Quantum Autoencoder: An efficient approach to quantum feature map generation

Shengxin Zhuang[1,6,†], **Yusen Wu**[2,†], **Xavier F. Cadet**[3], **Du Q. Huynh**[4], **Wei Liu**[4], **Philippe Charton**[6,7], **Cedric Damour**[8], **Frédéric Cadet**[5,6,7,§], **Jingbo Wang**[1,‡]

1 Department of Physics, The University of Western Australia, Perth, WA, Australia
2 School of Artificial Intelligence, Beijing Normal University, Beijing, China
3 Department of Computing, Imperial College London, United Kingdom
4 Department of Computer Science and Software Engineering, The University of Western Australia, Perth, WA, Australia
5 PEACCEL, Artificial Intelligence Department, AI for Biologics, France
6 University of Paris City & University of Reunion, France
7 Laboratory of Excellence GR-Ex, France
8 EnergyLab, EA 4079, Faculty of Sciences and Technology, University of Reunion, France
[†]These authors contributed equally.
[‡] jingbo.wang@uwa.edu.au
[§]frederic.cadet.run@gmail.com

September 24, 2025

Abstract

Quantum machine learning methods often rely on fixed, hand-crafted quantum encodings that may not capture optimal features for downstream tasks. In this work, we study the power of quantum autoencoders in learning data-driven quantum representations. We first theoretically demonstrate that the quantum autoencoder method is efficient in terms of sample complexity throughout the entire training process. Then we numerically train the quantum autoencoder on 3 million peptide sequences, and evaluate their effectiveness across multiple peptide classification problems including anti-hypertensive peptide prediction, blood-brain barrier-penetration, and cytotoxic activity detection. The learned representations were compared against Hamiltonian-evolved baselines using a quantum kernel with support vector machines. Results show that quantum autoencoder learned representations achieve accuracy improvements ranging from 0.4% to 8.1% over Hamiltonian baselines across seven datasets, demonstrating effective generalization to diverse downstream datasets with pre-training enabling effective transfer learning without task-specific fine-tuning. This work establishes that quantum autoencoder architectures can effectively learn from large-scale datasets (3 million samples) with compact parameterizations ($\sim$900 parameters), demonstrating their viability for practical quantum applications.

1 Introduction

Quantum machine learning (QML) has emerged as a promising field that integrates principles of quantum computing with classical machine learning objectives, offering the potential for advantage expressivity and pattern recognition for high-dimensional learning tasks [1]. Central to QML model performance is the encoding of classical data into quantum states, a critical preprocessing step that fundamentally determines the model's representational capacity and generalization ability [2, 3]. Whether in quantum support vector machines (QSVMs) [4, 5, 6], variational quantum classifiers (VQCs) [7], or quantum generative models [8, 9, 10], the choice of quantum state representation is often a key bottleneck that determines downstream performance.

In practice, most QML pipelines rely on fixed, task-agnostic encodings to convert classical data into quantum states. Common strategies include *angle encoding* [11], where scalar features are mapped to qubit rotations; *amplitude encoding* [12], which encodes normalized feature vectors into the amplitude of quantum states; and *Hamiltonian evolution* [13, 14], where classical data is used to parametrize a system Hamiltonian and the resulting time-evolved quantum state serves as the data representations. These approaches are appealing due to their computational simplicity and circuit efficiency. However, they suffer from a key limitation: they are static and unable to adapt to the statistical structure of specific datasets or optimize for the downstream learning task. In contrast, classical machine learning has achieved major performance gains through learned representations enabled by deep learning architectures, including unsupervised pre-training models like protein language models (PLMs) that are specifically adapted to both the data distribution and downstream tasks [15]. The quantum field has yet to fully embrace this data-driven paradigm. Most current quantum encodings remain manually designed and disconnected from the structure of the input data or the requirements of the learning task, potentially limiting the effectiveness of QML models in practice.

This contrast has prompted interest in quantum analogs of representation learning. Among these, the quantum autoencoder has been proposed as a variational circuit architecture that learns to compress quantum states by discarding redundant qubits. While quantum autoencoders were originally developed for tasks such as quantum state compression, they offer an appealing framework for learning quantum representations for classical data. Despite this, their role in supervised QML pipelines, especially in learning general-purpose representations for downstream classification, remains underexplored. Moreover, few works have investigated the scalability of quantum autoencoder training on large classical datasets encoded into quantum states.

In this work, our primary objective is not to demonstrate quantum advantage over classical approaches but to establish that quantum autoencoders can learn useful representations that provide measurable improvements in supervised peptide classification tasks compared to fixed Hamiltonian encodings. This focus isolates the contribution of representation learning within quantum models themselves, providing an explicit test of whether unsupervised quantum pre-training offers benefits beyond static, hand-crafted encodings. To this end, we pre-trained quantum autoencoders with a large dataset of over 3 million peptide sequences and evaluated the learned representations across seven datasets in computational biology. Comparisons to classical PLMs are included only as a separate contextual benchmark, where PLMs are evaluated against one-hot encodings using classical classifiers. We do not compare quantum autoencoders directly against PLMs or other classical methods; rather, the central contribution of this work is the demonstration that quantum autoencoders improve over Hamiltonian baselines within the quantum settings. This work also establishes a methodology for unsupervised quantum representation learning that may generalize to other QML applications.

We train quantum autoencoders with different compression schemes and circuit depths, then extract the learned compressed quantum states as fixed representations for each input from the downstream datasets. These representations are evaluated using support vector machines (SVMs) [16] with trace distance kernels. As a baseline, we compare against Hamiltonian evolved quantum states encoded from the same downstream datasets. This setup allows us to isolate the impact of the learned representation on classification performance while keeping the downstream classifier and data consistent.

Our results show that quantum autoencoder learned representations consistently outperform Hamiltonian-based encodings by 0.4% to 8.1% in test accuracy across all tasks while demonstrating strong transferability: models trained in an unsupervised fashion on large-scale data generalize effectively to diverse downstream tasks without requiring fine-tuning. This establishes quantum autoencoders as effective pre-trained quantum representation learners, establishing a quantum analog to classical representation learning models.

In summary, this work presents an empirical study of quantum autoencoders as a practical component in QML workflows. By applying quantum autoencoders to large-scale peptide datasets and benchmarking their learned representations across several bioinformatics classification tasks, we demonstrate their potential in real-world settings where data structures are complex and hand-crafted encodings may fall short. To the best of our knowledge, the quantum autoencoders presented here are pre-trained on 3 million peptide sequences, representing one of the largest datasets used for representation learning in QML to date. Our findings suggest that quantum autoencoders offer a scalable and effective approach to representation learning, with implications for transfer learning and hybrid quantum-classical pipelines.

2 Related Work

2.1 Quantum Data Encoding

Encoding classical data into quantum states is a fundamental step in QML pipelines, as it determines the subspace of the Hilbert space explored by the models. Common approaches include:

- **Angle encoding:** For a data vector $\boldsymbol{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,n})$, each scalar value $x_{i,j}$ is used to parametrize a single-qubit rotation [11],

$$|0\rangle^{\otimes n} \mapsto \bigotimes_{j=1}^{n} R_y(x_{i,j})\,|0\rangle = \bigotimes_{j=1}^{n} \left(\cos(x_{ij}/2)|0\rangle + \sin(x_{ij}/2)|1\rangle\right).$$

- **Amplitude encoding:** The normalized vector $\boldsymbol{x}_i \in \mathbb{R}^{2^n}$ is encoded directly into the amplitudes of a quantum state [12],

$$\boldsymbol{x}_i \;\mapsto\; |\psi(\boldsymbol{x}_i)\rangle = \sum_{j=0}^{2^n-1} x_{i,j}|j\rangle, \quad \|\boldsymbol{x}_i\|_2 = 1.$$

- **Hamiltonian evolution encoding:** The vector $\boldsymbol{x}_i$ parametrizes a Hamiltonian $H(\boldsymbol{x}_i)$, and the encoded state is obtained by time evolution [13, 14],

$$|\psi(\boldsymbol{x}_i)\rangle = e^{-iH(\boldsymbol{x}_i)t}\,|0\rangle^{\otimes n}.$$

These encodings have been central to early QML experiments because of their simplicity and relatively low circuit depth. In this work, we selected the Hamiltonian evolution encoding as a fixed baseline against which to compare the learned, data-driven quantum representation.

