Private training in quantum machine learning

Tigran Sedrakyan[1], Frédéric Grosshans[1], Elham Kashefi[1,2]

[1]Sorbonne Université, CNRS, LIP6, Paris, 75005 France.
[2]School of Informatics, University of Edinburgh, Edinburgh, EH8 9AB United Kingdom.

Abstract

With the emergence of machine learning (ML) models trained on large datasets containing potentially sensitive data, a major question in AI safety is how to make learning private with respect to the training data. Similar to classical machine learning, quantum machine learning (QML) models are not devoid of privacy vulnerabilities. Differential privacy (DP) is a standard tool for training ML models on sensitive data, but its impact in QML remains poorly understood. In this work we study private training in hybrid variational QML models using a classical private DP-SGD optimizer applied to pipelines with classical inputs and outputs. We analyze the interplay between gradient clipping and calibrated noise addition in DP-SGD, and its impact on optimization and accuracy for noisy and noiseless quantum models. We first explain why quantum noise does not provide a satisfactory replacement for the calibrated noise in DP-SGD for ensuring privacy. We then show how the deterministic bounds on gradient norms for a wide class of quantum models translate into explicit control of the detrimental clipping bias introduced by DP-SGD. Finally, we formulate a numerical comparison protocol under fixed clipping threshold and privacy budget and evaluate it on synthetic and image-classification tasks for equivalent quantum and classical models. Our results suggest that quantum models can retain higher accuracy in private-training regimes where the formal privacy guarantee is ensured by a classical DP-SGD mechanism.

Keywords: differential privacy, quantum machine learning, DP-SGD

Contents

1 Introduction **3**

2 Background **4**
2.1 Quantum machine learning 5
2.2 Differential privacy 6
2.3 DP-SGD and gradient clipping 8

3 Results **12**
3.1 Noise propagation 13
3.2 Clipping bias bound 14

4 Numerical simulations **16**
4.1 Effect of Clipping 17
4.2 Private training 19

5 Conclusion **21**

6 Statements and Declarations **23**

A Proofs of Theorems **29**

1 Introduction

Learning from sensitive data while providing rigorous privacy guarantees is central to the emerging field of AI safety. Differential privacy (DP) offers an analytical way to protect individuals, bounding how much one record in a dataset can affect the output distribution of a randomized algorithm, by adding carefully calibrated noise to the outputs of a mechanism [1]. Over the last decade, DP has been brought into practice for training deep models through DP stochastic gradient descent (DP-SGD) [2–4]. DP-SGD computes per-example gradients, clcalips each of them to a fixed norm threshold, and then adds calibrated Gaussian noise yielding rigorous privacy guarantees composed across training iterations, but at a nontrivial utility (accuracy) cost. In particular, per-example clipping is not merely a technical device for bounding sensitivity: it also distorts the optimization and can become a major source of performance degradation, which has been studied previously [5–7].

On the other hand, a rapidly growing body of literature studies quantum machine learning (QML) for classical data [8, 9], where classical data are encoded in quantum states and processed using parameterized quantum circuits. In this context, differentially private QML routines have been demonstrated, typically by combining classical DP-SGD with hybrid models, and early studies suggest that utility can be retained while enforcing a meaningful privacy budget (ε, δ) [10–12]. The division of labor in these models is convenient analytically: it makes explicit where sensitivity arises, keeps the privacy notion identical to the one used in classical machine learning, and lets us ask whether QML enjoys a structural utility advantage at fixed privacy budget.x

In our work, we first discuss whether quantum hardware noise and shot noise can play a beneficial role in private training. We conclude that quantum noise may marginally improve empirical utility by symmetrizing the pre-clipping gradient distribution, but by itself it does not provide a practically tight replacement for standard DP-SGD in the record-level privacy setting studied here. In light of this, we focus on a simple question with practical consequences: is it possible to have an empirical advantage in privacy-utility trade-offs, by applying classical DP mechanisms to QML pipelines, compared to classical neural networks? Our starting point is that quantum machine learning (QML) is structurally different. In the class of variational quantum models considered here, the trainable quantum component evolves unitarily by design, while the model outputs are extracted through expectation values of bounded observables. Using these facts, we study privacy-preserving training on classical datasets and isolate a mechanism implying deterministic bound of DP-SGD clipping bias due to norm-controlled quantum gradients. We do not present novel privacy guarantees here; rather, using simple arguments, we support numerically the thesis that off-the-shelf classical DP mechanisms transfer naturally to hybrid QML settings, where they perform well due to structural reasons.

