Письма в ЖЭТФ, том 122, вып. 11, с. 774 – 781

Обучение квантовых рекуррентных нейронных сетей на джозефсоновской интегральной схеме

*С. С. Самарин[+*1)], А. Е. Толстобров[+*], Ш. В. Сандуляну[*×], Г. П. Фёдоров[*×], А. А. Рябов[+], В. В. Вановский[+*], Д. А. Калачева[*+], А. Н. Болгар[*], В. Б. Лубсанов[*+], Ш. В. Кадырметов[*], А. М. Муравьев[*], Е. С. Алексеева[*], П. Ю. Шлыков[+*], А. М. Еремеев[*], А. В. Васенин[*+], А. Ю. Дмитриев[*×], О. В. Астафьев[+*×]*

[+] *Сколковский институт науки и технологий, 121205 Москва, Россия*

[*] *Московский физико-технический институт, 141700 Долгопрудный, Россия*

[×] *Институт радиотехники и электроники им. В. А. Котельникова РАН, 125009 Москва, Россия*

Поступила в редакцию 28 октября 2025 г.
После переработки 28 октября 2025 г.
Принята к публикации 29 октября 2025 г.

Данная работа посвящена решению задачи обработки числовых последовательностей при помощи вариационных квантовых алгоритмов, реализуемых на шумном квантовом компьютере среднего масштаба. Платформой для квантовых вычислений является интегральная схема на основе сверхпроводящих искусственных атомов. Архитектура квантовой рекуррентной нейронной сети сконструирована с использованием одно- и двухкубитных операций. На этапе эмулирования была изучена обучаемость модели в зависимости от числа кубитов, а также объема и метода кодирования данных в состояние кубитов. Сравнение с классическими архитектурами показало, что на современных квантовых процессорах возможно достичь качества предсказаний в выбранной задаче машинного обучения, сравнимого с качеством реализуемых на классических процессорах моделей.

DOI: 10.31857/S0370274X25120114

1. Введение. Квантовое машинное обучение (*quantum machine learning*, QML) – это развивающаяся междисциплинарная область, соединяющая вариационные квантовые алгоритмы и классическое машинное обучение. Решаются задачи классификации, регрессии, прогнозирования, приготовления состояния. В работах [1–3] представлены квантовые аналоги таких классических алгоритмов, как метод опорных векторов, k ближайших соседей, а также нейронные сети от примитивных до сверточных, рекуррентных и трансформеров. Было показано, что маломасштабные квантовые нейросети могут превосходить классические аналоги [4–6].

Однако реализация большинства из них на существующих сегодня шумных квантовых устройствах промежуточного масштаба (*noisy intermediate-scale quantum*, NISQ) либо невозможна, либо требует огромных временных и ресурсных затрат. Это связано с техническими сложностями реализации процессов управления и измерения кубитов, кодирования данных, зависящих от физических особенностей установок. Источниками ошибок в квантовых вычислениях служат: декогеренция (потеря информации, закодированной в состояние кубитов, из-за взаимодействия с окружающей средой), неточности выполнения квантовых операций и считывания состояния системы. Тем не менее, алгоритмы, построенные из параметризованных квантовых цепочек (*parametrized quantum circuit*, PQC) [7, 8] являются одним из самых перспективных направлений в QML, так как их дизайн может быть адаптирован под конкретное квантовое устройство в соответствии с его возможностями.

На сегодняшний день среди множества физических платформ для квантовых вычислений наиболее популярны сверхпроводниковые искусственные атомы, нейтральные атомы, ионы в ловушках и фотоны [9–15]. Сверхпроводниковые интегральные схемы обладают высокой чувствительностью к внешним шумам, однако на их основе удается реализовать операции над кубитами с длительностью всего несколько десятков наносекунд при сохранении высокой точности (fidelity) $> 99.9\,\%$ [16, 17]. Алгоритмы, использующие PQC, оказываются наиболее устойчивы к ошибкам, так как оптимизация параметров происходит на основе зашумленных измерений [18, 19].

2. Квантовая рекуррентная нейронная сеть. В данной работе квантовая рекуррентная нейронная

1)e-mail: samarin.ss@phystech.edu

сеть (quantum recurrent neural network, QRNN) используется для решения задачи обработки временных рядов, представленных в виде числовых последовательностей $\{x_i\}_{i=1}^{L}$. Модель предсказывает следующий элемент последовательности по предшествующим T элементам. Далее следует описание процессов обработки данных и обучения модели.

Сначала значения ряда приводятся к диапазону $[-1, 1]$ по формуле $x_{\text{scaled}} = \dfrac{x - x_{\min}}{x_{\max} - x_{\min}}$, затем они разделяются на тренировочный и тестовый сегменты (см. "Приложение. Используемые данные"). Каждый элемент данных для обучения представляет собой пару из подпоследовательности длины T и следующего за ней элемента, который необходимо предсказать, $(\{x_{i+1}, x_{i+2}, ..., x_{i+T}\}, x_{i+T+1})$, где $i = 0, 1, ..., L_{\text{train}} - T - 1$. Каждая подпоследовательность далее преобразуется в $\{\xi_i\}_{i=1}^{\tau}$ сверткой методом скользящего окна, после чего эти значения поочередно поступают на вход квантовой цепочке (см. рис. 1a). По окончании обработки всех ξ_i производится считывание состояния одного из кубитов, результат которого будет предсказанием модели

$$\tilde{x}_{T+1}(x_1, x_2, ..., x_T, \boldsymbol{\theta}) = \langle \sigma_z^{(1)}(x_1, x_2, ..., x_T, \boldsymbol{\theta}) \rangle = \\ = \mathrm{Tr}\Big[\rho_\tau(x_1, x_2, ..., x_T, \boldsymbol{\theta})\, \sigma_z^{(1)}\Big],$$

