Analyzing the barren plateau phenomenon in training quantum neural network with the ZX-calculus

Chen Zhao[1,2] and Xiao-Shan Gao[1,2]

[1]Academy of Mathematics and Systems Science, Chinese Academy of Sciences

[2]University of Chinese Academy of Sciences

In this paper, we propose a general scheme to analyze the barren plateau phenomenon in training quantum neural networks with the ZX-calculus. More precisely, we extend the barren plateaus theorem from unitary 2-design circuits to any parameterized quantum circuits under certain reasonable assumptions. The main technical contribution of this paper is representing certain integrations as ZX-diagrams and computing them with the ZX-calculus. The method is used to analyze four concrete quantum neural networks with different structures. It is shown that, for the hardware efficient ansatz and the MPS-inspired ansatz, there exist barren plateaus, while for the QCNN and the tree tensor network ansatz, there exists no barren plateau.

1 Introduction

In recent years, the hybrid quantum-classical algorithms are widely used in quantum chemistry [1–4], combinatorial optimization [5, 6], and quantum machine learning [7–12]. In these hybrid quantum-classical algorithms, the goal is usually training parameterized quantum circuits (PQCs) with classical optimizers. The PQC will be applied to an initial state and then the state will be measured on a quantum device. The classical optimizer will update the parameters of the PQC according to the measurement results. As the PQC can be run on noisy intermediate-scale quantum (NISQ [13]) devices, these algorithms are regarded as near-term practical quantum algorithms with potential quantum advantages.

There exist many methods to train PQCs. Some of these are gradient-based [14–16] and some are not [17, 18]. In quantum machine learning, gradient-based methods are widely used. When using gradient-based methods to train PQCs, one may suffer from the barren plateaus (BP) phenomenon which was first studied in [19]. The BP phenomenon is that the gradient of parameters of the PQC will vanish exponentially in the system size. It was proved that if the PQC is 2-design then the barren plateau phenomenon exists [19]. Even if the PQC is shallow and locally 2-design, the BP phenomenon also exists if the cost-function is global [20]. It was also proved that if the cost-function is local, then the PQC with $\log(n)$-depth is trainable [20]. Too much entanglement will induce barren plateaus [21]. And the noise from quantum hardware also causes barren plateaus, which is called noise-induced barren plateaus [22].

The above results are obtained under certain assumptions of unitary t-design and it is still difficult to analyze the BP phenomenon for PQCs besides those containing t-design parts. In this paper, we develop a general scheme to analyze whether there exist BP phenomena when training a concrete PQC. We focus on BP phenomena induced by the structure of PQCs and noise-induced barren plateaus are not considered in this paper. The most important tool used is the *ZX-calculus*, a graphical language for describing and reasoning quantum processes. ZX-calculus was developed by Coecke and Duncan in [23, 24], which has various applications including

Chen Zhao: zhaochen17@mails.ucas.ac.cn
Xiao-Shan Gao: xgao@mmrc.iss.ac.cn

quantum circuit synthesis [25–28], measurement-based quantum computing [29, 30], quantum error correction [31, 32], condensed matter physics [33], and quantum natural language processing [34]. In the ZX-calculus, the objects to deal with are ZX-diagrams, which consist of two kinds of tensors, Z-spiders and X-spiders. And ZX-diagrams can be rewritten with ZX-calculus rules. Moreover, every quantum circuit can be converted into a ZX-diagram.

Let $\vec{\theta} = (\theta_1, \ldots, \theta_m)$ be a set of parameters. To analyze the gradient of a PQC $U(\vec{\theta})$ with respect to a Hamiltonian H, we need to estimate the following expectation and the variance

$$\mathbf{E}\left[\frac{\partial \langle H \rangle}{\partial \theta_j}\right], \ \mathbf{Var}\left[\frac{\partial \langle H \rangle}{\partial \theta_j}\right] \tag{1}$$

where $\langle H \rangle$ is defined in (3). It will be shown that the expectation in (1) is always zero. The PQC is said to have *barren plateaus* if the variance in (1) vanishes exponentially in terms of the size of the PQC. The PQC is said to have *no barren plateau* or *trainable* if the variance in (1) vanishes polynomially in terms of the size of the PQC.

To estimate the expectation and variance in (1), we first represent them as ZX-diagrams. Since the expectation and the variance are integrations, the main technical contribution of this paper is representing these integrations as ZX-diagrams and computing them with the ZX-calculus. More precisely, with the rewriting rules in the ZX-calculus, we prove that (1) is equal to the contraction of a tensor network with a similar structure as the PQC. Hence, the existence of barren plateaus is totally characterized by the scaling property of the tensor network.

We use these techniques to analyze whether there exist BP phenomena in the hardware-efficient ansatz [2], the QCNN [35], the tree tensor network ansatz [36], and the MPS-inspired ansatz [37]. We show that there exist barren plateaus in hardware-efficient ansatz and MPS-inspired ansatz, and there is no barren plateau in the QCNN and the tree tensor network ansatz.

This paper is organized as follows. A brief introduction to the PQC, the BP phenomenon, and the ZX-calculus will be given in section 2. We will prove the main result which characterizes (1) in section 3. And the analysis of 4 concrete PQCs is given in section 4.

2 Preliminary

2.1 Hybrid quantum-classical algorithms

In a hybrid quantum-classical algorithm, there will be an ansatz, which is a PQC of the form

$$U(\vec{\theta}) = \prod_{j=1}^{M} [U_j(\theta_j) \cdot V_j]. \tag{2}$$

Here $U_j(\theta_j), j = 1, \ldots, M$ are parameterized gates, for example the rotation gates R_X, R_Y, R_Z. And $V_j, j = 1, \ldots, M$ are non-parameterized gates, for example the Hadamard gate H and the CNOT gate. The PQC will be applied to an initial state $|\psi_0\rangle$ and then the state will be measured. The above procedure, which is called the quantum part of the algorithm, will be run on quantum processors,

Meanwhile, there will be a classical part which consists of classical processors to optimize the parameters of the PQC in the quantum part. A cost-function $L(\vec{\theta})$ will be estimated in the classical part based on the measurement results. Usually, the expectation

$$\langle H \rangle = \langle \psi_0 | U^\dagger(\vec{\theta}) H U(\vec{\theta}) | \psi_0 \rangle \tag{3}$$

of a given Hamiltonian H will be regarded as the cost-function in many tasks.

As demonstrated in figure 1, the quantum part runs the PQC and gets the measurement results and the classical part estimates the cost-function and updates the parameters. After several iterations, the cost-function may converge and be optimized. Then the training will be stopped. This is the main idea of the hybrid quantum-classical algorithm.

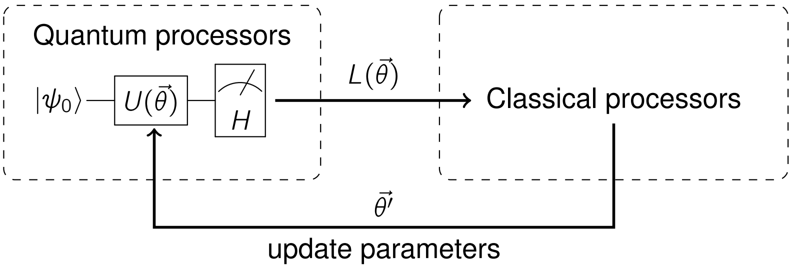


Figure 1: The hybrid quantum-classical algorithm

2.2 Barren plateau phenomenon

When the parameterized gates are of the form

$$U_j(\theta_j) = e^{-i\frac{\theta}{2}H_j},$$

where H_j satisfies $H_j^2 = I$, the gradient $\frac{\partial\langle H\rangle}{\partial\theta_j}$ can be estimated by the parameter shifting rule without changing the structure of the PQC [14]. Once we get the gradient, we can use gradient-based optimization methods, such as gradient descent, to optimize the parameters.

Ideally, if the gradient does not vanish too fast as the size of the PQC grows, then the gradient could be estimated efficiently and the PQC could be trained easily. However, the BP phenomenon tells us that in many cases, the gradient vanishes exponentially as the system grows up. When this happens, the PQC will be difficult to be trained. The first rigorous proof of the BP phenomenon is shown below.

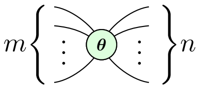
Theorem 1 ([19])**.** *Consider a PQC $U(\vec{\theta}) = V(\theta_M, \dots, \theta_{j+1})U(\theta_j)W(\theta_{j-1}, \dots, \theta_1)$ and a Hamiltonian H. The expectation of gradient is 0 if V and W are 1-design. And the variance of gradient* $\mathbf{Var}\left[\frac{\partial\langle H\rangle}{\partial\theta_j}\right]$ *vanishes exponentially in the number of qubits if V or W is 2-design.*

Hence, when designing the ansatz PQC for a hybrid quantum-classical algorithm, we should analyze whether there exist BP phenomena in it to ensure that it can be trained efficiently.

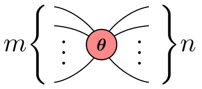
2.3 The ZX-calculus

We provide a briefly introduction to the ZX-calculus here. For more details, please refer to [38, 39].

In the ZX-calculus, quantum states and their transformations are represented as ZX-diagrams which consist of 2 kinds of tensors, *Z-spiders* and *X-spiders*. The Z-spider is denoted as the green node, and the X-spider is denoted as the red node. They can be written explicitly in the Dirac notation as follows.

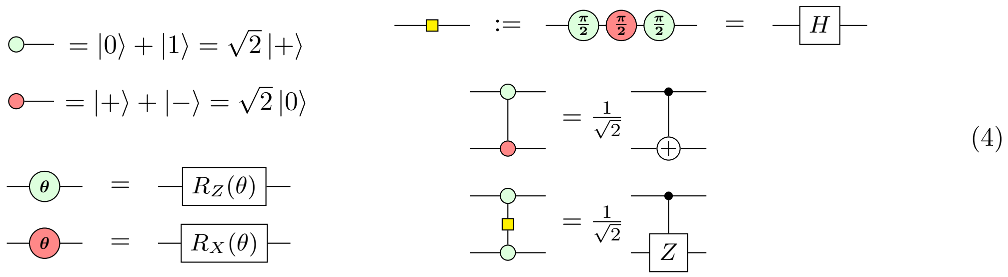


$$:= \underbrace{|0\dots0\rangle}_{n}\underbrace{\langle 0\dots0|}_{m} + e^{i\theta}\underbrace{|1\dots1\rangle}_{n}\underbrace{\langle 1\dots1|}_{m}$$



$$:= \underbrace{|+\dots+\rangle}_{n}\underbrace{\langle +\dots+|}_{m} + e^{i\theta}\underbrace{|-\dots-\rangle}_{n}\underbrace{\langle -\dots-|}_{m}$$

For a spider, the edges on the left are called *input* and the edges on the right are called *output*. The angle θ is called the *phase* of the spider. For simplicity, we will omit the phase when it is zero. Spiders can be connected with wires. Hence, ZX-diagrams can be regarded as tensor networks generated with Z-spiders and X-spiders. For example, we can use ZX-diagrams to represent the following quantum states and quantum gates.



Here we introduce a new notation, the yellow box, to represent the Hadamard gate

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Since the gates set $\{R_Z, R_X, \mathrm{CNOT}\}$ is universal for quantum computing, in principle, one can convert every quantum circuit to a ZX-diagram with the equations in (4).

Moreover, the ZX-calculus is a powerful tool for reasoning. There are several rewriting rules in the ZX-calculus with which one can rewrite a ZX-diagram to another equivalent form. Figure 2 gives some basic rewriting rules in the ZX-calculus. Here, two ZX-diagrams A, B are said to be equivalent if and only if there exists a non-zero constant $c \in \mathbb{C}$, such that $A = c \cdot B$.

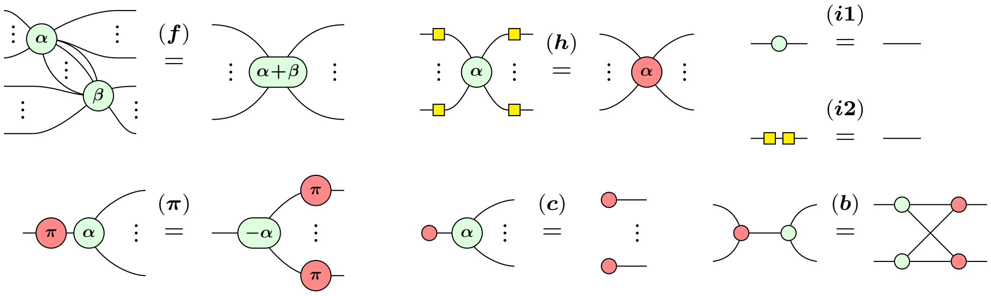


Figure 2: Some basic rewriting rules in the ZX-calculus. Here, '. . . ' means 0 or more. (This figure is from [25].)

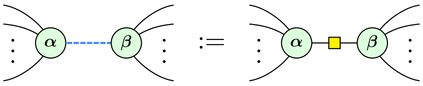
Note that the ZX-calculus is *universal*. It means that any linear transformations can be represented as ZX-diagrams. Moreover, the rules in figure 2 are *complete* for the stabilizer quantum mechanics where phases can only be multiples of $\frac{\pi}{2}$ [40, 41]. That is, if two ZX-diagrams are equivalent, then there exists a set of rewriting rules in figure 2 that rewrites one into another. There are also completeness results for the Clifford+T quantum mechanics, where phases can be multiples of $\frac{\pi}{4}$, and for arbitrary ZX-diagrams [42–45].

In this paper, we will focus on a canonical form of the ZX-diagram, the *graph-like ZX-diagram* which is defined in [25].

Definition 1 ([25])**.** *A ZX-diagram is graph-like if*

1. *All spiders are Z-spiders.*
2. *Z-spiders are only connected via Hadamard edges.*
3. *There exist no parallel Hadamard edges or self-loops.*
4. *Every input or output is connected to a Z-spider and every Z-spider is connected to at most one input or output.*

Two spiders being connected via a Hadamard edge means that they are connected with a Hadamard box. Alternatively, we will also use the dashed blue edge to represent a Hadamard edge.



All X-spiders can be rewritten to Z-spiders by using the rule $(\boldsymbol{h})$ in figure 2. Connected Hadamard boxes can be canceled with the rule $(\boldsymbol{i2})$ and normal edges can be canceled with the rule $(\boldsymbol{f})$. Furthermore, parallel Hadamard edges and self-loops can be canceled with rules in the figure 3. Hence, every ZX-diagram is equivalent to a graph-like ZX-diagram [25].

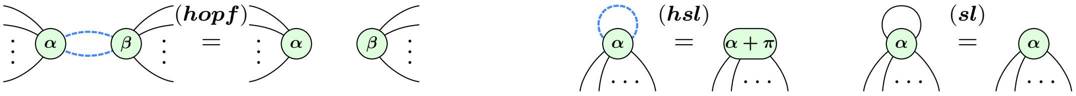


Figure 3: Rules for canceling parallel edges and self-loops [25].

3 Analyzing the BP phenomenon with the ZX-calculus

In this section, we will show how to analyze the BP phenomenon with the ZX-calculus. More precisely, we will show how to estimate the expectation and the variance of the gradient of the cost function of a PQC with respect to a Hamiltonian with the ZX-calculus. The main technique we used is to compute integration over unitarians with the ZX-calculus.

Scalars are ignored in the rules in section 2.3. However, to consider the BP phenomenon, the scalar is necessary. Hence, we first give the precise rules with scalars in figure 4.

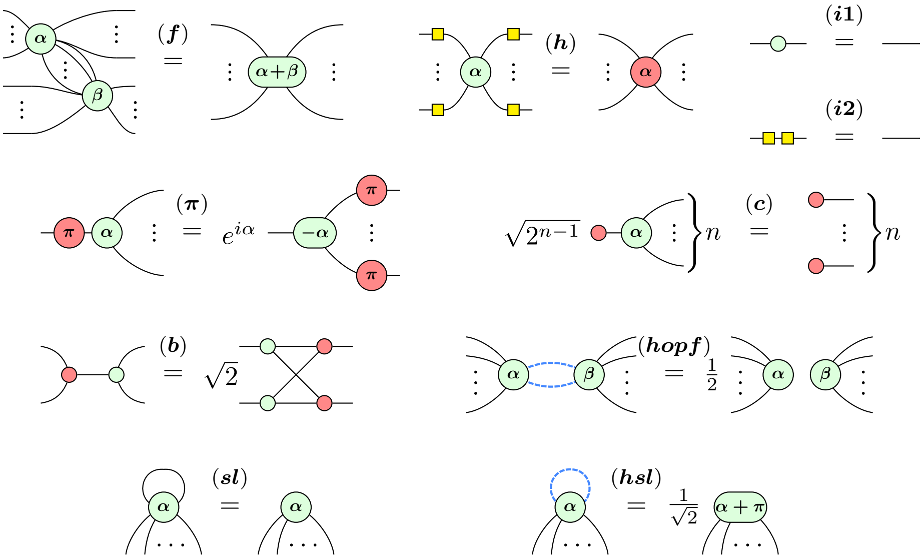


Figure 4: Rewriting rules with scalars.

In this paper, we consider PQCs under the following assumptions.

Assumption 1. *The PQC $U(\vec{\theta})$ satisfies*

1. *Each gate in U is one of $\{R_X, R_Z, H, \text{CNOT}\}$.*
2. *The parameters in $\vec{\theta} = (\theta_1, \ldots, \theta_m)$ are independent uniform random variables in the interval $[-\pi, \pi]$.*

3.1 Representing gradients as ZX-diagrams

Consider a PQC $U(\vec{\theta})$ of n-qubits and a Hamiltonian H. Without loss of generality, we also assume that we apply this PQC to an initial state $|0\rangle$. Then the expectation of H can be expressed as

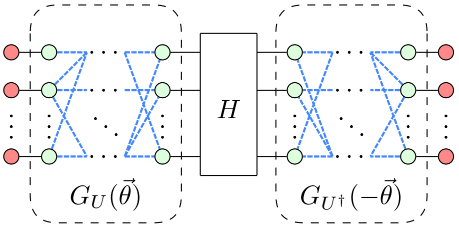
$$\langle H\rangle = \langle 0| U^\dagger(\vec{\theta}) H U(\vec{\theta}) |0\rangle . \tag{5}$$

As shown in section 2.3, we can convert the PQC $U(\vec{\theta})$ to a parameterized graph-like ZX-diagram $G_U(\vec{\theta})$ with (4). Suppose that

$$U(\vec{\theta}) = c \cdot G_U(\vec{\theta})$$

for a constant c, then $\langle H\rangle$ can also be expressed as a ZX-diagram as demonstrated in the following equation.

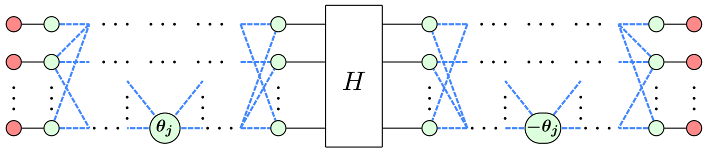
$$\langle H\rangle = \frac{|c|^2}{2^n} \cdot \tag{6}$$



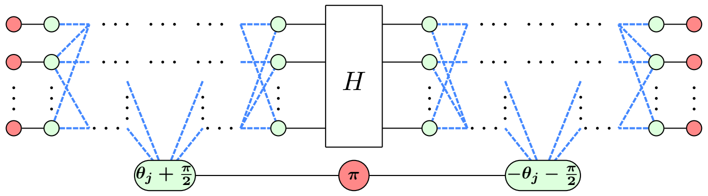
If we expand the spider by the definition of the Z-spider, we can prove that the gradient $\frac{\partial\langle H\rangle}{\partial\theta_j}$ can be represented as a ZX-diagram.

Theorem 2. *The gradient can be represented as the following equation.*

$$\frac{2^n}{|c|^2} \cdot \frac{\partial\langle H\rangle}{\partial\theta_j} = \frac{\partial}{\partial\theta_j}$$



$$=$$



Proof. The proof is given in appendix A. □

To analyze the BP phenomenon, we need to compute the expectation

$$\mathbf{E}(\frac{\partial\langle H\rangle}{\partial\theta_j}) = \int_{\vec{\theta}} p(\vec{\theta}) \frac{\partial\langle H\rangle}{\partial\theta_j} \mathrm{d}\vec{\theta}, \tag{7}$$

and the variance

$$\mathbf{Var}(\frac{\partial\langle H\rangle}{\partial\theta_j}) = \mathbf{E}(\left|\frac{\partial\langle H\rangle}{\partial\theta_j}\right|^2) - \left(\mathbf{E}(\frac{\partial\langle H\rangle}{\partial\theta_j})\right)^2 = \int_{\vec{\theta}} p(\vec{\theta}) \left|\frac{\partial\langle H\rangle}{\partial\theta_j}\right|^2 \mathrm{d}\vec{\theta} - \left(\mathbf{E}(\frac{\partial\langle H\rangle}{\partial\theta_j})\right)^2, \tag{8}$$

for $j = 1, \ldots, m$. Here $p(\vec{\theta})$ is the probability of the parameters $\vec{\theta}$.

By assumption 1, (7) and (8) can be written as

$$\mathbf{E}(\frac{\partial\langle H\rangle}{\partial\theta_j}) = \frac{1}{(2\pi)^m} \int_{\theta_1} \cdots \int_{\theta_m} \frac{\partial\langle H\rangle}{\partial\theta_j} \mathrm{d}\theta_1 \ldots \mathrm{d}\theta_m, \tag{9}$$

and

$$\mathbf{Var}(\frac{\partial \langle H\rangle}{\partial \theta_j}) = \frac{1}{(2\pi)^m}\int_{\theta_1}\cdots\int_{\theta_m}\left|\frac{\partial \langle H\rangle}{\partial \theta_j}\right|^2 \mathrm{d}\theta_1\ldots\mathrm{d}\theta_m - \left(\mathbf{E}(\frac{\partial \langle H\rangle}{\partial \theta_j})\right)^2, \tag{10}$$

for $j = 1, \ldots, m$.

We will compute the expectation and variance of the gradients in the next two sections.