2.2 Quantum Autoencoders

Quantum autoencoders are variational quantum circuits designed to efficiently compress quantum states by removing redundant degrees of freedom while retaining the information necessary to reconstruct the original state [17]. The typical architecture partitions the qubits into a latent register, which retains compressed information, and a trash register, which is discarded after compression. The parameterized unitary is trained to maximize the fidelity between the trash register and a reference state via a SWAP test [18].

Originally, quantum autoencoders were proposed for efficient compression of quantum data [17]. Subsequent work has extended quantum autoencoders to denoising quantum data [19] and experimental demonstrations with photonic systems [20], where data compression has been realized using photons.

Beyond compression, the quantum variational autoencoder (QVAE) [21] adapts the autoencoder concept for generative modeling. In the QAVE, the latent distribution is modeled by a quantum Boltzmann machine, enabling both data generation and representation learning. This allows sampling new data from quantum enhanced latent variables, shifting the focus from compressing quantum states to learning quantum enhanced generative processes.

Other recent work has also explored autoencoder-based quantum architectures in molecular domains. A quantum variational autoencoder with spherical latent variable learning has been applied to 3D molecule generation, where the goal is to learn latent variables that capture molecular geometry and enable the synthesis of novel molecular structures [22]. In parallel, MolQAE [23] introduced a quantum autoencoder framework for molecular representation learning, focusing on compressing SMILES-based molecular encodings into a latent quantum register. Their evaluation was based on the reconstruction fidelity, demonstrating that compressed states could retain information about the original input. However, the learned representations were not applied to downstream supervised tasks such as property prediction, leaving their practical usefulness for machine learning unexplored. In contrast, our work explicitly evaluates quantum autoencoder representations in peptide classification tasks, benchmarking against Hamiltonian kernels. This direct use of learned representations for prediction differentiates our study, showing not only that the quantum autoencoder can compress quantum states but also that the resulting representations improve downstream performance.

2.3 Representation Learning for Sequence Data

The success of machine learning algorithms generally depends on the choice of data representation, which can hide or reveal the different explanatory factors of variation behind the data [24]. The learned representations have driven major advances in machine learning by automatically learning features that capture richer patterns than manual engineering and enable effective transfer learning.

The bioinformatics field has begun to adopt this approach in recent years. Early computational approaches to peptide and protein modeling relied on fixed encodings such as amino acid composition or physicochemical descriptors. While useful in certain contexts, these handcrafted features were limited in their ability to capture the complex dependencies that govern biological sequence-function relationships. More recently, PLMs such as ESM, ProtBERT, UniRep [25, 26, 27], which are trained in an unsupervised (more precisely, self-supervised) manner on millions of sequences, as well as PeptideBERT, which fine-tunes ProtBERT for peptide property prediction tasks including hemolysis and solubility [28]. These models demonstrate that unsupervised pre-training on large unlabeled datasets can produce general purpose representations that transfer effectively to diverse supervised prediction problems.

Most existing QML literature has focused on demonstrating theoretical quantum advantage or on developing efficient state preparation methods that exploit symmetries or well-defined mathematical structures [29, 30], whereas in many real-world problems the underlying structure is unclear or poorly defined. For example, in biological sequence modeling, the relationship between peptide or protein sequence and function is highly complex and remains poorly understood, making it difficult to capture with fixed encodings. This motivates the need for adaptive, data-driven quantum representation learning approaches, such as quantum autoencoders, which can make use of large unlabeled datasets to learn representations that could potentially improve downstream task performance.

3 Methods

Our workflow consists of two main stages:

(1) unsupervised pre-training of a quantum autoencoder on Hamiltonian encoded quantum states from 3 million unlabeled peptide sequences;

(2) downstream evaluation of the learned representation on seven peptide supervised classification tasks.

The Hamiltonian encoded states play two roles: (i) they serve as the input to the quantum autoencoder; (ii) they are also used to construct a kernel that trains classical SVMs to provide a baseline for comparison. The quantum autoencoder compresses quantum states by discarding trash qubits, with the remaining subsystem serving as the learned representation. These representations are then used to construct a kernel for SVMs classification. A schematic of the overall workflow is shown in Fig. 1.

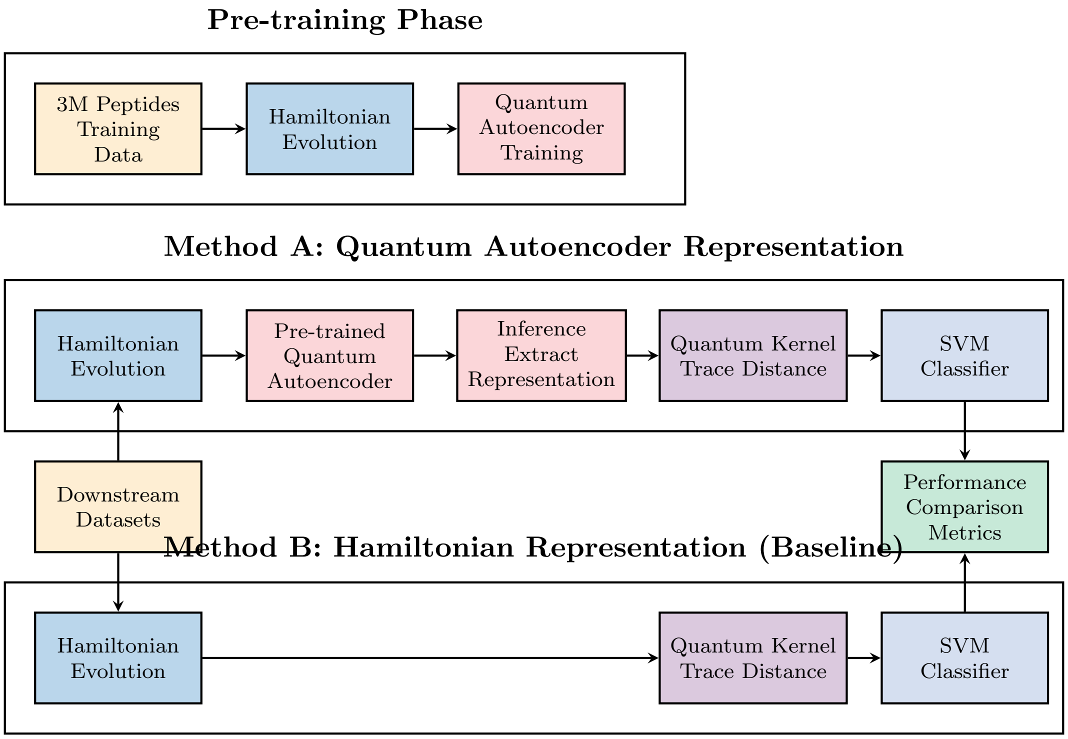
Pre-training Phase
3M Peptides Training Data
Hamiltonian Evolution
Quantum Autoencoder Training
Method A: Quantum Autoencoder Representation
Hamiltonian Evolution
Pre-trained Quantum Autoencoder
Inference Extract Representation
Quantum Kernel Trace Distance
SVM Classifier
Downstream Datasets
Performance Comparison Metrics
Method B: Hamiltonian Representation (Baseline)
Hamiltonian Evolution
Quantum Kernel Trace Distance
SVM Classifier

Figure 1: Methodology Overview: Quantum Autoencoder vs Hamiltonian Baseline Comparison. The framework consists of three phases: (1) Pre-training on 3M peptide sequences to learn quantum representations, (2) Method A using pre-trained quantum autoencoders for downstream tasks, and (3) Method B using direct Hamiltonian evolution as baselines. Both methods employ quantum trace distance as a kernel for SVM classification, enabling direct performance comparison.

3.1 Representation Pipelines

All experiments used the same peptide datasets, but the initial data representation varied by method. Quantum approaches (the quantum autoencoder and Hamiltonian baseline) and classical baselines started from one-hot encoding matrices, where rows correspond to amino acid type and columns to sequence positions. These matrices were flattened into binary vectors for subsequent processing. On the other hand, PLMs operated directly on raw amino acid sequences, applying their own tokenization before producing representations. Detailed dataset collection, preparation, and preprocessing steps are provided in Appendix D.1 (pre-training) and Appendix D.2 (downstream).

Hamiltonian encoding baseline. The flattened one-hot vectors parametrized a system Hamiltonian, and the resulting time-evolved states were used directly to construct a kernel for support vector machines (SVMs). This serves as the fixed quantum encoding baseline (see Section 3.4.2 for kernel construction details).

