
Robust Integrated Learning and Pauli Noise Mitigation for Parametrized Quantum Circuits

Md Mobasshir Arshed Naved

Department of Computer Science
Purdue University
West Lafayette, USA
naved@purdue.edu

Wenbo Xie

Department of Computer Science
Purdue University
West Lafayette, USA
xie401@purdue.edu

Wojciech Szpankowski

Department of Computer Science
Purdue University
West Lafayette, USA
szpan@purdue.edu

Ananth Grama

Department of Computer Science
Purdue University
West Lafayette, USA
ayg@cs.purdue.edu

Abstract

We propose a novel gradient-based framework for learning parameterized quantum circuits (PQCs) in the presence of Pauli noise in gate operation. The key innovation in our framework is the simultaneous optimization of model parameters and learning of an inverse noise channel, specifically designed to mitigate Pauli noise. Our parametrized inverse noise model utilizes the Pauli-Lindblad equation and relies on the principle underlying the Probabilistic Error Cancellation (PEC) protocol to learn an effective and scalable mechanism for noise mitigation. In contrast to conventional approaches that apply predetermined inverse noise models during execution, our method systematically mitigates Pauli noise by dynamically updating the inverse noise parameters in conjunction with the model parameters, facilitating task-specific noise adaptation throughout the learning process. We employ proximal stochastic gradient descent (proximal SGD) to ensure that updates are bounded within a feasible range to ensure stability. This approach allows the model to converge efficiently to a stationary point, balancing the trade-off between noise mitigation and computational overhead, resulting in a highly adaptable quantum model that performs robustly in noisy quantum environments.

1 Introduction

Quantum systems offer the potential to solve computationally hard problems in simulation, optimization, and machine learning. However, they pose significant challenges on NISQ devices [22]. The presence of quantum noise, which leads to errors in state preparation, gate operation, and measurement [12, 20], severely constrains the performance of variational quantum algorithms (VQAs) and quantum machine learning (QML) tasks. A major contributor to quantum noise in NISQ devices is gate noise, commonly modeled as Pauli noise [29]. Pauli noise occurs when unintended Pauli operators (X , Y , Z) are applied to quantum states during gate operations. These disruptions degrade the fidelity of quantum circuits and, in turn, the performance of parameterized quantum circuits (PQCs) commonly used in optimization and machine learning. Effective noise mitigation techniques are necessary to realize the potential of quantum computations on NISQ devices.

Pauli noise mitigation strategies, such as zero noise extrapolation (ZNE) and probabilistic error cancellation (PEC), have several practical drawbacks. ZNE mitigates Pauli noise by purposely

amplifying it, either by stretching gate durations or by adding noisy operations [11]. It then estimates zero-noise results through carefully designed cancellation strategies. This approach relies on precise calibration based on the hardware’s noise characteristics. To find the zero-noise result, ZNE runs circuits repeatedly at different noise levels, which introduces overhead for complex circuits or highly stochastic noise profiles. PEC utilizes detailed noise characterization, often by full or partial tomography to construct inverse noise operations [26]. For this reason, it is hard to implement for large systems [5] and highly variable noise profiles.

Van Den Berg’s Sparse Pauli Lindblad Method [28] provides an efficient approach to model and mitigate Pauli noise by utilizing sparse representations of Lindblad operators. It puts forth a principled framework for modeling quantum noise by integrating the dynamics of open quantum systems. This approach eliminates the need for resource-intensive full-noise tomography but relies on the careful selection of dominant noise terms, which requires domain expertise. It also presumes stable noise profiles and has limited adaptability to rapidly changing noise environments. This limits its applicability for NISQ hardware, where noise is unpredictable and dynamic. All of these methods aim to mitigate noise while being oblivious to the specific quantum task/ operator.

In this paper, we propose a novel gradient-based framework to mitigate Pauli noise in PQCs addressing the challenges of adaptivity, efficiency, and efficacy. Unlike conventional approaches, *our method tightly integrates the process of learning parameterized inverse noise models, constructed using a sparse Lindblad-based formulation [28], with the optimization of PQCs in the training process.* This simultaneous learning approach allows our framework to adapt dynamically to task-specific objectives and noise characteristics. We present detailed derivations and theoretical foundations of our method, establishing its superiority, and demonstrate its performance in the context of real-world quantum machine learning scenarios. Our findings demonstrate that adaptive noise mitigation improves accuracy compared to state of the art baselines. Our integrated learning approach improves the reliability of quantum machine learning on NISQ hardware.

Main Contributions: We propose a joint gradient-based framework that optimizes parameterized quantum circuits (PQCs) while mitigating Pauli noise arising during unitary gate operations, thereby enabling noise-aware training. As with all gradient-based approaches, efficient gradient computation is essential to our method. While gradients w.r.t. PQC parameters can be efficiently estimated using the parameter shift rule [17], estimating gradients for inverse noise parameters is more challenging due to exponential scaling with the qubit count. To address this, we introduce a novel and efficient gradient estimation method tailored to inverse noise parameters enabling scalable and robust optimization of PQCs.

The remainder of the paper is organized as follows: Section 2 introduces the necessary background. Section 2.3 presents our proposed integrated framework for PQC optimization. Section 3.2.1 focuses on gradient derivation while Section 3.2.2 describes our quantum algorithm to estimate gradients of all model parameters, including the inverse noise parameters, a key component of our efficient joint optimization approach. Sample complexity and convergence results are discussed in Section 3.2.3 and Section 3.3, respectively. Numerical results are presented in Section 4, with related work and concluding remarks in Section 5 and Section 6.

2 Preliminaries

2.1 Parameterized quantum Circuit (PQC)

A PQC is a crucial element in the design of Variational Quantum Algorithms (VQAs). It is represented as a parameterized unitary operator $U_{\vec{b}}(\vec{\theta})$ defined by:

$$U_{\vec{b}}(\vec{\theta}) = \prod_{l=1}^L U_{l,b_l}, \quad U_{l,b_l} = \begin{cases} U_l(\theta_l) & b_l = 1 \\ V_l & o.w. \end{cases} \quad (1)$$

where $U_l(\theta_l) = \exp\{i\theta_l G_l\}$ is a parameterized quantum gate generated by a Pauli string $G_l \in \{I, X, Y, Z\}^{\otimes n}$, θ_l are learnable parameters, V_l are non-Pauli constant unitaries, and $\vec{b} \in \{0, 1\}^L$ is a constant binary vector.

2.2 Noise Model

Pauli noise is a form of quantum noise frequently observed in quantum computing and quantum information processing. It originates from stochastic errors in quantum gate operations, where each qubit in a quantum system may experience a noise process characterized by one of the Pauli operators. This type of noise can be described as a probabilistic application of Paulis— X , Y or Z to a qubit, each occurring with a certain probability. Since any Markovian noise can be approximated using Pauli noise via Pauli twirling [8], the dominant noise affecting gate operations can be effectively modeled as Pauli noise. Based on this observation, we adopt the Pauli-Lindblad noise model introduced in [28] to define the noise model over a noise-free quantum state ρ as:

$$\Lambda(\bullet)(\rho) = \left(\bigcirc_{k \in \mathcal{K}} \left(\omega_k I \bullet I + (1 - \omega_k) P_k \bullet P_k^\dagger \right) \right) (\rho) \quad (2)$$

where $\omega_k = (1 + \exp \{-2\lambda_k\})/2$, $\lambda_k \geq 0$.

Here, $\bullet$ serves as a placeholder indicating that the map is applied to ρ , i.e., $\Lambda(\bullet)(\rho) = \Lambda(\rho)$. The notation $\bigcirc_{k \in \mathcal{K}}$ represents the composition of maps where each map corresponds to a noise model term in $\mathcal{K}$ associated with a Pauli string $P_k \in \{I, X, Y, Z\}^{\otimes n}$, and a model coefficient λ_k . These model terms reflect various noise interactions in the quantum system.

By definition, the non-physical inverse noise model to mitigate the aforementioned noise can be written as:

$$\Lambda^{-1}(\bullet)(\rho) = \left(\gamma \bigcirc_{k \in \mathcal{K}} \left(q_k I \bullet I - (1 - q_k) P_k \bullet P_k^\dagger \right) \right) (\rho) \quad (3)$$

where, $q_k = (1 + \exp \{-2\sigma_k\})/2$, $\gamma = \exp \left\{ 2 \sum_{k \in \mathcal{K}} \sigma_k \right\}$, $\sigma_k \geq 0$.

Here, values of σ_k correspond to the coefficients of inverse noise model w.r.t. Paulis P_k .

Remark 2.1. Throughout this paper, the notation for nested map composition $\bigcirc_{j=1}^L f_j(\bullet)$ is defined as

$$f_L \circ f_{L-1} \circ \cdots \circ f_2 \circ f_1(\bullet) = f_L(f_{L-1}(\cdots f_2(f_1(\bullet))))$$

for any collection of maps $f_1, \cdots, f_L$.

2.3 Noise-Mitigated PQCs

A noise-mitigated parameterized quantum circuit (nmPQC) is an augmentation of the PQC, as described in Equation (1). It is carefully designed to minimize the adverse impact of noise. In contrast to the conventional design where each layer contains only unitary transformations, each layer of the nmPQC consists of the gate U_{l,b_l} , the noise model Λ_l and its corresponding inverse noise model Λ_l^{-1} , as illustrated in Figure 1. With this design, the noise Λ_l can be mitigated by the inverse noise operator Λ_l^{-1} , thereby improving the performance of the PQC.

Formally, an L -layer nmPQC, denoted by $\mathcal{U}_R$ is defined as follows:

$$\mathcal{U}_R(\bullet) = \bigcirc_{l=1}^L \Lambda_l^{-1} \circ \Lambda_l \circ \text{Ad}_{U_{l,b_l}}(\bullet) \quad (4)$$

where $\text{Ad}_{U_{l,b_l}}(\bullet) = U_{l,b_l} \bullet U_{l,b_l}^\dagger$, U_{l,b_l} is the l -th gate unitary. Additionally, Λ_l and Λ_l^{-1} represent the noise and inverse noise models at gate l , as defined in Equation (2) and Equation (3), respectively. The noise model Λ_l is parameterized by unknown noise parameters $\lambda_{l,k}$, while the inverse noise model Λ_l^{-1} is parameterized by learnable inverse noise parameters $\sigma_{l,k}$ w.r.t. Paulis $P_k^{(l)}$, associated with each unitary U_{l,b_l} .

3 Integrated Learning Framework for nmPQC

In this section, we propose a gradient-based framework to train a parameterized quantum circuit on noisy quantum hardware, where the noise is modeled by a Pauli-Lindblad model. We begin by formalizing the learning objective and outlining the assumptions underlying our analysis.

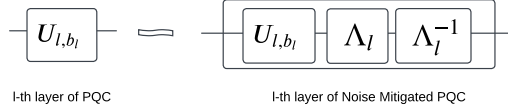


Figure 1: The diagram above illustrates a single layer of the nmPQC. The complete nmPQC is constructed by concatenating L such blocks of this structure.

3.1 Problem Statement and Assumptions

Let (ρ_t, y_t) be elements of the data set $\mathcal{D} = \{(\rho_t, y_t) \in \mathcal{B}(\mathbb{C}^N) \times \{-1, 1\}\}_{t=1}^{|\mathcal{D}|}$, where $N = 2^n$. We assume that (ρ_t, y_t) are independent and identically distributed (i.i.d.) samples drawn from a known distribution. In order to train the model $\mathcal{U}_R$ defined in Equation (4), we formulate the following optimization problem:

$$\min_{\vec{\sigma}, \vec{\theta} \in \mathbb{R}^{|\langle \vec{\sigma}, \vec{\theta} \rangle|}} \mathcal{L}(\vec{\sigma}, \vec{\theta}) + \mathcal{G}(\vec{\sigma}, \vec{\theta}) \quad (5)$$

where the loss function $\mathcal{L} : \mathbb{R}^{|\langle \vec{\sigma}, \vec{\theta} \rangle|} \rightarrow \mathbb{R}$ is defined as :

$$\mathcal{L}(\vec{\sigma}, \vec{\theta}) = \frac{1}{|\mathcal{D}|} \sum_{t=1}^{|\mathcal{D}|} \mathcal{L}_t(\vec{\sigma}, \vec{\theta}) = \frac{1}{|\mathcal{D}|} \sum_{t=1}^{|\mathcal{D}|} (y_t - \text{tr}(\mathcal{M}\mathcal{U}_R(\rho_t)))^2 \quad (6)$$

which follows a squared loss formulation and $\text{tr}(\mathcal{M}\mathcal{U}_R(\rho_t))$ denotes the predicted measurement outcome corresponding to the observable $\mathcal{M}$. The function $\mathcal{G} : \mathbb{R}^{|\langle \vec{\sigma}, \vec{\theta} \rangle|} \rightarrow \mathbb{R} \cup \{+\infty\}$ enforces constraints on the learnable parameters by restricting $\vec{\sigma}$ and $\vec{\theta}$ to predefined domains. Specifically, $\mathcal{X}_{\text{sigma}}$ constrains the inverse noise parameters $\vec{\sigma}$ for each gate, while $\mathcal{X}_{\text{theta}}$ bounds the variational PQC parameters $\vec{\theta}$, ensuring they lie within a specified range and preserve parameter feasibility during optimization. The constraint sets are defined by $\mathcal{X}_{\text{sigma}} = \prod_{l=1}^L \prod_{k \in \mathcal{K}_l} [0, \mathfrak{B}^{(l,k)}]$, $\mathcal{X}_{\text{theta}} = \prod_{b_l=1}^L [\mathfrak{B}_0^{(l)}, \mathfrak{B}_1^{(l)}]$, $\mathfrak{B}^{(l,k)} \in \mathbb{R}_{\geq 0}$, and $\mathfrak{B}_0^{(l)}, \mathfrak{B}_1^{(l)}$ satisfy $-\infty < \mathfrak{B}_0^{(l)} \leq \mathfrak{B}_1^{(l)} < \infty$.

Using these definitions, $\mathcal{G}(\vec{\sigma}, \vec{\theta})$ can be written as:

$$\mathcal{G}(\vec{\sigma}, \vec{\theta}) = \begin{cases} 0 & \vec{\sigma}, \vec{\theta} \in \mathcal{X}_{\text{sigma}} \times \mathcal{X}_{\text{theta}} \\ +\infty & \text{o.w.} \end{cases} \quad (7)$$

In this framework, the learning task is comprised of: (i) optimizing the standard PQC parameters that define the unitary transformations; and (ii) learning the inverse noise parameters $\sigma_{l,k}$, which mitigate the unknown noise effect of Λ_l . The PQCs are restricted to c -local Pauli operators, ensuring that each unitary U_{l,b_l} acts on at most c qubits. As a result, the number of inverse noise parameters $\sigma_{l,k}$ per unitary is constrained to at most 4^c , reflecting the overall number of Pauli strings in a c -qubit system.

To ensure feasibility and facilitate problem formulation, we make the following assumptions:

Assumption 3.1. The noise parameters $\lambda_{l,k}$ are constrained within the domain $\mathcal{X}_{\text{sigma}}$, which ensures that each $\lambda_{l,k}$ lies within the interval $[0, \mathfrak{B}^{(l,k)}] \subseteq \mathbb{R}_{\geq 0}$.

Assumption 3.2. The single-qubit Paulis, I , X , Y , and Z , are considered noiseless.

Assumption 3.3. The State Preparation and Measurement (SPAM) noise is absent.

Assumption 3.4. $\mathcal{M}$ is Hermitian with eigenvalues 1 and -1.

3.2 Gradient and its Estimation

As discussed in the previous section, obtaining the gradient of the objective function is essential for all gradient-based methods. In this subsection, we derive the gradient of the objective function defined in Equation (6) and present a quantum algorithm for its estimation.

3.2.1 Derivation of Gradient

First of all, we derive gradients of the per-sample loss function, $\mathcal{L}_t(\vec{\sigma}, \vec{\theta}) = (y_t - \text{tr}(\mathcal{M}\mathcal{U}_R(\rho_t)))^2$. Using the parameter shift rule as described in [17], we compute the partial derivative of $\mathcal{L}_t(\vec{\sigma}, \vec{\theta})$ w.r.t. the PQC parameter θ_j (from Appendix A.2.2) as:

$$\frac{\partial \mathcal{L}_t(\vec{\sigma}, \vec{\theta})}{\partial \theta_j} = -2(y_t - \text{tr}(\mathcal{M}\mathcal{U}_R(\rho_t))) \cdot \text{tr}\left(\mathcal{M}\mathcal{U}_{t,>j}\left(\Lambda_j^{-1} \circ \Lambda_j \circ \mathcal{U}_{j,\frac{\pi}{2}}\left(\rho_t^{(j-1)}\right)\right)\right). \quad (8)$$

Here, we denote $\mathcal{U}_{j,a} = \frac{1}{2}\text{Ad}_{U_j(\theta_j+a)} - \frac{1}{2}\text{Ad}_{U_j(\theta_j-a)}$ for $a \in \mathbb{R}$, and introduce the intermediate result $\rho_t^{(j-1)}$ defined by: $\rho_t^{(j-1)} = \left(\bigcirc_{l=1}^{j-1} \Lambda_l^{-1} \circ \Lambda_l \circ \text{Ad}_{U_{l,b_l}}\right)(\rho_t)$.

Next, differentiating $\mathcal{L}_t(\vec{\sigma}, \vec{\theta})$ w.r.t. an inverse noise parameter $\sigma_{j,q}$ yields (see Appendix A.2.1):

$$\frac{\partial \mathcal{L}_t(\vec{\sigma}, \vec{\theta})}{\partial \sigma_{j,q}} = -4(y_t - \text{tr}(\mathcal{M}\mathcal{U}_R(\rho_t))) \cdot \text{tr}\left(\mathcal{M}\mathcal{U}_{t,>j}\left(\Lambda_j^{-1}\left(\sum_{g \in \mathbf{G}_q^{(j)}} \alpha_{j,g} g\right)\right)\right). \quad (9)$$

where $\mathbf{G}_q^{(j)} = \{g \in \{I, X, Y, Z\}^{\otimes n} \mid \{P_q^{(j)}, g\} = 0\}$ is the subset of Pauli strings that anti-commute with $P_q^{(j)}$, and the terms within the innermost parentheses arise from the Pauli decomposition of the Hermitian matrix $\rho_{t,j} = \Lambda_j \circ \text{Ad}_{U_{j,b_j}} \circ \left(\bigcirc_{l=1}^{j-1} \Lambda_l^{-1} \circ \Lambda_l \circ \text{Ad}_{U_{l,b_l}}\right)(\rho_t)$. Plugging in the identity stated in Proposition A.14, we simplify further to arrive at:

$$\frac{\partial \mathcal{L}_t(\vec{\sigma}, \vec{\theta})}{\partial \sigma_{j,q}} = -4(y_t - \text{tr}(\mathcal{M}\mathcal{U}_R(\rho_t))) \cdot \text{tr}\left(\mathcal{M}\mathcal{U}_{t,>j}\left(\Lambda_j^{-1} \circ \mathcal{U}_{P_q^{(j)}}(\rho_{t,j})\right)\right), \quad (10)$$

where $\mathcal{U}_{P_q^{(j)}}(\bullet) = \frac{1}{2}(I \bullet I - P_q^{(j)} \bullet (P_q^{(j)})^\dagger)$.

3.2.2 Universal Estimation Algorithm

By examining equations Equation (8) and Equation (10), it becomes clear that the primary computational challenge in evaluating these partial derivatives arises from estimating the trace terms. These trace terms can be categorized into three distinct types: $\text{tr}(\mathcal{M}\mathcal{U}_R(\rho_t))$, $\text{tr}\left(\mathcal{M}\mathcal{U}_{t,>j}\left(\Lambda_j^{-1} \circ \text{Ad}_{P_q^{(j)}}(\rho_{t,j})\right)\right)$, and $\text{tr}\left(\mathcal{M}\mathcal{U}_{t,>j}\left(\text{Ad}_{U(\theta_j+a)}(\rho_{t,j})\right)\right)$. To facilitate the estimation of the relevant trace terms, we reduce each one to a standard form and present the derivations in Equation (28), Equation (29), and Equation (30). Upon analysis of these expressions, we observe a common structural form across them:

$$\text{tr}\left(\mathcal{M} \bigcirc_{l=1}^L \Lambda_l^{-1} \circ \text{Ad}_{W_l} \circ \Lambda_l \circ \text{Ad}_{U_l}(\rho)\right), \quad (11)$$

where W_l is a Pauli string and U_l is an unitary. Building on this observation, we introduce the universal estimation algorithm, described in Algorithm 2, designed to unbiasedly approximate the expression in Equation (11) (see Lemma A.16 for details). The corresponding quantum circuit implementation is illustrated in Figure 2.

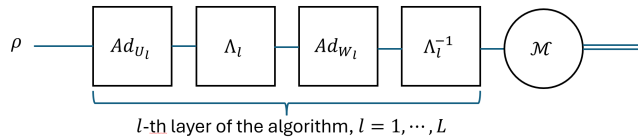


Figure 2: (1) The quantum circuit implementing the universal estimation algorithm: operator W_l corresponds to a fixed Pauli string, while U_l are constant or parameterized (variational) c -local non-Pauli quantum operations. (2) Note that the inverse channels Λ_l^{-1} are non-physical and cannot be directly realized on quantum hardware. Consequently, the implementation of the circuit requires classical preprocessing. Further details are provided in Appendix A.3.1.

Importantly, this quantum-classical hybrid algorithm enables unbiased estimation of these trace terms on noisy quantum hardware without requiring explicit acquisition of the structure of noise channels Λ_L , which is typically computationally expensive.

3.2.3 Sample Complexity

The universal estimation algorithm serves as the fundamental building block for the partial derivative estimation procedures defined in Algorithm 3 and Algorithm 4, corresponding to the partial derivatives of $\mathcal{L}_t$ given in Equation (8) and Equation (10). In essence, the universal estimation algorithm is executed repeatedly to obtain unbiased approximations of the quantities defined in Equation (8) and Equation (10), up to a prescribed accuracy (see Proposition A.17 for details). Ultimately, by iteratively applying Algorithm 3 and Algorithm 4, we obtain an approximation of the gradient of $\mathcal{L}_t$ at a given point $(\vec{\theta}, \vec{\sigma})$.

A natural question arises regarding sample complexity: how many measurements are required to obtain a reliable estimate of the gradient at $(\vec{\sigma}, \vec{\theta})$? Referring to Lemma A.22, we observe that, with probability at least $1 - \delta$, the number of measurements (or number of runs) required to estimate

a single partial derivative within an additive error ε is no less than $\frac{32^2}{2((\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \varepsilon)^2} \ln \left(\frac{2|(\vec{\sigma}, \vec{\theta})|}{\delta} \right)$,

where $\Gamma_{\vec{\sigma}} = \exp \left\{ 2 \sum_{l=1}^L \sum_{k \in \mathcal{K}_l} \sigma_{l,k} \right\}$. This expression shows that the sample complexity (or the measurement complexity) can grow rapidly if ε or δ is too small or if the number of parameters $|(\vec{\sigma}, \vec{\theta})|$ becomes large, making scalability a significant challenge.

