Dead Gate Elimination

Yanbin Chen[0000−0002−1123−1432], Christian B. Mendl[0000−0002−6386−0230], and Helmut Seidl[0000−0002−2135−1593]

School of CIT, Technical University of Munich, Garching 85748, Germany
`{yanbin.chen, christian.mendl, helmut.seidl}@tum.de`

Abstract. Hybrid quantum algorithms combine the strengths of quantum and classical computing. Many quantum algorithms, such as the variational quantum eigensolver (VQE), leverage this synergy. However, quantum circuits are executed in full, even when only subsets of measurement outcomes contribute to subsequent classical computations. In this manuscript, we propose a novel circuit optimization technique that identifies and removes dead gates. We prove that the removal of dead gates has no influence on the probability distribution of the measurement outcomes that contribute to the subsequent calculation result. We implemented and evaluated our optimization on a VQE instance, a quantum phase estimation (QPE) instance, and hybrid programs embedded with random circuits of varying circuit width, confirming its capability to remove a non-trivial number of dead gates in real-world algorithms. The effect of our optimization scales up as more measurement outcomes are identified as non-contributory, resulting in a proportionally greater reduction of dead gates.

Keywords: Quantum compilation · Dynamic circuit optimization.

1 Introduction

In recent efforts to address complex real-world problems, researchers are increasingly integrating quantum and classical computing to use the unique strengths of both paradigms [18]. In such interdisciplinary development, domain specialists are exploring ways to implement or even accelerate specific subroutines through quantum circuits tailored to quantum processing units (QPUs) [2, 3, 13, 17, 26]. Concurrently, quantum experts may incorporate classical computing procedures, given the wealth of sophisticated classical computing procedures that have been developing over decades [8, 31, 32]. A popular algorithm framework that allows to take advantage of the strength of both quantum and classical computers is *hybrid programs* [19]. In hybrid programs, quantum circuits are embedded as subroutines into programs from a classical host language. Usually, the classical host program handles optimization, control, and data processing, while the quantum circuits are used for specific calculations that may benefit from quantum speedups.

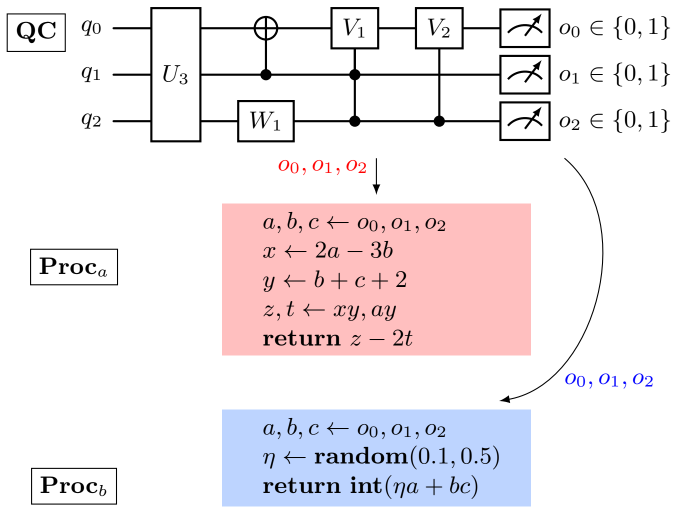
However, this integration can present challenges [10,11,28]. When researchers work beyond their core expertise, the interplay between classical and quantum

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97632-2_10

components may be suboptimal. This imperfect coupling risks inefficient resource utilization.

An implicit assumption is often made that the circuits are executed as external entities and all qubits are measured in the end and their outcomes are collected for purposes that are not of interest to circuits. So, circuits are fully executed even if not all measurement outcomes contribute to later calculations. However, in the following example, we see the potential for circuit simplification when knowing that some measurement outcomes are not needed.

Example 1. In the hybrid program in Fig. 1, the measurement outcome o_0 does not contribute to final results: in $\mathbf{Proc}_a$, the initial value of variable a, i.e. o_0, gets canceled out in the expression $z - 2t$; in $\mathbf{Proc}_b$ the initial value of variable a has no impact on the return value, because $0 \le \eta a \le 0.5$ and thus the ηa part is always rounded down to 0 by $\mathbf{int}(\cdot)$ operator. So, if we execute the circuit Fig. 2, where the measurement outcome from q_0 is always discarded, and we assign an arbitrary value from $\{0, 1\}$ to o_0, the results of the both $\mathbf{Proc}_a$ and $\mathbf{Proc}_b$ will not be influenced. Then, we could optimize the program by running the simplified circuit Fig. 3 instead of circuit Fig. 2. We call gates removed by this analysis dead gates. We will formally justify that this simplification will never influence calculation results of hybrid programs in Section 3.



```
$a, b, c \leftarrow o_0, o_1, o_2$
$x \leftarrow 2a - 3b$
$y \leftarrow b + c + 2$
$z, t \leftarrow xy, ay$
**return** $z - 2t$
```

```
$a, b, c \leftarrow o_0, o_1, o_2$
$\eta \leftarrow \mathbf{random}(0.1, 0.5)$
**return int**$(\eta a + bc)$
```

Fig. 1: An example of a hybrid program, where a quantum circuit **QC** of 3 qubits are first executed and then the measurement outcomes o_0, o_1, o_2 from qubits q_0, q_1, q_2, respectively, are dispatched to one of the two classical computing procedures, $\mathbf{Proc}_a$, or $\mathbf{Proc}_b$.

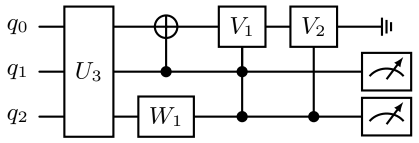


Fig. 2: A 3-qubit circuit. The measurement outcome of the top qubit is discarded.

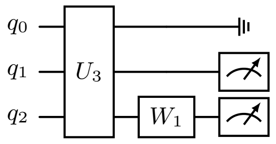


Fig. 3: Simplification of the circuit in Fig. 2. The probability distribution on measurement outcomes that are not discarded remains unchanged.

