Quantum Data Representation via Circuit Partitioning and Reintegration

Ziqing Guo*†, Jan Balewski†, Kewen Xiao‡, Ziwen Pan*

*Texas Tech University, Lubbock, TX, USA

†National Energy Research Scientific Computing Center, Lawrence Berkeley National Laboratory, Berkeley, CA, USA

‡Rochester Institute of Technology, Henrietta, NY, USA

***Abstract*—Quantum data encoding (QDE) enables faster computations than classical algorithms through superposition and entanglement. Circuit cutting and knitting are effective techniques for ameliorating current noisy quantum processing unit (QPUs) errors via a divide-and-conquer approach that splits quantum circuits into subcircuits and recombines them using classical postprocessing. Unfortunately, the existing QDE frameworks fail to consider quantum hardware limitations, such as the topology of the chip. Designing a computation model that supports the algorithm level of quantum computation and optimizes non-all-to-all connected quantum circuit simulations remains underdeveloped. In this study, we introduce shardQ, a method that leverages the SparseCut algorithm with matrix product state (MPS) compilation and a global knitting technique to mitigate the quantum error rates. This method elucidates the optimal trade-off between the computational time and error rate for quantum encoding with a theoretical proof, evidenced by an ablation analysis using an IBM Heron-type QPUs with 15% error reduction. This study also presents the results of quantum image encoding readiness. The proposed model advances the current quantum computation towards the fault-tolerant regime as QDE is the input of grand unified quantum algorithms.**

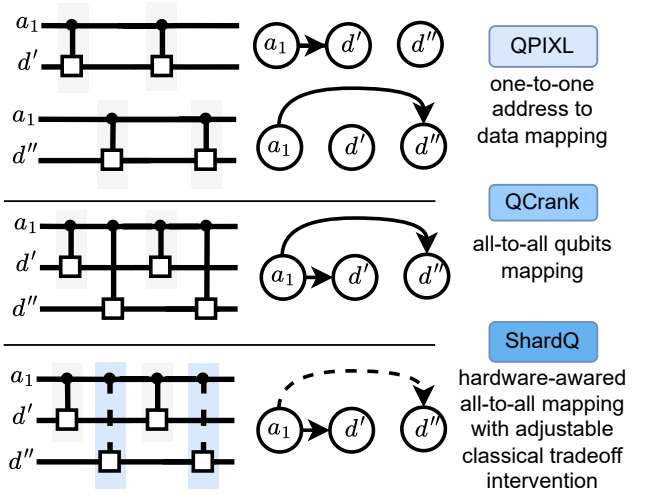


Fig. 1: Overview of the proposed method compared with conventional quantum data encoding techniques.

I. Introduction

Scaling quantum computation to the utility regime, where quantum processors deliver consistent application-level advantages, remains a central challenge in the noisy intermediate-scale quantum (**NISQ**) era [1]–[3]. Despite progress in quantum algorithms [4]–[6] and hardware [7]–[9], limited qubit connectivity [10], [11], high two-qubit gate error rates [12], and overhead of long-distance entanglement operations [13], [14] hinder the practical execution of large-scale quantum circuits on current superconducting QPUs. These constraints are particularly problematic for contemporary quantum workflows; therefore, large quantum circuits require to be partitioned, executed, and recombined for more efficient simulations.

Overcoming these limitations would enable the execution of resource-intensive quantum algorithms such as Grover's search [15], Shor's factoring [16], and Quantum Fourier Transform (**QFT**) [17] on near-term devices, unlocking practical applications in cryptography [18], optimization [19], and scientific simulation [20]. Among these algorithms, quantum block data encoding [21] provides a fundamental methodology for singular value transformation [22] which is broadly used by quantum walk [23] and quantum machine learning algorithms [24]. The capability of tensor network quantum state encoding to efficiently represent limited entanglement is coherently interfered with by the non-unitary phenomenon of quantum measurement, which enables a poly logarithmic speed-up compared with the classical approach [25], [26]. Notably, circuit cutting and knitting techniques that utilize quasi-probability decomposition (**QPD**) [27], [28] offer a promising route for reducing hardware requirements, without sacrificing algorithmic universality. This method is crucial for hybrid high-performance computing (**HPC**)-integrated quantum platforms [29], in which quantum resources must be tightly integrated with classical postprocessing.

State-of-the-art circuit decomposition methods, including space cut [30] and time cut [31], have demonstrated compatibility with universal two-qubit gate architectures [32], but they often neglect hardware-aware optimization for connectivity-limited QPUs. Alternative approaches, such as the hardware-

aware cutting framework [33], variational quantum clustering [34] and quantum embedding analysis [35], provide valuable insights into data representation but are not designed for direct classical data encoding and require iterative quantum-classical optimization. Consequently, these methods either fail to fully exploit **QPD** in practical **NISQ** systems or incur prohibitive resource overhead.

While **QPD** enables resource trade-offs through local operations and classical communication (LOCC) [36], [37], it introduces a sampling overhead and error amplification, particularly in algorithms requiring deep entanglement. Moreover, superconducting QPUs typically have finicky connectivity, forcing the use of additional SWAP operations or long-distance entanglement gates, which increases the circuit depth and error probability.

Although current quantum encoding methods offer a compact way to compress classical inputs for faster linear operations within the Hibert space, the current challenges are as follows: (1) The 2D grid non-all-to-all connecting superconducting layout introduces a physical distance that is aware after the ideal quantum circuits are transpiled. Consequently, long-distance data-encoded qubits exacerbate crosstalk errors when running on actual quantum hardware. (2) The cutting and knitting methodology requires an algorithmic design to select the best entangled qubit pairs and reconstruct the simulation results. Hence, a scalable and adjustable cutting protocol and global reconstruction technique should be considered as required by the partitioned quantum block data encoding.

In this study, we introduce **shardQ**, an end-to-end partition-to-recomposition quantum data encoding model specifically optimized for superconducting quantum chips in the **NISQ** era. The state-of-the-art quantum data encoding schemes of **QPIXL** [25] and (**QCrank**) [38] shown in Fig. 1 still require high-dimensional correlated superposition states that are costly to implement on real hardware with two-qubit operations. Our approach addresses these challenges by employing dynamic cut control based on the number of qubits to minimize two-qubit entanglement gate error rates, integrating hardware-aware circuit knitting for restricted qubit connectivity, and supporting HPC-integrated quantum workflow execution, thereby enabling the decomposition, distribution, and recombination of large-scale quantum algorithms with reduced error rates and circuit depths. Although this method is tailored to superconducting architectures, its principles can be adapted to other platforms with constrained connectivity.

To the best of our knowledge, the contributions of this study can be summarized as follows:

- We identify core limitations in current quantum data-encoding techniques and formalize a classical–quantum trade-off that guides large-scale simulation of encoding circuits.
- We introduce **SparseCut**, a hardware-aware circuit-cutting algorithm that suppresses long-distance entanglement via a user-defined interaction range and deterministically selects cut locations on the hardware connectivity graphb based on the number of qubits.
- We develop a global reconstruction technique that stitches subcircuit results using matrix-product-state (MPS) tensor-network contraction, reducing classical overhead and scaling to larger circuits and cut counts.
- We demonstrate on real quantum hardware that our proposed computation model reduces end-to-end error by about 15% relative to a standard transpilation baseline, while keeping the classical stitching time practical.
- We validate practical utility on grayscale quantum image–encoding benchmarks, indicating a viable path toward future fault-tolerant, fully quantum data-processing pipelines.

The remainder of this paper is organized as follows. § II reviews related work and motivates our cutting-and-knitting approach. § IV details the **shardQ** architecture and the algorithms for partitioning and recomposition. § V reports experimental results on quantum hardware. § VI discusses limitations and concludes the paper. The theoretical foundations and proofs are provided in the Appendix.

II. Related Work and Motivation

The concept underlying our approach traces back to the clifford-gate group simulation protocol [39], which reduces the classicalquantum computation overhead and strengthens the hybrid paradigm [27]. This foundation has driven both empirical [31] and theoretical [40] advances in clustered simulations of molecular systems, which were later enhanced by maximum-likelihood fragment tomography [41]. Subsequent studies examined the overhead of circuit cutting and knitting for entanglement gates [42]–[45] and error mitigation in low-depth entanglement circuits [46], leading to practical frameworks such as CutQC [47] and Qiskit Circuit Knitting [48]. However, these frameworks lack algorithm-level QPU-aware optimization for hardware with limited connectivity. Recent innovations include heuristic randomized measurement cutting for quantum approximate optimization algorithms (**QAOA**) [49] and high-fidelity cutting with gradient-based reconstruction [50]. The Qdislib framework [51] extends the cutting to distributed settings by integrating HPC with QPUs. However, the scalability and efficiency of such cutting methods, particularly for quantum data encoding in the **NISQ** era, remain uncertain. Our proposed approach is motivated by the error mitigation benefits of tightly coupling QPUs with classical HPC resources [52] to overcome the limitation of deterministic conventional quantum block data encoding methods. More importantly, the state-of-the-art classical-quantum interconnect communication platform NVQLink [1] can further optimize the postprocessing of classical overhead tradeoff indroduced by our model.