где $\boldsymbol{\theta}$ – все обучаемые параметры модели, ρ_τ — матрица плотности кубитов цепочки после обработки всех ξ_i. Во время эмулирования на классическом компьютере $\langle \sigma_z^{(1)} \rangle$ вычисляется аналитически. На квантовом процессоре это значение вычисляется усреднением многократных запусков и измерений квантовой цепочки путем дисперсионного считывания состояний кубитов [20] при фиксированных обучаемых параметрах.

Рассматриваемые нами квантовые цепочки представляют из себя набор послойно применяющихся однокубитных и двухкубитных операций. Они различаются количеством кубитов в системе и слоев (глубиной), набором исполняемых операций (гейтов) [21, 22]. Мы используем однокубитные $R_x(\varphi) = e^{-i\varphi\sigma_x/2}$, $R_y(\varphi) = e^{-i\varphi\sigma_y/2}$ и двухкубитные $\mathrm{iSWAP} = e^{-i\frac{\pi}{4}(\sigma_x \otimes \sigma_x + \sigma_y \otimes \sigma_y)}$ операции, которые формируют блоки квантовых операций (см. рис. 1b).

На первом слое классические данные кодируются в состояние квантовой системы. Идейно существует несколько способов кодирования [18]. В этой работе используется кодирование значений временного ряда ξ_t в состояние кубита через угол поворота $R_x(\xi_t, \boldsymbol{\gamma}) = R_x(\gamma_1 \cdot \xi_t + \gamma_2)$, γ_1 и γ_2 – обучаемые параметры. Данные кодируются в половину кубитов системы (регистровые кубиты). Перед началом выполнения очередного блока целесообразно сбрасывать регистровые кубиты в основное состояние $|0\rangle$. Вторая половина отвечает за "память" цепочки: после запутывания с первой половиной состояние этих кубитов будет также нести информацию о закодированных данных [23]. После кодирования следуют слои R_x и R_y поворотов с обучаемыми параметрами $\boldsymbol{\beta}$. Следом применяется закольцованная "лестница" iSWAP, которая запутывает состояния отдельных кубитов между собой, что позволяет представлять более сложные нелокальные корреляции между кубитами и расширяет класс достижимых унитарных преобразований цепочки. Среди всех возможных двухкубитных гейтов iSWAP обладает максимально возможной запутывающей способностью наравне с CNOT [24]. На сверхпроводниковых платформах iSWAP выполняется в процессе обмена энергией между кубитами, приведенными в частотный резонанс потоковым импульсом на определенное время, зависящее от энергии связи между ними [17].

Процесс кодирования ξ_t, поворотов и запутывания кубитов затем повторяется вновь $n_{\text{reuploadings}}$ раз. В недавних исследованиях было показано, что многократная перезагрузка одних и тех же данных (*reuploading*) значительно повышает описательную способность модели [15, 25–27], что послужило основанием для применения данного метода в нашей работе. После блока квантовых операций $S(\xi_t)$ матрица плотности N кубитной системы преобразуется к виду

$$\rho_t = S(\xi_t)\rho_{t-1}S^\dagger(\xi_t).$$

Обучение QRNN происходит путем обновления параметров $\boldsymbol{\theta}$ методом стохастического градиентного спуска (SGD) со среднеквадратической ошибкой в качестве оптимизируемой функции потерь

$$\mathcal{L}(\tilde{\boldsymbol{x}}(\boldsymbol{\theta}), \boldsymbol{x}) = \frac{\sum_{i=T+1}^{L} \left(\tilde{x}_i(\boldsymbol{\theta}) - x_i\right)^2}{L}. \tag{1}$$

Для минимизации $\mathcal{L}$ используется *adaptive momentum optimizer*. В квантовых цепочках градиент по обучаемым параметрам находится по *правилу сдвига параметров (parameter-shift rule)* [28, 29]. Согласно этому правилу, для вычисления частной производной по одному из параметров θ_j необходимо выполнить два измерения дифференцируемой величины с углами $\boldsymbol{\theta} \pm \frac{\pi}{2} e_j$

$$\frac{\partial \langle A_k \rangle_{\boldsymbol{\theta}}}{\partial \theta_j} = \frac{\langle A_k \rangle_{\boldsymbol{\theta} + \frac{\pi}{2} e_j} - \langle A_k \rangle_{\boldsymbol{\theta} - \frac{\pi}{2} e_j}}{2}. \tag{2}$$