Non-contributory measurement outcomes can also occur in scenarios where the classical computing procedure only queries a subset of the available measurement results. This could make the same simplification possible, as we will demonstrate in the VQE and QPE examples in Section 4.

In addtion, qubits that are not explicitly measured, such as ancilla qubits used in intermediate computations can also be interpreted in the same manner. While not explicitly measured at the end of circuits, these qubits could be considered as implicitly measured, with their outcomes being discarded immediately. In this perspective, such qubits fit naturally into the consideration of our paper, as their measurement outcomes do not influence subsequent classical computations.

Several existing works address certain aspects of the matter discussed in this paper. The partial equivalence checking proposed in [4], verifies whether two circuits yield the same probability distribution for a given set of measurement outcomes, but it does not provide a way to simplify circuits while preserving these distributions. Moreover, it requires explicit global unitary operators, the computation of which is infeasible for large-scale circuits due to the inherent complexity of circuit simulation. QuTracer proposed in [16] optimizes circuits by eliminating gates that do not affect a subset of measured qubits, but it lacks a formal framework for this process and may fail to recognize redundant operations—such as SWAP gates that merely permute qubits without altering measurement distributions. In [1], it is mentioned that a measurement outcome depends only on its causal light cone, yet it does not provide a systematic method to exploit this insight for circuit simplification. Crucially, existing approaches overlook a key optimization opportunity: in hybrid quantum-classical workflows, some measurement outcomes become non-contributory to subsequent classical computations.

In this manuscript, we introduce a novel approach to simplify circuits that uses context information from the classical computing components of the hybrid program. By propagating the contextual information that some measurement outcomes are not contributory, our method identifies and removes dead gates in circuits without changing the semantics of the entire hybrid program, leading to more resource-efficient circuits and quantum-classical integration. We evaluate our method by running it on instances of VQE and QPE algorithm, and on random circuits in Section 4.

2 Preliminaries

This manuscript assumes that readers are familiar with the basics of quantum computing. For a detailed introduction to quantum computing, we recommend the following literature [14, 22, 27]. In this section, we explain some notations that we will use later.

For an n-qubit circuit C, we use C.**gates**() to denote the set of all gates in C. Each gate in C.**gates**() is an object storing information, including gate type, the set of qubits it acts on, and the set of gates it depends on. We use C to also denote the unitary matrix of circuit C if no ambiguity is produced. For example, when applying C to a n-qubit state S, the resulting state is CS. We use the following $-$ as an operator to remove one gate from a circuit.

Definition 1 ($-$ operator). *For an n-qubit circuit C and a gate $g \in C$.**gates**(), $C - g$ represents a n-qubit circuit obtained by removing the gate g from C, namely $(C-g)$.**gates**() $:= C$.**gates**()$\setminus\{g\}$.*

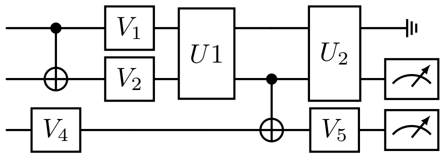
We use the following notation to describe the probability of the measurement outcomes of a subsystem of a quantum state being a given binary string.

Definition 2 (Probability distribution of subsystem measurement outcomes). *For an n-qubit state S_n on a set of qubits $Q_n = \{q_0, \ldots, q_{n-1}\}$, and a binary string k of length $|k| = |Q|$, where Q is a subset of qubits $\{q_{i_0}, \ldots, q_{i_{|k|-1}}\} = Q \subseteq Q_n$ where $i_0 < \cdots < i_{|k|-1}$, $\mathcal{P}^k_{i_0 \ldots i_{|k|-1}}[S_n]$ denotes the probability of q_{i_j} measuring $k[j]$ for all $j \in \{0, \ldots, |k|-1\}$, where $k[j]$ is the j-th element of k.*

Example 2. Consider a 2-qubit state $|\Phi\rangle$ on qubits q_0 and q_1, where $|\Phi\rangle = \alpha_0 |00\rangle + \alpha_1 |01\rangle + \alpha_2 |10\rangle + \alpha_3 |11\rangle$. $\mathcal{P}^{01}_{01}[|\Phi\rangle]$ represents the probability of measuring 0 on q_0 and 1 on q_1, namely the probability of the state collapsing to $|01\rangle$, which is $|\alpha_1|^2$. Similarly, $\mathcal{P}^{10}_{01}[|\Phi\rangle] = |\alpha_2|^2$, $\mathcal{P}^{00}_{01}[|\Phi\rangle] = |\alpha_0|^2$. $\mathcal{P}^{1}_{0}[|\Phi\rangle]$ represents the probability of measuring 1 on q_0, which is $|\alpha_2|^2 + |\alpha_3|^2$, because when q_0 is measured 1, q_1 could be measured either 0 or 1.

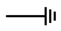
Definition 3 (Frontier). *Given a circuit C, its* frontier *is a set $\mathcal{F}_C$ satisfying: (a) $\mathcal{F}_C \subseteq C$.**gates**(); (b) for any gate $g \in \mathcal{F}_C$, any output wire of g is no input of any other gates.*

Example 3. The frontier of the following circuit only consists of V_5 and U_2.



3 Method

In this work, We restrict our discussion to circuits that contain no mid-circuit measurements or resets. We start by introducing some concepts that we will use in later discussions.

For a quantum circuit C, we assume that for the outcomes we collect by measuring all qubits, a subset of them has no contribution to the classical computing procedures that come later. We explicitly mark such measurement outcomes as discarded, and we call them *discarded measurement outcomes*. The following notation is put at the end of a qubit wire to denote that the measurement outcome on that qubit is discarded:  . On the contrary, a *valid measurement outcome* is the one that is not discarded.