Similar structural reasons have been explored in classical learning literature as well. Classical models sometimes impose orthogonal or unitary parameterizations to stabilize gradients, use bounded activation functions or explicitly constrain operator norms of every layer in the network for robustness and sensitivity reasons, as observed in Lipschitz neural networks [13–16]. In such models, norm preservation is an additional architectural constraint, which eliminates the need for clipping entirely,

but comes with certain expressivity caveats. In the hybrid quantum setting considered here, analogous control is inherited from the model class itself through unitary evolution together with bounded readout. In numerical simulations we compare how quantum models compare to classical gradient norm-constrained Lipschitz networks.

Our work should also be distinguished from the existing privacy literature around quantum learning. Quantum differential privacy (QDP) extends privacy guarantees to quantum channels and output states [17, 18]. The broader privacy context around QML includes privacy guarantees in federated settings, as well as the role of hardware and measurement noise. Some proposals explore whether device-level randomness can be used for DP, for instance by calibrating to measurement noise or by composing device channels with classical noise to meet a target privacy budget [12]. Some works extend this idea to private inference and adversarial robustness in QML [19]. Others consider formal bridges between quantum and classical DP where the privacy of a quantum output implies the privacy of all classical statistics derived from it, facilitating end-to-end reasoning about hybrid learning pipelines [20]. Surveys and position papers are beginning to map the broader landscape, including risks particular to QML, evaluation gaps, and the significance of auditing [21, 22]. Importantly, in our work we do not consider QDP applied to quantum states, but rather focus on training-time privacy of classical training data in hybrid QML pipelines, with classical inputs and outputs.

In summary, we make three contributions. First, we show that using quantum noise as the only noise source in DP-SGD is too restrictive and dilutes the privacy budget beyond practical thresholds; this claim hinges on the propagation of quantum noise to the loss gradients and a gradient bound for the class of variational quantum models considered here. Second, we show that the presence of this deterministic gradient bound allows for tighter control of the clipping bias induced by DP-SGD, providing a concrete structural explanation for why quantum models may tolerate smaller clipping thresholds than matched classical baselines under the same privacy mechanism. Finally, we formulate an experimental comparison protocol for matched quantum and classical models under identical clipping and privacy parameters, linking the theory to the observed utility in numerical study.

The rest of the paper is organized as follows. Section 2 reviews the relevant background on quantum machine learning, differential privacy, and DP-SGD. Section 3 fixes the learning setting and presents the main analytical results: we first analyse the propagation of quantum shot and hardware noise to loss gradients, and then derive a deterministic clipping-bias bound for quantum models. We also explain why native quantum noise is not a satisfactory replacement for the formal privacy mechanism even though it may improve utility. Section 4 describes the numerical studies.

2 Background

This section reviews the concepts used throughout the paper. We keep the discussion self-contained while emphasizing the aspects that matter most for private training: the structure of quantum machine learning models, the definition of differential privacy, and the role of clipping in DP-SGD.

2.1 Quantum machine learning

Quantum machine learning (QML) refers to machine learning where data are processed by parameterized quantum circuits and the resulting measurement statistics, usually Pauli expectation values, form the basis for an objective function, which is then minimized using classical gradient-based methods. For a recent and comprehensive overview of techniques and methods in QML we refer to [9]. Here, we overview a common supervised learning scenario with classical data, with training dataset D and entries $(x_i, y_i) \in D$, where x_i are the features and y_i the label.

The first step in any hybrid QML pipeline working with classical data is the encoding. Let $\mathcal{X} \subseteq \mathbb{R}^n$ be the input domain. An encoding is a map $\rho : \mathcal{X} \to \mathcal{H}$. A common example is amplitude encoding $x \mapsto \rho(x) = |\phi(x)\rangle\langle\phi(x)|$ with $|\phi(x)\rangle = \sum_{j=1}^{n} x_j|j\rangle$ for normalized x with $\|x\|_2 = 1$. Other common encoding choices include angle, basis encodings and data reuploading schemes [23, 24]. Encodings control both expressivity and stability, and they determine how neighboring inputs appear in the state space: amplitude or angle encodings implement data-dependent feature maps whose induced kernels can be highly expressive, while concentrated encodings may separate classes better but also increase sensitivity.

After the encoding, a variational circuit applies a sequence of parametrized gates interleaved with fixed entanglers defining the ansatz, and a set of observables is measured to define the model output.

