3673.9 ± 442.3
Humanoid	**UCR** (ours)	1	648.3	636.3	525.8

95% CI by Student's $t_{0.975,n-1} \cdot$ SE.

The feasibility results on Ant-v5 and Humanoid-v5 (Fig. 13 and Table 12) are used to verify whether UCR can be

constructed and trained under high-dimensional observations, rather than for performance claims in the main text.

The observation dimension of Ant-v5 is 105, and UCR uses 8 qubits to complete PPO training and generate a steadily rising learning curve. Its Best Eval Return is 1445.2 ± 358.3, lower than Classical's 2473.2 ± 87.2, but this demonstrates that the UCR frontend can be integrated into the standard PPO workflow with hundred-dimensional inputs while maintaining non-degenerate learning.

Humanoid-v5 requires caution. For this task with 348 observation dimensions, UCR uses 10 qubits. Single-seed results show a Best Eval Return of 648.3, far below the Classical result of 5976.6 ± 87.2. This indicates that in higher-dimensional tasks with more complex dynamics, the current UCR configuration still faces significant challenges in representation and optimization. It demonstrates "runnability" and "resource accounting", rather than "fully effective high-dimensional performance."

Therefore, UCR possesses the engineering feasibility to scale to high-dimensional observation tasks, but to achieve a performance advantage, it requires stronger quantum encoding designs, more appropriate training-task optimization designs, and further validation with additional seeds.

A2. Additional VQC-Gradient Diagnostics

This section adopts the set of common retained parameters $\mathcal{J}_{\text{common}}$ from Eq. (46) in the main text; all small gradients arising from the input states themselves are still retained. As a supplement to the gradient variance analysis in the main text, we further report the overall root-mean-square gradient magnitude and the fraction of near-zero gradients:

$$G_{\text{RMS}} = \sqrt{\underset{m,r,j\in\mathcal{J}_{\text{common}}}{\text{mean}} g_{rmj}^2} = \sqrt{\frac{1}{MR\,|\mathcal{J}_{\text{common}}|}\sum_{m=1}^{M}\sum_{r=1}^{R}\sum_{j\in\mathcal{J}_{\text{common}}} g_{rmj}^2},$$
$$\rho_{<10^{-6}} = \underset{m,r,j\in\mathcal{J}_{\text{common}}}{\text{mean}} \mathbf{1}\left(|g_{rmj}| < 10^{-6}\right). \tag{54}$$

Here, m indexes the $M = 32$ input observations, r indexes the $R = 200$ independent random VQC initializations, and j indexes the commonly retained non-structurally-zero parameters; G_{RMS} measures the overall gradient magnitude, while $\rho_{<10^{-6}}$ measures the proportion of retained gradients that are close to zero and thus unlikely to provide effective update signals.

When $Q = 10$ and $L = 1$, the proportions of near-zero gradients for UCR, Angle, and Amplitude are 2.8%, 12.1%, and 0.2%, respectively. Therefore, although the policy-gradient variance of UCR decreases rapidly as Q increases, within the scope of this study's testing, this decrease has not yet manifested as widespread numerical vanishing of parameter gradients. The depth sweep further indicates that the gradient differences among the three encoding schemes primarily occur in shallow circuits and gradually diminish as the VQC depth increases; under local loss conditions, the difference in gradient variance among the three encoding schemes is less than approximately 5% when $L = 4$. Therefore, the encoding scheme primarily affects the gradient scale in shallow circuits, while circuit structure and the loss function gradually become more significant factors as the circuit deepens.

Confidence intervals for the gradient variance statistics were obtained using $10\,000$ bootstrap resamples, with a random seed of 2029. For a fixed Q, the three encoding schemes were compared using a statistical paired comparison with the same observations and weight configurations; resampling was performed independently for different Q values, without assuming a one-to-one correspondence between different qubit scales.