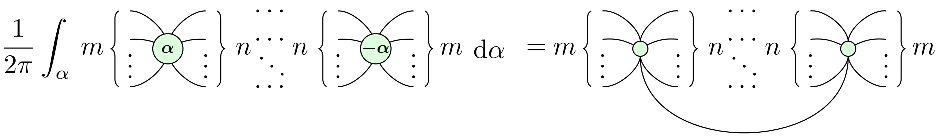
3.2 The expectation of gradients

In this section, we will compute the expectation in (9). As shown in theorem 2, the integration

$$\frac{1}{2\pi}\int_{\theta_k}\frac{\partial \langle H\rangle}{\partial \theta_j}\mathrm{d}\theta_k,$$

for $k = 1, \ldots, m$, is also an integration of a ZX-diagram over its parameter θ_k. With the following lemma, the integration can be represented as a ZX-diagram again.

Lemma 1. *The following equation holds.*



Proof. The proof is given in appendix B. □

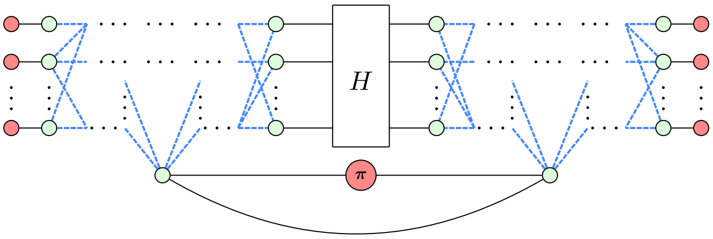
With this lemma, we can prove the following theorem.

Theorem 3. *Under assumption 1, the integration*

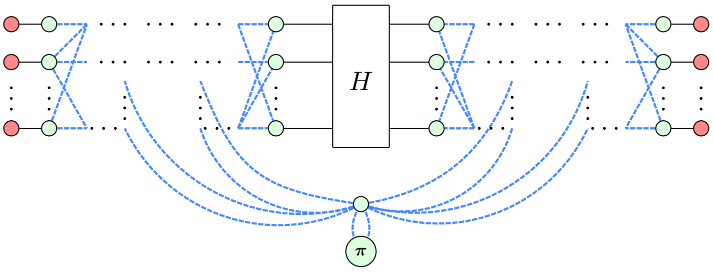
$$\frac{1}{2\pi}\int_{\theta_j}\frac{\partial \langle H\rangle}{\partial \theta_j}\mathrm{d}\theta_j = 0.$$

Proof. Using the relation in lemma 1 on the ZX-diagram in theorem 2, we have

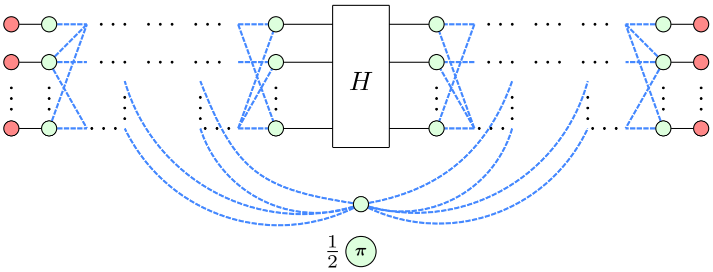
$\frac{2^n}{|c|^2}\cdot\frac{1}{2\pi}\int_{\theta_j}\frac{\partial\langle H\rangle}{\partial\theta_j}\mathrm{d}\theta_j \;=$



$=$



$=$



$= 0.$

□

As a corollary, the expectation of the gradient in (9) is zero.

Corollary 1. *Under assumption 1, the expectation*

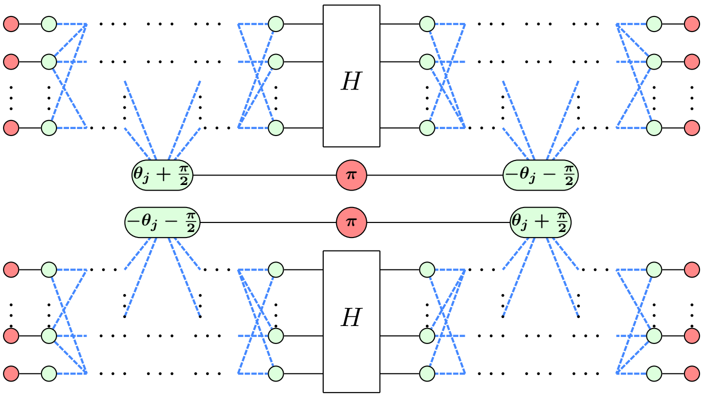
$$\mathbf{E}(\frac{\partial\langle H\rangle}{\partial\theta_j}) = \frac{1}{(2\pi)^m}\int_{\theta_1}\cdots\int_{\theta_m}\frac{\partial\langle H\rangle}{\partial\theta_j}\mathrm{d}\theta_1\ldots\mathrm{d}\theta_m = 0,\ \textit{for } j = 1,\ldots,m.$$

3.3 The variance of gradients

In this section, we will compute the variance in (8).

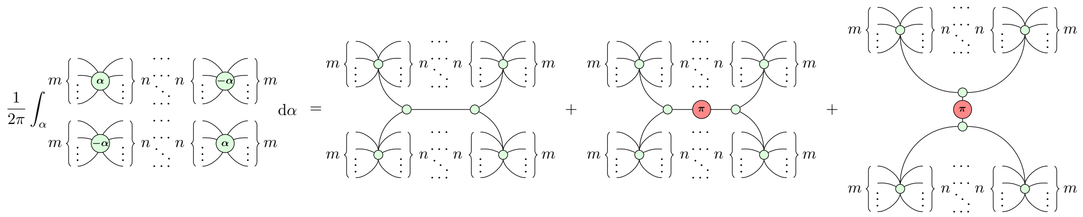
Because the gradient $\frac{\partial\langle H\rangle}{\partial\theta_j}$ is a real number and the expectation is 0, by (8), the variance is the expectation of $\left(\frac{\partial\langle H\rangle}{\partial\theta_j}\right)^2$, which can be represented as follows by theorem 2.

$\frac{4^n}{|c|^2}\cdot\left[\frac{\partial\langle H\rangle}{\partial\theta_j}\right]^2 =$ (11)



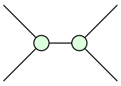
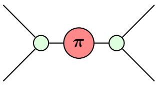
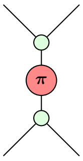
Similar to lemma 1, we can prove the following lemma.

Lemma 2. *The following equation holds.*



Proof. Refer to appendix C. □

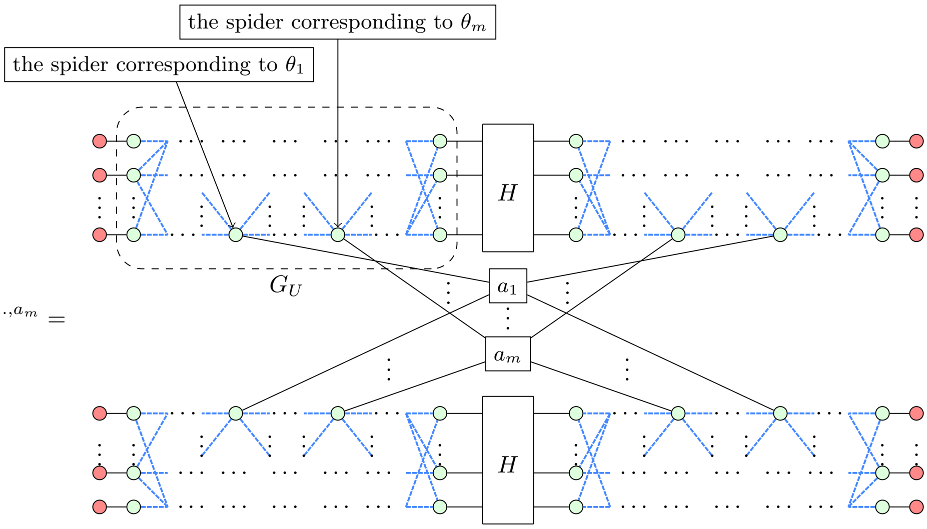
There exist three terms after integration. Hence, computing the variance of gradients is much more complicated than computing the expectation. We denote the three ZX-diagrams in lemma 2 as

$T_1 =$  , $T_2 =$  , $T_3 =$  . (12)

And we introduce a new notation

$$V_U^{a_1,\ldots,a_m},\ a_j \in \{T_1, T_2, T_3\},$$

to represent the following ZX-diagram.

$V_U^{a_1,\ldots,a_m} =$  (13)

the spider corresponding to θ_m
the spider corresponding to θ_1
G_U
H
a_1
a_m
H

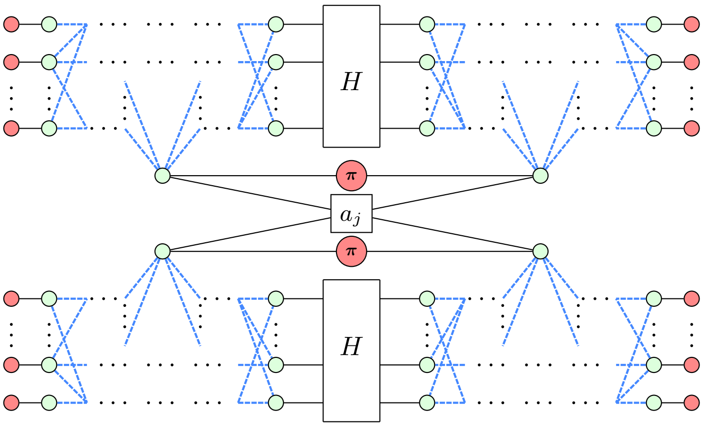
Here, $U(\theta_1, \ldots, \theta_m)$ is a PQC with m parameters and G_U is the graph-like ZX-diagram corresponding to U. With this notation, we have the following theorem.

Theorem 4. *Under assumption 1, the following equation holds.*

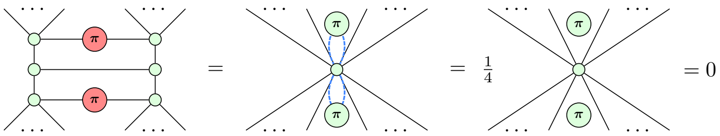
$$\textbf{Var}\left(\frac{\partial\langle H\rangle}{\partial\theta_j}\right) = \frac{|c|^2}{4^n} \cdot \sum_{a_k\in\{T_1,T_2,T_3\},\ k\neq j} V_U^{a_1,\ldots,a_{j-1},T_2,a_{j+1},\ldots,a_m}.$$

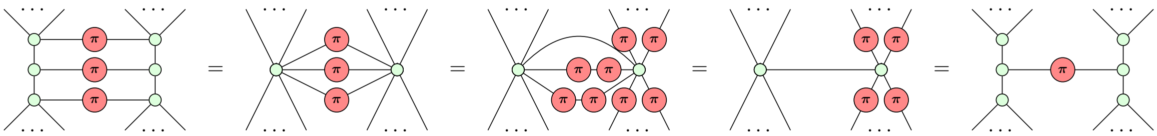
Proof. By (11) and lemma 2, we have the following equation.

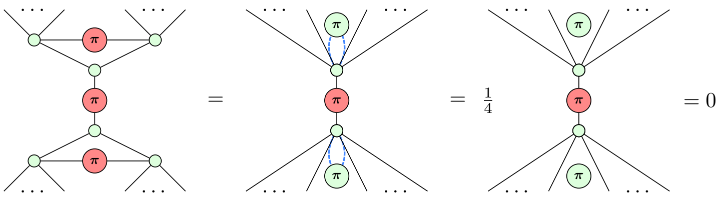
$$\frac{4^n}{|c|^2} \cdot \frac{1}{2\pi} \int_{\theta_j} \left[\frac{\partial \langle H \rangle}{\partial \theta_j} \right]^2 \mathrm{d}\theta_j = \sum_{a_j \in \{T_1, T_2, T_3\}}$$



And by the following equations,

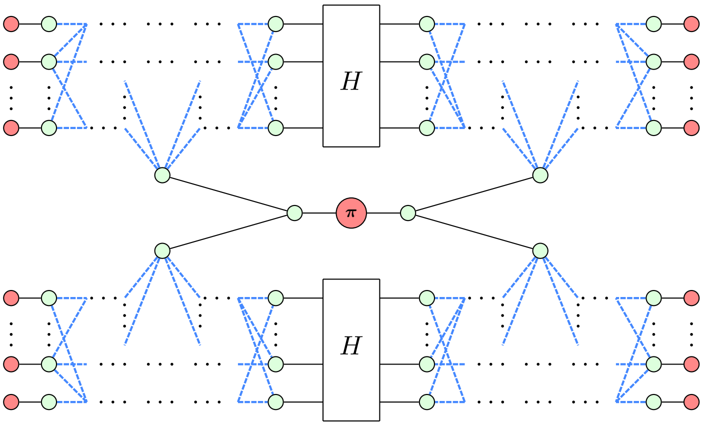






we have

$$\frac{4^n}{|c|^2} \cdot \frac{1}{2\pi} \int_{\theta_j} \left[\frac{\partial \langle H \rangle}{\partial \theta_j} \right]^2 \mathrm{d}\theta_j =$$



Then, by the definition of $V_U^{a_1,\ldots,a_m}$, we get

$$\mathbf{Var}\left(\frac{\partial \langle H \rangle}{\partial \theta_j} \right) = \frac{|c|^2}{4^n} \cdot \sum_{a_k \in \{T_1, T_2, T_3\},\ k \neq j} V_U^{a_1,\ldots,a_{j-1},T_2,a_{j+1},\ldots,a_m}.$$

□

Hence, to compute the variance, we need to sum over 3^{m-1} terms of the tensor

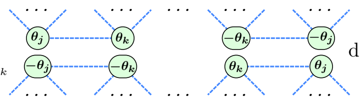
$$V_U^{a_1,\ldots,a_{j-1},T_2,a_{j+1},\ldots,a_m}.$$

It seems inaccessible when m is large. But in many cases, we have simple ways to compute this sum.

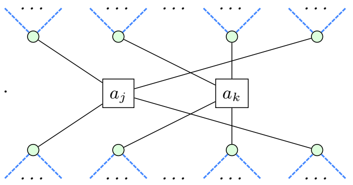
Let us consider two spiders W_j, W_k corresponding to the parameters θ_j, θ_k in G_U. Suppose that W_j and W_k are connected with a Hadamard edge. Then by the following lemma, the Hadamard edge can be removed after integration over θ_j and θ_k.

Lemma 3. *The following equation holds.*

$$\frac{1}{(2\pi)^2}\int_{\theta_j}\int_{\theta_k}$$



$$\mathrm{d}\theta_j\mathrm{d}\theta_k = \sum_{a_j,a_k\in\{T_1,T_2,T_3\}} M_{a_j,a_k}\cdot$$



Here M_{a_j,a_k} *is the* 3×3 *matrix*

$$M = \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & 1 \end{pmatrix}$$

Proof. Refer to appendix D. □

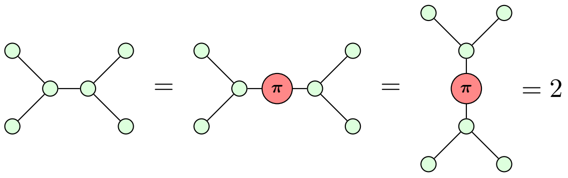
Applying this lemma to the variance recursively, we can remove all the Hadamard edges connecting two parameterized spiders. And the big tensor

$$V_U^{a_1,\ldots,a_{j-1},T_2,a_{j+1},\ldots,a_m}$$

will be broken into smaller tensors that are connected with M. It is a new tensor network whose structure is similar to G_U. To compute the variance, the only thing we need to do is contracting this new tensor network. Figure 5 demonstrates the above procedure for the case that all spiders in G_U are parameterized and are connected with Hadamard edges. The tensor $\tilde{I}_{a_1,\ldots,a_k}$ is related to the input state, while the tensor $\tilde{H}_{c_1,\ldots,c_m}$ is related to the Hamiltonian H. And P_2 is a projection that has only one non-zero entry. That is

$$P_2(2,2,\ldots,2) = 1.$$

Also note that there is a scalar 2 for each internal copy tensor. This scalar comes from the following equation.



(14)

In conclusion, computing the variance of gradients is reduced to contracting a tensor network corresponding to the circuit. In the next section, we will use these techniques to analyze the BP phenomenon several for concrete PQCs.

4 Analyzing the BP phenomenon for four PQCs

In this section, we will analyze the BP phenomenon for 4 PQCs with the techniques introduced in section 3.

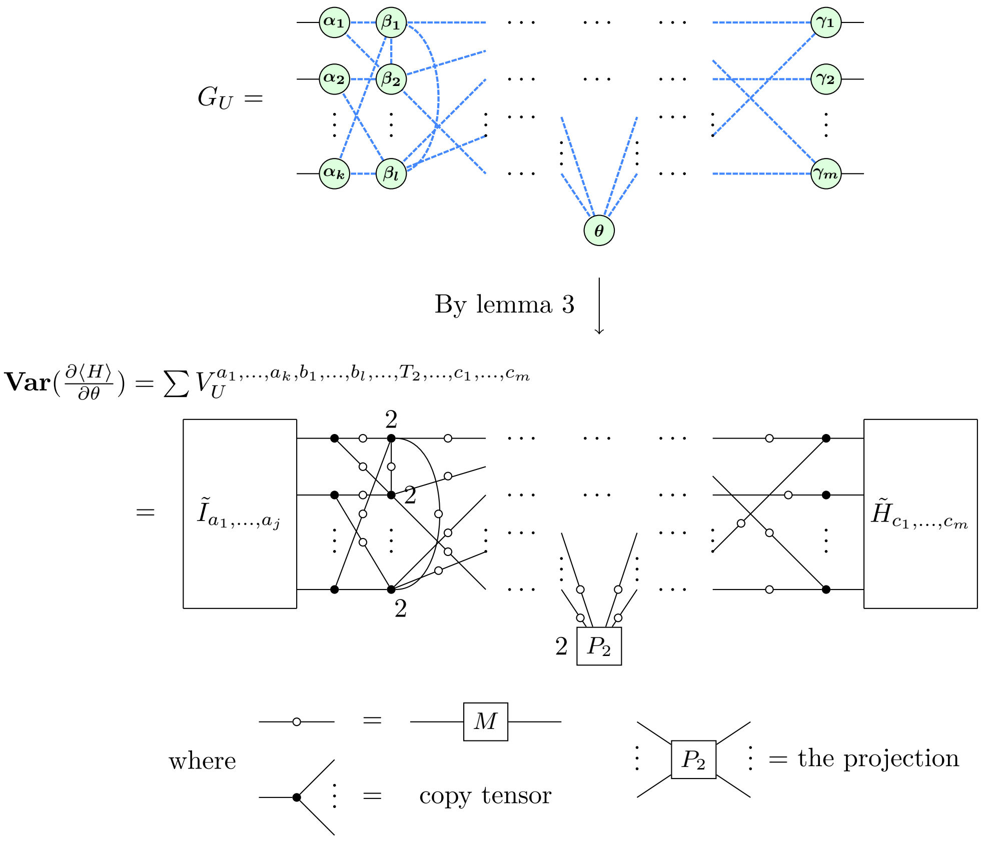
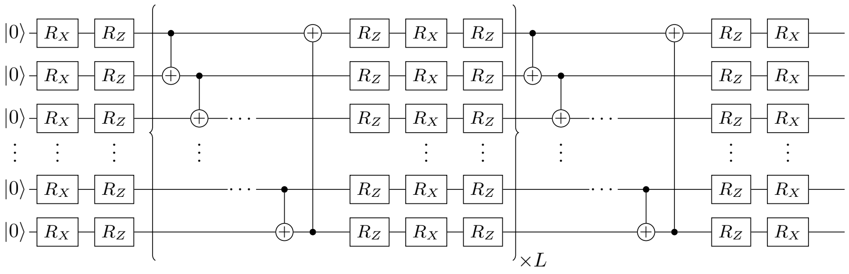

$G_U =$
α_1
α_2
α_k
β_1
β_2
β_l
θ
γ_1
γ_2
γ_m
By lemma 3
$\mathbf{Var}(\frac{\partial\langle H\rangle}{\partial\theta}) = \sum V_U^{a_1,\ldots,a_k,b_1,\ldots,b_l,\ldots,T_2,\ldots,c_1,\ldots,c_m}$
$=$
$\tilde{I}_{a_1,\ldots,a_j}$
2
2
2
2
P_2
$\tilde{H}_{c_1,\ldots,c_m}$
where
$=$
M
$=$
copy tensor
P_2
$=$ the projection

Figure 5: Computing the variance with the tensor network

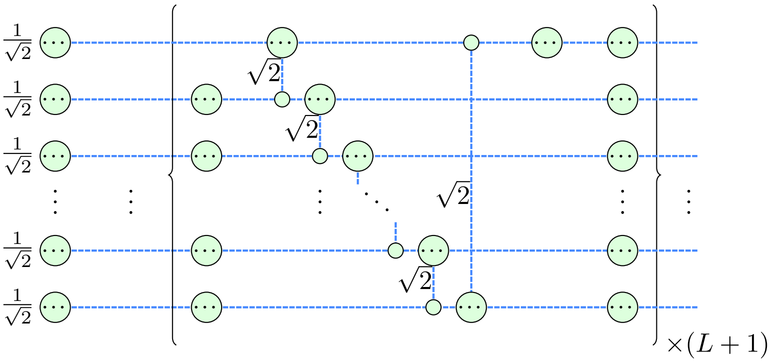
4.1 Hardware-efficient ansatz

Consider a hardware-efficient ansatz [2] PQC of the following form.



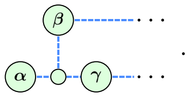
(15)

Suppose the circuit is of n-qubits. By the conversion rules in (4), we get a graph-like ZX-diagram where all spiders are parameterized.



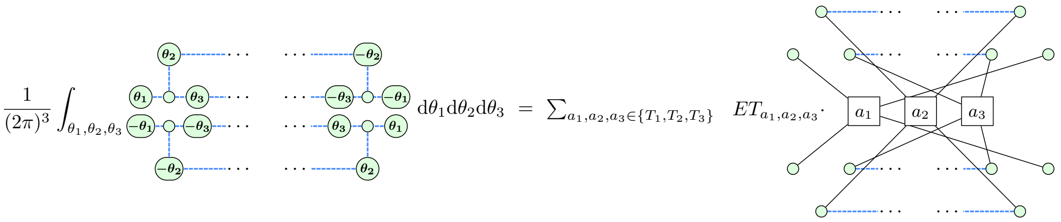
Here, the Z-spider with "…" represents a Z-spider with a parameter.

