

Leveraging hardware-control imperfections for error mitigation via generalized quantum subspace

Yasuhiro Ohkura

March 2023

Master's thesis
Keio University
Graduate School of Media and Governance

Abstract

In the era of noisy intermediate-scale quantum (NISQ) computing, it is essential to suppress noise effects to realize practical quantum computation. Generalized quantum subspace expansion (GSE) method was introduced as a unified framework of virtual distillation (VD), quantum subspace expansion (QSE) and error-extrapolation method. It effectively constructs an error-mitigated quantum state with copies of quantum states with different noise levels. In this work, we further explore the potential of the fault-subspace method by leveraging the hardware-oriented noise, GSE-DA which utilizes the decoherence effect that occurs in processors to suppress the noise. We demonstrate the validity of our proposals via experiments using quantum devices provided by IBM. Compared to competitive techniques, the GSE-DA method shows remarkable performance even with the operational overheads derived from hardware restriction.

Contents

1	Introduction	3
1.1	Background	3
1.2	Research Contribution	3
1.3	Thesis structure	4
2	Preliminaries	5
2.1	Complex Number for Quantum Computing	5
2.1.1	The imaginary unit i and complex plane	5
2.1.2	Euler's Formula	6
2.1.3	Maclaurin Series Expansion	8
2.1.4	Euler's Formula Again	8
2.2	Linear Algebra for Quantum Computing	9
2.2.1	Bra and Ket notations	9
2.2.2	Norm of Vectors	10
2.2.3	Inner Product of Vectors	11
2.2.4	Product of Matrices	13
2.2.5	Tensor Product	13
2.2.6	Transpose / Conjugate / Conjugate Transpose	14
2.2.7	Unitary Matrix (Operator)	14
2.2.8	Hermitian Matrix (Operator)	15
2.2.9	Matrix Exponential Function	15
2.3	Hello Quantum world	16
2.3.1	Quantum bit	16
2.3.2	State vector	16
2.3.3	Quantum Gate	18
2.3.4	Quantum Measurement	19
2.3.5	Observable and Expectation Value	21
2.3.6	Quantum Circuit Model	22
2.3.7	Density Matrix representation	23
2.3.8	Bloch Sphere	24
2.4	Quantum Noise	30
2.4.1	Kraus Operator	30

2.4.2	Examples of Quantum Error Channel	30
2.5	Noisy Intermediate-Scale Quantum (NISQ)	34
2.6	Cloud Quantum Computing	34
3	Literature Review	36
3.1	Quantum Error Mitigation	36
3.1.1	Readout Error Mitigation	36
3.1.2	Quantum Subspace Expansion	38
3.1.3	Virtual Distillation / Error Suppression by Derangement . . .	39
3.2	Generalized Quantum Subspace Expansion	43
3.2.1	Fault subspace	44
4	Methodology	45
4.1	Objective	45
4.2	Fault Subspace Design	45
4.3	Implementation on a Real Device	47
4.4	Evaluation	50
4.5	Noise Models	51
5	Results	52
5.1	Results of Numerical Simulation	52
5.1.1	The Performance of Alternating SWAP	52
5.1.2	The Impact of Stretching Noise factor	53
5.2	Result of Experiments on the Real Device	57
5.3	Scalability	58
6	Conclusions	59
6.1	Summary	59
6.2	Future Work	59
	Acknowledgement	60
A	Variations of techniques for boosting hardware noise	62
A.1	Gate Repetition (GR)	62
A.2	Crosstalk Boost (CB)	62

Chapter 1

Introduction

1.1 Background

In the era of Noisy Intermediate-Scale Quantum (NISQ) computing, as we look towards Fault-Tolerant Quantum Computing (FTQC), to achieve the quantum advantage over the classical counterpart, it is crucial to realize reliable computation in the presence of noise. Several research fields aimed to achieve reliable quantum computation, such as more efficient quantum algorithms, more sophisticated hardware structure and implementation, the software approach to avoid unwanted errors, quantum error correction (QEC), and quantum error mitigation (QEM). The last of these is the main focus of this thesis.

In previous studies, fault-subspace error mitigation as a subclass of the *generalized quantum subspace expansion (GSE)* method was introduced as a noise-agnostic quantum error mitigation technique [48]. It effectively constructs an error-mitigated quantum state by employing multiple copies of the quantum state, each of which has a different level of quantum noise as an extended subspace; it thus allows for efficient suppression of stochastic, coherent, and algorithmic errors. A few approaches exist for the creation of the different levels of quantum noise such as repeat unitary gate or re-scaling Hamiltonian by stretching pulse strength and width to introduce errors in the context of extrapolation methods [12] but have not yet been discussed carefully in the literature.

1.2 Research Contribution

This thesis further explores the GSE-based fault subspace technique to leverage imperfection in hardware control, instead of straightforwardly trying to reduce their intensity. To implement the fault subspace technique on the physical hardware, we propose the *Decoherence Amplification (GSE-DA)* method that increases the intensity of the decoherence effect of the quantum state occurring in time. When it

comes to implementation on real hardware, the operational overhead for resolving hardware restrictions may cause serious performance degradation of quantum computation [34]. To avoid this situation, we also propose an efficient implementation, *Alternating SWAP* technique that helps to resolve this hardware restriction. This thesis demonstrates the performance the validity of our proposals via both numerical simulations that employ the hardware noise profiles reported on IBM’s platform and experiments on a quantum processor provided by IBM. In the experiments, our proposed techniques outperform a competitive technique in both numerical simulation and real-world experiments.

1.3 Thesis structure

The remaining thesis is constructed as follows. In Chapter 2, preliminaries quantum information is provided. In Chapter 3, the current status of related work is explained. In Chapter 4, the practical design of fault subspace employs imperfections in hardware controls is explained. In Chapter 5, details regarding the numerical simulation and experimental results on a real quantum processor are provided. Finally, in Chapter 6, this thesis is concluded with some discussions regarding future work.

Chapter 2

Preliminaries

2.1 Complex Number for Quantum Computing

Complex numbers play an important role in quantum computing and therefore we give a brief introduction.

2.1.1 The imaginary unit i and complex plane

The imaginary unit i is defined as square root of -1 :

$$i = \sqrt{-1} \quad (2.1)$$

$$i^2 = \sqrt{-1} * \sqrt{-1} = -1 \quad (2.2)$$

A complex number z is explained as the sum of a real number x , and an imaginary number iy . Where x is called the real part and y is called the imaginary part, and both x and y are real numbers ($x, y \in \mathbb{R}$):

$$z = x + iy \quad (2.3)$$

As shown in Fig. 2.1, a complex number z can be plotted. In the Cartesian coordinate system, the horizontal axis corresponds to the real part $Re(z)$, and the vertical axis corresponds to the imaginary part $Im(z)$. A complex number always has a *complex conjugate*:

$$z^* = x - iy \quad (2.4)$$

that is also denoted as \bar{z} .

The norm of the complex number is defined as the square root of the product of it and its conjugate:

$$\begin{aligned} |z| &= \sqrt{zz^*} \\ &= \sqrt{(x + iy)(x - iy)} \\ &= \sqrt{x^2 + y^2} \end{aligned} \quad (2.5)$$

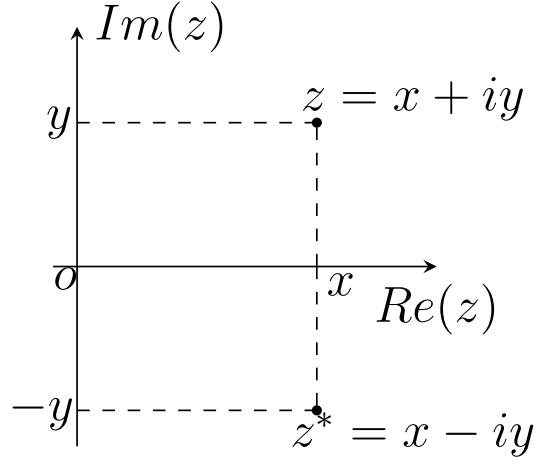


Figure 2.1: **Complex plain on Cartesian coordinate system**

2.1.2 Euler's Formula

We can write a complex number in its polar coordinates

$$\begin{aligned} x &= r \cos \theta \\ y &= r \sin \theta \end{aligned} \tag{2.6}$$

where r is length between z and original point o as shown in Fig. 2.2.

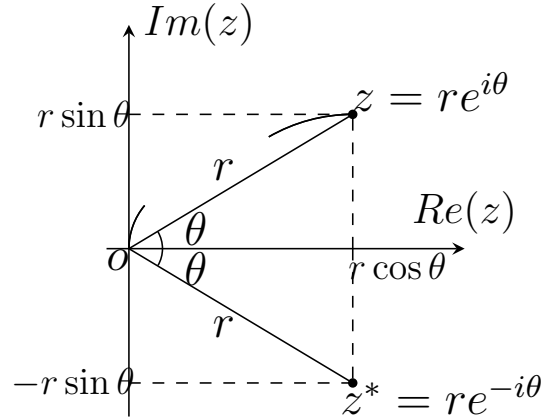


Figure 2.2: **complex plane on Polar coordinate** By applying the polar coordinate system, complex numbers are represented with r , trigonometric functions cosine and sine, and imaginary unit i .

Then let's think about a product of complex numbers z_1 and z_2 :

$$\begin{aligned} z_1 &= x_1 + iy_1 = r_1(\cos \theta_1 + i \sin \theta_1) \\ z_2 &= x_2 + iy_2 = r_2(\cos \theta_2 + i \sin \theta_2) \end{aligned} \tag{2.7}$$

$$\begin{aligned}
z_1 z_2 &= (x_1 + iy_1)(x_2 + iy_2) \\
&= (x_1 x_2 - y_1 y_2) - i(x_1 y_2 + y_1 x_2) \\
&= r_1 r_2 [(\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2) - i(\cos \theta_1 \sin \theta_2 + \sin \theta_1 \cos \theta_2)] \\
&= r_1 r_2 (\cos(\theta_1 + \theta_2) - i \sin(\theta_1 + \theta_2))
\end{aligned} \tag{2.8}$$

This equation uses notations of both the Cartesian coordinates and the polar coordinates. The product of z_1 and z_2 becomes a product of the lengths r_1 and r_2 and the sum of angles θ_1 and θ_2 . This transformation is similar to *the Law of Exponents*:

$$a^m \times a^n = a^{m+n} \tag{2.9}$$

$$a^m / a^n = a^{m-n} \tag{2.10}$$

$$(a^m)^n = a^{mn} \tag{2.11}$$

$$a^n / b^n = (a/b)^n \tag{2.12}$$

$$a^0 = 1 \tag{2.13}$$

$$a^{-m} = 1/a^m \tag{2.14}$$

If the product of complex numbers follows these rules, you can think of the complex number as the exponential function. Let's verify it by using differentiation. By setting a $\cos \theta + i \sin \theta = Z(\theta)$, we have:

$$Z(\theta) = \cos \theta + i \sin \theta \tag{2.15}$$

$$\begin{aligned}
(Z(\theta))' &= -\sin \theta + i \cos \theta \\
&= i(\cos \theta + i \sin \theta) \\
&= iZ(\theta)
\end{aligned} \tag{2.16}$$

therefore we get the same form $Z(\theta)$ except for the factor i . In general, the differentiation of exponential function is defined as $(a^x)' = a^x \log a$, and in the special case of $a = e$, $(e^x)' = e^x$. Therefore:

$$\log a = i, a = e^i \tag{2.17}$$

$$e^{i\theta} = \cos \theta + i \sin \theta \tag{2.18}$$

Eq. (2.18) is called *Euler's formula*. Therefore, the complex number can be rewritten as:

$$z = r e^{i\theta} \tag{2.19}$$

2.1.3 Maclaurin Series Expansion

When the function $f(x)$ is infinitely differentiable, it can be expanded as an infinite sum of the function's derivatives.

$$\begin{aligned} f(x) &= f(0) + f'(0)x + \frac{f''(0)}{2!}x^2 + \frac{f^{(3)}(0)}{3!}x^3 \dots \\ &= \sum_{k=0}^{\infty} f^{(k)}(0) \frac{x^k}{k!} \end{aligned} \quad (2.20)$$

where $f'(0)$ denotes first-order differentiation of $f(0)$, $f''(0)$ is second-order differentiation, and $f^{(3)}(0)$ is 3rd-order differentiation and so forth.

If so, what kind of functions can be differentiable an infinite number of times? Here we show a few example functions:

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \frac{x^{10}}{10!} + \dots \quad (2.21)$$

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \frac{x^{11}}{11!} + \dots \quad (2.22)$$

$$i \sin x = ix - \frac{ix^3}{3!} + \frac{ix^5}{5!} - \frac{ix^7}{7!} + \frac{ix^9}{9!} - \frac{ix^{11}}{11!} + \dots \quad (2.23)$$

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \frac{x^6}{6!} + \frac{x^7}{7!} + \frac{x^8}{8!} + \dots \quad (2.24)$$

$$e^{ix} = 1 + ix - \frac{x^2}{2!} - \frac{ix^3}{3!} + \frac{x^4}{4!} + \frac{ix^5}{5!} - \frac{x^6}{6!} - \frac{ix^7}{7!} + \frac{x^8}{8!} \dots \quad (2.25)$$

2.1.4 Euler's Formula Again

From Eq. (2.21), Eq. (2.23), Eq. (2.25), we can introduce Euler's formula in Eq. (2.18) again:

$$\begin{aligned} &\cos \theta + i \sin \theta \\ &= \left(1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} - \frac{\theta^6}{6!} + \frac{\theta^8}{8!} \dots\right) + \left(i\theta - \frac{i\theta^3}{3!} + \frac{i\theta^5}{5!} - \frac{i\theta^7}{7!} + \frac{i\theta^9}{9!} \dots\right) \\ &= 1 + i\theta - \frac{\theta^2}{2!} - \frac{i\theta^3}{3!} + \frac{\theta^4}{4!} + \frac{i\theta^5}{5!} - \frac{\theta^6}{6!} - \frac{i\theta^7}{7!} + \frac{\theta^8}{8!} + \frac{i\theta^9}{9!} \dots \\ &= e^{i\theta} \end{aligned} \quad (2.26)$$

Now, let's rewrite the complex number in polar coordinates as follows,

$$z = x + iy = r(\cos \theta + i \sin \theta) = re^{i\theta}, \quad (2.27)$$

$$z^* = x - iy = r(\cos \theta - i \sin \theta) = re^{-i\theta}, \quad (2.28)$$

$$(2.29)$$

where

$$\begin{aligned} r &= \sqrt{x^2 + y^2}, \\ \begin{cases} x = r \cos \theta \\ y = r \sin \theta \end{cases} \end{aligned} \quad (2.30)$$

2.2 Linear Algebra for Quantum Computing

In this section, we give a brief introduction to the basic linear algebra necessary for quantum computing. The main object of linear algebra is dealing with *vectors* and *matrices*.

A vector $\vec{v} \in \mathbb{R}^n$ is a list of n real numbers, often written in column form, (r_1, r_2, \dots, r_n) .

$$\vec{v} = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{pmatrix} \quad (2.31)$$

where v is a label, any symbol including a , 1 , ψ , \mathfrak{v} is acceptable.

In quantum mechanics, the quantum states and operators are represented in vector and matrix form respectively. As we explained in Sec. 2.1 in quantum mechanics, we are more interested in complex numbers. A vector $\vec{v} \in \mathbb{C}^n$ means n -dimensional vector where each element is the complex number (c_1, c_2, \dots, c_n) .

$$\vec{v} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} \quad (2.32)$$

The *transpose conjugate* of the vector is defined as a row vector. By this operation, the column elements of the vector are rewritten as rows and each element is a complex conjugate of the original one:

$$(\vec{v}^*)^T = (c_1^* \ c_2^* \ \cdots \ c_n^*) \quad (2.33)$$

2.2.1 Bra and Ket notations

As we mentioned in the previous section, in quantum mechanics we utilize vectors and matrices to represent quantum states and operators. We use the notation called

ket

$$|v\rangle = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} \quad (2.34)$$

and *bra*,

$$\langle v| = (c_1^* \ c_2^* \ \dots \ c_n^*) \quad (2.35)$$

to represent vectors. The bra notation is the transpose of the complex conjugate of ket notation.

In quantum information, we usually represent *qubit* 0 which is the quantum analogue of bit 0, as a vector $|0\rangle$ or an *up spin* of electron $|\uparrow\rangle$, and qubit 1 as a $|1\rangle$ or $|\downarrow\rangle$ *down spin*:

$$|0\rangle = |\uparrow\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad (2.36)$$

$$|1\rangle = |\downarrow\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (2.37)$$

These two vectors are *orthogonal* (Sec. 2.2.3) to each other explained in detail later in Sec. 2.3.1.

2.2.2 Norm of Vectors

The *norm* is the magnitude of the vector, i.e. distance from the origin point to its endpoint. The general definition is:

$$|a\rangle = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}, \text{ then } ||a|| = \sqrt{\langle a|a\rangle} = \sqrt{a_1^2 + a_2^2 + \dots + a_n^2} \quad (2.38)$$

For example, the length of $|a\rangle = \begin{pmatrix} 3 \\ 1 - 2i \end{pmatrix}$ is:

$$||a|| = \sqrt{\langle a|a\rangle} = \sqrt{\begin{pmatrix} 3 & 1 + 2i \end{pmatrix} \begin{pmatrix} 3 \\ 1 - 2i \end{pmatrix}} = \sqrt{14} \quad (2.39)$$

Vectors of norm 1 are called *unit vectors*. In quantum computation, pure states of qubits are represented as unit vectors explained in detail later in Sec. 2.3.1.

2.2.3 Inner Product of Vectors

Inner product is a some what confusing idea. It represents the multiplication of the **norm of one vector** and the **norm of projection of one to another** as shown in Fig. 2.3. An inner product of two vectors \vec{a} and \vec{b} is:

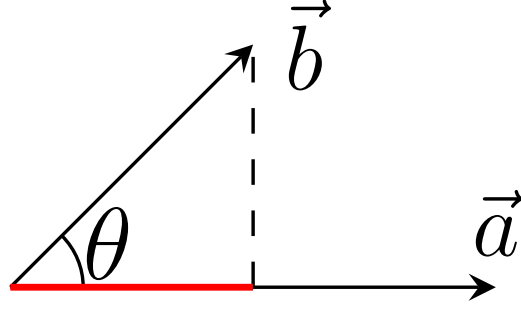


Figure 2.3: **Inner Product between \vec{a} and \vec{b} is equal to $|\vec{a}||\vec{b}|\cos\theta$.** A red line represents projection of vector \vec{b} to \vec{a} , whose norm is $|\vec{b}|\cos\theta$.

$$\vec{a} \cdot \vec{b} = |\vec{a}||\vec{b}|\cos\theta \quad (2.40)$$

where θ is the angle between them.

It can be transformed by using elements of each vector. When two vectors are defined as $\vec{a} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$ and $\vec{b} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$, and angles between \vec{a} and \vec{b} respectively as shown in Fig. 2.4, the inner product between them is:

$$\begin{aligned} \vec{a} \cdot \vec{b} &= |\vec{a}||\vec{b}|\cos\theta \\ &= |\vec{a}||\vec{b}|\cos(\beta - \alpha) \\ &= |\vec{a}||\vec{b}|(\cos\alpha\cos\beta + \sin\alpha\sin\beta) \\ &= \sqrt{a_1^2 + a_2^2}\sqrt{b_1^2 + b_2^2} \left(\frac{a_1}{\sqrt{a_1^2 + a_2^2}} \frac{b_1}{\sqrt{b_1^2 + b_2^2}} + \frac{a_2}{\sqrt{a_1^2 + a_2^2}} \frac{b_2}{\sqrt{b_1^2 + b_2^2}} \right) \\ &= a_1b_1 + a_2b_2 \end{aligned} \quad (2.41)$$

where unit vectors \vec{e}_1 and \vec{e}_2 are orthogonal and vectors \vec{a} and \vec{b} are represented linear combination of these unit vectors.