Quantum autoencoder. The same Hamiltonian-encoded states were provided as input to the quantum autoencoder, which compressed them by tracing out designated trash qubits. The remaining subsystem was treated as the learned representation. Reduced density matrices from this subsystem were then used to construct kernels for SVM classification (see

Section 3.4.1 for details).

Classical baselines. The flattened one-hot vectors were used as input to five candidate classifiers: SVM with an RBF kernel, SVM with a linear kernel, random forest [31], XG-Boost [32], and logistic regression [33]. For each dataset, the best-performing model was selected based on cross-validated accuracy.

Protein language models (PLMs). Raw amino acid sequences were provided as input to pre-trained PLMs, including ESM, ProtBERT, and related models [25, 26]. These models applied their own tokenization and generated sequence representations, which were subsequently evaluated with the same set of five classical classifiers used for the classical baselines. The best performing PLM-classifier combination was reported for each dataset, and the full list of PLMs considered is provided in Appendix C.

This setup enables two types of comparisons. In the quantum setting, we evaluate whether quantum autoencoder representations provide improvements over fixed Hamiltonian encoding baselines. In the classical setting, we evaluate whether pre-trained PLMs provide improvements over traditional one-hot encodings when combined with classical classifiers. These paired comparisons focus on the relative contribution of learned versus fixed representations in each domain, with all classifiers evaluated under 5-fold cross-validation to ensure fairness and consistency.

3.2 Quantum Autoencoder

The quantum autoencoder consists of a parametrized encoder circuit that compresses the input state by tracing out some trash qubits, leaving the remaining subsystem as the learned representation. During training, for each one-hot vector $\boldsymbol{x}_i \in \mathcal{D}$, we first map the classical data $\boldsymbol{x}_i$ to n-qubit Hilbert space $|\psi(\boldsymbol{x}_i)\rangle$ by using the Hamiltonian encoding method. Then, a n-qubit, d-layer variational quantum circuit $U(\vec{\boldsymbol{\theta}})$ is performed on $|\psi(\boldsymbol{x}_i)\rangle$, followed by a computational basis measurement to the m trash qubits, where $m = \mathcal{O}(1)$ and $m \ll n$. The probability of obtaining all $|0\rangle$ outcomes is used as the loss function, that is

$$\mathcal{L}(\vec{\boldsymbol{\theta}}) = \mathrm{Tr}\left[\Pi_m U(\vec{\boldsymbol{\theta}}) \mathbb{E}_{\boldsymbol{x}_i \sim \mathcal{D}} \left(|\psi(\boldsymbol{x}_i)\rangle\langle\psi(\boldsymbol{x}_i)|\right) U^\dagger(\vec{\boldsymbol{\theta}})\right], \tag{1}$$

where the m-qubit local observable $\Pi_m = |0\rangle\langle 0|^{\otimes m}$. The encoder parameters $\vec{\boldsymbol{\theta}}$ are optimized to maximize the probability $\mathcal{L}(\vec{\boldsymbol{\theta}})$, effectively pushing information away from the trash subsystem and into the remaining subsystem, which then serves as the learned representation. Notably, we only train the encoder in the training process, which is mathematically equivariant to train both encoder and decoder when they share the same quantum circuit. Specifically, let $\boldsymbol{\theta}^* = \arg\max_{\vec{\boldsymbol{\theta}}} \mathcal{L}(\vec{\boldsymbol{\theta}})$, the learned quantum feature state (density matrix) is

$$\phi(\boldsymbol{x}_i) = \mathrm{Tr}_m \left[U(\boldsymbol{\theta}^*)|\psi(\boldsymbol{x}_i)\rangle\langle\psi(\boldsymbol{x}_i)|U^\dagger(\boldsymbol{\theta}^*)\right]. \tag{2}$$

In the following, we provide theoretical guarantees for the quantum autoencoder method. Our first result focuses on the quantum circuit depth of $U(\vec{\boldsymbol{\theta}})$ used in the quantum autoencoder.

Theorem 1 (Informal)**.** *Given the Hamiltonian encoding method $|\psi(\boldsymbol{x}_i)\rangle = e^{-iH(\boldsymbol{x}_i)t}|0^n\rangle$, where the evolution time $t = \mathcal{O}(1)$, the quantum circuit depth of $U(\vec{\boldsymbol{\theta}})$ satisfies $d \leq$* $\mathrm{poly}\log(nt/\epsilon)$ *sufficing to implement the quantum autoencoder task.*

Proof. The fundamental idea of a quantum autoencoder is to approximately decouple m qubits from the whole quantum system. Without loss of generality, we assume the Hamiltonian $H(\boldsymbol{x}_i)$ is defined on a constant-dimensional lattice, and that $\max\{\|\boldsymbol{x}_i\|_\infty |t|\}$ is a constant which does not increase with the system size. According to the Lieb-Robinson bound [34], the required quantum circuit depth for the quantum feature state $|\psi(\boldsymbol{x}_i)\rangle$ is $\operatorname{poly}\log(nt/\epsilon)$. On the other hand, we label the target qubits to be trashed as $Q = \{q_1, \cdots, q_m\}$. Since the information only spreads within the light cone, as a result, a quantum circuit $U(\vec{\boldsymbol{\theta}})$ with circuit depth $d = \operatorname{poly}\log(nt/\epsilon)$ suffices to decouple the qubit set Q back to $|0^m\rangle$. □

In general, training the variational quantum circuit $U(\vec{\boldsymbol{\theta}})$ would be classically hard in the worst-case scenario [35]. Here, we rigorously prove that the sample complexity in training the quantum autoencoder is efficient, which can be summarized as the following result.

Theorem 2. *For any quantum autoencoder task with loss function $\mathcal{L}(\vec{\boldsymbol{\theta}})$, then*

$$N = \mathcal{O}\left(\frac{3^{\log(1/\epsilon\delta)}\log(1/\epsilon\delta)\log(n)}{\epsilon^2}\right) \tag{3}$$

samples drawn from $\mathbb{E}_{\boldsymbol{x}_i\sim\mathcal{D}}\left[|\psi(\boldsymbol{x}_i)\rangle\langle\psi(\boldsymbol{x}_i)|\right]$ suffice to estimate $\mathcal{L}(\vec{\boldsymbol{\theta}})$ throughout the entire training process, with success probability $\geq 1-\delta$. In other words, N-samples are sufficient to guarantee an ϵ-approximation to $\mathcal{L}(\vec{\boldsymbol{\theta}})$ for all candidate variational parameters $\vec{\boldsymbol{\theta}}$.

The overall structure of the quantum autoencoder circuit is illustrated in Fig. 2.

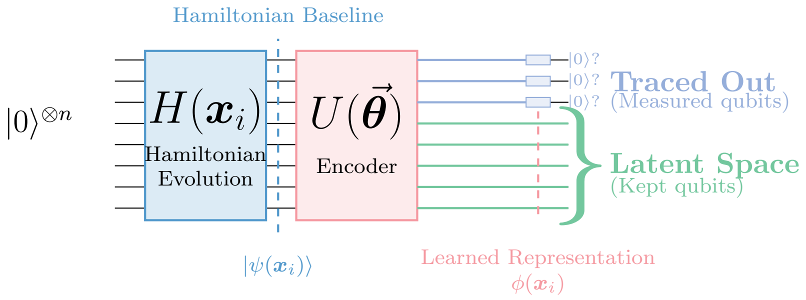

Figure 2: Quantum autoencoder architecture for unsupervised representation learning. The circuit processes input peptide sequences in two stages: (1) Hamiltonian evolution $H(\boldsymbol{x}_i)$ encodes classical data into quantum states, and (2) a parameterized encoder $U(\vec{\boldsymbol{\theta}})$ performs dimensional compression. The encoder discards qubits (blue) while preserving latent information (green), producing compressed quantum representations $\phi(\boldsymbol{x}_i)$ for downstream peptide classification tasks.

3.3 Pre-training Setup

Training was performed on a large-scale dataset of 3 million peptide sequences, represented as flattened one-hot vectors and then mapped into quantum states via Hamiltonian evolution. We used mini-batch training with a batch size of 1024, optimizing with Adam [36] at a learning rate of 10^{-2}. Models with 8 input qubits were trained for up to 10 epochs, while models with 10 input qubits were trained for up to 5 epochs. To ensure fairness, we trained 18 quantum autoencoder variants covering all combinations of input size (8 or 10 qubits), circuit depth (10, 20, or 30 layers), and number of traced-out qubits (1, 2, or 3). The best-performing variant for each dataset was selected based on cross-validation accuracy, analogous to hyperparameter optimization in classical machine learning. The Hamiltonian baseline, by contrast, involves no trainable parameters and serves as a fixed reference encoding, but was evaluated under the same cross-validation splits for comparability.