By lemma 3, we can remove most Hadamard edges. Then the remaining part consists of



By the similar techniques in lemma 3, we can prove the following lemma.

Lemma 4. *The following equation holds.*

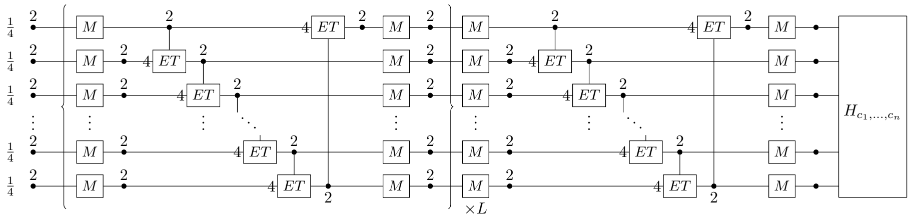


Here ET_{a_1,a_2,a_3} *can be regarded as a* $3\times 3\times 3$ *tensor*

$$ET[1,\cdot,\cdot]=\frac{1}{8}\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix},\quad ET[2,\cdot,\cdot]=\frac{1}{8}\begin{pmatrix}0&1&0\\1&0&0\\0&0&0\end{pmatrix},\quad ET[3,\cdot,\cdot]=\frac{1}{8}\begin{pmatrix}0&0&1\\0&0&0\\1&0&0\end{pmatrix}.$$

Proof. Refer to appendix E. □

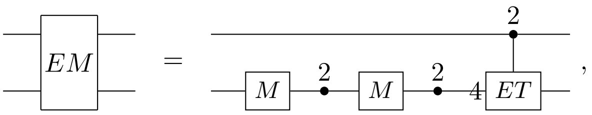
Then we can construct a tensor network which is simlar to figure 5 as follows.



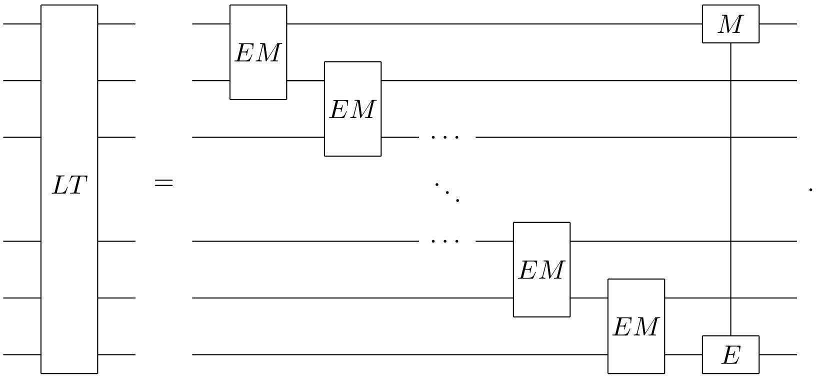
(16)

And if we want to compute the variance $\mathbf{Var}(\frac{\partial\langle H\rangle}{\partial\theta_j})$, we can just simply replace the copy tensor corresponding to θ_j in the above tensor network with the projection P_2.

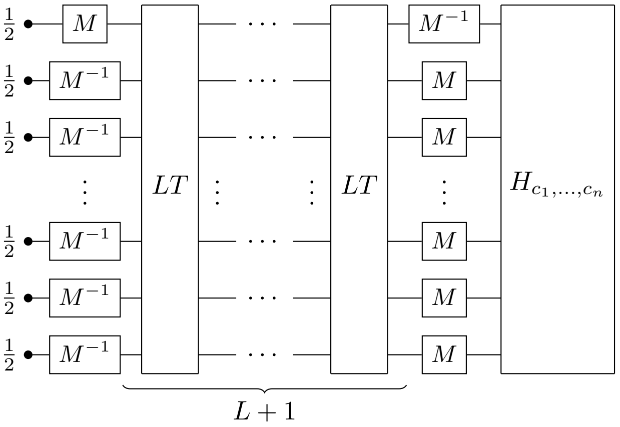
If we denote



then a layer can be represented as



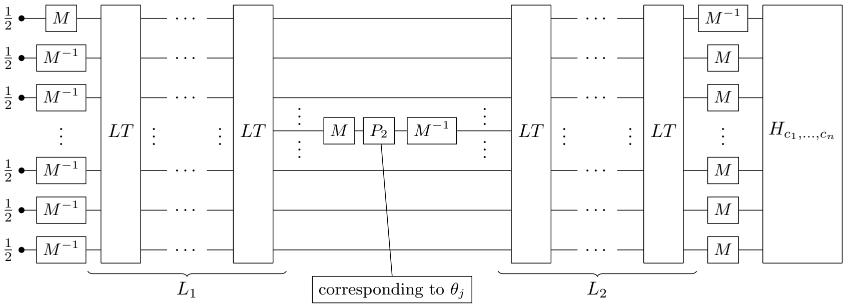
And the whole tensor network in (16) is



.

Hence, the variance will be

$$\mathbf{Var}\left(\frac{\partial\langle H\rangle}{\partial\theta_j}\right) =$$



$$. \tag{17}$$

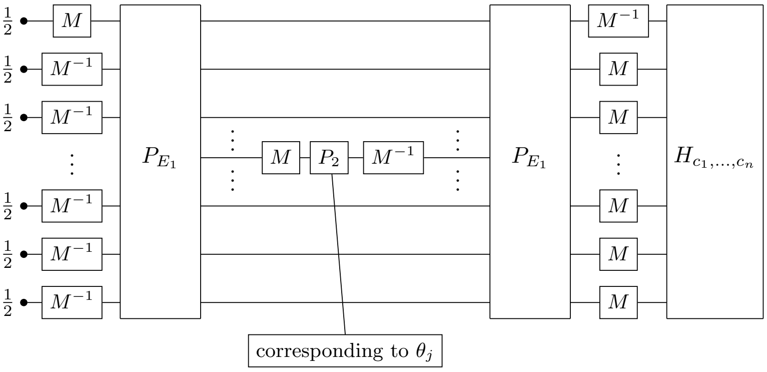
We can prove that only 2 eigenvalues of the matrix LT are 1 and the norms of other eigenvalues are less than 1 (for the complete proof, please refer to appendix F). Moreover, the eigenspace corresponding to the eigenvalue 1 is generated with two vectors

$$E_1 = \operatorname{span}\{v_{1,2} \otimes \cdots \otimes v_{1,2}, v_{1,3} \otimes \cdots \otimes v_{1,3}\}, \quad v_{1,2} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \ v_{1,3} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}. \tag{18}$$

Hence, LT^d will converge to P_{E_1}, the projection to the eigenspace E_1, exponentially, as $d \to \infty$.

If we replace LT^{L_1} and LT^{L_2} with the projection P_{E_1}, then the (17) will become

$$\lim_{L_1,L_2\to\infty} \mathbf{Var}\left(\frac{\partial\langle H\rangle}{\partial\theta_j}\right) =$$



$$. \tag{19}$$

This term is

$$4 \cdot \frac{1}{4^n} \operatorname{Tr}(H^2), \tag{20}$$

which is exponentially small. Thus, there exist barren plateaus in the hardware-efficient ansatz. More precisely, the variance $\mathbf{Var}\left(\frac{\partial\langle H\rangle}{\partial\theta}\right)$ is exponentially (in L) close to a exponentially small (in n) value (20).

Theorem 5. *The variance of gradients in the hardware-efficient ansatz defined in* (15) *vanishes exponentially as the qubit number n and the layer number L grow up.*

Note that the above analysis can be generalized to any hardware-efficient ansätze if the entangler connects all of the qubits.

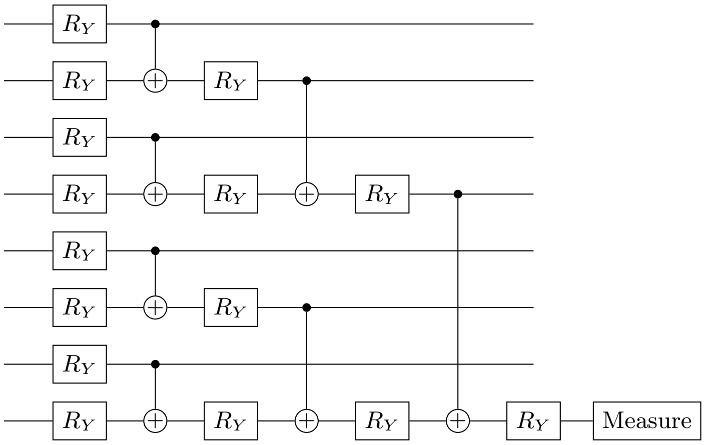
4.2 Tree tensor network ansatz

The tree tensor network is a special kind of tensor networks with tree structures. And the quantum analog of the tree tensor network is developed in [36]. In [46], it was proved that the sum of the variance

$$\sum_{j=1}^{n} \mathbf{Var}\left(\frac{\partial \langle H \rangle}{\partial \theta_j}\right)$$

will not vanish exponentially. In this section, we will prove that not only the sum of the variance but also the variance of each parameter vanishes polynomially.

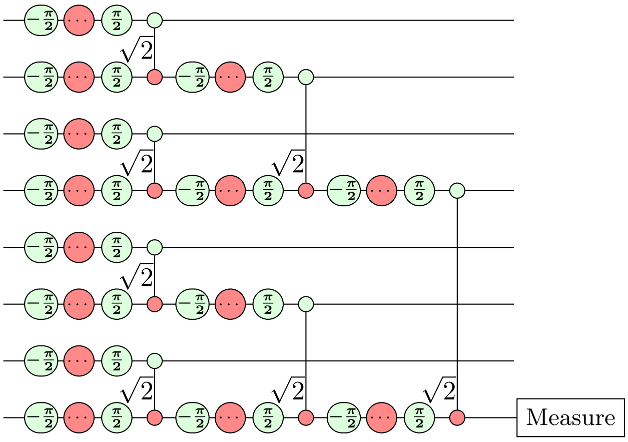
Consider the tree tensor network ansatz with n-qubit of the following form.



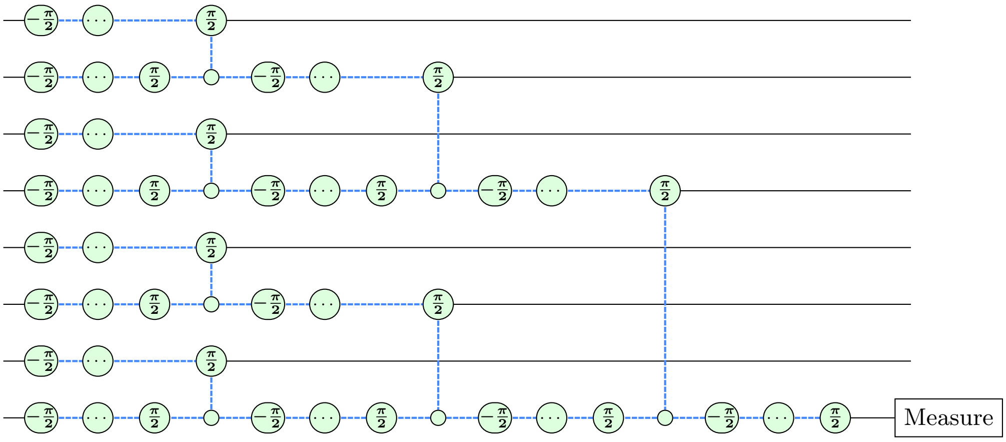
To analyse the BP phenomenon in this ansatz, we first use the gate decomposition

$$R_Y(\theta) = R_Z(\frac{\pi}{2})R_X(\theta)R_Z(-\frac{\pi}{2})$$

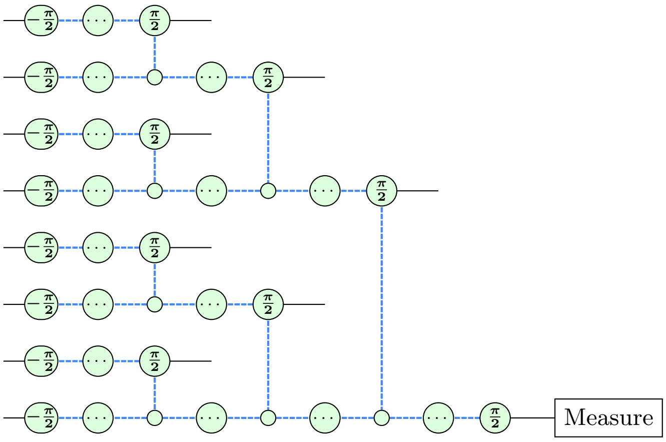
to convert the PQC to a ZX-diagram as follows.



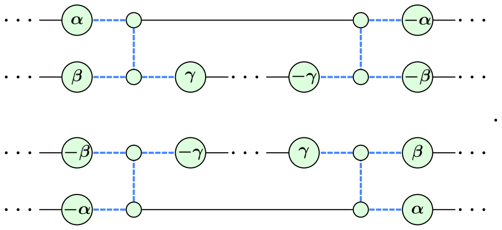
The X-spiders with phase "..." are spiders with parameters. And the ZX-diagram can be rewritten to a graph-like ZX-diagram as follows.



By using the rewriting rule (**lc**) in [25], we can remove the spiders with phases $\pm\frac{\pi}{2}$.

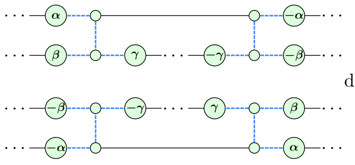


By (11), the building block of the variance $\frac{\partial\langle H\rangle}{\partial\theta_j}^2$ is

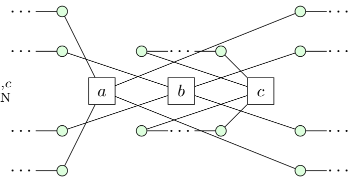


We can prove that (for the complete proof, please refer to appendix F), after integration over the parameters α, β, γ, the building block will become

$$\frac{1}{(2\pi)^3}\int_{\alpha,\beta,\gamma}$$



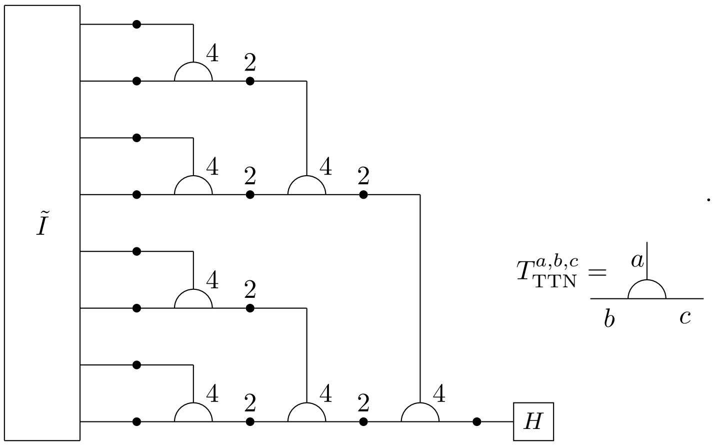
$$\mathrm{d}\alpha\mathrm{d}\beta\mathrm{d}\gamma = \sum_{a,b,c\in T_1,T_2,T_3} T_{\mathrm{TTN}}^{a,b,c} \qquad (21)$$



Here, T_{TTN} is a rank-3 tensor defined as follows.

$$T_{\mathrm{TTN}}[1,\cdot,\cdot] = \frac{1}{16}\begin{pmatrix}1 & 0 & 1\\ 0 & 1 & 0\\ 1 & 0 & 1\end{pmatrix}, \quad T_{\mathrm{TTN}}[2,\cdot,\cdot] = \frac{1}{16}\begin{pmatrix}1 & 0 & -1\\ 0 & 1 & 0\\ -1 & 0 & 1\end{pmatrix}, \quad T_{\mathrm{TTN}}[3,\cdot,\cdot] = \frac{1}{16}\begin{pmatrix}1 & 0 & 1\\ 0 & 1 & 0\\ 1 & 0 & 1\end{pmatrix}.$$

Hence, the variance of $\frac{\partial \langle H \rangle}{\partial \theta_j}$ can be obtained by replacing one of the copy tensor with the projection P_2 in the following tensor network.



. (22)

Now let us analyse this tensor network.

Since the Hamiltonian H is a 1-qubit Hermitian operator, it can be expressed as

$$H = k_0 I + k_1 X + k_2 Y + k_3 Z, \quad k_j \in \mathbb{R}.$$

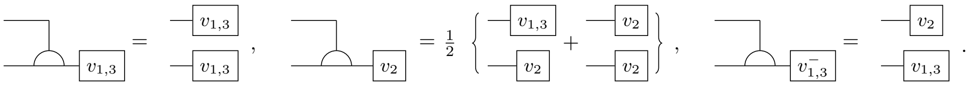
Then

$$\tilde{H} = 2k_0^2 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + 2(k_1^2 + k_3^2) \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + 2k_2^2 \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}.$$

We denote

$$v_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad v_{1,3} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \quad v_{1,3}^{-} = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}.$$

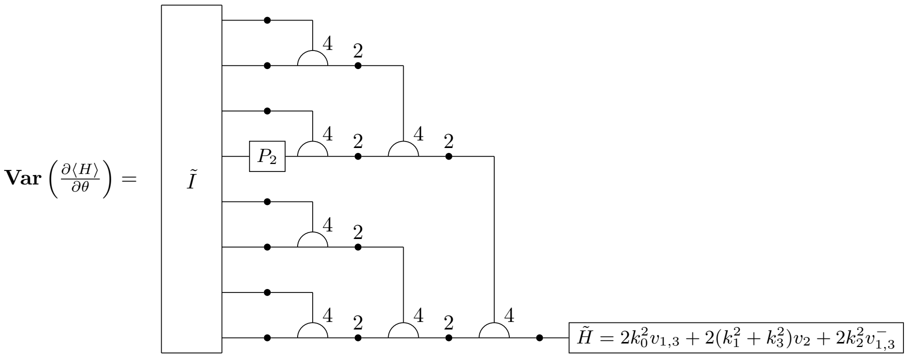
Note that the building block of this tensor network is $8 \cdot T_{\mathrm{TTN}}$. By the definition of T_{TTN}, we have



(23)

With the above equations, we can compute the variance simply.

For example, consider the following variance.

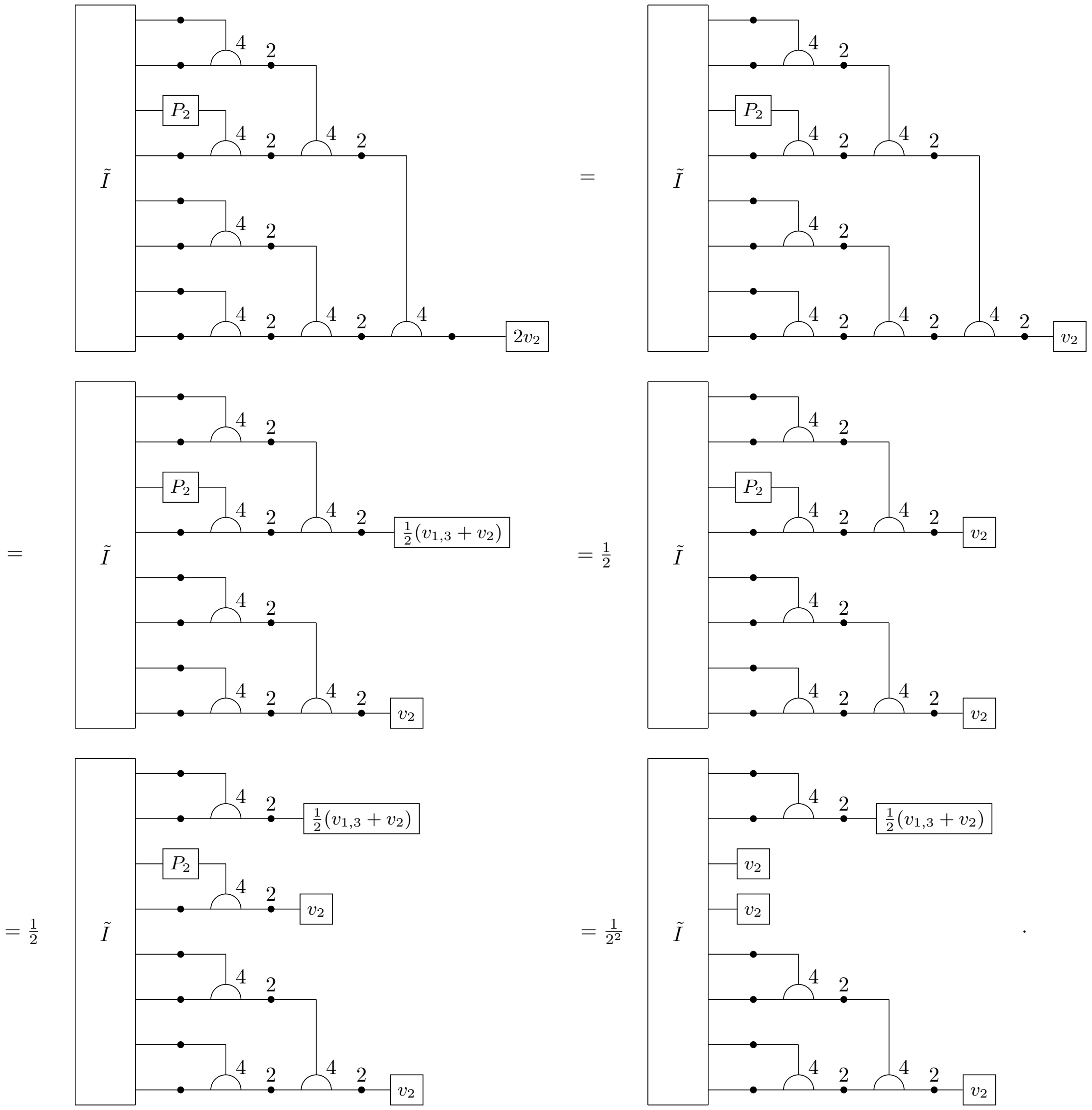

$\mathbf{Var}\left(\frac{\partial \langle H \rangle}{\partial \theta}\right) =$
$\tilde{I}$
P_2
4
2
4
2
4
2
4
2
4
2
4
2
4
$\tilde{H} = 2k_0^2 v_{1,3} + 2(k_1^2 + k_3^2) v_2 + 2k_2^2 v_{1,3}^{-}$

.