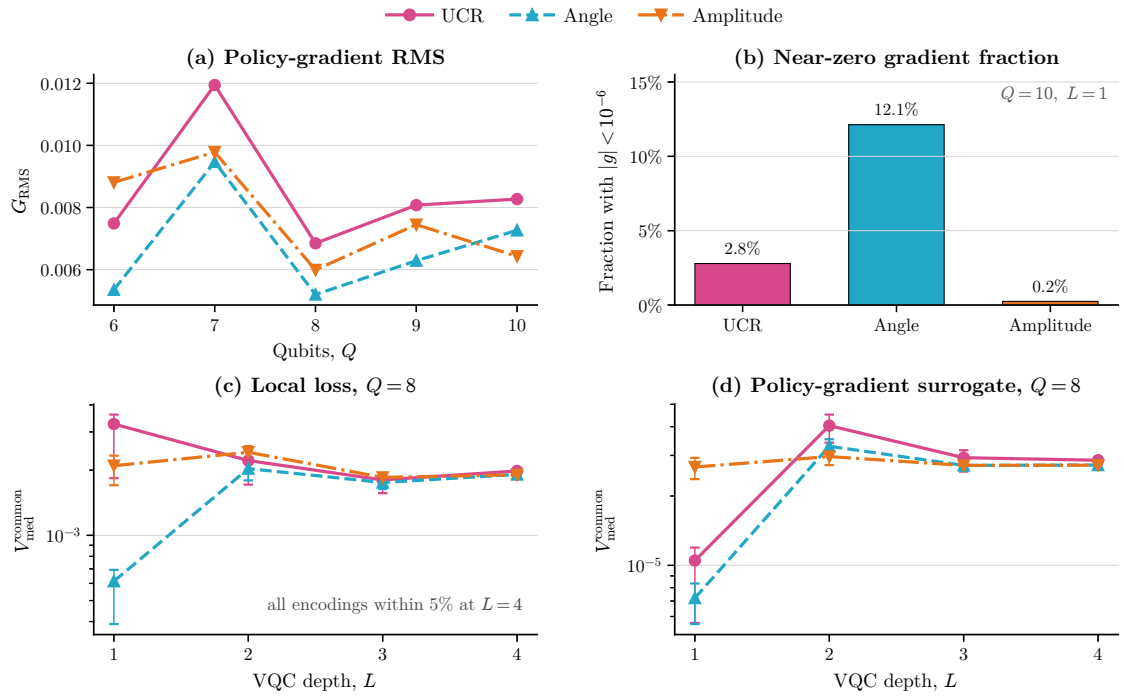


Fig. 14. VQC-gradient diagnostics. (a) Overall root-mean-square policy-gradient magnitude G_{RMS} (Eq. (54)), over $Q = 6$–10. It tests whether the overall gradient magnitude remains numerically usable. (b) Fraction of retained policy-gradient entries with $|g| < 10^{-6}$ at $Q = 10$, $L = 1$, measuring how many gradients provide almost no update signal. (c) and (d) Depth sweeps at $Q = 8$ for the local loss and policy-gradient surrogate, respectively, testing whether encoding-dependent differences persist in deeper circuits. The $L = 1$ values in (c) and (d) coincide with the $Q = 8$ points in Fig. 10(b) and (c). Error bars denote 95% bootstrap confidence intervals.

A3. Full Task-Level Noise Scan

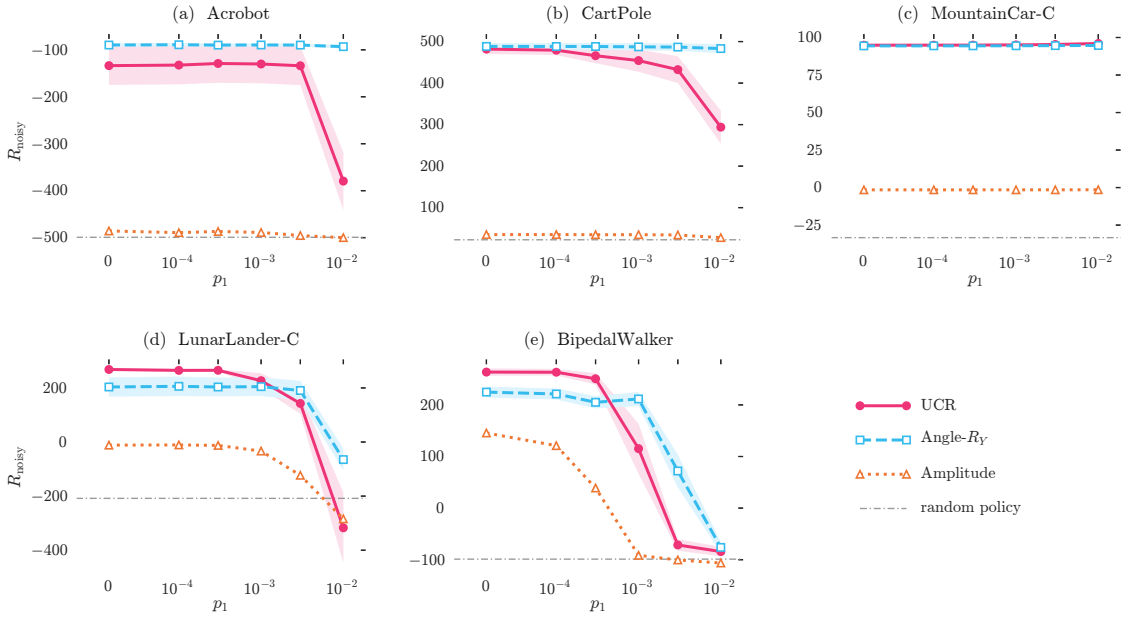


Fig. 15. Full gate-level depolarizing-noise scan across the five main tasks and three quantum encoding interfaces. Curves show mean final-checkpoint noisy returns over 10 training seeds; shaded bands denote seed-level standard errors and dash-dotted lines denote random-policy returns. The channel parameters satisfy $p_2 = 10p_1$.