III. Background

In this section, we present the essential background for the proposed **shardQ** protocol. We first introduce the tensor network simulation technique. Subsequently, we outline the

[1] https://www.nvidia.com/en-us/solutions/quantum-computing/nvqlink/

approximate quantum data encoding approach. Finally, we provide the fundamental concepts underlying circuit cutting and the knitting process.

A. Quantum circuit simulation

MPS compilation. The quantum state can be written as

$$|\psi\rangle = \sum_{i=0}^{2^n-1} c_i |i\rangle, \tag{1}$$

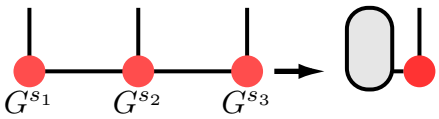
where classical hardware requires exponentially growing random access memory (RAM) to simulate the state vector, specifically 2^n complex amplitudes for n qubits. Matrix product states (MPS), based on tensor networks (TN), are widely used to mitigate RAM overhead. For example, the Greenberger–Horne–Zeilinger (GHZ) state can be represented as

$$\mathcal{G}_{s_1 s_2 s_3} = [A^{s_1} A^{s_2} A^{s_3}], \tag{2}$$

where $\mathcal{G}_{s_1 s_2 s_3}$ yields the computational basis amplitude for each $(s_1, s_2, s_3) \in \{0,1\}^3$. The contraction of a tensor network representing a quantum circuit is given by

$$\mathcal{A} = \sum_{i_k} \prod_j T^{[j]}_{i_{j-1}, i_j}, \tag{3}$$

where $T^{[j]}$ is the local tensor at site j, and indices $\{i_k\}$ are contracted according to the network (see Diagram (4)). Here, the contraction of three tensors produces an effective tensor by summing the internal indices.



(4)

An important advantage of this method is that, for circuits exhibiting structured entanglement, the time complexity is $O(\text{poly}(N) \cdot 2^w)$ [53], where w represents the minimal width determined by the circuit connectivity.

B. Gate-based quantum data encoding

Data encoding is essential for embedding and image encoding because it enables a compact Hilbert space representation of classical data. Quantum data encoding transforms classical information into quantum states using three methods. *Basis encoding* maps discrete values directly to computational basis states using binary representation; for example, $[01, 11]$ becomes $|x^1\rangle = |01\rangle$ and $|x^2\rangle = |11\rangle$. *Amplitude encoding* embeds normalized data $\vec{x}$ into quantum amplitudes $\sum_i \alpha_i |i\rangle$, subject to normalization. *Angle encoding* maps features to qubit states using rotation gates, so each feature is encoded as $\cos(\theta_i/2)|0\rangle + \sin(\theta_i/2)|1\rangle$, with θ_i rescaled to $[0, \pi]$. The resulting quantum state is

$$|\psi\rangle = \bigotimes_{i=1}^{n} \left(\cos\frac{\theta_i}{2} |0\rangle + \sin\frac{\theta_i}{2} |1\rangle \right). \tag{5}$$

Following angle encoding, **QCrank** encodes classical data with the Unitary Controlled Rotation (UCR_y) gate

$$\text{UCR}_y(\theta) = \begin{pmatrix} R_y(\theta_0) & & \\ & \ddots & \\ & & R_y(\theta_{2^{n_a}-1}) \end{pmatrix}. \tag{6}$$

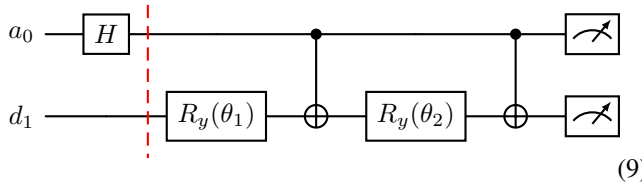
Each address qubit configuration (representing the position of each classical data input) $|i\rangle$ is prepared in a superposition using the Walsh–Hadamard Transform (WHT). For each address, the associated data values are encoded onto the data qubits by applying the single-qubit rotations. Specifically, for every address, each data qubit receives a rotation by an angle corresponding to the classical data value assigned to that address and the data qubit.

$$|c_{i,j}\rangle = \cos(\theta_{i,j}/2)\,|0\rangle + \sin(\theta_{i,j}/2)\,|1\rangle\,. \tag{7}$$

The UCR_y gate applies these rotations in a block-diagonal form, which is controlled by the address qubits. Thus, the **QCrank** encoder produces

$$|\psi_{\text{qcrank}}(\vec{\theta})\rangle = \frac{1}{\sqrt{2^{n_a}}} \sum_{i=0}^{2^{n_a}-1} |i\rangle \otimes |c_{i,0}\rangle \otimes \cdots \otimes |c_{i,n_d-1}\rangle. \tag{8}$$

with each $|c_{i,j}\rangle$ angle-encoded via UCR_y, n_a address qubits, and n_d data qubits. For example, encoding the 3D tensor $(1, 2, 1)$ yields the circuit



(9)

resulting in $|\psi_{\text{qcrank}}\rangle = \frac{1}{\sqrt{2}}(|0\rangle|0\rangle + |1\rangle[c|0\rangle + s|1\rangle])$, where $c = \cos\left(\frac{\theta_1+\theta_2}{2}\right)$, $s = \sin\left(\frac{\theta_1+\theta_2}{2}\right)$. To retrieve classical data, projective measurements estimate $|c|^2$ and $|s|^2$, and the encoded angle is reconstructed by

$$\theta_1 + \theta_2 = 2\arctan\left(\sqrt{\frac{|s|^2}{|c|^2}}\right), \tag{10}$$

Assuming that the angle is rescaled to $[0, \pi]$ using $arccos(0, 1)$ via EVEN encoding [54].

C. Circuit Cutting and Knitting

The circuit knit-and-cut technique enables the efficient simulation of large quantum circuits by dividing them into smaller, manageable subcircuits (SC), which are simulated separately and then recombined to construct the global observables. This approach exploits the fact that the expectation value of an operator on the entire circuit, such as $\mathcal{S}(\mathbf{U}_1 \otimes \mathbf{U}_2)$ [30], can be represented as a linear combination of the expectation values of each subcircuit, each weighted by classical coefficients, where $\mathcal{S}$ represents the operator in Pauli groups. For two

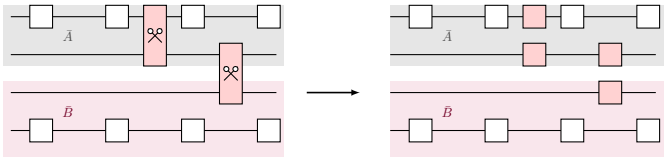


Fig. 2: The generic example of cutting two qubit gates into separate one qubit gates. Note that, the right side only shows one of the subcircuits.

arbitrary unitaries U, the observable in the QPD framework is reconstructed as

$$\mathcal{S}(\mathbf{U}_1 \otimes \mathbf{U}_2) = \mathbf{c}^T \mathbf{S} + \frac{1}{4}\sin(2\theta)\sin(2\phi)\sum_{\boldsymbol{\alpha}} \alpha_1 \alpha_2 \, \mathcal{R}(\boldsymbol{\alpha}). \tag{11}$$

Here, $\mathbf{S}$ contains the expectation values of the subcircuits with different operator insertions at the cut, and $\mathbf{c}$ provides the corresponding weights of the expectation values. The sum over $\boldsymbol{\alpha}$ captures the correlations introduced by the cut (see Appendix A in [40]). This technique relies on efficient classical post-processing to recover the outcome of the original circuit (see Table 1 in [36]). Here, we provide the general QPD sampling overhead for a CX gate for our protocol

Definition 1 (QDP overhead for CX gate)**.** *Let C be a quantum circuit; the QDP overhead is*

$$O_{\mathrm{QDP}(C)} = 3^{2k}, \tag{12}$$

where k is the number of cut qubits, 2 symbolizes the decomposition of the control and target gates, and each factor of base 3 reflects the three nontrivial Pauli insertions (X, Y, Z) per cut qubit, as shown in Fig. 2.

IV. Methods

Fig. 3 shows the end-to-end **shardQ** protocol that leverages circuit cutting algorithm, quantum approximate compilation, and global result reconstruction. Note that the proposed protocol is for a gate-based quantum simulation. Specifically, we provide a general cutting strategy, **SparseCut**, for the quantum encoder, which further allows the best MPS compilation. In addition, we propose a generic global measured bit string reconstruction algorithm.