It is a linear function of $\tilde{H}$. Hence, we can analyze each term of $\tilde{H}$ individually.

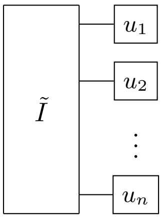
Since $P_2 v_{1,3} = 0$, the first term $2k_0^2 v_{1,3}$ in $\tilde{H}$ will become 0.

Now let us consider the second term $2(k_1^2 + k_3^2)v_2$. With (23), we can expand the variance as follows.



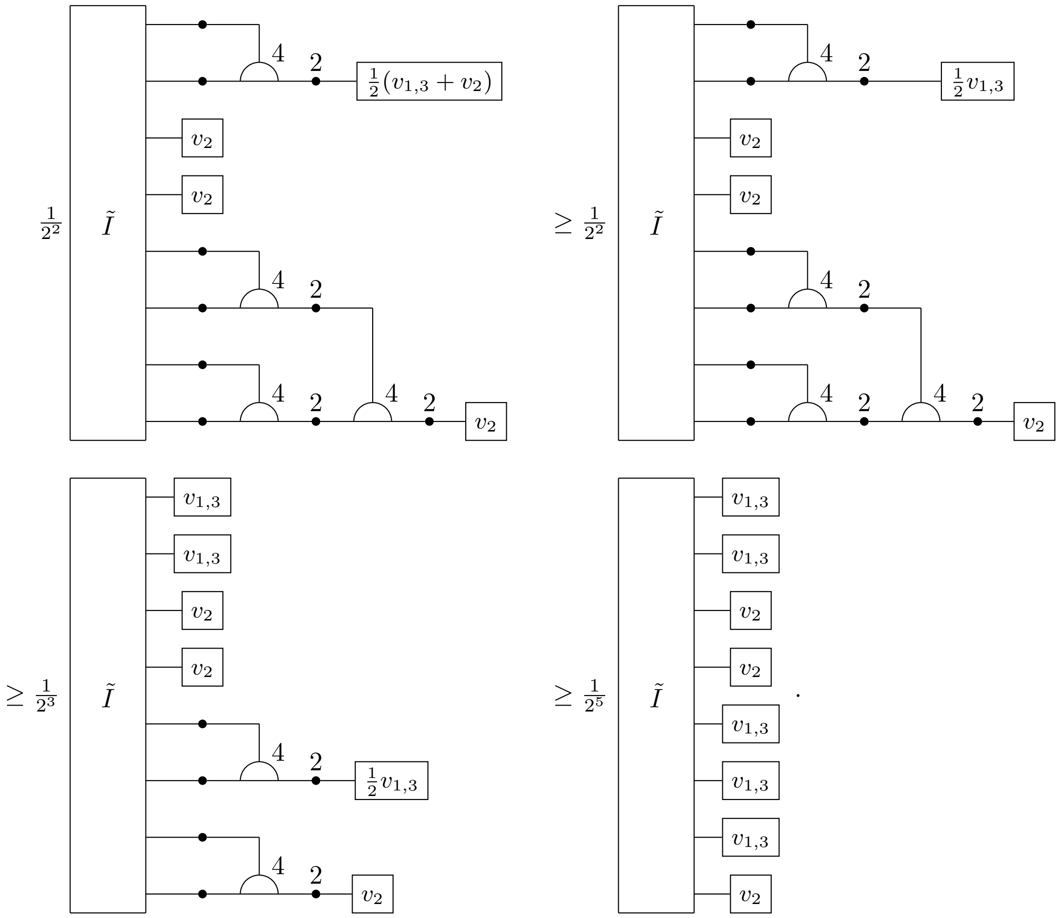
Expanding it recursively, the variance can be represented as a summation of terms of the following form,

$$\tilde{I}(u_1, \ldots, u_n) =$$



$$, \text{ where } u_j \in \{v_2, v_{1,3}\}. \tag{24}$$

And by the definition of $\tilde{I}$, each of the term $\tilde{I}(u_1,\dots,u_n) \geq 0$. Hence, we can get a lower bound,



Similarly, we have a lower bound for the term $v_{1,3}^{-}$.

For the general case of n-qubit, we can prove that it has a lower bound.

Theorem 6. *For tree tensor network ansatz shown in section 4.2, if*

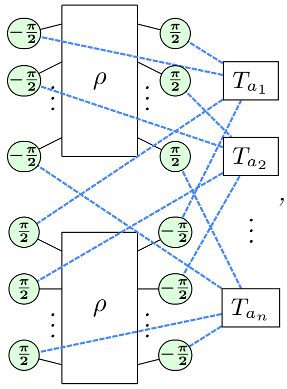
$$H = k_0 I + k_1 X + k_2 Y + k_3 Z,$$

then we have

$$\mathbf{Var}\left(\frac{\partial \langle H \rangle}{\partial \theta}\right) \geq \frac{k_1^2 + k_3^2}{n^2}\tilde{I}(u_1,\dots,u_n) + \frac{k_2^2}{n^2}\tilde{I}(w_1,\dots,w_n), \tag{25}$$

for some $u_j, w_j \in \{v_{1,3}, v_2, v_{1,3}^{-}\}$. Here $\tilde{I}(u_1,\dots,u_n)$ is defined in (24). And $\tilde{I}$ is a 3^n dimensional tensor which only depends on the input state. If the input state is ρ, then $\tilde{I}$ is defined as follows.

$\tilde{I}^{a_1,\dots,a_n} \quad =$



, where $a_j \in \{1, 2, 3\}$.

Proof. See appendix F.2. □

Note that $\tilde{I}(u_1, \ldots, u_n)$ only depends on the input state. If

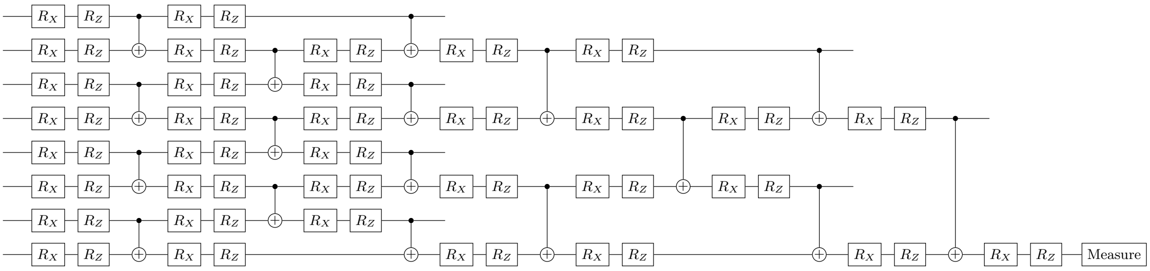
$$\tilde{I}(u_1, \ldots, u_n) \in \Omega(\frac{1}{\mathrm{poly}(n)}) \text{ or } \tilde{I}(w_1, \ldots, w_n) \in \Omega(\frac{1}{\mathrm{poly}(n)}),$$

there exist no barren plateau in the tree tensor network ansatz.

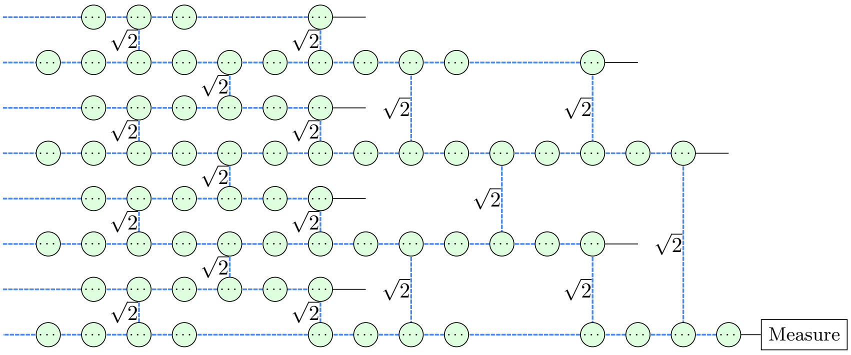
4.3 QCNN

QCNN was developed in [35]. It was proved that there exists no barren plateau in the QCNN ansatz if the subblocks are unitary 2-design [47]. In this section, we will use the ZX-calculus to analyze the BP phenomenon in a QCNN ansatz without the assumption of unitary 2-design.

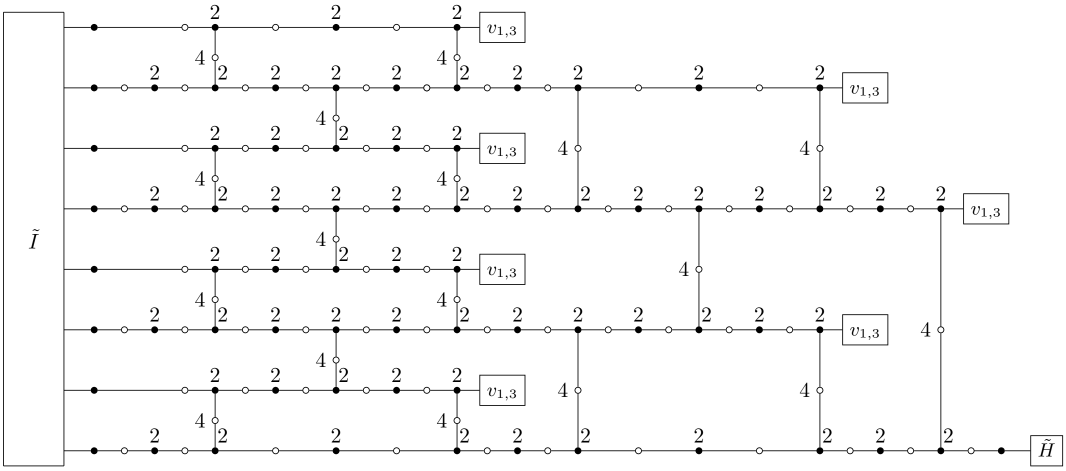
Consider a QCNN ansatz as follows.



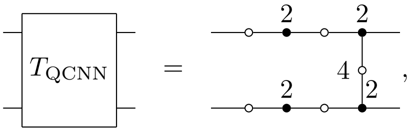
It can be represented as the following ZX-diagram.



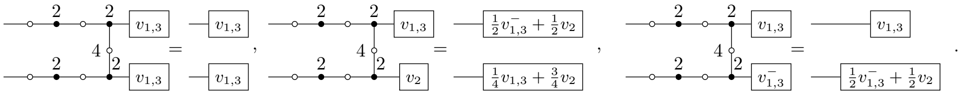
where the Z-spiders with "..." are parameterized. Note that this is a graph-like ZX-diagram whose spiders are all parameterized. Hence, by using lemma 3, the variance can be obtained by replacing one of the copy tensors with the projection P_2 in the following tensor network.



If we denote



then we have the following equations

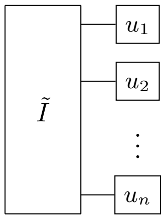


2
2
4
2
2
$v_{1,3}$
$v_{1,3}$
=
$v_{1,3}$
$v_{1,3}$
,
$v_{1,3}$
v_2
=
$\frac{1}{2}v_{1,3}^{-}+\frac{1}{2}v_2$
$\frac{1}{4}v_{1,3}+\frac{3}{4}v_2$
,
$v_{1,3}$
$v_{1,3}^{-}$
=
$v_{1,3}$
$\frac{1}{2}v_{1,3}^{-}+\frac{1}{2}v_2$
.

(26)

By using (26), we can expand the variance as a sum of terms of the following form.

$\tilde{I}(u_1,\ldots,u_n) =$



, where $u_j \in \{v_2, v_{1,3}, v_{1,3}^{-}\}$.

And each of these terms is non-negative. Hence, similar to the analysis of tree tensor network ansatz, we can prove that the variance of gradients in the QCNN ansatz has a lower bound.

Theorem 7. *For the QCNN ansatz shown in section 4.3, if*

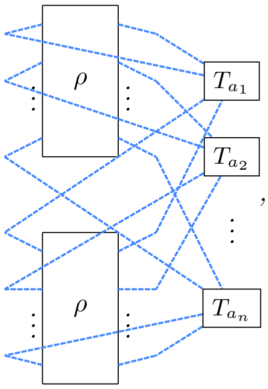
$$H = k_0 I + k_1 X + k_2 Y + k_3 Z,$$

then we have

$$\mathbf{Var}\left(\frac{\partial\langle H\rangle}{\partial\theta}\right) \geq \frac{k_2^2+k_3^2}{n^9}\tilde{I}(u_1,\ldots,u_n) + \frac{k_1^2}{n^9}\tilde{I}(w_1,\ldots,w_n), \tag{27}$$

for some $u_j, w_j \in \{v_{1,3}, v_2, v_{1,3}^{-}\}$. Here $\tilde{I}$ is a 3^n dimensional tensor which only depends on the input state ρ.

$\tilde{I}^{a_1,\ldots,a_n} =$



, *where $a_j \in \{1,2,3\}$.*

Proof. See appendix F.3. □

Hence, if provided

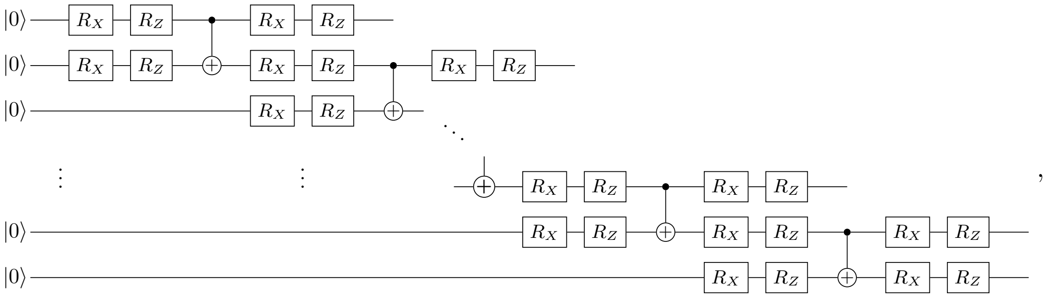
$$\tilde{I}(u_1,\ldots,u_n) \in \Omega(\frac{1}{\mathrm{poly}(n)}) \text{ or } \tilde{I}(w_1,\ldots,w_n) \in \Omega(\frac{1}{\mathrm{poly}(n)}),$$

then there exists no barren plateau in the QCNN ansatz.

4.4 MPS-inspired ansatz

The matrix product state (MPS) is a special structure of tensor networks. And it is widely used in quantum physics and machine learning [48, 49]. There are also PQCs with a similar structure as MPS, and we call it MPS-inspired ansatz. It has been shown that MPS-inspired ansatz can be implemented efficiently in quantum computers with fewer qubits [37]. We will analyze the BP phenomenon in MPS-inspired ansatz in this section.

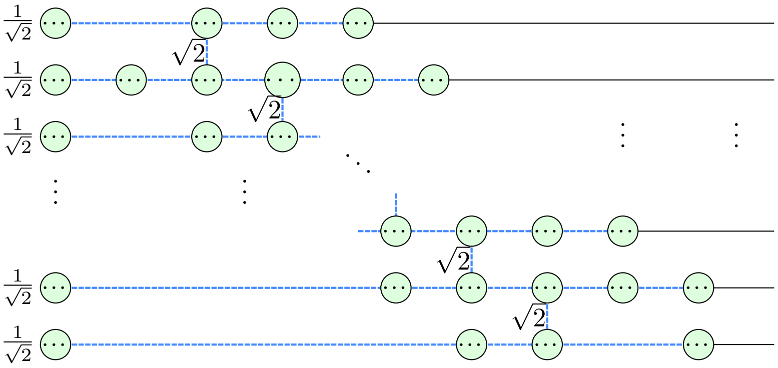
Let us consider the following MPS-inspired ansatz



,

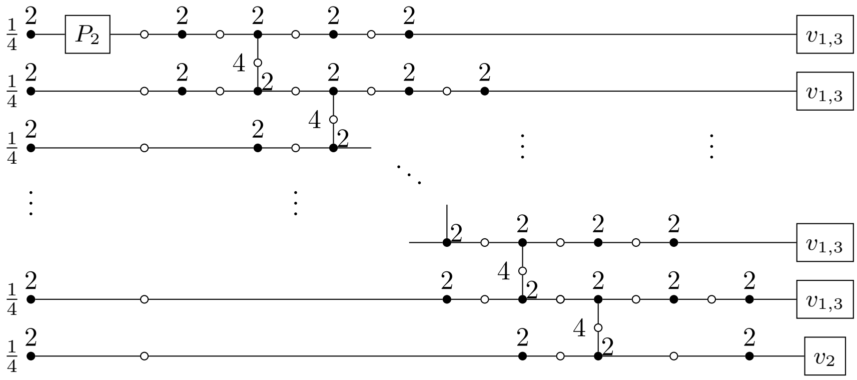
and the Hamiltonian $H = I \otimes I \cdots \otimes I \otimes X$. We will prove that the variance $\mathbf{Var}\left(\frac{\partial \langle H \rangle}{\partial \theta_1}\right)$ is exponentially small. Here θ_1 is the parameter of the first R_X gate applying on the first qubit.

Firstly, we convert the PQC into a ZX-diagram as follows.



This is a graph-like ZX-diagram whose spiders are all parameterized. We can use lemma 3 to represent the variance as the following tensor network.

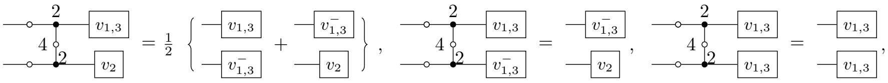
$$\mathbf{Var}\left(\frac{\partial \langle H \rangle}{\partial \theta_1}\right) =$$



By using

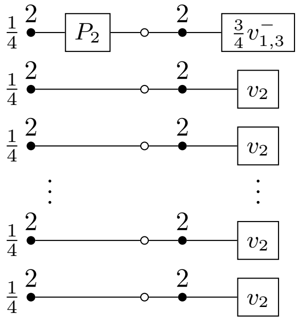
$$2Mv_2 = \frac{1}{2}\left(v_2 + v_{1,3}^{-}\right), \quad 2Mv_{1,3} = v_{1,3},$$

and



we can simplify the variance as

$$\mathbf{Var}\left(\frac{\partial\langle H\rangle}{\partial\theta_1}\right) =$$



$$= \frac{1}{2^{2n-1}} \tag{28}$$

It is exponential in qubit number n. Hence, there exist barren plateaus in the MPS-inspired ansatz.

5 Discussion

We developed powerful techniques to analyze the BP phenomenon in certain quantum neural networks training with the ZX-calculus. The quantum neural networks under consideration are PQCs and the cost function is the expectation $\langle H\rangle$ of the PQC with respect to a given Hamiltonian H. The basic idea of the method is to represent the PQC, the cost function $\langle H\rangle$, and the gradients $\frac{\partial\langle H\rangle}{\partial\theta_j}$ as ZX-diagrams. And then computing the expectation and the variance of the gradient of $\langle H\rangle$ becomes computing the integration of certain ZX-diagrams. We show that these integrations are also ZX-diagrams which can be computed explicitly in many cases.

In principle, these techniques can be used to any given ansatz under assumption 1. We remark that these techniques can be used to analyze the BP phenomenon for PQCs which contain t-design sub-blocks, for example, the PQCs considered in [20, 47]. Because the t-design sub-blocks can be replaced with concrete t-design PQCs and then the techniques proposed in this paper can be applied. In conclusion, we extend the barren plateaus theorem from unitary 2-design circuits to any parameterized quantum circuits under assumption 1.

Using the techniques proposed in this paper, we analyzed 4 kinds of ansätze, including the hardware-efficient ansatz, the tree tensor network ansatz, the QCNN ansatz, and the MPS-inspired ansatz. It is shown that there exist barren plateaus in the hardware-efficient-ansatz and the MPS-inspired ansatz, while there exists no barren plateau in the tree tensor network ansatz and the QCNN ansatz.

Acknowledgment. This work is partially supported by a NSFC grant No.11688101 and by a NKRDP grant No.2018YFA0306702.

References

[1] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O'brien. A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, 5:4213, 2014.

[2] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M Chow, and Jay M Gambetta. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature*, 549(7671):242–246, 2017.

[3] Yudong Cao, Jonathan Romero, Jonathan P Olson, Matthias Degroote, Peter D Johnson, Mária Kieferová, Ian D Kivlichan, Tim Menke, Borja Peropadre, Nicolas PD Sawaya, et al. Quantum chemistry in the age of quantum computing. *Chemical reviews*, 119(19):10856–10915, 2019.

[4] Bela Bauer, Sergey Bravyi, Mario Motta, and Garnet Kin Chan. Quantum algorithms for quantum chemistry and quantum materials science. *arXiv preprint arXiv:2001.03685*, 2020.

[5] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.

[6] Leo Zhou, Sheng-Tao Wang, Soonwon Choi, Hannes Pichler, and Mikhail D Lukin. Quantum approximate optimization algorithm: performance, mechanism, and implementation on near-term devices. *arXiv preprint arXiv:1812.01041*, 2018.

[7] Jin-Guo Liu and Lei Wang. Differentiable learning of quantum circuit born machines. *Physical Review A*, 98(6):062324, 2018.

[8] Seth Lloyd and Christian Weedbrook. Quantum generative adversarial learning. *Physical review letters*, 121(4):040502, 2018.

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

[10] Maria Schuld, Alex Bocharov, Krysta M Svore, and Nathan Wiebe. Circuit-centric quantum classifiers. *Physical Review A*, 101(3):032308, 2020.

[11] Marcello Benedetti, Erika Lloyd, Stefan Sack, and Mattia Fiorentini. Parameterized quantum circuits as machine learning models. *Quantum Science and Technology*, 4(4):043001, 2019.

[12] Chen Zhao and Xiao-Shan Gao. Qdnn: Dnn with quantum neural network layers. *arXiv preprint arXiv:1912.12660*, 2019.

[13] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.

[14] Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. Evaluating analytic gradients on quantum hardware. *Physical Review A*, 99(3):032331, 2019.

[15] Andrea Mari, Thomas R Bromley, and Nathan Killoran. Estimating the gradient and higher-order derivatives on quantum hardware. *Physical Review A*, 103(1):012405, 2020.