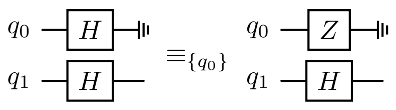
From now on, if a measurement outcome is valid, we omit the symbol of measurement at the end of the qubit wire in the circuit diagram for conciseness.

Definition 4 (Dead/Valid qubit). *A qubit is a* dead qubit *if its measurement outcome is discarded. A qubit is a* valid qubit *if it is not dead.*

Next, we establish an equivalent relation among circuits that is based on measurements performed only on valid qubits. That is, in this equivalence, we consider two circuits to be equal if their probability distributions of valid measurement outcomes are identical.

Definition 5 (Equivalence relative to valid outcomes). *Given two circuits C_1 and C_2 applied on the same set of qubits $Q_n = \{q_0, \dots, q_{n-1}\}$, for a subset $D \subseteq Q_n$ where all qubits in D are dead, C_1 and C_2 are equivalent relative to D, denoted by $C_1 \equiv_D C_2$, if and only if for any n-qubit state S and any binary string k of length $|k| = |Q_n \backslash D|$, $\mathcal{P}^k_{i_0 \dots i_{|k|-1}}[C_1 S] = \mathcal{P}^k_{i_0 \dots i_{|k|-1}}[C_2 S]$, where $Q_n \backslash D = \{q_{i_0}, \dots, q_{i_{|k|-1}}\}$ and $i_0 < \dots < i_{|k|-1}$.*

Example 4. The following two circuits are equivalent relative to their valid outcomes, because the probability of measuring 0 *on the valid qubit, q_1, is the same in both circuits.*

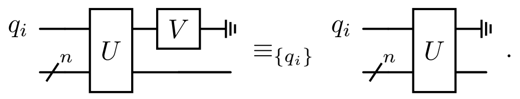


Then, we move on to concepts of dead gates, which are essential to our method. Given the knowledge that some measurement outcomes do not influence subsequent calculations and we discard them explicitly, we define a gate as dead if removing it only affects the probability distribution of these discarded measurement outcomes.

Definition 6 (Dead gate). *Given a circuit C and a gate g in C, and a set of dead qubits D, g is a* dead gate *if and only if $C \equiv_D C' := C - g$, where $-$ is defined by Definition 1.*

Since removing dead gates does not change the probability distribution on valid measurement outcomes, we could simplify circuits by removing such dead gates. By Theorem 1, Theorem 2, and Theorem 3, we present our approach to identify dead gates and prove that removing these dead gates does not influence the results of calculation, therefore justifying the correctness of our method.

Theorem 1. *Given any operator U acting on $n+1$ qubits and any operator V acting on a single qubit q_i, and q_i is dead, it holds that*

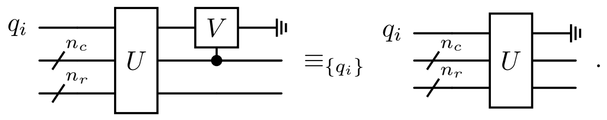


(1)

Proof. This is a special case of Theorem 2. □

Remark 1. By Definition 6, gate V in Eq. (1) is a dead gate, so we could optimize the circuit by removing it.

Theorem 2. *Given any operator U acting on $n+1$ qubits and any operator V acting on a single qubit q_i, and V is controlled by n_c qubits, where $n_c + n_r = n$, and q_i is dead, it holds that*



(2)

Proof. Let the circuit on the left be C_1, and the circuit on the right be C_2. W.l.o.g, we assume that $i=0$, and the gate V is controlled by qubits $q_1, \ldots, q_{n_c}$. Suppose the gate V is defined by $V|0\rangle = \alpha_{v_0}|0\rangle + \beta_{v_0}|1\rangle$ and $V|1\rangle = \alpha_{v_1}|0\rangle + \beta_{v_1}|1\rangle$. For any input state S, we assume that $|\Phi\rangle = C_2 S = \sum_{j=0}^{N-1} c_j |j\rangle$, where $N = 2^{n+1}$, $c_j \in \mathbb{C}$. Then, for any n-bit binary string k, we have $\mathcal{P}^k_{1\ldots n}[|\Phi\rangle] = |c_{0\oplus k}|^2 + |c_{1\oplus k}|^2$, where $\oplus$ is string concatenation (E.g., $00 \oplus 11 = 0011$ and $110 \oplus 1 = 1101$). In fact, $|\Phi\rangle$ could be rewritten as

$$
\begin{aligned}
|\Phi\rangle = \sum_{j=0}^{N-1} c_j |j\rangle = &\sum_{\substack{|s|=n_c,\\ 0\in s}} \sum_{|t|=n_r} c_{0\oplus s\oplus t} |0\oplus s\oplus t\rangle + \\
&\sum_{\substack{|s|=n_c,\\ 0\notin s}} \sum_{|t|=n_r} c_{0\oplus s\oplus t} |0\oplus s\oplus t\rangle + \sum_{\substack{|s|=n_c,\\ 0\in s}} \sum_{|t|=n_r} c_{1\oplus s\oplus t} |1\oplus s\oplus t\rangle + \\
&\sum_{\substack{|s|=n_c,\\ 0\notin s}} \sum_{|t|=n_r} c_{1\oplus s\oplus t} |1\oplus s\oplus t\rangle
\end{aligned}
\tag{3}
$$