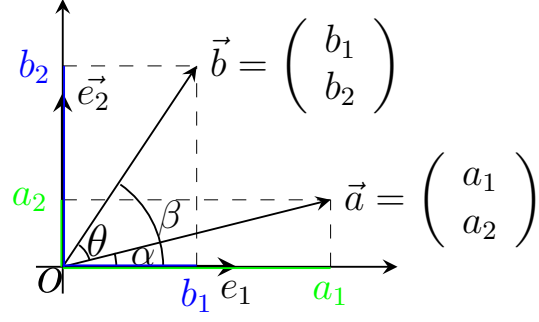


Figure 2.4: **Inner Product between \vec{a} and \vec{b} is equal to $a_1b_1 + a_2b_2$.** Unit vectors \vec{e}_1 and \vec{e}_2 are orthogonal. And vectors \vec{a} and \vec{b} can be represented as a linear combination of these two unit vectors. $\cos \theta = \cos (\beta - \alpha)$.

The general definition using bra-ket notation is:

$$\langle \phi | \psi \rangle = \langle \phi | | \psi \rangle = \begin{pmatrix} \phi_1^* & \phi_2^* & \dots & \phi_n^* \end{pmatrix} \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix} = \sum_{k=1}^n \phi_k^* \psi_k \quad (2.42)$$

The inner product of vectors satisfies the five properties below:

1. **Non-negative**

$$\langle \psi | \psi \rangle \geq 0 \quad (2.43)$$

2. If norm equals 0, then zero vector.

$$\sqrt{\langle \psi | \psi \rangle} = 0 \iff |\psi\rangle = 0 \quad (2.44)$$

3. **Complex conjugate**

$$\langle \psi | \phi \rangle^* = \langle \phi | \psi \rangle \quad (2.45)$$

4. **Linearity**

$$\langle \psi | (a_1 |\phi_1\rangle + a_2 |\phi_2\rangle) = a_1 \langle \psi | \phi_1 \rangle + a_2 \langle \psi | \phi_2 \rangle \quad (2.46)$$

5. Two vectors $|\psi\rangle$ and $|\phi\rangle$ are **orthogonal**, when both are non-zero vector and satisfies:

$$\langle \psi | \phi \rangle = 0 \quad (2.47)$$

2.2.4 Product of Matrices

The product of $N_1 \times N_2$ matrix \mathbf{A}_{N_1, N_2} and $N_2 \times N_3$ matrix \mathbf{B}_{N_2, N_3} is the $N_1 \times N_3$ sized matrix defined as:

$$(\mathbf{AB})_{ij} = \sum_{l=1} \mathbf{A}_{il} \mathbf{B}_{lj} \quad (2.48)$$

Here I show two special cases: when $N_1 = n$, $N_2 = 1$ and $N_3 = n$, it represents product of column vector $|\psi\rangle$ and row vector $\langle\phi|$:

$$|\phi\rangle\langle\psi| = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix} \begin{pmatrix} \phi_1^* & \phi_2^* & \dots & \phi_n^* \end{pmatrix} = \begin{pmatrix} \psi_1\phi_1^* & \psi_1\phi_2^* & \dots & \psi_1\phi_n^* \\ \psi_2\phi_1^* & \psi_2\phi_2^* & \dots & \psi_2\phi_n^* \\ \vdots & \vdots & & \vdots \\ \psi_n\phi_1^* & \psi_n\phi_2^* & \dots & \psi_n\phi_n^* \end{pmatrix} \quad (2.49)$$

and this form is called the *outer product* of vectors.

The other case is when $N_1 = 1$, $N_2 = n$ and $N_3 = 1$, it is the product of row vector and column vector (**inner product**).

In general, the product of two matrices does not *commute* ($\mathbf{AB} \neq \mathbf{BA}$) except for the special cases. When the matrix \mathbf{A} commutes the matrix \mathbf{B} , you can write down the relationship between them as below:

$$[\mathbf{A}, \mathbf{B}] = \mathbf{AB} - \mathbf{BA} = 0 \quad (2.50)$$

2.2.5 Tensor Product

By employing a linear algebra technique called the *tensor product*, we can combine vector spaces together and represent larger vector spaces. For the 1-D matrix (vector) case, the tensor product expands the system as follows:

$$\vec{a} \otimes \vec{b} := \begin{pmatrix} a_i \vec{b} \end{pmatrix} = \begin{pmatrix} a_1 \vec{b} \\ a_2 \vec{b} \\ \vdots \end{pmatrix} \quad (2.51)$$

For the 2-D matrix case, the tensor product expands the system as follows:

$$\mathbf{A} \otimes \mathbf{B} := \begin{pmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \dots \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}. \quad (2.52)$$

2.2.6 Transpose / Conjugate / Conjugate Transpose

Transpose Matrix is a matrix that reflects the elements along with its diagonal. Transpose of a n by m matrix $\mathbf{M}_{n,m}$ is defined as:

$$\mathbf{M}_{n,m}^\top = \begin{pmatrix} \sigma_{1,1} & \cdots & \sigma_{1,n} \\ \vdots & \ddots & \vdots \\ \sigma_{m,1} & \cdots & \sigma_{n,m} \end{pmatrix} \quad (2.53)$$

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & j \end{pmatrix}^\top = \begin{pmatrix} a & d & g \\ b & e & h \\ c & f & j \end{pmatrix} \quad (2.54)$$

Conjugate matrix is a matrix that is obtained by taking the complex conjugate of each element of the original matrix.

$$\mathbf{M}_{n,m}^* = \begin{pmatrix} \overline{\sigma_{1,1}} & \cdots & \overline{\sigma_{n,1}} \\ \vdots & \ddots & \vdots \\ \overline{\sigma_{1,m}} & \cdots & \overline{\sigma_{n,m}} \end{pmatrix} \quad (2.55)$$

$$\begin{pmatrix} a+ix & b & c \\ d & e & f+iy \\ g & h-iz & j \end{pmatrix}^* = \begin{pmatrix} a-ix & b & c \\ d & e & f-iy \\ g & h+iz & j \end{pmatrix} \quad (2.56)$$

When we take both conjugate and transpose of the matrix, which is also known as the dagger of matrix $(M^*)^\top = M^\dagger$.

2.2.7 Unitary Matrix (Operator)

Unitary matrix \mathbf{U} is the square matrix that conjugate transpose (dagger) is also its inverse,

$$\mathbf{U}^\dagger \mathbf{U} = \mathbf{U} \mathbf{U}^\dagger = \mathbf{U} \mathbf{U}^{-1} = \mathbf{I} \quad (2.57)$$

where \mathbf{I} is the *identity matrix* where all diagonal elements are one and all the others are zero. It means when a performing unitary operator just after its dagger operator, these two operations are cancelled out and produce the same vector as before the operations were performed (reversible property). And the unitary operator preserves the norm of a vector when it is applied to complex vectors (unitary property). Those two properties of unitary operation are the important concept of quantum computation.

2.2.8 Hermitian Matrix (Operator)

A $n \times n$ square matrix which equals to hermitian conjugate of itself, is defined as *hermitian matrix*:

$$\mathbf{H}^\dagger = \mathbf{H} \quad (2.58)$$

The *eigenvalue equation* is given as

$$\mathbf{H} |\lambda_i\rangle = \lambda_i |\lambda_i\rangle, \quad (2.59)$$

where $|\lambda\rangle$ are non-zero n vectors called *eigenvectors* and λ_i is called *eigenvalue* an real number corresponding to eigenvector $|\lambda_i\rangle$. Hermitian Matrix is an important asset of quantum information.

2.2.9 Matrix Exponential Function

How can we calculate an exponential function of a square matrix?

$$e^{\mathbf{A}}, \text{ where } \mathbf{A} = \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix} \quad (2.60)$$

This form is called *Matrix Exponential Function*. From the example of Maclaurin series expansion in Eq. (2.25), the usual exponential function of x is expanded as:

$$\begin{aligned} e^x &= 1 + x + \frac{1}{2!}x^2 + \frac{1}{3!}x^3 + \frac{1}{4!}x^4 + \dots \\ &= \sum_{k=0}^{\infty} \frac{1}{k!}x^k \end{aligned} \quad (2.61)$$

In the same way, by the scalar x replacing with a square matrix \mathbf{A} , the matrix exponential function is expanded as follows:

$$\begin{aligned} e^{\mathbf{A}} &= 1 + \mathbf{A} + \frac{1}{2!}\mathbf{A}^2 + \frac{1}{3!}\mathbf{A}^3 + \frac{1}{4!}\mathbf{A}^4 + \dots \\ &= \sum_{k=0}^{\infty} \frac{1}{k!}\mathbf{A}^k \end{aligned} \quad (2.62)$$

which is the same size as the square matrix \mathbf{A} . Matrix exponential functions have properties as below:

1. $e^{\mathbf{0}} = \mathbf{I}$ where \mathbf{I} is an identity matrix
2. $e^{\mathbf{P}^{-1}\mathbf{A}\mathbf{P}} = \mathbf{P}^{-1}e^{\mathbf{A}}\mathbf{P}$

3. $e^{\mathbf{D}} = \begin{pmatrix} e^a & 0 \\ 0 & e^b \end{pmatrix}$, where $\mathbf{D} = \begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix}$ is Diagonal matrix
4. $\frac{d}{dt}e^{\mathbf{A}t} = \mathbf{A}e^{\mathbf{A}t}$
5. If $\mathbf{AB} = \mathbf{BA}$, then $e^{\mathbf{A}}e^{\mathbf{B}} = e^{\mathbf{A}+\mathbf{B}}$,
where \mathbf{A} , \mathbf{B} are the same sized square matrices.

2.3 Hello Quantum world

From this section, we introduce quantum information and its principles.

2.3.1 Quantum bit

The unit of ordinary computation (i.e. classical computation in contrast to the quantum) is called *bit* that can be represented using two distinct states e.g. a head and a tail of a coin, on and off of the switch, true and false, or 0 and 1. The quantum analogue of a bit is called a *quantum bit* or *qubit*.

Here we also introduce a visual representation of the 1-qubit state called the *Bloch sphere*. Even though its mechanics is a little bit complicated and the detail will be explained later in Sec. 2.3.8, this Bloch sphere notation will help your understanding of the unit of quantum computation.

2.3.2 State vector

In physics, the unit of quantum information, the qubit, is represented by utilizing two-level quantum systems, and we have a bunch of physical systems to represent qubits. On the contrary, representation in math is common through any physical system. The quantum state, especially pure quantum states, can be denoted using vector notation, called a state vector. Here we introduce example of one-qubit states,

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (2.63)$$

$$|+\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \quad |-\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \quad (2.64)$$

$$|i\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ i \end{pmatrix} = \frac{|0\rangle + i|1\rangle}{\sqrt{2}}, \quad |-i\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -i \end{pmatrix} = \frac{|0\rangle - i|1\rangle}{\sqrt{2}} \quad (2.65)$$

All these three pairs of state vectors are orthogonal to each other. In a two-level quantum system, such pairs of the orthogonal states are called basis vectors. The

linear combination of basis vectors can represent any quantum state. The general superposition one-qubit state of $|0\rangle$ and $|1\rangle$ basis is represented as follows,

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad (2.66)$$

where $\alpha, \beta \in \mathbb{C}$. When you take an inner product of a quantum state with itself, we get the following expression,

$$\begin{aligned} \langle\psi|\psi\rangle &= (\alpha^* \langle 0| + \beta^* \langle 1|) \alpha |0\rangle + \beta |1\rangle \\ &= \alpha^* \alpha \langle 0|0\rangle + \beta^* \beta \langle 1|1\rangle \\ &= |\alpha|^2 + |\beta|^2 \\ &= 1. \end{aligned} \quad (2.67)$$

This equality shows that a quantum pure state preserves probability. Therefore, we can rewrite Eq. (2.66) as follows:

$$|\psi\rangle = e^{i\gamma} \left(\cos \frac{\vartheta}{2} |0\rangle + e^{i\varphi} \sin \frac{\vartheta}{2} |1\rangle \right), \quad (2.68)$$

where ϑ , φ and γ are the real numbers and we can ignore the factor of $e^{i\gamma}$, because it has no observable effects called *global phase*. The numbers ϑ and φ define a coordinate of point on the unit of three-dimensional sphere called the *Bloch sphere*, which provides useful insight about quantum states, that we will explain it later in Sec. 2.3.8.

We can denote multipartite systems as a tensor product of individual vector spaces. By putting vector spaces $|a\rangle$ and $|b\rangle$ together, we can form larger vector spaces as follows,

$$|a\rangle|b\rangle = |a\rangle \otimes |b\rangle = |ab\rangle \quad (2.69)$$

For instance, we can describe the two-qubits state $|00\rangle$ as,

$$|00\rangle = |0\rangle \otimes |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \times \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ 0 \times \begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (2.70)$$

where the vector representation is expanded to 4 dimension. To generalize the state vector representation, we can describe arbitrary pure quantum states as follows,

$$|\psi\rangle = \sum_i c_i |\psi_i\rangle, \quad (2.71)$$

where $\forall c \in \mathbb{C}$ and $\sum_i |c_i|^2 = 1$, that preserves probability. The way to describe quantum state affected by noise is explained in Sec. 2.3.7.

2.3.3 Quantum Gate

How can we control the quantum states and execute the quantum computation? One important component is quantum gate operation. So far we have introduced how to denote the quantum states as vectors. As we saw in the previous sections, we can modify the state vector by the matrix operation called a *quantum gate*. Here is an example of bit flip of a quantum state by the *Pauli X gate*.

$$\mathbf{X} |0\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle \quad (2.72)$$

Unlike classical computation, the quantum states can have more information such as superposition state (amplitude of each basis), and phase information. That is why we have a variety of quantum gate operators. Here are two more examples, phase flip of arbitrary one qubit state by *Pauli Z gate*.

$$\mathbf{Z}(\alpha |0\rangle + \beta |1\rangle) = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \alpha \\ -\beta \end{pmatrix} \quad (2.73)$$

$$= \alpha |0\rangle - \beta |1\rangle, \quad (2.74)$$

where $|\alpha|^2 + |\beta|^2 = 1$. Pauli **Z** gate flips phase information from positive to negative and so forth. The *Hadamard gate* (**H**) turns the state into superposition.

$$\mathbf{H} |0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \quad (2.75)$$

$$\mathbf{H} |1\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle). \quad (2.76)$$

The interesting fact is that when we apply Hadamard gate to $|0\rangle$ or $|1\rangle$, we get superposition states in both cases but the phase is different. When we apply the Hadamard gate, **Z** gate and Hadamard gate again, we get the same result as bit flip.

$$\mathbf{H}\mathbf{Z}\mathbf{H} |0\rangle = \mathbf{H}\mathbf{Z} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \mathbf{H}\mathbf{Z} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \mathbf{H} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle. \quad (2.77)$$

This equation uses the fact that $\mathbf{H}\mathbf{Z}\mathbf{H} = \mathbf{X}$.

We also introduce an example of a two qubit gate, *Control-NOT (CNOT/CX) gate*. Though the truth table of the family of control gate is a little bit tricky but the rule is pretty simple. If the qubit of the control side is on ($|1\rangle$ state), perform the gate on the target side. For example, if we apply a CX gate to the $|10\rangle$ state, the operation flips the second qubit into $|1\rangle$ as follows,

$$\mathbf{CX} |10\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = |11\rangle \quad (2.78)$$

The quantum gate operation should preserve the norm of vector due to the preservation of probability of quantum states. Which means quantum gates are unitary matrix introduced in Sec. 2.2.7. So when we apply the same quantum gate twice, it cancel outs and do nothing to the state, and which is same as applying *identity gate*. In the Tab. 2.1, we provide a cheat sheet of quantum gates with the circuit diagram we introduce in Sec. 2.3.6.

2.3.4 Quantum Measurement

So far, we've learned how to control quantum state by gate operations. The ideal quantum state evolution is unitary, which means it doesn't interact with the outside world. But eventually, we need to interact with quantum states to know what the state is to complete computation, in other words, observe it. Due to the stochastic nature of the quantum state, we can observe that the quantum state result shows probabilistic behavior. To characterize the stochastic behavior of the quantum state, we understand the results statistically, i.e. repeating the experiments to collect the results (sampling) as a probability distribution. A very simple way to observe the single qubit state is the measurement of a qubit in the computational basis. For a single qubit state, we obtain binary values, +1 or -1 about the probability as a measurement result. For example, when we measure the single qubit state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ in the $|0\rangle$ and $|1\rangle$ basis derived from Pauli Z operator, the probability to obtain +1 and -1 are respectively,

$$\begin{aligned} \text{Prob}\{+1\} &= |\langle 0|\psi\rangle|^2 = |\langle 0|(\alpha|0\rangle + \beta|1\rangle)|^2 = |\alpha|^2 \\ \text{Prob}\{-1\} &= |\langle 1|\psi\rangle|^2 = |\langle 1|(\alpha|0\rangle + \beta|1\rangle)|^2 = |\beta|^2. \end{aligned} \quad (2.79)$$

The middle part of the equation above can be transformed as follows,

$$|\langle 0|\psi\rangle|^2 = \langle \psi|0\rangle \langle 0|\psi\rangle = \langle \psi| \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} |\psi\rangle = \langle \psi| P_0 |\psi\rangle \quad (2.80)$$

where P_0 and P_1 are called measurement operators. In Sec. 2.2.4, we show the outer product as a special case of the dot product in Eq. (2.49). The measurement operators have following properties,

$$\begin{aligned} P_i P_j &= \delta_{ij} P_i \\ P_i^2 &= P_i \\ \sum_i P_i &= I. \end{aligned} \quad (2.81)$$

The binary values we obtain as measurement outcomes are corresponding to the eigenvalues and eigenvectors of an observable. In the example above, the observable is the Pauli Z operator. Here we show the spectral decomposition form of Pauli

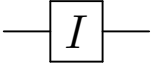
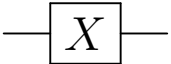


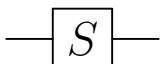
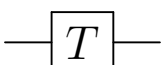
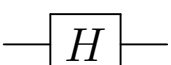
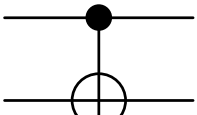
Quantum Gate	Circuit diagram	Matrix representation	Truth table	
			Input	Output
Identity gate		$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$ 0\rangle$ $ 1\rangle$	$ 0\rangle$ $ 1\rangle$
X gate		$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	$ 0\rangle$ $ 1\rangle$	$ 1\rangle$ $ 0\rangle$
Y gate		$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$	$ 0\rangle$ $ 1\rangle$	$i 1\rangle$ $-i 0\rangle$
Z gate		$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	$ 0\rangle$ $ 1\rangle$	$ 0\rangle$ $- 1\rangle$
S gate		$S = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{2}} \end{pmatrix}$	$ 0\rangle$ $ 1\rangle$	$ 0\rangle$ $e^{i\frac{\pi}{2}} 1\rangle$
T gate		$S = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}$	$ 0\rangle$ $ 1\rangle$	$ 0\rangle$ $e^{i\frac{\pi}{4}} 1\rangle$
H gate		$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	$ 0\rangle$ $ 1\rangle$	$\frac{ 0\rangle+ 1\rangle}{\sqrt{2}}$ $\frac{ 0\rangle- 1\rangle}{\sqrt{2}}$
Control-NOT gate		$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	$ 0\rangle 0\rangle$ $ 0\rangle 1\rangle$ $ 1\rangle 0\rangle$ $ 1\rangle 1\rangle$	$ 0\rangle 0\rangle$ $ 0\rangle 1\rangle$ $ 1\rangle 1\rangle$ $ 1\rangle 0\rangle$

Table 2.1: Examples of Quantum Gate [27].

operators as follows,

$$\begin{aligned}\mathbf{Z} &= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = 1 \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + (-1) \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \\ &= 1|0\rangle\langle 0| + (-1)|1\rangle\langle 1| = 1\mathbf{P}_0 + (-1)\mathbf{P}_1\end{aligned}\quad (2.82)$$

$$\mathbf{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = 1|+\rangle\langle +| + (-1)|-\rangle\langle -| \quad (2.83)$$

$$\mathbf{Y} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = 1|i\rangle\langle i| + (-1)|-i\rangle\langle -i| \quad (2.84)$$

where +1 and -1 are the eigenvalues and the following vectors are the eigenvectors. About the observable, we are going to explain the observable in the next section Sec. 2.3.5. Of course we can measure the quantum state in any other basis. For example, we still have other Pauli members, Pauli X basis $|+\rangle$ and $|-\rangle$, and Pauli Y basis $|i\rangle$ and $|-i\rangle$. The probability of obtain +1 and -1 measured in Pauli X basis is as follows,

$$\begin{aligned}\text{Prob}\{+1\} &= |\langle +|\psi\rangle|^2 = \frac{|\alpha + \beta|^2}{2} \\ \text{Prob}\{-1\} &= |\langle -|\psi\rangle|^2 = \frac{|\alpha - \beta|^2}{2}.\end{aligned}\quad (2.85)$$

The probabilities obtained in the Pauli Y basis are,

$$\begin{aligned}\text{Prob}\{+1\} &= |\langle i|\psi\rangle|^2 = \frac{|\alpha + i\beta|^2}{2} \\ \text{Prob}\{-1\} &= |\langle -i|\psi\rangle|^2 = \frac{|\alpha - i\beta|^2}{2}.\end{aligned}\quad (2.86)$$

2.3.5 Observable and Expectation Value

In the physical world, objects can be characterized quantitatively from several points of view, e.g. the position of the object and its momentum. Such a physical quantity is called an *observable*. In quantum physics, an observable is defined as the object of measurement and described as a hermitian matrix.