[16] James Stokes, Josh Izaac, Nathan Killoran, and Giuseppe Carleo. Quantum natural gradient. *Quantum*, 4:269, 2020.

[17] Ken M Nakanishi, Keisuke Fujii, and Synge Todo. Sequential minimal optimization for quantum-classical hybrid algorithms. *Physical Review Research*, 2(4):043158, 2020.

[18] Jonas M Kübler, Andrew Arrasmith, Lukasz Cincio, and Patrick J Coles. An adaptive optimizer for measurement-frugal variational algorithms. *Quantum*, 4:263, 2020.

[19] Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature communications*, 9 (1):1–6, 2018.

[20] Marco Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J Coles. Cost-function-dependent barren plateaus in shallow quantum neural networks. *arXiv preprint arXiv:2001.00550*, 2020.

[21] Carlos Ortiz Marrero, Mária Kieferová, and Nathan Wiebe. Entanglement induced barren plateaus. *arXiv preprint arXiv:2010.15968*, 2020.

[22] Samson Wang, Enrico Fontana, Marco Cerezo, Kunal Sharma, Akira Sone, Lukasz Cincio, and Patrick J Coles. Noise-induced barren plateaus in variational quantum algorithms. *arXiv preprint arXiv:2007.14384*, 2020.

[23] Bob Coecke and Ross Duncan. Interacting quantum observables. In *Proceedings of the 37th International Colloquium on Automata, Languages and Programming (ICALP)*, Lecture Notes in Computer Science, 2008. DOI: 10.1007/978-3-540-70583-3“˙25.

[24] Bob Coecke and Ross Duncan. Interacting quantum observables: categorical algebra and diagrammatics. *New Journal of Physics*, 13:043016, 2011. DOI: 10.1088/1367-2630/13/4/043016.

[25] Ross Duncan, Aleks Kissinger, Simon Pedrix, and John van de Wetering. Graph-theoretic Simplification of Quantum Circuits with the ZX-calculus. *Quantum*, 4:279, 6 2020. ISSN 2521-327X. DOI: 10.22331/q-2020-06-04-279.

[26] Aleks Kissinger and John van de Wetering. Reducing T-count with the ZX-calculus. *Physical Review A*, 102:022406, 8 2020. DOI: 10.1103/PhysRevA.102.022406.

[27] Alexander Cowtan, Silas Dilkes, Ross Duncan, Will Simmons, and Seyon Sivarajah. Phase Gadget Synthesis for Shallow Circuits. In Bob Coecke and Matthew Leifer, editors, Proceedings 16th International Conference on *Quantum Physics and Logic,* Chapman University, Orange, CA, USA., 10-14 June 2019, volume 318 of *Electronic Proceedings in Theoretical Computer Science*, pages 213–228. Open Publishing Association, 2020. DOI: 10.4204/EPTCS.318.13.

[28] Michael Hanks, Marta P. Estarellas, William J. Munro, and Kae Nemoto. Effective Compression of Quantum Braided Circuits Aided by ZX-Calculus. *Physical Review X*, 10:041030, 2020. DOI: 10.1103/PhysRevX.10.041030.

[29] Ross Duncan. A graphical approach to measurement-based quantum computing. In Mehrnoosh Sadrzadeh Chris Heunen and Edward Grefenstette, editors, *Quantum Physics and Linguistics: A Compositional, Diagrammatic Discourse*. 2013. ISBN 9780199646296. DOI: 10.1093/acprof:oso/9780199646296.001.0001.

[30] Miriam Backens, Hector Miller-Bakewell, Giovanni de Felice, Leo Lobski, and John van de Wetering. There and back again: A circuit extraction tale. *arXiv preprint arXiv:2003.01664*, 2020.

[31] Nicholas Chancellor, Aleks Kissinger, Joschka Roffe, Stefan Zohren, and Dominic Horsman. Graphical Structures for Design and Verification of Quantum Error Correction. *arXiv preprint arXiv:1611.08012*, 2016.

[32] Niel de Beaudrap and Dominic Horsman. The ZX calculus is a language for surface code lattice surgery. *Quantum*, 4, 2020. DOI: 10.22331/q-2020-01-09-218.

[33] Richard D. P. East, John van de Wetering, Nicholas Chancellor, and Adolfo G. Grushin. AKLT-states as ZX-diagrams: diagrammatic reasoning for quantum states. *arXiv preprint arXiv:2012.01219*, 2020.

[34] Bob Coecke, Giovanni de Felice, Konstantinos Meichanetzidis, and Alexis Toumi. Foundations for Near-Term Quantum Natural Language Processing. *arXiv preprint arXiv:2012.03755*, 2020.

[35] Iris Cong, Soonwon Choi, and Mikhail D Lukin. Quantum convolutional neural networks. *Nature Physics*, 15(12):1273–1278, 2019.

[36] Edward Grant, Marcello Benedetti, Shuxiang Cao, Andrew Hallam, Joshua Lockhart, Vid Stojevic, Andrew G Green, and Simone Severini. Hierarchical quantum classifiers. *npj Quantum Information*, 4(1):1–8, 2018.

[37] Jin-Guo Liu, Yi-Hong Zhang, Yuan Wan, and Lei Wang. Variational quantum eigensolver with fewer qubits. *Phys. Rev. Research*, 1:023025, Sep 2019. DOI: 10.1103/PhysRevResearch.1.023025. URL `https://link.aps.org/doi/10.1103/PhysRevResearch.1.023025`.

[38] Bob Coecke and Aleks Kissinger. *Picturing Quantum Processes*. Cambridge University Press, 2017. DOI: 10.1007/978-3-319-91376-6“˙6.

[39] John van de Wetering. ZX-calculus for the working quantum computer scientist. *arXiv preprint arXiv:2012.13966*, 2020.

[40] Miriam Backens. The ZX-calculus is complete for stabilizer quantum mechanics. *New Journal of Physics*, 16(9):093021, 2014. DOI: 10.1088/1367-2630/16/9/093021.

[41] Miriam Backens. Making the stabilizer ZX-calculus complete for scalars. In Chris Heunen, Peter Selinger, and Jamie Vicary, editors, *Proceedings of the 12th International Workshop on Quantum Physics and Logic (QPL 2015)*, volume 195 of *Electronic Proceedings in Theoretical Computer Science*, pages 17–32, 2015. DOI: 10.4204/EPTCS.195.2.

[42] Emmanuel Jeandel, Simon Perdrix, and Renaud Vilmart. A Complete Axiomatisation of the ZX-Calculus for Clifford+T Quantum Mechanics. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '18, pages 559–568, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5583-4. DOI: 10.1145/3209108.3209131.

[43] Emmanuel Jeandel, Simon Perdrix, and Renaud Vilmart. Diagrammatic Reasoning Beyond Clifford+T Quantum Mechanics. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '18, pages 569–578, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5583-4. DOI: 10.1145/3209108.3209139.

[44] Quanlong Wang. *Completeness of the ZX-calculus*. PhD thesis, University of Oxford, 2018.

[45] Emmanuel Jeandel, Simon Perdrix, and Renaud Vilmart. Completeness of the zx-calculus. *Logical Methods in Computer Science*, 6 2020. DOI: 10.23638/LMCS-16(2:11)2020.

[46] Kaining Zhang, Min-Hsiu Hsieh, Liu Liu, and Dacheng Tao. Toward trainability of quantum neural networks. *arXiv preprint arXiv:2011.06258*, 2020.

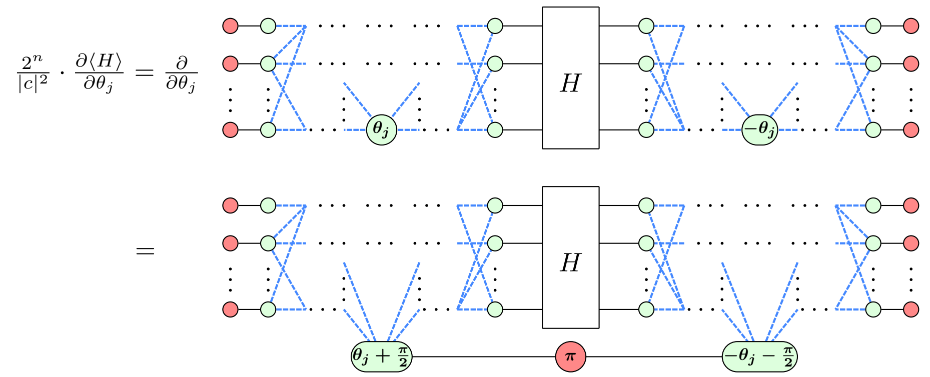
[47] Arthur Pesah, M Cerezo, Samson Wang, Tyler Volkoff, Andrew T Sornborger, and Patrick J Coles. Absence of barren plateaus in quantum convolutional neural networks. *arXiv preprint arXiv:2011.02966*, 2020.

[48] F. Verstraete, V. Murg, and J.I. Cirac. Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems. *Advances in Physics*, 57(2):143–224, 2008. DOI: 10.1080/14789940801912366. URL https://doi.org/10.1080/14789940801912366.

[49] Zhao-Yu Han, Jun Wang, Heng Fan, Lei Wang, and Pan Zhang. Unsupervised generative modeling using matrix product states. *Phys. Rev. X*, 8:031012, Jul 2018. DOI: 10.1103/PhysRevX.8.031012. URL https://link.aps.org/doi/10.1103/PhysRevX.8.031012.

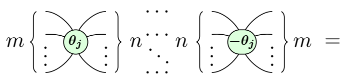
A Proof of theorem 2

Theorem 2. *The gradient can be represented as the following equation.*



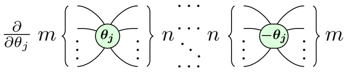
$\frac{2^n}{|c|^2} \cdot \frac{\partial \langle H \rangle}{\partial \theta_j} = \frac{\partial}{\partial \theta_j}$
θ_j
H
$-\theta_j$
$=$
H
$\theta_j + \frac{\pi}{2}$
π
$-\theta_j - \frac{\pi}{2}$

Proof. Consider the spiders corresponding to θ_j. We expand the spiders as follows.

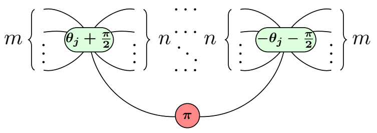


$$= \begin{array}{c} |0\rangle^{\otimes n}\langle 0|^{\otimes m} \cdots\ |0\rangle^{\otimes n}\langle 0|^{\otimes m} + e^{i\theta_j}\ |1\rangle^{\otimes n}\langle 1|^{\otimes m} \cdots\ |0\rangle^{\otimes n}\langle 0|^{\otimes m} \\ +e^{-i\theta_j}\ |0\rangle^{\otimes n}\langle 0|^{\otimes m} \cdots\ |1\rangle^{\otimes n}\langle 1|^{\otimes m} + \ |1\rangle^{\otimes n}\langle 1|^{\otimes m} \cdots\ |1\rangle^{\otimes n}\langle 1|^{\otimes m} \ . \end{array}$$

We take the partial derivative of θ_j on the two sides. We get



$$\begin{aligned} &= ie^{i\theta_j}\ |1\rangle^{\otimes n}\langle 1|^{\otimes m} \cdots\ |0\rangle^{\otimes n}\langle 0|^{\otimes m} - ie^{-i\theta_j}|0\rangle^{\otimes n}\langle 0|^{\otimes m} \cdots\ |1\rangle^{\otimes n}\langle 1|^{\otimes m} \\ &= e^{i(\theta_j + \frac{\pi}{2})}|1\rangle^{\otimes n}\langle 1|^{\otimes m} \cdots\ |0\rangle^{\otimes n}\langle 0|^{\otimes m} + e^{-i(\theta_j + \frac{\pi}{2})}\ |0\rangle^{\otimes n}\langle 0|^{\otimes m} \cdots\ |1\rangle^{\otimes n}\langle 1|^{\otimes m} \\ &= \end{aligned}$$

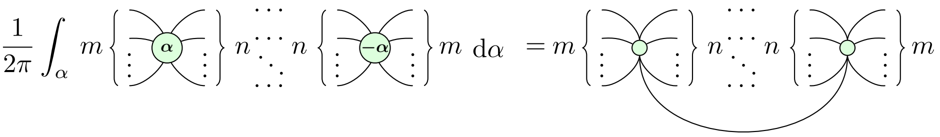


.

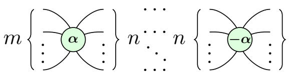
□

B Proof of lemma 1

Lemma 1. *The following equation holds.*



Proof. We expand the spiders as follows.



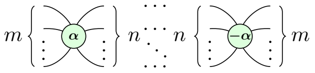
$$= \begin{array}{c} |0\rangle^{\otimes n}\langle 0|^{\otimes m} \cdots\ |0\rangle^{\otimes m}\langle 0|^{\otimes n} + e^{i\alpha}|1\rangle^{\otimes n}\langle 1|^{\otimes m} \cdots\ |0\rangle^{\otimes m}\langle 0|^{\otimes n} \\ +e^{-i\alpha}\ |0\rangle^{\otimes n}\langle 0|^{\otimes m} \cdots\ |1\rangle^{\otimes m}\langle 1|^{\otimes n} + \quad |1\rangle^{\otimes n}\langle 1|^{\otimes m} \cdots\ |1\rangle^{\otimes m}\langle 1|^{\otimes n} \ . \end{array}$$

By

$$\int_{-\pi}^{\pi} e^{k\alpha}\, d\alpha = 0, \quad k = \pm 1,$$

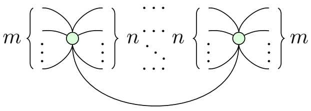
we have

$$\tfrac{1}{2\pi}\int_\alpha \mathrm{d}\alpha$$



$$= |0\rangle^{\otimes n}\langle 0|^{\otimes m}\cdots|0\rangle^{\otimes m}\langle 0|^{\otimes n} + |1\rangle^{\otimes n}\langle 1|^{\otimes m}\cdots|1\rangle^{\otimes m}\langle 1|^{\otimes n}$$

$$=$$

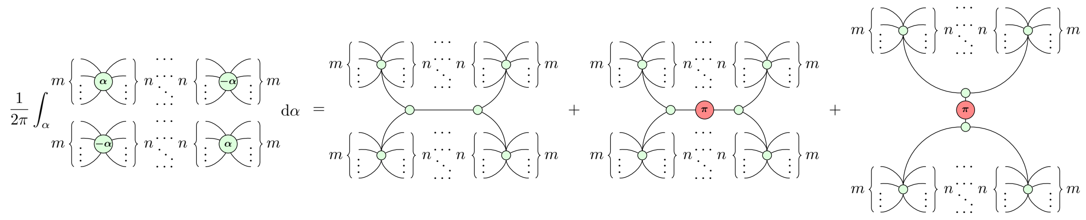


.

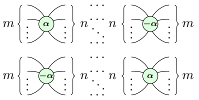
□

C Proof of lemma 2

Lemma 2. *The following equation holds.*



Proof. We expand each spider on the left hand side of the equation as follows.



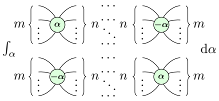
$$
\begin{aligned}
= {} & \begin{matrix}|0\rangle^{\otimes n}\langle 0|^{\otimes m}\cdots|0\rangle^{\otimes m}\langle 0|^{\otimes n}\\ |0\rangle^{\otimes n}\langle 0|^{\otimes m}\cdots|0\rangle^{\otimes m}\langle 0|^{\otimes n}\end{matrix}
+e^{i\alpha}\begin{matrix}|1\rangle^{\otimes n}\langle 1|^{\otimes m}\cdots|0\rangle^{\otimes m}\langle 0|^{\otimes n}\\ |0\rangle^{\otimes n}\langle 0|^{\otimes m}\cdots|0\rangle^{\otimes m}\langle 0|^{\otimes n}\end{matrix}
+e^{-i\alpha}\begin{matrix}|0\rangle^{\otimes n}\langle 0|^{\otimes m}\cdots|1\rangle^{\otimes m}\langle 1|^{\otimes n}\\ |0\rangle^{\otimes n}\langle 0|^{\otimes m}\cdots|0\rangle^{\otimes m}\langle 0|^{\otimes n}\end{matrix}
+\begin{matrix}|1\rangle^{\otimes n}\langle 1|^{\otimes m}\cdots|1\rangle^{\otimes m}\langle 1|^{\otimes n}\\ |0\rangle^{\otimes n}\langle 0|^{\otimes m}\cdots|0\rangle^{\otimes m}\langle 0|^{\otimes n}\end{matrix}\\
& +e^{-i\alpha}\begin{matrix}|0\rangle^{\otimes n}\langle 0|^{\otimes m}\cdots|0\rangle^{\otimes m}\langle 0|^{\otimes n}\\ |1\rangle^{\otimes n}\langle 1|^{\otimes m}\cdots|0\rangle^{\otimes m}\langle 0|^{\otimes n}\end{matrix}
+\begin{matrix}|1\rangle^{\otimes n}\langle 1|^{\otimes m}\cdots|0\rangle^{\otimes m}\langle 0|^{\otimes n}\\ |1\rangle^{\otimes n}\langle 1|^{\otimes m}\cdots|0\rangle^{\otimes m}\langle 0|^{\otimes n}\end{matrix}
+e^{-i2\alpha}\begin{matrix}|0\rangle^{\otimes n}\langle 0|^{\otimes m}\cdots|1\rangle^{\otimes m}\langle 1|^{\otimes n}\\ |1\rangle^{\otimes n}\langle 1|^{\otimes m}\cdots|0\rangle^{\otimes m}\langle 0|^{\otimes n}\end{matrix}
+e^{-i\alpha}\begin{matrix}|1\rangle^{\otimes n}\langle 1|^{\otimes m}\cdots|1\rangle^{\otimes m}\langle 1|^{\otimes n}\\ |1\rangle^{\otimes n}\langle 1|^{\otimes m}\cdots|0\rangle^{\otimes m}\langle 0|^{\otimes n}\end{matrix}\\
& +e^{i\alpha}\begin{matrix}|0\rangle^{\otimes n}\langle 0|^{\otimes m}\cdots|0\rangle^{\otimes m}\langle 0|^{\otimes n}\\ |0\rangle^{\otimes n}\langle 0|^{\otimes m}\cdots|1\rangle^{\otimes m}\langle 1|^{\otimes n}\end{matrix}
+e^{i2\alpha}\begin{matrix}|1\rangle^{\otimes n}\langle 1|^{\otimes m}\cdots|0\rangle^{\otimes m}\langle 0|^{\otimes n}\\ |0\rangle^{\otimes n}\langle 0|^{\otimes m}\cdots|1\rangle^{\otimes m}\langle 1|^{\otimes n}\end{matrix}
+\begin{matrix}|0\rangle^{\otimes n}\langle 0|^{\otimes m}\cdots|1\rangle^{\otimes m}\langle 1|^{\otimes n}\\ |0\rangle^{\otimes n}\langle 0|^{\otimes m}\cdots|1\rangle^{\otimes m}\langle 1|^{\otimes n}\end{matrix}
+e^{i\alpha}\begin{matrix}|1\rangle^{\otimes n}\langle 1|^{\otimes m}\cdots|1\rangle^{\otimes m}\langle 1|^{\otimes n}\\ |0\rangle^{\otimes n}\langle 0|^{\otimes m}\cdots|1\rangle^{\otimes m}\langle 1|^{\otimes n}\end{matrix}\\
& +\begin{matrix}|0\rangle^{\otimes n}\langle 0|^{\otimes m}\cdots|0\rangle^{\otimes m}\langle 0|^{\otimes n}\\ |1\rangle^{\otimes n}\langle 1|^{\otimes m}\cdots|1\rangle^{\otimes m}\langle 1|^{\otimes n}\end{matrix}
+e^{i\alpha}\begin{matrix}|1\rangle^{\otimes n}\langle 1|^{\otimes m}\cdots|0\rangle^{\otimes m}\langle 0|^{\otimes n}\\ |1\rangle^{\otimes n}\langle 1|^{\otimes m}\cdots|1\rangle^{\otimes m}\langle 1|^{\otimes n}\end{matrix}
+e^{-i\alpha}\begin{matrix}|0\rangle^{\otimes n}\langle 0|^{\otimes m}\cdots|1\rangle^{\otimes m}\langle 1|^{\otimes n}\\ |1\rangle^{\otimes n}\langle 1|^{\otimes m}\cdots|1\rangle^{\otimes m}\langle 1|^{\otimes n}\end{matrix}
+\begin{matrix}|1\rangle^{\otimes n}\langle 1|^{\otimes m}\cdots|1\rangle^{\otimes m}\langle 1|^{\otimes n}\\ |1\rangle^{\otimes n}\langle 1|^{\otimes m}\cdots|1\rangle^{\otimes m}\langle 1|^{\otimes n}\end{matrix}
\end{aligned}
$$

Since

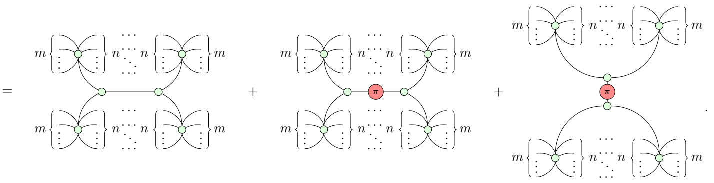
$$\int_0^{2\pi} e^{ik\alpha}\, d\alpha = 0, \quad k = \pm 1, \pm 2,$$

we integrate over α on each side and obtain

$$\tfrac{1}{2\pi}\int_\alpha$$



$$
\mathrm{d}\alpha = \left\{\begin{matrix}|0\rangle^{\otimes n}\langle 0|^{\otimes m}\cdots|0\rangle^{\otimes m}\langle 0|^{\otimes n}\\ |0\rangle^{\otimes n}\langle 0|^{\otimes m}\cdots|0\rangle^{\otimes m}\langle 0|^{\otimes n}\\ +\ |1\rangle^{\otimes n}\langle 1|^{\otimes m}\cdots|1\rangle^{\otimes m}\langle 1|^{\otimes n}\\ |1\rangle^{\otimes n}\langle 1|^{\otimes m}\cdots|1\rangle^{\otimes m}\langle 1|^{\otimes n}\end{matrix}\right\}
+\left\{\begin{matrix}|1\rangle^{\otimes n}\langle 1|^{\otimes m}\cdots|0\rangle^{\otimes m}\langle 0|^{\otimes n}\\ |1\rangle^{\otimes n}\langle 1|^{\otimes m}\cdots|0\rangle^{\otimes m}\langle 0|^{\otimes n}\\ +\ |0\rangle^{\otimes n}\langle 0|^{\otimes m}\cdots|1\rangle^{\otimes m}\langle 1|^{\otimes n}\\ |0\rangle^{\otimes n}\langle 0|^{\otimes m}\cdots|1\rangle^{\otimes m}\langle 1|^{\otimes n}\end{matrix}\right\}
+\left\{\begin{matrix}|0\rangle^{\otimes n}\langle 0|^{\otimes m}\cdots|0\rangle^{\otimes m}\langle 0|^{\otimes n}\\ |1\rangle^{\otimes n}\langle 1|^{\otimes m}\cdots|1\rangle^{\otimes m}\langle 1|^{\otimes n}\\ +\ |1\rangle^{\otimes n}\langle 1|^{\otimes m}\cdots|1\rangle^{\otimes m}\langle 1|^{\otimes n}\\ |0\rangle^{\otimes n}\langle 0|^{\otimes m}\cdots|0\rangle^{\otimes m}\langle 0|^{\otimes n}\end{matrix}\right\}
$$

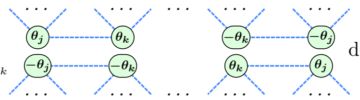


□

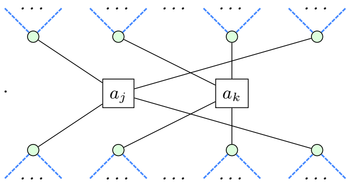
D Proof of lemma 3

Lemma 3. *The following equation holds.*

$$\frac{1}{(2\pi)^2}\int_{\theta_j}\int_{\theta_k}$$



$$\mathrm{d}\theta_j\mathrm{d}\theta_k = \textstyle\sum_{a_j,a_k\in\{T_1,T_2,T_3\}} \; M_{a_j,a_k}\cdot$$

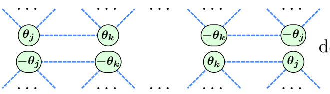


Here M_{a_j,a_k} is the 3×3 matrix

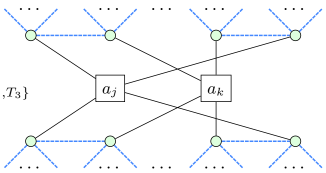
$$M = \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & 1 \end{pmatrix}$$

Proof. By lemma 2, we have

$$\frac{1}{(2\pi)^2}\int_{\theta_j}\int_{\theta_k}$$

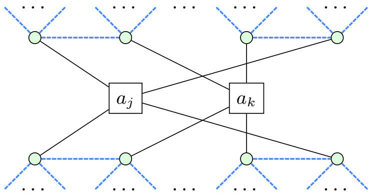


$$\mathrm{d}\theta_j\mathrm{d}\theta_k = \textstyle\sum_{a_j,a_k\in\{T_1,T_2,T_3\}}$$

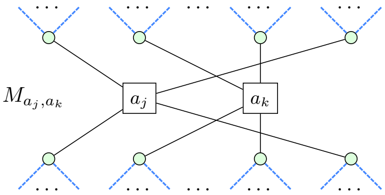


.