Definition 1 (Hybrid variational quantum classifier) Consider a K-class classification task. Let $\rho(x)$ be a data-encoding state on a Hilbert space $\mathcal{H}$, $U(\theta)$ a parameterised unitary with $\theta \in \mathbb{R}^p$, and $\{O_m\}_{m=1}^{M}$ a set of Hermitian observables on $\mathcal{H}$. The quantum part of the model produces the expectation values

$$q_m(x, \theta) \;=\; \mathrm{Tr}\big(O_m\, U(\theta)\, \rho(x)\, U(\theta)^\dagger\big), \qquad m = 1, \ldots, M \tag{1}$$

A fixed or trainable classical post-processing map $h_\phi \colon \mathbb{R}^M \to \mathbb{R}^K$ with parameters ϕ, turns these into class logits

$$f(x, \theta, \phi) \;=\; h_\phi\big(q_1(x, \theta), \ldots, q_M(x, \theta)\big), \tag{2}$$

which feed the loss $\ell(f(x, \theta, \phi),\, y)$ for a label $y \in \{1, \ldots, K\}$.

In the simplest case h_ϕ is the identity map (with $M = K$) and does not depend on ϕ, so $f(x, \theta) = \big(q_1(x, \theta), \ldots, q_K(x, \theta)\big)$. Variational classifiers then act as tunable non-classical feature maps whose outputs can be combined with standard losses and regularizers. In contrast to classical learning models, the output is not a vector of hidden activations but a set of expectation values of observables, which are optionally postprocessed classically and plugged into a classical optimisation loop.

With these ingredients, the generic supervised training setup proceeds as follows:

1. classical data x_i are encoded as quantum states $\rho(x_i)$;
2. a variational quantum circuit $U(\theta)$ is applied;
3. expectation values of one or more observables $\{O_m\}$ are computed;
4. the measurement results are post-processed to form the objective function $f(x_i, \theta)$ and the per-example loss $\ell(f(x_i, \theta), y_i)$, with the population loss defined as $\mathcal{L}(\theta) := \mathbb{E}_{(x_i, y_i)\sim D}[\ell(f(x_i, \theta), y_i)]$;

5. using a gradient-based (or gradient-free) optimizer, a classical subroutine updates θ in order to minimize $\mathcal{L}$

The same loop repeats across training iterations, also known as epochs. In mini-batch stochastic gradient descent (referred to simply as SGD), at every epoch the dataset is randomly subsampled into mini-batches, each of which is processed separately, after which the loss is averaged over the entries in the mini-batch. Once all entries in the dataset have been processed once, the epoch concludes. The stochastic gradient obtained over a batch B, $g_B = \frac{1}{|B|}\sum_{i\in B} \partial_\theta \ell(f(x_i,\theta), y_i)$, is an unbiased estimator of the true gradient, so that $\mathbb{E}[g_B] = \partial_\theta \mathcal{L}(\theta)$, meaning that SGD converges to the true optimum, provided enough training epochs.

The ansatz determines trainability and gradient concentration of the model: hardware-efficient and problem-inspired ansatzes can behave very differently with regard to barren plateaus and effective parameter counts. Highly non-local ansatzes are attractive for computational advantage in terms of expressivity, but are also vulnerable to barren plateaus, where gradients concentrate near zero as the number of qubits grows, slowing training or rendering it ineffective [25]. The primary reasons for barren plateaus are multiple: highly entangled initial states, overly expressive circuit ansatzes, global observable measurements, and noise can all contribute to their emergence. Conversely, architectures that are shallow and local are often devoid of barren plateaus and are easier to train, but may be closer to classical simulability. In this regime, gradient estimation and loss evaluation can often be performed efficiently on classical hardware, which weakens the case for a computational advantage even if learning is stable. In this work we consider quantum architectures within the classically simulable and trainable regime, so the potential for computational advantage is limited. In this scope, the quantum circuit may be executed on quantum hardware or simulated classically, but the claims apply independently of this choice.

2.2 Differential privacy

Differential privacy [1] formalizes the idea that the presence or absence of a single individual should not significantly change the distribution of an algorithm's outputs. Informally, if two datasets are identical except for one record, then a DP mechanism should make them almost indistinguishable to any observer, regardless of the available side information. The notion is parameterized by (ε, δ), where ε controls the strength of indistinguishability and δ allows for a small failure probability.

Definition 2 ((ε,δ)-differential privacy) Let $\mathcal{M}$ be a randomized mechanism mapping datasets to outputs in a measurable space. For parameters $\varepsilon \geq 0$ and $\delta \in [0,1]$, $\mathcal{M}$ is (ε,δ)-differentially private (DP) under a fixed adjacency relation on datasets if, for all adjacent $D \sim D'$ and all measurable $S \subseteq \operatorname{im}\mathcal{M}$,

$$\Pr[\mathcal{M}(D) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(D') \in S] + \delta \tag{3}$$

We will consider two datasets adjacent $D \sim D'$ (or neighboring) if they differ by exactly one record, which is present in one dataset and absent in the other, i.e.