Here we show some important example observables, the *Pauli operators*, and their spectral decomposed forms in Eq. (2.82). The general hermitian matrix \mathbf{A} is decomposed as follows,

$$\mathbf{A} = \sum_i \lambda_i |\lambda_i\rangle\langle \lambda_i| \quad (2.87)$$

where λ_i is the eigenvalue that is obtained when we measure the observable corresponding to eigenvector $|\lambda_i\rangle$, such that $\langle \lambda_j|\lambda_i\rangle = \delta_{ij}$. When we measure the observable

\mathbf{A} to the quantum state $|\psi\rangle = \sum c_i |\lambda_i\rangle$ decomposed in orthonormal basis $|\lambda_i\rangle$, the expectation value of observable $\langle \mathbf{A} \rangle$ is defined as follows,

$$\begin{aligned}\langle \mathbf{A} \rangle &= \langle \psi | \mathbf{A} | \psi \rangle \\ &= \sum_i c_i^* \langle \lambda_i | \sum_j \lambda_j |\lambda_j\rangle \langle \lambda_j | \sum_k c_k |\lambda_k\rangle \\ &= \sum_i |c_i|^2 \lambda_i\end{aligned}\tag{2.88}$$

Here, we show examples of expectation value of Pauli operators as follows,

$$\begin{aligned}\langle \psi | Z | \psi \rangle &= (\alpha^* \ \beta^*) \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = (+1)|\alpha|^2 + (-1)|\beta|^2 \\ \langle \psi | X | \psi \rangle &= (\alpha^* \ \beta^*) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha^* \beta + \alpha \beta^* = (+1) \frac{|\alpha + \beta|^2}{2} + (-1) \frac{|\alpha - \beta|^2}{2} \\ \langle \psi | Y | \psi \rangle &= (\alpha^* \ \beta^*) \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = i\alpha^* \beta - i\alpha \beta^* = (+1) \frac{|\alpha + i\beta|^2}{2} + (-1) \frac{|\alpha - i\beta|^2}{2}.\end{aligned}\tag{2.89}$$

2.3.6 Quantum Circuit Model

A quantum circuit is a model of quantum computation, that is composed of the quantum register(s) and classical register(s) described as line(s), a series of the quantum gate(s) as box(es), and measurement operation described as a meter connected to classical register(s) Fig. 2.5. Every quantum gate is defined as the unitary operation that means the quantum operation is reversible. On the contrary, the measurement of qubit(s) is a destructive and irreversible operation. It collapses the quantum state and converts it to classical (on/off) form.

In quantum computing, we can utilize special correlations between qubits called entanglement that particular to quantum physics. In general, quantum algorithms are executed as follows. 1). Initialize a qubit state as a superposition. 2). Apply quantum operations to qubits. In the quantum circuit model, those operations have implemented a combination of quantum gates Table 2.1 in time steps. 3). Finally, by applying a measurement operation, we can extract the final state as a classical probability distribution of bit strings.

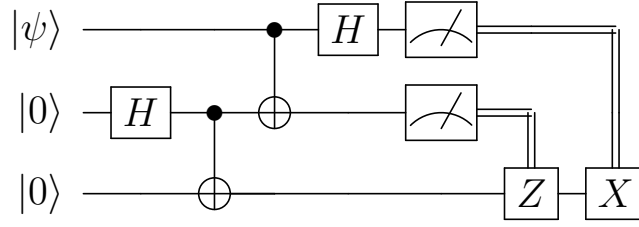


Figure 2.5: **Example of Quantum Circuit: Quantum Teleportation** [23].

2.3.7 Density Matrix representation

So far we have considered the quantum state as a state vector which represents a closed quantum system. However in real-world situations, quantum states are fragile and always exposed to the effects of external noise. We need to discuss then more general ways of representation that can handle commonly encountered scenarios in quantum mechanics. Noise here includes that affects the quantum state, forcing us to represent it in terms of classical probabilities. We consider the quantum state $\{p_i, |\psi_i\rangle\}$ we shall call *ensemble of pure states* that we obtain a state $|\psi_i\rangle$ with a certain probability p_i . Such a state cannot be represented only in the framework of state vectors. For this quantum state, when we measure the observable \mathbf{A} , the expectation value $\langle A \rangle$ is defined as follows:

$$\langle \mathbf{A} \rangle = \sum_i p_i \langle \psi_i | \mathbf{A} | \psi_i \rangle \quad (2.90)$$

We can think of each pure quantum state $|\psi_i\rangle$ as weighted by the classical probability p_i . This quantum state is called a classical mixture. The way to represent this classical mixture more conveniently is the *density matrix representation (density operator)*. The density matrix representation describe the quantum state we discussed above as following

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|. \quad (2.91)$$

This is the reason why we call it as a density "matrix": the quantum state is represented as the sum of the outer product of pure states. The density operator is a positive semi-definite matrix where all the eigenvalues are non-negative when it is spectrally decomposed. When we can represent quantum state without coefficients of classical probability as $\rho = |\psi\rangle\langle\psi|$ it is called *pure state*, other than that we call *mixed state*.

We again defined the expectation value of observable \mathbf{A} as follows:

$$\begin{aligned}
\langle \mathbf{A} \rangle &= \text{Tr} [\rho \mathbf{A}] \\
&= \sum_i p_i \text{Tr} [|\psi_i\rangle\langle\psi_i| \mathbf{A}] \quad (\text{Linear property}) \\
&= \sum_i p_i \sum_j \langle e_j | |\psi_i\rangle\langle\psi_i| \mathbf{A} | e_j \rangle \quad (\text{Definition of trace}) \\
&= \sum_i p_i \sum_j \langle \psi_i | \mathbf{A} | e_j \rangle \langle e_j | \psi_i \rangle \quad (\text{Completeness property : } \sum_j |e_j\rangle\langle e_j| = \mathbf{I}) \\
&= \sum_i p_i \langle \psi_i | \mathbf{A} | \psi_i \rangle.
\end{aligned} \tag{2.92}$$

2.3.8 Bloch Sphere

As the last part of Sec. 2.3, we introduce convenient notation of a quantum bit, the *Bloch sphere*. An arbitrary one qubit state can be considered as the coordinates of a point on a surface or inside of sphere as shown in Fig. 2.6. Bloch sphere gives us

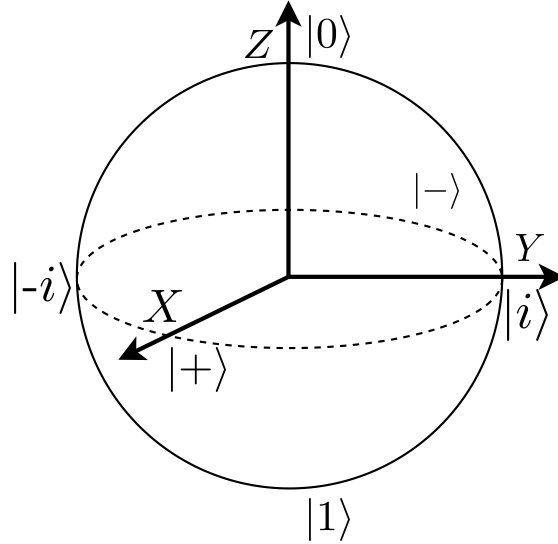


Figure 2.6: **Bloch sphere representation**

a useful insight about quantum state. The north and south poles represent $|0\rangle$ and $|1\rangle$ states respectively, both on the Z axis. At both ends of the X axis there are $|+\rangle$ and $|-\rangle$, and $|i\rangle$ and $|-i\rangle$ on the Y axis. Though visually those pairs of states locate on the ends of same axis, but they are orthogonal mathematically. Fig. 2.7 shows example of the Bloch sphere representation of some basic quantum states.

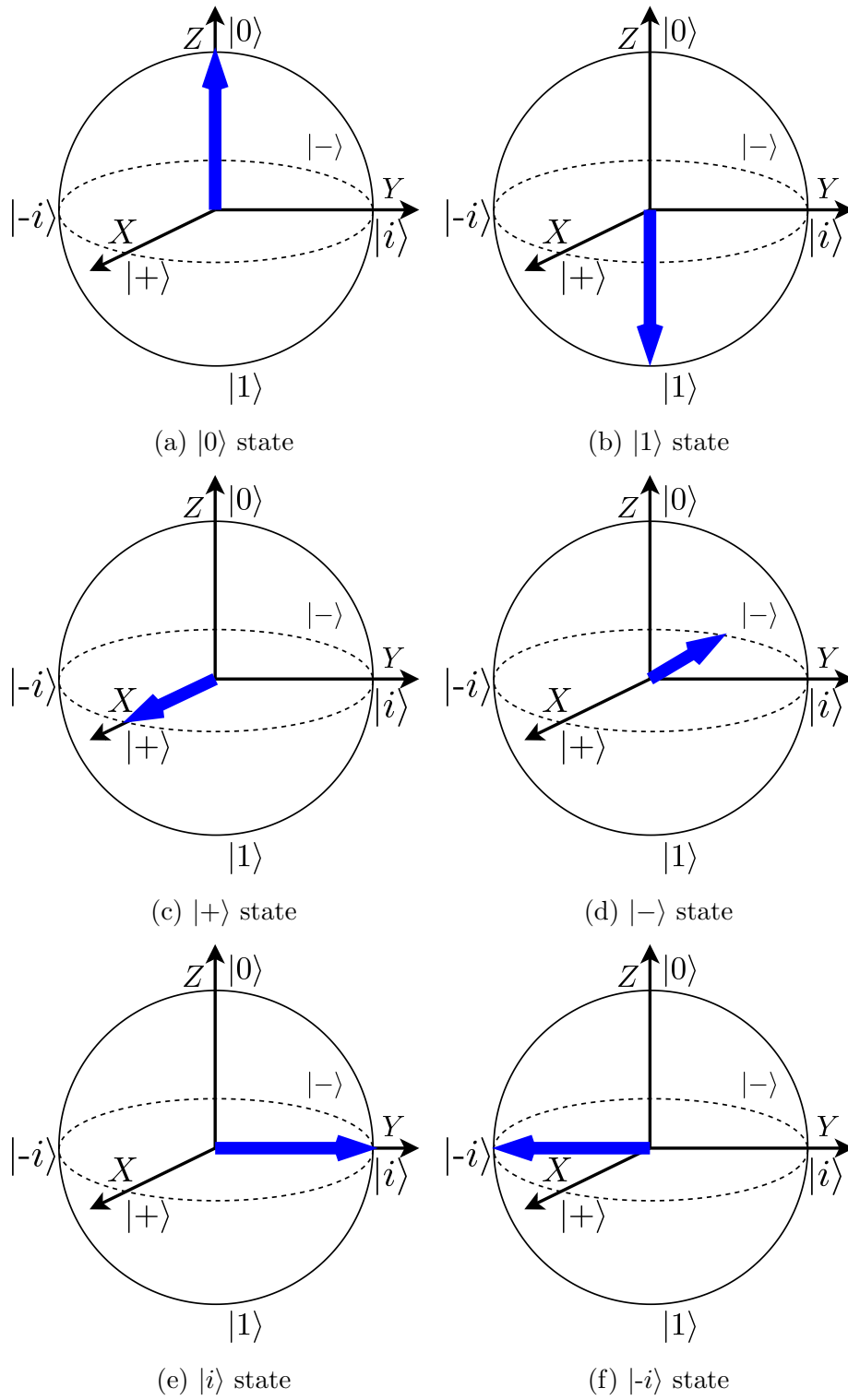


Figure 2.7: **Example of representation of quantum states on Bloch sphere**

As a reminder, as shown in Eq. (2.68), the arbitrary quantum state $|\psi\rangle$ is represented with numbers ϑ and φ which are the coordinates of

$$|\psi\rangle = \cos \frac{\vartheta}{2} |0\rangle + e^{i\varphi} \sin \frac{\vartheta}{2} |1\rangle \quad (2.93)$$

On the Bloch sphere shown in Fig. 2.8a. More general quantum state ρ can be represented as follows:

$$\rho = \frac{\mathbf{I} + rx\mathbf{X} + ry\mathbf{Y} + rz\mathbf{Z}}{2} \quad (2.94)$$

where r is the norm of the sphere and which is defined as 1, and the numbers rx , ry and rz define the point of Bloch sphere as shown in Fig. 2.8b. The number r

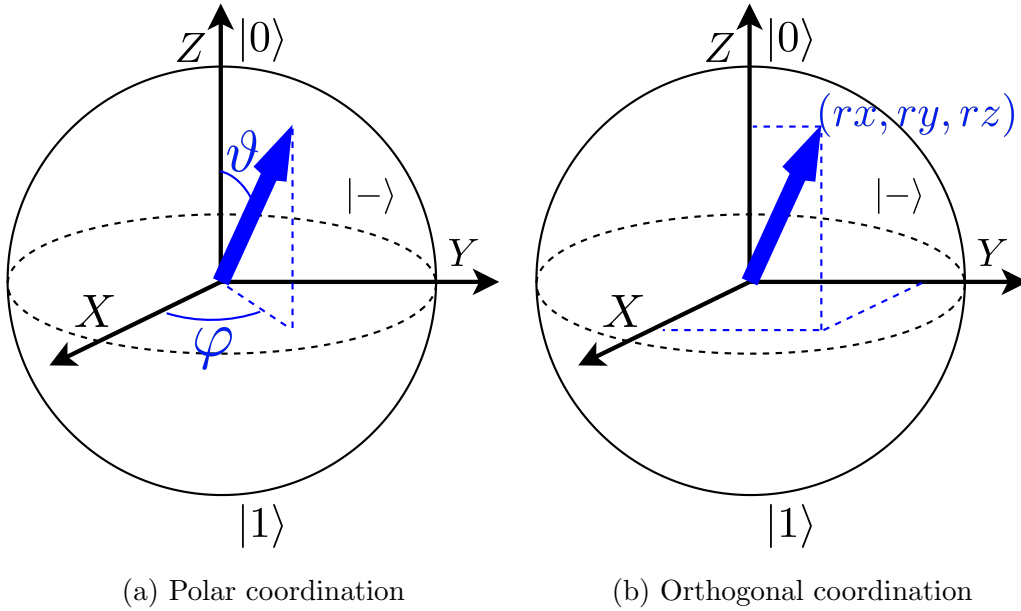


Figure 2.8: **Arbitrary one qubit quantum state**

corresponds to the polar coordinate (r, ϑ, φ) . From now on, we omit r . The numbers represent orthogonal coordinates (x, y, z) are defined as follows:

$$\begin{aligned}
 x &= \text{Tr}[X\rho] = \text{Tr}\left[\frac{\mathbf{X} + x\mathbf{X}^2 + y\mathbf{X}\mathbf{Y} + z\mathbf{X}\mathbf{Z}}{2}\right] \\
 &= \frac{1}{2} \left(\text{Tr} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} + \text{Tr} \begin{pmatrix} x & 0 \\ 0 & x \end{pmatrix} + \text{Tr} \begin{pmatrix} iy & 0 \\ 0 & -iy \end{pmatrix} + \text{Tr} \begin{pmatrix} 0 & iz \\ -iz & 0 \end{pmatrix} \right) \\
 &= x \\
 y &= \text{Tr}[Y\rho] \\
 z &= \text{Tr}[Z\rho]
 \end{aligned} \quad (2.95)$$

By utilizing the above definition of numbers (x, y, z) , for the case of a pure state, we can confirm those two representation of quantum state in polar coordinates and orthogonal coordinates are equal to each other.

$$\begin{aligned}
\rho &= |\psi\rangle\langle\psi| \\
&= (\cos \frac{\theta}{2} |0\rangle + e^{i\varphi} \sin \frac{\theta}{2} |1\rangle)(\cos \frac{\theta}{2} \langle 0| + e^{-i\varphi} \sin \frac{\theta}{2} \langle 1|) \\
&= \cos^2 \frac{\theta}{2} |0\rangle\langle 0| + e^{i\varphi} \cos \frac{\theta}{2} \sin \frac{\theta}{2} |1\rangle\langle 0| + e^{-i\varphi} \cos \frac{\theta}{2} \sin \frac{\theta}{2} |0\rangle\langle 1| + e^{i\varphi} e^{-i\varphi} \sin^2 \frac{\theta}{2} |1\rangle\langle 1| \\
&= \frac{1 + \cos \theta}{2} |0\rangle\langle 0| + \frac{\sin(\theta/2 + \theta/2) + \sin(\theta/2 - \theta/2)}{2} (\cos \varphi + i \sin \varphi) |1\rangle\langle 0| \\
&\quad + \frac{\sin(\theta/2 + \theta/2) + \sin(\theta/2 - \theta/2)}{2} (\cos \varphi - i \sin \varphi) |0\rangle\langle 1| + \frac{1 - \cos \theta}{2} |1\rangle\langle 1| \\
&= \frac{1}{2} [(|0\rangle\langle 0| + |1\rangle\langle 1|) + \sin \vartheta \cos \varphi (|1\rangle\langle 0| + |0\rangle\langle 1|) + i \sin \vartheta \sin \varphi (|1\rangle\langle 0| - |0\rangle\langle 1|) + \cos \theta (|0\rangle\langle 0| - |1\rangle\langle 1|)] \\
&= \frac{I + \sin \vartheta \cos \varphi \mathbf{X} + \sin \vartheta \sin \varphi \mathbf{Y} + \cos \theta \mathbf{Z}}{2}
\end{aligned} \tag{2.96}$$

That leads to the equations representing relationship between polar coordinates and orthogonal coordinates as follows,

$$\begin{aligned}
x &= \sin \vartheta \cos \varphi \\
y &= \sin \vartheta \sin \varphi \\
z &= \cos \vartheta.
\end{aligned} \tag{2.97}$$