To address the scalability challenge, we incorporate two key techniques. First, in each update round, we randomly select a subset of coordinate directions to update, where each direction is independently chosen with probability $1/p$ for some integer $p \geq 1$ (see Algorithm 5). This sampling strategy reduces the number of estimations required per iteration. The resulting sample complexity is summarized in the following theorem:

Theorem 3.1 (Informal Version of Theorem A.23). *Under Assumption 3.2 - Assumption 3.4, The expected sample complexity (measurement complexity) required for Algorithm 5 to obtain a estimation of the gradient with additive error $\varepsilon > 0$ with probability at least $1 - \delta$, where $\delta \in (0, 1)$ is at least*

$$\frac{|(\vec{\sigma}, \vec{\theta})|}{p} \cdot \frac{32^2}{2((\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \varepsilon)^2} \ln \left(\frac{2|(\vec{\sigma}, \vec{\theta})|}{\delta p} \right)$$

Second, when the current parameters $(\vec{\sigma}, \vec{\theta})$ are far from a stationary point, we relax the estimation accuracy to prioritize correct gradient directions over high precision. Concretely, we set the error tolerance to $\varepsilon = \Gamma_{\vec{\sigma}}^2 \kappa$ for some $\kappa \in (0, 1)$. In particular, by applying the bound established in

Theorem 3.1, the expected total number of measurements per update round is at least $\frac{|(\vec{\sigma}, \vec{\theta})|}{p} \cdot \frac{32^2}{2\kappa^2} \ln \left(\frac{2|(\vec{\sigma}, \vec{\theta})|}{\delta p} \right)$. This adaptive strategy ensures that early-stage updates remain computationally efficient while preserving convergence behavior.

3.3 Optimization and Convergence

To train the model introduced in Equation (4), we consider the optimization problem defined in Equation (5). Specifically, we optimize the objective function in Equation (5) using the proximal stochastic gradient descent (proximal SGD) method, as described in Algorithm 1. Unlike standard SGD, proximal SGD updates the parameters according to the rule

$$(\vec{\sigma}^{(t+1)}, \vec{\theta}^{(t+1)}) \leftarrow \text{prox}_{\mathcal{G}} \left((\vec{\sigma}^{(t)}, \vec{\theta}^{(t)}) - \eta^{(t)} \tilde{\nabla}_{\vec{\sigma}, \vec{\theta}}^{(t)} \mathcal{L}_{i_t} \right), \quad (12)$$

where $\tilde{\nabla}_{\vec{\sigma}, \vec{\theta}}^{(t)} \mathcal{L}_{i_t}$ denotes the estimated gradient of $\mathcal{L}_{i_t}$, generated by Algorithm 5 at the point $(\vec{\sigma}, \vec{\theta})$ for a randomly selected index $i_t \in [|\mathcal{D}|]$, and $\text{prox}_{\mathcal{G}}$ denotes the proximal operator associated with $\mathcal{G}$.

The proximal operator is defined as

$$\text{prox}_G(\sigma^{(t)}, \theta^{(t)}) = \underset{(\sigma^{(t')}, \theta^{(t')}) \in \mathcal{X}}{\text{argmin}} \left(G(\sigma^{(t')}, \theta^{(t')}) + \frac{1}{2} \|(\sigma^{(t')}, \theta^{(t')}) - (\sigma^{(t)}, \theta^{(t)})\|^2 \right).$$

where $\mathcal{X} \equiv \mathcal{X}_{\text{sigma}} \times \mathcal{X}_{\text{theta}}$ is the search space. This construction ensures that every element of the sequence $\left\{(\vec{\sigma}^{(t)}, \vec{\theta}^{(t)})\right\}_{t=1}^T$ generated by the update rule in Equation (12) remains within $\mathcal{X}$, guaranteeing feasibility throughout the optimization process.

Algorithm 1 Learning Algorithm

Dataset: $\mathcal{D} = \{(\rho_t, y_t)\}$

Input: nmPQC (number of layers L , $\vec{b} \in \mathbb{R}^L$), Measurement $\mathcal{M}$, Total iterations T , Learning rate schedule $\{\eta^{(t)}\}_{t=1}^T$, Number of shots $\tilde{N} \in (2\mathbb{Z})_{\geq 0}$, and positive integer p .

{The non-increasing learning rate schedule $\eta^{(t)}$ is such that $\eta^{(t)} \in (0, 1)$, and $\sum_{t=0}^T \eta^{(t)}$ is finite}

Initialization: $\forall l \in \left[\left\lceil \left\| \vec{b} \right\|_0 \right\rceil\right]$, $\theta_l \sim \text{Uniform}(\mathfrak{B}_0^{(l)}, \mathfrak{B}_1^{(l)}); \forall l \in [L], k \in \mathcal{K}_l, \sigma_{l,k} \leftarrow 0$.

for $t = 1, \dots, T$ **do**

$i_t \sim \text{uniform}(1, \dots, |\mathcal{D}|)$ {Sample a random index}

$\vec{p}_t \sim (\text{Bernoulli}(1/p))^{|\vec{\sigma}, \vec{\theta}|}$

$\tilde{\nabla}_{\vec{\sigma}, \vec{\theta}}^{(t)} \mathcal{L}_{i_t} \leftarrow \text{gradient_estimator} \left(i_t, M, \vec{\sigma}^{(t)}, \vec{\theta}^{(t)}, y_{i_t}, \tilde{N}, \vec{p}_t \right)$ {Algorithm 5}

$(\vec{\sigma}^{(t+1)}, \vec{\theta}^{(t+1)}) \leftarrow \text{prox}_G \left((\vec{\sigma}^{(t)}, \vec{\theta}^{(t)}) - \eta^{(t)} \tilde{\nabla}_{\vec{\sigma}, \vec{\theta}}^{(t)} \mathcal{L}_{i_t} \right)$

end for

We now analyze the convergence of the learning algorithm defined in Algorithm 1. To begin, we present several key properties of the objective function and the estimated gradient. First, in Lemma A.28, we show that the function $\mathcal{L}$ is locally Lipschitz continuous over the compact set $\mathcal{X}$, a key property for ensuring convergence. Next, in Lemma A.29, we demonstrate that $\mathcal{L} + \mathcal{G}$ is lower bounded, thereby ensuring the existence of an optimizer. Finally, in Proposition A.30, we prove that the estimated gradient produced by Algorithm 5 satisfies the bounded variance condition, which is a necessary requirement for bounded convergence behavior. Building on these properties, we summarize the convergence rate of the learning algorithm in the following theorem:

Theorem 3.2 (Informal Version of Theorem A.31). *Under Assumption 3.1, for the non-convex objective function $\mathcal{L} + \mathcal{G}$ defined in Equation (6), the convergence rate of the Algorithm 1 satisfies*

$$\left(\frac{1}{\sum_{t=1}^T \eta_t} \sum_{t=1}^T \eta_t \mathbb{E}[\mathcal{L}(w_t)] \right) - \mathcal{L}(\bar{w}) \leq \frac{p \cdot \|w_0 - \bar{w}\|_2^2}{2 \sum_{t=1}^T \eta_t} + \frac{3L_{\mathcal{X}} \sum_{t=1}^T \eta_t \mathfrak{B}_{\max}}{2 \sum_{t=1}^T \eta_t} + \frac{p \sum_{t=1}^T \eta_t^2 \mathcal{V}}{2 \sum_{t=1}^T \eta_t},$$

where $w_0 = (\vec{\sigma}^{(0)}, \vec{\theta}^{(0)})$ is the initial point, $\bar{w} = \overline{(\vec{\sigma}, \vec{\theta})}$ denotes a stationary point, and $w_t = (\vec{\sigma}^{(t)}, \vec{\theta}^{(t)})$ are the iterates generated by proximal SGD. The step size $\eta^{(t)} \in (0, 1)$ forms a monotonically decreasing sequence such that $\sum_{t=0}^{\infty} \eta_t < \infty$. The constants $\mathcal{V}$ and $\mathfrak{B}_{\max}$ are defined in Theorem A.31.

Remark 3.3. The factor p in the bound above arises from the random selection strategy employed in Algorithm 5. A larger value of p leads to a sparser estimated gradient $\tilde{\nabla}_{\vec{\sigma}, \vec{\theta}}^{(t)} \mathcal{L}_{i_t}$, which in turn results in a slower convergence rate.

4 Numerical Results

4.1 Experimental Setup and Evaluation

We evaluate our learning algorithm (Algorithm 1) on a binary classification task on the standard MNIST dataset as in prior related efforts [3, 30], focusing on digits 3 and 6. The data is preprocessed via principal component analysis (PCA) to reduce dimensionality, then normalized and encoded into

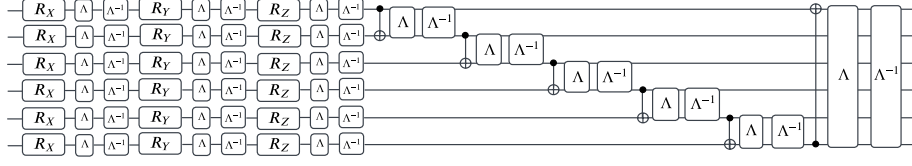


Figure 3: A sample HEA layer features a 6-qubit $R_X + R_Y + R_Z + \text{CNOT}$ design. Here, Λ and Λ^{-1} are gate-specific; one-qubit noise and its inverse are applied to R_X , R_Y and R_Z , while two-qubit noise models are used for the non-parameterized CNOT gates.

quantum states using amplitude encoding, with binary labels: -1 for digit 3 and 1 for digit 6. A total of 5000 input-label pairs are generated, which are randomly shuffled and split into 80% for training and 20% for testing. To isolate training-time noise and mitigation, we directly initialize the simulator to the amplitude-encoded PCA state (i.e., no explicit state-preparation circuit). For the nmPQC, we employ a noisy 6-qubit hardware-efficient ansatz (HEA), as illustrated in Figure 3, consisting of two layers of parameterized single-qubit rotation gates R_X , R_Y , and R_Z , combined with circular entangling CNOT gates as the base PQC. At the end of the circuit, all qubits are measured in Z -basis. We determine the initial learning rate, $\eta^{(1)}$, via a coarse-to-fine search: a wide log-spaced grid sweep to localize a promising interval, followed by a binary search within that interval to refine the value. During training, we maintain a monotonically decreasing learning-rate schedule. The framework is simulated on a multi-core CPU computing cluster, and the simulation is implemented using Qiskit [10].

To assess the effectiveness of the proposed mitigation framework which augments the PQC with nmPQC (as in Figure 3), we compare it with the following configurations: (1) A noiseless PQC, serving as a best-case performance benchmark. (2) A noisy PQC without mitigation, serving as a baseline. (3) A mitigated PQC using a probabilistic quantum noise model, adapted from Van Den Berg et al. [28]. We pre-estimate noise parameters using their procedure and fix them as inverse noise parameters during training. While their method is not originally intended for variational optimization, we adapt it as a benchmark, referring to it as “Van Den Berg et al.’s mitigation approach”. For training we use proximal SGD to optimize the model parameters along with the inverse noise parameter (for our method). Model performance is evaluated by accuracy, while noise mitigation efficacy is assessed using mean squared error (MSE) over training epochs.

4.2 Experimental Results

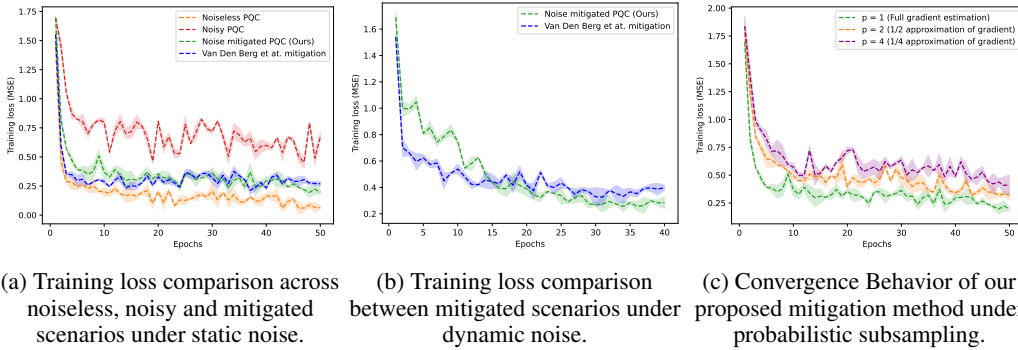


Figure 4: Training performance of the binary classification using MNIST dataset in the quantum noise context. Each epoch summarizes 50 iterations of the training process.

We evaluate the effectiveness of our proposed method by comparing its training loss and testing accuracy against alternative configurations in the trivial scenario (i.e., full gradient estimation). Each experiment is repeated three times per setting, and the standard deviation is used to report variability. Figure 4a shows the convergence behavior of different methods over epochs under a static noise setting, where all CNOT gates in the noisy and mitigated models are assigned fixed error rates. The

noiseless model serves as a best-case reference. As expected, the noiseless PQC achieves the lowest training loss. The noisy PQC without any mitigation mechanism exhibits significantly higher loss, highlighting the effects of quantum noise. Both mitigation strategies—our proposed method and that of Van Den Berg et al.[28] achieve an improved reduction in training loss compared to the noisy baseline. The Van Den Berg et al. approach pre-estimates noise models using the random benchmarking technique [9] and applies their inverses using the learned noise parameters during training, yielding an initial advantage. In contrast, our method learns the inverse noise parameters during training, eliminating the need for noise pre-characterization and calibration, and enabling streamlined, task-specific adaptive mitigation. We observe that Van Den Berg et al.’s approach achieves lower training loss initially, but as training continues, our method surpasses it, ultimately achieving slightly better overall performance. Table 1 summarizes the experimental accuracies of different approaches, showing that our method outperforms the baseline noisy PQC and Van Den Berg et al.’s.

To evaluate adaptability, we simulate dynamic noise by slightly increasing error rates of three CNOT gates shortly after training begins for mitigated methods only. While Van Den Berg et al.’s approach continues using its static pre-estimated noise model, our method dynamically updates the inverse noise parameters during training. As shown in Figure 4b, our method effectively learns the inverse noise parameters in response to dynamic changes and ultimately achieves lower MSE than the method of Van Den Berg et al.

Table 1: Classification accuracies across Different methods

	ACCURACY
NOISELESS PQC	94.13± 0.31
NOISY PQC (BASELINE)	79.69± 3.70
MITIGATED PQC (VAN DEN BERG ET.AL)	85.37± 3.44
OURS	86.43± 3.14

In addition, we investigate the trade-off between gradient estimation cost and optimization performance by evaluating the convergence behavior of our learning algorithm (Algorithm 1) under probabilistic subsampling (see Appendix A.3.3). This experiment uses the same static noise profile as in Figure 4a. In this setting, for a fixed parameter p , each coordinate direction (i.e., parameter) is selected independently with probability $\frac{1}{p}$ during each gradient update step. The case $p = 1$ corresponds to full gradient estimation, while larger p values introduce sparsity in the gradient estimates.

Figure 4c depicts the convergence behavior for various p values over 50 epochs. For $p = 1$, we observe faster and smoother convergence, with a steady decline in loss and minimal fluctuations, demonstrating the benefit of full gradient estimation. As p increases, convergence slows, and the loss curve becomes noisier with higher overall loss levels, reflecting the impact of sparser gradient updates.

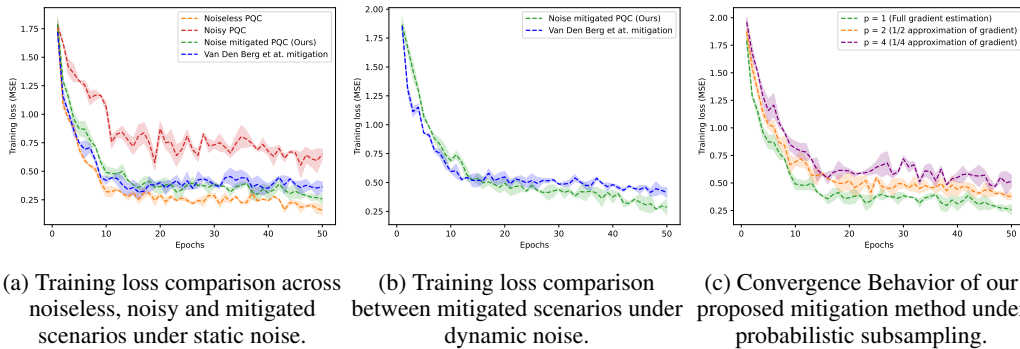


Figure 5: Training performance of the binary classification using Fashion-MNIST dataset (Pullover vs Shirt) in the quantum noise context. Each epoch summarizes 50 iterations of the training process.

Additional results. We repeat the same protocol on the Fashion-MNIST dataset (Pullover vs. Shirt) [32]; trends mirror MNIST under static (Figure 5a) and dynamic (Figure 5b) noise, and probabilistic subsampling behaves similarly (Figure 5c). Additional experimental details and the accuracy table appear in Appendix A.6.1. Furthermore, under the same dynamic-noise protocol across both datasets, we compare PEC and ZNE against our method (see Appendix A.6.2); in our experiments, our framework adapts online and attains the best final metric (lower loss/MSE).

5 Related Work

PQCs are central to many quantum machine learning models, particularly in VQAs [2], originally introduced for ground state preparation [21]. VQAs adopt a hybrid quantum-classical loop, where a quantum circuit evaluates an objective and a classical optimizer adjusts parameters. This framework supports applications such as quantum chemistry [15], combinatorial optimization [6], and quantum neural networks (QNNs) [24, 23, 1].

Despite their flexibility, PQCs are susceptible to quantum noise, including decoherence, gate noise, and SPAM errors [22]. Quantum Error Correction (QEC) offers robust protection [27], but its overhead remains prohibitive for NISQ devices. Instead, Quantum Error Mitigation (QEM) methods such as ZNE [14, 26] aim to reduce noise without heavy resource demands.

Pauli noise, a common model of gate errors, can exponentially suppress gradients in VQAs, impeding training [8, 31]. Several approaches have emerged to address this. PEC techniques model noise as an invertible stochastic process, correcting it via probabilistic sampling [26]. Learning-based strategies such as those by Strikis et al. [25] and Czarnik et al. [4] train on auxiliary circuits to estimate mitigated expectations. Van den Berg et al. [28] propose incorporating Lindblad-modeled noise directly into the optimization. A recent diffusion-inspired approach [19] mitigates noise via learned forward-backward dynamics.

6 Discussions

We propose a gradient-based framework for jointly learning parameterized quantum circuits (PQCs) and inverse noise operators, enabling noise parameter estimation during training. This approach supports rapid recalibration when the noise model changes, thereby avoiding full retraining and improving adaptability on near-term devices. A parameter-shift-like rule is introduced for scalable, hardware-efficient gradient estimation with respect to inverse noise parameters, ensuring robustness under Pauli noise. While effective, the method faces challenges: parameter growth under low locality (e.g., $c > \Omega(\log n)$) can increase computational cost, and the reliance on simple measurement may limit broader applicability. Future work will focus on reducing the complexity of inverse noise operators and extending applicability to general measurement schemes.

Acknowledgments

This work is partially supported by the NSF Center for Science of Information (CSoI) Grant CCF-0939370, and also by NSF Grants CCF-2006440 and CCF-2211423.

References

- [1] Samuel Beer, Bruno Voss, Alex Vartanian, and Ben Barak. Training deep quantum neural networks. *npj Quantum Information*, 6(1):1–7, 2020.
- [2] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021.
- [3] Samuel Yen-Chi Chen, Chih-Min Huang, Chia-Wei Hsing, and Ying-Jer Kao. Hybrid quantum-classical classifier based on tensor network and variational quantum circuit. *arXiv preprint arXiv:2011.14651*, 2020.

- [4] Piotr Czarnik, Andrew Arrasmith, Patrick J Coles, and Lukasz Cincio. Error mitigation with clifford quantum-circuit data. *Quantum*, 5:592, 2021.
- [5] Suguru Endo, Simon C Benjamin, and Ying Li. Practical quantum error mitigation for near-future applications. *Physical Review X*, 8(3):031027, 2018.
- [6] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. Quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.
- [7] Guillaume Garrigos and Robert M Gower. Handbook of convergence theorems for (stochastic) gradient methods. *arXiv preprint arXiv:2301.11235*, 2023.
- [8] Michael R Geller and Zhongyuan Zhou. Efficient error models for fault-tolerant architectures and the pauli twirling approximation. *Physical Review A—Atomic, Molecular, and Optical Physics*, 88(1):012314, 2013.
- [9] Jonas Helsen, Xiao Xue, Lieven MK Vandersypen, and Stephanie Wehner. A new class of efficient randomized benchmarking protocols. *npj Quantum Information*, 5(1):71, 2019.
- [10] Ali Javadi-Abhari, Matthew Treinish, Kevin Krsulich, Christopher J Wood, Jake Lishman, Julien Gacon, Simon Martiel, Paul D Nation, Lev S Bishop, Andrew W Cross, et al. Quantum computing with qiskit. *arXiv preprint arXiv:2405.08810*, 2024.
- [11] Abhinav Kandala, Kristan Temme, Antonio D Córcoles, Antonio Mezzacapo, Jerry M Chow, and Jay M Gambetta. Error mitigation extends the computational reach of a noisy quantum processor. *Nature*, 567(7749):491–495, 2019.
- [12] Emanuel Knill. Quantum computing with realistically noisy devices. *Nature*, 434(7029):39–44, 2005.
- [13] John M Lee and John M Lee. *Smooth manifolds*. Springer, 2012.
- [14] Ying Li and Simon C Benjamin. Efficient variational quantum simulator incorporating active error minimization. *Physical Review X*, 7(2):021050, 2017.
- [15] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023023, 2016.
- [16] Colin McDiarmid et al. On the method of bounded differences. *Surveys in combinatorics*, 141(1):148–188, 1989.
- [17] Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii. Quantum circuit learning. *Physical Review A*, 98(3):032309, 2018.
- [18] James Munkres. *Topology*.
- [19] Hoang-Quan Nguyen, Xuan Bac Nguyen, Samuel Yen-Chi Chen, Hugh Churchill, Nicholas Borys, Samee U Khan, and Khoa Luu. Diffusion-inspired quantum noise mitigation in parameterized quantum circuits. *arXiv preprint arXiv:2406.00843*, 2024.
- [20] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010.
- [21] 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(1):4213, 2014.
- [22] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.
- [23] Maria Schuld, Ronny Bechtel, and Johan Meyer. Circuit-centric quantum classifiers. *Quantum Machine Learning*, pages 37–74, 2020.
- [24] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. An introduction to quantum machine learning. *Contemporary Physics*, 56(2):172–185, 2015.