Число обучаемых параметров QRNN в зависимости

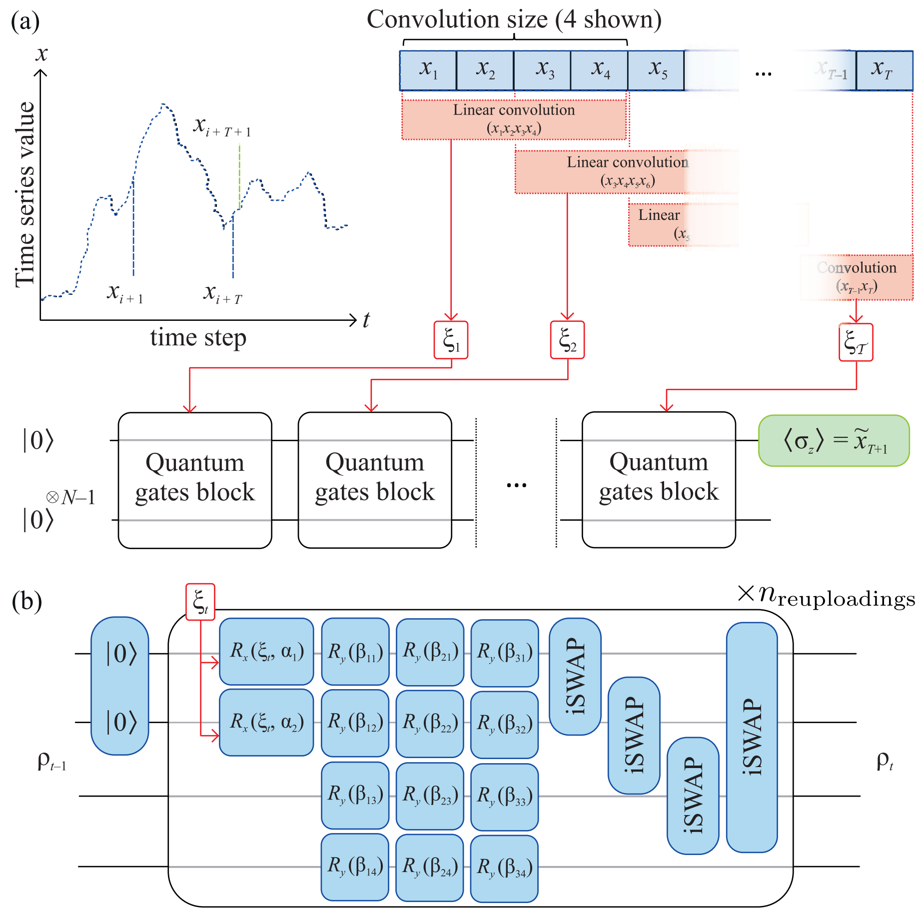

Рис. 1. (Цветной онлайн) Алгоритм QRNN. (a) – Преобразование участка временного ряда $\{x_i\}_{i=1}^{T}$ посредством свертки с обучаемыми параметрами в ряд $\{\xi_i\}_{i=1}^{\tau}$ и его последующая обработка в квантовой цепочке. Измеряемая $\langle\sigma_z^{(1)}\rangle$ является предсказанием $\tilde{x}_{T+1}$ следующего значения исходного временного ряда. (b) – Блок квантовых операций $S(\xi_t)$ (показан для $N=4$ кубитов). Регистровые кубиты приводятся в основное состояние $|0\rangle$. Затем кодирование ξ_t, параметризованные однокубитные повороты и запутывающие iSWAP применяются $n_{\text{reuploadings}}$ раз, преобразуя ρ_{t-1} к ρ_t

от значений гиперпараметров расчитывается по формуле:

$$n_{\text{parameters}} = \tau\cdot(\text{convolution size}+1)+ \\ +n_{\text{reuploadings}}\cdot N\cdot(n_{\text{features}}+3)+3, \qquad (3)$$

где $\tau = \dfrac{2T}{\text{convolution size}} - 1$ – длина вектора классических данных на входе квантовой цепочки, convolution size – ширина окна свертки, а n_{features} – размерность вектора x_i (в этой работе $n_{\text{features}}=1$).

3. Сверхпроводниковый квантовый процессор. Экспериментальное устройство содержит множество искусственных атомов – трансмонов [30], взаимодействующих с соседями (см. рис. 2). Каждый трансмон оснащен индивидуальным измерительным резонатором и совмещенной линией управления для изменения проходящего через сверхпроводящее квантовое интерференционное устройство (SQUID) трансмона магнитного потока и для его возбуждения микроволновым излучением. Для обучения QRNN использовалось четыре кубита (см. табл. 1).

Таблица 1. Параметры трансмонов, используемых в запуске QRNN

Transmon	1	2	3	4
$\omega_{ge}/2\pi$, GHz	4.26	5.45	4.13	5.68
T_1, μs	25.9	11.2	15.1	16.2
T_2^*, μs	9.7	7.2	3.6	4.3

4. Моделирование. Проектирование квантовых цепочек для архитектуры квантовой рекуррентной нейросети и их автоматическое дифференцирование реализовано с использованием библиотеки *Pennylane* [31]. Для снижения трудоемкости эксперимента пределы качества предсказаний, оцениваемые по средне-

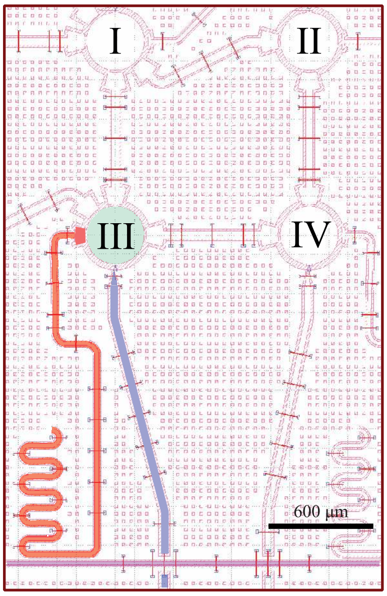

Рис. 2. (Цветной онлайн) Фрагмент чертежа сверхпроводниковой квантовой интегральной микросхемы процессора с квазидвумерной топологией связей между трансмонами в корпусе. Цифрами отмечены задействованные в эксперименте трансмоны. Показана одна из считывающих линий (розовый), резонатор для дисперсионного считывания кубита (красный), совмещенная линия управления потоком и микроволновым излучением (синий) и шунтирующая емкость острова трансмона (зеленый)