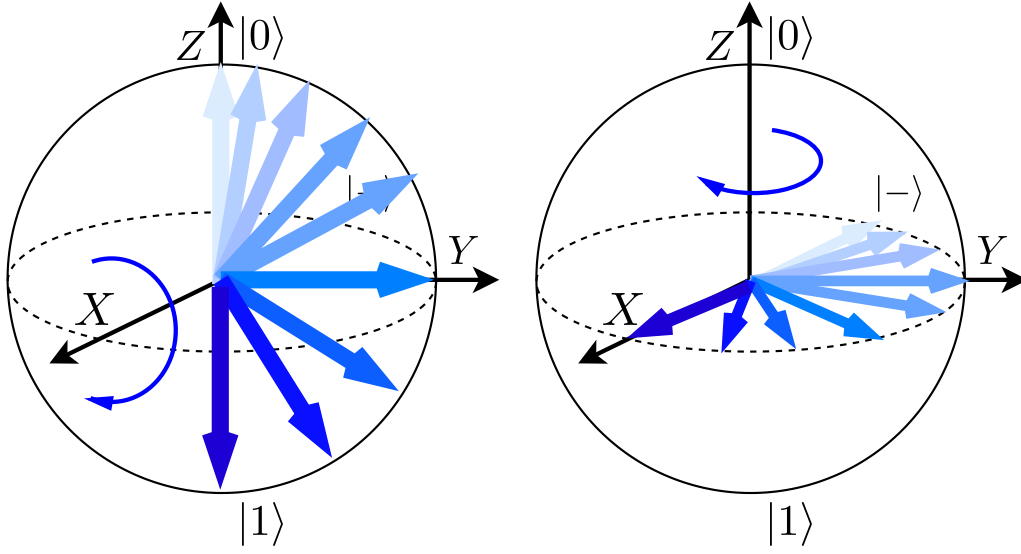
The Bloch sphere can also represent transition of quantum state by quantum gate operations. For example, the transition of quantum state by Pauli operations correspond to the rotation about each axis. Fig. 2.9 shows visual image of example quantum gates. The motion of Hadamard gate is little bit tricky. It rotates quantum state about the axis between Z and X axes shown in Fig. 2.9c. By considering the definition of the unitary gate, we can handle an arbitrary rotation angle in arbitrary axis. The rotation operation along with σ_i is Pauli operator axis can be defined as follows,

$$\begin{aligned}
R(\vartheta)_{\sigma_i} &= e^{-i\frac{\vartheta}{2}\sigma_i} \\
&= \cos \frac{\vartheta}{2} \mathbf{I} - i \sin \frac{\vartheta}{2} \sigma_i.
\end{aligned} \tag{2.98}$$

For example, the operator for rotate ϑ in Z axis can be represented as:

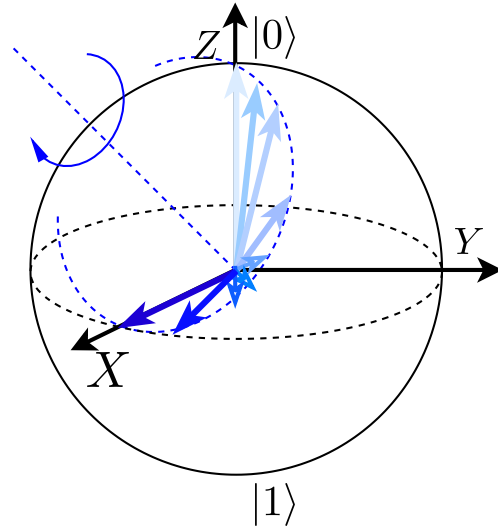
$$e^{-i\frac{\vartheta}{2}\mathbf{Z}} = \begin{pmatrix} e^{-i\frac{\vartheta}{2}} & 0 \\ 0 & e^{i\frac{\vartheta}{2}} \end{pmatrix} \tag{2.99}$$

$$\begin{aligned}
e^{-i\frac{\vartheta}{2}\mathbf{Z}}|+\rangle &= \frac{1}{\sqrt{2}}\left(e^{-i\frac{\vartheta}{2}}|0\rangle + e^{i\frac{\vartheta}{2}}|1\rangle\right) \\
&= e^{-\frac{\vartheta}{2}}\frac{|0\rangle + e^{i\vartheta}|1\rangle}{\sqrt{2}} \\
&= \frac{|0\rangle + e^{i\vartheta}|1\rangle}{\sqrt{2}}
\end{aligned} \tag{2.100}$$



(a) **Rotation about the X axis.**

(b) **Rotation about the Z axis.**



(c) **Hadamard gate.**

Figure 2.9: **Rotation of quantum state** The X gate rotate quantum state from $|0\rangle$ to $|1\rangle$ state through Y axis. The Z gate rotate quantum state from $|-\rangle$ to $|+\rangle$ state in the XY plane. Fig. 2.9c shows move of Hadamard gate from $|0\rangle$ to $|+\rangle$ state. It rotates about an axis between the Z and X axes.

2.4 Quantum Noise

In this section, we describe the formalism of the noise effects on the quantum state. In any real-world situations, a lot of unwanted factors change ideal quantum states into the noisy states. To describe that phenomenon, we employ the concept of a *noise channel* $\mathcal{E}(\rho)$ which maps one quantum state into another ρ' .

2.4.1 Kraus Operator

We can represent the noise channel in a discrete way as the *Kraus operators* \mathbf{E}_k . The Kraus operator can represent the transition of quantum states by discrete operations like gate operation. The Kraus operator is defined as follows,

$$\mathcal{E}(\rho) = \sum_k \mathbf{E}_k \rho \mathbf{E}_k^\dagger \quad (2.101)$$

where $\sum_k \mathbf{E}_k^\dagger \mathbf{E}_k = \mathbf{I}$ and this equation is satisfied by quantum operations which are *trace-preserving*. To describe the continuous noise model, we need to employ the *Lindblad Master equation* that is derived from Schrödinger equation, but we don't explain that equation in this thesis.

2.4.2 Examples of Quantum Error Channel

Here, we introduce examples of quantum noise channels.

The *depolarizing channel* flips quantum state isotropically and uniformly with probability p . That means to replace the state with a completely mixed state $\mathbf{I}/2$ stochastically. The depolarizing channel is defined as follows,

$$\mathcal{E}(\rho) = \frac{p\mathbf{I}}{2} + (1-p)\rho \quad (2.102)$$

We can rewrite the channel in the Bloch sphere representation by using $\frac{\mathbf{I}}{2} = \frac{\rho + \mathbf{X}\rho\mathbf{X} + \mathbf{Y}\rho\mathbf{Y} + \mathbf{Z}\rho\mathbf{Z}}{4}$ and Eq. (2.94) omitting $r = 1$ as follows

$$\begin{aligned} \mathcal{E}_{\text{depol}}(\rho) &= \frac{p\mathbf{I}}{2} + (1-p)\rho \\ &= \frac{p(\rho + \mathbf{X}\rho\mathbf{X} + \mathbf{Y}\rho\mathbf{Y} + \mathbf{Z}\rho\mathbf{Z})}{4} + (1-p)\rho \\ &= \frac{\mathbf{I} + (1-p)x\mathbf{X} + (1-p)y\mathbf{Y} + (1-p)z\mathbf{Z}}{2}. \end{aligned} \quad (2.103)$$

This transition explains depolarizing channel as it loses quantum information in all the axes (X, Y, and Z) of the Bloch sphere with probability p , $(x, y, z) \rightarrow ((1-p)x, (1-p)y, (1-p)z)$. Fig. 2.10a shows a one qubit example in Bloch sphere notation.

The *bit flip channel* flips the state of qubits from $|0\rangle$ to $|1\rangle$ and vice versa with probability p .

$$\mathcal{E}_{\text{bit}}(\rho) = p\rho + (1-p)\mathbf{X}\rho\mathbf{X} \quad (2.104)$$

Note that this channel eventually let loses the quantum information on the X-axis of Bloch sphere notation.

The amplitude damping is time dependent transition that represents the loss of energy of quantum states that decays from the higher energy state $|1\rangle$ to the ground state $|0\rangle$. The energy is transmitted to, or leaked to the environment and lost from the qubit. T_1 is the measure related to this energy dissipation. It is also called *relaxation time*, *thermal relaxation*, *spontaneous emission time*, etc. This process is defined for single qubits as follows

$$\mathcal{E}_{T_1}(\rho) = \mathbf{E}_0^\dagger \rho \mathbf{E}_0 + \mathbf{E}_1^\dagger \rho \mathbf{E}_1 + \mathbf{E}_2^\dagger \rho \mathbf{E}_2 + \mathbf{E}_3^\dagger \rho \mathbf{E}_3 \quad (2.105)$$

and where the four operators \mathbf{E}_k are defined as

$$\begin{aligned} \mathbf{E}_0 &= \sqrt{p} \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{pmatrix} \\ \mathbf{E}_1 &= \sqrt{p} \begin{pmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{pmatrix} \\ \mathbf{E}_2 &= \sqrt{1-p} \begin{pmatrix} \sqrt{1-\gamma} & 0 \\ 0 & 1 \end{pmatrix} \\ \mathbf{E}_3 &= \sqrt{1-p} \begin{pmatrix} 0 & 0 \\ \sqrt{\gamma} & 0 \end{pmatrix} \end{aligned} \quad (2.106)$$

where p represents the population of the excited state of quantum state ρ and $\gamma = 1 - e^{-t/T_1}$ that t is time, and T_1 is some constant that represents the speed of decay of physical qubits. Fig. 2.10b shows how the amplitude damping affects on single qubit state geometrically.

The *dephasing channel* is a time-dependent transition of quantum states. It is also known as *phase damping*, T_2 *dephasing*, e.t.c. It lets the quantum state lose phase information in time. This process is defined for single qubits as follows

$$\mathcal{E}_{T_1}(\rho) = \mathbf{E}_0^\dagger \rho \mathbf{E}_0 + \mathbf{E}_1^\dagger \rho \mathbf{E}_1 \quad (2.107)$$

and where those two operators \mathbf{E}_0 and \mathbf{E}_1 are defined as

$$\begin{aligned} \mathbf{E}_0 &= \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\lambda} \end{pmatrix} \\ \mathbf{E}_1 &= \begin{pmatrix} 0 & 0 \\ 0 & \sqrt{\lambda} \end{pmatrix} \end{aligned} \quad (2.108)$$

where $\lambda = 1 - e^{-t/T_2}$ with t being time, and T_2 some constant represents speed of scattering phase information. Fig. 2.10c shows how the amplitude damping affects on single qubit state geometrically.

Both T_1 and T_2 are critical measures of the quantum devices that characterize the decay of quantum states in time. We can easily say if we don't have any other errors except for decoherence in time, we can correct noise that occurred in the quantum circuit that takes a shorter time than expiring those T_1 and T_2 time.

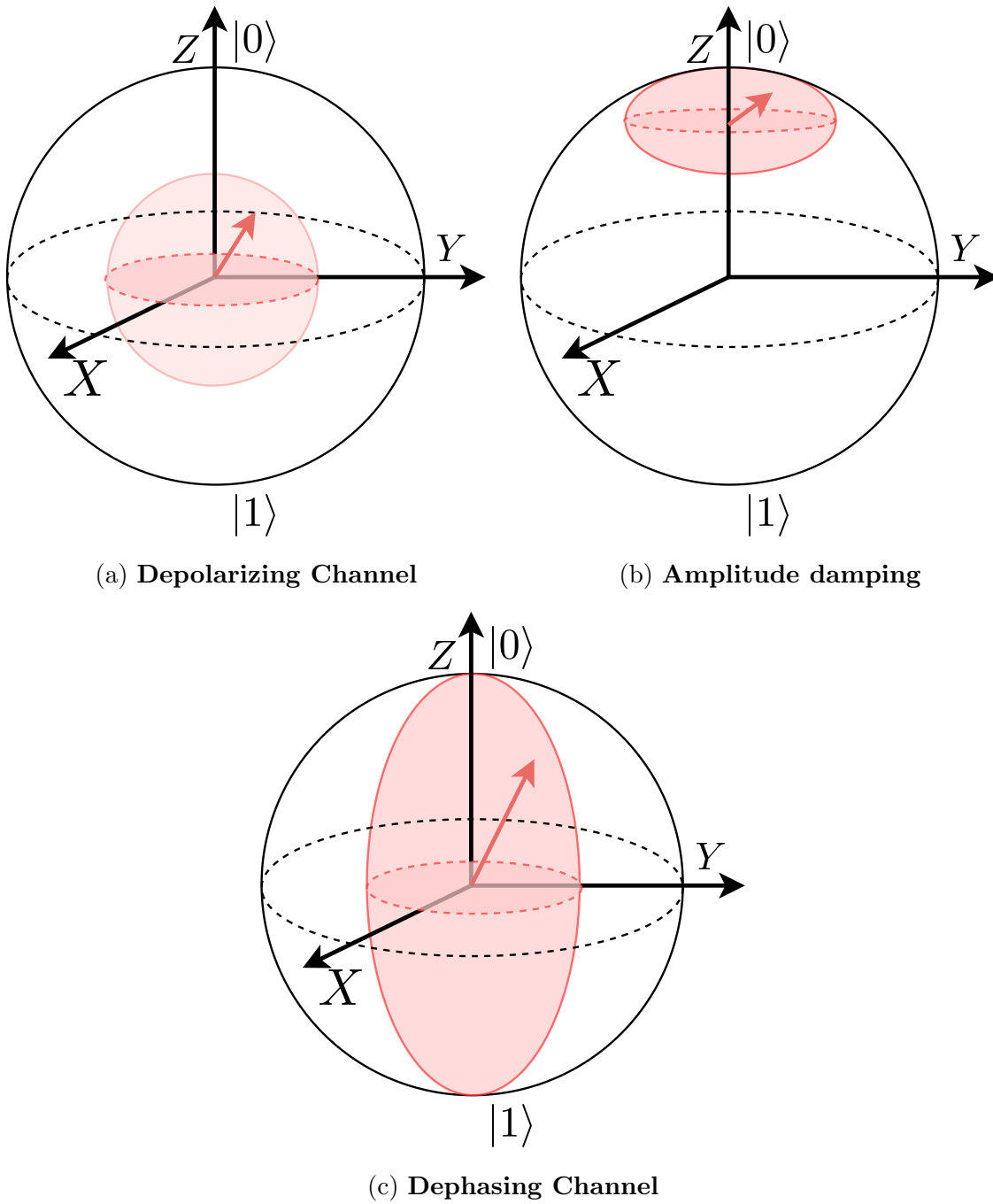


Figure 2.10: **Example of Error Channel:** As the noise affects on quantum state, the state vector shrinks as represented by the red-lined area and loses information. Whereas the depolarizing channel loses information isotropically in the sphere, the others shrink in characteristic ways.

2.5 Noisy Intermediate-Scale Quantum (NISQ)

Current processors, called Noisy Intermediate-Scale Quantum (NISQ) [39], do not have enough capability of error correction, i.e. not immune to noise, which causes a high error rate and greatly affect the reliability of the computation.

Though many quantum algorithms guarantee a so-called quantum advantage in theoretical computational complexity, they require large-scale quantum circuits and reliable computation [5]. With *Quantum Error Correction (QEC)* technique, those promised quantum algorithms including Shor’s algorithm [43], Grover’s algorithm [18] provide the quantum acceleration against classical counterparts. However, QEC requires physical qubits overhead [17], [13]. It is quite hard to succeed in the NISQ era.

With the advent of NISQ processors, there is the development of various NISQ-friendly techniques such as Variational Quantum Eigensolver (VQE) [38], and the *Quantum Error Mitigation* techniques which suit for NISQ processors to suppress noise explained in Sec. 3.1 is in progress. The way towards the quantum computing with capability of QEC, which is called *Fault-Tolerant Quantum Computing (FTQC)*, there are a lots of challenges to improve the performance of quantum processor (hardware), the software, quantum algorithms, theory, integration of Error Correction and Error Mitigation [44] and so on.

2.6 Cloud Quantum Computing

With the advent of cloud quantum systems i.e. quantum computing as a service (QCaaS) [2, 8, 16, 22, 30, 40, 47], many researchers and developers from a variety of domains are becoming quantum users. QCaaS provides the quantum resource that allows opportunities ranging from conducting basic experiments [9] to developing applications that include quantum simulation, quantum machine learning, and optimization [31].

Cloud quantum computing architectures consist of components that include Quantum Processors, Analog Digital control, Real-time control systems, WEB servers and classical databases, the Internet, and End users, as in Fig. 2.11.

Why use a cloud quantum system instead of a local server or a quantum laptop? In general, there are several reasons for the migration of the service from desktop and corporate server rooms to a cloud platform [20, 3]. For the individual users, the total control of the software, OS and low-level utility, and subsequent revisions to other programs comes with a price. For the service providers, the internet-based service can be developed, tested, and operated on the platform provider’s choice instead of coping with various user’s environments. In the case of quantum computation, the development and operation, which includes daily calibration, of a quantum computer are very expensive and specialized tasks [45, 29]. Languages, tools, and environments for the development of the quantum program are still not sufficient. By providing them

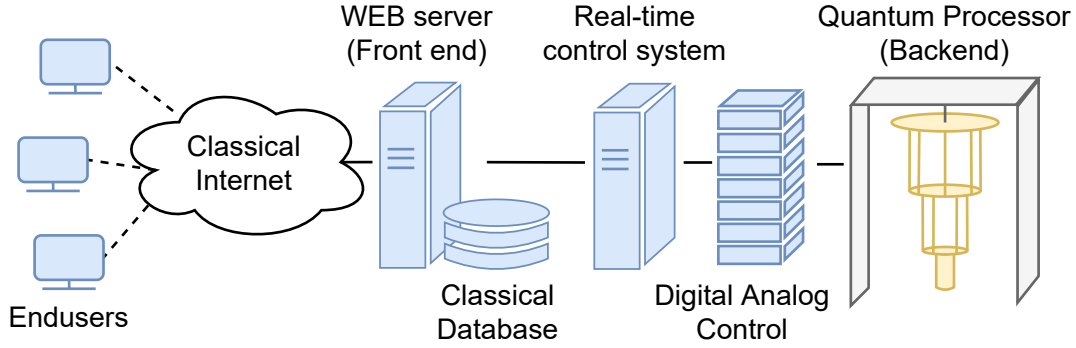


Figure 2.11: **Cloud Quantum Computing Architecture** Various implementations of quantum processors have been proposed [24, 7, 25, 35]. As the core of the service, the quantum processor provides its computing resources by performing quantum operations and measurements. Digital Analog control exchanges the quantum and classical information by converting program instruction into analog signals and measurement results into classical data. The real-time control system is responsible for classical-quantum interaction, mid-circuit measurements, and feed-forward operations. Another end of the cloud computing is the browser-based user interface. End-users create the requests (jobs) on a web browser, send them to the system via a web server through the Internet, and receive the results of the computation. The browser-based user interface provides AAA (authentication, authorization, accountability), and in some cases, quantum programming tool and its development environment, such as a GUI-based quantum circuit composer.

comprehensively as cloud services, users can utilize the quantum resources without being bothered by maintenance.

The rapid increase in users, urgent access for limited quantum resources, and the number of queued jobs are becoming serious issues. To deal with this problem, Ohkura proposed compilation scheme for quantum multi-programming [36].

Chapter 3

Literature Review

3.1 Quantum Error Mitigation

In the era of NISQ computing, it is vital to mitigate the effect of quantum noise in order to achieve meaningful computation. Quantum error mitigation (QEM) is a collection of modern techniques that aim to reduce unwanted computational bias in exchange for additional sampling cost [46, 28, 12, 6].