So, by applying C_1 to S, the output state $|\Psi\rangle = C_1 S = (C^{n_c}V)C_2 S$ is

$$|\Psi\rangle = C^{n_c}V \ |\Phi\rangle = \sum_{\substack{|s|=n_c,\\ 0\notin s}} \sum_{|t|=n_r} \sum_{b=0}^{1} c_{b\oplus s\oplus t}\alpha_{v_b} |0\oplus s\oplus t\rangle + \sum_{\substack{|s|=n_c,\\ 0\notin s}} \sum_{|t|=n_r} \sum_{b=0}^{1} c_{b\oplus s\oplus t}\beta_{v_b} |1\oplus s\oplus t\rangle + \sum_{\substack{|s|=n_c,\\ 0\in s}} \sum_{|t|=n_r} \sum_{b=0}^{1} c_{b\oplus s\oplus t} |b\oplus s\oplus t\rangle \tag{4}$$

where $C^{n_c}V$ denotes the multi-controlled gate V.
If $\exists l \in \{1,\ldots,n_c\}$: $k[l]=0$, then $\mathcal{P}^k_{1\ldots n}[C_1 S]$ is calculated by

$$\sum_{b=0}^{1} \sum_{s\oplus t=k} |c_{b\oplus s\oplus t}|^2 = |c_{0\oplus k}|^2 + |c_{1\oplus k}|^2 = \mathcal{P}^k_{1\ldots n}[C_2 S] \tag{5}$$

If $\forall l \in \{1,\ldots,n_c\}$: $k[l]=1$, then $\mathcal{P}^k_{1\ldots n}[C_1 S]$ is calculated by

$$\sum_{b=0}^{1} \sum_{s\oplus t=k} |c_{b\oplus s\oplus t}\alpha_{v_b}|^2 + |c_{b\oplus s\oplus t}\beta_{v_b}|^2 = \sum_{b=0}^{1} \sum_{s\oplus t=k} |c_{b\oplus s\oplus t}|^2(|\alpha_{v_b}|^2 + |\beta_{v_b}|^2) \\ = |c_{0\oplus k}|^2(|\alpha_{v_0}|^2 + |\beta_{v_0}|^2) + |c_{1\oplus k}|^2(|\alpha_{v_1}|^2 + |\beta_{v_1}|^2) = \mathcal{P}^k_{1\ldots n}[C_2 S] \tag{6}$$

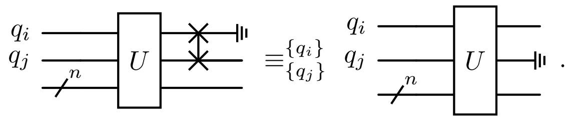
Since our choice of S is arbitrary, by Definition 5, Eq. (2) holds. $\square$

It could happen that removing some gate makes some dead qubits valid and some valid qubits dead, while the probability distribution of valid measurement outcomes is unchanged. For our analysis to encompass this case, we need to extend the equivalence in Definition 5 and the dead gate in Definition 6.

Definition 7 (Extended equivalence relative to valid outcomes). *Given circuits C_1 and C_2 applying on the set of qubits $Q_n = \{q_0,\ldots,q_{n-1}\}$, for $D1, D2 \subseteq Q_n$ where $|D_1| = |D_2|$ and all qubits in $D1$ and D_2 are dead, $C_1 \equiv^{D_1}_{D_2} C_2$ iff for any n-qubit state S and any binary string k of length $|k| = |Q_n \backslash D_1| = |Q_n \backslash D_2|$, $\mathcal{P}^k_{i_0\ldots i_{|k|-1}}[C_1 S] = \mathcal{P}^k_{[e_1/f_1,\ldots,e_m/f_m](i_0\ldots i_{|k|-1})}[C_2 S]$, where $Q_n \backslash D_1 = \{q_{i_0},\ldots,q_{i_{|k|-1}}\}$, $i_0 < \cdots < i_{|k|-1}$, $D_1 \backslash (D_1 \cap D_2) = \{e_1,\ldots,e_m\}$, $D_2 \backslash (D_1 \cap D_2) = \{f_1,\ldots,f_m\}$, and $[b_1/a_1,\ldots,b_p/a_p]s$ denotes a string obtained by for each $l \in \{1,\ldots,p\}$ replacing a_l in string s with b_l (E.g., $[1/4, 2/5, 3/6]456 = 123$).*

Definition 8 (Extended dead gate). *Given a circuit C and a gate g in C acting on a set of qubits Q, g is a dead gate if and only if there exist subsets $D_1, D_2 \subseteq Q$ such that $C \equiv^{D_1}_{D_2} C' = C - g$, where $-$ is defined by Definition 1.*

Theorem 3. *Given any operator U acting on $n+2$ qubits and a SWAP gate, it holds that*

 (7)

Proof. It follows directly the definition of SWAP gates and Definition 7. □

Remark 2. The SWAP gate in Eq. (7) is a dead gate by Definition 8 and can be removed. After removing a SWAP gate, we also need to adapt the qubit mapping, if the qubit mapping/routing is performed at an earlier stage.

Our optimization algorithm is shown in Algorithm 1, of which the asympotic bound is given in Theorem 4.

Algorithm 1: Dead gates removal

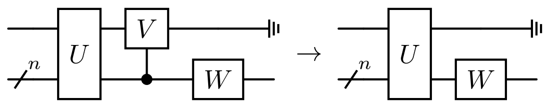
```
Data: $C \in circuits$
Result: $C_{opt}$
$C_{opt} \leftarrow C$, $terminate \leftarrow$ False;
while $terminate \neq True \wedge \emptyset \neq \mathcal{F}_\mathcal{C} \leftarrow C_{opt}.$frontier() do
    $terminate \leftarrow$ True;
    for $g \in \mathcal{F}_\mathcal{C}$ do
        if $g$ is a dead gate by Theorem 1, Theorem 2, or Theorem 3 then
            $C_{opt} \leftarrow C_{opt} - g$, $terminate \leftarrow$ False;
        end
    end
end
```

Theorem 4 (Algorithm 1 is polynomial). *The time complexity of Algorithm 1 is* $\mathcal{O}(|C.\textbf{gates}()|^2)$.