$x \notin D', D = \{x\} \cup D'$. The two prerequisites for enforcing DP are global sensitivity and noise calibration. For a vector-valued query on the dataset $Q(D) \in \mathbb{R}^d$, the ℓ_2 global sensitivity is $\Delta_2(Q) = \sup_{D \sim D'} \|Q(D) - Q(D')\|_2$. The most commonly used Gaussian mechanism for DP releases $\mathcal{M}(D) = Q(D) + \xi$ with $\xi \sim \mathcal{N}(0, \sigma^2 I_d)$; by definition (ε, δ) corresponds to a noise scale defined by:

$$\sigma_{\varepsilon,\delta} \geq \frac{\sqrt{2\ln(1.25/\delta)}\Delta_2(Q)}{\varepsilon} \tag{4}$$

Some important properties of DP make differentially private training of machine learning models possible. Along with the definition of DP, these properties provide a quantitative privacy guarantee that is robust to arbitrary side information and does not depend on the details of a particular privacy attack. First, DP is closed under post-processing, which formalizes that once a privatized object is released, any further, potentially randomized transformation that does not look back at the raw data preserves the guarantee. For machine learning, this is of particular importance, since gradient descent optimizers often employ some form of post-processing, such as momentum and learning-rate schedules.

Property 1 (Post-processing) If $\mathcal{M}$ is (ε, δ)-differentially private and $\mathcal{T}$ is any (possibly randomized) map that does not access the raw dataset D, then $\mathcal{T} \circ \mathcal{M}$ is also (ε, δ)-differentially private.

Second, the composition rules ensure that when composed, private mechanisms applied to the same dataset still form a private mechanism, albeit with different privacy parameters. In iterative training approaches such as gradient descent, this is of utmost importance and allows for estimating privacy costs across training iterations.

Property 2 (Composition) If mechanisms $\mathcal{M}_1, \ldots, \mathcal{M}_T$ are applied to the same dataset and each $\mathcal{M}_t$ is $(\varepsilon_t, \delta_t)$-differentially private, then the composed mechanism is $\left(\sum_t \varepsilon_t, \sum_t \delta_t\right)$-differentially private.

One obvious drawback of direct composition is that for a fixed (ε, δ) privacy budget of the composed mechanism, at each step t one is forced to calibrate ε_t to the δ_t of that step, often chosen to be $\delta_t = \delta/T$, making each ε_t larger and loosening the total privacy guarantees ε. Sharper composition theorems exist, such as advanced composition, which allows a quadratically better scaling for ε at the expense of increased δ, but recent private machine learning pipelines typically use Rényi differential privacy [3] because it composes tightly and converts cleanly to (ε, δ). It is based on Rényi divergence defined on probability measures P and Q:

$$D_\alpha(P\|Q) \;=\; \frac{1}{\alpha - 1} \log \mathbb{E}_{x \sim Q}\left[\left(\frac{dP}{dQ}(x)\right)^\alpha\right]$$

for order $\alpha > 1$ whenever P is absolutely continuous with respect to Q.

Definition 3 (Rényi differential privacy) A mechanism $\mathcal{M}$ satisfies $(\alpha, \varepsilon(\alpha))$-RDP if, for all adjacent datasets D, D', the distributions $\mathcal{M}(D)$ and $\mathcal{M}(D')$ obey $D_\alpha(\mathcal{M}(D)\|\mathcal{M}(D')) \leq \varepsilon(\alpha)$.

In particular for the Gaussian mechanism, $\varepsilon = \frac{\alpha\Delta_t^2}{2\sigma_t^2}$. RDP also composes additively: if $\mathcal{M}_t$ are applied on the same data, with per-step RDP curves $\varepsilon_t(\alpha)$, then the composition satisfies $\varepsilon_{\text{tot}}(\alpha) = \sum_{t=1}^{T} \varepsilon_t(\alpha)$-RDP at order α. RDP converts to (ε, δ) via

$$\varepsilon(\delta) \;=\; \min_{\alpha>1} \left\{ \varepsilon_{\text{tot}}(\alpha) \;+\; \frac{\log(1/\delta)}{\alpha - 1} \right\}$$

Private machine learning routines harness RDP by using RDP-accountants, which maintain tight composition bounds across training epochs. They achieve this by keeping track of the curve $\varepsilon_{\text{tot}}(\alpha)$, without needing to choose δ_t for each training iteration and only converting back to (ε, δ)-DP for a given δ when reporting the privacy budget.

2.3 DP-SGD and gradient clipping

In many machine learning applications, the training set contains information about individuals, such as medical records, financial transactions, or user behavior logs, and a trained model can leak part of this information even when the raw data are never released. Such leakage can occur through model parameters, confidence scores, or repeated query access, and is commonly studied through attacks such as membership inference [26–28]. In this context, privacy of the training data can be ensured at different points in the model training and deployment pipeline as shown in Figure 1.