- [25] Armands Strikis, Dayue Qin, Yanzhu Chen, Simon C Benjamin, and Ying Li. Learning-based quantum error mitigation. *PRX Quantum*, 2(4):040330, 2021.
- [26] Kristan Temme, Sergey Bravyi, and Jay M Gambetta. Error mitigation for short-depth quantum circuits. *Physical review letters*, 119(18):180509, 2017.
- [27] Barbara M Terhal. Quantum error correction for quantum memories. *Reviews of Modern Physics*, 87(2):307–346, 2015.
- [28] Ewout Van Den Berg, Zlatko K Mineev, Abhinav Kandala, and Kristan Temme. Probabilistic error cancellation with sparse pauli–lindblad models on noisy quantum processors. *Nature physics*, 19(8):1116–1121, 2023.
- [29] Joel J Wallman and Joseph Emerson. Noise tailoring for scalable quantum computation via randomized compiling. *Physical Review A*, 94(5):052325, 2016.
- [30] Hanrui Wang, Yongshan Ding, Jiaqi Gu, Yujun Lin, David Z Pan, Frederic T Chong, and Song Han. Quantumnas: Noise-adaptive search for robust quantum circuits. In *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 692–708. IEEE, 2022.
- [31] Samson Wang, Enrico Fontana, Marco Cerezo, Kunal Sharma, Akira Sone, Lukasz Cincio, and Patrick J Coles. Noise-induced barren plateaus in variational quantum algorithms. *Nature communications*, 12(1):6961, 2021.
- [32] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The abstract and the introduction point out the major contribution and scope of the paper.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: The limitations of the work is mentioned in Section 6. Besides, the assumptions are also discussed in Section 2.3.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: The paper provides complete proof for the theoretical results along with the assumptions where necessary.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: Source code with reproduction instruction is provided as supplementary materials.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The source code is made available along with the submission.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: They required information are provided along with the source code as a markdown file.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The error bars are denoted as shadows in the figures for experimental results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: The required information is provided along with the source code as supplementary material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The authors have made a concerted effort to comply with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: This work is not expected to have any direct societal impact, to the best of our understanding.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: As far as we can tell, the paper does not involve any identifiable risks as mentioned above.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The dataset used in this work along with the code for generating existing Linblad noise mitigation techniques have been cited properly.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: The source code and relevant information are made available.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[NA\]](#)

Justification: This work does not involve research involving human subjects or crowdsourcing experiments.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [\[NA\]](#)

Justification: [\[NA\]](#)

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

A Appendix

In the following appendices, we provide additional technical details that complement the findings presented in the main manuscript. In Appendix A.1, we offer a concise overview of key concepts in quantum computing and properties of the Pauli-Lindblad noise model. In Appendix A.2, we present a detailed derivation of the gradient of the loss function. In Appendix A.3, we describe the algorithms used for gradient estimation. In Appendix A.4, we discuss the sample complexity of the quantum algorithm. Finally, in Appendix A.5, we characterize the convergence behavior of the proposed quantum learning approach and provide an analysis of its performance.

A.1 Preliminary

A.1.1 Overview

We present a brief overview of some key concepts and definitions in quantum computing.

Definition A.1 (Space of Linear Operators). If $\mathcal{H}$ be a finite-dimensional Hilbert space, then let $\mathcal{B}(\mathcal{H})$ be the set of all linear operators on $\mathcal{H}$, that is:

$$\mathcal{B}(\mathcal{H}) = \{A : \mathcal{H} \rightarrow \mathcal{H}\}.$$

Remark A.2. In this paper, we assume $\mathcal{H} = \mathbb{C}^N$, where $N = 2^n$ and n is the number of qubits in the system. In other words, $\mathcal{B}(\mathbb{C}^N)$ is the set of all linear operators from $\mathbb{C}^N$ to $\mathbb{C}^N$.

Definition A.3 (c -local). If a Hamiltonian H is supported on at most c of the $n \geq c$ qubits (i.e., H acts on qubits in $S \subseteq [n]$, where $|S| = c$), it is called a c -**local** Hamiltonian (on n qubits).

Remark A.4. In this paper, we say the locality is high when $c = O(\log n)$, where n is the number of qubits.

Theorem A.5 (Quantum Channel [20]). A map $\mathcal{E} : \mathcal{B}(\mathcal{H}) \rightarrow \mathcal{B}(\mathcal{H})$ is a quantum channel if and only if $\mathcal{E}$ satisfies the following three conditions:

1. (**Trace Preserving**) $\text{tr}(\mathcal{E}(\rho)) = 1$ for all mixed states ρ on $\mathcal{H}$.
2. (**Convex**) $\mathcal{E}(\sum_i p_i \rho_i) = \sum_i p_i \mathcal{E}(\rho_i)$, where $\{p_i\}$ is a probability distribution.
3. (**Completely Positive**) For all positive Hermitian operators $A \in \mathcal{B}(\mathcal{H})$, $\mathcal{E}(A)$ is also a positive operator.

A.1.2 Pauli Matrices and Pauli Strings

The Pauli matrices I, X, Y, Z are defined as the following 2-by-2 traceless complex matrices:

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

A Pauli string P of length n is an element in $\{I, X, Y, Z\}^{\otimes n}$. Since $I = 1 \cdot (X)^0 (Z)^0$, $X = 1 \cdot (X)^1 (Z)^0$, $Y = i \cdot (X)^1 (Z)^1$, and $Z = 1 \cdot (X)^0 (Z)^1$, any Pauli string $P \in \{I, X, Y, Z\}^{\otimes n}$ can be uniquely expressed as follows:

$$P = \xi \bigotimes_{j=1}^n (X)^{\vec{x}(j)} (Z)^{\vec{z}(j)},$$

where $\xi \in \{\pm 1, \pm i\}$, $\vec{x}, \vec{z} \in \{0, 1\}^n$ are binary vectors, and $\vec{x}(j)$ ($\vec{z}(j)$) represents the j -th component of the vector $\vec{x}$ ($\vec{z}$). Given this observation, we can now define the symplectic inner product between two Pauli strings.

Definition A.6 (Symplectic Inner Product (Less General)). If $P_1 = \xi_1 \bigotimes_{j=1}^n (X)^{\vec{x}_1(j)} (Z)^{\vec{z}_1(j)}$ and $P_2 = \xi_2 \bigotimes_{j=1}^n (X)^{\vec{x}_2(j)} (Z)^{\vec{z}_2(j)}$ are any two Pauli strings, the symplectic inner product between P_1 and P_2 , denoted by $(P_1, P_2)_{sp}$ is defined by:

$$(P_1, P_2)_{sp} = (\vec{x}_1 \cdot \vec{z}_2 + \vec{z}_1 \cdot \vec{x}_2) \mod 2.$$

The following proposition is a direct application of Definition A.6.

Proposition A.7. *Let $P_1 = \xi_1 \bigotimes_{j=1}^n (X)^{\vec{x}_1(j)} (Z)^{\vec{z}_1(j)}$ and $P_2 = \xi_2 \bigotimes_{j=1}^n (X)^{\vec{x}_2(j)} (Z)^{\vec{z}_2(j)}$ be any two Pauli strings. Then,*

$$P_1 P_2 = (-1)^{(P_1, P_2)_{sp}} P_2 P_1.$$

Sketch Proof of Proposition A.7. First, we note that for any $b_1, b_2 \in \{0, 1\}$,

$$(X)^{b_1} (Z)^{b_2} = (-1)^{b_1 b_2} (Z)^{b_2} (X)^{b_1}$$

Using the properties of Pauli matrices and Kronecker product, we obtain

$$\begin{aligned} & P_1 P_2 \\ &= \left(\xi_1 \bigotimes_{j=1}^n (X)^{\vec{x}_1(j)} (Z)^{\vec{z}_1(j)} \right) \left(\xi_2 \bigotimes_{j=1}^n (X)^{\vec{x}_2(j)} (Z)^{\vec{z}_2(j)} \right) \\ &= \xi_1 \xi_2 \bigotimes_{j=1}^n (X)^{\vec{x}_1(j)} (Z)^{\vec{z}_1(j)} (X)^{\vec{x}_2(j)} (Z)^{\vec{z}_2(j)} \\ &= \xi_1 \xi_2 \bigotimes_{j=1}^n (-1)^{\vec{z}_1(j) \cdot \vec{x}_2(j)} \cdot (-1)^{\vec{x}_1(j) \cdot \vec{z}_2(j)} (X)^{\vec{x}_2(j)} (Z)^{\vec{z}_2(j)} (X)^{\vec{x}_1(j)} (Z)^{\vec{z}_1(j)} \\ &= \xi_1 \xi_2 (-1)^{\sum_{j=1}^n \vec{z}_1(j) \cdot \vec{x}_2(j)} \cdot (-1)^{\sum_{j=1}^n \vec{x}_1(j) \cdot \vec{z}_2(j)} \bigotimes_{j=1}^n (X)^{\vec{x}_2(j)} (Z)^{\vec{z}_2(j)} (X)^{\vec{x}_1(j)} (Z)^{\vec{z}_1(j)} \\ &= \xi_1 \xi_2 (-1)^{\vec{z}_1 \cdot \vec{x}_2 + \vec{x}_1 \cdot \vec{z}_2} \bigotimes_{j=1}^n (X)^{\vec{x}_2(j)} (Z)^{\vec{z}_2(j)} (X)^{\vec{x}_1(j)} (Z)^{\vec{z}_1(j)} \\ &= \xi_1 \xi_2 (-1)^{(\vec{z}_1 \cdot \vec{x}_2 + \vec{x}_1 \cdot \vec{z}_2) \bmod 2} \bigotimes_{j=1}^n (X)^{\vec{x}_2(j)} (Z)^{\vec{z}_2(j)} (X)^{\vec{x}_1(j)} (Z)^{\vec{z}_1(j)} \\ &= (-1)^{(P_1, P_2)_{sp}} P_2 P_1. \end{aligned}$$

□

The following corollary follows immediately from Proposition A.7.

Corollary A.8. *Two Pauli strings commute if and only if their symplectic inner product is 0.*

A.1.3 Pauli-Lindblad Noise Model

As suggested in [28], the Pauli-Lindblad equation with no internal Hamiltonian dynamics can be written as:

$$\frac{\partial}{\partial t} \rho(t) = \mathcal{L} \rho(t), \quad \text{where } \mathcal{L}(\rho) = \sum_{k \in \mathcal{K}} \lambda_k (P_k \rho P_k^\dagger - \rho) \quad (13)$$

and the noise model derived from Equation (13) can be expressed as the following quantum channel:

$$\Lambda(\rho) = \bigcirc_{k \in \mathcal{K}} \left(\omega_k \rho + (1 - \omega_k) P_k \rho P_k^\dagger \right)$$

for some $\mathcal{K} \subseteq [4^n - 1]$, where $\omega_k = (1 + \exp \{-2\lambda_k\})/2$. To construct the inverse channel, [28] used the probabilistic error cancellation (PEC) protocol. The PEC approach allows for the application of an inverse noise channel to correct the errors introduced by the noise channel. The inverse channel is typically constructed by taking a probabilistic approach to undo the effects of the noise. The inverse noise model is by definition a non physical map and expressed as:

$$\Lambda^{-1}(\rho) = \gamma \bigcirc_{k \in \mathcal{K}} \left(\omega_k \rho - (1 - \omega_k) P_k \rho P_k^\dagger \right)$$

with $\gamma = \exp \left\{ \sum_{k \in \mathcal{K}} 2\lambda_k \right\}$ represents sampling overhead. For each $k \in \mathcal{K}$ the identity is sampled with probability ω_k or the Pauli P_k is applied. The number of non-identity Paulis are counted and the final Paulis are computed as the product of sampled terms. This process is repeated for all layers and the corresponding number of non-identity Paulis is recorded along with the sampling overhead. For layer $i = 1 \cdots L$, let the respective count be m_i and overhead be γ_i , then the final measurement outcome is multiplied by $\prod_{i=1}^L (-1)^{m_i} \gamma_i$ to get an unbiased expectation value on average.

For our purpose, we reiterate some useful properties of the operator $(aI \bullet I + bP_k \bullet P_k^\dagger)$ for any Pauli string P_k and $a, b \in \mathbb{R}$.

Proposition A.9. *Let P_k be any Pauli string. Then, $(aI \bullet I + bP_k \bullet P_k^\dagger)$ is linear.*

Sketch Proof of Proposition A.9. Let $A, B \in \mathcal{B}(\mathcal{H})$ be any two linear operators. Let $c, d \in \mathbb{R}$. Then,

$$\begin{aligned} & (aI \bullet I + bP_k \bullet P_k^\dagger)(cA + dB) \\ &= aI(cA + dB)I + bP_k(cA + dB)P_k^\dagger \\ &= caIAI + cbP_kAP_k^\dagger + daIBI + dbP_kBP_k^\dagger \\ &= c(aI \bullet I + bP_k \bullet P_k^\dagger)(A) + d(aI \bullet I + bP_k \bullet P_k^\dagger)(B). \end{aligned}$$

□

Proposition A.10. *Let P_{k_1}, P_{k_2} be any two Pauli strings. Let $a, b, c, d \in \mathbb{R}$ be any real values. Then, $aI \bullet I + bP_{k_1} \bullet P_{k_1}^\dagger$ and $cI \bullet I + dP_{k_2} \bullet P_{k_2}^\dagger$ commute.*

Sketch Proof of Proposition A.10. The proof is basically the same as the one used in [28]. First, we write $aI \bullet I + bP_{k_1} \bullet P_{k_1}^\dagger$ and $cI \bullet I + dP_{k_2} \bullet P_{k_2}^\dagger$ in their matrix form as the follows:

$$aI \otimes I + bP_{k_1} \otimes (P_{k_1}^\dagger)^T,$$

and

$$cI \otimes I + dP_{k_2} \otimes (P_{k_2}^\dagger)^T.$$

Then, by applying Proposition A.7, we obtain

$$\begin{aligned} & \left(aI \otimes I + bP_{k_1} \otimes (P_{k_1}^\dagger)^T \right) \left(cI \otimes I + dP_{k_2} \otimes (P_{k_2}^\dagger)^T \right) \\ &= ac(I \otimes I)(I \otimes I) + ad(I \otimes I) \left(P_{k_2} \otimes (P_{k_2}^\dagger)^T \right) \\ & \quad + bc \left(P_{k_1} \otimes (P_{k_1}^\dagger)^T \right) (I \otimes I) + bd \left(P_{k_1} \otimes (P_{k_1}^\dagger)^T \right) \left(P_{k_2} \otimes (P_{k_2}^\dagger)^T \right) \\ &= ac(I \otimes I)(I \otimes I) + ad(I \otimes I) \left(P_{k_2} \otimes (P_{k_2}^\dagger)^T \right) \\ & \quad + bc \left(P_{k_1} \otimes (P_{k_1}^\dagger)^T \right) (I \otimes I) + bd(-1)^{2(P_{k_1}, P_{k_2})_{sp}} \left(P_{k_2} \otimes (P_{k_2}^\dagger)^T \right) \left(P_{k_1} \otimes (P_{k_1}^\dagger)^T \right) \\ &= \left(cI \otimes I + dP_{k_2} \otimes (P_{k_2}^\dagger)^T \right) \left(aI \otimes I + bP_{k_1} \otimes (P_{k_1}^\dagger)^T \right). \end{aligned}$$

□

Remark A.11. The composite map $\Lambda^{-1} \circ \Lambda$ can be physical under specific conditions. Using the definitions of Λ and Λ^{-1} (defined in Equation (2) and Equation (3)), together with the commuting property (Proposition A.10), we can rewrite the composite as:

$$\Lambda^{-1} \circ \Lambda = \bigcirc_{k \in \{1, \dots, 4^c\}} \left(\frac{1 + e^{2(\sigma_k - \lambda_k)}}{2} I \cdot I + \frac{1 - e^{2(\sigma_k - \lambda_k)}}{2} P_k \cdot P_k \right),$$

where λ_k are noise parameters and σ_k are inverse noise parameters. It follows that $\Lambda^{-1} \circ \Lambda$ is implicitly a valid quantum channel if $\sigma_k \leq \lambda_k$ for all $k \in \mathcal{K}$. Unfortunately, verifying this inequality without prior knowledge of λ_k may require exponential effort.

A.2 Gradient

A.2.1 Derivation of $\frac{\partial \text{tr}(\mathcal{M}\mathcal{U}_R(\rho_t))}{\partial \sigma_{j,q}}$

Using the linearity of $\mathcal{U}_R$ and operator $\frac{\partial}{\partial \sigma_{j,q}}$, we can show that

$$\frac{\partial \text{tr}(\mathcal{M}\mathcal{U}_R(\rho_t))}{\partial \sigma_{j,q}} = \text{tr} \left(\mathcal{M} \frac{\partial \mathcal{U}_R(\rho_t)}{\partial \sigma_{j,q}} \right) = \text{tr} \left(\mathcal{M} \frac{\partial \mathcal{U}_{t,>j}(\Lambda_j^{-1}(\rho_{t,j}))}{\partial \sigma_{j,q}} \right) \quad (14)$$

where $\rho_{t,j} = \Lambda_j \circ \text{Ad}_{U_{j,b_j}} \circ \left(\bigcirc_{l=1}^{j-1} \Lambda_l^{-1} \circ \Lambda_l \circ \text{Ad}_{U_{l,b_l}} \right) (\rho_t)$ and $\mathcal{U}_{t,>j}(\bullet) = \left(\bigcirc_{l=j+1}^L \Lambda_l^{-1} \circ \Lambda_l \circ \text{Ad}_{U_{l,b_l}} \right) (\bullet)$.

Proposition A.12. *Let $c_1, c_2, c_3, c_4 \in \mathbb{R}$ be any constant real values. Let P be any complex matrix. Suppose $A : \mathbb{R} \rightarrow \mathbb{C}^{N \times N}$ is any function depending on the variable $x \in \mathbb{R}$. Then,*

$$\begin{aligned} & \frac{\partial}{\partial x} \left((c_1 + c_2)I \bullet I + (c_3 + c_4)P \bullet P^\dagger \right) (A(x)) \\ &= \left((c_1 + c_2)I \bullet I + (c_3 + c_4)P \bullet P^\dagger \right) \left(\frac{\partial}{\partial x} A(x) \right). \end{aligned}$$

Proof of Proposition A.12. Plugging in the matrix $A(x)$ into $((c_1 + c_2)I \bullet I + (c_3 + c_4)P \bullet P^\dagger)$ on the left-hand side, we obtain:

$$((c_1 + c_2)I \bullet I + (c_3 + c_4)P \bullet P^\dagger) (A(x)) = (c_1 + c_2)A(x) + (c_3 + c_4)PA(x)P^\dagger.$$

Taking the derivative with respect to x , we get,

$$\frac{\partial}{\partial x} \left((c_1 + c_2)A(x) + (c_3 + c_4)PA(x)P^\dagger \right) = (c_1 + c_2) \left(\frac{\partial A(x)}{\partial x} \right) + (c_3 + c_4)P \left(\frac{\partial A(x)}{\partial x} \right) P^\dagger,$$

which is exactly the same as the expression on the right-hand side. $\square$

Using Proposition A.12, it is straightforward to verify that operator $\mathcal{U}_{t,>j}$ commutes with $\frac{\partial}{\partial \sigma_{j,q}}$. Therefore, Equation (14) simplifies to

$$\text{tr} \left(\mathcal{M} \frac{\partial \mathcal{U}_{t,>j}(\Lambda_j^{-1}(\rho_{t,j}))}{\partial \sigma_{j,q}} \right) = \text{tr} \left(\mathcal{M} \mathcal{U}_{t,>j} \left(\frac{\partial \Lambda_j^{-1}(\rho_{t,j})}{\partial \sigma_{j,q}} \right) \right). \quad (15)$$

Next, applying the Pauli decomposition of $\rho_{t,j}$, we obtain

$$\rho_{t,j} = \sum_{g \in \{I, X, Y, Z\}^{\otimes n}} \alpha_{j,g} g. \quad (16)$$

Combining Equation (15), Equation (16), and Proposition A.9, we arrive at

$$\frac{\partial \text{tr}(\mathcal{M}\mathcal{U}_R(\rho_t))}{\partial \sigma_{j,q}} = \sum_{g \in \{I, X, Y, Z\}^{\otimes n}} \alpha_{j,g} \text{tr} \left(\mathcal{M} \mathcal{U}_{t,>j} \left(\frac{\partial \Lambda_j^{-1}(g)}{\partial \sigma_{j,q}} \right) \right). \quad (17)$$

To proceed, we need to compute the partial derivative, $\frac{\partial \Lambda_j^{-1}(g)}{\partial \sigma_{j,q}}$. First, let us simplify $\Lambda_j^{-1}(g)$ using the following proposition.

Proposition A.13. *If P, g are any two Pauli strings, then*

$$(aI \bullet I + bP \bullet P^\dagger) (g) = \begin{cases} (a + b)g & [P, g] = 0 \\ (a - b)g & \text{o.w.}, \end{cases}$$

for any $a, b \in \mathbb{R}$.