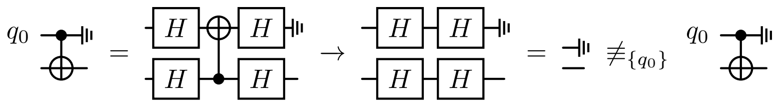
Proof. In each **while** iteration, $\mathcal{O}(|C.\textbf{gates}()|)$-many gates are checked to see whether they are dead, and since at least one gate is removed in every iteration (except the last iteration), there are at most $\mathcal{O}(|C.\textbf{gates}()|)$-many iterations. □

One should be cautious when considering circuit simplification based on the knowledge about non-contributory measurement outcomes. There are cases where it seems that some gate only "writes" on dead qubits and looks like a dead gate, but it is not a dead gate and thus cannot be removed, as demonstrated in the following example.

Example 5. The following simplification would, in general, lead to a changed probability distribution on valid measurement outcomes.



To see this, we consider a special case of it shown as follows:



4 Evaluation

We conduct 3 sets of experiments to evaluate our method. In the first set, our method is applied to the quantum algorithm VQE. In the second set, an instance of QPE is optimized with our method. In the third set, our method is applied to random circuits. The demo implementation of optimization and experiments is accessible at `https://github.com/i2-tum/demo-dead-gate-elimination`.

VQE algorithm The VQE is a hybrid quantum-classical algorithm that finds the ground state energy of a quantum system, and it is widely used in areas like quantum chemistry and material science [12]. In each iteration of the VQE algorithm, an *Ansatz*, a parameterized circuit selected from a diverse range of designs, is executed, and measurements are performed on all output qubits. These measurement outcomes are then scheduled to an *optimizer*, a classical computing procedure. This procedure uses them to calculate expectation values of Hamiltonian terms, which are then used to update parameters in the Ansatz.

Due to the broad range of applications of VQE, it has been integrated into well-established toolchains such as Qiskit, allowing it to be utilized as a black-box subroutine [25]. While this facilitates the use of quantum computers for domain experts, it also introduces the risk of misalignment between quantum circuits and classical computing procedures, particularly because there are already numerous choices of Ansatz with different focuses, and many more are expected to be developed in the future [20, 24, 29].

Consider the instance of the VQE algorithm constructed in Fig. 4. In each iteration, the 4-qubit Ansatz A_1 is executed and measured. The resulting measurement outcomes, o_i ($i \in 0, \dots, 3$), are then sent to the optimizer. There, two expectation values, $\mathbb{E}_Z$ and $\mathbb{E}_X$, are computed and combined. The resulting values are used to update the parameters in the Ansatz—namely, $\overrightarrow{\theta_r}$ and θ_j ($j \in 1, \dots, 8$)—which are adapted for the next iteration.

Here, we observe that the calculation of $\mathbb{E}_Z$ and $\mathbb{E}_X$ only depends on the measurement outcomes o_2 and o_3, meaning q_0 and q_1 are dead qubits. By applying Algorithm 1 to the Ansatz A_1, a simplified Ansatz A_2 shown in Fig. 5 is obtained. Thus, we can optimize the VQE instance by replacing A_1 with A_2, reducing the number of parameterized gates by 4 and the number of two-qubit gates by 3 in each iteration. Considering that VQE requires many iterations to converge, the reduction in gate operations becomes even more significant.

QPE algorithm The QPE algorithm determines the phase associated with an eigenvalue of a given unitary operator [9,15,22]. QPE is employed as a subroutine in various quantum algorithms, among which one of the most famous examples is Shor's algorithm [31].

Consider an instance of QPE constructed in Fig. 6, where the QPE circuit is executed and its measurement outcomes constitute the estimated phase $\theta \in [0, 1)$ received by the classical computing procedure $\mathbf{Proc}_c$. However, we can observe that the most significant bit of θ, namely θ_0, is always subtracted away in the expression $\lambda - \lfloor \lambda \rfloor$. Hence, the initial value of θ_0, i.e., o_0, is not contributory, and

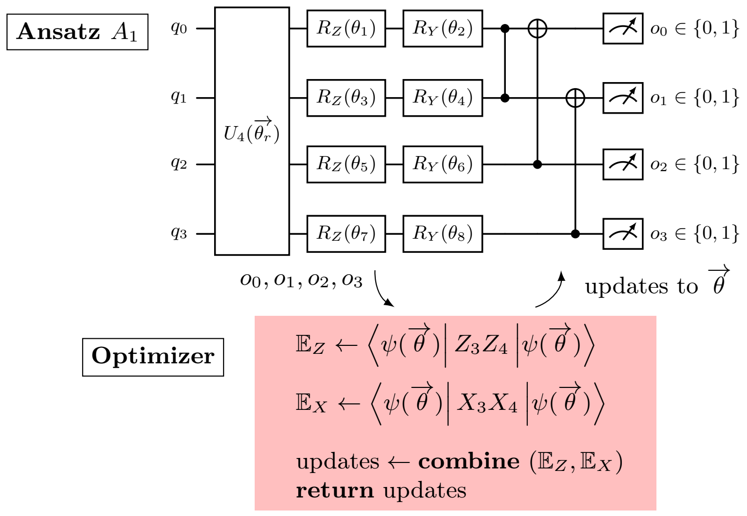

Ansatz A_1
q_0
q_1
q_2
q_3
$U_4(\overrightarrow{\theta_r})$
$R_Z(\theta_1)$ $R_Y(\theta_2)$
$R_Z(\theta_3)$ $R_Y(\theta_4)$
$R_Z(\theta_5)$ $R_Y(\theta_6)$
$R_Z(\theta_7)$ $R_Y(\theta_8)$
$o_0 \in \{0,1\}$
$o_1 \in \{0,1\}$
$o_2 \in \{0,1\}$
$o_3 \in \{0,1\}$
o_0, o_1, o_2, o_3
updates to $\overrightarrow{\theta}$
Optimizer
$\mathbb{E}_Z \leftarrow \left\langle \psi(\overrightarrow{\theta}) \middle| Z_3 Z_4 \middle| \psi(\overrightarrow{\theta}) \right\rangle$
$\mathbb{E}_X \leftarrow \left\langle \psi(\overrightarrow{\theta}) \middle| X_3 X_4 \middle| \psi(\overrightarrow{\theta}) \right\rangle$
updates $\leftarrow$ **combine** $(\mathbb{E}_Z, \mathbb{E}_X)$
return updates

Fig. 4: An instance of VQE algorithm.