The most intuitive approach is to perturb the training data first and then rely on the DP post-processing Property 1. This can be useful in settings such as federated learning, where each client privatizes its local data before sending it to a central aggregator, which can be untrusted. This is the local-DP regime, which is stronger than the global DP setting considered here because it requires less trust in the aggregator training algorithm, but it also tends to reduce the accuracy of learned models more severely [1, 29, 30].

If training data itself was not privatized, its privacy can be ensured at the training stage or alternatively during inference. Here we focus on the former. In private training the entire training algorithm is the private mechanism, so once the model has been released, any subsequent inference query is only post-processing and does not consume additional privacy budget. By contrast, if one trains a non-private model and injects noise only at inference time, then privacy losses compose across prediction queries via Property 2, which makes that approach poorly suited for protecting training data in publicly accessible models. Inference-time noise can still be useful for other purposes, such as adversarial robustness [19, 31] but it is not the main object of this paper.

Differentially private training can be achieved by integrating randomized mechanisms into the training process so that a single entry cannot disproportionately affect the trained model. One possibility is to inject randomness directly at the model level or into the intermediate outputs of the model during training. In practice, however, because gradient computation is the last step of every optimization iteration, privacy

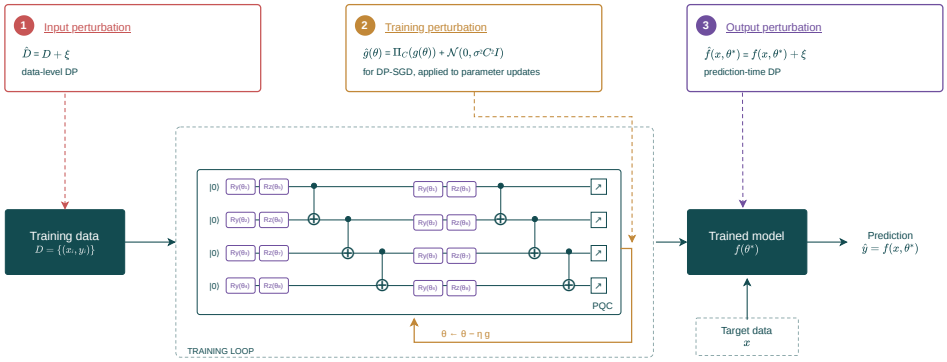

Fig. 1: Schematic overview of differential privacy mechanisms in machine learning pipelines, highlighting where privatization can be applied during training and inference. Noise can be injected at input (1), training (2) or output (3) levels. For private training, shown here is DP-SGD, applied to a training loop on a parametrized quantum circuit (PQC); however, training perturbation can happen differently and the underlying model can belong to any quantum or classical training paradigm.

composition is most naturally analyzed at the level of the loss gradients themselves, which also ensures the privacy of labels y_i, which enter the loss function directly. Privatizing the gradient updates directly is the primary principle behind differentially private stochastic gradient descent (DP-SGD) first proposed in [2], which has emerged as the standard optimizer in the private deep learning literature because it provides clean and tight privacy bounds with relatively little loss of utility and does not require strong assumptions on the whole model. DP-SGD augments ordinary stochastic gradient descent in three steps. First, it computes per-example gradients so that each training record can be controlled individually. Second, it clips those gradients to a norm threshold C to bound sensitivity. Third, it adds calibrated Gaussian noise to the clipped batch average. Everything that happens afterward is post-processing and therefore does not change the privacy guarantee.

The query to be privatized is the stochastic batch gradient query

$$Q(B) = g_B(\theta) = \frac{1}{|B|} \sum_{i \in B} g_i(\theta), \tag{5}$$

where the per-example stochastic gradients

$$g_i(\theta) := \partial_\theta \ell(f(x_i, \theta), y_i)$$

are an unbiased estimator of the population gradient

$$g(\theta) := \mathbb{E}[g_i(\theta)] = \partial_\theta \mathcal{L}(\theta), \tag{6}$$

A common way to model the stochasticity of per-example gradients is to write

$$g_i(\theta) = g(\theta) + \xi_i(\theta), \qquad \mathbb{E}[\xi_i(\theta)] = 0, \tag{7}$$

where $\xi_i(\theta)$ denotes the centered stochastic fluctuation around the population gradient.