Our aim in training multiple models was not to analyze architectural effects individually, but to ensure coverage of a representative model space. This setup provides a fair test of whether the quantum autoencoder can successfully learn from large-scale peptide data and produce representations that outperform fixed Hamiltonian encodings.

3.4 Downstream Evaluation

3.4.1 Quantum Autoencoder Kernel

Once trained, the quantum autoencoder provides compressed representations obtained from the unmeasured subsystem of the quantum state. To evaluate these representations in supervised learning tasks, we construct a similarity kernel as follows. First, pairwise trace distances are computed between the reduced density matrices corresponding to different samples. These distances are then converted into similarities by applying $k_{\text{QAE}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1 - \text{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j)$, where the trace distance

$$\text{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{1}{2}\|\phi(\boldsymbol{x}_i) - \phi(\boldsymbol{x}_j)\|_1 \tag{4}$$

and the quantum feature state $\phi(\boldsymbol{x}_i)$ (a $(2^{n-m} \times 2^{n-m})$-sized density matrix) is given by Eq. 2. Generally, estimating the trace distance between mixed states are even quantum hard. However, the condition $m \ll n$ enables $\phi(\boldsymbol{x}_i)$ being a low-rank density matrix for $i \in [N]$, and the related trace distance can be efficiently estimated with the sample complexity of $\mathcal{O}(\text{rank}^2(\phi(\boldsymbol{x}_j))/\epsilon^5)$, where ϵ represents the additive error [37].

3.4.2 Hamiltonian Baseline Kernel

This quantum autoencoder based kernel is compared against a baseline kernel constructed directly from Hamiltonian-encoded states without training. Specifically, suppose $\mathcal{D} = \{(\boldsymbol{x}_i)\}_{i=1}^N$ representing the flattened one-hot vectors, the Hamiltonian encoding method achieves the map $\boldsymbol{x}_i \mapsto |\psi(\boldsymbol{x}_i)\rangle = e^{-iH(\boldsymbol{x}_i)t}|0^n\rangle$ [38]. Here, we suppose the one-hot vector $\boldsymbol{x}_i \in \mathbb{R}^L$ represents a M-dimensional real vector, the involved local Hamiltonian $H(\boldsymbol{x}_i) = \sum_{s=1}^L \boldsymbol{x}_i(s)P_s$, and P_s represents a 'constant-weight' Pauli operator non-trivially acting on constant number of qubits. The quantum kernel function is defined as $k_H(\boldsymbol{x}_i, \boldsymbol{x}_j) = |\langle\psi(\boldsymbol{x}_i)|\psi(\boldsymbol{x}_j)\rangle|^2$, and it can be computed efficiently by using the SWAP-test method [18]..

In both cases, the resulting kernel matrices are used to train classical SVMs for downstream classification. The classification accuracy obtained with the Hamiltonian baseline serves as the reference point for measuring improvements achieved by the learned representations. This

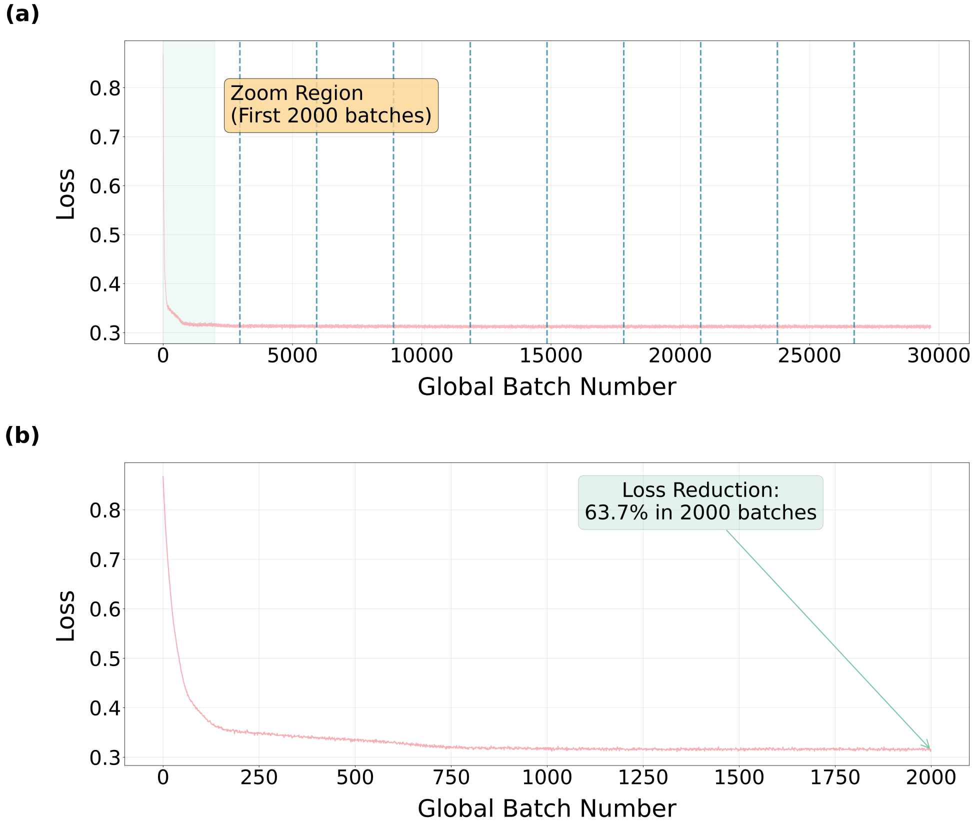

Figure 3: Training loss convergence during quantum autoencoder pre-training. (a) Complete training trajectory showing loss versus global batch number across all epochs, with the zoom region highlighted in green covering the first 2000 batches where rapid convergence occurs. Blue vertical dashed lines indicate epoch boundaries. (b) Detailed view of the first 2000 batches showing the rapid loss reduction, where the model achieves 63.7% loss reduction. The loss curve demonstrates fast initial convergence followed by stabilization.

procedure benchmarks whether unsupervised quantum representation learning can provide advantages over fixed encodings in peptide sequence classification.

4 Numerical Simulation Results

4.1 Quantum Autoencoders Training

Training of the quantum autoencoders converged rapidly across all 18 variants. Figure 3 shows the trajectory for one representative model. More than 60% of the total loss reduction occurs within the first 2000 batches, corresponding to less than a single epoch over the 3 million peptide corpus. After this initial phase, the training loss quickly stabilizes at a low value and remains consistent throughout the remainder of training.

Across all architectures, final training losses ranged from approximately 0.15 to 0.5, reflecting variation in model capacity and compression settings. Gradient norms remained stable throughout training, indicating that optimization was not affected by vanishing

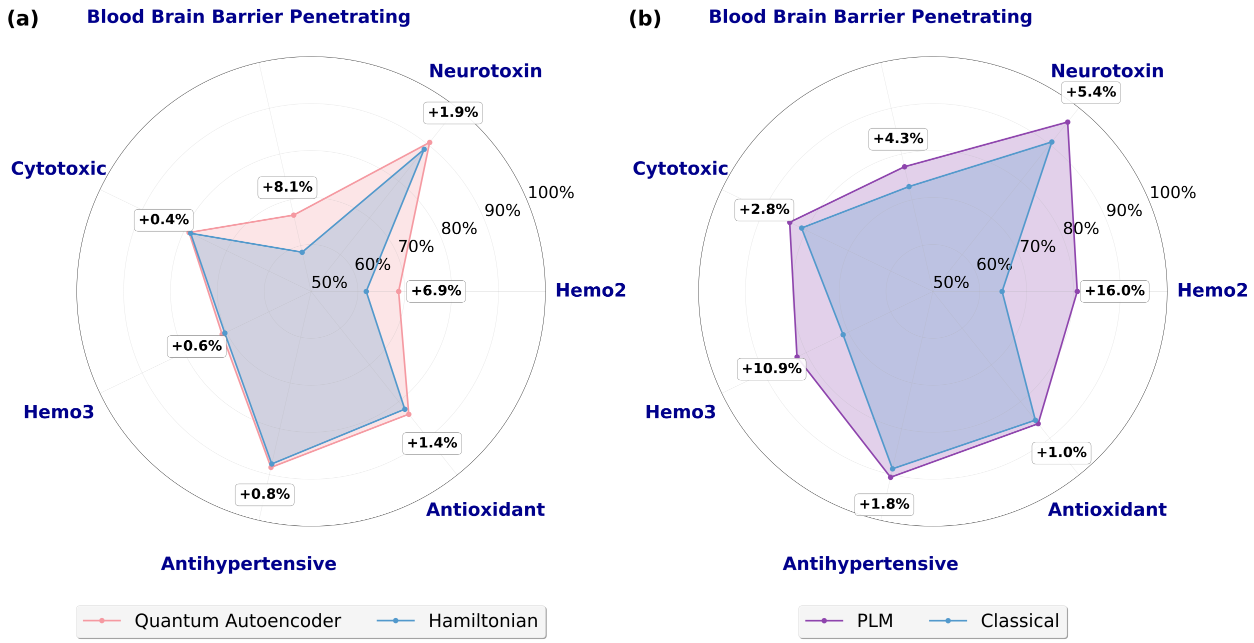

Figure 4: Classification accuracy comparison across seven peptide datasets. (a) Quantum autoencoder representations compared to Hamiltonian baselines. (b) PLM representations compared to classical baselines. The two comparisons use different baselines and are therefore not directly comparable, but both highlight the benefit of representation learning.