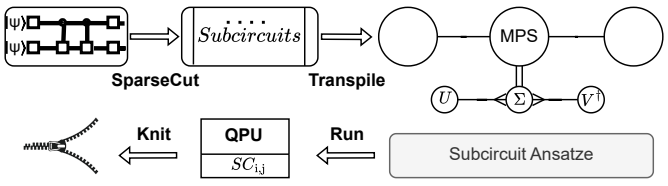


Fig. 3: The **shardQ** protocol is outlined as follows: Initially, the original data encoder circuit employs the **SparseCut** algorithm to divide the circuit into subcircuits (SC). Subsequently, approximate quantum compilation is used to transpile these subcircuits into the MPS. The ansatze are executed on the QPU, where index i denotes the partition and j represents the decomposed gates. Ultimately, the results are globally reconstructed into classical tensor data using local saved intermediate results.

A. Cutting Algorithm

Similar to the use of permuted controlled unitary operations in **QFT**, the **QCrank** encoder employs layers of gray-coded CX gates to facilitate data entanglement and connectivity in a high-dimensional Hilbert space. Therefore, we note that it is highly time consuming to split the circuit into small partitions with smaller clusters simulated with fewer qubits because of the scaling overhead shown in Eq. (12). To address this, we provide the **SparseCut** algorithm illustrated in Alg. 1, where an example of a circuit-cutting scheme with three address qubits and three data qubits is shown in Fig. 4. Here, we define a cutting selection rule and its goals.

Definition 2. *Given by the set of the cutting candidates*

$$\mathcal{G}_{cross} = \left\{ \begin{array}{l} g = (n_a, n_d), \\ n_a \in A,\ n_d \in D, \\ d(n_a, n_d) \end{array} \right\}. \tag{13}$$

Here, n_a and n_d refer to the address and data qubits, respectively; d denotes the distance; and the goal is to minimize the qubit map distance in the cutting pool $\mathcal{G}$. First, we recall that the address and data qubits encode Eq. (9) data $= \begin{bmatrix}\begin{bmatrix}\theta_{0,0,0}\\ \theta_{0,1,0}\end{bmatrix}\end{bmatrix}$. Here, we denote data$[c][a][d]$, where the parameters correspond to the circuit, address, and data indices. The coupling map of Fig. 5 is retrieved from the state-of-the-art IBM 156 qubit Pittsburgh QPU; the qubit indexes correspond to the latest layout [55]. By taking the address and data qubit indices,

Algorithm 1 SparseCut Selection

Require: circuit C, observable qubit sets A, D, maximum cuts max_cuts

$\mathcal{G}_{\text{cross}} \leftarrow \varnothing$
for each two–qubit gate $g = (q_i, q_j)$ in C **do**
  if $\big(q_i \in A \wedge q_j \in D\big) \ \vee \ \big(q_i \in D \wedge q_j \in A\big)$ **then**
    $d \leftarrow |i - j|$
    $\mathcal{G}_{\text{cross}} \leftarrow \mathcal{G}_{\text{cross}} \cup \{(\text{gate\_index}, d)\}$
sort $\mathcal{G}_{\text{cross}}$ by d in **descending** order
return first $\min\big(|\mathcal{G}_{\text{cross}}|, \text{ max\_cuts}\big)$ elements

Alg. 1 iteratively updates the current gate index (as shown in the 15th gate from left to right in our example in Fig. 4) based on the absolute distance between two sets A and D. The benefit of retrieving absolute differences becomes clearer when using different quantum platforms because the resulting quantum bit strings may be in the reverse order. We also introduce a hyperparameter `max_cuts` that defines the upper bound of the cutting protocols. In other words, in each recursion, the algorithm finds the longest entanglement distance based on the gray code law by looping the control and target qubit indices and returning the minimum number of gate indices in the group of cutting candidates. The absolute virtual qubit distance is calculated by subtracting the target and control qubit indices from each other. We refer to the example of the **QPD** case in the Appendix.

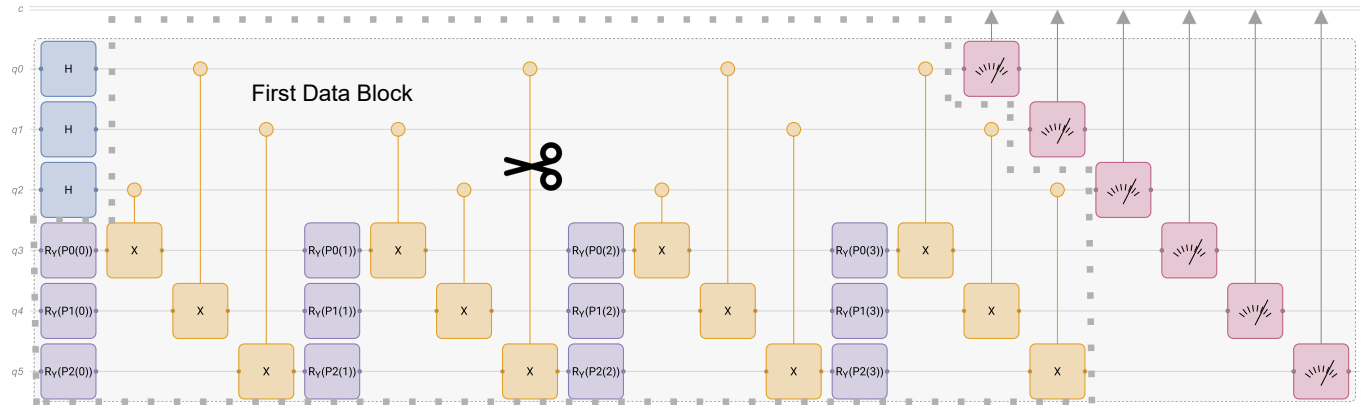


Fig. 4: Example of three-by-three tensor encoder circuit cutting paradigm. The scissors were placed under the longest entanglement gate, corresponding to the physical qubit mapping, as shown in Fig. 5. We only show the first data block, as indicated by the grey dashed box. We refer to the remaining encoding blocks in Fig. 1 (c) [38]. Note that, each data qubit (q_3, q_4, q_5) corresponding to first encoded dimension P with the address qubit encoded position as the rotation parameters noted by the indexes of P.

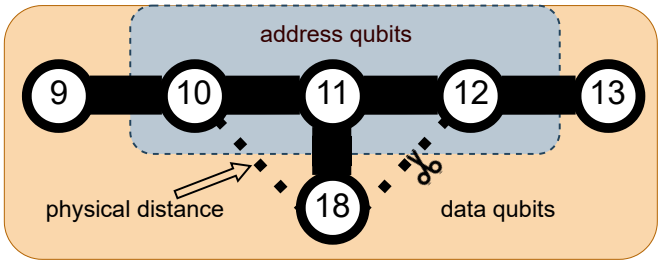


Fig. 5: The three-by-three tensor encoder physical qubit mapping diagram. We denote the address qubits corresponding to q_0, q_1, q_2 in Fig. 4 by the blue-shaded area. The yellow area represents the data qubits q_3, q_4, q_5. The longest entanglement cut is q_0 (the first address qubit) and q_5 (third data qubit) shown in the dashed diagonal lines.

B. MPS Compilation

In alignment with the principles of **SparseCut**, the algorithm is distinctively characterized by its adherence to universal optimality, as the selection of the shortest path for severing the longest nondirect connecting edge aligns with Dijkstra's algorithm [56]. Consequently, the approximate quantum compilation (**AQC**) technique [57] employed in our protocol facilitates a further reduction in the gate depth post-cut Alg. 1, thereby compressing the tensor encoder in Eq. (8). The complete encoder circuit C is decomposed into a prefix C_1 and suffix C_2 such that $C = C_2 C_1$, with only C_1 being compiled. All two-qubit operations that couple the address register A to the data register D are ranked according to their Manhattan distance d, and the $k = \texttt{max\_cuts}$ gates with the largest d from the cutting set $\mathcal{G}^{\star}_{\text{cross}}$. The removal of these gates results in the truncated circuit C_1^{trunc}, whose output state can be simulated as an MPS with bond dimension $\chi \ll 2^{|A|+|D|}$. It is important to note that MPS provides an explicit representation of the tensor encoder target state $|\psi_{\text{tar}}\rangle \approx |\psi_{\text{qcrank}}\rangle$, as demonstrated in Eq. (8).