For the following discussion, we will use notations $i \in B$ and $(x_i, y_i) \in B$ interchangeably. For two neighboring datasets $D \sim D'$, and neighboring batches $B \sim B'$ defined so that the differing entry is present in one batch but not the other, the sensitivity of this query is

$$\Delta_2(Q) = \sup_{B \sim B'} \|Q(B) - Q(B')\|_2 = \frac{1}{|B|} \sup_i \|g_i(\theta)\|_2 \tag{8}$$

Sensitivity directly affects the scale of the noise to be added in (4) and controlling it requires a bound on the per-example gradient norm. Since most standard models do not provide such a bound naturally, one clips each per-example gradient to have norm at most C:

$$\Pi_C(g_i(\theta)) := g_i(\theta) \min\{1, C/\|g_i(\theta)\|_2\} \tag{9}$$

and the sensitivity is then $\Delta_2(Q) = C/|B|$. The overall DP-SGD update over a batch B can be summarized as follows:

1. Sample a random batch $B \subset D$ and for every $i \in B$ compute the gradient $g_i(\theta) = \partial_\theta \ell(f(x_i, \theta), y_i)$.
2. Clip each gradient to the threshold C: $\bar{g}_i(\theta) = \Pi_C(g_i(\theta))$.
3. Aggregate the clipped gradients over the batch: $\bar{g}_B(\theta) = \frac{1}{|B|} \sum_{i \in B} \bar{g}_i(\theta)$.
4. Add Gaussian noise: $\widehat{g}_B(\theta) = \bar{g}_B(\theta) + Z$, where $Z \sim \mathcal{N}(0, \frac{\sigma^2 C^2}{|B|^2} I_p)$.
5. Update the parameters using $\widehat{g}_B(\theta)$ as the gradient estimate.

This gradient descent loop then repeats for T steps. One can express the noise multiplier σ required by the full mechanism in terms of the batch subsampling rate $q = |B|/N$, the number of steps T, and the target privacy level (ε, δ). In particular, there exist constants $c_1, c_2 > 0$ such that, for any $\varepsilon < c_1 q^2 T$ and $\delta > 0$, it suffices to choose

$$\sigma \geq c_2 \frac{q\sqrt{T \log(1/\delta)}}{\varepsilon}, \tag{10}$$

which yields (ε, δ)-DP for the mechanism composed across T steps.

The important point for optimization is that under a fixed privacy budget (ε, δ) the noise scale depends on batch size $|B|$, clipping threshold C and the noise multiplier σ, which in turn depends on subsampling rate $q = |B|/N$ and number of steps T. Intuitively, larger batches average more per-example gradients, which reduces the variance of the stochastic gradient estimate; however, large $|B|$ also increases the sampling rate q, which enters the privacy accounting in (10) and can force either fewer steps T (early stopping) or a larger noise multiplier σ. The expression for σ is different for RDP accountants which admit sharper subsampling formulas for Poisson or fixed-size batch sampling; however, the practical dependence on the sampling rate q remains [3, 4]. In practice, $|B|$ therefore trades off per-step utility (better updates) against per-step privacy cost (faster privacy budget spending).

Likewise, clipping bounds sensitivity, but it also sets the added-noise scale. This makes the role of C twofold: decreasing C reduces both the sensitivity and the noise scale, but it also increases the risk of clipping and therefore of bias, inherent to clipping in the stochastic setting. To compare the impact of DP-SGD on training and gradient updates for different models under identical clipping and privacy parameters, a convenient quantity is the mean-squared error (MSE) of the stochastic update with respect to the unaltered population gradient, which captures both clipping bias and DP noise. It can be written as

$$\begin{aligned}
\mathrm{MSE}(\theta) &= \mathbb{E}\big[\|\widehat{g}_B(\theta) - g(\theta)\|_2^2\big] \\
&= \mathbb{E}\big[\|\bar{g}_B(\theta) + Z - g(\theta)\|_2^2\big] \\
&= \mathbb{E}\big[\|\bar{g}_B(\theta) - g(\theta)\|_2^2\big] + 2\,\mathbb{E}\big[\langle \bar{g}_B(\theta) - g(\theta), Z\rangle\big] + \mathbb{E}\,\|Z\|_2^2 \\
&= \mathbb{E}\big[\|\bar{g}_B(\theta) - g(\theta)\|_2^2\big] + p\Big(\frac{\sigma C}{|B|}\Big)^2 \\
&= \mathrm{Tr}\big(\mathrm{Cov}[\bar{g}_B(\theta)]\big) + \|\,\mathbb{E}[\bar{g}_B(\theta)] - g(\theta)\|_2^2 + p\Big(\frac{\sigma C}{|B|}\Big)^2
\end{aligned} \tag{11}$$

where we used that Z is independent of the minibatch and centered, so the cross term vanishes and $\mathbb{E}\,\|Z\|_2^2 = p(\sigma C/|B|)^2$. Under i.i.d. sampling

$$\mathrm{Cov}(\bar{g}_B(\theta)) = \frac{1}{|B|}\,\mathrm{Cov}(\bar{g}_i(\theta)), \qquad \mathbb{E}[\bar{g}_B(\theta)] = \mathbb{E}[\bar{g}_i(\theta)]$$