gradients. Notably, we did not observe the exponentially small gradient magnitudes that are characteristic of barren plateaus, even in deeper circuits with up to 30 layers. Overall, the convergence pattern was highly consistent: every quantum autoencoder reached a stable loss in the early stages of training. These results confirm that the quantum autoencoder can be trained efficiently at scale, enabling subsequent evaluation of learned representations on downstream peptide classification tasks.

4.2 Classification Results

We evaluated the learned quantum autoencoder representations on seven peptide classification tasks and compared them with the Hamiltonian baseline. In parallel, we evaluated PLMs against classical baselines. The results of these two comparisons are shown in Figure 4.

Following the pre-training setup, classification performance was evaluated using the best-performing quantum autoencoder variant for each dataset, selected by cross-validation. The downstream evaluation procedure for quantum kernels is described in Section 3.4. Relative to the Hamiltonian kernel baseline, the quantum autoencoder improved classification accuracy across all datasets, with an average gain of +2.9%. This indicates that unsupervised quantum autoencoder pre-training can capture non-trivial features beyond those present in the fixed Hamiltonian encoding.

For the classical experiments, the baseline was chosen as the best of five candidate classifiers trained directly on flattened one-hot encoding, while the PLM results corresponds to the best of 12 pre-trained models. PLMs achieved larger improvements, with an average gain of +6.0%. The representation pipelines and evaluation procedure for these experiments are described in Section 3.1.

Although the two radar plots are not directly comparable due to different baselines, both highlight the importance of representation learning. In the quantum case, quantum autoencoder representations outperform Hamiltonian encodings, while in the classical case, PLM representations outperform one-hot encoding. Together, these results establish the quantum autoencoder as a proof of principle of a quantum analogue of PLM-style pre-training.

At the same time, several limitations remain. The quantum autoencoder improvements, while consistent, are smaller than those achieved by PLMs, and all experiments were conducted on simulators without device noise. Moreover, the Hamiltonian encoding may not be optimal for long peptide sequences, suggesting room for alternative encoding strategies. Another limitation is that quantum evaluations were restricted to kernel-based SVMs. Unlike PLM or one-hot encodings, which output feature vectors that can be fed directly to diverse classifiers, the quantum autoencoder produces reduced density matrices as representations. Extracting feature vectors from mixed states would require full state tomography [39], which is prohibitively expensive for large systems [40]. Although classical shadow provides an efficient alternative for predicting local properties of quantum states [41], its applicability to long-range entanglement properties still requires an exponential amount of samples. For this reason, quantum kernel methods currently provide the most straightforward way to benchmark quantum representations.

5 Conclusion

We presented a quantum autoencoder approach for representation learning on peptide sequences. Quantum autoencoders were shown to train efficiently on a large-scale peptide corpus without encountering barren plateaus, and consistently improved classification performance over Hamiltonian baselines across seven datasets. These results demonstrate that unsupervised quantum pre-training can extract non-trivial features from biological sequences, providing a quantum analogue of representation learning. At the same time, the magnitude of improvement was smaller than that achieved by PLMs over classical baselines, highlighting both the feasibility and the current limitations of quantum representation learning. More broadly, much of the existing QML literature still relies on hard-coded, task-agnostic encoding strategies for mapping classical data into quantum states. Such fixed encodings often fail to adapt to the statistical structure of real-world data, limiting their downstream effectiveness. Our results suggest that learned quantum representations, enabled by quantum autoencoder pre-training, offer a promising alternative. Potential future work includes scaling quantum autoencoders to larger architectures, developing hybrid quantum-classical pipelines, and integrating with PLMs to combine the strengths of both paradigms. Together, these directions have strong potential to advance QML in bioinformatics applications.

References

[1] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.

[2] Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo, Hartmut Neven, and Jarrod R McClean. Power of data in quantum machine learning. *Nature communications*, 12(1):2631, 2021.

[3] Maria Schuld, Ryan Sweke, and Johannes Jakob Meyer. Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Physical Review A*, 103(3):032430, 2021.

[4] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. *Physical review letters*, 113(13):130503, 2014.

[5] Yusen Wu, Bujiao Wu, Jingbo Wang, and Xiao Yuan. Quantum phase recognition via quantum kernel methods. *Quantum*, 2023.

[6] Vojtěch Havlíček, Antonio D Córcoles, Kristan Temme, Aram W Harrow, Abhinav Kandala, Jerry M Chow, and Jay M Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209–212, 2019.

[7] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021.

[8] Pierre-Luc Dallaire-Demers and Nathan Killoran. Quantum generative adversarial networks. *Physical Review A*, 98(1):012324, 2018.

[9] Juan Carrasquilla, Giacomo Torlai, Roger G Melko, and Leandro Aolita. Reconstructing quantum states with generative models. *Nature Machine Intelligence*, 1(3):155–161, 2019.

[10] Xun Gao, Z-Y Zhang, and L-M Duan. A quantum machine learning algorithm based on generative models. *Science advances*, 4(12):eaat9004, 2018.

[11] Maria Schuld and Nathan Killoran. Quantum machine learning in feature hilbert spaces. *Physical review letters*, 122(4):040504, 2019.

[12] Mikko Mottonen, Juha J Vartiainen, Ville Bergholm, and Martti M Salomaa. Transformation of quantum states using uniformly controlled rotations. *arXiv preprint quant-ph/0407010*, 2004.

[13] Dave Wecker, Matthew B Hastings, and Matthias Troyer. Progress towards practical quantum variational algorithms. *Physical Review A*, 92(4):042303, 2015.

[14] Roeland Wiersema, Cunlu Zhou, Yvette de Sereville, Juan Felipe Carrasquilla, Yong Baek Kim, and Henry Yuen. Exploring entanglement and optimization within the hamiltonian variational ansatz. *PRX quantum*, 1(2):020319, 2020.

[15] Sam Gelman, Bryce Johnson, Chase R Freschlin, Arnav Sharma, Sameer D'Costa, John Peters, Anthony Gitter, and Philip A Romero. Biophysics-based protein language models for protein engineering. *Nature Methods*, pages 1–12, 2025.

[16] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[17] Jonathan Romero, Jonathan P Olson, and Alan Aspuru-Guzik. Quantum autoencoders for efficient compression of quantum data. *Quantum Science and Technology*, 2(4):045001, 2017.

[18] Harry Buhrman, Richard Cleve, John Watrous, and Ronald De Wolf. Quantum fingerprinting. *Physical review letters*, 87(16):167902, 2001.

[19] Dmytro Bondarenko and Polina Feldmann. Quantum autoencoders to denoise quantum data. *Physical review letters*, 124(13):130502, 2020.

[20] Alex Pepper, Nora Tischler, and Geoff J Pryde. Experimental realization of a quantum autoencoder: The compression of qutrits via machine learning. *Physical review letters*, 122(6):060501, 2019.

[21] Amir Khoshaman, Walter Vinci, Brandon Denis, Evgeny Andriyash, Hossein Sadeghi, and Mohammad H Amin. Quantum variational autoencoder. *Quantum Science and Technology*, 4(1):014001, 2018.

[22] Huanjin Wu, Xinyu Ye, and Junchi Yan. QVAE-mole: The quantum vae with spherical latent variable learning for 3-d molecule generation. *Advances in Neural Information Processing Systems*, 37:22745–22771, 2024.