The four features of the error mitigation technique are:

1. No encoding is required, unlike quantum error correction. In other words, it can be realised with fewer quantum resources.
2. Sampling Cost increase according to the strength of noise and system size as shown in Fig. 3.1.
3. Additional classical and quantum operations are required that can be severe overheads for the performance.
4. There are error-agnostic and error-aware approaches that exist. For instance, following QEM techniques Readout Error Mitigation requires error information for mitigation, but Quantum Subspace Expansion and Virtual Distillation or Error Suppression by Derangement are error agnostic groups.

3.1.1 Readout Error Mitigation

In experiments of the quantum circuit model, quantum measurement results in sampling counts of quantum states along with measurement basis. For instance, measurement results of pure state $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ in Z basis with a number of sampling 1000, is $\{00 : 500, 11 : 500\}$ ideally, (however, it actually fluctuates even in theory because the nature of randomness of the measurement operation). Besides the randomness of measurements that results as the *shot noise*, for the NISQ environment, measurement results are fluctuated by an error occurs in the measurement

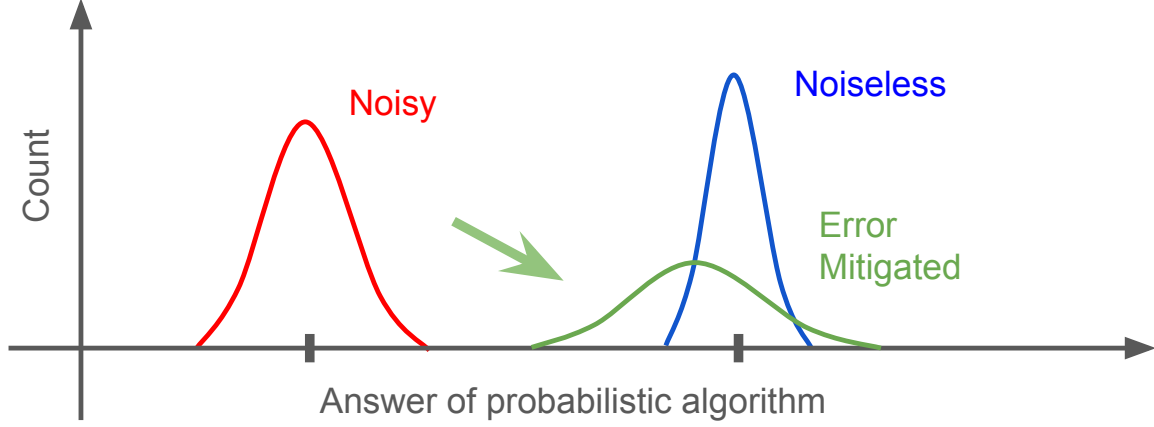


Figure 3.1: **Outline of quantum error mitigation** The vertical axis denotes sampling counts as solutions of quantum computation. QEM techniques aim to fix erroneous results (denoted as red distribution) caused by noises occurring in quantum processors. The modified results (green distribution) have a more significant variance compared to the original results. It means more sampling is required to achieve an original precision.

operation. Unlike general quantum states, the measurement results are explained in as binary manner, you can abstract the phenomena of readout errors as a matrix calculation. Once you obtain the probability $p_{\varepsilon(0 \rightarrow 1)}$ of flipping 0 to 1 and $p_{\varepsilon(1 \rightarrow 0)}$ vice versa, measurement error for the one-qubit system is written as follows:

$$\begin{pmatrix} q_0^{\text{noisy}} \\ q_1^{\text{noisy}} \end{pmatrix} = M_{\varepsilon} \begin{pmatrix} q_0^{\text{ideal}} \\ q_1^{\text{ideal}} \end{pmatrix}, \quad (3.1)$$

$$M_{\varepsilon} = \begin{pmatrix} 1 - p_{\varepsilon(0 \rightarrow 1)} & p_{\varepsilon(1 \rightarrow 0)} \\ p_{\varepsilon(0 \rightarrow 1)} & 1 - p_{\varepsilon(1 \rightarrow 0)} \end{pmatrix} \quad (3.2)$$

where q^{noisy} and q_0^{ideal} are the measurement counts of noisy and ideal case. Instead of applying a matrix with noise characteristics M_ε for obtaining noisy counts, you can get ideal counts by finding inverse matrix M_ε^{-1} ,

$$\begin{pmatrix} q_0^{\text{ideal}} \\ q_1^{\text{ideal}} \end{pmatrix} = M_\varepsilon^{-1} \begin{pmatrix} q_0^{\text{noisy}} \\ q_1^{\text{noisy}} \end{pmatrix}. \quad (3.3)$$

3.1.2 Quantum Subspace Expansion

The *Quantum Subspace Expansion* (QSE) is a technique that allows further explore the Hilbert space in variational simulation. It is often unable to directly prepare the optimal state that minimizes the objective function $\langle \psi(\theta) | H | \psi(\theta) \rangle$ with regard to the parametrized quantum state $|\psi(\theta)\rangle$ in variational methods. The QSE methods explore the optimal state as a linear combination of the set of linearly independent quantum states $\{|\phi_i\rangle\}$ (not necessarily orthogonal) as follows:

$$|\psi_{\vec{c}}\rangle_{\text{opt}} = \sum_i c_i |\phi_i\rangle \quad (3.4)$$

where $\vec{c} = (c_0, \dots)$ are coefficients to be determined by the QSE method to obtain optimal state $|\psi_{\vec{c}}\rangle_{\text{opt}}$. So the problem is to find a set of coefficients \vec{c} that minimize the energy:

$$\min_{c_i c_j^*} \langle \psi_{\vec{c}} | H | \psi_{\vec{c}} \rangle = \min_{c_i c_j^*} \sum_{ij} c_i c_j^* \langle \phi_j | H | \phi_i \rangle, \quad (3.5)$$

such that

$$\begin{aligned} \|\psi_{\vec{c}}\| &= \left\| \sum_i c_i |\phi_i\rangle \right\| \\ &= \sum_{ij} c_i c_j^* \langle \phi_j | \phi_i \rangle \\ &= 1. \end{aligned} \quad (3.6)$$

Though we can directly find a better quantum state with respect to \vec{c} in the manner of a variational scheme, this can be considered a quadratic programming problem (QP). By utilizing methods of Lagrange multiplier

$$\begin{aligned} \mathcal{L} &= \langle \psi_{\vec{c}} | H | \psi_{\vec{c}} \rangle - \lambda (\langle \psi_{\vec{c}} | \psi_{\vec{c}} \rangle - 1) \\ &= \sum_{ij} c_i^* c_j \langle \phi_i | H | \phi_j \rangle - \lambda \left(\sum_{ij} c_i^* c_j \langle \phi_i | \phi_j \rangle - 1 \right) \end{aligned} \quad (3.7)$$

where λ is Lagrange multiplier. By solving the following partial differential equation:

$$\frac{\partial}{\partial \vec{c}} \left(\sum_{ij} c_i^* c_j \langle \phi_i | H | \phi_j \rangle - \lambda \left(\sum_{ij} c_i^* c_j \langle \phi_i | \phi_j \rangle - 1 \right) \right) = 0 \quad (3.8)$$

the solution is given in the form of the following linear eigenvalue problem

$$\tilde{H} \vec{c} = \lambda \tilde{S} \vec{c}, \quad (3.9)$$

where \tilde{H} and \tilde{S} are the square matrices composed of $\tilde{H}_{ij} = \langle \phi_j | H | \phi_i \rangle$ and $\tilde{S}_{ij} = \langle \phi_j | \phi_i \rangle$, \vec{c} is the eigenvector and λ is a set of eigenvalues. The elements of \tilde{H} and \tilde{S} matrices are efficiently computed on a quantum computer with a SWAP test algorithm.

The key to the success of the QSE method is properly choosing the set of states $\{|\phi_i\rangle\}$. The usual way to select the states is based on the best state we can prepare in variational method $|\psi(\theta)\rangle$ before performing QSE.

3.1.3 Virtual Distillation / Error Suppression by Derangement

Stochastic noise channels turn an ideal pure state $\rho_0 = |\psi_0\rangle\langle\psi_0|$ into some noisy mixed state ρ . This mixed state ρ can be written in terms of the spectral decomposition as follows:

$$\rho = p_0 |\psi_0\rangle\langle\psi_0| + \sum_{k=1} p_k |\psi_k\rangle\langle\psi_k| \quad (3.10)$$

where $\langle\psi_i|\psi_j\rangle = \delta_{ij}$, $p_0 \geq p_1 \geq p_2 \geq p_3 \geq \dots \geq 0$ then $|\psi_0\rangle$ is a dominant vector which is closest pure state to ρ in the trace distance. Virtual Distillation (VD) [21] or Error Suppression by Derangement (ESD) [26] can effectively compute the expectation value of an observable from that dominant vector $|\psi_0\rangle$ with exponentially small error using M copies of ρ . Error mitigated state $\rho_{\text{QEM}}^{(M)}$ can be written as follows:

$$\rho_{\text{QEM}}^{(M)} = \frac{\rho^M}{\text{Tr}[\rho^M]} = \frac{1}{p_0^M + \sum_{i=1} p_i^M} \left(p_0^M |\psi_0\rangle\langle\psi_0| + \sum_{i=1} p_i^M |\psi_i\rangle\langle\psi_i| \right). \quad (3.11)$$

Under the assumption that p_0 is strictly greater than else, when increasing the number of copies M , error mitigated state $\rho_{\text{QEM}}^{(M)}$ gets closer to the dominant vector $|\psi_0\rangle$. The expectation value of interest observable O with this purified state is

$\text{Tr} [\rho_{\text{QEM}}^{(M)} O] = \text{Tr} [\rho^M O] / \text{Tr} [\rho^M]$ and this is obtained by separate measurements of $\text{Tr} [\rho^M O]$ and $\text{Tr} [\rho^M]$. In VD/ESD, by using *derangement* operator S to perform cyclic permutation between M copies of ρ , $\text{Tr} [\rho^M O]$ can be measured as:

$$\text{Tr} [\rho^M O] = \frac{1}{M} \sum_m \text{Tr} [S \rho^{\otimes M} O_m] \quad (3.12)$$

The derangement operation S shown as an orange dashed box is constructed based on the SWAP test.

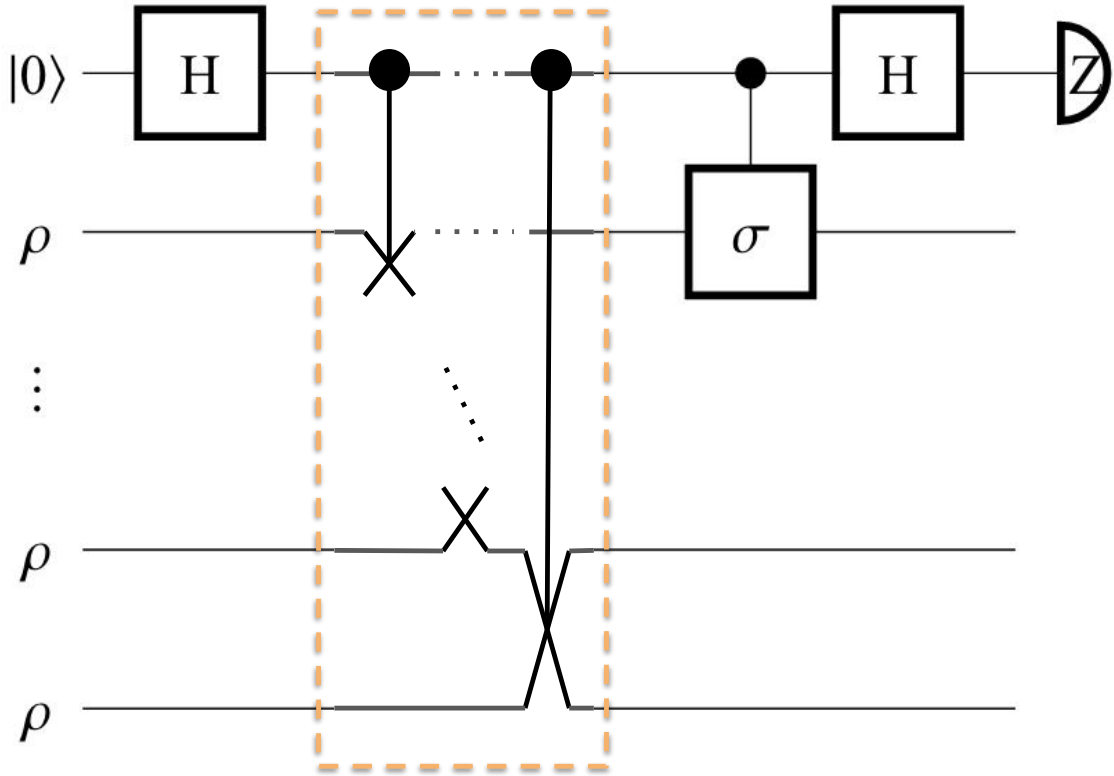


Figure 3.2: **Derangement circuit:** Quantum Circuit for evaluating $\text{Tr} [\rho^M H] = \sum_i \alpha_i \text{Tr} [\rho^M \sigma_i]$ where the hamiltonian $H = \sum_i \alpha_i \sigma_i$.

Fig. 3.2 shows quantum circuit to obtain $\text{Tr} [S \rho^{\otimes M} O_m]$. As with the SWAP test, derangement operation $S^{(M)}$ permute quantum state one by one:

$$\begin{aligned} & \langle \psi_0 \psi_1 \psi_2 \cdots \psi_m | S^{(M)} | \psi_0 \psi_1 \psi_2 \cdots \psi_m \rangle \\ &= \langle \psi_0 \psi_1 \psi_2 \cdots \psi_m | \psi_1 \psi_2 \psi_3 \cdots \psi_0 \rangle \\ &= \langle \psi_0 | \psi_1 \rangle \langle \psi_1 | \psi_2 \rangle \cdots \langle \psi_m | \psi_0 \rangle \\ &= 0 \end{aligned} \quad (3.13)$$

In the case of the density matrix, all the orthogonal states are cancelling out, and then the inner product of the same state vector and powered coefficients p_k^M are left as well as Eq. (3.11).

There is a well-known technique *destructive SWAP test*, a variety of SWAP tests with bell measurement instead of indirect measurement with an ancillary qubit [15]. Compared to performing control SWAP operation, direct measurement requires less quantum gate operation.

This is based on that SWAP operator S can be diagonalized with a bell basis:

$$\begin{aligned} S &= B_S^\dagger D_S B_S \\ &= |\Phi^+\rangle\langle\Phi^+| + |\Psi^+\rangle\langle\Psi^+| - |\Psi^-\rangle\langle\Psi^-| + |\Phi^-\rangle\langle\Phi^-|, \end{aligned} \quad (3.14)$$

where diagonalize matrix B_S is defined as:

$$\begin{aligned} B_S &= \begin{pmatrix} 1/\sqrt{2} & 0 & 1/\sqrt{2} & 0 \\ 0 & 1/\sqrt{2} & 0 & 1/\sqrt{2} \\ 0 & 1/\sqrt{2} & 0 & -1/\sqrt{2} \\ 1/\sqrt{2} & 0 & -1/\sqrt{2} & 0 \end{pmatrix}, \\ D_S &= \text{diag}(1, 1, -1, 1) \end{aligned} \quad (3.15)$$

and all types of bell states are:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle), \quad (3.16)$$

$$|\Phi^-\rangle = \frac{1}{\sqrt{2}} (|00\rangle - |11\rangle), \quad (3.17)$$

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}} (|01\rangle + |10\rangle), \quad (3.18)$$

$$|\Psi^-\rangle = \frac{1}{\sqrt{2}} (|01\rangle - |10\rangle) \quad (3.19)$$

Furthermore, the symmetrized single-qubit observable can be diagonalized as follows:

$$\begin{aligned} S\sigma_\alpha &= B_\alpha^\dagger D_\alpha B_\alpha, \\ D_\alpha &= \text{diag}(1, i, -1, -i), \end{aligned} \quad (3.20)$$

where pauli diagonalize matrices B_α are defined in Fig. 3.3.

This indicates that we must operate unitary B to every qubit pair after all quantum gates in the circuit are executed. In the numerical demonstration, we clarify the influence of errors present in the measurement-associated unitary circuit by comparing the results with and without noise in them.

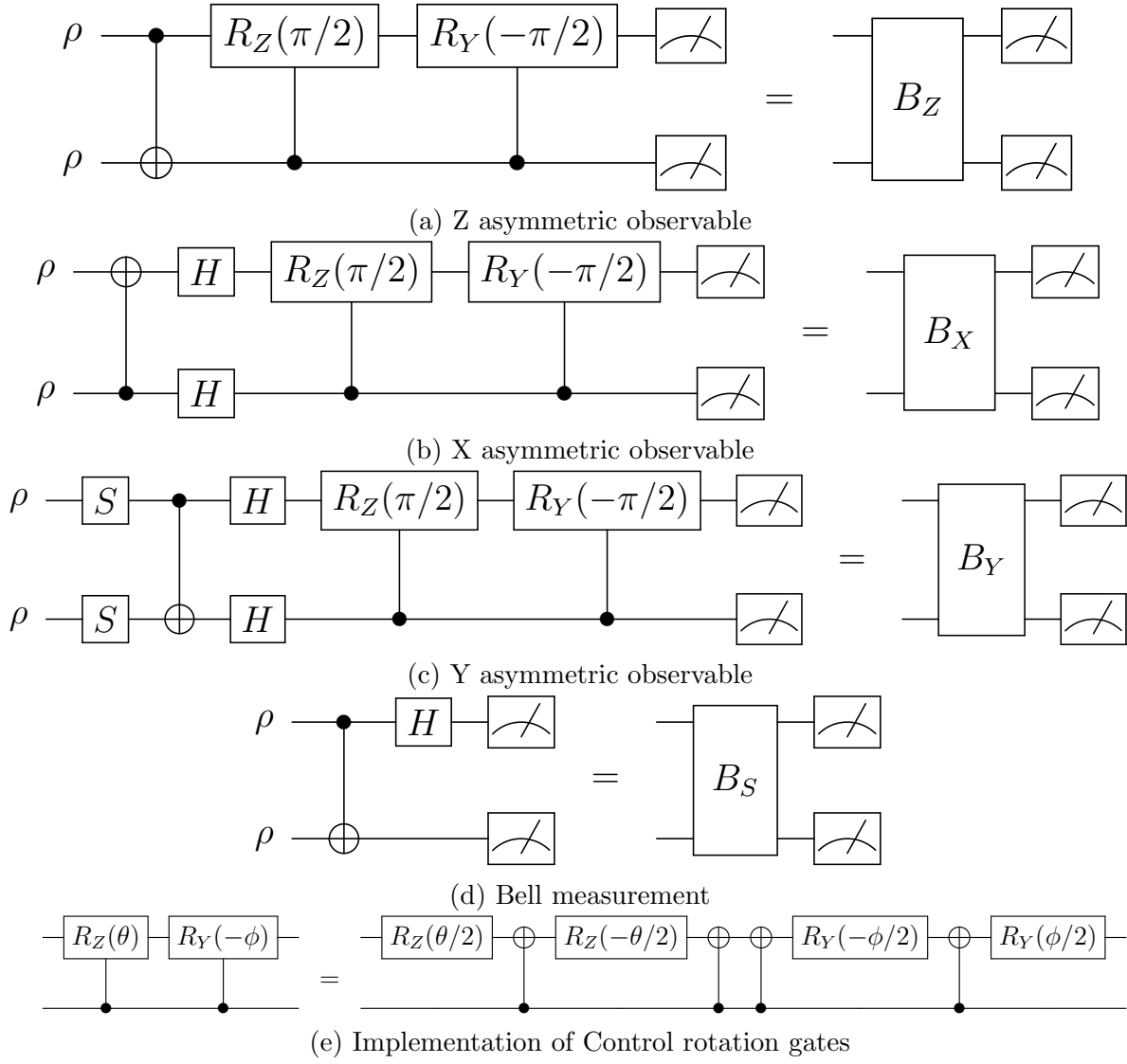


Figure 3.3: **Measuring asymmetric observable**

3.2 Generalized Quantum Subspace Expansion

In the previous study, the fault-subspace error mitigation as a subclass of the *generalized quantum subspace expansion* method was introduced as a noise-agnostic quantum error mitigation technique [48]. It effectively constructs an error-mitigated quantum state by employing multiple copies of the quantum state, each of which has a different level of quantum noise as an extended subspace; it thus allows for efficient suppression of stochastic, coherent, and algorithmic errors.