Proof of Proposition A.13. By definition, we have

$$(aI \bullet I + bP \bullet P^\dagger)(g) = a \cdot g + b \cdot PgP^\dagger$$

Considering two cases:

- **Case 1:** If $[P, g] = 0$, then

$$a \cdot g + b \cdot PgP^\dagger = a \cdot g + b \cdot gPP^\dagger = (a + b)g.$$

- **Case 2:** If $[P, g] \neq 0$, then by Proposition A.7 we know $gP = (-1)Pg$, which implies:

$$a \cdot g + b \cdot PgP = a \cdot g - b \cdot gPP^\dagger = (a - b)g.$$

□

Using Proposition A.13, we deduce, for all $g \in \{I, X, Y, Z\}^{\otimes n}$,

$$\begin{aligned} \Lambda_j^{-1}(g) &= \gamma_j \circ_{k \in \mathcal{K}_j} \left(q_{j,k} \cdot I \bullet I + (q_{j,k} - 1)P_k^{(j)} \bullet (P_k^{(j)})^\dagger \right)(g) \\ &= \gamma_j \cdot \left(\prod_{\substack{k \in \mathcal{K}_j \\ (P_k^{(j)}, g)_{sp} = 0}} (2q_{j,k} - 1) \cdot \prod_{\substack{k \in \mathcal{K}_j \\ (P_k^{(j)}, g)_{sp} = 1}} 1 \right) g \\ &= \gamma_j \cdot \prod_{\substack{k \in \mathcal{K}_j \\ (P_k^{(j)}, g)_{sp} = 0}} (2q_{j,k} - 1) g. \end{aligned}$$

Recall that $\gamma_j = \exp \left\{ 2 \sum_{k \in \mathcal{K}_j} \sigma_{j,k} \right\}$ and $q_{j,k} = \frac{1 + \exp \{-2\sigma_{j,k}\}}{2}$. Substituting the corresponding terms, we obtain

$$\begin{aligned} & \gamma_j \cdot \prod_{\substack{k \in \mathcal{K}_j \\ (P_k^{(j)}, g)_{sp} = 0}} (2q_{j,k} - 1) g \\ &= \exp \left\{ 2 \sum_{k \in \mathcal{K}_j} \sigma_{j,k} \right\} \cdot \prod_{\substack{k \in \mathcal{K}_j \\ (P_k^{(j)}, g)_{sp} = 0}} \left(2 \cdot \frac{1 + \exp \{-2\sigma_{j,k}\}}{2} - 1 \right) g \\ &= \exp \left\{ 2 \sum_{k \in \mathcal{K}_j} \sigma_{j,k} \right\} \cdot \prod_{\substack{k \in \mathcal{K}_j \\ (P_k^{(j)}, g)_{sp} = 0}} \exp \{-2\sigma_{j,k}\} \cdot g \\ &= \exp \left\{ 2 \sum_{k \in \mathcal{K}_j} \sigma_{j,k} \right\} \cdot \prod_{k \in \mathcal{K}_j} \exp \left\{ -2\sigma_{j,k} \cdot \mathbb{I} \left[(P_k^{(j)}, g)_{sp} = 0 \right] \right\} g \\ &= \exp \left\{ 2 \sum_{k \in \mathcal{K}_j} \sigma_{j,k} - \left(2 \sum_{k \in \mathcal{K}_j} \sigma_{j,k} \cdot \mathbb{I} \left[(P_k^{(j)}, g)_{sp} = 0 \right] \right) \right\} g. \end{aligned}$$

Here, $\mathbb{I}[\cdot]$ is the indicator function. By splitting $2 \sum_{k \in \mathcal{K}_j} \sigma_{j,k}$ into $2 \sum_{\substack{k \in \mathcal{K}_j \\ (P_k^{(j)}, g)_{sp}=0}} \sigma_{j,k}$ and $2 \sum_{\substack{k \in \mathcal{K}_j \\ (P_k^{(j)}, g)_{sp}=1}} \sigma_{j,k}$ and rearranging terms, we obtain

$$\begin{aligned} \Lambda_j^{-1}(g) &= \exp \left\{ 2 \sum_{\substack{k \in \mathcal{K}_j \\ (P_k^{(j)}, g)_{sp}=0}} \sigma_{j,k} - \sigma_{j,k} \right\} \exp \left\{ 2 \sum_{\substack{k \in \mathcal{K}_j \\ (P_k^{(j)}, g)_{sp}=1}} \sigma_{j,k} \right\} g \\ &= \exp \left\{ 2 \sum_{k \in \mathcal{K}_j} \sigma_{j,k} (P_k^{(j)}, g)_{sp} \right\} g. \end{aligned} \quad (18)$$

Next, applying chain rule to compute the partial derivative of $\Lambda_j^{-1}(g)$ w.r.t. $\sigma_{j,q}$, we get:

$$\begin{aligned} \frac{\partial \Lambda_j^{-1}(g)}{\partial \sigma_{j,q}} &= \frac{\partial}{\partial \sigma_{j,q}} \exp \left\{ 2 \sum_{k \in \mathcal{K}_j} \sigma_{j,k} \cdot (P_k^{(j)}, g)_{sp} \right\} g \\ &= \exp \left\{ 2 \sum_{k \in \mathcal{K}_j} \sigma_{j,k} \cdot (P_k^{(j)}, g)_{sp} \right\} \cdot 2 (P_q^{(j)}, g)_{sp} g. \end{aligned} \quad (19)$$

Combining Equation (17), Equation (18), and Equation (19), we have

$$\begin{aligned} &\frac{\partial \text{tr}(\mathcal{M}\mathcal{U}_R(\rho_t))}{\partial \sigma_{j,q}} \\ &= \sum_{g \in \{I, X, Y, Z\}^{\otimes n}} \alpha_{j,g} \text{tr} \left(\mathcal{M}\mathcal{U}_{t,>j} \left(\exp \left\{ 2 \sum_{k \in \mathcal{K}_j} \sigma_{j,k} \cdot (P_k^{(j)}, g)_{sp} \right\} \cdot 2 (P_q^{(j)}, g)_{sp} \cdot g \right) \right) \\ &= \sum_{g \in \{I, X, Y, Z\}^{\otimes n}} \alpha_{j,g} \cdot 2 (P_q^{(j)}, g)_{sp} \text{tr} \left(\mathcal{M}\mathcal{U}_{t,>j} \left(\exp \left\{ 2 \sum_{k \in \mathcal{K}_j} \sigma_{j,k} \cdot (P_k^{(j)}, g)_{sp} \right\} \cdot g \right) \right) \\ &= \sum_{\substack{g \in \{I, X, Y, Z\}^{\otimes n} \\ (P_q^{(j)}, g)_{sp}=1}} 2\alpha_{j,g} \text{tr} \left(\mathcal{M}\mathcal{U}_{t,>j} \left(\exp \left\{ 2 \sum_{k \in \mathcal{K}_j} \sigma_{j,k} \cdot (P_k^{(j)}, g)_{sp} \right\} \cdot g \right) \right) \\ &= \sum_{\substack{g \in \{I, X, Y, Z\}^{\otimes n} \\ (P_q^{(j)}, g)_{sp}=1}} 2\alpha_{j,g} \text{tr}(\mathcal{M}\mathcal{U}_{t,>j}(\Lambda_j^{-1}(g))) \\ &= 2 \text{tr} \left(\mathcal{M}\mathcal{U}_{t,>j} \left(\Lambda_j^{-1} \left(\sum_{\substack{g \in \{I, X, Y, Z\}^{\otimes n} \\ (P_q^{(j)}, g)_{sp}=1}} \alpha_{j,g} g \right) \right) \right). \end{aligned} \quad (21)$$

The expression presented in Equation (21) is impractical for direct use. To address this, we consider the following observation.

Proposition A.14. *If $P \in \{I, X, Y, Z\}^{\otimes n}$ is any Pauli string, then*

$$\sum_{\substack{g \in \{I, X, Y, Z\}^{\otimes n} \\ (P, g)_{sp}=1}} \alpha_{j,g} g = \frac{1}{2} (\rho_{t,j} - P \rho_{t,j} P^\dagger).$$

Proof of Proposition A.14. Again, applying Proposition A.7 the linearity of the adjoint operator, we obtain

$$P\rho_{t,j}P^\dagger = \sum_{\substack{g \in \{I,X,Y,Z\}^{\otimes n} \\ (P,g)_{sp}=0}} \alpha_{j,g}g - \sum_{\substack{g \in \{I,X,Y,Z\}^{\otimes n} \\ (P,g)_{sp}=1}} \alpha_{j,g}g. \quad (22)$$

Combining Equation (16) and Equation (22), we have

$$\begin{aligned} & \frac{1}{2} (\rho_{t,j} - P\rho_{t,j}P) \\ &= \frac{1}{2} \left(\left(\sum_{g \in \{I,X,Y,Z\}^{\otimes n}} \alpha_{j,g}g \right) - \left(\sum_{\substack{g \in \{I,X,Y,Z\}^{\otimes n} \\ (P,g)_{sp}=0}} \alpha_{j,g}g - \sum_{\substack{g \in \{I,X,Y,Z\}^{\otimes n} \\ (P,g)_{sp}=1}} \alpha_{j,g}g \right) \right) \\ &= \frac{1}{2} \left(\sum_{\substack{g \in \{I,X,Y,Z\}^{\otimes n} \\ (P,g)_{sp}=0}} \alpha_{j,g}g + \sum_{\substack{g \in \{I,X,Y,Z\}^{\otimes n} \\ (P,g)_{sp}=1}} \alpha_{j,g}g \right) \\ &\quad - \frac{1}{2} \left(\sum_{\substack{g \in \{I,X,Y,Z\}^{\otimes n} \\ (P,g)_{sp}=0}} \alpha_{j,g}g - \sum_{\substack{g \in \{I,X,Y,Z\}^{\otimes n} \\ (P,g)_{sp}=1}} \alpha_{j,g}g \right) \\ &= \frac{1}{2} \cdot 2 \cdot \left(\sum_{\substack{g \in \{I,X,Y,Z\}^{\otimes n} \\ (P,g)_{sp}=1}} \alpha_{j,g}g \right) = \sum_{\substack{g \in \{I,X,Y,Z\}^{\otimes n} \\ (P,g)_{sp}=1}} \alpha_{j,g}g \end{aligned}$$

□

By applying Proposition A.14, we are able to reformulate Equation (21) into a form analogous to the parameter-shift rule. Specifically, we obtain:

$$\frac{\partial \text{tr}(\mathcal{MU}_R(\rho_t))}{\partial \sigma_{j,q}} = 2 \cdot \frac{1}{2} \cdot \left[\text{tr}(\mathcal{MU}_{t,>j}(\Lambda_j^{-1}(\rho_{t,j}))) - \text{tr}(\mathcal{MU}_{t,>j}\Lambda_j^{-1}(P_q^{(j)}\rho_{t,j}(P_q^{(j)})^\dagger)) \right].$$

A.2.2 Derivation of $\frac{\partial \text{tr}(\mathcal{MU}_R(\rho_t))}{\partial \theta_j}$

We include both parameterized gate and constant gates in the PQC using Equation (1), therefore, we have to make sure that gradient calculation is exclusively performed for parameterized gates, i.e., model parameters only. As a result, we derive the expression for $\frac{\partial \text{tr}(\mathcal{MU}_R(\rho_t))}{\partial \theta_j}$ for all $1 \leq j \leq L$ such that $b_j = 1$ to ensure gradient calculation for appropriate model parameters.

Using the definition and linearity property (Proposition A.12), we express the partial derivative as follows:

$$\begin{aligned} \frac{\partial \text{tr}(\mathcal{MU}_R(\rho_t))}{\partial \theta_j} &= \text{tr} \left(\mathcal{MU}_{t,>j} \left(\Lambda_j^{-1} \circ \Lambda_j \left(\frac{\partial}{\partial \theta_j} \left(\text{Ad}_{U_{j,b_j}} \left(\rho_t^{(j-1)} \right) \right) \right) \right) \right) \\ &= \text{tr} \left(\mathcal{MU}_{t,>j} \left(\Lambda_j^{-1} \circ \Lambda_j \left(\frac{\partial}{\partial \theta_j} \left(\text{Ad}_{U_j(\theta_j)} \left(\rho_t^{(j-1)} \right) \right) \right) \right) \right) \\ &= \text{tr} \left(\mathcal{MU}_{t,>j} \left(\Lambda_j^{-1} \circ \Lambda_j \left(\frac{\partial}{\partial \theta_j} \left(U_j(\theta_j) \rho_t^{(j-1)} U_j^\dagger(\theta_j) \right) \right) \right) \right). \end{aligned}$$

where $\rho_t^{(j-1)} = \left(\bigcirc_{l=1}^{j-1} \Lambda_l^{-1} \circ \Lambda_l \circ \text{Ad}_{U_{l,b_l}} \right) (\rho_t)$. The partial derivative of $U_j(\theta_j) \rho_t^{(j-1)} U_j^\dagger(\theta_j)$ with respect to θ_l is given by

$$\begin{aligned} \frac{\partial U_j(\theta_j) \rho_t^{(j-1)} U_j^\dagger(\theta_j)}{\partial \theta_j} &= \frac{\partial U_j(\theta_j)}{\partial \theta_j} \rho_t^{(j-1)} U_j^\dagger(\theta_j) + U_j(\theta_j) \rho_t^{(j-1)} \frac{\partial U_j^\dagger(\theta_j)}{\partial \theta_j} \\ &= i G_j U_j(\theta_j) \rho_t^{(j-1)} U_j^\dagger(\theta_j) - i U_j(\theta_j) \rho_t^{(j-1)} U_j^\dagger(\theta_j) G_j \\ &= i \left[G_j, U_j(\theta_j) \rho_t^{(j-1)} U_j^\dagger(\theta_j) \right]. \end{aligned}$$

Applying the parameter-shift rule given in [17], we obtain:

$$\begin{aligned} &\frac{\partial U_j(\theta_j) \rho_t^{(j-1)} U_j^\dagger(\theta_j)}{\partial \theta_j} \\ &= \frac{1}{2} \left(U_j \left(\theta_j + \frac{\pi}{2} \right) \rho_t^{(j-1)} U_j^\dagger \left(\theta_j + \frac{\pi}{2} \right) \right) - \frac{1}{2} \left(U_j \left(\theta_j - \frac{\pi}{2} \right) \rho_t^{(j-1)} U_j^\dagger \left(\theta_j - \frac{\pi}{2} \right) \right). \end{aligned}$$

Altogether, $\frac{\partial \text{tr}(\mathcal{M}\mathcal{U}_R(\rho_t))}{\partial \theta_j}$ is given by:

$$\begin{aligned} &\frac{\partial \text{tr}(\mathcal{M}\mathcal{U}_R(\rho_t))}{\partial \theta_j} \\ &= \frac{1}{2} \text{tr} \left(\mathcal{M}\mathcal{U}_{t,>j} \left(\Lambda_j^{-1} \circ \Lambda_j \circ \text{Ad}_{U_j(\theta_j+\frac{\pi}{2})} \left(\rho_t^{(j-1)} \right) \right) \right) \\ &\quad - \frac{1}{2} \text{tr} \left(\mathcal{M}\mathcal{U}_{t,>j} \left(\Lambda_j^{-1} \circ \Lambda_j \circ \text{Ad}_{U_j(\theta_j-\frac{\pi}{2})} \left(\rho_t^{(j-1)} \right) \right) \right). \end{aligned}$$

A.2.3 The Gradient of Loss Function

Given the observations above, applying the chain rule, we obtain:

$$\frac{\partial (y_t - \text{tr}(\mathcal{M}\mathcal{U}_R(\rho_t)))^2}{\partial \sigma_{j,q}} = -4 (y_t - \text{tr}(\mathcal{M}\mathcal{U}_R(\rho_t))) \cdot \text{tr} \left(\mathcal{M}\mathcal{U}_{t,>j} \left(\Lambda_j^{-1} \circ \mathcal{U}_{P_q^{(j)}}(\rho_{t,j}) \right) \right), \quad (23)$$

where $\mathcal{U}_{P_q^{(j)}} = \frac{1}{2} \text{Ad}_I - \frac{1}{2} \text{Ad}_{P_q^{(j)}}$, and:

$$\frac{\partial (y_t - \text{tr}(\mathcal{M}\mathcal{U}_R(\rho_t)))^2}{\partial \theta_j} = -2 (y_t - \text{tr}(\mathcal{M}\mathcal{U}_R(\rho_t))) \cdot \text{tr} \left(\mathcal{M}\mathcal{U}_{t,>j} \left(\Lambda_j^{-1} \circ \Lambda_j \circ \mathcal{U}_{j,\frac{\pi}{2}} \left(\rho_t^{(j-1)} \right) \right) \right) \quad (24)$$

, where $\mathcal{U}_{j,a} = \frac{1}{2} \text{Ad}_{U_j(\theta_j+a)} - \frac{1}{2} \text{Ad}_{U_j(\theta_j-a)}$ for $a \in \mathbb{R}$.

A.3 Estimation of Gradient

A.3.1 Universal Estimation Algorithm

In this section, we introduce the universal estimation algorithm, a quantum-classical hybrid method designed to estimate the quantity

$$\text{tr} \left(\mathcal{M} \bigcirc_{l=1}^L \Lambda_l^{-1} \circ \text{Ad}_{W_l} \circ \Lambda_l \circ \text{Ad}_{U_l}(\rho) \right). \quad (25)$$

Here, $\mathcal{M}$ denotes a Hermitian observable, ρ is a density operator, W_l is a Pauli string, and U_l represents an arbitrary (non-Pauli) quantum operator. By construction, each composition $\text{Ad}_{W_l} \circ \Lambda_l \circ \text{Ad}_{U_l}$ defines a valid quantum channel, and can therefore be implemented directly on quantum hardware. The primary challenge lies in the realization of the non-physical inverse channels Λ_l^{-1} .

Fortunately, a simple classical preprocessing step enables their indirect implementation. Recall that each Λ_l^{-1} is defined as a linear combination of identity and conjugation by a Pauli operator, i.e., $q_{k,l} \bullet - (1 - q_{k,l}) P_k^{(l)} \bullet P_k^{(l)}$, where $q_{k,l} \in (0, 1)$. To realize Λ_l^{-1} , we sample from the Bernoulli distribution $\text{Bernoulli}(1 - q_{k,l})$. If the outcome is 0, we apply the identity operation; if the outcome is 1, we apply the Pauli string $P_k^{(l)}$ and record a classical multiplier of -1 . After measuring the quantum

state at the final stage of the algorithm, we multiply the final result by $(-1)^S$, where S is the number of such -1 recorded during the run. The detailed implementation is presented in Algorithm 2.

Algorithm 2 Universal Estimation Algorithm
(universal_estimator)

Input: $t \in [\mathcal{D}]$, Measurement $\mathcal{M}$, Parameters $\vec{\sigma} \in \mathcal{X}_{\text{sigma}}, \vec{\theta} \in \mathcal{X}_{\text{theta}}$.
Input: A set of constant Pauli strings $(W_1, \dots, W_L)$ {By assumption 3.2, all W_l are noiseless}
 $S \leftarrow 0$
 Prepare the quantum state ρ_t
for $l = 1$ to L **do**
 {Apply the inverse channel Λ_l^{-1} }
 $\rho_t \leftarrow \text{Ad}_{W_l} \circ \Lambda_l \circ \text{Ad}_{U_l}(\rho)$ {Evolve the Quantum State}
 {Here, for Ad_{U_l} , we look up the corresponding parameter θ in $\vec{\theta}$ when U_l is parameterized.}
 for $k \in \mathcal{K}_l$ **do**
 $q_{l,k} \leftarrow \frac{1 + \exp(-2\sigma_{l,k})}{2}$
 $b_{l,k} \sim \text{Bernoulli}(1 - q_{l,k})$
 if $b_{l,k} = 0$ **then**
 $\rho_t \leftarrow \rho$ {Apply Identity}
 else
 $\rho_t \leftarrow P_k^{(l)} \rho (P_k^{(l)})^\dagger$ {Apply $P_k^{(l)}$ }
 $S \leftarrow S + 1$
 end if
 end for
end for
 Measure the quantum state ρ on $\mathcal{M}$ and let $\mathbb{O}$ be the outcome of the measurement.
Return $\exp\left(2 \sum_{l \in [1, L], k \in \mathcal{K}_l} \sigma_{l,k}^{(t)}\right) \cdot (-1)^S \cdot \mathbb{O}$

Remark A.15. By construction, the universal estimation algorithm does not impose stringent requirements on the locality of the quantum operations involved (see Remark A.4). However, when the locality is too low—for instance, when $c = n$ —the structure of the inverse channel Λ_l^{-1} can become significantly more complex. In such cases, the cardinality of the corresponding index set $\mathcal{K}_l$ may grow exponentially with the number of qubits n , rendering the algorithm computationally intractable.

We now proceed to show that Algorithm 2 yields an unbiased estimator of the quantity $\text{tr}(\mathcal{M} \circ_{l=1}^L \Lambda_l^{-1} \circ \text{Ad}_{W_l} \circ \Lambda_l \circ \text{Ad}_{U_l}(\rho_t))$.

Lemma A.16. *universal_estimator* (Algorithm 2) provides an unbiased estimation of $\text{tr}(\mathcal{M} \circ_{l=1}^L \Lambda_l^{-1} \circ \text{Ad}_{W_l} \circ \Lambda_l \circ \text{Ad}_{U_l}(\rho_t))$, where $\mathcal{M}$ is a Hermitian and ρ_t is a density operator.