Hence, it is sufficient to prove

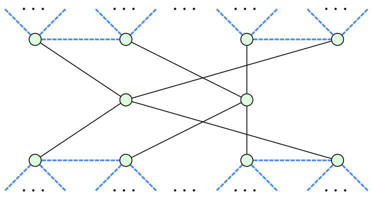


$$= M_{a_j,a_k}$$

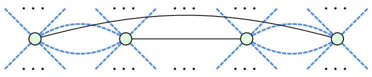


for $a_j, a_j \in \{T_1, T_2, T_3\}$.

For $a_j = a_k = T_1$, we have

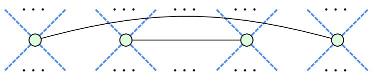


$$\overset{(\boldsymbol{f})}{=}$$

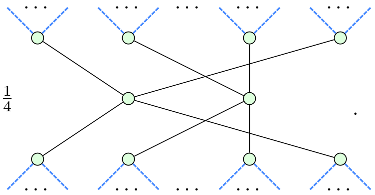


(29)

$$\overset{(\boldsymbol{hopf})}{=} \frac{1}{4}$$



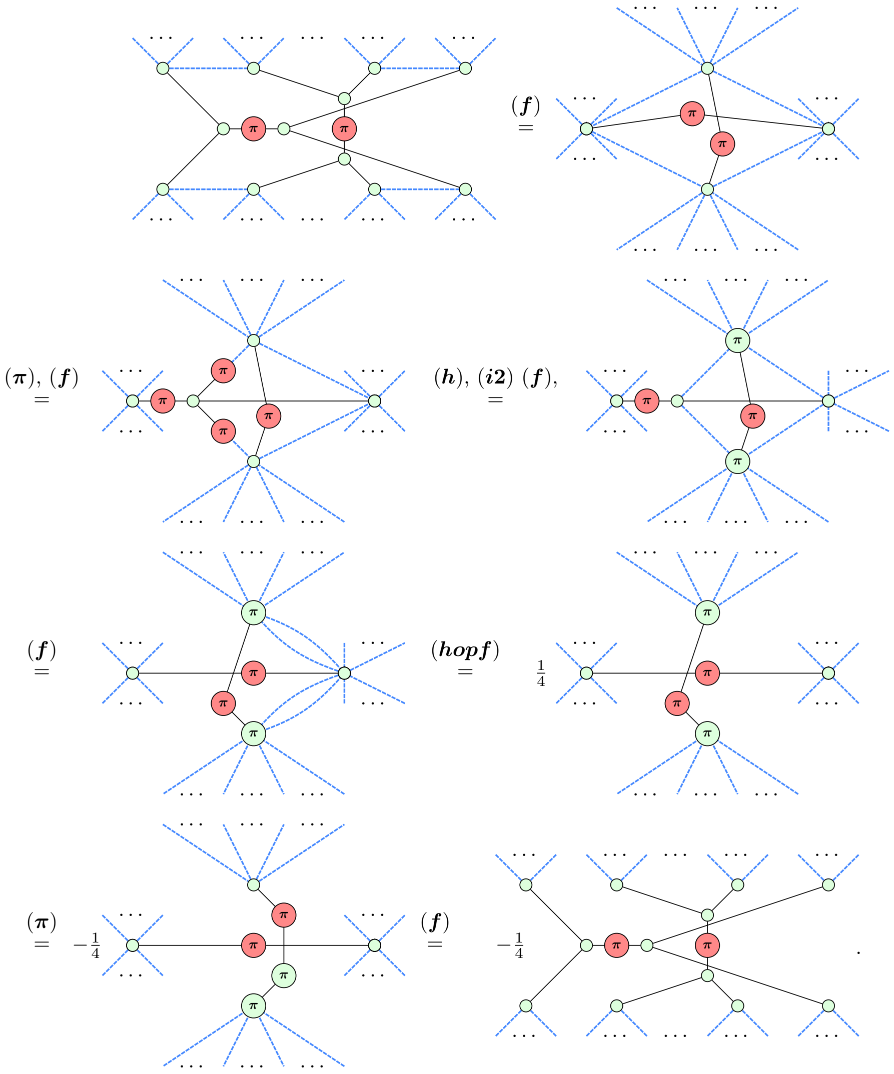
$$\overset{(\boldsymbol{f})}{=} \frac{1}{4}$$



.

And for $(a_j, a_k) \notin \{(T_2, T_3), (T_3, T_2)\}$, the proof is almost the same as that of (29).

Now, let us consider the case when $(a_j, a_k) = (T_2, T_3)$. We can use rules in figure 4 as follows.



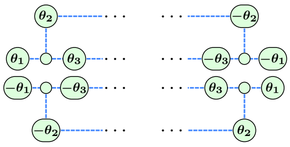
$\overset{(\boldsymbol{f})}{=}$
$\overset{(\boldsymbol{\pi}),\,(\boldsymbol{f})}{=}$
$\overset{(\boldsymbol{h}),\,(\boldsymbol{i2})\,(\boldsymbol{f}),}{=}$
$\overset{(\boldsymbol{f})}{=}$
$\overset{(\boldsymbol{hopf})}{=}$
$\frac{1}{4}$
$\overset{(\boldsymbol{\pi})}{=}$
$-\frac{1}{4}$
$\overset{(\boldsymbol{f})}{=}$
$-\frac{1}{4}$
π

□

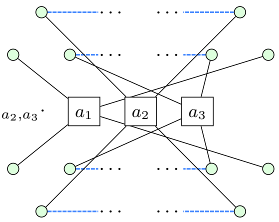
E Proof of lemma 4

Lemma 4. *The following equation holds.*

$$\frac{1}{(2\pi)^3}\int_{\theta_1,\theta_2,\theta_3}$$



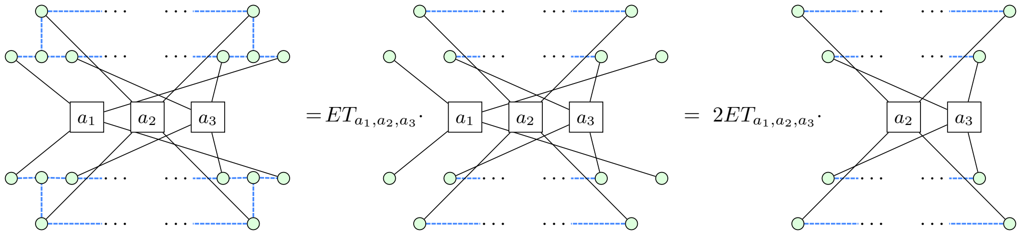
$$\mathrm{d}\theta_1\mathrm{d}\theta_2\mathrm{d}\theta_3 \;=\; \sum_{a_1,a_2,a_3\in\{T_1,T_2,T_3\}} ET_{a_1,a_2,a_3}\cdot$$



Here ET_{a_1,a_2,a_3} *can be regarded as a* $3\times3\times3$ *tensor*

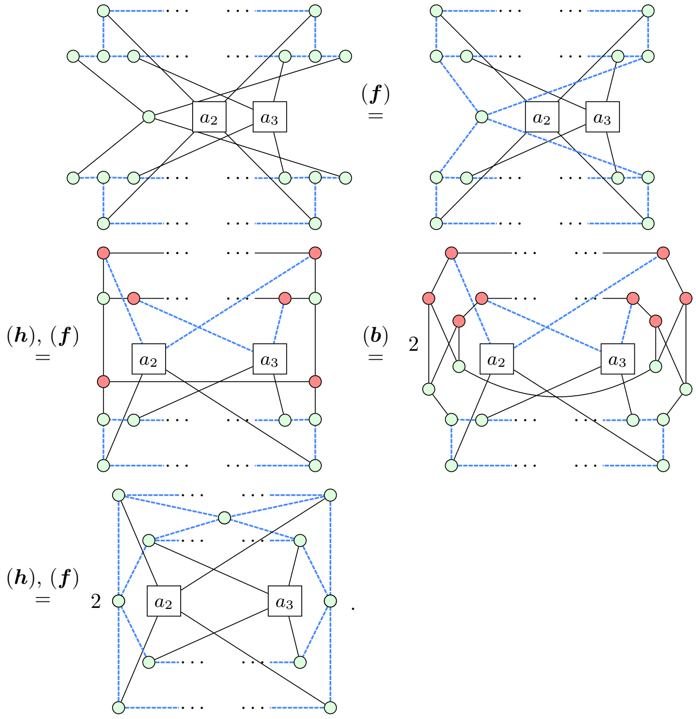
$$ET[1,\cdot,\cdot] = \frac{1}{8}\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}, \quad ET[2,\cdot,\cdot] = \frac{1}{8}\begin{pmatrix}0&1&0\\1&0&0\\0&0&0\end{pmatrix}, \quad ET[3,\cdot,\cdot] = \frac{1}{8}\begin{pmatrix}0&0&1\\0&0&0\\1&0&0\end{pmatrix}.$$

Proof. By lemma 2, it is sufficient to prove that

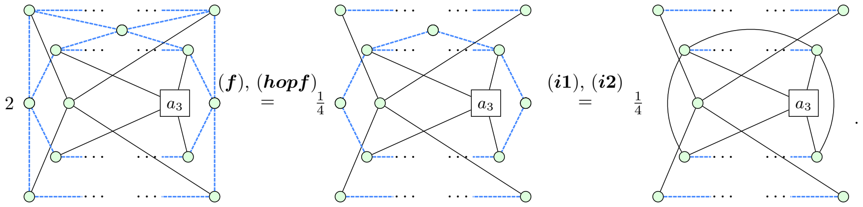


for $a_1, a_2, a_3 \in \{T_1, T_2, T_3\}$.

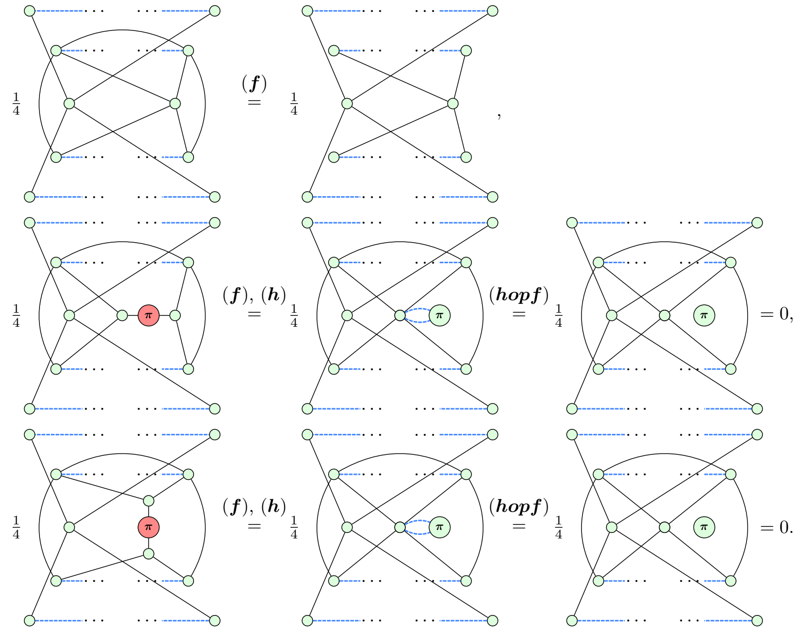
We first consider the case when $a_1 = T_1$. Since $a_1 = T_1$, we have


a_2
a_3
$\overset{(\boldsymbol{f})}{=}$
a_2
a_3
$\overset{(\boldsymbol{h}),\,(\boldsymbol{f})}{=}$
a_2
a_3
$\overset{(\boldsymbol{b})}{=}$
2
a_2
a_3
$\overset{(\boldsymbol{h}),\,(\boldsymbol{f})}{=}$
2
a_2
a_3
.

Now, if $a_2 = T_1$, then we have

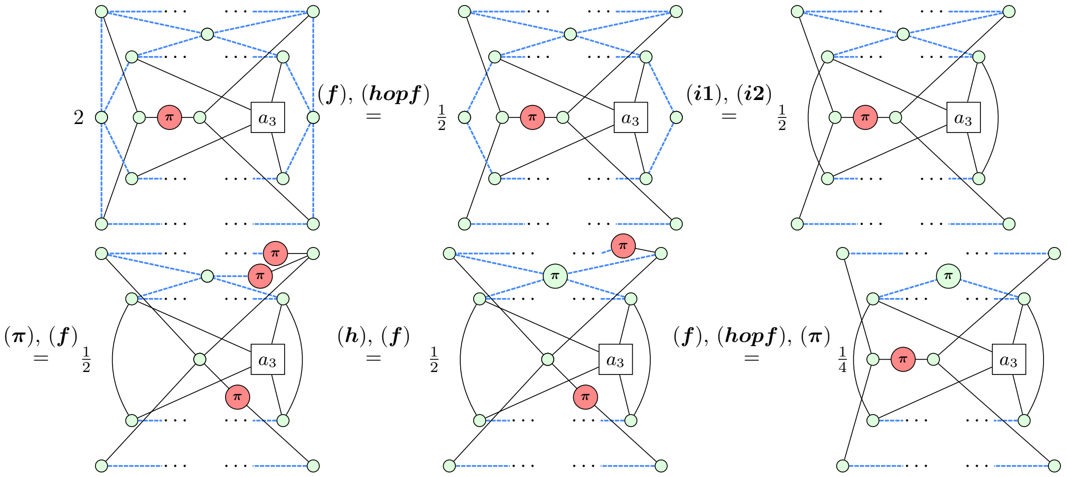

2
a_3
$\overset{(\boldsymbol{f}),\,(\boldsymbol{hopf})}{=}$
$\frac{1}{4}$
a_3
$\overset{(\boldsymbol{i1}),\,(\boldsymbol{i2})}{=}$
$\frac{1}{4}$
a_3
.

Hence,

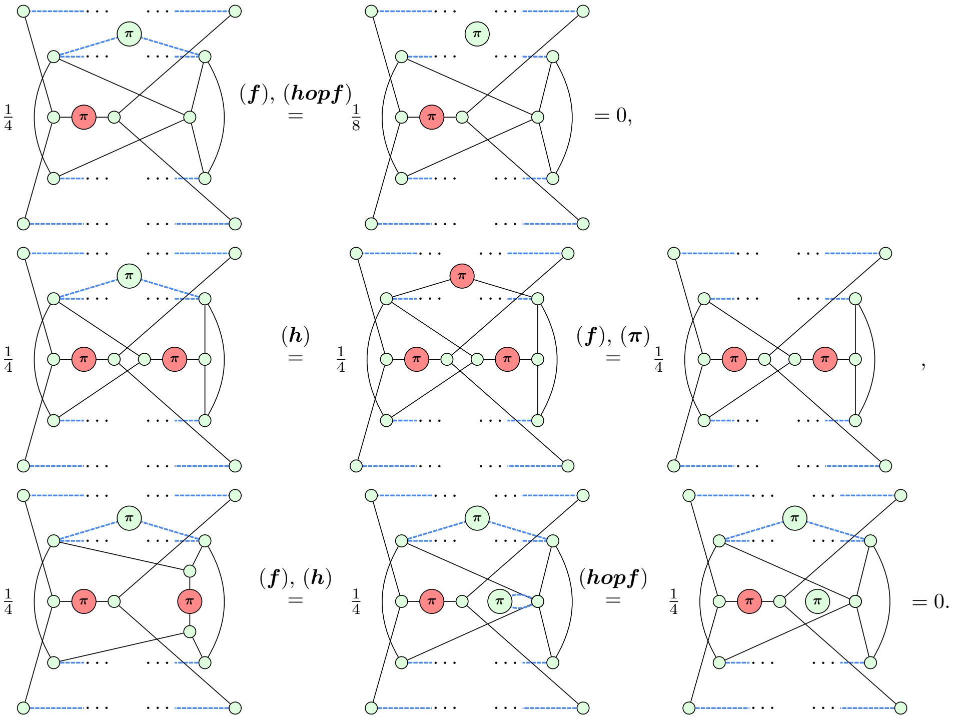


That is $ET[1, 1, \cdot] = \frac{1}{8}\begin{pmatrix}1 & 0 & 0\end{pmatrix}$.

If $a_2 = T_2$, then

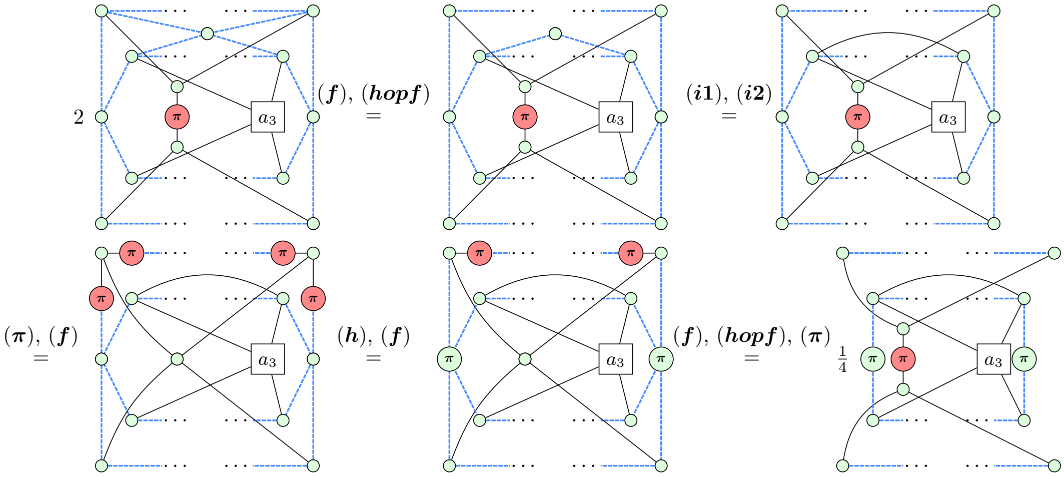


Hence,

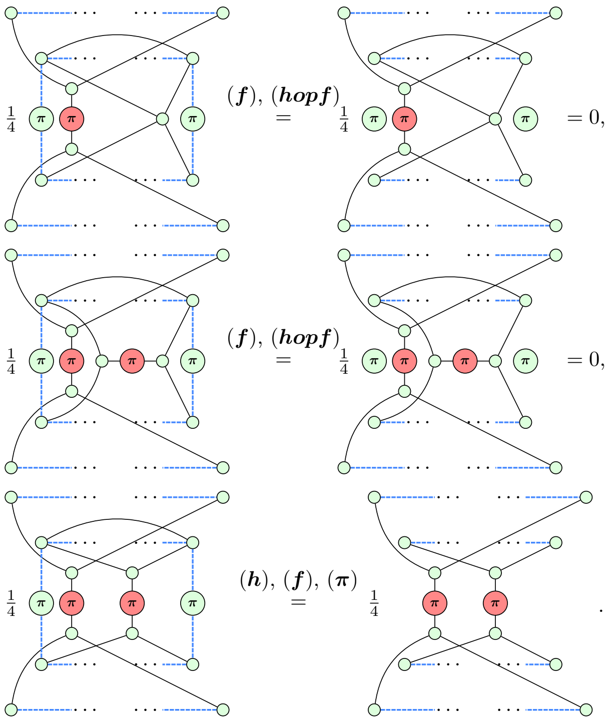


That is $ET[1,2,\cdot] = \frac{1}{8}\begin{pmatrix}0 & 1 & 0\end{pmatrix}$.