квадратичной ошибке (MSE), определялись на эмуляторе путем изменения таких гиперпараметров как число кубитов N, объем входных данных T, способ их сверточного преобразования τ и число повторений процедуры их записи $n_{\mathrm{reuploadings}}$. Обучение проводилось на шести временных рядах (см. "Приложение. Используемые данные"): трех аналитических функциях, заданных вручную: ARTIFICIAL 1, ARTIFICIAL 2, ARTIFICIAL 3, и трех реальных наборах данных, используемых в бенчмаркинге машинного обучения: PRESSURE, ELECTRICITY, TEMPERATURE.

Число кубитов в системе задает размерность гильбертова пространства, в котором осуществляется поиск решения. С одной стороны, увеличение размерности увеличивает описательную способность модели, но с другой – для охвата такого пространства требуется больше параметризованных операций (см. рис. 3). При фиксированной глубине квантовой цепочки увеличение числа кубитов приводит к улучшению качества обучения (MSE на тестовом сегменте временного ряда уменьшается), однако начиная с $N = 6$ кубитов наблюдается эффект насыщения, и относительное уменьшение MSE в сравнении с $N = 4$ кубитами становится незначительным.

Исследование зависимости MSE от объема входных данных показало, что при $T = 1$ (не представлено на графике) и $T = 2$ показатели точности не превышают уровень наивного прогноза вследствие недостаточной информативности входной последовательности. Использование $T = 50$ временных шагов (при различных параметрах свертки) значительно увеличивает время исполнения цепочки и, при этом, либо вовсе не приводит к улучшению метрики, либо замедляет скорость сходимости. Оптимальным с точки зрения баланса между точностью прогнозирования и скоростью сходимости является выбор $T = 10$ с последующей сверткой до $\tau = 4$ элементов (см. рис. 4). Применение свертки оправдано с точки с зрения обеспечения более плотного кодирования данных и уменьшения глубины цепочки, что особенно важно во время обучения на NISQ устройстве.

Внедрение процедуры перезагрузки (reuploading) данных (см. рис. 5) приводит к уменьшению ошибки прогнозов, однако для $N = 4$ кубитов и свертки от $T = 10$ к $\tau = 4$ это уменьшение оказывается не более чем в 3 раза. Для изначально менее удачных гиперпараметров (не изображено) перезагрузка может уменьшить ошибку на порядок. Из этих наблюдений следует, что метод перезагрузки при определенных настройках модели может быть эффективен и в случае квантовых нейронных сетей.

В таблице 2 приведены наилучшие метрики MSE, полученные при помощи различных архитектур: QRNN сравнивается с классическими рекуррентными архитектурами RNN, LSTM и GRU [32–34]. Количество параметров в сравниваемых моделях не превосходит 100.

Таблица 2. Сравнение QRNN с классическими архитектурами

MSE\architectures	QRNN	RNN	LSTM	GRU
ARTIFICIAL 1 (10^{-4})	5.35	0.78	0.89	**0.20**
ARTIFICIAL 2 (10^{-4})	1.54	0.29	0.31	**0.20**
ARTIFICIAL 3 (10^{-2})	**0.94**	1.04	1.21	1.11
PRESSURE (10^{-1})	2.51	**2.44**	2.54	2.46
ELECTRICITY (10^{0})	1.44	1.48	1.45	**1.41**
TEMPERATURE (10^{0})	**4.29**	4.41	4.35	4.36

5. Эксперимент. Для проведения эксперимента на сверхпроводящей интегральной схеме была выбрана задача прогнозирования атмосферного давления (PRESSURE). Для этой задачи выбраны оптимальные параметры модели, полученные на эмуляторе: 4 кубита (рис. 3d), параметризованная свертка