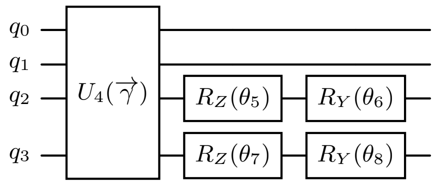

q_0
q_1
q_2
q_3
$U_4(\overrightarrow{\gamma})$
$R_Z(\theta_5)$ $R_Y(\theta_6)$
$R_Z(\theta_7)$ $R_Y(\theta_8)$

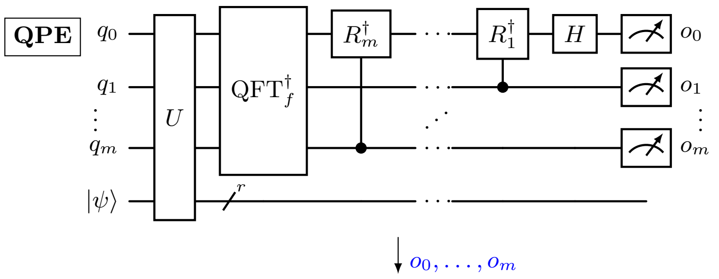
Fig. 5: A simplified Ansatz, A_2, that can replace A_1 in Fig. 4.

we identify q_0 as a dead qubit. Then, by running our optimization on the QPE circuit, we remove m two-qubit gates and a Hadamard gate and get a simplified circuit in Fig. 7. Thus we can optimize the QPE instance by replacing the circuit in Fig. 6 by the circuit Fig. 7.

Random circuits As an effort to ensure a broad and unbiased evaluation of our optimization algorithm, we also conduct experiments where Algorithm 1 is performed on randomly generated circuits. We consider hybrid programs consisting of alternating quantum and classical segments, illustrated as follows:

$$\begin{tikzcd}
\mathrm{QC}_0 \arrow[r] & \mathrm{C}_0 \arrow[r] & \cdots \arrow[r] & \mathrm{QC}_{b-1} \arrow[r] & \mathrm{C}_{b-1}
\end{tikzcd}$$

Each hybrid program consists of a sequence of b quantum-classical blocks, where each block comprises a quantum circuit QC_i followed by a classical computation C_i. The parameter b controls the number of such blocks in the hybrid program. In our evaluation, we set $b = 60$ and generate random circuits of *circuit width* ranging from 2 to 100 qubits, where the circuit width is defined as the number of



Proc$_c$

```
$\theta_0, \ldots \theta_m \leftarrow o_0, \ldots, o_m$
$\theta \leftarrow 0.\theta_0 \ldots \theta_m$
$\lambda \leftarrow 10\theta$
**return** $\lambda - \lfloor \lambda \rfloor$
```

Fig. 6: An instance of QPE, where U consists of Hadamard gates and a sequence of controlled oracles to prepare the state ready for the inverse quantum Fourier transform (QFT [7,22]), $\mathrm{QFT}_f^\dagger$ is the front part of the inverse QFT, measurement outcomes $o_i \in \{0,1\}$ for all i, and $\lfloor \cdot \rfloor$ is the floor function that maps a real value to the greatest integer less than or equal to it.

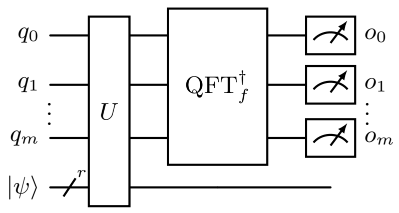


Fig. 7: A simplified QPE circuit to replace the circuit in Fig. 6.

qubits in the circuit. For each circuit, the total gate count is 100 times the circuit width. The circuits are constructed using the universal Clifford + T gate set with single-qubit gates comprising 10% of all generated gates, ensuring a reasonable balance between single- and multi-qubit operations. For each circuit width w, we generate 1000 hybrid programs. In each program, every quantum block QC_i is instantiated with a random circuit generated as described above. All gates are placed uniformly at random across the circuit, ensuring no positional bias in their distribution. These circuits are assumed to be first pre-optimized using circuit transpilers, such as Qiskit and t$|$ket$\rangle$, ensuring that the input circuits are highly optimized by the state-of-the-art compilation toolchain.

We then apply our optimization algorithm to every quantum circuit QC_i within each hybrid program under various settings of dead qubit constraints. Specifically, we consider five settings: 1, 2, or 3 dead qubits, as well as when the number of dead qubits is set to 10% or 20% of the circuit width. For each circuit width and each dead qubit setting, we evaluate the gate count reduction achieved by our algorithm across all circuits in all generated hybrid programs. The mean gate reduction serves as our primary performance metric, providing a robust and comprehensive assessment of the algorithm's effectiveness in realistic hybrid execution scenarios.

The result of our experiments is shown in Fig. 8. The experimental results show that our method consistently removes a non-trivial number of gates across settings. This is particularly notable given that our optimization is applied to circuits that are assumed to have already been pre-optimized using circuit transpilers. This demonstrates that our approach achieves further optimization beyond what is achievable with current state-of-the-art quantum compilation tools.

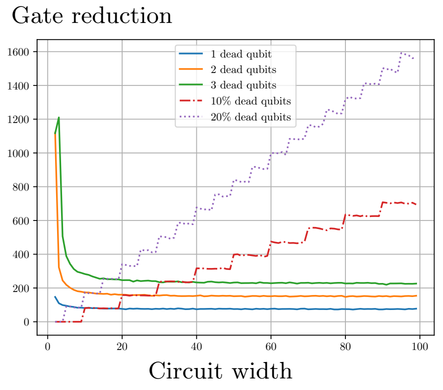
For the settings with a fixed number of dead qubits (1, 2, and 3), we observe that the number of removed gates is initially high when the circuit width is small, and then decreases and stabilizes as the circuit width increases. This trend reflects a key observation: in small circuits, a fixed number of dead qubits represents a large proportion of the total qubit count (e.g., 1 dead qubit out of 2 or 3 is 50%–33%), which creates substantial optimization opportunities. As the circuit grows wider, however, the proportion of dead qubits diminishes, leading to less pronounced impact from the optimization.