If $a_2 = T_3$, then



Hence.



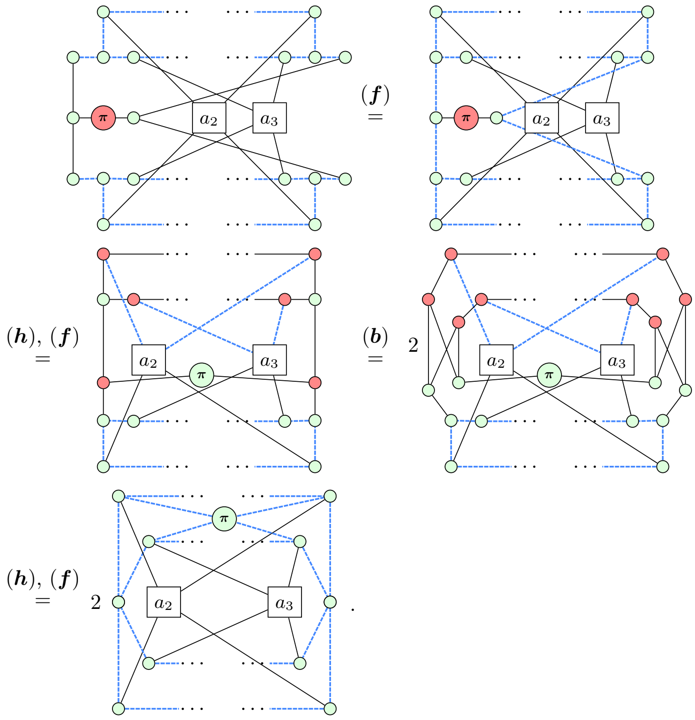
$\frac{1}{4}$
π
$(\boldsymbol{f})$, $(\boldsymbol{hopf})$
$=$
$\frac{1}{4}$
$= 0,$
$\frac{1}{4}$
$(\boldsymbol{f})$, $(\boldsymbol{hopf})$
$=$
$\frac{1}{4}$
$= 0,$
$\frac{1}{4}$
$(\boldsymbol{h})$, $(\boldsymbol{f})$, $(\boldsymbol{\pi})$
$=$
$\frac{1}{4}$
.

That is $ET[1,3,\cdot] = \frac{1}{8}\begin{pmatrix}0 & 0 & 1\end{pmatrix}$.

By now, we have proved that

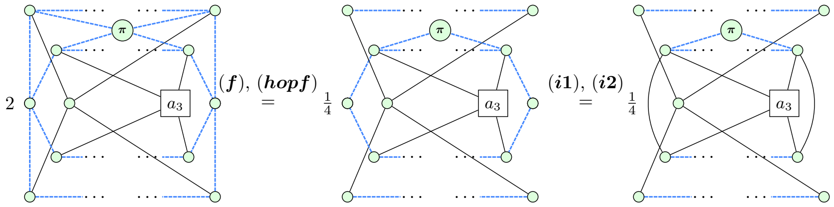
$$ET[1,\cdot,\cdot] = \frac{1}{8}\begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix}. \tag{30}$$

Now, let us consider the case when $a_1 = T_2$. Since $a_1 = T_2$, we have

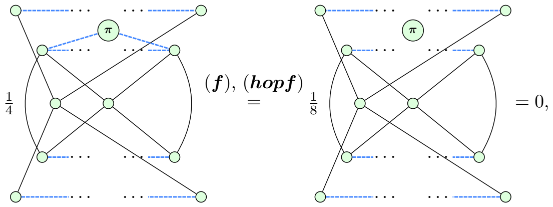


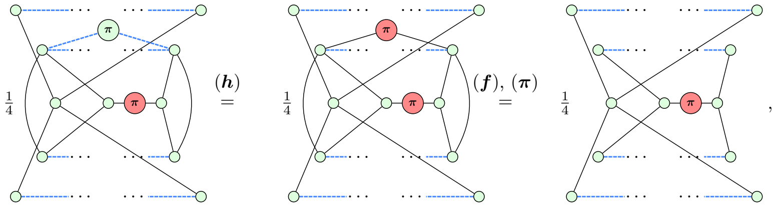
π
a_2
a_3
$(\boldsymbol{f})$
$=$
$(\boldsymbol{h})$, $(\boldsymbol{f})$
$=$
$(\boldsymbol{b})$
$=$
2
$(\boldsymbol{h})$, $(\boldsymbol{f})$
$=$
2
.

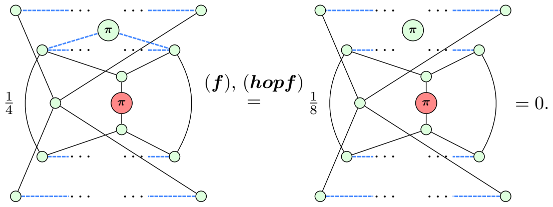
Now, if $a_2 = T_1$, then we have



Hence,

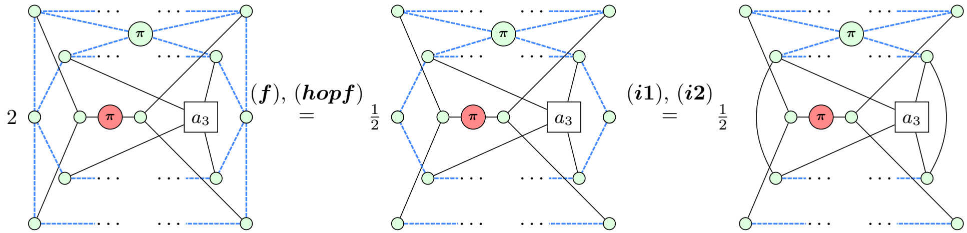


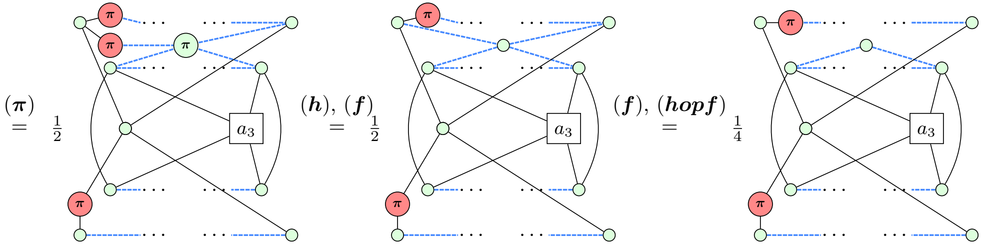


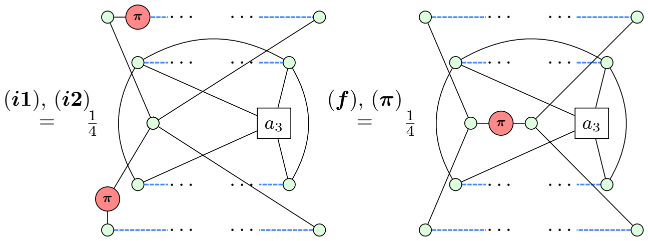


That is $ET[1,1,\cdot] = \frac{1}{8}\begin{pmatrix}0 & 1 & 0\end{pmatrix}$.

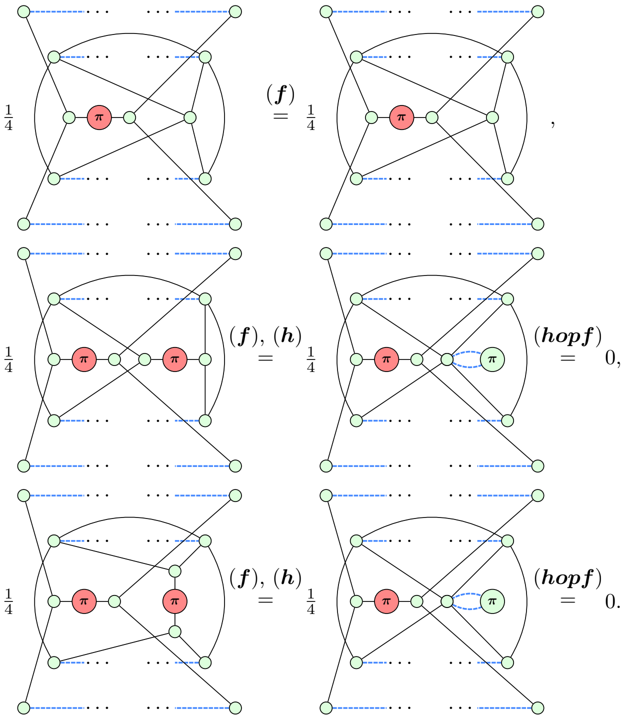
If $a_2 = T_2$, then





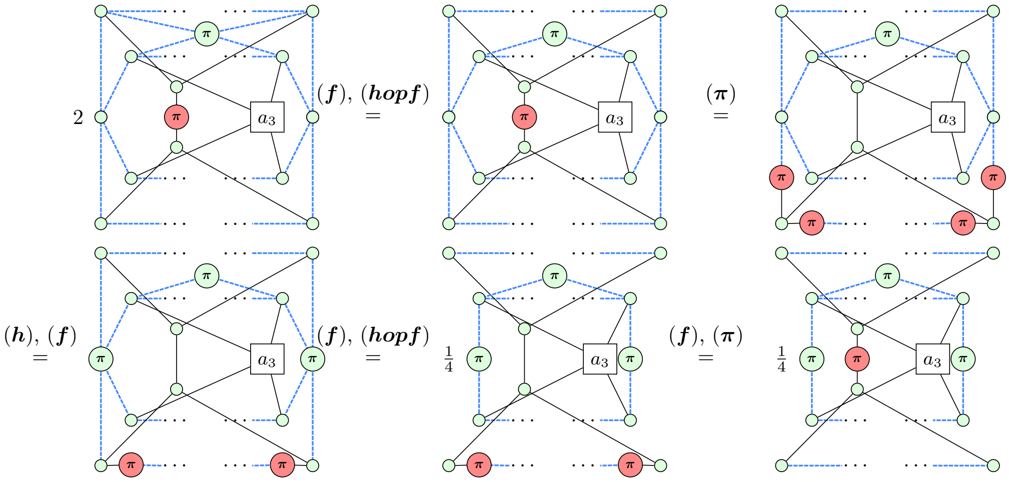


Hence,

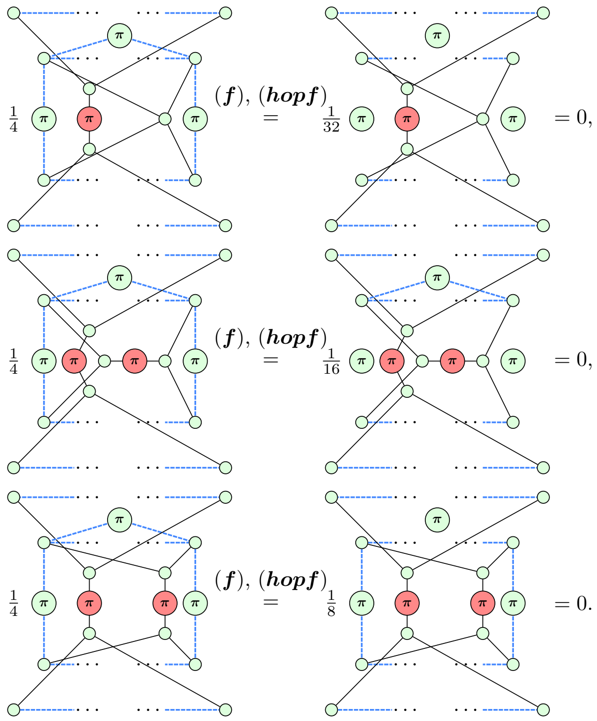


That is $ET[1,2,\cdot] = \frac{1}{8}\begin{pmatrix}1 & 0 & 0\end{pmatrix}$.

If $a_2 = T_3$, then



Hence.


$\frac{1}{4}$
π
$(\boldsymbol{f})$, $(\boldsymbol{hopf})$
$=$
$\frac{1}{32}$
$= 0,$
$\frac{1}{4}$
$(\boldsymbol{f})$, $(\boldsymbol{hopf})$
$=$
$\frac{1}{16}$
$= 0,$
$\frac{1}{4}$
$(\boldsymbol{f})$, $(\boldsymbol{hopf})$
$=$
$\frac{1}{8}$
$= 0.$

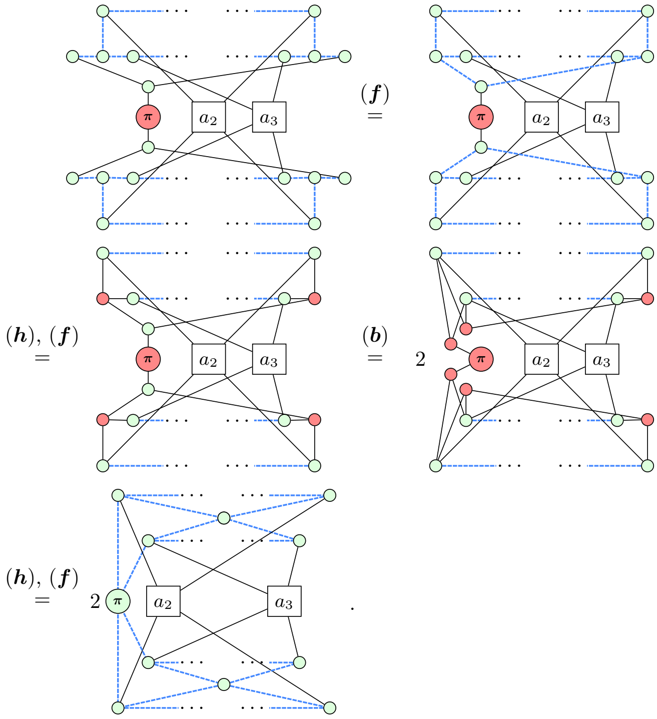
That is $ET[1,3,\cdot] = \frac{1}{8}\begin{pmatrix}0 & 0 & 0\end{pmatrix}$.

By now, we have proved that

$$ET[2,\cdot,\cdot] = \frac{1}{8}\begin{pmatrix}0 & 1 & 0\\ 1 & 0 & 0\\ 0 & 0 & 0\end{pmatrix}. \tag{31}$$

Let us consider the case when $a_1 = T_3$.

When $a_1 = T_3$, we have


π
a_2
a_3
$(\boldsymbol{f})$
$=$
$(\boldsymbol{h})$, $(\boldsymbol{f})$
$=$
$(\boldsymbol{b})$
$=$
2
$(\boldsymbol{h})$, $(\boldsymbol{f})$
$=$
2
.

According to the symmetry, we can use the result of the case when $a_1 = T_2$. And we get

$$ET[3,\cdot,\cdot] = \frac{1}{8}\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}. \tag{32}$$

□

F Analysis of PQCs in section 4

F.1 Hardware-efficient ansatz

We will prove some property of LT which are used in the analysis in 4.1.

Theorem 8. *Suppose that $\lambda_1, \ldots, \lambda_{3^n}$ are eigenvalues of LT. And*

$$|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_{3^n}|.$$

Then we have

$$\lambda_1 = \lambda_2 = 1, \quad |\lambda_j| < 1 \textit{ for } j > 2.$$

Proof. By definition of EM, we can compute that

$$EM = \begin{pmatrix} \frac{3}{4} & \frac{1}{4} & \frac{1}{4} & & & & & & \\ \frac{1}{4} & \frac{3}{4} & -\frac{1}{4} & & & & & & \\ \frac{1}{4} & -\frac{1}{4} & \frac{3}{4} & & & & & & \\ & & & \frac{1}{4} & \frac{3}{4} & 0 & & & \\ & & & \frac{3}{4} & \frac{1}{4} & 0 & & & \\ & & & -\frac{1}{4} & \frac{1}{4} & 0 & & & \\ & & & & & & \frac{1}{4} & 0 & \frac{3}{4} \\ & & & & & & -\frac{1}{4} & 0 & \frac{1}{4} \\ & & & & & & \frac{3}{4} & 0 & \frac{1}{4} \end{pmatrix}. \tag{33}$$

By computation, EM can be diagonalized. Four of its eigenvalues are 1 and other eigenvalues are in the interval $(-1, 1)$. Moreover, the eigenspace of the eigenvalue 1 is

$$\operatorname{span}\{v_1 \otimes v_{1,2}, v_1 \otimes v_{1,3}, v_{1,2} \otimes v_{1,2}, v_{1,3} \otimes v_{1,3}\}, \tag{34}$$

where

$$v_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, v_{1,2} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, v_{1,3} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}.$$

LT is an operator on the tensor product of n of $\mathbb{R}^3$. We denote the operator EM on the i-th and j-th $\mathbb{R}^3$ as $EM_{i,j}$. Then the eigenspace of LT corresponding to the eigenvalue 1 is the intersection of the eigenspaces corresponding to the eigenvalue 1 of

$$EM_{1,2},\ EM_{2,3},\ \ldots, EM_{n-1,n},\ EM_{n,1}.$$

Hence, by eq. (34), we have

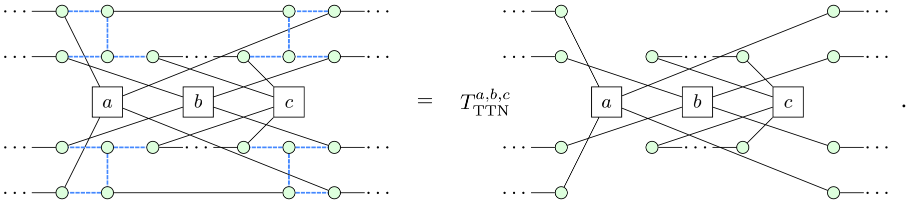
$$E_1 = \operatorname{span}\{v_{1,2} \otimes \cdots \otimes v_{1,2}, v_{1,3} \otimes \cdots \otimes v_{1,3}\}.$$

□

F.2 Tree tensor network ansatz

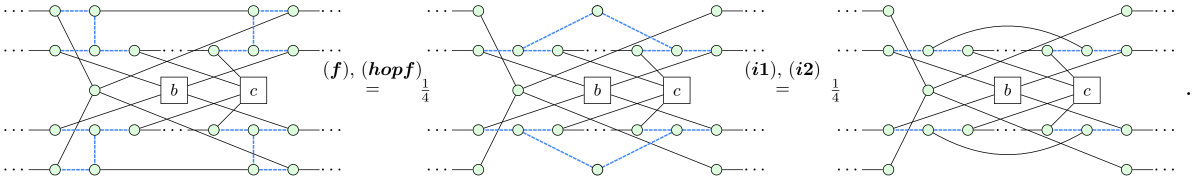
In section 4.2, we used (21). Here we will prove this equation.

Proof of (21). By lemma 2, we only need to prove that

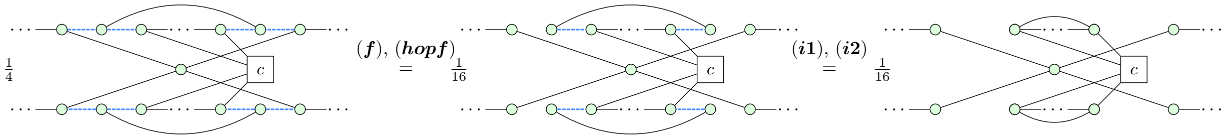


Similar to the proof of lemma 4, we first consider the case when $a = T_1$.

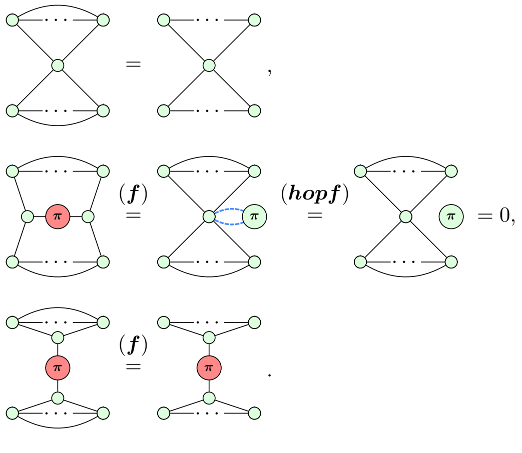
When $a = T_1$, we have


b
c
$(\boldsymbol{f})$, $(\boldsymbol{hopf})$ $=$ $\frac{1}{4}$
b
c
$(\boldsymbol{i1})$, $(\boldsymbol{i2})$ $=$ $\frac{1}{4}$
b
c
.