Proof of Lemma A.16. Let $(b_{l,k}) \in \{0, 1\}^{|\vec{\sigma}|}$ be a set of independent Bernoulli random variables with parameter $1 - q_{l,k} = 1 - \frac{1 + \exp(-2\sigma_{l,k})}{2}$. We denote by $\rho_{(b_{l,k})}$, the quantum state right before the measurement $\mathcal{M}$, which depends on the choice of $b_{l,k}$. From the algorithm definition, we note that $(-1)^S$ and $\rho_{(b_{l,k})}$ are random variables depending on $b_{l,k}$, and they can be re-written as follows:

$$(-1)^S = (-1)^{\sum_{l,k} b_{l,k}},$$

and

$$\rho_{(b_{l,k})} = \bigcirc_{l=1}^L \left(\bigcirc_{k \in \mathcal{K}_l} \text{Ad}_{(P_k^{(l)})^{b_{l,k}}} \right) \circ \text{Ad}_{W_l} \circ \Lambda_l \circ \text{Ad}_{U_l}(\rho_t). \quad (26)$$

Here, $\text{Ad}_A(\bullet) = A \bullet A^\dagger$ is the adjoint map and the bullet $\bullet$ stands as a placeholder. For clarity, we re-express Equation (26) by vectorizing it using the identity, $\text{vec}[ABC] = (A \otimes C^T) \text{vec}[B]$. We then obtain:

$$\text{vec}[\rho_{(b_{l,k})}] = \left(\prod_{l=L}^1 \left(\prod_{k \in \mathcal{K}_l} \left(P_k^{(l)} \otimes ((P_k^{(l)})^\dagger)^T \right)^{b_{l,k}} \right) \mathcal{N}_l \right) \text{vec}[\rho_t],$$

where $\mathcal{N}_l$ is defined by:

$$\left(\prod_{k \in \mathcal{K}_l} \left((\omega_{l,k}) I \otimes I + (1 - \omega_{l,k}) P_k^{(l)} \otimes ((P_k^{(l)})^\dagger)^T \right) \right) \left(W_l \otimes (W_l^\dagger)^T \right) \left(U_{l,b_l} \otimes (U_{l,b_l}^\dagger)^T \right).$$

Taking expectation over $b_{l,k}$, we arrive at:

$$\begin{aligned} & \mathbb{E}_{b_{l,k}} \left[(-1)^{\sum_{l,k} b_{l,k}} \cdot \text{vec} [\rho_{(b_{l,k})}] \right] \\ &= \mathbb{E}_{b_{l,k}} \left[(-1)^{\sum_{l,k} b_{l,k}} \cdot \left(\prod_{l=L}^1 \left(\prod_{k \in \mathcal{K}_l} \left(P_k^{(l)} \otimes ((P_k^{(l)})^\dagger)^T \right)^{b_{l,k}} \right) \mathcal{N}_l \right) \text{vec} [\rho_t] \right] \\ &= \mathbb{E}_{b_{l,k}} \left[\left(\prod_{l=L}^1 \left(\prod_{k \in \mathcal{K}_l} \left(-P_k^{(l)} \otimes ((P_k^{(l)})^\dagger)^T \right)^{b_{l,k}} \right) \mathcal{N}_l \right) \text{vec} [\rho_t] \right]. \end{aligned}$$

Since $b_{l,k}$ are independent, it follows that the terms $\left(-P_k^{(l)} \otimes ((P_k^{(l)})^\dagger)^T \right)^{b_{l,k}}$ are also independent. Moreover, Λ_l and ρ_t do not depend on $b_{l,k}$. Therefore, applying properties of expected value, we obtain:

$$\begin{aligned} & \mathbb{E}_{b_{l,k}} \left[(-1)^{\sum_{l,k} b_{l,k}} \cdot \text{vec} [\rho_{(b_{l,k})}] \right] \\ &= \left(\prod_{l=L}^1 \left(\prod_{k \in \mathcal{K}_l} \mathbb{E}_{b_{l,k}} \left[\left(-P_k^{(l)} \otimes ((P_k^{(l)})^\dagger)^T \right)^{b_{l,k}} \right] \right) \mathcal{N}_l \right) \text{vec} [\rho_t] \\ &= \left(\prod_{l=L}^1 \left(\prod_{k \in \mathcal{K}_l} \left(q_{l,k} \cdot \left(-P_k^{(l)} \otimes ((P_k^{(l)})^\dagger)^T \right)^0 \right. \right. \right. \\ & \quad \left. \left. \left. + (1 - q_{l,k}) \cdot \left(-P_k^{(l)} \otimes ((P_k^{(l)})^\dagger)^T \right)^1 \right) \right) \mathcal{N}_l \right) \text{vec} [\rho_t] \\ &= \left(\prod_{l=L}^1 \left(\prod_{k \in \mathcal{K}_l} \left(q_{l,k} \cdot I \otimes I - (1 - q_{l,k}) \cdot P_k^{(l)} \otimes ((P_k^{(l)})^\dagger)^T \right) \right) \mathcal{N}_l \right) \text{vec} [\rho_t]. \quad (27) \end{aligned}$$

Re-writing Equation (27), we express the expectation as:

$$\begin{aligned} & \mathbb{E}_{b_{l,k}} \left[(-1)^{\sum_{l,k} b_{l,k}} \cdot \rho_{(b_{l,k})} \right] \\ &= \left(\bigcirc_{l=1}^L \left(\bigcirc_{k \in \mathcal{K}_l} \left(q_{l,k} \text{Ad}_I - (1 - q_{l,k}) \text{Ad}_{P_k^{(l)}} \right) \right) \circ \text{Ad}_{W_l} \circ \Lambda_l \circ \text{Ad}_{U_l} \right) (\rho_t) \\ &= \left(\bigcirc_{l=1}^L \left(\frac{1}{\gamma_l} \cdot \gamma_l \bigcirc_{k \in \mathcal{K}_l} \left(q_{l,k} \text{Ad}_I - (1 - q_{l,k}) \text{Ad}_{P_k^{(l)}} \right) \right) \circ \text{Ad}_{W_l} \circ \Lambda_l \circ \text{Ad}_{U_l} \right) (\rho_t) \\ &= \left(\bigcirc_{l=1}^L \left(\frac{1}{\gamma_l} \Lambda_l^{-1} \right) \circ \text{Ad}_{W_l} \circ \Lambda_l \circ \text{Ad}_{U_l} \right) (\rho_t), \end{aligned}$$

where $\gamma_l = \exp \{ 2 \sum_{k \in \mathcal{K}_l} \sigma_{l,k} \}$. For a fixed $b_{l,k}$, one can easily verify that $\rho_{(b_{l,k})}$ is a density operator. Using the property of quantum measurement, we note that the expected measurement outcome $\odot$ is given by $\text{tr} (\mathcal{M} \rho_{(b_{l,k})})$. Combining these results, the expected value of the output of

the universal estimation algorithm is given by:

$$\begin{aligned}
& \mathbb{E} \left[\exp \left(2 \sum_{l \in [1, L], k \in \mathcal{K}_l} \sigma_{l,k}^{(t)} \right) \cdot (-1)^S \cdot \mathbb{O} \right] \\
&= \exp \left(2 \sum_{l \in [1, L], k \in \mathcal{K}_l} \sigma_{l,k}^{(t)} \right) \cdot \mathbb{E} \left[(-1)^{\sum_{l,k} b_{l,k}} \cdot \mathbb{O} \right] \\
&= \exp \left(2 \sum_{l \in [1, L], k \in \mathcal{K}_l} \sigma_{l,k}^{(t)} \right) \cdot \mathbb{E}_{b_{l,k}} \left[\mathbb{E} \left[(-1)^{\sum_{l,k} b_{l,k}} \cdot \mathbb{O} \middle| b_{l,k} \right] \right] \\
&= \exp \left(2 \sum_{l \in [1, L], k \in \mathcal{K}_l} \sigma_{l,k}^{(t)} \right) \cdot \mathbb{E}_{b_{l,k}} \left[(-1)^{\sum_{l,k} b_{l,k}} \cdot \text{tr}(\mathcal{M} \rho_{(b_{l,k})}) \right] \\
&= \exp \left(2 \sum_{l \in [1, L], k \in \mathcal{K}_l} \sigma_{l,k}^{(t)} \right) \cdot \text{tr} \left(\mathcal{M} \cdot \mathbb{E}_{b_{l,k}} \left[(-1)^{\sum_{l,k} b_{l,k}} \cdot \rho_{(b_{l,k})} \right] \right) \\
&= \left(\prod_{l=L}^1 \gamma_l \right) \cdot \text{tr} \left(\mathcal{M} \left(\bigcirc_{l=1}^L \left(\frac{1}{\gamma_l} \Lambda_l^{-1} \right) \circ \text{Ad}_{W_l} \circ \Lambda_l \circ \text{Ad}_{U_l} \right) (\rho_t) \right) \\
&= \text{tr} \left(\mathcal{M} \left(\bigcirc_{l=1}^L \left(\gamma_l \cdot \frac{1}{\gamma_l} \Lambda_l^{-1} \right) \circ \text{Ad}_{W_l} \circ \Lambda_l \circ \text{Ad}_{U_l} \right) (\rho_t) \right) \\
&= \text{tr} \left(\mathcal{M} \bigcirc_{l=1}^L \Lambda_l^{-1} \circ \text{Ad}_{W_l} \circ \Lambda_l \circ \text{Ad}_{U_l} (\rho_t) \right).
\end{aligned}$$

□

This algorithm serves as a foundational component in the estimation of partial derivatives, as discussed in the following section. There, we will show how the partial derivative estimation problem can be reduced to the trace estimation problem stated in Equation (25).

A.3.2 Estimation of Partial Derivative with the Universal Estimation Algorithm

To estimate the partial derivatives with respect to specific parameters, we first require efficient estimations of the trace terms: $\text{tr}(\mathcal{M} \mathcal{U}_R(\rho_t))$, $\text{tr}(\mathcal{M} \mathcal{U}_{t,>j}(\Lambda_j^{-1} \circ \text{Ad}_{P_q^{(j)}}(\rho_{t,j})))$, and $\text{tr}(\mathcal{M} \mathcal{U}_{t,>j}(\text{Ad}_{U(\theta_j+c)}(\rho_{t,j})))$. To facilitate their estimation, we first reformulate them. In particular, the term $\text{tr}(\mathcal{M} \mathcal{U}_R(\rho_t))$ can be expressed as

$$\text{tr} \left(\mathcal{M} \bigcirc_{l=1}^L \Lambda_l^{-1} \circ \text{Ad}_I \circ \Lambda_l \circ \text{Ad}_{U_{l,b_l}} (\rho_t) \right) \quad (28)$$

the term $\text{tr}(\mathcal{M} \mathcal{U}_{t,>j}(\Lambda_j^{-1} \circ \text{Ad}_{P_q^{(j)}}(\rho_{t,j})))$ as

$$\begin{aligned}
& \text{tr} \left(\mathcal{M} \left(\bigcirc_{l=j+1}^L \Lambda_l^{-1} \circ \text{Ad}_I \circ \Lambda_l \circ \text{Ad}_{U_{l,b_l}} \right) \circ \left(\Lambda_j^{-1} \circ \text{Ad}_{P_q^{(j)}} \circ \Lambda_j \circ \text{Ad}_{U_{j,b_j}} \right) \right. \\
& \quad \left. \circ \left(\bigcirc_{l=1}^{j-1} \Lambda_l^{-1} \circ \text{Ad}_I \circ \Lambda_l \circ \text{Ad}_{U_{l,b_l}} \right) (\rho_t) \right), \quad (29)
\end{aligned}$$

and the term $\text{tr}(\mathcal{M} \mathcal{U}_{t,>j}(\text{Ad}_{U(\theta_j+c)}(\rho_{t,j})))$ as

$$\begin{aligned}
& \text{tr} \left(\mathcal{M} \left(\bigcirc_{l=j+1}^L \Lambda_l^{-1} \circ \text{Ad}_I \circ \Lambda_l \circ \text{Ad}_{U_{l,b_l}} \right) \circ \left(\Lambda_j^{-1} \circ \text{Ad}_I \circ \Lambda_j \circ \text{Ad}_{U_{j,b_j}} \right) \right. \\
& \quad \left. \circ \left(\bigcirc_{l=1}^{j-1} \Lambda_l^{-1} \circ \text{Ad}_I \circ \Lambda_l \circ \text{Ad}_{U_{l,b_l}} \right) (\rho_t) \right). \quad (30)
\end{aligned}$$

Building on the transformations described above, we can efficiently estimate the three trace terms using the universal estimation algorithm presented in Algorithm 2. To estimate the partial derivatives defined in Equation (23) and Equation (24), integrating the observations discussed above, we introduce

two estimators: sigma_grad_est and theta_grad_est, explicitly detailed in Algorithm 3 and Algorithm 4, respectively.

Algorithm 3 Estimation of $\frac{\partial(y_t - \text{tr}(\mathcal{M}\mathcal{U}_R(\rho_t)))^2}{\partial\sigma_{j,q}}$ (sigma_grad_est)

Input: $t \in [|\mathcal{D}|]$, Measurement $\mathcal{M}$, Parameters $\vec{\sigma}^{(t)} \in \mathcal{X}_{\text{sigma}}$, $\vec{\theta}^{(t)} \in \mathcal{X}_{\text{theta}}$, true label y_t , $j \in [L]$, $q \in \mathcal{K}_j$, and positive integers $\tilde{N}_1^{(j,q)}, \tilde{N}_2^{(j,q)}$.
 $\mathcal{O}_t^{(1)}, \mathcal{O}_t^{(2,\sigma_{j,q})} \leftarrow 0, 0$
for $\alpha = 1$ to $\tilde{N}_1^{(j,q)}$ **do**
 $o_\alpha^{(1)} \leftarrow \text{universal_estimator}\left(t, \mathcal{M}, \vec{\sigma}, \vec{\theta}, (I, \dots, I)\right)$ {Algorithm 2}
 $\mathcal{O}_t^{(1)} \leftarrow \mathcal{O}_t^{(1)} + o_\alpha^{(1)}$
end for
for $\alpha = 1$ to $\tilde{N}_2^{(j,q)}$ **do**
 $b \sim \text{Bernoulli}(\frac{1}{2})$
if $b = 1$ **then**
 $o_\alpha^{(2,\sigma_{j,q})} \leftarrow \text{universal_estimator}\left(t, \mathcal{M}, \vec{\sigma}, \vec{\theta}, (I, \dots, I)\right)$
else
 $o_\alpha^{(2,\sigma_{j,q})} \leftarrow (-1) \cdot \text{universal_estimator}\left(\rho_t, \mathcal{M}, \vec{\sigma}, \vec{\theta}, (I, \dots, I, P_q^{(j)}, I, \dots, I)\right)$
{Here, $(I, \dots, I, P_q^{(j)}, I, \dots, I)$ is a L -tuple of identity except the one at (j) }
end if
 $\mathcal{O}_t^{(2,\sigma_{j,q})} \leftarrow \mathcal{O}_t^{(2,\sigma_{j,q})} + o_\alpha^{(2,\sigma_{j,q})}$
end for
Return $-4 \left(y_t - \frac{1}{\tilde{N}_1^{(j,q)}} \mathcal{O}_t^{(1)} \right) \cdot \frac{1}{\tilde{N}_2^{(j,q)}} \mathcal{O}_t^{(2,\sigma_{j,q})}$

Algorithm 4 Estimation of $\frac{\partial(y_t - \text{tr}(\mathcal{M}\mathcal{U}_R(\rho_t)))^2}{\partial\theta_j}$ (theta_grad_est)

Input: $t \in [|\mathcal{D}|]$, Measurement $\mathcal{M}$, Parameters $\vec{\sigma}^{(t)} \in \mathcal{X}_{\text{sigma}}$, $\vec{\theta}^{(t)} \in \mathcal{X}_{\text{theta}}$, true label y_t , $j \in [|\vec{\theta}|]$, and a positive integers $\tilde{N}_1^{(j)}, \tilde{N}_2^{(j)}$.
 $\mathcal{O}_t^{(1)}, \mathcal{O}_t^{(2,\theta_j)} \leftarrow 0, 0$
for $\alpha = 1$ to $\tilde{N}_1^{(j)}$ **do**
 $o_\alpha^{(1)} \leftarrow \text{universal_estimator}\left(t, \mathcal{M}, \vec{\sigma}, \vec{\theta}, (I, \dots, I)\right)$ {Algorithm 2}
 $\mathcal{O}_t^{(1)} \leftarrow \mathcal{O}_t^{(1)} + o_\alpha^{(1)}$
end for
for $\alpha = 1$ to $\tilde{N}_2^{(j)}$ **do**
 $b \sim \text{Bernoulli}(\frac{1}{2})$
if $b = 1$ **then**
 $o_\alpha^{(2,\theta_j)} \leftarrow \text{universal_estimator}\left(t, \mathcal{M}, \vec{\sigma}, \vec{\theta} + \frac{\pi}{2} \vec{e}_j, (I, \dots, I)\right)$
{Here, $\vec{e}_j \in \mathbb{R}^L$ is the vector that has 1 in the j^{th} component and 0 elsewhere.}
else
 $o_\alpha^{(2,\theta_j)} \leftarrow (-1) \cdot \text{universal_estimator}\left(\rho_t, \mathcal{M}, \vec{\sigma}, \vec{\theta} - \frac{\pi}{2} \vec{e}_j, (I, \dots, I)\right)$
end if
 $\mathcal{O}_t^{(2,\theta_j)} \leftarrow \mathcal{O}_t^{(2,\theta_j)} + o_\alpha^{(2,\theta_j)}$
end for
Return $-2 \left(y_t - \frac{1}{\tilde{N}_1^{(j)}} \mathcal{O}_t^{(1)} \right) \cdot \frac{1}{\tilde{N}_2^{(j)}} \mathcal{O}_t^{(2,\theta_j)}$

We proceed to demonstrate that both algorithms produce unbiased estimators, as stated in the following proposition.

Proposition A.17. *sigma_grad_est (Algorithm 3) and theta_grad_est (Algorithm 4) provide unbiased estimators of $\frac{\partial(y_t - \text{tr}(\mathcal{MU}_R(\rho_t)))^2}{\partial \sigma_{j,q}}$ and $\frac{\partial(y_t - \text{tr}(\mathcal{MU}_R(\rho_t)))^2}{\partial \theta_j}$, respectively.*

Proof of Proposition A.17. We will show that Algorithm 3 is an unbiased estimator of $\frac{\partial(y_t - \text{tr}(\mathcal{MU}_R(\rho_t)))^2}{\partial \sigma_{j,q}}$. Then, by applying a similar reasoning, one can demonstrate that the estimation produced by Algorithm 4 is also unbiased.

To start, let us show $\mathbb{E} \left[-4 \left(y_t - \frac{1}{\tilde{N}_1^{(j,q)}} \mathcal{O}_t^{(1)} \right) \right] = -4 (y_t - \text{tr}(\mathcal{MU}_R(\rho_t)))$. Let $o_\alpha^{(1)}$ and $o_\alpha^{(2, \sigma_{j,q})}$ be the random variables defined in Algorithm 3. Applying linearity of expectation, we obtain:

$$\mathbb{E} \left[-4 \left(y_t - \frac{1}{\tilde{N}_1^{(j,q)}} \mathcal{O}_t^{(1)} \right) \right] = -4 \left(y_t - \frac{1}{\tilde{N}_1^{(j,q)}} \sum_{\alpha=1}^{\tilde{N}_1^{(j,q)}} \mathbb{E} [o_\alpha^{(1)}] \right).$$

Applying Lemma A.16, we know $\mathbb{E} [o_\alpha^{(1)}] = \text{tr}(\mathcal{MU}_R(\rho_t))$. Hence,

$$\mathbb{E} \left[-4 \left(y_t - \frac{1}{\tilde{N}_1^{(j,q)}} \mathcal{O}_t^{(1)} \right) \right] = -4 (y_t - \text{tr}(\mathcal{MU}_R(\rho_t))).$$

Continuing, we show that,

$$\begin{aligned} & \mathbb{E} \left[\frac{1}{\tilde{N}_2^{(j,q)}} \cdot \mathcal{O}_t^{(2, \sigma_{j,q})} \right] \\ &= \frac{1}{2} \cdot \left(\text{tr} \left(\mathcal{MU}_{t, > j} \left(\Lambda_l^{-1} \left(\rho_t^{(j-1)} \right) \right) \right) - \text{tr} \left(\mathcal{MU}_{t, > j} \left(\Lambda_l^{-1} \left(P_q^{(j)} \rho_{t,j} (P_q^{(j)})^\dagger \right) \right) \right) \right). \end{aligned}$$

Applying again linearity of expectation and incorporating the definition of $o_\alpha^{(2, \sigma_{j,q})}$, we arrive at:

$$\begin{aligned} & \mathbb{E} \left[\frac{1}{\tilde{N}_2^{(j,q)}} \cdot \mathcal{O}_t^{(2, \sigma_{j,q})} \right] \\ &= \frac{1}{\tilde{N}_2^{(j,q)}} \sum_{\alpha=1}^{\tilde{N}_2^{(j,q)}} \mathbb{E} [o_\alpha^{(2, \sigma_{j,q})}] \\ &= \frac{1}{\tilde{N}_2^{(j,q)}} \sum_{\alpha=1}^{\tilde{N}_2^{(j,q)}} \left(\frac{1}{2} \mathbb{E} [\text{universal_estimator}(\rho_t, \mathcal{M}, \vec{\sigma}, \vec{\theta}, (I, \dots, I))] \right. \\ & \quad \left. - \frac{1}{2} \mathbb{E} [\text{universal_estimator}(\rho_t, \mathcal{M}, \vec{\sigma}, \vec{\theta}, (I, \dots, I, P_q^{(j)}, I, \dots, I))] \right). \end{aligned}$$

By Lemma A.16, we know that,

$$\begin{aligned}
& \mathbb{E} \left[\frac{1}{\tilde{N}_2^{(j,q)}} \cdot \mathcal{O}_t^{(2,\sigma_{j,q})} \right] \\
&= \frac{1}{\tilde{N}_2^{(j,q)}} \sum_{\alpha=1}^{\tilde{N}_2^{(j,q)}} \left(\frac{1}{2} \text{tr} \left(\mathcal{M}\mathcal{U}_{t,>j} \left(\Lambda_l^{-1} \left(\rho_t^{(j-1)} \right) \right) \right) - \frac{1}{2} \text{tr} \left(\mathcal{M}\mathcal{U}_{t,>j} \left(\Lambda_l^{-1} \left(P_q^{(j)} \rho_{t,j} (P_q^{(j)})^\dagger \right) \right) \right) \right) \\
&= \frac{1}{2} \left(\frac{1}{\tilde{N}_2^{(j,q)}} \sum_{\alpha=1}^{\tilde{N}_2^{(j,q)}} \text{tr} \left(\mathcal{M}\mathcal{U}_{t,>j} \left(\Lambda_l^{-1} \left(\rho_t^{(j-1)} \right) \right) \right) \right) \\
&\quad - \frac{1}{2} \left(\frac{1}{\tilde{N}_2^{(j,q)}} \sum_{\alpha=1}^{\tilde{N}_2^{(j,q)}} \text{tr} \left(\mathcal{M}\mathcal{U}_{t,>j} \left(\Lambda_l^{-1} \left(P_q^{(j)} \rho_{t,j} (P_q^{(j)})^\dagger \right) \right) \right) \right) \\
&= \frac{1}{2} \text{tr} \left(\mathcal{M}\mathcal{U}_{t,>j} \left(\Lambda_l^{-1} \left(\rho_t^{(j-1)} \right) \right) \right) - \frac{1}{2} \text{tr} \left(\mathcal{M}\mathcal{U}_{t,>j} \left(\Lambda_l^{-1} \left(P_q^{(j)} \rho_{t,j} (P_q^{(j)})^\dagger \right) \right) \right).
\end{aligned}$$

Since $-4(y_t - \mathcal{O}_t^{(1)})$ and $\mathcal{O}_t^{(2,\sigma_{j,q})}$ are independent, combining results above, we obtain:

$$\begin{aligned}
& \mathbb{E} \left[-4 \left(y_t - \frac{1}{\tilde{N}_1^{(j,q)}} \cdot \mathcal{O}_t^{(1)} \right) \cdot \left(\frac{1}{\tilde{N}_2^{(j,q)}} \cdot \mathcal{O}_t^{(2,\sigma_{j,q})} \right) \right] \\
&= \mathbb{E} \left[-4 \left(y_t - \frac{1}{\tilde{N}_1^{(j,q)}} \cdot \mathcal{O}_t^{(1)} \right) \right] \cdot \mathbb{E} \left[\frac{1}{\tilde{N}_2^{(j,q)}} \cdot \mathcal{O}_t^{(2,\sigma_{j,q})} \right] \\
&= -4 \left(y_t - \text{tr}(\mathcal{M}\mathcal{U}_R(\rho_t)) \right) \cdot \text{tr} \left(\mathcal{M}\mathcal{U}_{t,>j} \left(\Lambda_j^{-1} \circ \left(\frac{1}{2} \text{Ad}_I - \frac{1}{2} \text{Ad}_{P_q^{(j)}} \right) (\rho_{t,j}) \right) \right)
\end{aligned}$$

□

A.3.3 Gradient Estimation via Probabilistic Subsampling: Algorithm and Approximation Lemma

By the definition of Algorithm 1, each step involves estimating the gradient of $\mathcal{L}_t$ at $(\vec{\sigma}, \vec{\theta})$ for a given $t \in [|\mathcal{D}|]$. To this end, we introduce the following estimation algorithm, which leverages stochastic sampling to estimate gradient w.r.t. both the inverse noise parameters $\vec{\sigma}$ and PQC parameters $\vec{\theta}$.