The full scan in Fig. 15 shows that UCR's low-noise advantage is primarily evident in LunarLanderContinuous-v3 and BipedalWalker-v3, where its noise-free performance is already superior. Acrobot-v1 and CartPole-v1 provide examples

in which Angle-R_Y is more stable, illustrating that this advantage is task-dependent.

A4. Multi-Channel Noise Results

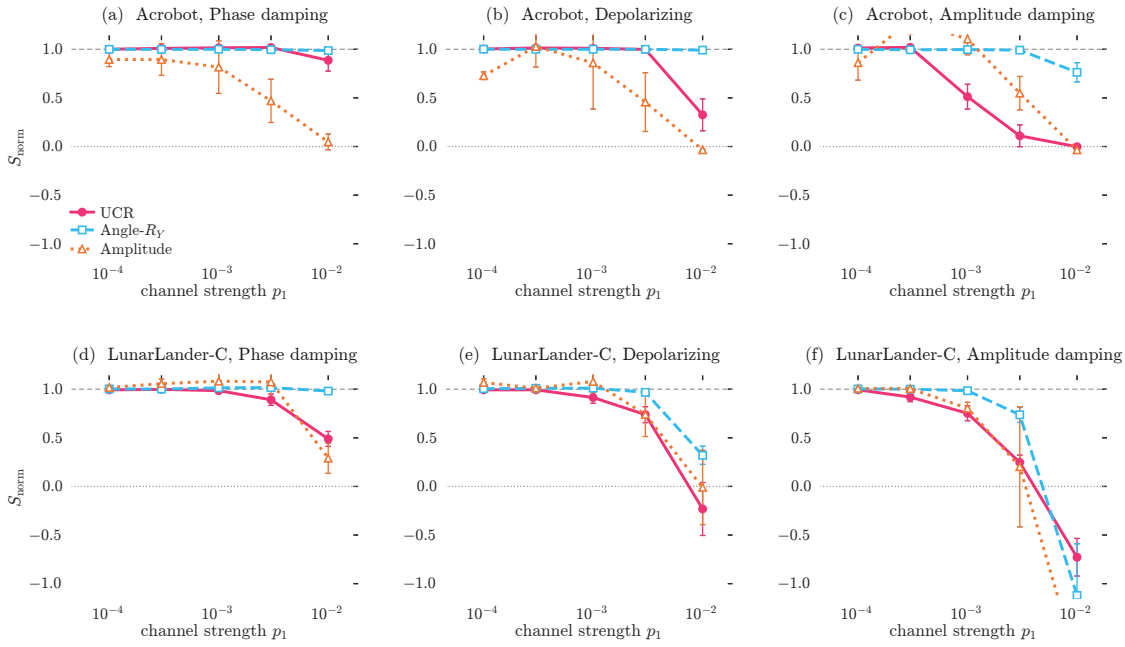


Fig. 16. Normalized performance retention S_{norm} (Eq. (43)) of UCR, Angle-R_Y and amplitude encoding under phase-damping, depolarizing and amplitude-damping channels in panels (a)–(c) for Acrobot-v1 and in panels (d)–(f) for LunarLanderContinuous-v3. S_{norm} is computed per seed and then averaged; error bars denote seed-level SEs. Valid-seed counts $n_{\text{valid}}/10$: Acrobot-v1, UCR 9/10, Angle-R_Y 10/10, amplitude 4/10; LunarLanderContinuous-v3, 10/10 for all encodings. The displayed range is limited to $[-1, 1]$; the amplitude-encoding value on LunarLanderContinuous-v3 under amplitude damping at $p_1 = 10^{-2}$ ($S_{\text{norm}} = -1.81$) lies below the panel.

Fig. 16 extends the three-channel comparison from the single task described in the main text to the Acrobot-v1 and LunarLanderContinuous-v3 tasks and three encoding methods. This figure is intended solely to compare the sensitivity of the encodings to different channels and is not intended to establish a ranking of physical severity.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (No.61673389).