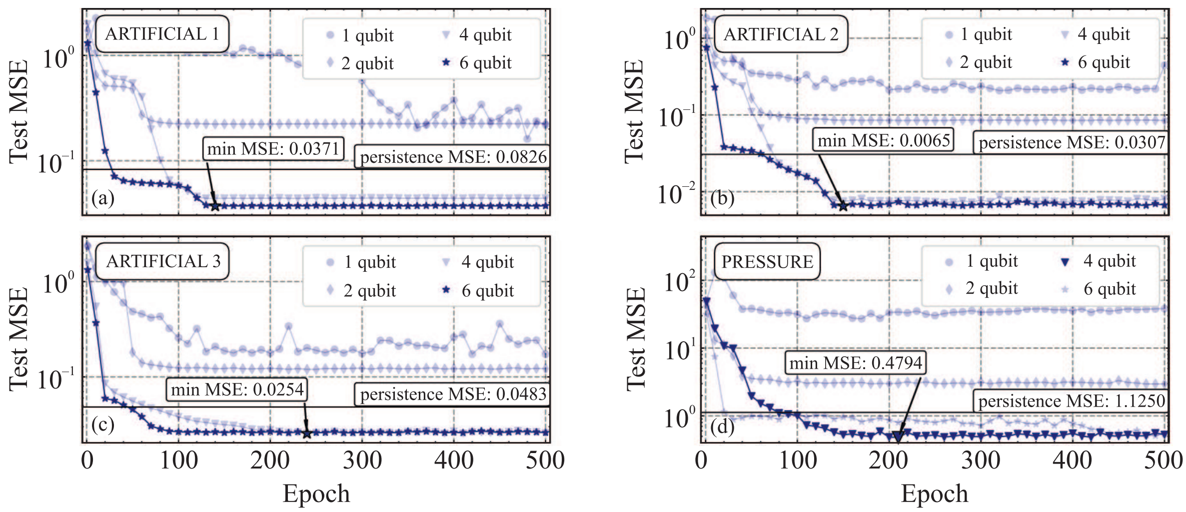

Рис. 3. (Цветной онлайн) Зависимости метрики MSE на тестовом сегменте временных рядов (a)–(d) от числа кубитов в алгоритме. Чтобы выявить однозначно “неудачные” параметры для запуска, на графиках была отображена метрика, соответсвующая предсказаниям наивной модели, которая в качестве предсказания на следующий временной шаг выдает значение последнего полученного на вход элемента данных (отмечена черной линией и подписью persistence MSE). Свертка и повторная загрузка данных здесь не используется

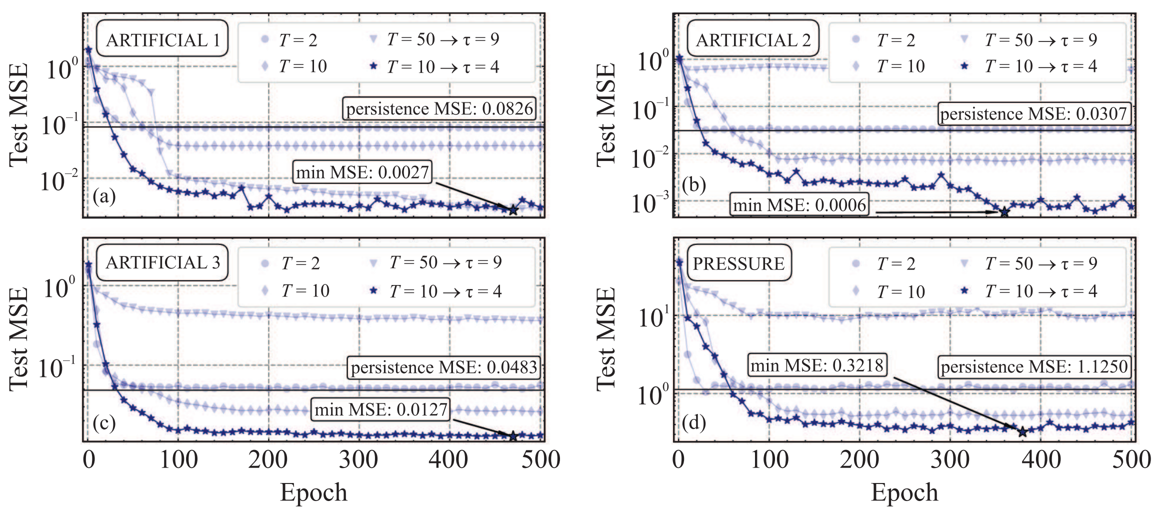

Рис. 4. (Цветной онлайн) Метрика MSE в зависимости от числа временных шагов T, используемых при построении прогноза, а также размера свертки. С увеличением значения T наблюдается рост качества прогноза до некоторого порогового уровня; дальнейшее улучшение достигается за счет использования свертки (схема представлена на рис. 1a). Число кубитов $N = 4$ зафиксировано, перезагрузка данных не используется

$T = 10$ значений ряда в $\tau = 4$ (рис. 4d) и перезагрузка данных $n_{\text{reuploadings}} = 3$ раза (рис. 5d). При выбранных гиперпараметрах цепочка содержит 71 обучаемый параметр и имеет глубину 99 слоев квантовых операций. В эксперименте сброс состояния регистровых кубитов после каждого блока операций не производился. Цепочка из операций, послойно реализуемых при помощи микроволновых сигналов, выполняется за $t_{\text{PQC}} = 5$ мкс, после чего состояние одного кубита считывается за $t_{\text{readout}} = 0.5$ мкс, и система релаксирует в основное состояние за $t_{\text{equilibrium}} = 100$ мкс. Процесс повторяется $n_{\text{shots}} = 1000$ раз для получения $\langle\sigma_z^{(1)}\rangle$. Вычисление градиента выбранной модели при помощи правила сдвига параметров (2), заключающееся в варьировании каждого из углов однокубитных операций на $\pm\pi/2$ требует $2\cdot n_{\text{1q gates}} + 1 = 2\cdot 171 + 1 = 343$ измерений (каждое получается усреднением по n_{shots} запускам), $t_{\text{waveform}} = 100$ мс требуется для записи сигнала при изменении угла какого-либо однокубитного поворота. Та-