Algorithm 2 Reconstruct global counts

Require: job set $\mathcal{R} = \{r_1, \ldots, r_m\}$; QPD coeffs. $\mathcal{C}$
Ensure: global counter $\mathcal{G}\colon \Sigma^* \to \mathbb{R}$

$\mathcal{Y} \leftarrow \emptyset$ ▷ multiset of local counters.
for all $r \in \mathcal{R}$ **do**
  $\mathbf{o} \leftarrow \text{LABELS}^1(r.\text{observables})$, $\mathbf{q} \leftarrow \text{LABELS}(r.\text{qpd})$
  $\mathsf{cnt} \leftarrow [\, s \mapsto 0 \,]$
  for $k = 1$ **to** $|\mathbf{o}|$ **do**
    $s \leftarrow \mathbf{o}_k \parallel \mathbf{q}_k$; $\mathsf{cnt}(s) \mathrel{+}= 1$
  $\mathcal{Y} \cup = \{\mathsf{cnt}\}$
$n_o \leftarrow n_q^{\text{tot}}$; $n_q \leftarrow |\text{dom}(\mathcal{Y}[1])| - n_o$
$\mathcal{G} \leftarrow [\, s \mapsto 0 \,]$
for all $(\mathsf{cnt}, c) \in \mathcal{Y} \times \mathcal{C}$ **do**
  for all $(s, n) \in \mathsf{cnt}$ **do**
    obs $\leftarrow s[1{:}n_o]$, qpd $\leftarrow s[n_o{+}1{:}]$
    $\sigma \leftarrow \begin{cases} \text{PARITY}^2(\text{qpd}) & n_q > 0 \\ 1 & \text{otherwise} \end{cases}$
    $\mathcal{G}(\text{reverse}(\text{obs})) \mathrel{+}= c\,\sigma\,n$
for all $s \in \text{dom}(\mathcal{G})$ **do**
  $\mathcal{G}(s) \leftarrow \max\big(0, \lfloor \mathcal{G}(s) + 0.5 \rfloor\big)$
return $\mathcal{G}$

The ansatz $\tilde{C}_{\boldsymbol{\theta}}$, which is hardware-native and incorporates only nearest-neighbor couplings, is derived from C_1^{trunc} through a KAK-based [58] block factorization. This ensures that the initial parameter vector $\boldsymbol{\theta}_0$ precisely reconstructs C_1^{trunc}, except for the global phase. Subsequently, optimization is conducted by minimizing the infidelity cost function

$$\mathcal{L}(\boldsymbol{\theta}) = 1 - \left|\langle \psi_{\text{tar}} | \psi(\boldsymbol{\theta}) \rangle\right|^2, \quad |\psi(\boldsymbol{\theta})\rangle = \tilde{C}_{\boldsymbol{\theta}} |0\rangle^{\otimes L}, \tag{14}$$

where the gradients are obtained by automatic differentiation through tensornetwork contraction. Because long-range gates in $\mathcal{G}^{\star}_{\text{cross}}$ are absent from the simulation, their entangling effect is reproduced variationally by the ansatz parameters, allowing

[1] LABELS converts a bit-array (rows of 0/1 or non-negative integers) into a list of binary strings.
[2] PARITY returns $(-1)^{\#1\text{'s}}$ of its input.

χ to remain small while still capturing the dominant short-range correlations inside A and D. After convergence the compiled circuit is reconstructed as

$$C_{\mathrm{AQC}} = C_2\, \mathcal{G}^{\star}_{\mathrm{cross}}\, \tilde{C}_{\boldsymbol{\theta}^{\star}}, \tag{15}$$

which approximates the original encoder with fidelity exceeding $1-\varepsilon$ while containing k fewer long-range CX layers than the original encoder.

C. Global Reconstruction

We provide the pseudocode in Alg. 2 for the global bit string reconstruction. Note that `dom` represents the domain of the Hilbert Space. To recover the original circuit statistics, we start with basic **QPD** recursion. For a single Pauli cut, the quasi-probability expansion of an observable $\mathcal{M}$ acting on circuit $\mathcal{C}$ is

$$T(\mathcal{C}, \mathcal{M}) = \sum_{i=1}^{8} c_i\, T(\mathcal{C}', \mathcal{M}_i), \tag{16}$$

where $T(\mathcal{C}, \mathcal{M}) = \mathrm{Tr}[\mathcal{M}\,\mathcal{C}(\rho)]$ and $\mathcal{C}'$ denotes the circuit after inserting one Pauli completion. By iterating the rule over M cuts gives

$$T(\mathcal{C}, \mathcal{M}) = \sum_{\boldsymbol{\alpha}\in\{1,\ldots,8\}^M} \Big(\prod_{m=1}^{M} c_{\alpha_m}\Big)\, T\big(\mathcal{C}', \mathcal{M}_{\boldsymbol{\alpha}}\big), \tag{17}$$

with $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_M)$ enumerating the 8^M completion patterns (see Eq. (29)). Removing every cut edge splits $\mathcal{C}'$ into K independent fragments,

$$\mathcal{C}' = C^{(1)} \sqcup C^{(2)} \sqcup \cdots \sqcup C^{(K)}. \tag{18}$$

Because of the independent expectation values, we provide

$$T\big(\mathcal{C}', \mathcal{M}_{\boldsymbol{\alpha}}\big) = \prod_{k=1}^{K} T\Big(C^{(k)}, \mathcal{M}^{(k)}_{\boldsymbol{\alpha}_{\mathrm{cuts}(k)}}\Big), \tag{19}$$

from Eq. (17) and Eq. (18) where $\boldsymbol{\alpha}_{\mathrm{cuts}(k)} \subset \boldsymbol{\alpha}$ holds only the remaining components that act on fragment k. We note that $\mathrm{SC}_{i,j}, i = 1, \ldots, K,\ j = 1, \ldots, 8^{m_i}$, in Fig. 3 represents the subcircuits that runs fragment $C^{(i)}$ with the j-th Pauli completion of its m_i local cuts. Executing all $\mathrm{SC}_{i,j}$ on the QPU provides the conditional probabilities $P^{(i)}\big(b_i \mid \boldsymbol{\alpha}_{\mathrm{cuts}(i)}\big)$. Inserting these probabilities into the squared-modulus of (19) yields the knitted distribution

$$P(\mathbf{b}) = \sum_{\boldsymbol{\alpha}} \Big(\prod_{m=1}^{M} c_{\alpha_m}\Big) \prod_{i=1}^{K} P^{(i)}\big(b_i \mid \boldsymbol{\alpha}_{\mathrm{cuts}(i)}\big). \tag{20}$$

Each non-Clifford gate within $\mathcal{C}$ can be expressed through the QPD expansion $U(\theta) = \sum_j c_j(\theta)\, G_j$, with an associated overhead $\Gamma(\theta) = \sum_j |c_j(\theta)|$, such Pauli measurement pairs with a cut set S increases the Monte Carlo variance by at most $\prod_{g\in S} \Gamma(\theta_g) \le 9^{|S|}$. Given that each Pauli completion G_j is a Clifford gate, every $\mathrm{SC}_{i,j}$ can be efficiently simulated classically in polynomial time as per the Gottesman–Knill theorem. Consequently, the total post-processing effort is scaled as $\mathcal{O}\big(9^{|S|}\,\mathrm{poly}(n)\big)$. The final global result can be represented by the trigonometric function $\mathcal{G}[s] \leftarrow \mathcal{G}[s] + c\_i \cdot \mathrm{sign}(q) \cdot \mathcal{P}$, where the sign is given by the PARITY function and $\mathcal{P}$ is the probability of the expected results of the sub-experiments calculated by the measured bit strings based on the shots. This is because achieving optimal local operation classical communication (LOCC) [37] overhead requires internal communication in each subcircuit. In the realistic noisy quantum simulation scenario, we emphasize that the advantage of using the cutting pool $\mathcal{G}$ defined in Def. 2 allows the reconstruction algorithm to process the quantum bit string results with one pass to substitute sequential processing.

V. Result

A. Performance Analysis

$Q^{\star}$: Does the **shardQ** protocol facilitate state-of-the-art **QPD** results in the context of three-dimensional tensor encoding?

To address this, we demonstrate that the protocol effectively reduces crosstalk errors in the current noisy QPU, where the goal is to compare the **shardQ** protocol with and without the original encoder. Here, we perform the benchmark test for the performance analysis with the selection of two address and data qubits.

ShardQ Protocol Analysis. We demonstrate that the protocol provides a lower error rate for quantum circuit simulation because the protocol physically cuts the longest entanglement gate into local unitary operations, as shown in Fig. 6. These benefits arise for two reasons. First, the idle qubit performs single-qubit Pauli operations after the cut, which allows for a longer coherence time because of the probe of the pulse in the conductor of the superconducting circuit hardware. Second, our MPS-enabled compilation further reduces the transpiled circuit depth, which limits the entanglement gates, allowing the results to have better locality using shallower subcircuits because of the tensor approximate contraction. We emphasize that the root mean square error (RMSE) of the quantum reconstructed data and true data (classical inputs) trend reveals that the cut simulation constantly outperforms the original uncut simulation.