Hence the same identity can be written equivalently as

$$\mathrm{MSE}(\theta) = \frac{1}{|B|}\,\mathrm{Tr}\big(\mathrm{Cov}[\bar{g}_i(\theta)]\big) + p\Big(\frac{\sigma C}{|B|}\Big)^2 + \|\,\mathbb{E}[\bar{g}_i(\theta)] - g(\theta)\|_2^2 \tag{12}$$

The first term corresponds to the sampling variance which always appears under stochastic minibatch descent. The second term is the variance of the added noise and is a result of the noise addition mechanism in DP-SGD, while the third term is the squared norm of the clipping bias

$$b_C(\theta) := \mathbb{E}\big[\bar{g}_i(\theta) - g_i(\theta)\big] = \mathbb{E}[\bar{g}_i(\theta)] - g(\theta) \tag{13}$$

where the expectation is taken over the data distribution. The concept of clipping bias is inherent to clipped stochastic gradient descent, and it measures how much clipping shifts the expected update away from the unclipped gradient field before any privacy noise is added. Note that, unlike the other terms in (12), the clipping bias does not improve with increasing batch size. Previous literature explored the harmful effects of clipping bias, which can accumulate into a non-vanishing optimization error floor [5–7]. Under clipping, the stochastic optimizer converges only to a neighborhood of a stationary point, with the size of that neighborhood controlled by the bias. But the magnitude of the bias alone is not sufficient: for optimization it also matters whether clipping preserves the descent direction. This can be measured using

directional alignment

$$A_C(\theta) := \frac{\langle \bar{g}_B(\theta),\, g(\theta)\rangle}{\|\bar{g}_B(\theta)\|_2\, \|g(\theta)\|_2} \in [-1, 1] \tag{14}$$

When $A_C(\theta) < 1$, stochastic clipping rotates the mean update away from the true gradient direction, which is a harmful regime for optimization. When per-example clipping is minimal, we have $\|b_C(\theta)\|_2 \approx 0$ and $A_C(\theta) \approx 1$. However, the directional alignment can still be favorable for optimization, $A_C(\theta) \approx 1$, even when clipping is happening aggressively, but only causing a rescaling of the expected update. As shown in [5], the compound effect of gradient clipping on directional alignment throughout training is determined by the symmetry of the per-example gradient distribution around the true population mean, that is the distance of ξ_i in (7) from the closest geometrically symmetric reference distribution centered at 0. In the perfectly symmetric case, the asymmetry-induced directional misalignment caused by clipping disappears. When ξ_i is only approximately symmetric, it can still affect the training in the earlier stages of optimization, even if to a lesser extent for the impact on the convergence and, therefore, eventual accuracy. As a workaround, the authors propose adding noise to the per-example gradients before clipping, which has this desirable symmetrizing effect:

$$\widehat{\partial_\theta \ell} = \partial_\theta \ell + \mathcal{N}(0, v) \tag{15}$$

Thus, the central question in what follows is not only how much Gaussian noise DP-SGD adds, but also how much clipping distorts the stochastic gradient before that noise is added. Boundedness of quantum model gradients can be favorable because it makes it easier to control the clipping bias.

3 Results

We work in a supervised K-class classification setting with features $x \in \mathbb{R}^n$ and labels $y \in \{1, \ldots, K\}$. In the quantum pipeline, encoding produces a state $\rho(x) = |\phi(x)\rangle\langle\phi(x)|$. For an entry (x, y) a parameterized circuit

$$U(\theta) = \prod_{j=1}^{p} e^{-i\theta_j H_j}$$

with parameter vector $\theta \in \mathbb{R}^p$ acts on $\rho(x)$, and bounded observables O_y are measured to define the logits

$$f_y(x, \theta) = \mathrm{Tr}\big(O_y\, U(\theta)\, \rho(x)\, U(\theta)^\dagger\big) \tag{16}$$