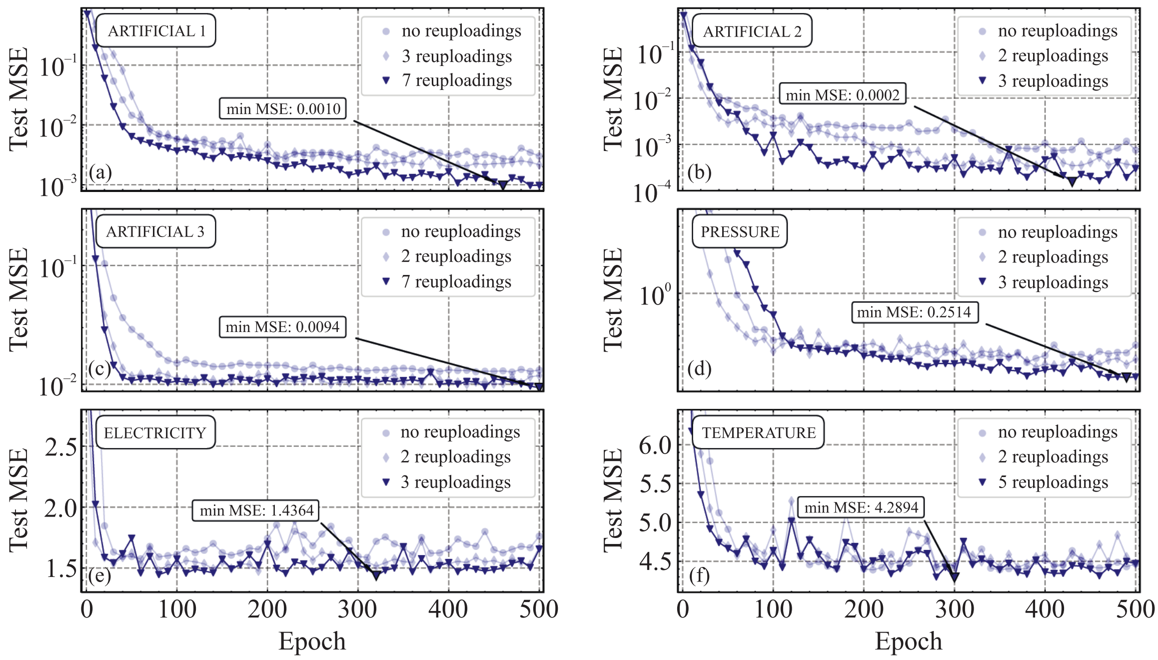

Рис. 5. (Цветной онлайн) Влияние повторной загрузки входных данных на MSE. Используется свертка $T = 10 \to \tau = 4$. Уменьшение ошибки с применением упомянутого метода составило: ARTIFICIAL 1 – в 1.7 раза, ARTIFICIAL 2 – в 2.7 раза, ARTIFICIAL 3 – на 35 %, PRESSURE – на 28 %, ELECTRICITY – на 10 %, TEMPERATURE – на 2 %

ким образом, для обработки одного элемента из тренировочного сегмента последовательности потребуется $t_{1\ \mathrm{element}} = \big(n_{\mathrm{shots}} \cdot (t_{\mathrm{PQC}} + t_{\mathrm{readout}} + t_{\mathrm{equilibrium}}) + t_{\mathrm{waveform}}\big) \cdot (2\,n_{\mathrm{gates}} + 1) \approx 71$ с. Время, затрачиваемое на эпоху обучения, соответственно, равно $t_{\mathrm{epoch}} = t_{1\ \mathrm{element}} \cdot (L - T) \approx 5$ ч.

Во время обучения на сверхпроводниковом квантовом процессоре наблюдается влияние ограниченных времен когерентности и неидеальности квантовых операций, приводящее к смещению экспериментальной кривой относительно результатов эмуляции (см. рис. 6). В пределах 25 эпох функция потерь демонстрирует немонотонное поведение, однако ее общий тренд остается убывающим. Кривая обучения модели не меняется на качественном уровне, и, следовательно, при дальнейшем обучении на квантовом процессоре можно ожидать достижения метрик, сопоставимых с результатами эмуляции и классических архитектур.

6. Заключение. В данной работе произведены эмуляция на классическом компьютере и эксперимент на сверхпроводящей интегральной схеме по обучению квантовой рекуррентной нейронной сети в задаче прогнозирования временных рядов. Продемонстрирована эффективность применения классической свертки данных во время их кодирования в состояние кубитов и многократной загрузки данных.

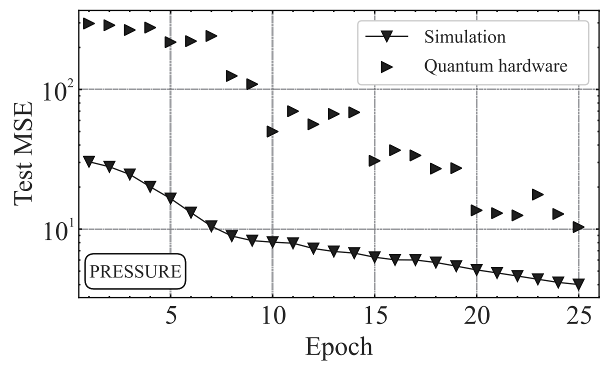

Рис. 6. Сравнение обучения QRNN на эмуляторе и сверхпроводниковом чипе

Большое внимание уделено исследованию обучаемости модели на эмуляторе в зависимости от задаваемых гиперпараметров. В эпоху шумных квантовых процессоров времена когерентности кубитов могут оказаться сравнимы с временами, затрачиваемыми на выполнение квантовой цепочки. Возникает необходимость баланса между глубиной алгоритма, увеличивающей описательную способность модели, и сохранением состояний кубитов.