Overhead Evaluation. However, we note that classical simulation overhead is unavoidable because of the QPD technique. In the ablation test, we demonstrate that beyond two cuts, diminishing performance returns coupled with exponential time growth demonstrate computational intractability, validating the practical selection of the number of data qubits as the optimum configuration for real-world deployment, as shown in Fig. 7. Here, we indicate that the optimum settings of the two cuts also have the best trade-off with respect to the classical overhead and error rates. This is because we have two address and data qubits in the simulated quantum circuit, where each data qubit introduces a non-avoidable long depth entanglement that can be eliminated by our model. Note that classical postprocessing was conducted using one 80 GB A100 GPU. We refer to the work [59] to extend the simulation across

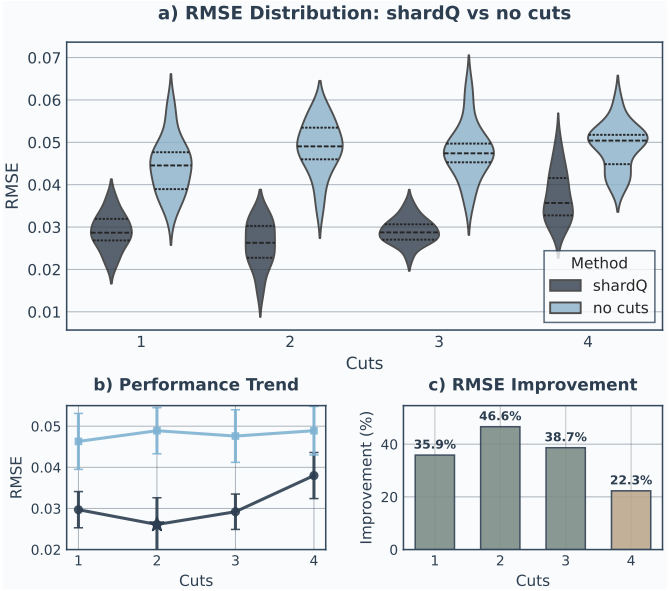


Fig. 6: Ablation study evaluating **shardQ** performance across cuts. (a) the middle dashed line symbolize the median with two dashed lines above and below indicate the 25% and 75% percentile. (b) RMSE performance trending line shown with the error bars. (c) Relative RMSE improvement quantification with color-coded bars indicating improvement magnitude (beige: 15-25%, sage: $> 25\%$). Error bars represent standard deviation across independent trials.

multiple GPU nodes to improve the scalability of the number of cuts.

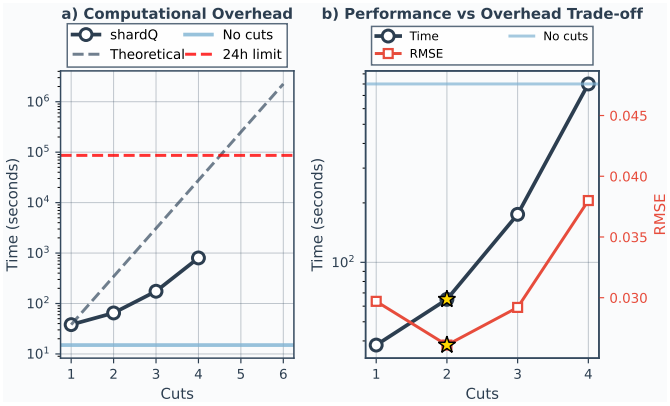


Fig. 7: Computational overhead analysis and performance trade-off evaluation for **shardQ** method. (a) Exponential computational scaling showing measured execution times (circles) closely following Eq. (12) with 24-hour practical limit (red dashed line) exceeded beyond 4 cuts. Baseline no-cuts method maintains constant $\sim$ 15s execution time. (b) The gold star indicates the optimum trade-off corresponding with Fig. 6.

B. Application

Furthermore, we demonstrate that the optimal two-cut enabled quantum data encoding simulation on the IBM ideal simulator facilitates near-perfect reconstruction of a grayscale image, as shown in Fig. 8. The total encoded tensor length is determined by $2^{n_a} * n_d$; thus, for an image comprising

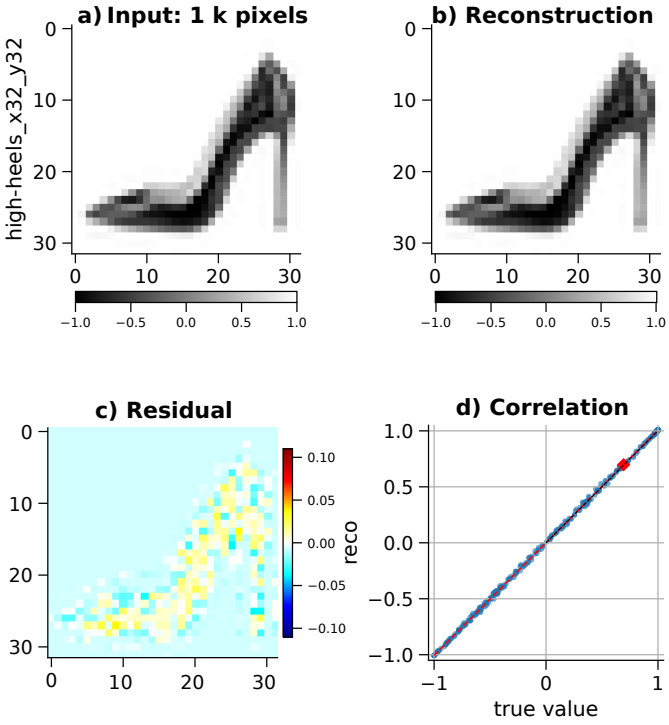


Fig. 8: The correlation between the reconstructed values and the ground truth using **shardQ** model is linearly correlated; noted as the reconstructed value fidelity (RVF).

1,000 pixels, we select nine address qubits and two data qubits to encode the image. We allocate 3,000 shots per classical data-encoded position (2^9), aligning with the methodology of employing ideal GPU quantum image simulation [59]; consequently, the total number of shots per subcircuit amounted to 1.5 million. The results indicate that the **shardQ** protocol yields a quantum-encoded image with an error rate of less than 1% and standard deviation of four orders of magnitude. Note that the quantum image encoding result is set up with an all-to-all qubit connection; therefore, the performance is only affected by the `max_cuts`.

Additionally, we use **ibm_pittsburgh** and **ibm_marrakesh** as the primary noise models to benchmark with the non-partitioned quantum encoding methodology, listed in Table I. Because the **QPD** overhead scales exponentially with cuts, we employ four distributed GPU nodes to cut each data qubit to address the long-distance entanglement issue. Consequently, the CZ-depth of the simulated quantum circuit is shorter than that of the baseline method because the SWAP gates are significantly lower after transpilation. We showcase our model, which provides RVF enhancement on real quantum hardware by over 15%.

VI. Discussion

The **shardQ** protocol presents an **NISQ**-friendly framework for quantum data encoding circuits, particularly within gate-based quantum platforms such as IBM, and can be extended to any topological-based quantum chips. By utilizing the

TABLE I: Summary of simulated experiments using **QCrank** and **shardQ** with the noise model explicitly substituted by the best performing IBM QPU noise characteristics. We denote that all the **shardQ** experiments are conducted based on the best-performing two cuts protocol. Experiment 1-2 verify the baseline benchmark aligned with Appendix C in [38]. Experiment 3-4 demonstrates the scaling of the encoded data results. Experiment 5 further explores the previous Heron-2 type of chip result.

Experiment	#1 **QCrank**	#2 **shardQ**	#3 **QCrank**	#4 **shardQ**	#5 **shardQ**
Objective	RVF on Heron-3				RVF on Heron-2
Noisy QPUs	**IBM ibm_pittsburgh Noise Model**[†]				**ibm_marrakesh**
addr. qubits (n_a)	4	4	5 – 7	5 – 7	4
data qubits (n_d)	8	8	8	8	8
number of addresses	16	16	32 – 128	32 – 128	16
input (bits)	128	128	256 – 1,024	256 – 1,024	128
RVF (Result)	0.79	**0.90**	0.75	**0.88**	0.75

[†] The noise model is derived from the current calibration data of the **IBM ibm_pittsburgh** Heron r3 processor (best performance QPU provided with median controlled-Z (CZ) error at $1.46 * 10^{-3}$) and median square root of X (SX) Error at $1.623 * 10^{-4}$.

SparseCut and global bit string reconstruction techniques, our approach addresses a significant challenge: extending quantum circuit simulation to fault-tolerant quantum computing (**FTQC**). This is crucial because future HPC-integrated quantum platforms will necessitate the division of quantum circuit simulations and approximations across different hardware. Our experimental findings validate the feasibility of the optimal cut strategy and low-error-rate quantum image encoding. We anticipate that our protocol will facilitate the future development of quantum computers capable of executing deeper and more intricately structured entangled circuits, thereby producing more reliable results.

VII. Acknowledgements

This research used resources of the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231 using NERSC award NERSC DDR-ERCAP0034486.

References

[1] J. Preskill, "Quantum computing in the nisq era and beyond," *Quantum*, vol. 2, p. 79, 2018.