Algorithm 5 Gradient Estimator (gradient_estimator)

Input: Positive Integer t , Measurement $\mathcal{M}$, Parameters $\vec{\sigma}^{(t)} \in \mathcal{X}_{\text{sigma}}$, $\vec{\theta}^{(t)} \in \mathcal{X}_{\text{theta}}$, true label y_t , an even positive integer $\tilde{N}$, and a binary vector $(\dots, p^{(l,k)}, \dots, p^{(l)}, \dots)$ of length $\left| \left(\vec{\sigma}, \vec{\theta} \right) \right|$.

Assumption: Assumption 3.2 - Assumption 3.4

```

for  $l \in [L]$  do
  for  $k \in \mathcal{K}_l$  do
    if  $p^{(l,k)} = 1$  then
       $\left( \tilde{\nabla} \mathcal{L}_t \left( \vec{\sigma}, \vec{\theta} \right) \right)_{l,k}^{\sigma} \leftarrow \text{sigma\_grad\_est}(t, \mathcal{M}, \vec{\sigma}, \vec{\theta}, y_t, l, k, \tilde{N}/2, \tilde{N}/2)$ 
    else
       $\left( \tilde{\nabla} \mathcal{L}_t \left( \vec{\sigma}, \vec{\theta} \right) \right)_{l,k}^{\sigma} \leftarrow 0$ 
    end if
  end for
end for
for  $l \in \left[ \left\lceil \left| \vec{\theta} \right| \right\rceil \right]$  do
  if  $p^{(l)} = 1$  then
     $\left( \tilde{\nabla} \mathcal{L}_t \left( \vec{\sigma}, \vec{\theta} \right) \right)_l^{\theta} \leftarrow \text{theta\_grad\_est}(t, \mathcal{M}, \vec{\sigma}, \vec{\theta}, y_t, l, \tilde{N}/2, \tilde{N}/2)$ 
  else
     $\left( \tilde{\nabla} \mathcal{L}_t \left( \vec{\sigma}, \vec{\theta} \right) \right)_l^{\theta} \leftarrow 0$ 
  end if
end for
Return  $\left( \dots, \left( \tilde{\nabla} \mathcal{L}_t \left( \vec{\sigma}, \vec{\theta} \right) \right)_{l,k}^{\sigma}, \dots, \left( \tilde{\nabla} \mathcal{L}_t \left( \vec{\sigma}, \vec{\theta} \right) \right)_l^{\theta}, \dots \right)$ 

```

In the following lemma, we demonstrate that Algorithm 5 produces an expected $(1/p)$ -approximation of the true gradient of $\mathcal{L}_t$ at $(\vec{\sigma}, \vec{\theta})$.

Lemma A.18. *Let $p \in \mathbb{Z}_{\geq 1}$. `gradient_estimator` (Algorithm 5) is a $(1/p)$ -approximation of the true gradient of $\mathcal{L}_t(\vec{\sigma}, \vec{\theta})$ in expectation.*

Proof of Lemma A.18. For simplicity, define $w = (\vec{\sigma}, \vec{\theta})$ and $|w|$ to be the length of w . Let $\vec{p} \sim (\text{Bernoulli}(1/p))^{|w|}$ be a random binary vector in $\{0, 1\}^{|w|}$. By the definition of Algorithm 5, we note that its output can be represented by a random vector $\vec{p} \odot \tilde{\nabla} \mathcal{L}_t(w)$, where $\odot$ is the element-wise vector multiplication and $\tilde{\nabla} \mathcal{L}_t(w)$ is the estimation of the true full gradient $\nabla \mathcal{L}_t(w)$ generated by Algorithm 3 and Algorithm 4.

Taking the expected value conditioned on $\vec{p}$, we get:

$$\mathbb{E} \left[\vec{p} \odot \tilde{\nabla} \mathcal{L}_t(w) \middle| \vec{p} \right] = \left(\dots, (\vec{p})_{\alpha} \cdot \mathbb{E} \left[\left(\tilde{\nabla} \mathcal{L}_t(w) \right)_{\alpha} \right], \dots \right).$$

Here, $\left(\tilde{\nabla} \mathcal{L}_t(w) \right)_{\alpha}$ is the α^{th} component of the vector $\tilde{\nabla} \mathcal{L}_t(w)$. From Proposition A.17, we know that $\tilde{\nabla} \mathcal{L}_t(w)$ is unbiased, meaning that,

$$\mathbb{E} \left[\vec{p} \odot \tilde{\nabla} \mathcal{L}_t(w) \middle| \vec{p} \right] = \left(\dots, (\vec{p})_{\alpha} \cdot (\nabla \mathcal{L}_t(w))_{\alpha}, \dots \right) = \vec{p} \odot \nabla \mathcal{L}_t(w).$$

Taking the full expectation,

$$\begin{aligned}
\mathbb{E} \left[\vec{p} \odot \tilde{\nabla} \mathcal{L}_t(w) \right] &= \sum_{\vec{p} \in \{0,1\}^{|w|}} \mathbb{E} \left[\vec{p} \odot \tilde{\nabla} \mathcal{L}_t(w) \middle| \vec{p} \right] \cdot \Pr[\vec{p}] \\
&= \sum_{\vec{p} \in \{0,1\}^{|w|}} (\vec{p} \odot \nabla \mathcal{L}_t(w)) \cdot \Pr[\vec{p}] \\
&= \left(\sum_{\vec{p} \in \{0,1\}^{|w|}} \Pr[\vec{p}] \cdot \vec{p} \right) \odot \nabla \mathcal{L}_t(w) \\
&= (\cdots, \mathbb{E}[(\vec{p})_\alpha], \cdots)^T \odot \nabla \mathcal{L}_t(w) \\
&= (\cdots, \Pr[(\vec{p})_\alpha = 0] \cdot 0 + \Pr[(\vec{p})_\alpha = 1] \cdot 1, \cdots)^T \odot \nabla \mathcal{L}_t(w) \\
&= (1/p, \cdots, 1/p)^T \odot \nabla \mathcal{L}_t(w) \\
&= \frac{1}{p} \cdot \nabla \mathcal{L}_t(w).
\end{aligned}$$

Here, $(\bullet)_\alpha$ is the α^{th} component of the input vector. □

Remark A.19. The scalar $p \in \mathbb{Z}_{\geq 1}$ must not exceed the total number of parameters (inverse noise parameters and PQC parameters) being optimized in order to guarantee that the expected density of the estimation given by $\mathbb{E} \left[\left\| \vec{p}_t \odot \tilde{\nabla} \mathcal{L}_t(\vec{\sigma}, \vec{\theta}) \right\|_0 \right]$ is not 0. In particular, if $p = 1$, $\vec{p}_t$ is, by definition, an all-ones vector. Consequently, `gradient_estimator` (Algorithm 5) provides an unbiased estimate of the full gradient of $\mathcal{L}_t$ at $(\vec{\sigma}, \vec{\theta})$.

A.4 Sample Complexity

A review of Algorithm 5 indicates that it employs a stochastic selection strategy to reduce the overall sample complexity. In the remainder of this section, we show that the total sample complexity decreases by a factor of $1/p$ in expectation. We begin by demonstrating that a mapping of the form $\mathcal{C} \left(y - \frac{1}{n} \sum_{i=1}^n \mathbb{X}_i \right) \left(\frac{1}{n} \sum_{i=n+1}^{2n} \mathbb{X}_i \right)$ satisfies the bounded difference property.

Proposition A.20. *Let $n \in \mathbb{Z}_{>0}$ be some positive number. Let $\mathbb{X}_1, \dots, \mathbb{X}_{2n} \in \{-1, 1\}$ be $2n$ independent random variables. Define $F : \{-1, 1\}^{2n} \rightarrow \mathbb{R}$ by $F(\mathbb{X}_1, \dots, \mathbb{X}_{2n}) = \mathcal{C} \left(y - \frac{1}{n} \sum_{i=1}^n \mathbb{X}_i \right) \left(\frac{1}{n} \sum_{i=n+1}^{2n} \mathbb{X}_i \right)$ for some constant $\mathcal{C} \in \mathbb{R}$ and $y \in \{-1, 1\}$. Then, for all $1 \leq j \leq 2n$,*

$$\max_{\mathbb{X}_j, \mathbb{X}'_j \in \{-1, 1\}} \left| F(\mathbb{X}_1, \dots, \mathbb{X}_j, \dots, \mathbb{X}_{2n}) - F(\mathbb{X}_1, \dots, \mathbb{X}'_j, \dots, \mathbb{X}_{2n}) \right| \leq \frac{|4\mathcal{C}|}{n}.$$

Proof of Proposition A.20. By the definition of F , we consider the following two cases:

- **Case 1:** Suppose $1 \leq j \leq n$. Then, for any $\mathbb{X}_j, \mathbb{X}'_j \in \{-1, 1\}$, we obtain:

$$\begin{aligned}
& \left| F(\mathbb{X}_1, \dots, \mathbb{X}_j, \dots, \mathbb{X}_n, \dots, \mathbb{X}_{2n}) - F(\mathbb{X}_1, \dots, \mathbb{X}'_j, \dots, \mathbb{X}_n, \dots, \mathbb{X}_{2n}) \right| \\
&= \left| \mathcal{C} \left(y - \frac{1}{n} \sum_{\substack{i=1 \\ i \neq j}}^n \mathbb{X}_i - \frac{\mathbb{X}_j}{n} \right) \left(\frac{1}{n} \sum_{i=n+1}^{2n} \mathbb{X}_i \right) \right. \\
&\quad \left. - \mathcal{C} \left(y - \frac{1}{n} \sum_{\substack{i=1 \\ i \neq j}}^n \mathbb{X}_i - \frac{\mathbb{X}'_j}{n} \right) \left(\frac{1}{n} \sum_{i=n+1}^{2n} \mathbb{X}_i \right) \right| \\
&= \frac{|\mathcal{C}|}{n} \cdot |\mathbb{X}'_j - \mathbb{X}_j| \cdot \left| \left(\frac{1}{n} \sum_{i=n+1}^{2n} \mathbb{X}_i \right) \right| \\
&\leq \frac{|\mathcal{C}|}{n} \cdot |\mathbb{X}'_j - \mathbb{X}_j| \cdot \frac{1}{n} \sum_{i=n+1}^{2n} |\mathbb{X}_i| \quad (\text{triangle inequality}) \\
&\leq \frac{|\mathcal{C}| \cdot 2}{n^2} \cdot \sum_{i=n+1}^{2n} 1 = \frac{2|\mathcal{C}|}{n}.
\end{aligned}$$

- **Case 2:** Suppose $n+1 \leq j \leq 2n$. Again, for any $\mathbb{X}_j, \mathbb{X}'_j \in \{-1, 1\}$, we obtain:

$$\begin{aligned}
& \left| F(\mathbb{X}_1, \dots, \mathbb{X}_n, \dots, \mathbb{X}_j, \dots, \mathbb{X}_{2n}) - F(\mathbb{X}_1, \dots, \mathbb{X}_n, \dots, \mathbb{X}'_j, \dots, \mathbb{X}_{2n}) \right| \\
&= \left| \mathcal{C} \left(y - \frac{1}{n} \sum_{i=1}^n \mathbb{X}_i \right) \left(\frac{1}{n} \sum_{\substack{i=n+1 \\ i \neq j}}^{2n} \mathbb{X}_i - \frac{\mathbb{X}_j}{n} \right) \right. \\
&\quad \left. - \mathcal{C} \left(y - \frac{1}{n} \sum_{i=1}^n \mathbb{X}_i \right) \left(\frac{1}{n} \sum_{\substack{i=n+1 \\ i \neq j}}^{2n} \mathbb{X}_i - \frac{\mathbb{X}'_j}{n} \right) \right| \\
&= |\mathcal{C}| \cdot \left| y - \frac{1}{n} \sum_{i=1}^n \mathbb{X}_i \right| \cdot \left| \frac{\mathbb{X}_j}{n} - \frac{\mathbb{X}'_j}{n} \right| \\
&\leq |\mathcal{C}| \cdot \left(|y| + \frac{1}{n} \sum_{i=1}^n |\mathbb{X}_i| \right) \cdot \frac{2}{n} \quad (\text{triangle inequality}) \\
&\leq |\mathcal{C}| \cdot (1+1) \cdot \frac{2}{n} = \frac{4|\mathcal{C}|}{n}.
\end{aligned}$$

Together, we get

$$\max_{\mathbb{X}_j, \mathbb{X}'_j \in \{-1, 1\}} \left| F(\mathbb{X}_1, \dots, \mathbb{X}_j, \dots, \mathbb{X}_{2n}) - F(\mathbb{X}_1, \dots, \mathbb{X}'_j, \dots, \mathbb{X}_{2n}) \right| \leq \max \left\{ \frac{2|\mathcal{C}|}{n}, \frac{4|\mathcal{C}|}{n} \right\} = \frac{4|\mathcal{C}|}{n}.$$

□

A.4.1 Probabilistic Bound on Estimation Error for Gradient Estimators

In the following lemma, we establish a probabilistic bound for estimating a partial derivative of $\mathcal{L}_t$ with deviation $\varepsilon > 0$.

Lemma A.21. Let $\widetilde{\partial}_{\sigma_{j,q}} \mathcal{L}_t(\vec{\sigma}, \vec{\theta})$ and $\widetilde{\partial}_{\theta_j} \mathcal{L}_t(\vec{\sigma}, \vec{\theta})$ denote the estimated partial derivative of $\mathcal{L}_t$ with respect to $\sigma_{j,q}$ and θ_j at $(\vec{\sigma}, \vec{\theta})$, which are generated by Algorithm 3 or Algorithm 4, respectively. Similarly, let $\partial_{\sigma_{j,q}} \mathcal{L}_t(\vec{\sigma}, \vec{\theta})$ and $\partial_{\theta_j} \mathcal{L}_t(\vec{\sigma}, \vec{\theta})$ denote the corresponding true partial

derivatives. Then, for all $\varepsilon > 0$, the probabilities $\Pr \left[\left| \widetilde{\partial_{\sigma_{j,q}} \mathcal{L}_t}(\vec{\sigma}, \vec{\theta}) - \partial_{\sigma_{j,q}} \mathcal{L}_t(\vec{\sigma}, \vec{\theta}) \right| \geq \varepsilon \middle| t \right]$ and $\Pr \left[\left| \widetilde{\partial_{\theta_j} \mathcal{L}_t}(\vec{\sigma}, \vec{\theta}) - \partial_{\theta_j} \mathcal{L}_t(\vec{\sigma}, \vec{\theta}) \right| \geq \varepsilon \middle| t \right]$ are both upper-bounded by $2 \exp \left\{ \frac{-2(\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \varepsilon^2 \tilde{N}}{32^2} \right\}$, where $\Gamma_{\vec{\sigma}}^{-1} = \exp \left\{ -2 \sum_{l=1}^L \sum_{k \in \mathcal{K}_l} \sigma_{l,k} \right\}$ and $\tilde{N}$ is a positive even integer.

Proof of Lemma A.21. We begin by establishing the bound for $\Pr \left[\left| \widetilde{\partial_{\sigma_{j,q}} \mathcal{L}_t}(\vec{\sigma}, \vec{\theta}) - \partial_{\sigma_{j,q}} \mathcal{L}_t(\vec{\sigma}, \vec{\theta}) \right| \geq \varepsilon \middle| t \right]$. Since $\Gamma_{\vec{\sigma}}^{-1} > 0$, we have

$$\begin{aligned} \left| \widetilde{\partial_{\sigma_{j,q}} \mathcal{L}_t}(\vec{\sigma}, \vec{\theta}) - \mathbb{E} \left[\widetilde{\partial_{\sigma_{j,q}} \mathcal{L}_t}(\vec{\sigma}, \vec{\theta}) \right] \right| &\geq \varepsilon \\ \iff (\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \left| \widetilde{\partial_{\sigma_{j,q}} \mathcal{L}_t}(\vec{\sigma}, \vec{\theta}) - \mathbb{E} \left[\widetilde{\partial_{\sigma_{j,q}} \mathcal{L}_t}(\vec{\sigma}, \vec{\theta}) \right] \right| &\geq (\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \varepsilon. \end{aligned}$$

Thus, the probability can be written as:

$$\begin{aligned} &\Pr \left[\left| \widetilde{\partial_{\sigma_{j,q}} \mathcal{L}_t}(\vec{\sigma}, \vec{\theta}) - \mathbb{E} \left[\widetilde{\partial_{\sigma_{j,q}} \mathcal{L}_t}(\vec{\sigma}, \vec{\theta}) \right] \right| \geq \varepsilon \middle| t \right] \\ &= \Pr \left[(\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \left| \widetilde{\partial_{\sigma_{j,q}} \mathcal{L}_t}(\vec{\sigma}, \vec{\theta}) - \mathbb{E} \left[\widetilde{\partial_{\sigma_{j,q}} \mathcal{L}_t}(\vec{\sigma}, \vec{\theta}) \right] \right| \geq (\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \varepsilon \middle| t \right] \\ &= \Pr \left[\left| (\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \widetilde{\partial_{\sigma_{j,q}} \mathcal{L}_t}(\vec{\sigma}, \vec{\theta}) - (\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \mathbb{E} \left[\widetilde{\partial_{\sigma_{j,q}} \mathcal{L}_t}(\vec{\sigma}, \vec{\theta}) \right] \right| \geq (\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \varepsilon \middle| t \right]. \end{aligned}$$

Next, we aim to show that the random variable $(\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \widetilde{\partial_{\sigma_{j,q}} \mathcal{L}_t}(\vec{\sigma}, \vec{\theta})$ satisfies the bounded difference property. From the definitions in Algorithm 3, we observe that $\widetilde{\partial_{\sigma_{j,q}} \mathcal{L}_t}(\vec{\sigma}, \vec{\theta})$ can be re-written as a random map from $\{-\Gamma_{\vec{\sigma}}, \Gamma_{\vec{\sigma}}\}^{\tilde{N}}$ to $\mathbb{R}$ defined as the following:

$$\begin{aligned} &\left(\widetilde{\partial_{\sigma_{j,q}} \mathcal{L}_t}(\vec{\sigma}, \vec{\theta}) \right) \left(o_1^{(1)}, \dots, o_{\tilde{N}/2}^{(1)}, o_1^{(2, \sigma_{j,q})}, \dots, o_{\tilde{N}/2}^{(2, \sigma_{j,q})} \right) \\ &= (-4) \left(y_t - \frac{1}{\tilde{N}/2} \sum_{\beta=1}^{\tilde{N}/2} o_{\beta}^{(1)} \right) \left(\frac{1}{\tilde{N}/2} \sum_{\beta=\tilde{N}/2+1}^{\tilde{N}} o_{\beta}^{(2, \sigma_{j,q})} \right). \end{aligned}$$

From the definitions in Algorithm 2, we observe that both $o_{\beta}^{(1)}$ and $o_{\beta}^{(2, \sigma_{j,q})}$ are in the set $\{-\Gamma_{\vec{\sigma}}, \Gamma_{\vec{\sigma}}\}$. This implies that the random variables $\Gamma_{\vec{\sigma}}^{-1} \cdot o_{\beta}^{(1)}$ and $\Gamma_{\vec{\sigma}}^{-1} \cdot o_{\beta}^{(2, \sigma_{j,q})}$ lie within $\{-1, 1\}$. Consequently, $(\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \left(\widetilde{\partial_{\sigma_{j,q}} \mathcal{L}_t}(\vec{\sigma}, \vec{\theta}) \right)$ becomes a random map from $\{-1, 1\}^{\tilde{N}}$ to $\mathbb{R}$. For convenience, let us re-express $(\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \left(\widetilde{\partial_{\sigma_{j,q}} \mathcal{L}_t}(\vec{\sigma}, \vec{\theta}) \right)$ as a mapping $F : \{-1, 1\}^{\tilde{N}} \rightarrow \mathbb{R}$, defined as follows:

$$F \left(\mathbb{Y}_1^{(F)}, \dots, \mathbb{Y}_{\tilde{N}/2}^{(F)}, \dots, \mathbb{Y}_{\tilde{N}}^{(F)} \right) = (-4) \left(\Gamma_{\vec{\sigma}}^{-1} \cdot y_t - \frac{1}{\tilde{N}/2} \sum_{\beta=1}^{\tilde{N}/2} \mathbb{Y}_{\beta}^{(F)} \right) \left(\frac{1}{\tilde{N}/2} \sum_{\beta=\tilde{N}/2+1}^{\tilde{N}} \mathbb{Y}_{\beta}^{(F)} \right).$$

where $\mathbb{Y}_{\beta}^{(F)} = \Gamma_{\vec{\sigma}}^{-1} \cdot o_{\beta}^{(1)}$ for $1 \leq \beta \in \tilde{N}/2$, and $\mathbb{Y}_{\beta}^{(F)} = \Gamma_{\vec{\sigma}}^{-1} \cdot o_{\beta}^{(2, \sigma_{j,q})}$ for $\tilde{N}/2 + 1 \leq \beta \in \tilde{N}$. By applying Proposition A.20, we observe the following bound on the difference when changing a single input:

$$\begin{aligned} &\max_{\mathbb{Y}_{\zeta}^{(F)}, (\mathbb{Y}_{\zeta}^{(F)})' \in \{-1, 1\}} \left| F \left(\mathbb{Y}_1^{(F)}, \dots, \mathbb{Y}_{\zeta}^{(F)}, \dots, \mathbb{Y}_{\tilde{N}}^{(F)} \right) - F \left(\mathbb{Y}_1^{(F)}, \dots, (\mathbb{Y}_{\zeta}^{(F)})', \dots, \mathbb{Y}_{\tilde{N}}^{(F)} \right) \right| \\ &\leq \frac{16}{\tilde{N}/2} = \frac{32}{\tilde{N}}. \end{aligned}$$

This shows that changing a single input alters the value of $(\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \left(\widetilde{\partial_{\sigma_{j,q}} \mathcal{L}_t}(\vec{\sigma}, \vec{\theta}) \right)$ by at most $\frac{32}{\tilde{N}}$, meaning that $(\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \left(\widetilde{\partial_{\sigma_{j,q}} \mathcal{L}_t}(\vec{\sigma}, \vec{\theta}) \right)$ satisfies the bounded differences property. As a result,

by applying McDiarmid's inequality [16], we know that for any $\varepsilon > 0$, we deduce,

$$\begin{aligned} & \Pr \left[\left| (\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \widetilde{\partial_{\sigma_{j,q}} \mathcal{L}_t}(\vec{\sigma}, \vec{\theta}) - (\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \mathbb{E} \left[\widetilde{\partial_{\sigma_{j,q}} \mathcal{L}_t}(\vec{\sigma}, \vec{\theta}) \right] \right| \geq (\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \varepsilon \right] \\ & \leq 2 \exp \left\{ -2 \left((\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \varepsilon \right)^2 / \left(\sum_{i=1}^{\tilde{N}} \left(32/\tilde{N} \right)^2 \right) \right\} \\ & = 2 \exp \left\{ \frac{-2 \left((\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \varepsilon \right)^2 \tilde{N}}{32^2} \right\}. \end{aligned}$$

By Proposition A.17, we note that

$$\mathbb{E} \left[\widetilde{\partial_{\sigma_{j,q}} \mathcal{L}_t}(\vec{\sigma}, \vec{\theta}) \right] = \partial_{\sigma_{j,q}} \mathcal{L}_t(\vec{\sigma}, \vec{\theta}).$$

Therefore, we obtain the following probability bound:

$$\Pr \left[\left| \widetilde{\partial_{\sigma_{j,q}} \mathcal{L}_t}(\vec{\sigma}, \vec{\theta}) - \partial_{\sigma_{j,q}} \mathcal{L}_t(\vec{\sigma}, \vec{\theta}) \right| \geq \varepsilon \right] \leq 2 \exp \left\{ \frac{-2 \left((\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \varepsilon \right)^2 \tilde{N}}{32^2} \right\}.$$

Using similar arguments as above, we identify $(\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \widetilde{\partial_{\theta_j} \mathcal{L}_t}(\vec{\sigma}, \vec{\theta})$ as a random mapping from $\{-1, 1\}^{\tilde{N}}$ to $\mathbb{R}$. From Proposition A.20, we observe that changing a single input alters the value of $(\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \widetilde{\partial_{\theta_j} \mathcal{L}_t}(\vec{\sigma}, \vec{\theta})$ by at most $\frac{16}{\tilde{N}} < \frac{32}{\tilde{N}}$. Again, by applying McDiarmid's inequality and Proposition A.17, we obtain:

$$\Pr \left[\left| \widetilde{\partial_{\theta_j} \mathcal{L}_t}(\vec{\sigma}, \vec{\theta}) - \partial_{\theta_j} \mathcal{L}_t(\vec{\sigma}, \vec{\theta}) \right| \geq \varepsilon \right] \leq 2 \exp \left\{ \frac{-2 \left((\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \varepsilon \right)^2 \tilde{N}}{32^2} \right\}.$$

□

Now returning to the context of Algorithm 1, in each update round, exactly one $\vec{p}$ is sampled and passed into Algorithm 5. For simplicity, we define $\vec{p} \odot \tilde{\nabla} \mathcal{L}_t(\vec{\sigma}, \vec{\theta})$ as the vector of estimated partial derivatives generated by Algorithm 5. Given a particular update round t and $\vec{p}$, the α^{th} entry of $\vec{p} \odot \tilde{\nabla} \mathcal{L}_t(\vec{\sigma}, \vec{\theta}) \neq 0$ only when the α^{th} entry in $\vec{p}$ is 1. Thus, to quantify the estimation error, it suffices to measure the deviation between the non-zero entries of $\vec{p} \odot \tilde{\nabla} \mathcal{L}_t(\vec{\sigma}, \vec{\theta})$ and their corresponding true partial derivative.