The early-stage behavior of settings with a fixed number of dead qubits directly parallels the trends observed in the percentage-based dead qubit settings (10% and 20%). In these settings, the number of removed gates grows stepwise with the circuit width, as the absolute number of dead qubits increases discretely with circuit size. These steps correspond to the increase in dead qubit count, and each jump leads to a corresponding spike in optimization gain. This confirms that our optimization method's effectiveness is primarily driven by the proportion of dead qubits rather than their absolute number alone.

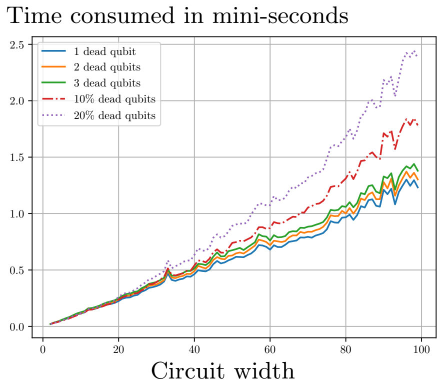
To assess the practical efficiency of our method, we evaluate its runtime as a function of circuit width, as shown in Fig. 8. The results indicate that the execution time increases approximately linearly with the circuit width, even in cases with a significant proportion of dead qubits. This empirical observation suggests that our approach remains efficient in practice. While the asymptotic complexity analysis in Theorem 4 establishes a worst-case quadratic dependence on the number of gates, our experimental results demonstrate that, for realistic circuits, the algorithm exhibits near-linear scaling with circuit width. This suggests that our method is practically efficient and scalable.

5 Related works

The concept of dead gates is inspired by the concept of dead variables in liveness analysis in compiler constructions of classical programming languages [21, 30]. Static analysis stands among the most potential approaches to automate the detection of dead qubits from the context of circuits, and they have proven to be effective in bug detection, program analysis, and circuit optimization [5, 6, 23, 33, 34].



(a)



(b)

Fig. 8: (a) Gate reduction obtained by performing our optimization Algorithm 1 with different dead-qubit settings on hybrid programs embedded with random circuits of circuit width ranging from 2 to 100. (b) The corresponding time consumed in mini seconds.

6 Conclusion and future works

Our method demonstrates a practical approach to optimizing quantum circuits in hybrid programs by taking into account context from the classical host. By identifying dead gates and simplifying circuits accordingly, we achieve significant reductions in gate count when quantum-classical integration is suboptimal. The evaluation of our method on the VQE and QPE instances and on random circuits confirms its potential to improve the quality of circuits. It would be an interesting future work to investigate how to construct the dead gate analysis on dynamic circuits. We think one of the challenges there is that mid-circuit measurements come with side effect even if they are performed on dead qubits.

Acknowledgments. This work is supported by the Bavarian state government through the project Munich Quantum Valley with funds from the Hightech Agenda Bayern Plus.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Abedi, E., Beigi, S., Taghavi, L.: Quantum lazy training. Quantum **7**, 989 (Apr 2023). `https://doi.org/10.22331/q-2023-04-27-989`, `http://dx.doi.org/10.22331/q-2023-04-27-989`
2. Bermot, E., Zoufal, C., Grossi, M., Schuhmacher, J., Tacchino, F., Vallecorsa, S., Tavernelli, I.: Quantum Generative Adversarial Networks For Anomaly Detection In High Energy Physics . In: 2023 IEEE International Conference on Quantum Computing and Engineering (QCE). pp. 331–341. IEEE Computer Society,

Los Alamitos, CA, USA (Sep 2023). https://doi.org/10.1109/QCE57702.2023.00045, https://doi.ieeecomputersociety.org/10.1109/QCE57702.2023.00045