[2] D. P. DiVincenzo, "Thirty years of quantum computing," *Quantum Science and Technology*, vol. 10, no. 3, p. 030501, 2025.

[3] J. Preskill, "Quantum computing 40 years later," in *Feynman Lectures on Computation*, pp. 193–244, CRC Press, 2023.

[4] A. Montanaro, "Quantum algorithms: an overview," *npj Quantum Information*, vol. 2, no. 1, pp. 1–8, 2016.

[5] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," *Phys. Rev. Lett.*, vol. 103, p. 150502, Oct 2009.

[6] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," 2014.

[7] N. P. De Leon, K. M. Itoh, D. Kim, K. K. Mehta, T. E. Northup, H. Paik, B. Palmer, N. Samarth, S. Sangtawesin, and D. W. Steuerman, "Materials challenges and opportunities for quantum computing hardware," *Science*, vol. 372, no. 6539, p. eabb2823, 2021.

[8] Y.-Y. Jiang, C. Deng, H. Fan, B.-Y. Li, L. Sun, X.-S. Tan, W. Wang, G.-M. Xue, F. Yan, H.-F. Yu, *et al.*, "Advancements in superconducting quantum computing," *National Science Review*, vol. 12, no. 8, p. nwaf246, 2025.

[9] H. Aghaee Rad, T. Ainsworth, R. Alexander, B. Altieri, M. Askarani, R. Baby, L. Banchi, B. Baragiola, J. Bourassa, R. Chadwick, *et al.*, "Scaling and networking a modular photonic quantum computer," *Nature*, vol. 638, no. 8052, pp. 912–919, 2025.

[10] P. Yuan and S. Zhang, "Full characterization of the depth overhead for quantum circuit compilation with arbitrary qubit connectivity constraint," *Quantum*, vol. 9, p. 1757, 2025.

[11] Y. Kim, H. Kim, J. Kang, W. Choi, and Y. Kwon, "Effectiveness of the syndrome extraction circuit with flag qubits on ibm quantum hardware," *Quantum*, vol. 9, p. 1893, 2025.

[12] D. Hothem, J. Hines, C. Baldwin, D. Gresh, R. Blume-Kohout, and T. Proctor, "Measuring error rates of mid-circuit measurements," *Nature Communications*, vol. 16, no. 1, p. 5761, 2025.

[13] S. Saha, M. Shalaev, J. O'Reilly, I. Goetting, G. Toh, A. Kalakuntla, Y. Yu, and C. Monroe, "High-fidelity remote entanglement of trapped atoms mediated by time-bin photons," *Nature Communications*, vol. 16, no. 1, p. 2533, 2025.

[14] B.-C. Ciobanu, L. P. Verzotti, and P. G. Popescu, "Optimal and scalable entanglement distribution over crossbar quantum networks," *Scientific Reports*, vol. 14, no. 1, p. 11714, 2024.

[15] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, (New York, NY, USA), p. 212–219, Association for Computing Machinery, 1996.

[16] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM review*, vol. 41, no. 2, pp. 303–332, 1999.

[17] D. Coppersmith, "An approximate fourier transform useful in quantum factoring," *arXiv preprint quant-ph/0201067*, 2002.

[18] S. Pirandola, R. Laurenza, C. Ottaviani, and L. Banchi, "Fundamental limits of repeaterless quantum communications," *Nature communications*, vol. 8, no. 1, p. 15043, 2017.

[19] A. Abbas, A. Ambainis, B. Augustino, A. Bärtschi, H. Buhrman, C. Coffrin, G. Cortiana, V. Dunjko, D. J. Egger, B. G. Elmegreen, *et al.*, "Challenges and opportunities in quantum optimization," *Nature Reviews Physics*, pp. 1–18, 2024.

[20] M. K. Bhaskar, S. Hadfield, A. Papageorgiou, and I. Petras, "Quantum algorithms and circuits for scientific computing," *arXiv preprint arXiv:1511.08253*, 2015.

[21] D. Camps and R. Van Beeumen, "Fable: Fast approximate quantum circuits for block-encodings," in *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pp. 104–113, IEEE, 2022.

[22] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, "Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics," in *Proceedings of the 51st annual ACM SIGACT symposium on theory of computing*, pp. 193–204, 2019.

[23] A. M. Childs, "Universal computation by quantum walk," *Physical review letters*, vol. 102, no. 18, p. 180501, 2009.

[24] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195–202, 2017.

[25] M. G. Amankwah, D. Camps, E. W. Bethel, R. Van Beeumen, and T. Perciano, "Quantum pixel representations and compression for n-dimensional images," *Scientific reports*, vol. 12, no. 1, p. 7712, 2022.

[26] E. Tang, "A quantum-inspired classical algorithm for recommendation systems," in *Proceedings of the 51st Annual ACM SIGACT Sympo-*

sium on Theory of Computing, STOC 2019, (New York, NY, USA), p. 217–228, Association for Computing Machinery, 2019.
[27] S. Bravyi, G. Smith, and J. A. Smolin, "Trading classical and quantum computational resources," *Physical Review X*, vol. 6, no. 2, p. 021043, 2016.
[28] H. Pashayan, J. J. Wallman, and S. D. Bartlett, "Estimating outcome probabilities of quantum circuits using quasiprobabilities," *Physical review letters*, vol. 115, no. 7, p. 070501, 2015.
[29] M. Mohseni, A. Scherer, K. G. Johnson, O. Wertheim, M. Otten, N. A. Aadit, Y. Alexeev, K. M. Bresniker, K. Y. Camsari, B. Chapman, *et al.*, "How to build a quantum supercomputer: Scaling from hundreds to millions of qubits," *No journal information*, 11 2024.
[30] K. Mitarai and K. Fujii, "Constructing a virtual two-qubit gate by sampling single-qubit operations," *New Journal of Physics*, vol. 23, no. 2, p. 023021, 2021.
[31] T. Peng, A. W. Harrow, M. Ozols, and X. Wu, "Simulating large quantum circuits on a small quantum computer," *Physical review letters*, vol. 125, no. 15, p. 150504, 2020.
[32] D. P. DiVincenzo, "Two-bit gates are universal for quantum computation," *Physical Review A*, vol. 51, no. 2, p. 1015, 1995.
[33] X. Ren, M. Zhang, and A. Barbalace, "A hardware-aware gate cutting framework for practical quantum circuit knitting," in *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*, pp. 1–9, 2024.
[34] P. Bermejo and R. Orús, "Variational quantum and quantum-inspired clustering," *Scientific Reports*, vol. 13, no. 1, p. 13284, 2023.
[35] M. Rath and H. Date, "Quantum data encoding: A comparative analysis of classical-to-quantum mapping techniques and their impact on machine learning accuracy," *EPJ Quantum Technology*, vol. 11, no. 1, p. 72, 2024.
[36] L. Schmitt, C. Piveteau, and D. Sutter, "Cutting circuits with multiple two-qubit unitaries," *Quantum*, vol. 9, p. 1634, 2025.
[37] L. Brenner, C. Piveteau, and D. Sutter, "Optimal wire cutting with classical communication," *arXiv preprint arXiv:2302.03366*, 2023.
[38] J. Balewski, M. G. Amankwah, R. Van Beeumen, E. W. Bethel, T. Perciano, and D. Camps, "Quantum-parallel vectorized data encodings and computations on trapped-ion and transmon qpus," *Scientific Reports*, vol. 14, no. 1, p. 3435, 2024.
[39] S. Bravyi and A. Kitaev, "Universal quantum computation with ideal clifford gates and noisy ancillas," *Physical Review A—Atomic, Molecular, and Optical Physics*, vol. 71, no. 2, p. 022316, 2005.
[40] K. Mitarai and K. Fujii, "Methodology for replacing indirect measurements with direct measurements," *Physical Review Research*, vol. 1, no. 1, p. 013006, 2019.
[41] M. A. Perlin, Z. H. Saleem, M. Suchara, and J. C. Osborn, "Quantum circuit cutting with maximum-likelihood tomography," *npj Quantum Information*, vol. 7, no. 1, p. 64, 2021.
[42] C. Piveteau and D. Sutter, "Circuit knitting with classical communication," *IEEE Transactions on Information Theory*, vol. 70, no. 4, pp. 2734–2745, 2023.
[43] S. Yang and P. Murali, "Understanding the scalability of circuit cutting techniques for practical quantum applications," *arXiv preprint arXiv:2411.17756*, 2024.
[44] G. Gentinetta, F. Metz, and G. Carleo, "Overhead-constrained circuit knitting for variational quantum dynamics," *Quantum*, vol. 8, p. 1296, 2024.
[45] M. Jing, C. Zhu, and X. Wang, "Circuit knitting facing exponential sampling-overhead scaling bounded by entanglement cost," *Physical Review A*, vol. 111, no. 1, p. 012433, 2025.
[46] K. Temme, S. Bravyi, and J. M. Gambetta, "Error mitigation for short-depth quantum circuits," *Physical review letters*, vol. 119, no. 18, p. 180509, 2017.
[47] W. Tang and M. Martonosi, "Cutting quantum circuits to run on quantum and classical platforms," *arXiv preprint arXiv:2205.05836*, 2022.
[48] I. Shehzad, E. Pednault, J. R. Garrison, C. Johnson, B. Fuller, and J. R. Glick, "Automated cut finding and circuit knitting on large quantum circuits," in *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*, vol. 2, pp. 406–407, IEEE, 2024.
[49] A. Lowe, M. Medvidović, A. Hayes, L. J. O'Riordan, T. R. Bromley, J. M. Arrazola, and N. Killoran, "Fast quantum circuit cutting with randomized measurements," *Quantum*, vol. 7, p. 934, 2023.
[50] M. Hart and J. McAllister, "Reconstructing cut quantum circuits maximising fidelity between quantum states," in *Proceedings of the 21st ACM International Conference on Computing Frontiers*, pp. 224–231, 2024.
[51] M. Tejedor, B. Casas, J. Conejero, A. Cervera-Lierta, and R. M. Badia, "Distributed quantum circuit cutting for hybrid quantum-classical high-performance computing," *arXiv preprint arXiv:2505.01184*, 2025.
[52] A. Carrera Vazquez, C. Tornow, D. Riste, S. Woerner, M. Takita, and D. J. Egger, "Combining quantum processors with real-time classical communication," *Nature*, vol. 636, no. 8041, pp. 75–79, 2024.
[53] A. Berezutskii, M. Liu, A. Acharya, R. Ellerbrock, J. Gray, R. Haghshenas, Z. He, A. Khan, V. Kuzmin, D. Lyakh, *et al.*, "Tensor networks for quantum computing," *Nature Reviews Physics*, pp. 1–13, 2025.
[54] J. Balewski, M. G. Amankwah, E. Bethel, T. Perciano, and R. Van Beeumen, "Ehands: Quantum protocol for polynomial computation on real-valued encoded states," *arXiv preprint arXiv:2502.15928*, 2025.
[55] IBM Quantum, "Ibm quantum system: Pittsburgh." https://quantum.cloud.ibm.com/computers?system=ibm_pittsburgh, 2025. Accessed: November 19, 2025.
[56] B. Haeupler, R. Hladík, V. Rozhoň, R. E. Tarjan, and J. Tetěk, "Universal optimality of dijkstra via beyond-worst-case heaps," in *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 2099–2130, IEEE, 2024.
[57] N. Robertson, A. Akhriev, J. Vala, and S. Zhuk, "Approximate quantum compiling for quantum simulation: A tensor network based approach," *ACM Transactions on Quantum Computing*, vol. 6, no. 3, pp. 1–15, 2025.
[58] J. Zhang, J. Vala, S. Sastry, and K. B. Whaley, "Geometric theory of nonlocal two-qubit operations," *Physical Review A*, vol. 67, no. 4, p. 042313, 2003.
[59] Z. Guo, Z. Pan, and J. Balewski, "Q-gear: Improving quantum simulation framework," *arXiv preprint arXiv:2504.03967*, 2025.
[60] S. Faro, A. Pavone, C. Viola, *et al.*, "Families of constant-depth quantum circuits for rotations and permutations," in *Proceedings of the 25nd Italian Conference on Theoretical Computer Science, Torino, Italy*, 2024.
[61] A. Bookstein, V. A. Kulyukin, and T. Raita, "Generalized hamming distance," *Information Retrieval*, vol. 5, pp. 353–375, 2002.