Let $\vec{p} \odot \nabla \mathcal{L}_t(\vec{\sigma}, \vec{\theta})$ denote the vector of the corresponding true partial derivatives. Here, $\nabla \mathcal{L}_t(\vec{\sigma}, \vec{\theta})$ is the full true gradient of $\mathcal{L}_t$ at $(\vec{\sigma}, \vec{\theta})$, and the value of each entry in $\vec{p} \odot \nabla \mathcal{L}_t(\vec{\sigma}, \vec{\theta})$ is determined according to the entries of $\vec{p}$. In the following lemma, we establish a probabilistic bound for $\left\| \vec{p} \odot \tilde{\nabla} \mathcal{L}_t(\vec{\sigma}, \vec{\theta}) - \vec{p} \odot \nabla \mathcal{L}_t(\vec{\sigma}, \vec{\theta}) \right\|_2$ with deviation $\varepsilon > 0$ conditioned on $\vec{p}$ and t .

Lemma A.22. *Let $p \leq |(\vec{\sigma}, \vec{\theta})|$ be a positive integer. For all $\varepsilon > 0$ and $\tilde{N} \in (2\mathbb{Z})_{>0}$,*

$$\begin{aligned} & \Pr \left[\left\| \vec{p} \odot \tilde{\nabla} \mathcal{L}_t(\vec{\sigma}, \vec{\theta}) - \vec{p} \odot \nabla \mathcal{L}_t(\vec{\sigma}, \vec{\theta}) \right\|_2 \geq \varepsilon \mid t, \vec{p} \right] \\ & \leq \|\vec{p}\|_0 \cdot 2 \exp \left\{ \frac{-2 \left((\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \varepsilon \right)^2 \tilde{N}}{32^2} \right\}, \end{aligned}$$

where $\Gamma_{\vec{\sigma}}^{-1} = \exp \left\{ -2 \sum_{l=1}^L \sum_{k \in \mathcal{K}_l} \sigma_{l,k} \right\}$ and $\vec{p} \in \{0, 1\}^{(\vec{\sigma}, \vec{\theta})}$.

Proof of Lemma A.22. For simplicity and clarity, we now denote $(\vec{\sigma}, \vec{\theta})$ by w and the length of w by $|w|$.

First, using the fact $\|x\|_\infty \leq \|x\|_2$, we note that $\|x\|_\infty \geq \varepsilon \implies \|x\|_2 \geq \varepsilon$, which implies that:

$$\begin{aligned} & \Pr \left[\left\| \vec{p} \odot \tilde{\nabla} \mathcal{L}_t(w) - \vec{p} \odot \nabla \mathcal{L}_t(w) \right\|_2 \geq \varepsilon \mid t, \vec{p} \right] \\ & \leq \Pr \left[\left\| \vec{p} \odot \tilde{\nabla} \mathcal{L}_t(w) - \vec{p} \odot \nabla \mathcal{L}_t(w) \right\|_\infty \geq \varepsilon \mid t, \vec{p} \right] \\ & = \Pr \left[\max_{\substack{\alpha=1, \dots, |w| \\ (\vec{p})_\alpha \neq 0}} \left| \left(\tilde{\nabla} \mathcal{L}_t(w) \right)_\alpha - \left(\nabla \mathcal{L}_t(w) \right)_\alpha \right| \geq \varepsilon \mid t, \vec{p} \right] \\ & \leq \Pr \left[\exists \alpha \in \mathcal{I}_{\vec{p}} : \left| \left(\tilde{\nabla} \mathcal{L}_t(w) \right)_\alpha - \left(\nabla \mathcal{L}_t(w) \right)_\alpha \right| \geq \varepsilon \mid t, \vec{p} \right], \end{aligned}$$

where $\mathcal{I}_{\vec{p}} = \{\alpha \in [|w|] \mid (\vec{p})_\alpha \neq 0\}$. Applying the union bound, we have:

$$\begin{aligned} & \Pr \left[\exists \alpha \in \mathcal{I}_{\vec{p}} : \left| \left(\tilde{\nabla} \mathcal{L}_t(w) \right)_\alpha - \left(\nabla \mathcal{L}_t(w) \right)_\alpha \right| \geq \varepsilon \mid t, \vec{p} \right] \\ & \leq \sum_{\alpha \in \mathcal{I}_{\vec{p}}} \Pr \left[\left| \left(\tilde{\nabla} \mathcal{L}_t(w) \right)_\alpha - \left(\nabla \mathcal{L}_t(w) \right)_\alpha \right| \geq \varepsilon \mid t \right] \end{aligned}$$

Then, since $\left(\tilde{\nabla} \mathcal{L}_t(w) \right)_\alpha$ are generated by either Algorithm 3 and Algorithm 4, by applying the Lemma A.21, we derive:

$$\Pr \left[\left| \left(\tilde{\nabla} \mathcal{L}_t(w) \right)_\alpha - \left(\nabla \mathcal{L}_t(w) \right)_\alpha \right| \geq \varepsilon \mid t \right] \leq 2 \exp \left\{ \frac{-2 \left((\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \varepsilon \right)^2 \tilde{N}}{32^2} \right\}.$$

Combining all of the results above, we obtain:

$$\begin{aligned} \Pr \left[\left\| \vec{p} \odot \tilde{\nabla} \mathcal{L}_t(w) - \vec{p} \odot \nabla \mathcal{L}_t(w) \right\|_2 \geq \varepsilon \mid t, \vec{p} \right] & \leq \sum_{\alpha \in \mathcal{I}_{\vec{p}}} 2 \exp \left\{ \frac{-2 \left((\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \varepsilon \right)^2 \tilde{N}}{32^2} \right\} \\ & = 2 |\mathcal{I}_{\vec{p}}| \cdot \exp \left\{ \frac{-2 \left((\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \varepsilon \right)^2 \tilde{N}}{32^2} \right\}. \end{aligned}$$

By definition of $\mathcal{I}_{\vec{p}}$, we note that $|\mathcal{I}_{\vec{p}}| = \|\vec{p}\|_0$. Therefore,

$$\Pr \left[\left\| \vec{p} \odot \tilde{\nabla} \mathcal{L}_t(w) - \vec{p} \odot \nabla \mathcal{L}_t(w) \right\|_2 \geq \varepsilon \mid t, \vec{p} \right] \leq 2 \|\vec{p}\|_0 \cdot \exp \left\{ \frac{-2 \left((\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \varepsilon \right)^2 \tilde{N}}{32^2} \right\}. \quad (31)$$

□

A.4.2 Expected Bound on the Total Number of Measurements

In the following theorem, we present an expected bound for the total number of measurements required to ensure that the deviation $\left\| \vec{p} \odot \tilde{\nabla} \mathcal{L}_t(\vec{\sigma}, \vec{\theta}) - \vec{p} \odot \nabla \mathcal{L}_t(\vec{\sigma}, \vec{\theta}) \right\|_2$ remains within ε with probability at least $1 - \delta$ over the distribution of $\vec{p}$.

Theorem A.23. *Given a positive integer $p \leq \left| (\vec{\sigma}, \vec{\theta}) \right|$, $\varepsilon > 0$, and $\delta \in (0, 1)$, the expected total number of measurements required for Algorithm 5 to estimate the gradient of $\mathcal{L}_t$ with deviation at most ε and probability $1 - \delta$ over the distribution of $\vec{p}$ is at least*

$$\frac{\left| (\vec{\sigma}, \vec{\theta}) \right|}{p} \cdot \frac{32^2}{2 \left((\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \varepsilon \right)^2} \ln \left(\frac{2 \left| (\vec{\sigma}, \vec{\theta}) \right|}{\delta p} \right)$$

Proof of Theorem A.23. For simplicity, we again denote $(\vec{\sigma}, \vec{\theta})$ by w , where $|w|$ represents its length. Taking the expected value of Equation (31) over the distribution of $\vec{p}$, we observe:

$$\begin{aligned}
& \Pr \left[\left\| \vec{p} \odot \tilde{\nabla} \mathcal{L}_t(w) - \vec{p} \odot \nabla \mathcal{L}_t(w) \right\|_2 \geq \varepsilon \middle| t \right] \\
&= \sum_{\vec{p} \in \{0,1\}^{|w|}} \Pr \left[\left\| \vec{p} \odot \tilde{\nabla} \mathcal{L}_t(w) - \vec{p} \odot \nabla \mathcal{L}_t(w) \right\|_2 \geq \varepsilon \middle| t, \vec{p} \right] \cdot \Pr[\vec{p}] \\
&\leq 2 \exp \left\{ \frac{-2 \left((\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \varepsilon \right)^2 \tilde{N}}{32^2} \right\} \cdot \left(\sum_{\vec{p} \in \{0,1\}^{|w|}} |\mathcal{I}_{\vec{p}}| \cdot \left(1 - \frac{1}{p} \right)^{|w| - |\mathcal{I}_{\vec{p}}|} \cdot \left(\frac{1}{p} \right)^{|\mathcal{I}_{\vec{p}}|} \right) \\
&= 2 \exp \left\{ \frac{-2 \left((\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \varepsilon \right)^2 \tilde{N}}{32^2} \right\} \cdot \left(\sum_{k=0}^{|w|} k \cdot \binom{|w|}{k} \cdot \left(1 - \frac{1}{p} \right)^{|w| - k} \cdot \left(\frac{1}{p} \right)^k \right)
\end{aligned}$$

Next, Using the identity $\sum_{i=0}^n i \cdot \binom{n}{i} (1 - 1/p)^{n-i} (1/p)^i = n/p$ for all $n \in \mathbb{Z}_{\geq 1}$ and $p \geq 1$, we arrive at:

$$\Pr \left[\left\| \vec{p} \odot \tilde{\nabla} \mathcal{L}_t(w) - \vec{p} \odot \nabla \mathcal{L}_t(w) \right\|_2 \geq \varepsilon \middle| t \right] \leq 2 \exp \left\{ \frac{-2 \left((\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \varepsilon \right)^2 \tilde{N}}{32^2} \right\} \cdot \frac{|w|}{p} \quad (32)$$

Then, let us determine the expected number of measurements required to estimate each selected partial derivative (i.e., $\tilde{N}$) over the distribution of $\vec{p}$. Setting right hand side of Equation (32) to at most δ and solving for $\tilde{N}$, we obtain:

$$\tilde{N} \geq \frac{32^2}{2 \left((\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \varepsilon \right)^2} \ln \left(\frac{2|w|}{\delta p} \right)$$

Moreover, since each entry of $\vec{p}$ is sampled from a Bernoulli($1/p$) distribution, we know that $\mathbb{E}[\|\vec{p}\|_0] = \frac{|w|}{p}$. In other words, there are $|w|/p$ number of selected partial derivatives in expectation.

Combining the above results, we conclude that over the distribution of $\vec{p}$, the expected total number of measurements (i.e., $(|w|/p) \cdot \tilde{N}$) required is at least:

$$\frac{|w|}{p} \cdot \frac{32^2}{2 \left((\Gamma_{\vec{\sigma}}^{-1})^2 \cdot \varepsilon \right)^2} \ln \left(\frac{2|w|}{\delta p} \right) \quad (33)$$

□

Remark A.24. Equation (33) suggests that a small ε results in an exponential sample size. To mitigate this, one approach is to select ε large enough to determine the direction of the gradient reliably.

To determine an appropriate value for ε , we consider the value of the partial derivatives at $(\vec{\sigma}, \vec{\theta})$. From Equation (3), Equation (8), and Equation (10), we observe that the partial derivatives are asymptotically equal to $\Theta(\Gamma_{\vec{\sigma}}^2 x)$ for some $x \in [-1, 1]$ when $(\vec{\sigma}, \vec{\theta})$ is not too close to a stationary point. Moreover, their directions are determined by x .

Thus, selecting ε as $\Gamma_{\vec{\sigma}}^2 \cdot \kappa$ for some $\kappa \in (0, 1)$ suffices to ensure a reliable determination of the gradient direction. Consequently, the expected total number of measurements required becomes at least:

$$\frac{|w|}{p} \cdot \frac{32^2}{2\kappa^2} \ln \left(\frac{2|w|}{\delta p} \right) \quad (34)$$

A.5 Convergence

A.5.1 Preliminary

We begin with a review of fundamental results in calculus.

Proposition A.25 (Extreme Value Theorem [18]). *Let $F : X \rightarrow Y$ be continuous, where Y is an ordered set in the order topology. If X is compact, then there exist points c and d in X such that $F(c) \leq F(x) \leq F(d)$ for every $x \in X$.*

Proposition A.26 (Estimation of Lipschitz constant on C^1 functions [13]). *Let $U \subseteq \mathbb{R}^n$ be an open subset, and suppose $F : U \rightarrow \mathbb{R}^m$ is of class C^1 . Then F is Lipschitz continuous on every compact convex subset $K \subset U$. The Lipschitz constant can be taken to be $\sup_{x \in K} \|\nabla F(x)\|_2$, where $\nabla F(x)$ is the Jacobian matrix.*

Lemma A.27 (Decent Lemma). *If $F : \mathbb{R}^d \rightarrow \mathbb{R}$ is of class of C^1 and K is any compact and convex subset of $\mathbb{R}^d$, then:*

$$F(y) \leq F(x) + \nabla F(x)(y - x) + \frac{L_K}{2} \|y - x\|_2^2,$$

for all $x, y \in K$, where L_K is the Lipschitz constant.

Sketch Proof of Lemma A.27. By applying Proposition A.26, we know that F is Lipschitz continuous on K . Furthermore, the Lipschitz constant L_K is bounded by $\sup_{x \in K} \|\nabla F(x)\|$. Then, using the inequality established in the proof of Lemma 2.25 in [7], for any $x, y \in K$, we know that:

$$F(y) \leq F(x) + (\nabla F(x))^T (y - x) + \int_0^1 \|\nabla F(x + t(y - x)) - \nabla F(x)\|_2 \cdot \|y - x\|_2 dt.$$

Since K is convex, we know that $x + t(y - x) \in K$ for all $t \in [0, 1]$. By the definition of local Lipschitz continuous, we can write:

$$\|\nabla F(x + t(y - x)) - \nabla F(x)\|_2 \leq L_K \|x + t(y - x) - x\|_2 = L_K t \|y - x\|_2,$$

for all $t \in [0, 1]$. Together, we conclude that:

$$\begin{aligned} F(y) &\leq F(x) + (\nabla F(x))^T (y - x) + \int_0^1 L_K t \|y - x\|_2^2 dt \\ &= F(x) + (\nabla F(x))^T (y - x) + L_K \|y - x\|_2^2 \left(\int_0^1 t dt \right) \\ &= F(x) + (\nabla F(x))^T (y - x) + \frac{L_K}{2} \|y - x\|_2^2. \end{aligned}$$

□

Based on the results above, we present the following two properties of our objective function: Lemma A.28 and Lemma A.29.

A.5.2 Properties of the Objective Function and Estimated Gradient

In this section, we present several important properties of the objective functions $\mathcal{L}$ and $\mathcal{L} + \mathcal{G}$, defined in Section 3, as well as the properties of the estimated gradient obtained from Algorithm 5.

Lemma A.28. *Let $\mathcal{L}(\vec{\sigma}, \vec{\theta})$ be the function defined in Equation (6). On the compact set $\mathcal{X} = \mathcal{X}_{\text{sigma}} \times \mathcal{X}_{\text{theta}}$, $\mathcal{L}$ is Lipschitz continuous.*

Proof of Lemma A.28. The proof of this lemma directly follows from Proposition A.26. Since maps $\frac{\partial \mathcal{L}}{\partial \sigma_{j,q}}$ given in Equation (10) and $\frac{\partial \mathcal{L}}{\partial \theta_j}$ given in Equation (8) are defined on $\mathbb{R}^{|\langle \vec{\sigma}, \vec{\theta} \rangle|}$, we know $\mathcal{L}$ is of class C^1 . Then, applying Proposition A.26, we know $\mathcal{L}$ is Lipschitz continuous on $\mathcal{X}$. □

Lemma A.29. $\mathcal{L}(\vec{\sigma}, \vec{\theta}) + \mathcal{G}(\vec{\sigma}, \vec{\theta})$ is lower-bounded on $\mathbb{R}^{|\langle \vec{\sigma}, \vec{\theta} \rangle|}$.

Proof of Lemma A.29. Given the definition of $\mathcal{L}(\vec{\sigma}, \vec{\theta})$ and $\mathcal{G}(\vec{\sigma}, \vec{\theta})$, we know that $\mathcal{L}(\vec{\sigma}, \vec{\theta}) + \mathcal{G}(\vec{\sigma}, \vec{\theta})$ is finite and well-defined when $(\vec{\sigma}, \vec{\theta}) \in \mathcal{X}$. So, it suffices to check $\mathcal{L}(\vec{\sigma}, \vec{\theta}) + \mathcal{G}(\vec{\sigma}, \vec{\theta})$ is lower bounded on $\mathcal{X}$. Given the definition of $\mathcal{L}$ and $\mathcal{G}$, we know that $\mathcal{L}(\vec{\sigma}, \vec{\theta}) + \mathcal{G}(\vec{\sigma}, \vec{\theta}) = \mathcal{L}(\vec{\sigma}, \vec{\theta})$ on $\mathcal{X}$. Since $\mathcal{L}(\vec{\sigma}, \vec{\theta})$ is continuous and $\mathcal{X}$ is compact, by the extreme value theorem (Proposition A.25), we know $\mathcal{L}(\vec{\sigma}, \vec{\theta})$ is lower-bounded, which implies $\mathcal{L}(\vec{\sigma}, \vec{\theta}) + \mathcal{G}(\vec{\sigma}, \vec{\theta})$ is lower-bounded. □

Proposition A.30. Let $\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(\vec{\sigma}, \vec{\theta})$ be an estimation of gradient of $\mathcal{L}_{i_t}$ at some fixed point $(\vec{\sigma}, \vec{\theta})$ generated by Algorithm 5, where $i_t \sim \text{uniform}\{1, \dots, |\mathcal{D}|\}$ and $\vec{p}_t \sim (\text{Bernoulli}(1/p))^{|(\vec{\sigma}, \vec{\theta})|}$. Then, $\mathbb{V}[\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(\vec{\sigma}, \vec{\theta})] = \mathbb{E}[\|\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(\vec{\sigma}, \vec{\theta}) - \frac{1}{p} \cdot \nabla \mathcal{L}(\vec{\sigma}, \vec{\theta})\|_2^2]$ is bounded.