[23] Yi Pan, Hanqi Jiang, Wei Ruan, Dajiang Zhu, Xiang Li, Yohannes Abate, Yingfeng Wang, and Tianming Liu. Molqae: Quantum autoencoder for molecular representation learning. *arXiv preprint arXiv:2505.01875*, 2025.

[24] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[25] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[26] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. Prottrans: towards cracking the language of life's code through self-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44:7112–7127, 2021.

[27] Ethan C Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M Church. Unified rational protein engineering with sequence-based deep representation learning. *Nature methods*, 16(12):1315–1322, 2019.

[28] Chakradhar Guntuboina, Adrita Das, Parisa Mollaei, Seongwon Kim, and Amir Barati Farimani. Peptidebert: A language model based on transformers for peptide property prediction. *The Journal of Physical Chemistry Letters*, 14(46):10427–10434, 2023.

[29] Hsin-Yuan Huang, Michael Broughton, Jordan Cotler, Sitan Chen, Jerry Li, Masoud Mohseni, Hartmut Neven, Ryan Babbush, Richard Kueng, John Preskill, et al. Quantum advantage in learning from experiments. *Science*, 376(6598):1182–1186, 2022.

[30] Hsin-Yuan Huang, Richard Kueng, and John Preskill. Information-theoretic bounds on quantum advantage in machine learning. *Physical Review Letters*, 126(19):190505, 2021.

[31] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[32] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[33] Peter McCullagh. *Generalized linear models*. Routledge, 2019.

[34] Jeongwan Haah, Matthew B Hastings, Robin Kothari, and Guang Hao Low. Quantum algorithm for simulating real time evolution of lattice hamiltonians. *SIAM Journal on Computing*, 52(6):FOCS18–250, 2021.

[35] Lennart Bittel and Martin Kliesch. Training variational quantum algorithms is np-hard. *Physical Review Letters*, 127(12):120502, 2021.

[36] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[37] Qisheng Wang and Zhicheng Zhang. Fast quantum algorithms for trace distance estimation. *IEEE Transactions on Information Theory*, 70(4):2720–2733, 2023.

[38] Shengxin Zhuang, John Tanner, Yusen Wu, Du Huynh, Wei Liu, Xavier Cadet, Nicolas Fontaine, Philippe Charton, Cedric Damour, Frederic Cadet, and Jingbo Wang. Nonhemolytic peptide classification using a quantum support vector machine. *Quantum Inf. Process.*, 23(11), November 2024.

[39] K Vogel and H Risken. Determination of quasiprobability distributions in terms of probability distributions for the rotated quadrature phase. *Physical Review A*, 40(5):2847, 1989.

[40] Jeongwan Haah, Aram W Harrow, Zhengfeng Ji, Xiaodi Wu, and Nengkun Yu. Sample-optimal tomography of quantum states. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 913–925, 2016.

[41] Hsin-Yuan Huang, Richard Kueng, and John Preskill. Predicting many properties of a quantum system from very few measurements. *Nature Physics*, 16(10):1050–1057, 2020.

[42] Armando Angrisani, Alexander Schmidhuber, Manuel S Rudolph, M Cerezo, Zoë Holmes, and Hsin-Yuan Huang. Classically estimating observables of noiseless quantum circuits. *arXiv preprint arXiv:2409.01706*, 2024.

[43] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

[44] Ahmed Elnaggar, Hazem Essam, Wafaa Salah-Eldin, Walid Moustafa, Mohamed Elkerdawy, Charlotte Rochereau, and Burkhard Rost. Ankh: Optimized protein language model unlocks general-purpose modelling. *arXiv preprint arXiv:2301.06568*, 2023.

[45] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021.

[46] Uniprot: the universal protein knowledgebase in 2025. *Nucleic acids research*, 53(D1):D609–D617, 2025.

[47] Gabriel Cabas-Mora, Anamaría Daza, Nicole Soto-García, Valentina Garrido, Diego Alvarez, Marcelo Navarrete, Lindybeth Sarmiento-Varón, Julieta H Sepúlveda Yañez, Mehdi D Davari, Frederic Cadet, et al. Peptipedia v2. 0: A peptide sequence database and user-friendly web platform. a major update. *Database*, 2024:baae113, 2024.

[48] Wei-Chao Lin, Chih-Fong Tsai, Ya-Han Hu, and Jing-Shang Jhang. Clustering-based undersampling in class-imbalanced data. *Information Sciences*, 409:17–26, 2017.

[49] Qingping Liang, Zhemin Liu, Mengshi Xiao, Wei Zhou, Dongxing Yu, Jiaming Liu, Le Wang, and Haijin Mou. Improving activity of ace inhibitory peptide by sequence-based rational design: Properties, function on endothelial cells, and potential antihypertensive mechanism. *Food Frontiers*, 2025.

[50] Natalia Ichim, Francisco Marín, and Esteban Orenes-Piñero. Potential impact of bioactive peptides from foods in the treatment of hypertension. *Molecular Nutrition & Food Research*, 68(14):2400084, 2024.

[51] Tang-Bin Zou, Tai-Ping He, Hua-Bin Li, Huan-Wen Tang, and En-Qin Xia. The structure-activity relationship of the antioxidant peptides from natural proteins. *Molecules*, 21(1):72, 2016.

[52] Hong-Yu Zhang, Hui-Zhen Li, Tian-Wei Zhang, and Zhi-Jun Zhang. Research progress on the mechanism of antioxidant peptides. 2022.

[53] Yiyun Zhu, Kang Wang, Xinyi Jia, Caili Fu, Haining Yu, and Yipeng Wang. Antioxidant peptides, the guardian of life from oxidative stress. *Medicinal Research Reviews*, 44(1):275–364, 2024.

[54] Vinod Kumar, Sumeet Patiyal, Rajesh Kumar, Sukriti Sahai, Dilraj Kaur, Anjali Lathwal, and Gajendra PS Raghava. B3pdb: an archive of blood–brain barrier-penetrating peptides. *Brain Structure and Function*, 226(8):2489–2495, 2021.

[55] Zhi-Feng Gu, Yu-Duo Hao, Tian-Yu Wang, Pei-Ling Cai, Yang Zhang, Ke-Jun Deng, Hao Lin, and Hao Lv. Prediction of blood–brain barrier penetrating peptides based on data augmentation with augur. *BMC biology*, 22(1):86, 2024.

[56] Xin Luan, Ye Wu, Yi-Wen Shen, Hong Zhang, Yu-Dong Zhou, Hong-Zhuan Chen, Dale G Nagle, and Wei-Dong Zhang. Cytotoxic and antitumor peptides as novel chemotherapeutics. *Natural product reports*, 38(1):7–17, 2021.

[57] Diana Rafieezadeh and Goli Esfandyari. Marine bioactive peptides with anticancer potential, a narrative review. *International Journal of Biochemistry and Molecular Biology*, 15(4):118, 2024.

[58] Ankur Gautam, Kumardeep Chaudhary, Sandeep Singh, Anshika Joshi, Priya Anand, Abhishek Tuknait, Deepika Mathur, Grish C Varshney, and Gajendra PS Raghava. Hemolytik: a database of experimentally determined hemolytic and non-hemolytic peptides. *Nucleic acids research*, 42(D1):D444–D449, 2014.

[59] Florence Jungo, Lydie Bougueleret, Ioannis Xenarios, and Sylvain Poux. The uniprotkb/swiss-prot tox-prot program: a central hub of integrated venom protein data. *Toxicon*, 60(4):551–557, 2012.

[60] Malak Pirtskhalava, Andrei Gabrielian, Phillip Cruz, Hannah L Griggs, R Burke Squires, Darrell E Hurt, Maia Grigolava, Mindia Chubinidze, George Gogoladze, Boris Vishnepolsky, et al. Dbaasp v. 2: an enhanced database of structure and antimicrobial/cytotoxic activity of natural and synthetic peptides. *Nucleic acids research*, 44(D1):D1104–D1112, 2016.

[61] Xi Zhou, Haiyi Chen, Shuijiao Peng, Yuxin Si, Gaoang Wang, Li Yang, Qing Zhou, Minjuan Lu, Qiaoling Xie, Xi He, et al. Inherent fast inactivation particle of nav channels as a new binding site for a neurotoxin. *The EMBO Journal*, 44(11):3180–3209, 2025.

[62] Byungjo Lee, Min Kyoung Shin, In-Wook Hwang, Junghyun Jung, Yu Jeong Shim, Go Woon Kim, Seung Tae Kim, Wonhee Jang, and Jung-Suk Sung. A deep learning approach with data augmentation to predict novel spider neurotoxic peptides. *International journal of molecular sciences*, 22(22):12291, 2021.