APPENDIX

In this section, we present a detailed proof of the decomposition of the CX gate within our protocol, which involves two address qubits and one data-qubit tensor data encoder. Notably, the permutation of UCR_y gates represents the current state-of-the-art approach for differential data encoding blocks, which can be realized through the constant-depth quantum circuits demonstrated in [60]. Specifically, the single rotation gate and controlled Y rotation gate are

$$\begin{aligned} R_y(\theta) &= \cos\left(\frac{\theta}{2}\right) I - i \sin\left(\frac{\theta}{2}\right) Y, \\ CRY(\theta) &= |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes R_y(\theta). \end{aligned} \tag{21}$$

The commutative principle is expressed in Eq. (18-21) of [38], where the CRY can be decomposed into a R_y with a CX gate. Such a decomposition enables the construction of the encoder circuit with two address qubits and one data qubit, denoted as $\mathcal{C}_{21}$ shown in Fig. 9. Note that the permutation of CX is encoded with the gray code [61] that optimizes the Hamming distance between the neighbors with the maximum value of one. The advantages become more apparent when we have more address qubits because the traversal of control configurations allows more efficient updating of CX control patterns; only one control bit changes between consecutive operations, which minimizes the number of CX gates required to reconfigure the multi-controlled rotations. We recall that the number of the address qubits is n_a, hence, the R_y gates are parameterized by $P_0\{a_{00}\}, \ldots, P_0\{binary(2^{n_a}\}$. Specifically, the data-to-angle encoding is given by

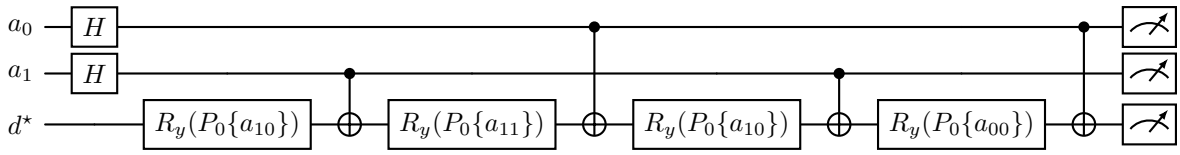


Fig. 9: The example of two address and one data qubits **QCrank**.

$$\theta^{\text{final}} = \text{Gray}\left(\text{FWHT}\left(\arccos(\mathbf{d})\right)\right) \tag{22}$$

where $\mathbf{d} \in [-1,1]^N$ is the input data vector, $\arccos(\cdot)$ maps data to angles, FWHT is the scaled fast Walsh-Hadamard transform, and the gray code permutation is used to optimize the control pattern for efficient quantum circuit simulation.

Given that the simplest maximum cut number set is one using Alg. 1, we obtain the cutting gate index as the sixth from circuit $\mathcal{C}_{21}$. Then, considering that the uncut circuit is decomposed into six subcircuits because of the Pauli group, each subcircuit is attained through the probabilistic Clifford gate representation. Here, the KAK decomposition for $\mathcal{C}_{21}$ is given by implementing a uniformly controlled rotation with address qubits q_0, q_1 and data qubit q_2 denoted by

$$|\psi\rangle = \frac{1}{2}\sum_{i=0}^{3} |i\rangle \otimes \left(\cos\frac{\alpha_i}{2}|0\rangle + \sin\frac{\alpha_i}{2}|1\rangle\right). \tag{23}$$

Then, by recalling that § IV-C, each controlled rotation gate can be decomposed to

$$\text{UCR}_y(\vec{\alpha}) = \sum_{a_1,a_2\in\{\pm 1\}} c_{a_1,a_2} \cdot P_{a_1}^{(0)} \otimes P_{a_2}^{(1)} \otimes R_y(\theta_{a_1,a_2}). \tag{24}$$

where P is the Pauli measurement projector. We note that the number of cut indices results in the $\mathcal{O}(n)$ linearly increasing of the QPD measurement stored as the temporary results with the coefficient shown in Eq. (29) because **SparseCut** allows cuts in the data block level, as shown in Fig. 3. To prove the CX cut, we first recall the CZ gate decomposition shown in Fig. 6 of [31]. Additionally, the CX can be represented by CZ with

$$CX \;=\; (I\otimes H)\,\text{CZ}\,(I\otimes H). \tag{25}$$

However, only three terms (Pauli X, Pauli Z, and Hadamard gate) are required for the **QCrank**. Here, CZ can be summed by R_z gate because of the two virtual qubit gate decomposition principle [40]

$$\text{CZ} = \sum_{a_1,a_2\in\{\pm 1\}^2} a_1 a_2 \left\{RZ\left(\frac{a_1\pi}{2}\right) \otimes RZ\left(\frac{a_2\pi}{2}\right)\right\}. \tag{26}$$

With Eq. (26), it applies the conjugation to Eq. (25)

$$CX \;= (I\otimes H)\cdot\left[\sum_{a_1,a_2\in\{\pm 1\}^2} a_1 a_2 \Big\{ RZ\left(\frac{a_1\pi}{2}\right) \otimes RZ\left(\frac{a_2\pi}{2}\right)\Big\}\right]\cdot(I\otimes H) \tag{27}$$