The GSE method is a variational computation technique to approximate an eigenstate of a given Hamiltonian in the following ansatz via postprocessing:

$$\rho_{\text{GSE}} = \frac{P^\dagger A P}{\text{Tr}[P^\dagger A P]}, \quad (3.21)$$

where $P = \sum_i c_i P_i$ ($c_i \in \mathbb{C}$) is a superposition of generic operators P_i and A is any positive-semidefinite Hermitian operator. By allowing the circuit to execute entangled measurements between multiple quantum states, the bases of the subspace P_i may generally take the following form:

$$P_i = \sum_k \beta_k^{(i)} \prod_{l=1}^{L_k} W_{lk}^{(i)} \rho_{lk}^{(i)} V_{lk}^{(i)}, \quad (3.22)$$

where $\beta_k^{(i)} \in \mathbb{C}$, $\rho_{lk}^{(i)}$ is assumed to be a quantum state, and $W_{lk}^{(i)}$ and $V_{lk}^{(i)}$ are some generic quantum operators. We require that, given fixed coefficients $\vec{c} = \{\dots c_i \dots\}_i$, a physical observable $O = \sum_Q c_Q P_Q$ ($c_Q \in \mathbb{C}$, $P_Q \in \{I, X, Y, Z\}^{\otimes N}$) expressed as a sum of polynomially many Pauli operators can be measured from polynomially many measurement circuits. Concretely, the expectation value of physical observable is given as

$$\langle O \rangle = \frac{\vec{c}^\dagger \tilde{O} \vec{c}}{\vec{c}^\dagger \tilde{S} \vec{c}}, \quad (3.23)$$

where the matrix elements are given as $\tilde{O}_{ij} = \text{Tr}[P_i^\dagger A P_j O]$ and $\tilde{S}_{ij} = \text{Tr}[P_i^\dagger A P_j]$, and we require that such quantities can be efficiently computed using quantum computers.

The coefficients \vec{c} are determined so that the ansatz (3.21) represents an eigenstate of a given Hamiltonian. In the case when we aim for the ground state, this can be understood as minimizing the energy expectation value. By analytically solving a variational principle (Ritz principle), we straightforwardly see that the energy minimization is equivalent to solving the following generalized eigenvalue problem:

$$\tilde{H} \vec{c} = E \tilde{S} \vec{c}, \quad (3.24)$$

where the truncated Hamiltonian is given as $\tilde{H}_{ij} = \text{Tr}[P_i^\dagger A P_j H]$, E denoting the approximated energy.

3.2.1 Fault subspace

Here, the fault-subspace method is defined by

$$P_i = \rho(\varepsilon_i), \tag{3.25}$$

$$A = I/2^N \tag{3.26}$$

where ε_i denotes a set of tunable error rates in the quantum circuit. This choice of the ansatz allows us to represent the error mitigated state $\rho_{\text{GSE}} \propto \bar{\rho}^\dagger \bar{\rho}$ with $\bar{\rho} = \sum_Q c_Q \rho(\varepsilon_Q)$. Note that this includes the virtual distillation (VD) method with $M = 2$ copies as the special case; by taking a single noisy state to span the subspace, we obtain $\rho_{\text{GSE}} = \rho_{\text{VD}} \propto \rho^2$ [26, 21]. As discussed in Ref. [48], by choosing ε_i to be governed by a homogeneous error stretch factor over the entire gate operations, the fault subspace would inherit the advantage of both zero-noise extrapolation method and the VD method as well as overcoming their drawbacks.

Chapter 4

Methodology

4.1 Objective

As explained in Chapter 3, a previous study shows fault-subspace error mitigation techniques effectively construct error-mitigated quantum state by employing multiple copies of the quantum state [48]. However, the effect of noise occurring in a real environment and the overhead of implementation have not yet been discussed carefully.

The GSE fault-subspace technique employs additional quantum noise on the copy state to mitigate the noise effects on the mitigated result similar to commonly used error extrapolation techniques. In a previous study, Yoshioka *et al.* show the performance of the GSE technique on the artificial erroneous environment created by the gaussian noise. In this study, we demonstrate how to employ realistic noises, that have more complex effects on quantum states, and how those noises impact on the performance of GSE techniques.

Another discussion is how efficiently realize the GSE technique in a physically restricted environment. Treating multiple copies of the quantum state on the physical quantum device requires more operational overhead and may lead to catastrophic performance degradation. In this study, we discuss how to implement it efficiently to improve the performance that the derangement operation performs interaction over the copy states by employing many entangling gates shown in Sec. 3.1.3.

4.2 Fault Subspace Design

In this section, we introduce the ***decoherence amplification (DA)*** technique as the hardware-oriented construction of the fault subspace, which is highly practical for experimental demonstration in superconducting qubits. We may occasionally discriminate this as the ***GSE-DA*** method to make the context clear.

The common error source in quantum devices is the qubit lifetime, or the deco-

herence time, of the individual quantum register. The decoherence time is commonly represented by two measures, T_1 and T_2 (or T_2^*) introduced in Sec. 4.5. In this hardware-oriented error boost technique, we intentionally insert buffer time between operations to induce and stretch this decoherence effect. It is simple to control the noise strength we just stretch the buffer (See Fig. 4.1). Note that the DA method does not require any additional hardware parameter tuning, calibration or stretching of the pulse that controls qubits.



Figure 4.1: The decoherence amplification technique. Just before the measurement operation, a certain amount of buffer time is inserted to induce amplitude damping (T_1) and dephasing (T_2) in each qubit.

It is informative to explicitly write down the effect of T_1 and T_2 relaxations. In our numerical simulation, we employ the definition of *thermal_relaxation_error* from Qiskit [1] function as explaining both T_1 and T_2 effects as follows,

$$\mathcal{E}_{T_1 T_2}(\rho) = \mathbf{E}_I^\dagger \rho \mathbf{E}_I + \mathbf{E}_Z^\dagger \rho \mathbf{E}_Z + \mathbf{E}_{\text{res}}^{(0)\dagger} \rho \mathbf{E}_{\text{res}}^{(0)} + \mathbf{E}_{\text{res}}^{(1)\dagger} \rho \mathbf{E}_{\text{res}}^{(1)} \quad (4.1)$$

where four Kraus operators are defined as follows

$$\begin{aligned} \mathbf{E}_I &= \sqrt{p_I} \mathbf{I} \\ \mathbf{E}_Z &= \sqrt{p_Z} \mathbf{Z} \\ \mathbf{E}_{\text{res}}^{(0)} &= \sqrt{p_{\text{res}}} \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \\ \mathbf{E}_{\text{res}}^{(1)} &= \sqrt{p_{\text{res}}} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \end{aligned} \quad (4.2)$$

where $p_{\text{res}} = 1 - e^{-t/T_1}$ is the probability of resetting the qubit to $|0\rangle$ state, $p_Z = (1 - p_{\text{res}})(1 - e^{-t(1/T_2 - 1/T_1)})$ is the probability of applying phase flip to the qubit, and $p_I = 1 - p_{\text{res}} - p_Z$ is the probability of doing nothing.

Note that T_1 and T_2 can be determined by, e.g., measuring the lifetime of the excited state and Hahn echo experiment, respectively. We can estimate the effect of amplified noise from such qubit characteristics.

4.3 Implementation on a Real Device

Before running a program as a quantum circuit on the hardware, the compiler transforms a program circuit to fit the hardware’s native operation. The quantum circuit compilation considers not only the optimization of the quantum circuit structure itself but also constraints of physical limitations of hardware, which usually introduce additional qubits and quantum operations as an overhead. The quantum compiler ultimately aims to minimize this overhead under certain conditions. There are various types of this overhead, for instance, the number of two-qubit gates.

In this study, we consider an implementation that maximizes the fidelity of the proposed error mitigation technique, especially focusing on the depth of CX gates, which is the dominant source of error among the instruction set of a device provided by IBM Quantum. The total number of CX gates G_{tot} in the pre-compiled quantum circuit is

$$G_{tot} = G_{VQE} + G_{derange} \quad (4.3)$$

where G_{VQE} and $G_{derange}$ are the number of CX gates required to implement the VQE and derangement operation of the VD technique.

When considering the compilation phase, a quantum circuit is described as a graph, and the routing task transforms it to fit hardware topology.

In this study, we consider the linearly connected physical qubit on the quantum processor which is the case of the most sparse physical connection. In such a case, the quantum circuit compiler adds some complex entangling operations as an overhead to resolve the qubit routing, and then that may cause catastrophic performance degradation. IBM Kawasaki the quantum processor we use for evaluation has a relatively sparse topology as shown in Fig. 4.3 and it is quite important to resolve this qubit routing problem efficiently to increase the performance.

We show the mapping and routing technique that hierarchically performs SWAP operations to rearrange qubits, which we call ***alternating SWAP***. This technique was developed to reduce operational overheads in the case of the most sparse two-qubit connection, i.e. linearly connected topology. The procedure of this technique is described as follows:

- Step 1. We perform operations to construct target quantum state ρ next to each other on physical qubits.
- Step 2. Perform an alternating SWAP operation before the derangement to rearrange the copy quantum states side by side,
- Step 3. Then, perform derangement and measurement operations on two qubits pairs (red and blue in Fig. 4.2) which is the component of each copy state.

The overhead of the alternating SWAP method depends on the size of the system qubits. Fig. Fig. 4.2 The total number of CX gates is

$$G_{tot} = G_{VQE} + \sum_n^{N_Q} (n-2)G_{\text{swap}} + G_{\text{derange}} \quad (4.4)$$

where N_Q is size of prepared quantum state ρ .

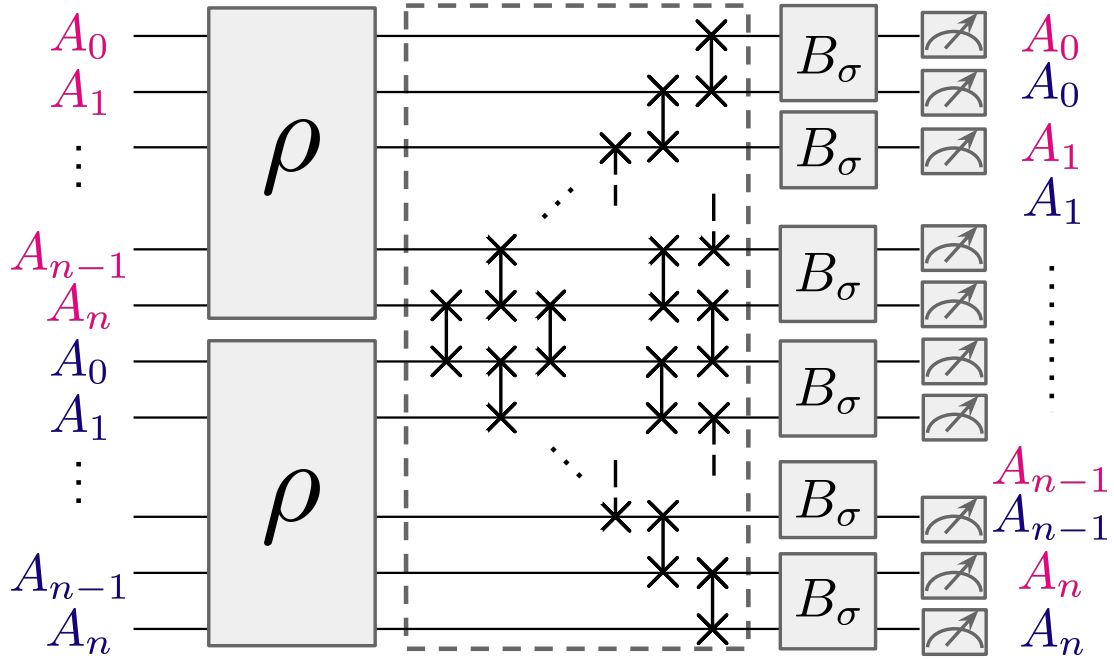


Figure 4.2: Alternating SWAP

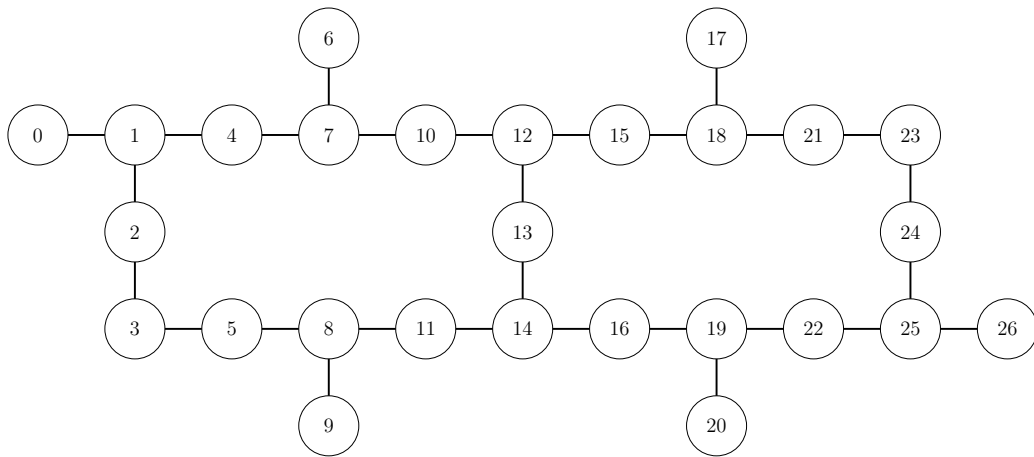


Figure 4.3: **Topology of IBM Kawasaki** Each qubit has a label for convenience. In the experiments, we map qubits 3, 5, 8 for the first copy of state ρ , and 11, 14, and 16 for the second copy of the state.

4.4 Evaluation

To evaluate, we conducted both numerical simulations and experiments on the real device to inspect the performance of our proposal GSE-DA technique.

We prepare the one-dimensional transverse-field Ising (1d TFI) model, whose Hamiltonian is given as $H = -\sum_r Z_r Z_{r+1} + h \sum_r X_r$ where X_r and Z_r denote the x-and z-components of the Pauli matrix acting on the r -th site and h is the amplitude of the transverse magnetic field. We set $h = 1$ in the following. First, we prepare parameters of variational circuit to construct quantum state ρ that reflects ground-state energy via the VQE algorithm on noiseless quantum simulations as a baseline. We run that variational circuit with obtained parameters on both simulators with noise models and real devices to calculate noisy value denoted as a red line in the figures in Chapter 5.

From the viewpoint of scalability, we prepared several sizes of variational circuits as shown in Fig. 4.4.

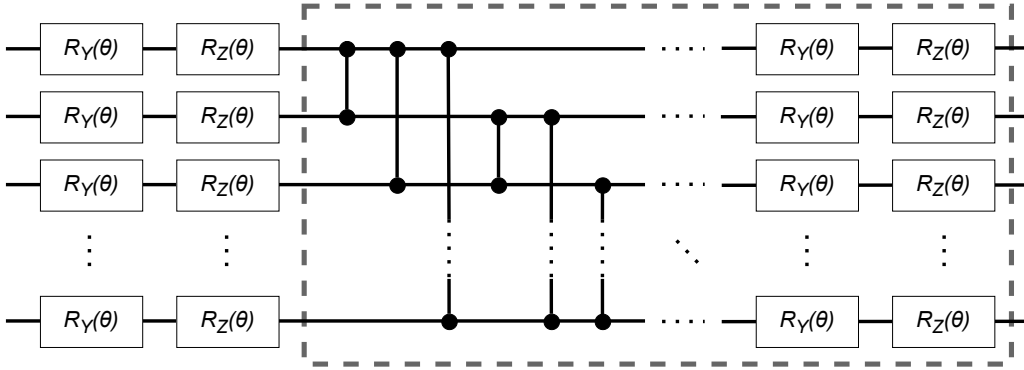


Figure 4.4: **Variational Ansatz:** The structure of the variational ansatz is considered in our demonstration. We employ the structure of a fully-entangled circuit, in which CZ gates construct entangling layers on all the two-qubit pairs of the state. It repeats d times that the gates inside the dashed line where d refers to the depth of entangling layers. In the experiments, we use a 3-qubit sized variational circuit, depth from 1 to 10 both in simulation and real device experiments, and 5, 7, and 9 qubits sized circuits with depth 1 to 5.

In the entire experiment, we focus on the precision of the error-mitigated ground-state energy $\text{tr}[\rho_{\text{GSE}} H]$. In the figures, we plot the followings:

- (I) Exact ground energy (**Exact**), as gray dashed lines,
- (II) Energy gained from a noisy quantum circuit with rotation angles given by VQE, as red lines,

- (III) Error mitigated energy obtained from the virtual distillation (**VD**) method using $M = 2$ copies of states, as blue lines,
- (IV) Error mitigated energy obtained from the GSE method based on fault subspace with hardware-oriented implementation (**GSE-DA**) as green lines.

In the numerical demonstration, we clarify the influence of errors present in the measurement-associated unitary circuit and measurement operation itself by comparing the results with (dashed lines) and without noise (solid lines) in them.

4.5 Noise Models

The noise model utilized in numerical simulation highly reflects the noise characteristics of a real quantum device IBM Kawasaki. Here, we describe how to concretely introduce noise for numerical demonstration in a hardware-oriented fashion. The overall procedure can be summarized as follows:

- Step 1. Determine the high-level representation of the quantum circuit.
- Step 2. Compile the circuit so that all the gates are expressed by native gates.
- Step 3. Introduce noise channel for every native gate.
- (Step 3'. Introduce SPAM noise channel, shot noise, or other realistic errors.)
- Step 4. Run the simulation.

In the current work, we assume that the most low-rank operations in the quantum circuits are given by \sqrt{X} , R_z , and CX gates:

Furthermore, we assume that the R_z gates are operated “virtually” by tuning the phase of the gate pulse, and no actual operation is executed on the hardware. Note that such a framework is employed in platforms such as fixed-frequency transmon qubits.

Chapter 5

Results

In this chapter, we show the evaluation of our proposal techniques both numerical simulation and experiments on a real quantum processor.

5.1 Results of Numerical Simulation

Now we proceed to provide the numerical results. We conducted two simulations to inspect the impact of our proposal techniques; (1) the alternating SWAP technique to resolve the qubit routing problem and (2) the noise stretch factors introduced by the GSE-DA technique.

5.1.1 The Performance of Alternating SWAP

We compiled the QEM circuit VD and GSE-DA to resolve constraints of the real device IBM Kawasaki (such as physical two-qubit connection, the error rate of qubits, gate time, etc.) by adding overheads of gate operations. In this simulation, we assess how our proposal technique, Alternating SWAP, suppresses performance degradation caused by operational overheads on the real device. The solid lines show the performance of the case that no errors occur in the derangement and measurement process. The dashed lines show the case of errors and gate overheads that comes from quantum circuit compilation in the entangle measurement process. The **GSE (Alternating SWAP)** denote the case of using alternating SWAP technique, and **GSE (Qiskit's Transpiler)** is the case of the optimized circuit with the Qiskit's function.