References

[1] Park S and Kim J 2024 *ETRI J.* **46** 748

[2] Hohenfeld H, Heimann D, Wiebe F and Kirchner F 2024 *IEEE Access* **12** 87217

[3] Lin D, Wang J, Tseng H H, Zhao T, Yue M and Yoo S 2025 *IEEE Open Access J. Power Energy* **12** 833

[4] Kim G S, Chung J and Park S 2024 *IEEE Trans. Veh. Technol.* **73** 12252

[5] Dalla Pozza N, Buffoni L, Martina S and Caruso F 2022 *Quantum Mach. Intell.* **4** 11

[6] Jin Y X, Wang Z W, Xu H Z, Zhuang W F, Hu M J and Liu D E 2025 arXiv:2501.07085v1 [quant-ph]

[7] Jin Y X, Xu H Z, Wang Z A, et al. 2024 *Chin. Phys. B* **33** 050301

[8] Kim G S, Lee S, Cho I S, Park S and Kim J 2025 *IEEE Internet Things J.* **12** 28986

[9] Andrés E, Cuéllar M P and Navarro G 2023 *IEEE Access* **11** 104534

[10] Roca-Jerat S, Román-Roche J and Zueco D 2024 *Mach. Learn.: Sci. Technol.* **5** 015057

[11] Clader B D, Dalzell A M, Stamatopoulos N, Salton G, Berta M and Zeng W J 2022 *IEEE Trans. Quantum Eng.* **3** 3103323

[12] Wang X, Wang Y, Qi B and Wu R 2025 arXiv:2503.01545v1 [quant-ph]

[13] Escudero E T, Alamo D A, Gómez O M and Bringas P G 2023 *Hybrid Artificial Intelligent Systems: 18th International Conference, HAIS 2023*, September 5–7, 2023, Salamanca, Spain, p. 314

[14] Möttönen M, Vartiainen J J, Bergholm V and Salomaa M M 2005 *Quantum Inf. Comput.* **5** 467

[15] Han J, DiBrita N S, Cho Y, Luo H and Patel T 2025 *Proceedings of the 62nd ACM/IEEE Design Automation Conference (DAC 2025)*, June 22–25, 2025, San Francisco, CA, USA, p. 1

[16] Camps D and Van Beeumen R 2022 *Proceedings of the IEEE International Conference on Quantum Computing and Engineering (QCE 2022)*, September 18–23, 2022, Broomfield, Colorado, USA, p. 104

[17] Nghiem N A 2025 arXiv:2501.07891v1 [quant-ph]

[18] Sünderhauf C, Campbell E and Camps J 2024 *Quantum* **8** 1226

[19] Wan L C, Yu C H, Pan S J, Qin S J, Gao F and Wen Q Y 2021 *Phys. Rev. A* **104** 062414

[20] Lin L 2022 arXiv:2201.08309v1 [quant-ph]

[21] Balewski J, Pestano C, Amankwah M G, Bethel E W, Perciano T and Van Beeumen R 2025 arXiv:2502.15928v3 [quant-ph]

[22] Stamatopoulos N, Clader B D, Woerner S and Zeng W J 2024 arXiv:2404.10088v1 [quant-ph]

[23] Gui K, Dalzell A M, Achille A, Suchara M and Chong F T 2024 *Quantum* **8** 1257

[24] Shi H, Han C H, Wang P and Zhang M 2025 *Chin. Phys. B* **34** 120304

[25] Chen Z L, Guan Z J, Zhao S X and Cheng X Y 2025 *Chin. Phys. B* **34** 050305

[26] Ajibosin S S and Cetinkaya D 2024 *Software* **3** 498

[27] Yang F, Wang F, Xu X, Gao P, Xin T, Wei S J and Long G 2025 *Phys. Rev. Res.* **7** 013007

[28] Möttönen M, Vartiainen J J, Bergholm V and Salomaa M M 2004 *Phys. Rev. Lett.* **93** 130502

[29] Le P Q, Dong F and Hirota K 2011 *Quantum Inf. Process.* **10** 63

[30] Amankwah M G, Camps D, Bethel E W, Van Beeumen R and Perciano T 2022 *Sci. Rep.* **12** 7712

[31] Balewski J, Amankwah M G, Van Beeumen R, Bethel E W, Perciano T and Camps D 2024 *Sci. Rep.* **14** 3435

[32] Bergholm V, Izaac J, Schuld M, Gogolin C and Killoran N 2018 arXiv:1811.04968v1 [quant-ph]

[33] Shende V V, Bullock S S and Markov I L 2006 *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **25** 1000

[34] Plesch M and Brukner Č 2011 *Phys. Rev. A* **83** 032302

[35] Agarwal R, Schwarzer M, Castro P S, Courville A C and Bellemare M G 2021 *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, December 6–14, 2021, Virtual Conference, p. 29304

[36] McClean J R, Boixo S, Smelyanskiy V N, Babbush R and Neven H 2018 *Nat. Commun.* **9** 4812

[37] Cerezo M, Sone A, Volkoff T, Cincio L and Coles P J 2021 *Nat. Commun.* **12** 1791

[38] Page D N 1993 *Phys. Rev. Lett.* **71** 1291

[39] Ragone M, Bakalov B N, Sauvage F, Kemper A F, Ortiz Marrero C, Larocca M and Cerezo M 2024 *Nat. Commun.* **15** 7172