Представленная архитектура для предсказания одного признака требует измерения лишь одного кубита, однако увеличение размерности вектора признаков на выходе модели возможно путем использования мультиплексированного считывания нескольких кубитов.

Несмотря на тот факт, что при работе экспериментального чипа операции выполняются за время порядка 10 нс, что делает сверхпроводниковые архитектуры самой быстродействующей платформой для квантовых вычислений на текущий момент, на 25 эпох экспериментального обучения ушло более 100 ч. Обучение классических архитектур требует на порядки меньших времен. Улучшение может быть достигнуто за счет уменьшения времени релаксации квантовой системы $t_{\text{equilibrium}}$ до 1 мкс при использовании безусловного сброса состояния кубитов [35].

Приложение. Используемые данные

Длины временных рядов, использовавшихся в этой работе (см. рис. 1A и 2A), для тренировочного и тестового сегмента соответственно составляют: искусственные данные – 242 и 58 элементов, PRESSURE – 250 и 50 элементов, ELECTRICITY – 700 и 300 элементов, TEMPERATURE – 1100 и 342 элементов.

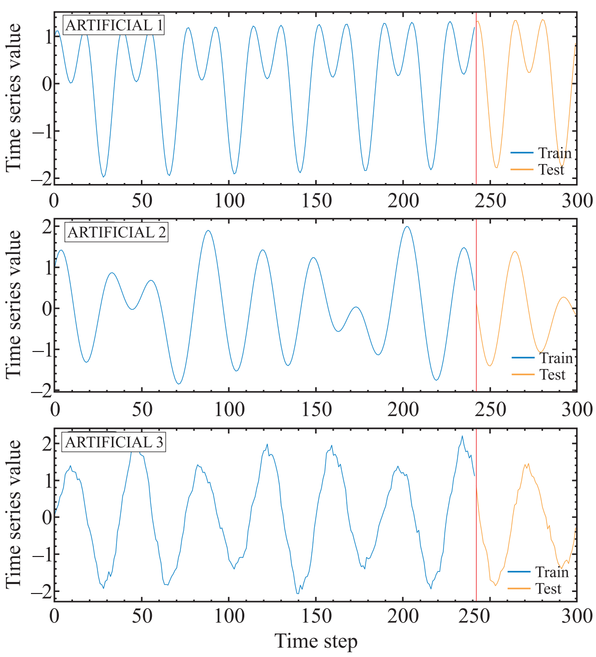

Рис. 1A. (Цветной онлайн) Искусственно заданные временные ряды, использованные в работе. Красная линия разделяет тренировочный и тестовый сегменты данных

Финансирование работы. Работа выполнена при финансовой поддержке Министерства науки и высшего образования Российской Федерации (Соглашение # 075-15-2024-538).

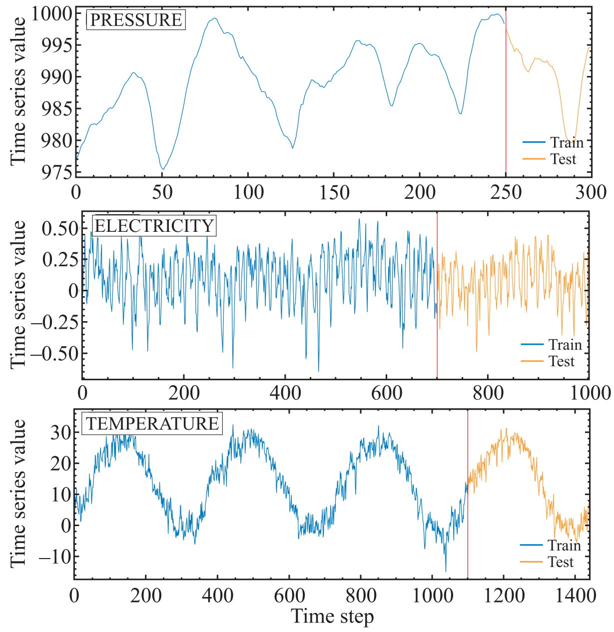

Рис. 2A. (Цветной онлайн) Реальные временные ряды, использованные в работе. Красная линия разделяет тренировочный и тестовый сегменты данных

Конфликт интересов. Авторы данной работы заявляют, что у них нет конфликта интересов.

1. T. Yin, Theoretical and Natural Science **51**, 34 (2024).
2. J. Li, J. Zhang, J. Zhang and S. Zhang, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **43**, 1332 (2024).
3. E. Cherrat, I. Kerenidis, N. Mathur, J. Landman, M. Strahm and Y. Li, Quantum **8**, 1265 (2024).
4. K. Bartkiewicz, C. Gneiting, A. Černoch, K. Jirakova, K. Lemr and F. Nori, Sci. Rep. **10**, 12356 (2020).
5. T. Pandey, V. Ravalekar, S. Nair, and S. Pradhan, Sci. Rep. **15**, 28443 (2025).
6. D. Risté, M. Silva, C. Ryan, A. Cross, J. Smolin, J. Gambetta, J. Chow, and B. Johnson, npj Quantum. Inf. **3**, 16 (2015).
7. M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, Quantum Science and Technology **4**, 043001 (2019); https://doi.org/10.1088/2058-9565/ab4eb5 (2019).
8. M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, Nat. Rev. Phys. **3**, 625 (2020).
9. L. Henriet, L. Beguin, A. Signoles, T. Lahaye, A. Browaeys, G.-O. Reymond, and C. Jurczak, Quantum **4**, 327 (2020).
10. A. Tolstobrov, G. Fedorov, S. Sanduleanu, S. Kadyrmetov, A. Vasenin, A. Bolgar, D. Kalacheva, V. Lubsanov, A. Dorogov, J. Zotova, P. Shlykov, A. Dmitriev, K. Tikhonov, and O. V. Astafiev, Phys. Rev. A **109**, 012411 (2024).