Fig. 5.1 shows the comparison of the performance of the GSE technique with/without compilation overheads. The case of no-overhead mitigates the noises nearly perfectly and that is the almost identical result as the original paper demonstrated. Compared to that, both results of compiled circuits are much worse though, our proposal for alternating SWAP technique suppresses the noise effect and shows a better solution than Qiskit compilation in the range of deeper circuits.

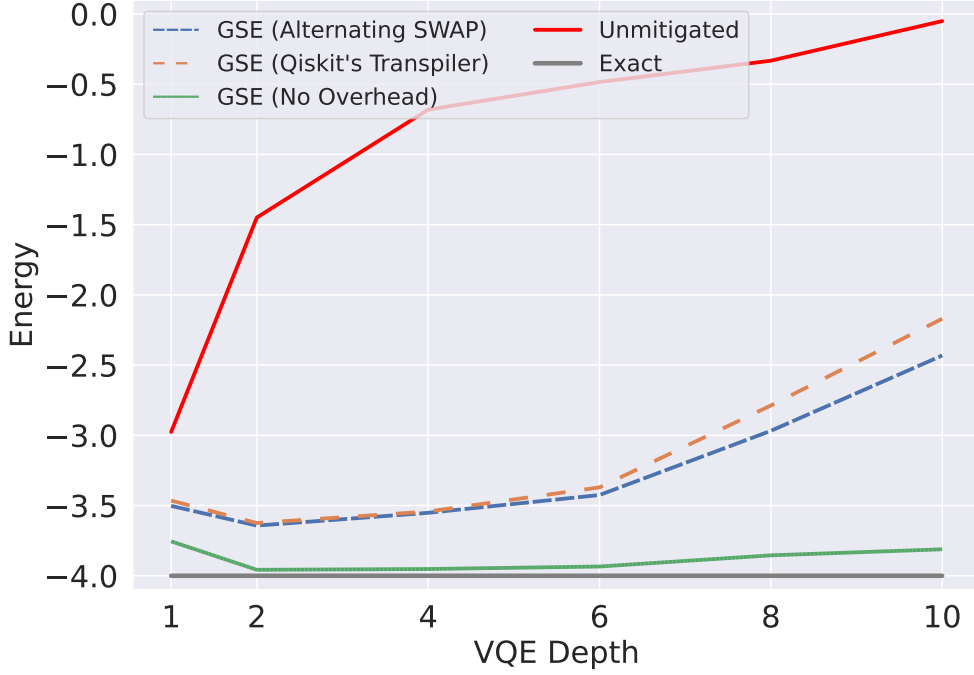


Figure 5.1: **Performance of Alternating SWAP:**

Fig. 5.2 shows the number of CX gates in the quantum circuits for the GSE error mitigation technique. *GSE (No Overhead)* denotes the minimum requirements of the number of CX gates to construct a quantum circuit to mitigate noise that occurs in target state ρ . **GSE (Alternating SWAP)** and **GSE (Qiskit's Transpiler)** show the amount beyond the minimum requirement as overheads come from quantum circuit compilation under the restriction of a physical processor. The overheads of each circuit depth are quite similar and fixed in size of quantum state ρ not the depth, and our proposal shows always better suppression of the increase of additional gates operations.

5.1.2 The Impact of Stretching Noise factor

In this simulation, we inspect the impact of decoherence in time as a noise stretch factor of the GSE-DA technique.

Fig. 5.3 shows the performance of error mitigation of each technique. We vary the buffer time inserted to the quantum circuits from 10^2 ns to 10^4 ns to induce different strengths of the noise. The order of CX gate time is 100 ns for the IBM Kawasaki.

We find that the ground-state energy mitigated by our proposal GSE-DA method

obeys a similar trend as the VD method up to a certain point, but shows better performance on the range of deeper depth circuits that causes more noise. We can understand this from the “effective dimension” of the fault subspace; the amplified decoherence gradually dominates compared to the other error sources that come from the circuit operation. And we find there is a certain sweet spot in the stretch factor of noise amplification. In this simulation, the case that induces additional decoherence $10 * 4\text{ns}$ shows the best performance among GSE-DA techniques.

Given that the current noisy simulation is based on the error profile on quantum devices IBM Kawasaki, it is natural to expect that the GSE-DA method shall perform quite robust even under experiments in hardware.

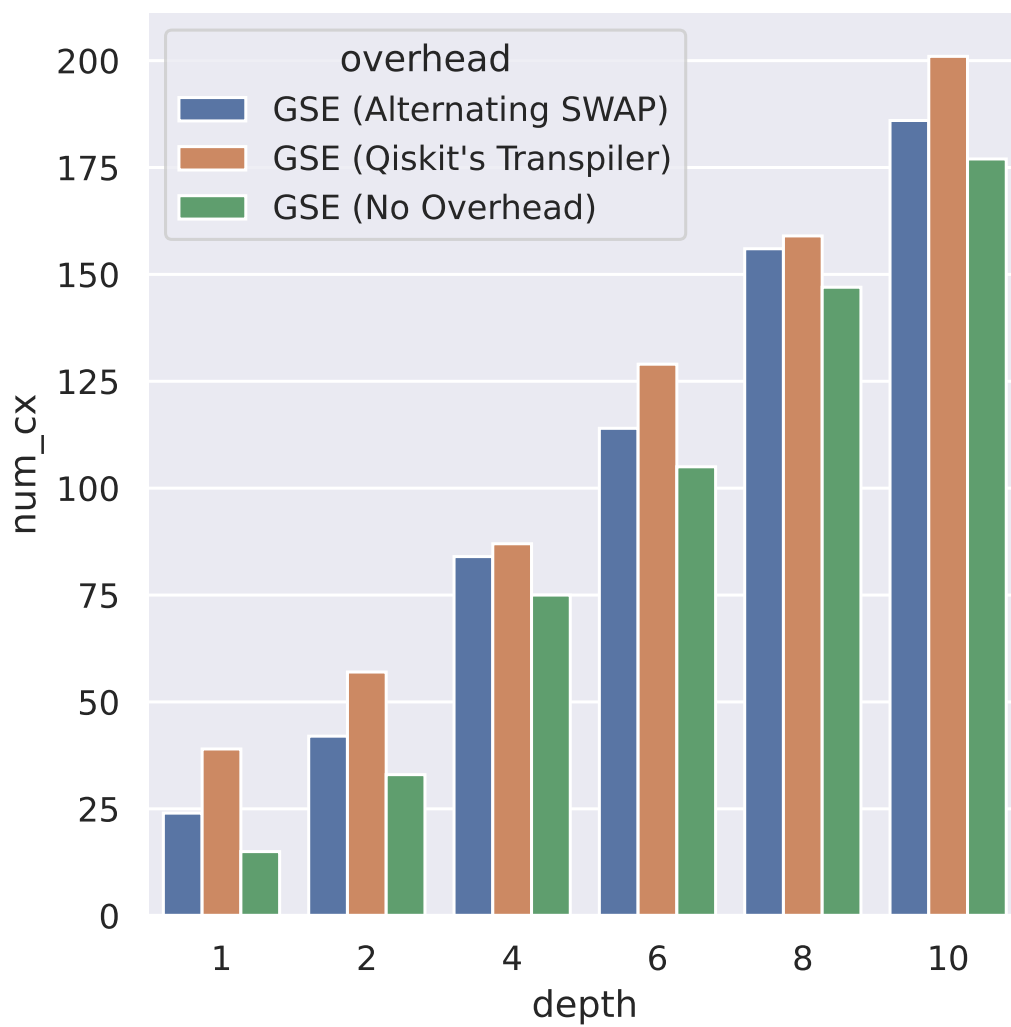


Figure 5.2: Operational Overheads in Quantum Circuit for Experiments:

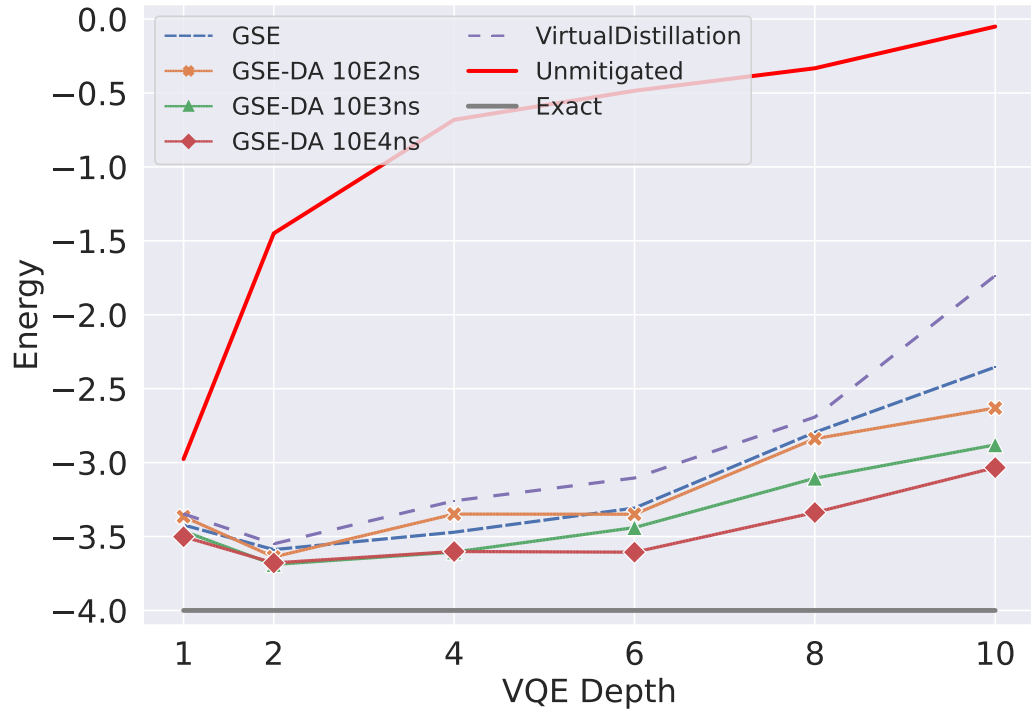


Figure 5.3: **The Impact of Stretching Noise Factor:** Numerical simulation to inspect the impact of noise stretch factor of GSE-DA. Note that each solid line shows the results with the buffer time varying from 10^2 ns to 10^4 ns increase by the order of 10.

5.2 Result of Experiments on the Real Device

Fig .5.4 shows the experimental result on the IBM Kawasaki device to compute expectation value for a 3-qubit sized variational circuit. And our experiments show a proposed fault subspace technique Decoherence Amplification defeats both noisy case and error mitigated expectation value by the VD technique.

We conducted experiments for a range from 10^2 ns to 10^4 ns, and the performance of noise stretch factors almost follows the result from the simulation shown in the previous section. This is understood from the “effective dimension” of the fault subspace; expansion of the difference between copy states allows us to find a better solution from the wider search spaces.

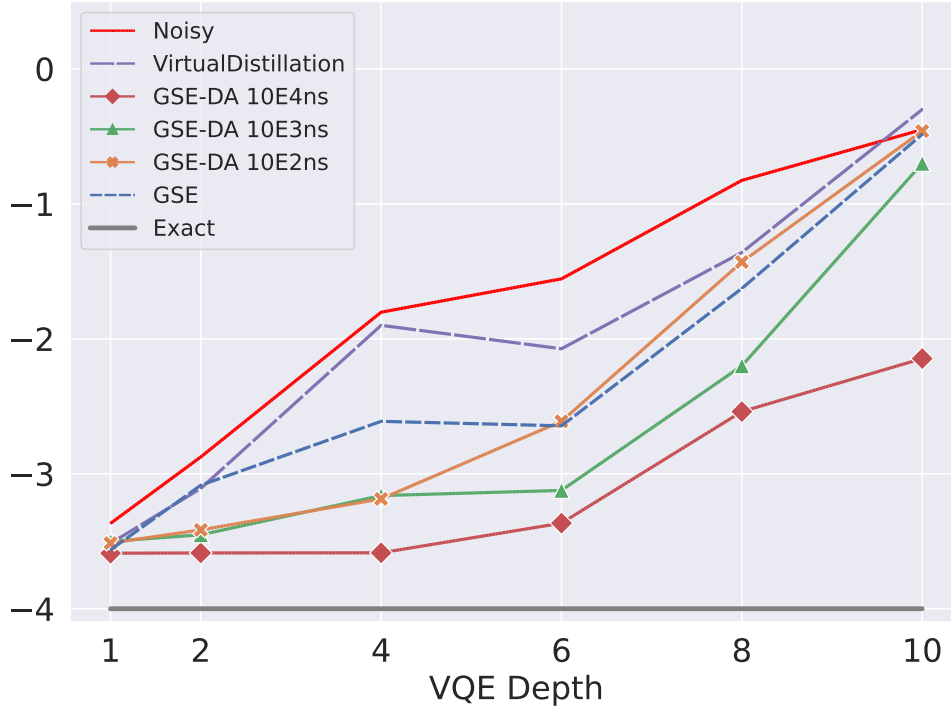


Figure 5.4: Raw expectation values (Red line) and the corresponding mitigated expectation values using Fault Subspace techniques, *Decoherence Amplification* (Green dashed lines). We vary duration time to introduce decoherence of quantum state from 10^2 ns to 10^4 ns. The ideal expectation value is -4 (thick dashed line).

5.3 Scalability

Since our proposal implementation requires additional gate overheads, it is worth discussing scalability. We conducted experiments up to a 9-qubit-sized problem. We prepare the fault subspace technique with 2 copies of quantum states, it occupies 18-qubit on the physical processor we used. Fig. 5.5 shows the comparison of performance for each problem size. We extend the problem size from 3 qubits to 5 and 7 qubits. Unlike the competitive method (VD) doesn't scales even less than a 5-qubit size problem, our proposal methods (green line) show stable mitigation of errors up to a circuit of 7-qubit.

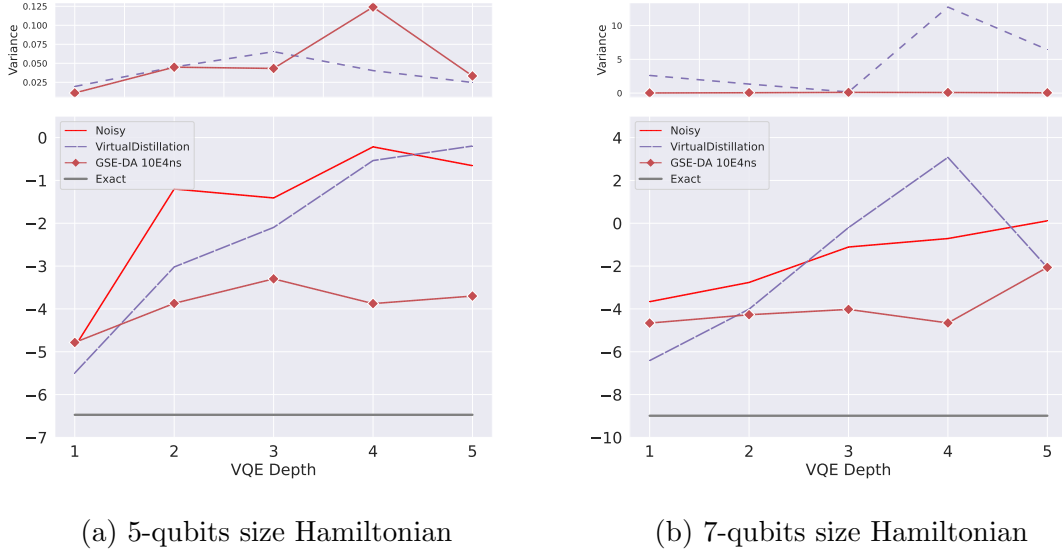


Figure 5.5: Evaluation of scalability on current Processor. We conduct the same experiments with larger qubit sizes. Due to utilizing 2 copies of quantum state ρ in subspace, our proposal technique works even at a 14-qubit size circuit at maximum. The ideal expectation values are -6.47, and -8.99 for the 5-qubits and 7-qubit size problems.

Chapter 6

Conclusions

6.1 Summary

This thesis has introduced novel techniques to leverage hardware imperfections to simulate eigenstates of quantum many-body systems by extending the framework of the fault subspace for the generalized quantum subspace expansion method. In total, three types of techniques to increase different characteristics of hardware errors (gate errors, decoherence in time, and crosstalk inside hardware) were demonstrated via both numerical simulation and experiments on a real device. We have verified that the quasi-SPAM error, namely the gate error present in the entangled measurement (derangement operation), and other overheads including alternating SWAP do not spoil the advantage of constructing error-boosted variational ansatz.

6.2 Future Work

We envision two main directions as future problems.

First, it may be highly practical to seek how to lower the overhead in the entangled measurement associated with the derangement operation. One potential solution is to improve the low-level implementation; we may for example design specific pulse control.

The method to increase hardware noise highly depends on the implementation methods and underlying physical system of processors. Though in this work we focused on the decoherence effect of the quantum state, we may utilize a variety of types of noises derived from different types of hardware.

Acknowledgements

First and foremost, I would like to thank to Professor Rodney Van Meter at Keio University for his long-term support whole my Bachelor's and Master's program. I have had wonderful opportunities at the AQUA lab to learn, which I could have never gained without you. Not only in the lab, but working as your SA/TA was tough, but I am glad that I was capable of working with you. It is, definitely a most valuable experiences in my life.

I would also like to thank to Project Assistant Professor Takahiko Satoh at Keio University for being always a supportive supervisor and spending a large fraction of your time without hesitation to share your knowledge. I enjoyed working with you on so many research projects, QiskitCamp (Snowy Mountain indeed), Subdivided-phase Grover, Mitou Target, quantum multi-programming, and so on. I truly enjoyed spending time with you, and I am lucky to have you throughout my Bachelor's and Master's program.

I would also like to thank to Project Assistant Professor Michal Hadjusek at Keio University for being always a supportive and kind supervisor. I have gained a lots of insight though your lectures and discussions with you. And I hope you and your family spend happy time in Chigasaki which is my favorite coast town.

I would also like to thank to Research Associate Professor Shota Nagayama. I have gained a lots of insight though your lectures and discussions with you. I enjoyed the time you invite me to Mercari office at Roppongi I remember.

I would also like to thank to Project Assistant Professor Bernard Ousmane Sane at Keio University. We spent an amazing amount of time together in this short period of time! I truly hope you and your family spend happy time in Japan.

I would also like to thank all the members of the AQUA lab, cocori, whit3z, zigen-san, sitong, shingy, syu, dave, zomi, sara, sam, poramet, ryom, coscos, syui, mason, kent-tea, soon and collodi which I enjoyed frequently hanging out outside the campus. I'm glad that I had you all. I also thank all the newcomers.

I also thank those that used to be a member of the AQUA lab. First, I have to thank Takaaki Matsuo for spending a lots of time both inside/outside of campus even after your graduation. I always wanted to beats you in video-games. You had teach us a lots of good and bad things. But you are always the best senpai in my life.

I shall thank Shin Nishio, who is my senpai in AQUA lab. You are so wise and having fascinating personalities. I've leaned a lots of valuable things especially through Mitou Target project from you. I hope you spend happy time in Okinawa with the beautiful sea and nature.

I am grateful to all members in Murai / Nakamura / Kusumoto / Takashio / Van Meter / Uehara / Mitsugi / Nakazawa / Tezuka / Takeda / Okoshi Joint Research Group for all the supports.

Thanks to the members of the IBMQ team, Qiskit Community. Thanks to them, I was able to conduct various experiments and learn.

I must also thank my wife Yuumi Okura for always being closest and supporting me.

Appendix A

Variations of techniques for boosting hardware noise

In this appendix, we discuss other potential techniques for boosting hardware-oriented noises besides Decoherence Amplification.