[63] Asmita Yadav, Damini Pandey, Ghulam Md Ashraf, and Rachana. Peptide based therapy for neurological disorders. *Current Protein and Peptide Science*, 22(9):656–665, 2021.

A Variational Circuit depth complexity of Quantum Autoencoder

Consider the initial quantum feature state $|\psi(\boldsymbol{x}_i)\rangle = e^{-iH(\boldsymbol{x}_i)t}|0^n\rangle$ with the local Hamiltonian matrix $H(\boldsymbol{x}_i) = \sum_{s=1}^{L} \boldsymbol{x}_i(s)P_s$, where P_s represents a 'constant-weight' Pauli operator non-trivially acting on constant number of qubits. The essential idea of quantum autoencoder aims to train a quantum circuit $U(\vec{\boldsymbol{\theta}})$, enabling

$$U(\vec{\boldsymbol{\theta}})\left(\mathbb{E}_{\boldsymbol{x}_i\sim\mathcal{D}}\left[|\psi(\boldsymbol{x}_i)\rangle\langle\psi(\boldsymbol{x}_i)|\right]\right)U^\dagger(\vec{\boldsymbol{\theta}}) \approx |0^m\rangle\langle 0^m| \otimes \sigma, \tag{5}$$

where σ represents a $(n-m)$-qubit density matrix, and the number of decoupled qubits m is a constant. In the main file, we suggested that the quantum autoencoder circuit $U(\vec{\boldsymbol{\theta}})$ can be trained via minimizing a loss function. We first evaluate the necessary quantum circuit depth of $U(\vec{\boldsymbol{\theta}})$.

Theorem 3 (Theorem 1 in the main text)**.** *Given the Hamiltonian encoding method $|\psi(\boldsymbol{x}_i)\rangle = e^{-iH(\boldsymbol{x}_i)t}|0^n\rangle$, where the Hamiltonian $H(\boldsymbol{x}_i) = \sum_{s=1}^{L} \boldsymbol{x}_i(s)P_s$, with P_s representing a 'constant-weight' Pauli operator non-trivially acting on constant number of qubits. Then the quantum circuit depth of $U(\vec{\boldsymbol{\theta}})$ satisfies $d \leq \operatorname{poly}\log(nt/\epsilon)$ sufficing to implement the quantum autoencoder task.*

Proof. The fundamental idea within quantum autoencoder is to approximately decouple m qubits from the whole quantum system. Without loss of generality, we assume the Hamiltonian $H(\boldsymbol{x}_i)$ is defined on a constant-dimensional lattice, and $\max\{\|\boldsymbol{x}_i\|_\infty |t|\}$ is a constant which does not increase with the system size. According to the Lieb-Robinson bound [34], the required quantum circuit depth for the quantum feature state $|\psi(\boldsymbol{x}_i)\rangle$ is $\operatorname{poly}\log(nt/\epsilon)$. On other hand, we label the target qubits to be trashed as $Q = \{q_1, \cdots, q_m\}$. Since the information only spreads within the light-cone, as a result, a quantum circuit $U(\vec{\boldsymbol{\theta}})$ with circuit depth $d = \operatorname{poly}\log(nt/\epsilon)$ suffices to decouple qubit set Q back to $|0^m\rangle$. □

B Sample Complexity of Quantum autoencoder

Let the density matrix $\rho_{\mathcal{D}} = \mathbb{E}_{\boldsymbol{x}_i\sim\mathcal{D}}\left(|\psi(\boldsymbol{x}_i)\rangle\langle\psi(\boldsymbol{x}_i)|\right)$, then one can rewrite the loss function

$$\mathcal{L}(\vec{\boldsymbol{\theta}}) = \operatorname{Tr}\left[\Pi_m U(\vec{\boldsymbol{\theta}})\rho_{\mathcal{D}}U^\dagger(\vec{\boldsymbol{\theta}})\right] = \operatorname{Tr}\left[U^\dagger(\vec{\boldsymbol{\theta}})\Pi_m U(\vec{\boldsymbol{\theta}})\rho_{\mathcal{D}}\right]. \tag{6}$$

Here, we aim to answer the question "*How many samples to $\rho_{\mathcal{D}}$ suffices to estimate $\mathcal{L}(\vec{\boldsymbol{\theta}})$ within ϵ additive error for all candidate $\vec{\boldsymbol{\theta}}$*," by studying the support size of the operator $U^\dagger(\vec{\boldsymbol{\theta}})\Pi_m U(\vec{\boldsymbol{\theta}})$. We first give the main theoretical result, then detail the proof.

Theorem 4 (Theorem 2 in the main text)**.** *For any quantum autoencoder task with loss function $\mathcal{L}(\vec{\boldsymbol{\theta}})$ (defined as Eq.(1)), then*

$$N = \mathcal{O}\left(\frac{3^{\log(1/\epsilon\delta)}\log(1/\epsilon\delta)\log(n)}{\epsilon^2}\right) \tag{7}$$

samples drawn from $\rho_{\mathcal{D}}$ suffice to estimate $\mathcal{L}(\vec{\boldsymbol{\theta}})$ throughout the entire training process, with success probability $\geq 1-\delta$. In other words, N-samples are sufficient to guarantee an ϵ-approximation to $\mathcal{L}(\vec{\boldsymbol{\theta}})$ for all candidate variational parameters $\vec{\boldsymbol{\theta}}$.

Proof. Suppose the variational quantum circuit

$$U(\vec{\boldsymbol{\theta}}) = U_d(\boldsymbol{\theta}_d)U(\boldsymbol{\theta}_{d-1})\cdots U_1(\boldsymbol{\theta}_1), \tag{8}$$

where each $U(\boldsymbol{\theta}_l)$ represents a layer of two-qubit gates $U_{j,l}$ for $l \in [d]$. Without loss of generality, we assume the distribution of each two-qubit gate $U_{j,l}$ is "locally scrambling", that is invariant under single-qubit rotations. This property is satisfied by a wide range of deep and shallow unstructured parameterised quantum circuits.

The proof is based on the backward propagation in the context of Heisenberg picture of the operator $O_{U(\vec{\boldsymbol{\theta}})} = U^{\dagger}(\vec{\boldsymbol{\theta}})\Pi_m U(\vec{\boldsymbol{\theta}})$. For the last layer of $U(\vec{\boldsymbol{\theta}})$, since $|\mathrm{supp}(\Pi_m)| = \mathcal{O}(1)$, and $U_d(\boldsymbol{\theta}_1)$ is a layer of two-qubit gates, it is straightforward to show that

$$O_d = U_d^{\dagger}(\boldsymbol{\theta}_1)\Pi_m U_d(\boldsymbol{\theta}_1) = \sum_{P\in\{I,X,Y,Z\}^{\otimes n},\ |P|\leq m} \mathrm{Tr}[PU_d^{\dagger}(\boldsymbol{\theta}_1)\Pi_m U_d(\boldsymbol{\theta}_1)]P/2^n. \tag{9}$$

Then for index $l \in \{d-1, d-2, \cdots, 1\}$, let the operator

$$O_j = \frac{1}{2^n}\sum_{P\in\{I,X,Y,Z\}^{\otimes n},\ |P|\leq k} \mathrm{Tr}[U_j^{\dagger}(\boldsymbol{\theta}_j)O_{j-1}U_j(\boldsymbol{\theta}_j)P]P, \tag{10}$$

until the final truncated observable

$$O_{U(\vec{\boldsymbol{\theta}})}^{(k)} = U_1^{\dagger}(\boldsymbol{\theta}_1)O_1U_1(\boldsymbol{\theta}_1). \tag{11}$$

According to the Lemma 11 in the Ref. [42], it is shown that

$$\mathbb{E}_{U(\vec{\boldsymbol{\theta}})}\left[\left|\mathrm{Tr}\left[\left(O_{U(\vec{\boldsymbol{\theta}})} - O_{U(\vec{\boldsymbol{\theta}})}^{(k)}\right)\rho\right]\right|\right] \leq \left(\frac{2}{3}\right)^{(k+1)/2} \|\Pi_m\|_{\mathrm{Pauli},2} \tag{12}$$

for any valid quantum state ρ, where the Pauli-2 norm is defined as

$$\|\Pi_m\|_{\mathrm{Pauli},2} = \left(\sum_P |\mathrm{Tr}[\Pi_m P]|^2\right)^{1/2}. \tag{13}$$

In our case, the operator Π_m non-trivially acts on constant number of qubits, resulting in $\|\Pi_m\|_{\mathrm{Pauli},2} = \mathcal{O}(1)$. Let $k = \mathcal{O}(\log(1/\epsilon\delta))$, the above result immediately yields

$$\left|\mathrm{Tr}\left[\left(O_{U(\vec{\boldsymbol{\theta}})} - O_{U(\vec{\boldsymbol{\theta}})}^{(k)}\right)\rho\right]\right| \leq \epsilon \tag{14}$$

for any variational quantum circuit $U(\vec{\boldsymbol{\theta}})$ except for a small fraction δ.