Now, a is disconnected with b and c. Hence, we can consider b and c individually. If $b = T_1$, then


$\frac{1}{4}$
c
$(\boldsymbol{f})$, $(\boldsymbol{hopf})$ $=$ $\frac{1}{16}$
c
$(\boldsymbol{i1})$, $(\boldsymbol{i2})$ $=$ $\frac{1}{16}$
c

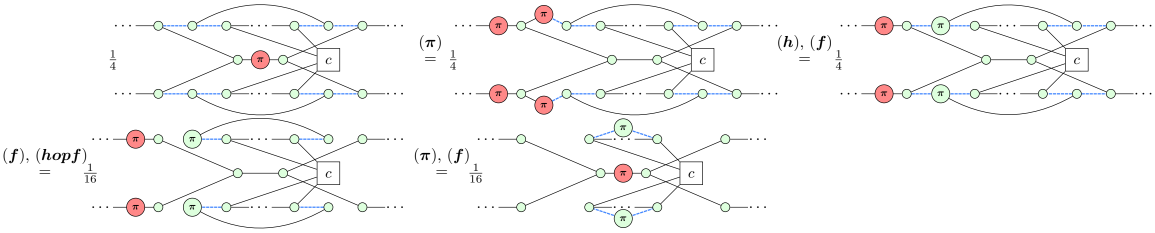
Hence,



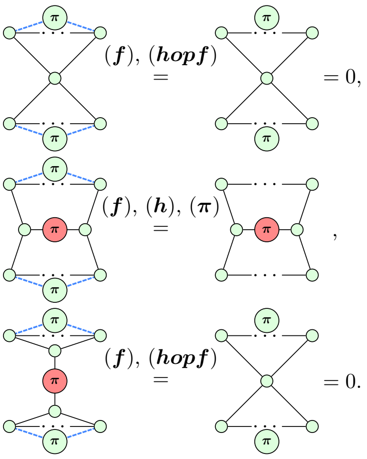
That it

$$T_{\mathrm{TTN}}[1,1,\cdot] = \frac{1}{16}\begin{pmatrix}1 & 0 & 1\end{pmatrix}.$$

If $b = T_2$, then


$\frac{1}{4}$
π
c
$(\boldsymbol{\pi})$ $=$ $\frac{1}{4}$
π
π
π
π
c
$(\boldsymbol{h})$, $(\boldsymbol{f})$ $=$ $\frac{1}{4}$
π
π
π
π
c
$(\boldsymbol{f})$, $(\boldsymbol{hopf})$ $=$ $\frac{1}{16}$
π
π
π
π
c
$(\boldsymbol{\pi})$, $(\boldsymbol{f})$ $=$ $\frac{1}{16}$
π
π
π
c

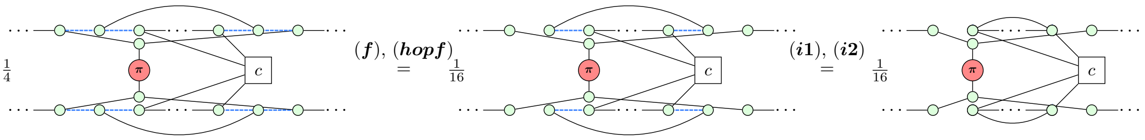
Hence,



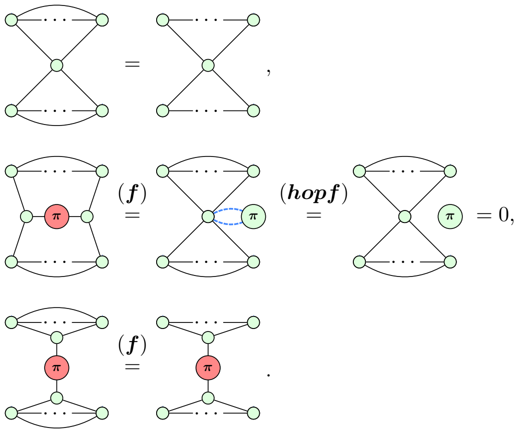
That it

$$T_{\mathrm{TTN}}[1,2,\cdot] = \frac{1}{16}\begin{pmatrix}0 & 1 & 0\end{pmatrix}.$$

If $b = T_3$, then



Hence,



That it

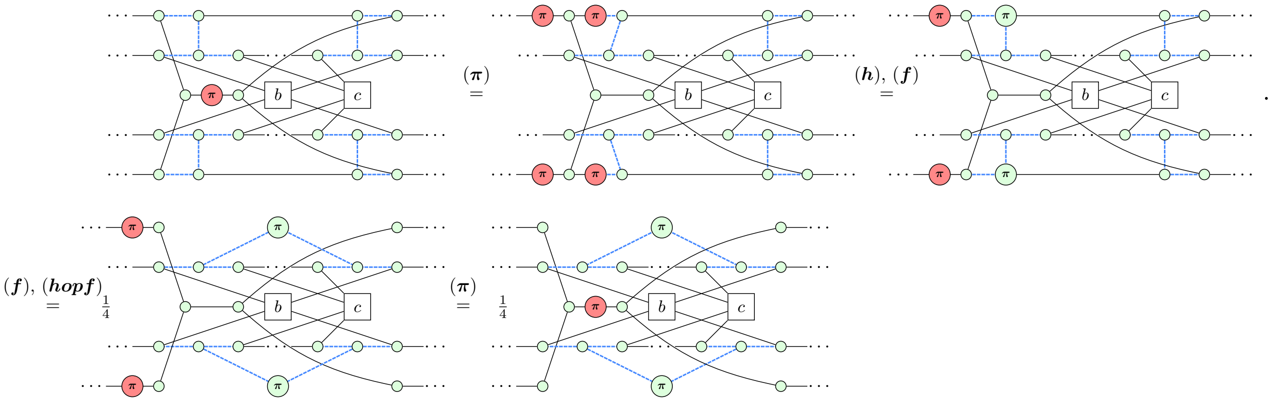
$$T_{\mathrm{TTN}}[1,3,\cdot] = \frac{1}{16}\begin{pmatrix}1 & 0 & 1\end{pmatrix}.$$

By now, we have proved that

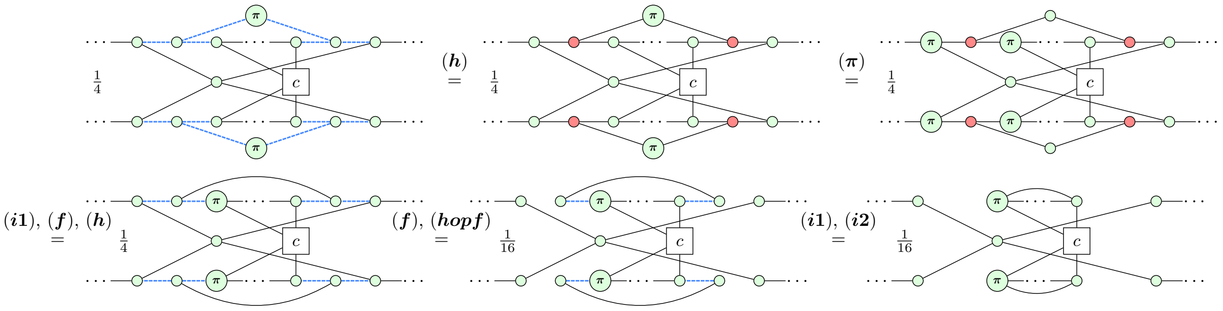
$$T_{\mathrm{TTN}}[1,\cdot,\cdot] = \frac{1}{16}\begin{pmatrix}1 & 0 & 1\\ 0 & 1 & 0\\ 1 & 0 & 1\end{pmatrix}. \tag{35}$$

Now, let us consider the case when $a = T_2$.

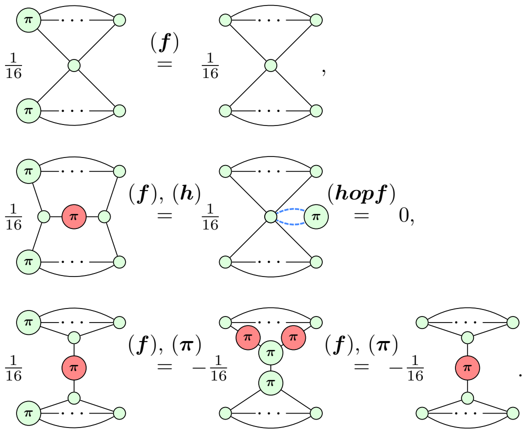
When $a = T_2$, we have



Now, a is disconnected with b and c. Hence, we can consider b and c individually. If $b = T_1$, then



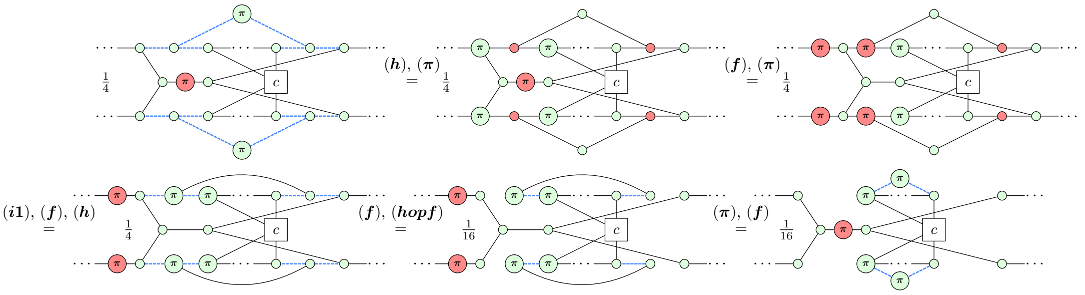
Hence,



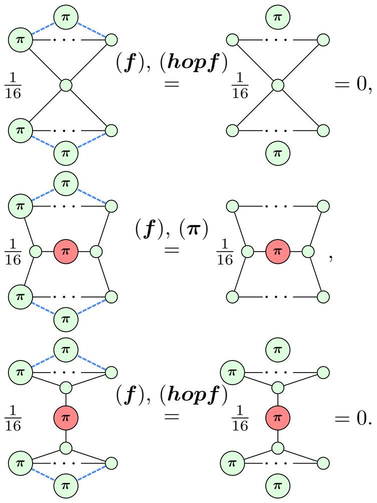
That it

$$T_{\mathrm{TTN}}[2,1,\cdot] = \frac{1}{16}\begin{pmatrix}1 & 0 & -1\end{pmatrix}.$$

If $b = T_2$, then



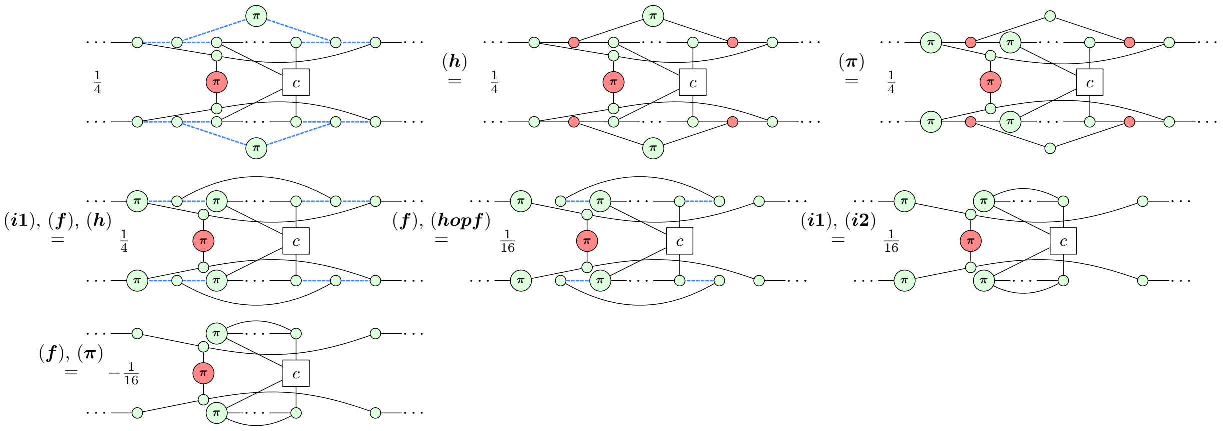
Hence,



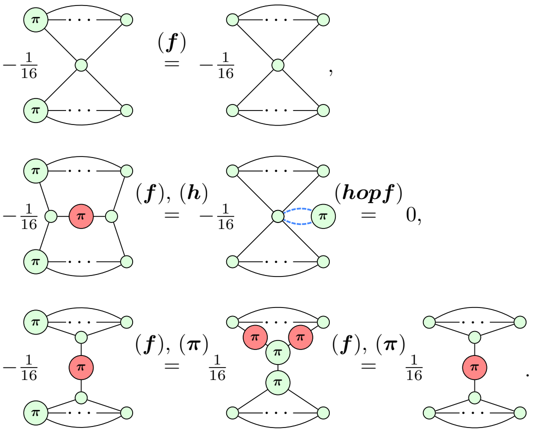
That it

$$T_{\text{TTN}}[2, 2, \cdot] = \frac{1}{16} \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} .ow$$

If $b = T_3$, then



Hence,



That it

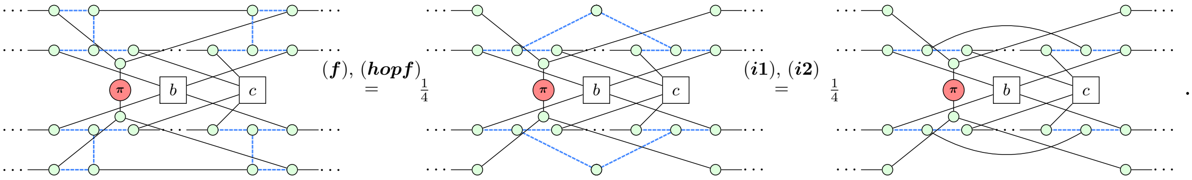
$$T_{\text{TTN}}[2, 3, \cdot] = \frac{1}{16} \begin{pmatrix} -1 & 0 & 1 \end{pmatrix} .$$

By now, we have proved that

$$T_{\text{TTN}}[2, \cdot, \cdot] = \frac{1}{16} \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} . \tag{36}$$

Now, let us consider the case when $a = T_3$.

When $a = T_3$, we have



Now, a is disconnected with b and c. And the part of b and c is same as the case when $a = T_1$. Hence, we have

$$T_{\mathrm{TTN}}[3, \cdot, \cdot] = \frac{1}{16} \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}. \tag{37}$$

□

Now we are going to prove that there is a lower bound for the variance of gradients in the tree tensor network ansatz.

Theorem 6. *For tree tensor network ansatz shown in section 4.2, if*

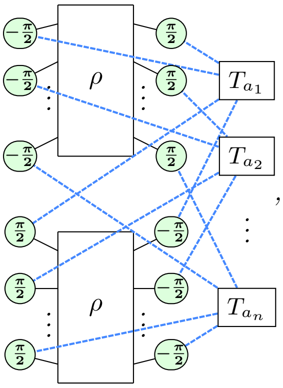
$$H = k_0 I + k_1 X + k_2 Y + k_3 Z,$$

then we have

$$\mathbf{Var}\left(\frac{\partial \langle H \rangle}{\partial \theta} \right) \geq \frac{k_1^2 + k_3^2}{n^2} \tilde{I}(u_1, \ldots, u_n) + \frac{k_2^2}{n^2} \tilde{I}(w_1, \ldots, w_n), \tag{25}$$

for some $u_j, w_j \in \{v_{1,3}, v_2, v_{1,3}^-\}$. Here $\tilde{I}(u_1, \ldots, u_n)$ is defined in (24). And $\tilde{I}$ is a 3^n dimensional tensor which only depends on the input state. If the input state is ρ, then $\tilde{I}$ is defined as follows.

$\tilde{I}^{a_1, \ldots, a_n} =$



, where $a_j \in \{1, 2, 3\}$.

Proof. We first prove that

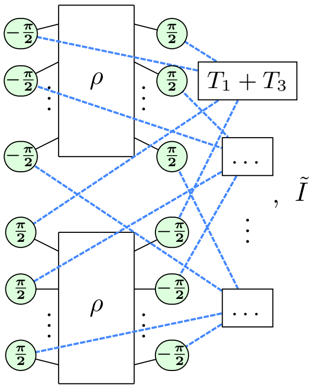
$$\tilde{I}(u_1, \ldots, u_n) \geq 0, \quad u_j \in \left\{ v_2, v_{1,3}, v_{1,3}^- \right\},$$

where

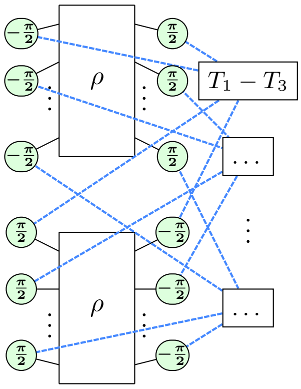
$$v_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad v_{1,3} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \quad v_{1,3}^- = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}.$$

Note that if u_1 is $v_{1,3}$ or $v_{1,3}^-$, then

$\tilde{I}(v_{1,3},\dots) =$



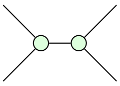
, $\tilde{I}(v_{1,3}^-,\dots) =$



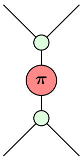
.

By

$T_1 + T_3 =$

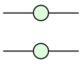


$+$



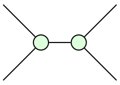
$= |00\rangle\langle 00| + |10\rangle\langle 10| + |01\rangle\langle 01| + |11\rangle\langle 11|$,

$= (|0\rangle\langle 0| + |1\rangle\langle 1|) \otimes (|0\rangle\langle 0| + |1\rangle\langle 1|) =$

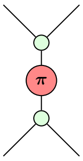


,

$T_1 - T_3 =$

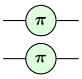


$+$



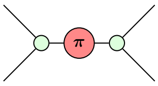
$= |00\rangle\langle 00| - |10\rangle\langle 10| - |01\rangle\langle 01| + |11\rangle\langle 11|$

$= (|0\rangle\langle 0| - |1\rangle\langle 1|) \otimes (|0\rangle\langle 0| - |1\rangle\langle 1|) =$



,

$T_2 =$



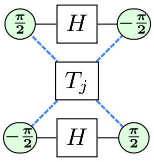
$$= \begin{matrix} -\boxed{|0\rangle\langle 1|}- \\ -\boxed{|0\rangle\langle 1|}- \end{matrix} + \begin{matrix} -\boxed{|1\rangle\langle 0|}- \\ -\boxed{|1\rangle\langle 0|}- \end{matrix} ,$$

we can expand $\tilde{I}(u_1,\dots,u_n)$ as a sum of squares. Thus, we have proved that

$$\tilde{I}(u_1,\dots,u_n) \geq 0.$$

Now let us consider the lower bound of $\mathbf{Var}\left(\frac{\partial\langle H\rangle}{\partial\theta}\right)$. By the graph-like ZX-diagram we get, we have $\tilde{H} = \begin{pmatrix} \tilde{H}_1 \\ \tilde{H}_2 \\ \tilde{H}_3 \end{pmatrix}$ which is defined as follows.

$\tilde{H}_j =$



, for $j = 1, 2, 3$.

Suppose that $H = k_0 I + k_1 X + k_2 Y + k_3 Z$. Then we can get

$$\tilde{H} = 2k_0^2 v_{1,3} + 2(k_1^2 + k_3^2)v_2 + 2k_2^2 v_{1,3}^-. \tag{38}$$

And by the definition of T_{TTN}, we can get (23). Hence, we can expand the variance as

$$\mathbf{Var}\left(\frac{\partial\langle H\rangle}{\partial\theta}\right) = \sum_{u_j \in \{v_2, v_{1,3}, v_{1,3}^-\}} a(u_1,\dots,u_n) \cdot \tilde{I}(u_1,\dots,u_n),$$

where $a(u_1,\dots,u_n)$ is a non-negative number. We will prove that there exists one choise of $(u_1,\dots,u_n)$ such that

$$a(u_1,\dots,u_n)\in\Omega(\frac{1}{\mathrm{poly}(n)}).$$

When we use (23) to expand the variance, the only case that will cause a coefficient <1 is the case when we use the second equation in (23). Hence, the coefficients $a(u_1,\dots,u_n)$ only depend on the number of times that we use the second equation. And it depends on the location of θ. Note that the worst case is that θ is the parameter of the first gate applying to the first qubit. In this case, we need to use the second equation in (23) for $2\log(n)-1$ times. That means we have a lower bound for the variance

$$\mathbf{Var}\left(\frac{\partial\langle H\rangle}{\partial\theta}\right)\geq\frac{k_1^2+k_3^2}{n^2}\tilde{I}(u_1,\dots,u_n)+\frac{k_2^2}{n^2}\tilde{I}(w_1,\dots,w_n),$$

for some $u_j,w_j\in\{v_{1,3},v_2,v_{\bar{1},3}\}$.

Note that $I(u_1,\dots,u_n)$ only depends on the input state ρ. If given that

$$\tilde{I}(u_1,\dots,u_n)\in\Omega(\frac{1}{\mathrm{poly}(n)})\text{ or }\tilde{I}(w_1,\dots,w_n)\in\Omega(\frac{1}{\mathrm{poly}(n)}),$$

then there exist no barren plateau in the tree tensor network ansatz.

□

F.3 QCNN

Theorem 7. *For the QCNN ansatz shown in section 4.3, if*

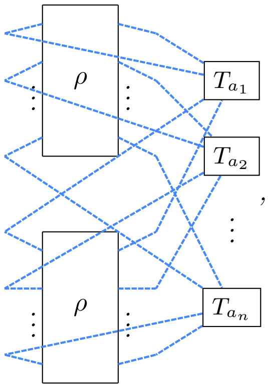
$$H=k_0I+k_1X+k_2Y+k_3Z,$$

then we have

$$\mathbf{Var}\left(\frac{\partial\langle H\rangle}{\partial\theta}\right)\geq\frac{k_2^2+k_3^2}{n^9}\tilde{I}(u_1,\dots,u_n)+\frac{k_1^2}{n^9}\tilde{I}(w_1,\dots,w_n),\tag{27}$$

for some $u_j,w_j\in\{v_{1,3},v_2,v_{\bar{1},3}\}$. Here $\tilde{I}$ is a 3^n dimensional tensor which only depends on the input state ρ.

$\tilde{I}^{a_1,\dots,a_n}$ =



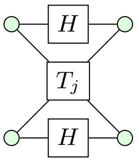
, *where $a_j\in\{1,2,3\}$.*

Proof. The proof is similar to that of the tree tensor network ansatz (Theorem 6), so we will only give a sketch of the proof. We also have

$$\tilde{I}(u_1,\dots,u_n)\geq 0,\quad u_j\in\{v_{1,3},v_2,v_{\bar{1},3}\}.$$

From the graph-like ZX-diagram, we have

$\tilde{H}_j$ =



, for $j=1,2,3$.

Hence,

$$\tilde{H} = 2k_0^2 v_{1,3} + 2(k_1^2 + k_2^2)v_2 + 2k_3^2 v_{1,3}^-.$$

We will analyze each term of $\tilde{H}$ in the variance.

Using (26), the term $2k_0^2 v_{1,3}$ will become 0 by

$$P_2 v_{1,3} = 0.$$

The terms $2(k_1^2 + k_2^2)v_2$ and $2k_3^2 v_{1,3}^-$ will generate terms containing v_2 after expanding using (26). And each time after generating terms containing v_2, a coefficient $\geq \frac{1}{8}$ will be multiplied to the variance. Hence, if we want to bring v_2 to P_2, we need to generate terms containing v_2 for l times, where l is a path from the location of v_2 to the location of P_2.

By the structure of the QCNN ansatz, we have

$$l \geq 3\log(n).$$

It will generated a coefficient $\geq \frac{1}{8^{3\log(n)}}$.

Hence we have

$$\textbf{Var}\left(\frac{\partial \langle H \rangle}{\partial \theta}\right) \geq \frac{k_2^2 + k_3^2}{n^9}\tilde{I}(u_1, \dots, u_n) + \frac{k_1^2}{n^9}\tilde{I}(w_1, \dots, w_n),$$

for some $u_j, w_j \in \{v_{1,3}, v_2, v_{1,3}^-\}$. □