3. Bouland, A., van Dam, W., Joorati, H., Kerenidis, I., Prakash, A.: Prospects and challenges of quantum finance (2020), https://arxiv.org/abs/2011.06492
4. Chen, T.F., Jiang, J.H.R., Hsieh, M.H.: Partial equivalence checking of quantum circuits. In: 2022 IEEE International Conference on Quantum Computing and Engineering (QCE). p. 594–604. IEEE (Sep 2022). https://doi.org/10.1109/qce53715.2022.00082, http://dx.doi.org/10.1109/QCE53715.2022.00082
5. Chen, Y., Fulginiti, I., Mendl, C.B.: Probabilistic Circuit Model. In: 2024 International Conference on Quantum Computing and Engineering (9 2024). https://doi.org/10.1109/QCE60285.2024.10379
6. Chen, Y., Stade, Y.: Quantum constant propagation. In: Hermenegildo, M.V., Morales, J.F. (eds.) Static Analysis. pp. 164–189. Springer Nature Switzerland, Cham (2023), https://link.springer.com/chapter/10.1007/978-3-031-44245-2_9
7. Coppersmith, D.: An approximate fourier transform useful in quantum factoring (2002), https://arxiv.org/abs/quant-ph/0201067
8. De Luca, G.: A survey of nisq era hybrid quantum-classical machine learning research. Journal of Artificial Intelligence and Technology **2**(1), 9–15 (Dec 2021). https://doi.org/10.37965/jait.2021.12002, https://ojs.istp-press.com/jait/article/view/60
9. Dobšíček, M.: Quantum computing, phase estimation and applications. arXiv preprint arXiv:0803.0909 (2008)
10. Elsharkawy, A., To, X.T.M., Seitz, P., Chen, Y., Stade, Y., Geiger, M., Huang, Q., Guo, X., Ansari, M.A., Mendl, C.B., Kranzlmüller, D., Schulz, M.: Integration of quantum accelerators with high performance computing – a review of quantum programming tools (2023), https://arxiv.org/abs/2309.06167
11. Elsharkawy, A., To, X.T.M., Seitz, P., Chen, Y., Stade, Y., Geiger, M., Huang, Q., Guo, X., Ansari, M.A., Ruefenacht, M., Schulz, L., Karlsson, S., Mendl, C.B., Kranzlmüller, D., Schulz, M.: Challenges in hpcqc integration. In: 2023 IEEE International Conference on Quantum Computing and Engineering (QCE). vol. 02, pp. 405–406 (2023). https://doi.org/10.1109/QCE57702.2023.10304
12. Fedorov, D.A., Peng, B., Govind, N., Alexeev, Y.: Vqe method: a short survey and recent developments. Materials Theory **6**(1), 2 (2022)
13. Jojo, J., Khandelwal, A., Chandra, M.G.: Quantum algorithms for tensor-svd (2024), https://arxiv.org/abs/2405.19485
14. Kaye, P., Laflamme, R., Mosca, M.: An introduction to quantum computing. OUP Oxford (2006)
15. Kitaev, A.Y.: Quantum measurements and the abelian stabilizer problem. arXiv preprint quant-ph/9511026 (1995)
16. Li, P., Liu, J., Gonzales, A., Saleem, Z.H., Zhou, H., Hovland, P.: Qutracer: Mitigating quantum gate and measurement errors by tracing subsets of qubits (2024), https://arxiv.org/abs/2404.19712
17. Matondo-Mvula, N., Elleithy, K.: Advances in quantum medical image analysis using machine learning: Current status and future directions. In: 2023 IEEE International Conference on Quantum Computing and Engineering (QCE). vol. 01, pp. 367–377 (2023). https://doi.org/10.1109/QCE57702.2023.00049
18. McCaskey, A., Dumitrescu, E., Liakh, D., Humble, T.: Hybrid programming for near-term quantum computing systems. In: 2018 IEEE International Conference on Rebooting Computing (ICRC). pp. 1–12 (2018). https://doi.org/10.1109/ICRC.2018.8638598

19. McCaskey, A., Dumitrescu, E., Liakh, D., Humble, T.: Hybrid programming for near-term quantum computing systems. In: 2018 IEEE international conference on rebooting computing (ICRC). pp. 1–12. IEEE (2018)
20. McClean, J.R., Romero, J., Babbush, R., Aspuru-Guzik, A.: The theory of variational hybrid quantum-classical algorithms. New Journal of Physics **18**(2), 023023 (feb 2016). https://doi.org/10.1088/1367-2630/18/2/023023, https://dx.doi.org/10.1088/1367-2630/18/2/023023
21. Muchnick, S.: Advanced compiler design implementation. Morgan kaufmann (1997)
22. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press, 1 edn. (Jun 2012). https://doi.org/10.1017/CBO9780511976667
23. Paltenghi, M., Pradel, M.: Analyzing quantum programs with lintq: A static analysis framework for qiskit. Proceedings of the ACM on Software Engineering **1**(FSE), 2144–2166 (2024)
24. Peruzzo, A., McClean, J., Shadbolt, P., Yung, M.H., Zhou, X.Q., Love, P.J., Aspuru-Guzik, A., O'brien, J.L.: A variational eigenvalue solver on a photonic quantum processor. Nature communications **5**(1), 4213 (2014)
25. Qiskit contributors: Qiskit: An open-source framework for quantum computing (2023). https://doi.org/10.5281/zenodo.2573505
26. Quetschlich, N., Forster, T., Osterwind, A., Helms, D., Wille, R.: Towards equivalence checking of classical circuits using quantum computing (2024), https://arxiv.org/abs/2408.14539
27. Rieffel, E., Polak, W.: An introduction to quantum computing for non-physicists. ACM Computing Surveys (CSUR) **32**(3), 300–335 (2000)
28. Rohe, T., Grätz, S., Kölle, M., Zielinski, S., Stein, J., Linnhoff-Popien, C.: From problem to solution: A general pipeline to solve optimisation problems on quantum hardware (2024), https://arxiv.org/abs/2406.19876
29. Romero, J., Babbush, R., McClean, J.R., Hempel, C., Love, P.J., Aspuru-Guzik, A.: Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz. Quantum Science and Technology **4**(1), 014008 (oct 2018). https://doi.org/10.1088/2058-9565/aad3e4, https://dx.doi.org/10.1088/2058-9565/aad3e4
30. Seidl, H., Wilhelm, R., Hack, S.: Compiler Design: Analysis and Transformation. Springer (2012)
31. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Journal on Computing **26**(5), 1484–1509 (1997). https://doi.org/10.1137/S0097539795293172, https://doi.org/10.1137/S0097539795293172
32. Veshchezerova, M., Somov, M., Bertsche, D., Limmer, S., Schmitt, S., Perelshtein, M., Joshi Tripathi, A.: A Hybrid Quantum-Classical Approach to the Electric Mobility Problem . In: 2023 IEEE International Conference on Quantum Computing and Engineering (QCE). pp. 636–641. IEEE Computer Society, Los Alamitos, CA, USA (Sep 2023). https://doi.org/10.1109/QCE57702.2023.00078, https://doi.ieeecomputersociety.org/10.1109/QCE57702.2023.00078
33. Xia, S., Zhao, J.: Static entanglement analysis of quantum programs. In: 2023 IEEE/ACM 4th International Workshop on Quantum Software Engineering (Q-SE). pp. 42–49 (2023). https://doi.org/10.1109/Q-SE59154.2023.00013
34. Zhao, P., Wu, X., Li, Z., Zhao, J.: Qchecker: Detecting bugs in quantum programs via static analysis. In: 2023 IEEE/ACM 4th International Workshop on Quantum Software Engineering (Q-SE). pp. 50–57. IEEE (2023)