A.1 Gate Repetition (GR)

One of the most dominant error sources in the current (and probably in the near future) quantum computing device is the quantum gates. The gate repetition technique is a well-established technique for amplifying the effect of gate error by inserting a decomposed identity, given by a sequence of multiple gates as $\prod_n G_n = I$. Since the insertion typically does not require any hardware calibration or circuit compiling, it is considered to be one of the simplest techniques to artificially boost the noise in a quantum circuit.

While we can in general choose any gate sets $\{G_n\}$ at any position as long as their product is identity, the common choice established in experiments is to make entangling gates (CX or CZ) redundant, which is expected to be the main error source in the current setup [11]. After an entangling gate that satisfies $G^2 = I$, we insert an even number of entangling gates $G_1 = G_2 = \dots G_{2K} = G$ so that the error from G is multiplied by a factor of $2K + 1$. In Fig. A.1, we show an example of gate repetition for the CX gate.

A.2 Crosstalk Boost (CB)

Now we introduce the crosstalk, one of the most notoriously known and significant non-local error sources in quantum processors [27, 37]. The crosstalk is the unwanted interaction between qubits that are not intended to couple with each other. The error typically arises when one of the connected (or nearby) qubits are interacting

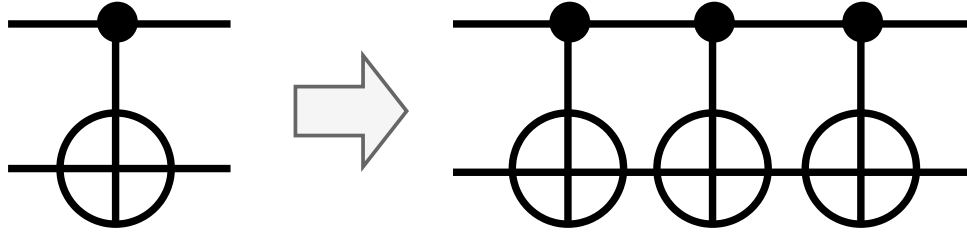


Figure A.1: Amplifying the circuit noise via gate repetition of CX gate. In this example, two CX gates are cancelled out, while the error is expected to increase by a factor of three.

with other ones. When the fabrication/geometry of the quantum register is not sufficiently isolating multi-qubit operations from each other, an entangling operation may simultaneously induce coupling between other qubit pairs. Such a "leak of coupling operation" is known to have a trade-off between the strength of qubit interaction and the magnitude of unwanted crosstalk noise [14, 42].

The suppression of the crosstalk becomes a significant roadblock when we aim to develop larger quantum processors [42, 41, 19], and there are already several approaches designed for superconducting qubits. In the case of quantum processors with tunable couplers including Google's Sycamore design [4], we can tune qubit frequencies or control specific couplers so that the crosstalk is shut down [32, 10]. In fixed frequency qubit systems including superconducting qubit devices provided by IBM, on the other hand, the circuit scheduler must be designed so that concurrent execution of entangling operation between potentially cross-talking qubits is avoided as much as possible [33].

On the contrary, we here focus on intentionally boosting crosstalk errors by amplifying the interference between system and environment qubits, so that the fault subspace spanned by erroneous states is readily expanded. Figure A.2 shows an example that generates crosstalk via cross-resonance gate operations on superconducting quantum processors. Here, we perform two-qubit gates on both the system qubits and the neighbouring environmental qubits at the same time. In devices with imperfect noise control, this operation can amplify the effect of the interaction with the environment, and therefore is complementary to the ordinary reduction approach. It is the opposite of the software approach to mitigate the crosstalk we discussed above.

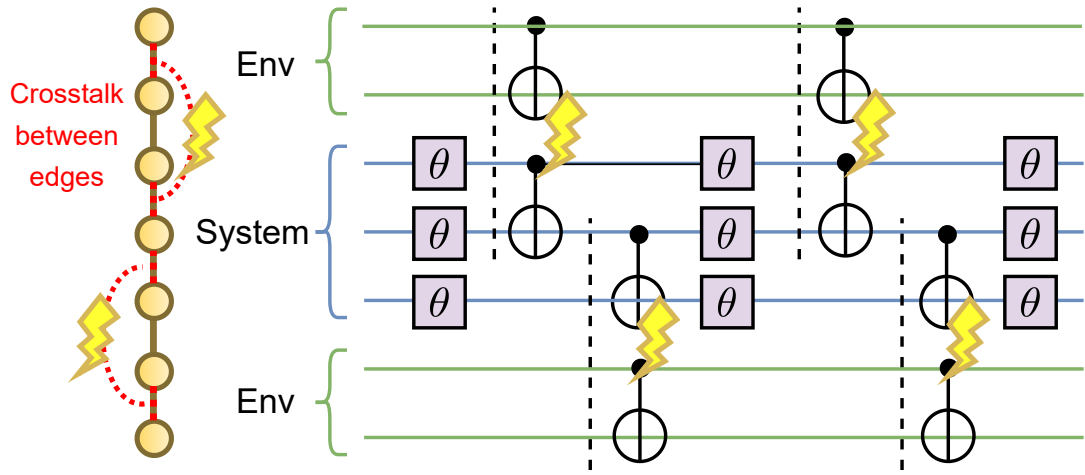


Figure A.2: A schematic illustration of the crosstalk boost method. (Left) Graph representing a 7-qubit processor. The nodes indicate physical qubits, and edges show the connectivity of the qubits, i.e., allow CX gates. (Right) The quantum circuit with three physical qubits as the simulation target system and four qubits of both ends are used as environments. By performing CX gates simultaneously on adjacent environments, we intentionally boost the error rate of CX gates via crosstalk.

Bibliography

- [1] A-tA-v et al. *Qiskit: An Open-source Framework for Quantum Computing*. 2021. DOI: 10.5281/zenodo.2573505.
- [2] Amazon. *Amazon Braket*. <https://aws.amazon.com/jp/braket/>.
- [3] Michael Armbrust et al. “A view of cloud computing”. en. In: *Communications of the ACM* 53.4 (Apr. 2010), pp. 50–58. ISSN: 0001-0782, 1557-7317. DOI: 10.1145/1721654.1721672. URL: <https://dl.acm.org/doi/10.1145/1721654.1721672>.
- [4] Frank Arute et al. “Quantum supremacy using a programmable superconducting processor”. en. In: *Nature* 574.7779 (Oct. 2019). Number: 7779 Publisher: Nature Publishing Group, pp. 505–510. ISSN: 1476-4687. DOI: 10.1038/s41586-019-1666-5. URL: <https://www.nature.com/articles/s41586-019-1666-5>.
- [5] Dave Bacon and Wim van Dam. “Recent progress in quantum algorithms”. In: *Communications of the ACM* 53.2 (Feb. 2010), pp. 84–93. ISSN: 0001-0782. DOI: 10.1145/1646353.1646375. URL: <https://doi.org/10.1145/1646353.1646375>.
- [6] Zhenyu Cai et al. *Quantum Error Mitigation*. arXiv:2210.00921 [quant-ph] version: 1. Oct. 2022. DOI: 10.48550/arXiv.2210.00921. URL: <http://arxiv.org/abs/2210.00921>.
- [7] John Clarke and Frank K. Wilhelm. “Superconducting quantum bits”. en. In: *Nature* 453.7198 (June 2008). Number: 7198 Publisher: Nature Publishing Group, pp. 1031–1042. ISSN: 1476-4687. DOI: 10.1038/nature07128. URL: <https://www.nature.com/articles/nature07128>.
- [8] D-wave. *D-wave Leap*. <https://cloud.dwavesys.com/leap/>.
- [9] Simon J. Devitt. “Performing quantum computing experiments in the cloud”. In: *Physical Review A* 94.3 (Sept. 2016). Publisher: American Physical Society, p. 032329. DOI: 10.1103/PhysRevA.94.032329. URL: <https://link.aps.org/doi/10.1103/PhysRevA.94.032329>.

- [10] Yongshan Ding et al. “Systematic Crosstalk Mitigation for Superconducting Qubits via Frequency-Aware Compilation”. In: *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. Oct. 2020, pp. 201–214. DOI: 10.1109/MICRO50266.2020.00028.
- [11] E. F. Dumitrescu et al. “Cloud Quantum Computing of an Atomic Nucleus”. en. In: *Physical Review Letters* 120.21 (May 2018), p. 210501. ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.120.210501. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.120.210501>.
- [12] Suguru Endo et al. “Hybrid Quantum-Classical Algorithms and Quantum Error Mitigation”. In: *Journal of the Physical Society of Japan* 90.3 (Mar. 2021). Publisher: The Physical Society of Japan, p. 032001. ISSN: 0031-9015. DOI: 10.7566/JPSJ.90.032001. URL: <https://journals.jps.jp/doi/10.7566/JPSJ.90.032001>.
- [13] Austin G. Fowler et al. “Surface codes: Towards practical large-scale quantum computation”. In: *Physical Review A* 86.3 (Sept. 2012). Publisher: American Physical Society, p. 032324. DOI: 10.1103/PhysRevA.86.032324. URL: <https://link.aps.org/doi/10.1103/PhysRevA.86.032324>.
- [14] Jay M. Gambetta et al. “Characterization of Addressability by Simultaneous Randomized Benchmarking”. In: *Physical Review Letters* 109.24 (Dec. 2012). Publisher: American Physical Society, p. 240504. DOI: 10.1103/PhysRevLett.109.240504. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.109.240504>.
- [15] Juan Carlos Garcia-Escartin and Pedro Chamorro-Posada. “The SWAP test and the Hong-Ou-Mandel effect are equivalent”. In: *Physical Review A* 87.5 (May 2013). arXiv:1303.6814 [quant-ph], p. 052330. ISSN: 1050-2947, 1094-1622. DOI: 10.1103/PhysRevA.87.052330. URL: <http://arxiv.org/abs/1303.6814>.
- [16] Google. *Quantum Computing Playground*. <https://experiments.withgoogle.com/quantum-computing-playground>.
- [17] Daniel Gottesman. *Stabilizer codes and quantum error correction*. en. 1997. URL: <https://www.proquest.com/openview/82213d22fc3a65405da6fe9a2106c432/1?pq-origsite=gscholar&cbl=18750&diss=y>.
- [18] Lov K. Grover. “A fast quantum mechanical algorithm for database search”. en. In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing - STOC '96*. Philadelphia, Pennsylvania, United States: ACM Press, 1996, pp. 212–219. ISBN: 978-0-89791-785-8. DOI: 10.1145/237814.237866. URL: <http://portal.acm.org/citation.cfm?doid=237814.237866>.

- [19] Robin Harper, Steven T. Flammia, and Joel J. Wallman. “Efficient learning of quantum noise”. en. In: *Nature Physics* 16.12 (Dec. 2020). Number: 12 Publisher: Nature Publishing Group, pp. 1184–1188. ISSN: 1745-2481. DOI: 10.1038/s41567-020-0992-8. URL: <https://www.nature.com/articles/s41567-020-0992-8>.
- [20] Brian Hayes. “Cloud computing”. In: *Communications of the ACM* 51.7 (July 2008), pp. 9–11. ISSN: 0001-0782. DOI: 10.1145/1364782.1364786. URL: <https://doi.org/10.1145/1364782.1364786>.
- [21] William J. Huggins et al. “Virtual Distillation for Quantum Error Mitigation”. In: *Physical Review X* 11.4 (Nov. 2021). Publisher: American Physical Society, p. 041036. DOI: 10.1103/PhysRevX.11.041036. URL: <https://link.aps.org/doi/10.1103/PhysRevX.11.041036>.
- [22] IBM. *IBM Quantum Experience*. <https://quantum-computing.ibm.com/>.
- [23] IBM. *Quantum Teleportation*. <https://qiskit.org/textbook/ch-algorithms/teleportation.html>.
- [24] Jonathan A. Jones. “Quantum computing and nuclear magnetic resonance”. en. In: *PhysChemComm* 4.11 (Jan. 2001). Publisher: The Royal Society of Chemistry, pp. 49–56. ISSN: 1460-2733. DOI: 10.1039/B103231N. URL: <https://pubs.rsc.org/en/content/articlelanding/2001/qu/b103231n>.
- [25] D. Kielpinski, C. Monroe, and D. J. Wineland. “Architecture for a large-scale ion-trap quantum computer”. en. In: *Nature* 417.6890 (June 2002). Number: 6890 Publisher: Nature Publishing Group, pp. 709–711. ISSN: 1476-4687. DOI: 10.1038/nature00784. URL: <https://www.nature.com/articles/nature00784>.
- [26] Bálint Koczor. “Exponential Error Suppression for Near-Term Quantum Devices”. In: *Physical Review X* 11.3 (Sept. 2021). Publisher: American Physical Society, p. 031057. DOI: 10.1103/PhysRevX.11.031057. URL: <https://link.aps.org/doi/10.1103/PhysRevX.11.031057>.
- [27] P. Krantz et al. “A quantum engineer’s guide to superconducting qubits”. In: *Applied Physics Reviews* 6.2 (June 2019). Publisher: American Institute of Physics, p. 021318. DOI: 10.1063/1.5089550. URL: <https://aip.scitation.org/doi/10.1063/1.5089550>.
- [28] Ying Li and Simon C. Benjamin. “Efficient Variational Quantum Simulator Incorporating Active Error Minimization”. en. In: *Physical Review X* 7.2 (June 2017), p. 021050. ISSN: 2160-3308. DOI: 10.1103/PhysRevX.7.021050. URL: <http://link.aps.org/doi/10.1103/PhysRevX.7.021050>.
- [29] David C. McKay et al. *Qiskit Backend Specifications for OpenQASM and Open-Pulse Experiments*. arXiv:1809.03452 [quant-ph]. Sept. 2018. DOI: 10.48550/arXiv.1809.03452. URL: <http://arxiv.org/abs/1809.03452>.

- [30] Microsoft. *Microsoft Azure Quantum*. <https://azure.microsoft.com/en-us/services/quantum/>.
- [31] Ashley Montanaro. “Quantum algorithms: an overview”. en. In: *npj Quantum Information* 2.1 (Jan. 2016). Number: 1 Publisher: Nature Publishing Group, pp. 1–8. ISSN: 2056-6387. DOI: 10.1038/npjqi.2015.23. URL: <https://www.nature.com/articles/npjqi201523>.
- [32] Pranav Mundada et al. “Suppression of Qubit Crosstalk in a Tunable Coupling Superconducting Circuit”. In: *Physical Review Applied* 12.5 (Nov. 2019). Publisher: American Physical Society, p. 054023. DOI: 10.1103/PhysRevApplied.12.054023. URL: <https://link.aps.org/doi/10.1103/PhysRevApplied.12.054023>.
- [33] Prakash Murali et al. “Software Mitigation of Crosstalk on Noisy Intermediate-Scale Quantum Computers”. In: *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*. ASPLOS ’20. New York, NY, USA: Association for Computing Machinery, Mar. 2020, pp. 1001–1016. ISBN: 978-1-4503-7102-5. DOI: 10.1145/3373376.3378477. URL: <https://doi.org/10.1145/3373376.3378477>.
- [34] Shin Nishio et al. “Extracting Success from IBM’s 20-Qubit Machines Using Error-Aware Compilation”. In: *ACM Journal on Emerging Technologies in Computing Systems* 16.3 (May 2020), 32:1–32:25. ISSN: 1550-4832. DOI: 10.1145/3386162. URL: <https://doi.org/10.1145/3386162>.
- [35] Jeremy L. O’Brien, Akira Furusawa, and Jelena Vučković. “Photonic quantum technologies”. en. In: *Nature Photonics* 3.12 (Dec. 2009). Number: 12 Publisher: Nature Publishing Group, pp. 687–695. ISSN: 1749-4893. DOI: 10.1038/nphoton.2009.229. URL: <https://www.nature.com/articles/nphoton.2009.229>.
- [36] Yasuhiro Ohkura, Takahiko Satoh, and Rodney Van Meter. “Simultaneous Execution of Quantum Circuits on Current and Near-Future NISQ Systems”. In: *IEEE Transactions on Quantum Engineering* 3 (2022). Conference Name: IEEE Transactions on Quantum Engineering, pp. 1–10. ISSN: 2689-1808. DOI: 10.1109/TQE.2022.3164716.
- [37] C. Ospelkaus et al. “Trapped-ion quantum logic gates based on oscillating magnetic fields”. eng. In: *Physical Review Letters* 101.9 (Aug. 2008), p. 090502. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.101.090502.
- [38] Alberto Peruzzo et al. “A variational eigenvalue solver on a photonic quantum processor”. en. In: *Nature Communications* 5.1 (July 2014). Number: 1 Publisher: Nature Publishing Group, p. 4213. ISSN: 2041-1723. DOI: 10.1038/ncomms5213. URL: <https://www.nature.com/articles/ncomms5213>.

- [39] John Preskill. “Quantum Computing in the NISQ era and beyond”. en-GB. In: *Quantum* 2 (Aug. 2018). Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften, p. 79. DOI: 10.22331/q-2018-08-06-79. URL: <https://quantum-journal.org/papers/q-2018-08-06-79/>.
- [40] Rigetti. *Rigetti Forest*. <https://www.rigetti.com/quantum-computing/>.
- [41] Kenneth Rudinger et al. “Probing Context-Dependent Errors in Quantum Processors”. In: *Physical Review X* 9.2 (June 2019). Publisher: American Physical Society, p. 021045. DOI: 10.1103/PhysRevX.9.021045. URL: <https://link.aps.org/doi/10.1103/PhysRevX.9.021045>.
- [42] Sarah Sheldon et al. “Procedure for systematically tuning up cross-talk in the cross-resonance gate”. In: *Physical Review A* 93.6 (June 2016). Publisher: American Physical Society, p. 060302. DOI: 10.1103/PhysRevA.93.060302. URL: <https://link.aps.org/doi/10.1103/PhysRevA.93.060302>.
- [43] Peter W. Shor. “Polynomial time algorithms for discrete logarithms and factoring on a quantum computer”. en. In: *Algorithmic Number Theory*. Ed. by Leonard M. Adleman and Ming-Deh Huang. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 1994, pp. 289–289. ISBN: 978-3-540-49044-9. DOI: 10.1007/3-540-58691-1_68.
- [44] Yasunari Suzuki et al. “Quantum Error Mitigation as a Universal Error Reduction Technique: Applications from the NISQ to the Fault-Tolerant Quantum Computing Eras”. In: *PRX Quantum* 3.1 (Mar. 2022). Publisher: American Physical Society, p. 010345. DOI: 10.1103/PRXQuantum.3.010345. URL: <https://link.aps.org/doi/10.1103/PRXQuantum.3.010345>.
- [45] Swamit S. Tannu and Moinuddin K. Qureshi. “Not All Qubits Are Created Equal: A Case for Variability-Aware Policies for NISQ-Era Quantum Computers”. In: *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*. ASPLOS ’19. New York, NY, USA: Association for Computing Machinery, Apr. 2019, pp. 987–999. ISBN: 978-1-4503-6240-5. DOI: 10.1145/3297858.3304007. URL: <https://doi.org/10.1145/3297858.3304007>.
- [46] Kristan Temme, Sergey Bravyi, and Jay M. Gambetta. “Error Mitigation for Short-Depth Quantum Circuits”. In: *Physical Review Letters* 119.18 (Nov. 2017). Publisher: American Physical Society, p. 180509. DOI: 10.1103/PhysRevLett.119.180509. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.119.180509>.
- [47] Xanadu. *Xanadu Quantum Cloud*. <https://www.xanadu.ai/>.
- [48] Nobuyuki Yoshioka et al. “Generalized Quantum Subspace Expansion”. In: *Physical Review Letters* 129.2 (July 2022). Publisher: American Physical Society, p. 020502. DOI: 10.1103/PhysRevLett.129.020502. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.129.020502>.