11. Z. Yin, I. Agresti, G. de Felice, D. Brown, A. Toumi, C. Pentangelo, S. Piacentini, A. Crespi, F. Ceccarelli, R. Osellame, B. Coecke, and P. Walther, Nat. Photonics **19**, 1020 (2025).
12. H.-L. Huang, Y. Du, M. Gong et al. (Collaboration), Phys. Rev. Appl. **16**, 024051 (2021).
13. D. Zhu, N. Linke, M. Benedetti, K. Landsman, N. Nguyen, C. Huerta Alderete, A. Perdomo-Ortiz, N. Korda, A. Garfoot, C. Brecque, L. Egan, O. Perdomo, and C. Monroe, Sci. Adv. **5**, eaaw9918 (2019).
14. W. Ren, W. Li, S. Xu et al. (Collaboration), Nature Computational Science **2**, 711 (2022).
15. T. Dutta, A. Pérez-Salinas, J. P. S. Cheng, J. I. Latorre, and M. Mukherjee, Phys. Rev. A **106**, 012411 (2022).
16. A. E. Tolstobrov, S. V. Kadyrmetov, G. P. Fedorov, S. V. Sanduleanu, V. B. Lubsanov, D. A. Kalacheva, A. N. Bolgar, A. Y. Dmitriev, E. V. Korostylev, K. S. Tikhonov, and O. V. Astafiev, Control. Radiophys. Quantum. **66**, 907 (2024).
17. P. Krantz, M. Kjaergaard, F. Yan, T. Orlando, S. Gustavsson, and W. Oliver, Appl. Phys. Rev. **6**, 021318 (2019).
18. M. Schuld and F. Petruccione, *Machine Learning with Quantum Computers*, Springer, Cham (2021).
19. E. Fontana, N. Fitzpatrick, D. M. Ramo, R. Duncan, and I. Rungger, Phys. Rev. A **104**, 022403 (2021).
20. A. Blais, R.-S. Huang, R.-S. Huang, A. Wallraff, S. M. Girvin, and R. J. Schoelkopf, Phys. Rev. A **69**, 062320 (2004).
21. S. Sim, P. D. Johnson, and A. Aspuru-Guzik, Advanced Quantum Technologies **2**, 1900070 (2019).
22. Y. Takaki, K. Mitarai, M. Negoro, K. Fujii, and M. Kitagawa, Phys. Rev. A **103**, 052414 (2021).
23. Y. Li, Z. Wang, R. Han, S. Shi, J. Li, R. Shang, H. Zheng, G. Zhong, and Y. Gu, Neural Netw. **166**, 148 (2023).
24. C. P. Williams and S. H. Clearwater, *Explorations in Quantum Computing*, 2nd ed., Texts in Computer Science, Springer-Verlag, Berlin, Heidelberg (1998).
25. A. Pérez-Salinas, D. López-Núnez, A. García-Sáez, P. Forn-Díaz, and J. I. Latorre, Phys. Rev. A **104**, 012405 (2021).
26. S. Jerbi, L. J. Fiderer, H. P. Nautrup, J. M. Kübler, H. J. Briegel, and V. Dunjko, Nat. Commun. **14**, 517 (2023).
27. M. Kölle, T. Witter, T. Rohe, G. Stenzel, P. Altmann, and T. Gabor, *A Study on Optimization Techniques for Variational Quantum Circuits in Reinforcement Learning*, in 2024 IEEE International Conference on Quantum Software (QSW) (2024), p. 157.
28. M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, Phys. Rev. A **99**, 032331 (2019).
29. N. Killoran, *The stochastic parameter-shift rule*, https://pennylane.ai/qml/demos/tutorial_stochastic_parameter_ shift, Date Accessed: 2025-11-03 (2020).
30. J. Koch, T. M. Yu, J. Gambetta, A. A. Houck, D. I. Schuster, J. Majer, A. Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf, Phys. Rev. A **76**, 042319 (2007).
31. V. Bergholm, J. A. Izaac, M. Schuld, C. Gogolin, A. Khandelwal, and N. Killoran, arXiv preprint arXiv:1811.04968 (2018).
32. J. J. Hopfield, Proceedings of the National Academy of Sciences of the United States of America **79**, 2554 (1982).
33. S. Hochreiter and J. Schmidhuber, Neural Comput. **9**, 1735 (1997).
34. K. Cho, B. van Merrienboer, C. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, *Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation*, in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (2014), p. 1724.
35. P. Magnard, P. Kurpiers, B. Royer, T. Walter, J.-C. Besse, S. Gasparinetti, M. Pechal, J. Heinsoo, S. Storz, A. Blais, and A. Wallraff, Phys. Rev. Lett. **121**, 060502 (2018).