Because $H \cdot RZ(\theta) \cdot H = RX(\theta)$ (referred to Clifford decomposition table Table II), this gives us separate terms conditioned based on the coefficients

$$H \cdot RZ\left(\frac{a_2\pi}{2}\right) \cdot H = RX\left(\frac{a_2\pi}{2}\right). \tag{28}$$

The general decomposition for the two virtual qubit gates can be written in the format of super operators with a single qubit operation sandwiching the observable density matrix. Let us expand the observables into three Pauli matrices and identity term, therefore, the coefficients are can be defined by

$$\begin{aligned}
&O_1 = I, && \rho_1 = |+\rangle\langle+|, && c_1 = +\tfrac{1}{2},\\
&O_2 = I, && \rho_2 = |-\rangle\langle-|, && c_2 = +\tfrac{1}{2},\\
&O_3 = X, && \rho_3 = |0\rangle\langle 0|, && c_3 = +\tfrac{1}{2},\\
&O_4 = X, && \rho_4 = |1\rangle\langle 1|, && c_4 = -\tfrac{1}{2},\\
&O_5 = Y, && \rho_5 = |-i\rangle\langle -i|, && c_5 = +\tfrac{1}{2},\\
&O_6 = Y, && \rho_6 = |+i\rangle\langle +i|, && c_6 = -\tfrac{1}{2},\\
&O_7 = Z, && \rho_7 = |+\rangle\langle+|, && c_7 = +\tfrac{1}{2},\\
&O_8 = Z, && \rho_8 = |-\rangle\langle-|, && c_8 = -\tfrac{1}{2}.
\end{aligned} \tag{29}$$

Here, $\mathcal{O}_i$ is the observable, ρ_i is the eigenstate, c_i is the

Gate	**Decomposition**
X	$H \cdot Z \cdot H$
Y	$H \cdot Z \cdot H \cdot Z$
Z	S^2
R_X	$H \cdot S^\dagger \cdot H$
R_Y	$S \cdot H \cdot S^\dagger \cdot H \cdot S^\dagger$
R_Z	$S^\dagger$
R_{YZ}	$H \cdot S^\dagger \cdot H \cdot Z$
R_{ZX}	$S^\dagger \cdot H \cdot S^\dagger \cdot H \cdot S^\dagger$
R_{XY}	$H \cdot Z \cdot H \cdot S^\dagger$
ΠX	$S \cdot H \cdot S \cdot H \cdot P_0 \cdot H \cdot S^\dagger \cdot H \cdot S^\dagger$
ΠY	$H \cdot S^\dagger \cdot H \cdot P_0 \cdot H \cdot S \cdot H$
ΠZ	P_0
Π_{YZ}	$S \cdot H \cdot S \cdot H \cdot P_0 \cdot H \cdot S \cdot H \cdot S^\dagger$
Π_{ZX}	$H \cdot S^\dagger \cdot H \cdot P_0 \cdot H \cdot S \cdot H \cdot Z$
Π_{XY}	$P_0 \cdot H \cdot Z \cdot H$

TABLE II: Decomposition of gates in terms of H Hadamard gate, S Phase gate, $S^\dagger$ Reverse phase gate, Z Pauli Z gate, and P_0 Z-based measurement gate used in controlled-rotation circuits. We note that the decomposition for the Pauli groups is defined in the Clifford gate group [39] except for the Z measurement gate. In local operator classic communication, **QPD** stores the mid-circuit measurement result for the global measurement reconstruction.

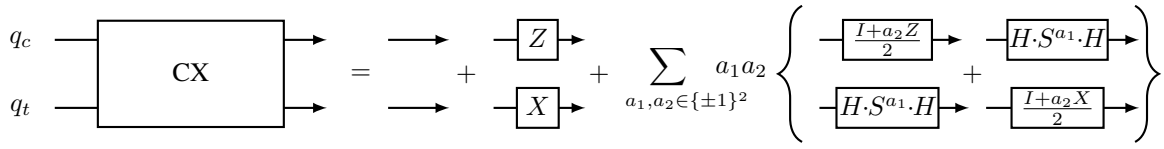


Fig. 10: CX–decomposition written exclusively with H, S, $S^\dagger$, Z, X and the projector $P_0 = \frac{1}{2}(I+Z)$. Every rotation $R_Z(\pm\pi/2)$ has been replaced by S or $S^\dagger$; every $R_X(\pm\pi/2)$ is implemented as the Clifford $H\,S^{(\dagger)}\,H$.

coefficient. Note that the cut gate has one prepared state and a measured observable. Specifically, on the preparation side, we apply a 1-qubit density matrix $\rho_\lambda = |\lambda\rangle\langle\lambda|$ that is the eigenstate of the Pauli as appearing in Eq. (29). On the measurement side, $s = \pm 1$ serves as the measured and recorded eigenvalues. In the experiment, we used a gate-based quantum circuit simulation. After shot-averaging,

$$\big\langle s \big\rangle = \mathrm{Tr}\big[P\,\rho_\lambda\big] = \lambda, \tag{30}$$

so the product of eigenvalue and shot average reproduces the Pauli and we denote the QPD Pauli representation of CX gate as

$$\mathrm{CX}_{c\to\,t} = \frac{1}{2}\Big(I_c{\otimes}I_t + Z_c{\otimes}I_t + I_c{\otimes}X_t - Z_c{\otimes}X_t\Big), \tag{31}$$

where c is the conditioned (control) qubit and t is the target qubit. Therefore, to generalize the eight Pauli observable and re-prepare for the global states, we categorize the six terms denoted in Table III. Here, Alg. 1 selects the Clifford

TABLE III: Post-measurement bases associated with the six transformed Pauli–Stinespring operators Φ_i'. S: unitary on both sides, M_A: measure & reprepare qubit 1, M_B: mid-measurement on qubit 2.

Φ_i' from (E1 [30])	Applied gates	Computation basis
$S(I\otimes I)$	no mid measurement	identity term
$S(A\otimes B)$	no mid measurement	$Z\otimes X$ term
$M_A\otimes S(e^{i\pi B/4})$	$S^\dagger H$ + Z-meas.	$S^\dagger$ basis ($\|\pm i\rangle$)
$M_A\otimes S(e^{-i\pi B/4})$	SH + Z-meas.	S basis ($\|\mp i\rangle$)
$S(e^{i\pi A/4})\otimes M_B$	I + Z-meas.	computational ($\|0\rangle, \|1\rangle$)
$S(e^{-i\pi A/4})\otimes M_B$	H + Z-meas.	Hadamard ($\|\pm\rangle$)

bases for efficient quantum hardware simulation because of the efficiency of the Clifford group overhead simulation. We note that rows 1 and 2 share the same hardware setting; the difference is an S vs. $S^\dagger$ gate; likewise 3 and 4 share Hadamard and no Hadamard settings. Each basis yields two possible classical outcomes, which correspond to the "$\pm$" states listed in the table. Because distinct quantum circuits (measurement settings) are required, the eight table rows are recovered by classical post-processing of the outcomes from the six circuits. Therefore, to produce the single cut as shown in the Fig. 9, we combine Eq. (31) and Eq. (29). Hereby we proved Fig. 10. Note that, in the transpiled version of quantum circuit mapping to the physical qubits, we do not consider the UCR_y gate in the **SparseCut** algorithm. To complete the proof for generalize UCR_y, therefore, by combining Eq. (24) and E1 [30], the complete UCR_y gate decomposes as

$$\mathrm{UCR}_y(\vec{\alpha}) = \frac{1}{4}\sum_{j=1}^{6} w_j \cdot \mathcal{M}_j \otimes \mathcal{R}_j \tag{32}$$

$$= \frac{1}{4}\begin{bmatrix} w_1\langle{+i}|\otimes R_y(\theta_1) + w_2\langle{-i}|\otimes R_y(\theta_2) \\ +w_3\langle{+i}|\otimes I\cdot R_y(\theta_3) + w_4\langle{-i}|\otimes Z\cdot R_y(\theta_4) \\ +w_5\langle 0|\otimes R_y(\theta_5) + w_6\langle{+}|\otimes R_y(\theta_6) \end{bmatrix} \tag{33}$$

For each subcircuit produces measurement outcomes with probabilities

$$P(m_j = \pm 1) = \frac{1 \pm \langle\psi|\sigma_j^{(0)}\otimes I^{(1)}\otimes I^{(2)}|\psi\rangle}{2} \tag{34}$$

where $|\phi\rangle$ is Eq. (23). Finally, we produce the global measurement reconstruction using Alg. 2. The original expectation values are recovered through

$$\langle R_y(\alpha_i)\rangle = \sum_{j=1}^{6} c_j \cdot \mathrm{Corr}(m_j^{(0)}, m_j^{(2)}) \tag{35}$$

where $\mathrm{Corr}(m_j^{(0)}, m_j^{(2)})$ represents the correlation between the address and data qubit measurements in subcircuit j, thereby completing the end-to-end quantum data encoder circuit partitioning and recomposition.