Proof of Proposition A.30. For clarity, let w denote $(\vec{\sigma}, \vec{\theta})$. First, by taking the full expectation of $\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(\vec{\sigma}, \vec{\theta})$ using Lemma A.18, we obtain:

$$\mathbb{E}[\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w)] = \frac{1}{|\mathcal{D}|} \sum_{i_t=1}^{|\mathcal{D}|} \mathbb{E}[\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w) | i_t] = \frac{1}{|\mathcal{D}|} \sum_{i_t=1}^{|\mathcal{D}|} \frac{1}{p} \cdot \nabla \mathcal{L}_{i_t}(w) = \frac{1}{p} \nabla \mathcal{L}(w), \quad (35)$$

which leads to:

$$\begin{aligned} & \mathbb{E}[\|\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w) - \mathbb{E}[\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w)]\|_2^2] \\ &= \mathbb{E}\left[\sum_{\alpha=1}^{|w|} \left(\left(\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w)\right)_{\alpha} - \left(\frac{1}{p} \nabla \mathcal{L}(w)\right)_{\alpha}\right)^2\right] \\ &= \sum_{\alpha=1}^{|w|} \mathbb{E}\left[\left(\left(\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w)\right)_{\alpha} - \left(\frac{1}{p} \nabla \mathcal{L}(w)\right)_{\alpha}\right)^2\right] \\ &= \sum_{\alpha=1}^{|w|} \text{var}\left[\left(\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w)\right)_{\alpha}\right]. \end{aligned}$$

Here, $(\cdot)_{\alpha}$ represents the α^{th} component of the input vector. From the observation of Algorithm 3, Algorithm 4, and Algorithm 5, we note that $\left(\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w)\right)_{\alpha} : \{-\Gamma_{\vec{\sigma}}, \Gamma_{\vec{\sigma}}\}^{\tilde{N}} \rightarrow \mathbb{R}$ defined by

$$\begin{aligned} & \left(\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w)\right)_{\alpha} (o_1, \dots, o_{\tilde{N}/2}, o_{\tilde{N}/2+1}, \dots, o_{\tilde{N}}) \\ &= -(\vec{p}_t)_{\alpha} \cdot \mathcal{C} \left(y_{i_t} - \frac{1}{\tilde{N}/2} \sum_{\beta=1}^{\tilde{N}/2} o_{\beta} \right) \left(\frac{1}{\tilde{N}/2} \sum_{\beta=\tilde{N}/2+1}^{\tilde{N}} o_{\beta} \right), \end{aligned}$$

where $\mathcal{C}$ is 2 or 4 and $\Gamma_{\vec{\sigma}} = \exp\left\{\sum_{j,k} \sigma_{j,k}\right\}$ and $\tilde{N}$ is an even number. Since the mapping $\left(\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w)\right)_{\alpha}$ has a finite domain, we note that:

$$\begin{aligned} & \max_{o_{\beta}, o'_{\beta} \in \{-\Gamma_{\vec{\sigma}}, \Gamma_{\vec{\sigma}}\}} \left| \left(\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w)\right)_{\alpha} (o_1, \dots, o_{\beta}, \dots, o_{\tilde{N}}) \right. \\ & \quad \left. - \left(\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w)\right)_{\alpha} (o_1, \dots, o'_{\beta}, \dots, o_{\tilde{N}}) \right| \end{aligned}$$

is finite for all i_t and $\vec{p}_t$. Define

$$\begin{aligned} \mathfrak{U} = & \max_{\substack{o_{\beta}, o'_{\beta} \in \{-\Gamma_{\vec{\sigma}}, \Gamma_{\vec{\sigma}}\} \\ \vec{p}_t \in \{0,1\}^{|w|} \\ i_t=1, \dots, |\mathcal{D}|}} \left| \left(\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w)\right)_{\alpha} (o_1, \dots, o_{\beta}, \dots, o_{\tilde{N}}) \right. \\ & \left. - \left(\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w)\right)_{\alpha} (o_1, \dots, o'_{\beta}, \dots, o_{\tilde{N}}) \right|. \end{aligned} \quad (36)$$

Using the Efron-Stetin inequality, we obtain

$$\begin{aligned}
& \text{var} \left[\left(\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w) \right)_\alpha \right] \\
& \leq \frac{1}{2} \sum_{\beta=1}^{\tilde{N}} \mathbb{E} \left[\left(\left(\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w) \right)_\alpha (o_1, \dots, o_\beta, \dots, o_{\tilde{N}}) \right. \right. \\
& \quad \left. \left. - \left(\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w) \right)_\alpha (o_1, \dots, o'_\beta, \dots, o_{\tilde{N}}) \right)^2 \right] \\
& \leq \frac{1}{2} \sum_{\beta=1}^{\tilde{N}} \mathfrak{U}^2 = \frac{\tilde{N} \cdot \mathfrak{U}^2}{2}.
\end{aligned}$$

Together, we have

$$\mathbb{V} \left[\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w) \right] \leq |w| \cdot \frac{\tilde{N} \cdot \mathfrak{U}^2}{2} < \infty.$$

□

A.5.3 Convergence bound of the Learning Algorithm

We now present the convergence bound in the following theorem.

Theorem A.31. *Let $\mathcal{L}(\vec{\sigma}, \vec{\theta}) + \mathcal{G}(\vec{\sigma}, \vec{\theta})$ be the objective function, where $\mathcal{L}$ and $\mathcal{G}$ are defined in Equation (6) and Equation (7). Let $\{w_t = (\vec{\sigma}^{(t)}, \vec{\theta}^{(t)})\}, t = 1, \dots, T\}$ be a sequence of parameters generated by the proximal-SGD (Algorithm 1). Suppose w_0 and $\bar{w}$ are the initial point and a stationary point in $\mathcal{X} = \mathcal{X}_{\text{sigma}} \times \mathcal{X}_{\text{theta}}$, respectively. Let $\{\eta_t | 0 < \eta_t < 1, t = 0, \dots, T\}$ be a sequence such that $\sum_{t=0}^{\infty} \eta_t < \infty$. Then,*

$$\left(\frac{1}{\sum_{t=1}^T \eta_t} \sum_{t=1}^T \eta_t \mathbb{E} [\mathcal{L}(w_t)] \right) - \mathcal{L}(\bar{w}) \leq \frac{p \cdot \|w_0 - \bar{w}\|_2^2}{2 \sum_{t=1}^T \eta_t} + \frac{3L_{\mathcal{X}} \sum_{t=1}^T \eta_t \mathfrak{B}_{\max}}{2 \sum_{t=1}^T \eta_t} + \frac{p \sum_{t=1}^T \eta_t^2 \mathcal{V}}{2 \sum_{t=1}^T \eta_t},$$

where $\mathcal{V} = \sup_{w_t} \mathbb{V} \left[\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w_t) \right]$, p is a positive integer, $L_{\mathcal{X}}$ is the local Lipschitz constant, and $\mathfrak{B}_{\max} = \sum_{l=1}^L \sum_{k \in \mathcal{K}_l} (\mathfrak{B}^{(l,k)})^2 + \sum_{b=1}^l (\mathfrak{B}_1^{(l)} - \mathfrak{B}_0^{(l)})^2$.

Proof of Theorem A.31. For simplicity and clarity, we define $w_t = (\vec{\sigma}_t, \vec{\theta}_t)$, $\mathcal{X} = \mathcal{X}_{\text{sigma}} \times \mathcal{X}_{\text{theta}}$, and $\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w_t)$ denotes the gradient estimation for $\mathcal{L}_{i_t}(w_t) = (y_{i_t} - \text{tr}(\mathcal{M}\mathcal{U}_R(\rho_{i_t})))^2$ generated by Algorithm 5. Here, $i_t \sim \text{uniform} \{1, 2, \dots, |\mathcal{D}|\}$ and $\vec{p}_t \sim (\text{Bernoulli}(1/p))^{|\mathcal{w}_t|}$. By definition, we know $w_{t+1} = \text{prox}_{\mathcal{G}}(w_t - \eta_t \vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w_t))$, which implies that $w_t \in \mathcal{X}$ for all $t = 0, \dots, T$. Applying lemma 8.17 from [7], we obtain the following result:

$$\begin{aligned}
\|w_{t+1} - \bar{w}\|_2^2 &= \left\| \text{prox}_{\mathcal{G}}(w_t - \eta_t \vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w_t)) - \text{prox}_{\mathcal{G}}(\bar{w}) \right\|_2^2 \\
&\leq \left\| w_t - \eta_t \vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w_t) - \bar{w} \right\|_2^2.
\end{aligned}$$

Using the fact $\|x - y\|_2^2 = (x - y)^T(x - y) = \|x\|_2^2 - 2x^T y + \|y\|_2^2$, we can expand the squared norm as below:

$$\begin{aligned}
& \left\| w_t - \eta_t \vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w_t) - \bar{w} \right\|_2^2 \\
&= \|w_t - \bar{w}\|_2^2 - 2\eta_t \left(\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w_t) \right)^T (w_t - \bar{w}) + \eta_t^2 \left\| \vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w_t) \right\|_2^2.
\end{aligned}$$

Together, we have

$$\|w_{t+1} - \bar{w}\|_2^2 - \|w_t - \bar{w}\|_2^2 \leq -2\eta_t \left(\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w_t) \right)^T (w_t - \bar{w}) + \eta_t^2 \left\| \vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w_t) \right\|_2^2.$$

Now, taking the expectation conditioned on w_t , we get

$$\begin{aligned}
& \mathbb{E} \left[\|w_{t+1} - \bar{w}\|_2^2 - \|w_t - \bar{w}\|_2^2 \middle| w_t \right] \\
& \leq \mathbb{E} \left[-2\eta_t \left(\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w_t) \right)^T (w_t - \bar{w}) + \eta_t^2 \left\| \vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w_t) \right\|_2^2 \middle| w_t \right] \\
& = -2\eta_t \mathbb{E} \left[\left(\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w_t) \right)^T (w_t - \bar{w}) \middle| w_t \right] + \eta_t^2 \mathbb{E} \left[\left\| \vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w_t) \right\|_2^2 \middle| w_t \right]. \quad (37)
\end{aligned}$$

For the first term in Equation (37), using Proposition A.17 and linearity of gradient, we arrive at:

$$\begin{aligned}
& \mathbb{E} \left[\left(\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w_t) \right)^T (w_t - \bar{w}) \middle| w_t \right] \\
& = \sum_{\vec{p}_t \in \{0,1\}^{|w_t|}} \left(\frac{1}{|\mathcal{D}|} \sum_{i_t=1}^{|\mathcal{D}|} \mathbb{E} \left[\left(\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w_t) \right)^T (w_t - \bar{w}) \middle| w_t, i_t, \vec{p}_t \right] \right) \cdot \Pr[\vec{p}_t] \\
& = \sum_{\vec{p}_t \in \{0,1\}^{|w_t|}} \left(\frac{1}{|\mathcal{D}|} \sum_{i_t=1}^{|\mathcal{D}|} (\vec{p}_t \odot \nabla \mathcal{L}_{i_t}(w_t))^T (w_t - \bar{w}) \right) \cdot \Pr[\vec{p}_t] \\
& = \sum_{\vec{p}_t \in \{0,1\}^{|w_t|}} (\vec{p}_t \odot \nabla \mathcal{L}(w_t))^T (w_t - \bar{w}) \cdot \Pr[\vec{p}_t] \\
& = \left(\left(\sum_{\vec{p}_t \in \{0,1\}^{|w_t|}} \vec{p}_t \cdot \Pr[\vec{p}_t] \right) \odot \nabla \mathcal{L}(w_t) \right)^T (w_t - \bar{w}) \\
& = \frac{1}{p} \cdot (\nabla \mathcal{L}(w_t))^T (w_t - \bar{w}).
\end{aligned}$$

For the second term in Equation (37), by Proposition A.30, we know that the value $\mathbb{V} \left[\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w_t) \right]$ conditioned on w_t is bounded. This implies that $\mathbb{E} \left[\left\| \vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w_t) \right\|_2^2 \middle| w_t \right]$ is also bounded, and specifically we have the inequality:

$$\mathbb{E} \left[\left\| \vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w_t) \right\|_2^2 \middle| w_t \right] \leq \mathbb{V} \left[\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w_t) \right].$$

Therefore, we have

$$\mathbb{E} \left[\|w_{t+1} - \bar{w}\|_2^2 - \|w_t - \bar{w}\|_2^2 \middle| w_t \right] \leq -\frac{2\eta_t}{p} (\nabla \mathcal{L}(w_t))^T (w_t - \bar{w}) + \eta_t^2 \mathbb{V} \left[\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w_t) \right].$$

By combining all the above and taking the full expectation, we get:

$$\begin{aligned}
\mathbb{E} \left[\|w_{t+1} - \bar{w}\|_2^2 - \|w_t - \bar{w}\|_2^2 \right] &= \mathbb{E} \left[\mathbb{E} \left[\|w_{t+1} - \bar{w}\|_2^2 - \|w_t - \bar{w}\|_2^2 \middle| w_t \right] \right] \\
&\leq \mathbb{E} \left[-\frac{2\eta_t}{p} (\nabla \mathcal{L}(w_t))^T (w_t - \bar{w}) \right] + \eta_t^2 \mathbb{E} \left[\mathbb{V} \left[\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w_t) \right] \right] \\
&\leq \mathbb{E} \left[-\frac{2\eta_t}{p} (\nabla \mathcal{L}(w_t))^T (w_t - \bar{w}) \right] + \eta_t^2 \mathcal{V},
\end{aligned}$$

where $\mathcal{V} = \sup_{w_t} \mathbb{V} \left[\vec{p}_t \odot \tilde{\nabla} \mathcal{L}_{i_t}(w_t) \right]$. Telescoping the both sides, we obtain:

$$\begin{aligned}
\sum_{t=1}^T \mathbb{E} \left[\|w_t - \bar{w}\|_2^2 - \|w_{t-1} - \bar{w}\|_2^2 \right] &= \mathbb{E} \left[\|w_T - \bar{w}\|_2^2 - \|w_0 - \bar{w}\|_2^2 \right] \\
&\leq -\sum_{t=1}^T \frac{2\eta_t}{p} \mathbb{E} \left[(\nabla \mathcal{L}(w_t))^T (w_t - \bar{w}) \right] + \sum_{t=1}^T \eta_t^2 \mathcal{V} \\
&= \sum_{t=1}^T \frac{2\eta_t}{p} \mathbb{E} \left[(\nabla \mathcal{L}(w_t))^T (\bar{w} - w_t) \right] + \sum_{t=1}^T \eta_t^2 \mathcal{V}. \quad (38)
\end{aligned}$$

Next, let us bound $(\nabla \mathcal{L}(w_t))^T (\bar{w} - w_t)$. Using the fact that $x^T y = \|x\| \|y\| \cos \theta$ for any x, y , we observe that:

$$(\nabla \mathcal{L}(w_t) - \nabla \mathcal{L}(\bar{w}))^T (\bar{w} - w_t) \leq \|\nabla \mathcal{L}(w_t) - \nabla \mathcal{L}(\bar{w})\|_2 \cdot \|\bar{w} - w_t\|_2.$$

On the left-hand side, we have:

$$(\nabla \mathcal{L}(w_t) - \nabla \mathcal{L}(\bar{w}))^T (\bar{w} - w_t) = (\nabla \mathcal{L}(w_t))^T (\bar{w} - w_t) - (\nabla \mathcal{L}(\bar{w}))^T (\bar{w} - w_t).$$

On the right hand side, since $\mathcal{L}$ is locally Lipschitz continuous on $\mathcal{X}$, by definition, we know that:

$$\|\nabla \mathcal{L}(w_t) - \nabla \mathcal{L}(\bar{w})\|_2 \cdot \|\bar{w} - w_t\|_2 \leq L_{\mathcal{X}} \|\bar{w} - w_t\|_2^2.$$

Combining and re-arranging terms, we get:

$$(\nabla \mathcal{L}(w_t))^T (\bar{w} - w_t) \leq (\nabla \mathcal{L}(\bar{w}))^T (\bar{w} - w_t) + L_{\mathcal{X}} \|\bar{w} - w_t\|_2^2.$$

Furthermore, applying the decent lemma (Lemma A.27), we have:

$$(\nabla \mathcal{L}(w_t))^T (\bar{w} - w_t) \leq \mathcal{L}(\bar{w}) - \mathcal{L}(w_t) + \frac{3L_{\mathcal{X}}}{2} \|\bar{w} - w_t\|_2^2. \quad (39)$$

Now, combining Equation (38) and Equation (39), we obtain:

$$\mathbb{E} \left[\|w_T - \bar{w}\|_2^2 - \|w_0 - \bar{w}\|_2^2 \right] \leq \sum_{t=1}^T \frac{2\eta_t}{p} \mathbb{E} \left[\mathcal{L}(\bar{w}) - \mathcal{L}(w_t) + \frac{3L_{\mathcal{X}}}{2} \|\bar{w} - w_t\|_2^2 \right] + \sum_{t=1}^T \eta_t^2 \mathcal{V},$$

which is equivalent to

$$\begin{aligned} & \sum_{t=1}^T \frac{2\eta_t}{p} (\mathbb{E} [\mathcal{L}(w_t)] - \mathcal{L}(\bar{w})) \\ & \leq \|w_0 - \bar{w}\|_2^2 - \mathbb{E} [\|w_T - \bar{w}\|_2^2] + \frac{3L_{\mathcal{X}}}{p} \sum_{t=1}^T \eta_t \mathbb{E} [\|\bar{w} - w_t\|_2^2] + \sum_{t=1}^T \eta_t^2 \mathcal{V} \\ & \leq \|w_0 - \bar{w}\|_2^2 + \frac{3L_{\mathcal{X}}}{p} \sum_{t=1}^T \eta_t \mathbb{E} [\|\bar{w} - w_t\|_2^2] + \sum_{t=1}^T \eta_t^2 \mathcal{V}. \end{aligned} \quad (40)$$

Since w_t and $\bar{w}$ are both in $\mathcal{X}$ and $\mathcal{X}$ is a hyper-rectangle, we know that:

$$\|\bar{w} - w_t\|_2^2 \leq \sum_{l=1}^L \sum_{k \in \mathcal{K}_l} \left(\mathfrak{B}^{(l,k)} \right)^2 + \sum_{\substack{l=1 \\ b_l=1}}^L \left(\mathfrak{B}_1^{(l)} - \mathfrak{B}_0^{(l)} \right)^2 = \mathfrak{B}_{max}.$$

Thus, Equation (40) can be written as:

$$\sum_{t=1}^T \frac{2\eta_t}{p} (\mathbb{E} [\mathcal{L}(w_t)] - \mathcal{L}(\bar{w})) \leq \|w_0 - \bar{w}\|_2^2 + \frac{3L_{\mathcal{X}}}{p} \cdot \sum_{t=1}^T \eta_t \mathfrak{B}_{max} + \sum_{t=1}^T \eta_t^2 \mathcal{V}.$$

Finally, dividing both sides of the inequality by $\frac{2 \sum_{t=1}^T \eta_t}{p}$, we obtain:

$$\left(\frac{1}{\sum_{t=1}^T \eta_t} \sum_{t=1}^T \eta_t \mathbb{E} [\mathcal{L}(w_t)] \right) - \mathcal{L}(\bar{w}) \leq \frac{p \cdot \|w_0 - \bar{w}\|_2^2}{2 \sum_{t=1}^T \eta_t} + \frac{3L_{\mathcal{X}} \sum_{t=1}^T \eta_t \mathfrak{B}_{max}}{2 \sum_{t=1}^T \eta_t} + \frac{p \sum_{t=1}^T \eta_t^2 \mathcal{V}}{2 \sum_{t=1}^T \eta_t}.$$

□

A.6 Additional Experiments

A.6.1 Evaluation on Fashion-MNIST Dataset

We evaluate our approach on the Fashion-MNIST dataset [32] (Pullover vs. Shirt; 5000 training samples; standard test split; three seeds). Each 28×28 image is standardized, reduced by PCA to a compact feature vector (64 dimensions for our 6-qubit scenario), and ℓ_2 -normalized to unit length.

In simulation, we directly initialize the statevector to the corresponding amplitude encoded state $|\psi(x)\rangle$, then apply a 6-qubit, two-layer HEA with parameterized R_X, R_Y, R_Z rotations and a circular CNOT entangler as in Figure 3. The labels are assigned to -1 (shirt) and $+1$ (pullover). This direct initialization avoids extra state preparation depth and associated gate noise, so our comparisons isolate training-time noise and mitigation effects. Besides, the initial learning rate ($\eta^{(1)}$) is selected using grid search across a wide range and then refined with binary search for better precision. Subsequently, the learning rate is annealed monotonically.

Under static noise, our inverse learning method initially lags Van den Berg et al.’s pre-characterized approach but surpasses it mid-training, achieving a lower final loss (Figure 5a). The noiseless PQC provides the best-case reference, while the baseline noisy PQC yields the highest error, confirming that both mitigation strategies effectively reduce noise impact. Consistent with these trends, Table 2 summarizes the test accuracies of the compared approaches. Under dynamic noise, we emulate hardware drift by slightly increasing the error rates of three CNOT gates early in training. While Van den Berg et al.’s method retains its fixed calibration, our approach updates the inverse parameters online and achieves lower MSE by tracking the drift (Figure 5b). This highlights the practical advantage of adaptive inverse learning over pre-calibrated approaches in realistic quantum hardware conditions. Finally, probabilistic subsampling analysis using the same static noise profile as Figure 5a, reveals the expected convergence degradation with increased sparsity ($p = 1 > p = 2 > p = 4$), with denser sampling providing more stable optimization, consistent with the MNIST results (Figure 5c).

Table 2: Classification accuracies across Different methods for Fashion-MNIST dataset

	ACCURACY
NOISELESS PQC	78.87 ± 1.32
NOISY PQC (BASELINE)	69.57 ± 3.35
MITIGATED PQC (VAN DEN BERG ET.AL)	72.37 ± 2.53
OURS	74.23 ± 2.84

A.6.2 Additional Results—PEC and ZNE under Dynamic Noise

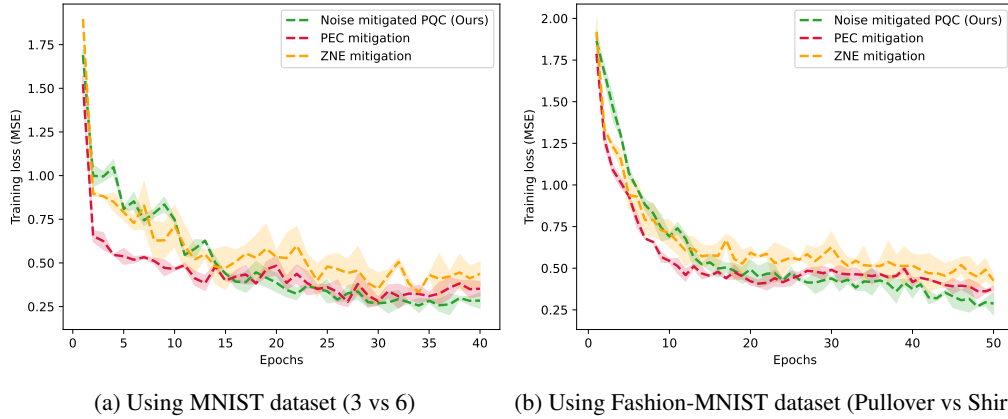


Figure 6: Training loss comparison among PEC, ZNE and our mitigation framework for binary classification task under dynamic noise. Here, each epoch summarizes 50 iterations of the training process.

Under dynamic noise, our adaptive inverse noise learning method updates the inverse noise parameters at every iteration and consistently achieves the lowest final MSE on both MNIST (Figure 6a) and Fashion-MNIST (Figure 6b). PEC often looks competitive because it applies the full inverse from a pre-calibrated noise map, but it has practical limits: (i) it is measurement intensive: the shot demand multiplies across mitigated gates, leading to an effective exponential growth with circuit depth; (ii) it does not track drift without recalibration; and (iii) it loses accuracy as the device moves further away from the calibration point. ZNE performs worst on average with higher variance, because

extrapolation assumes stable noise scaling, which breaks under time-varying noise. The convergence of our method across both datasets supports the conclusion that adaptive inverse noise learning is important for stable optimization on realistic, drifting hardware- capabilities that neither fixed PEC calibrations nor ZNE's static extrapolation provide without manual intervention.