Using this observation, we can further rewrite the quantum autoencoder loss function as

$$\mathcal{L}'(\vec{\boldsymbol{\theta}}) = \mathrm{Tr}\left[O_{U(\vec{\boldsymbol{\theta}})}^{(k)}\rho_{\mathcal{D}}\right], \tag{15}$$

where the operator $O_{U(\vec{\boldsymbol{\theta}})}^{(k)} = \sum_{|P|\leq k}\alpha_P P$, which contains $M = \mathcal{O}(n^k)$ local Pauli terms. By leveraging the classical shadow method [41], it is shown that $N = \mathcal{O}\left(3^{\max_P\{|\mathrm{supp}(P)|\}}\log(M)\epsilon^{-2}\right)$ samples suffice to estimate $\mathcal{L}'(\vec{\boldsymbol{\theta}})$. This completes the proof. □

C Protein Language Models Used

Table 1 lists all PLMs considered in this work, along with their representation dimensionalities.

Model	Representation Dimension
ESM-2 (t6_8M) [25]	320
ESM-2 (t12_35M) [25]	480
ESM-2 (t30_150M) [25]	640
PeptideBERT [28]	768
BioBERT [43]	768
Ankh-base [44]	768
ProtBERT [26]	1024
ProtBERT-BFD [26]	1024
ProtT5-XL [26]	1024
ESM-2 (t33_650M) [25]	1280
ESM-1b [45]	1280
Ankh-large [44]	1536

Table 1: List of pre-trained PLMs used in this study.

D Datasets

D.1 Pre-Training Dataset

UniProt Peptide Dataset. We derived our dataset from the UniProt Knowledgebase (UniProtKB) [46], which provides a comprehensive, curated, and regularly updated repository of protein sequences from diverse species. Using the UniProt search interface, we extracted 3,062,374 unique peptide sequences with lengths between 4 and 50 amino acids. Both Swiss-Prot (reviewed) and TrEMBL (unreviewed) entries were included, ensuring coverage of a wide range of species, biological functions, and environmental contexts. The query was performed on UniProt [14 March 2024], applying sequence-length filters to restrict results to the peptidome. Redundant entries were removed, and only sequences within the defined length window were retained. This dataset thus provides a broad and representative sample of the global peptidome, which we used as the basis for unsupervised training of our quantum autoencoder models.

D.2 Classification Datasets

Data collection process. All datasets were collected from the Peptipedia v2.0 database using its query system for biological activities[47]. Peptides were selected based on the desired function and further filtered to exclude sequences containing non-canonical residues or chemical modifications. Additional constraints were applied to retain only experimentally validated peptides with a maximum length of 100 residues. For the generation of negative

Table 2: Dataset statistics and preprocessing summary. All datasets are nearly balanced after preprocessing. Train/test splits show the actual positive and negative class counts.

Dataset	**#Sequences**	**Train (Pos/Neg)**	**Test (Pos/Neg)**
Antihypertensive	2,778	1,135 / 809	487 / 347
Antioxidant	1,933	745 / 608	319 / 261
Blood-Brain Barrier-Penetrating	650	248 / 207	107 / 88
Cytotoxic	2,651	1,017 / 838	436 / 360
Hemo 2	992	435 / 359	108 / 90
Hemo 3	1,587	692 / 577	173 / 145
Neurotoxin	1033	344 / 379	147 / 163

examples, the Peptipedia v2.0 database was also used to select peptides lacking the target activity, applying the same filtering criteria to ensure consistency across datasets[47].

Dataset preparation. To prepare the datasets for the training process, a homology reduction step was performed using CD-Hit [14]. Each dataset was first divided into positive and negative examples. For both subsets, CD-Hit was applied with a 90% sequence identity threshold to reduce redundancy. Subsequently, undersampling strategies were implemented to balance the datasets by adjusting the number of negative examples to match the positive ones [48]. To achieve this, sequences with reduced homology were randomly selected from the clusters generated by CD-Hit, ensuring a balanced representation across clusters.

Preprocessing. For each downstream dataset, we examined sequence length distributions using histograms. We observed that beyond a certain length threshold, all sequences were sparsely represented, so sequences longer than this threshold were removed (e.g., sequences longer than 65 amino acids in one dataset). The threshold varied across datasets and was chosen manually. After filtering, sequences were one-hot encoded, shorter sequences were padded to match the maximum sequence length within the dataset, and the resulting matrices were flattened into binary vectors for input to the models.

Antihypertensive. Antihypertensive peptides are short amino acid sequences that contribute to lowering blood pressure by modulating key biological and cellular processes involved in cardiovascular regulation [49]. Their activity is often linked to the inhibition of enzymes such as angiotensin-converting enzyme (ACE) or the modulation of signaling pathways that influence vascular tone and fluid balance [11]. Beyond direct blood pressure control, these peptides may also mitigate hypertension-associated inflammation and immune responses, thereby reducing the risk of long-term cardiovascular damage [50].

Antioxidant. Antioxidant peptides are short amino acid sequences with inherent bioactivity that protect cells and tissues from oxidative damage by counteracting reactive oxygen species (ROS) and other prooxidants [51]. Their activity arises from multiple mechanisms, including direct radical scavenging, inhibition of lipid peroxidation, and chelation of transition metals that catalyze oxidative reactions [52]. By mitigating oxidative stress, these peptides contribute to the preservation of cellular integrity and metabolic balance, which is essential for overall health and the prevention of chronic diseases linked to oxidative damage [53].

Blood-brain barrier-penetrating. Blood–brain barrier–penetrating peptides (BBBPps) are short amino acid sequences specifically designed or discovered to overcome the highly selective nature of the blood–brain barrier (BBB), a physiological shield that protects the central nervous system while severely limiting drug entry [54]. By interacting with receptors or transport systems on the endothelial cells that form the BBB, these peptides act as molecular shuttles that enable therapeutic or diagnostic molecules to reach the brain [55]. Their sequences often feature amphipathic or positively charged residues that promote membrane interactions, and they can be chemically optimized to resist enzymatic degradation in circulation [54].

Cytotoxic. Cytotoxic peptides are short amino acid sequences that induce cell death by directly interacting with and disrupting cell membranes, compromising their structural integrity [56]. This membrane-targeting mechanism allows them to bypass intracellular resistance pathways and makes them particularly effective against rapidly dividing or abnormal cells, such as cancer cells [56]. Because of their potent ability to trigger apoptosis or necrosis, cytotoxic peptides have been widely investigated as therapeutic agents in oncology, where they offer a promising alternative or complement to conventional chemotherapies [57].

Hemo 2 & Hemo 3. HemoPI-2 and HemoPI-3 datasets are compilations of experimentally validated peptides labeled as hemolytic or non-hemolytic which have been extracted from various sources, including the Hemolytik database [58], Swiss-Prot [59] (a curated protein sequence database which is part of UniProt [46]) and the Database of Antimicrobial Activity and Structure of Peptides (DBAASP) [60]. Hemolytic peptides are short amino acid sequences that have the ability to disrupt red blood cell membranes and cause cell lysis, whereas non-hemolytic peptides lack such activity and are considered safer for therapeutic applications.

Neurotoxin. A neurotoxin peptide is a short amino acid sequence that exerts toxic effects on the nervous system by disrupting neuronal communication and function. These peptides typically act by binding to and modulating ion channels, receptors, or neurotransmitter release machinery, leading to altered excitability, impaired synaptic transmission, or cell death [61]. Many are naturally produced by venomous organisms such as cone snails, scorpions, or spiders as a means of predation or defense, where their high potency and specificity allow them to immobilize prey or deter predators [62]. While inherently harmful, neurotoxin peptides are also valuable molecular tools for probing the physiology of neuronal signaling and are being explored as scaffolds for therapeutic applications, where their precision in targeting neural pathways can be redirected for beneficial outcomes [63].