For a temperature $\tau > 0$, define the temperature-scaled softmax probabilities

$$p_y(x, \theta) = \mathrm{softmax}_\tau(f_y(x, \theta)) = \frac{\exp(f_y(x, \theta)/\tau)}{\sum_{k=1}^{K} \exp(f_k(x, \theta)/\tau)},$$

and the cross-entropy loss

$$\ell(f(x,\theta),y) = -\log p_y(x,\theta)$$

Rewriting the cross-entropy loss as

$$\ell(f(x,\theta),y) = -\frac{f_y(x,\theta)}{\tau} + \log\left(\sum_{r=1}^{K} e^{f_r(x,\theta)/\tau}\right)$$

and differentiating with respect to θ_j gives

$$\frac{\partial \ell}{\partial \theta_j} = \frac{1}{\tau}\left(\frac{\sum_{k=1}^{K} e^{f_k/\tau}\,\frac{\partial f_k}{\partial \theta_j}}{\sum_{r=1}^{K} e^{f_r/\tau}} - \frac{\partial f_y}{\partial \theta_j}\right)$$

which simplifies to the closed-form loss-objective gradient relation

$$\frac{\partial \ell}{\partial \theta_j} = \frac{1}{\tau}\left(\sum_{k=1}^{K} p_k(x,\theta)\,\frac{\partial f_k(x,\theta)}{\partial \theta_j} - \frac{\partial f_y(x,\theta)}{\partial \theta_j}\right), \tag{17}$$

which we will refer to in what follows.

In our quantum models noiseless gradients are computed using backpropagation, while simulation of shot noise necessitates the application of parameter-shift rules [32] instead. In the classical baseline we assume the same loss and dataset, a comparable number of trainable parameters, and hyperparameters tuned separately for each setting. Differential privacy is then enforced using standard DP-SGD with subsampled Rényi accounting.

Below we proceed with an analysis of how shot noise and hardware noise propagate to loss gradients. This reveals why such quantum noise can look superficially similar to the Gaussian perturbation used in DP-SGD, but also why that resemblance is incomplete from a privacy perspective. We then introduce the structural ingredient that DP-SGD actually needs in the hybrid setting: a deterministic clipping bias bound for the loss gradients used in training. We finally return to the original noise question and explain why native quantum noise is not a satisfactory replacement for the formal privacy mechanism.

3.1 Noise propagation

There are several sources of randomness in private optimization that should be distinguished. Stochastic mini-batching affects per-example gradients before clipping and, unlike the Gaussian perturbation added after averaging over a batch, does not itself provide differential privacy. In addition to the artificial DP-SGD noise, quantum implementations also involve additional sources like shot noise and hardware noise. The purpose of this subsection is to understand how those quantum sources of randomness propagate to the loss gradients, and to clarify why they should be treated as secondary to the formal DP mechanism.

The idea of using quantum noise as a resource rather than a nuisance has been explored previously. In the privacy context, recent work has shown that quantum noise can protect quantum classifiers by providing adversarial robustness and inference-stage privacy [19]. Those guarantees, however, are formulated for neighboring quantum inputs or query-level perturbations, not for the dataset-level adjacency relevant to training-time privacy against membership inference. Thus, it is useful to propagate the effect of quantum noise to the end of the training iteration, in order to understand how it affects training-time privacy and where the analogy with DP-SGD breaks down.

Theorem 1 (Quantum noise propagation to loss gradients) *Consider a K-class quantum classification model with logits*

$$f_k(x,\theta) = \mathrm{Tr}\big(O_k\,\rho(\theta)\big)$$

for traceless Pauli observables O_k, and cross-entropy loss ℓ. Under global depolarizing noise

$$\mathcal{E}_\lambda(\rho) = (1-\lambda)\rho + \lambda\,\mathbb{1}/d$$

and finite-shot estimation with shot count S, the induced perturbation of the loss gradient can be approximated as additive Gaussian noise:

$$\widehat{\partial_\theta \ell} \approx \partial_\theta \ell + \mathcal{N}(m, v), \tag{18}$$

where the mean m and variance v are computable via 21 from the logits, their gradients, the noise strength λ, and the shot count S.

The proof with the exact expressions for m and v can be found in Appendix A. Theorem 1 shows that shot noise and hardware noise can, after propagation through the loss, approximately resemble an additive Gaussian perturbation on the gradients. The approximation is reasonable when $\|\xi\|/\tau$ is not too large, which is the case for small λ and large S or τ. However, this resemblance is not enough for a DP-SGD-style training guarantee.

Without an appropriate bound on $|\partial_\theta \ell|$, the sensitivity of gradient queries remains unbounded, necessitating clipping. In this case, note how when only shot noise is considered in the absence of depolarizing noise, we have $m = 0$ and (18) is identical to (15). As discussed, gradients of this form have nice geometric properties, which can reduce the clipping bias throughout the training and improve the utility under identical privacy budgets, by symmetrizing the pre-clipping gradient distribution so that the clipping step deforms the expected update less severely. Nevertheless, improvements from such a technique are expected to be marginal under reasonable assumptions on the scale of hardware noise and they can be easily simulated using a classical mechanism, as described in [5].

Alternatively, in order to obtain a formal privacy guarantee, one would still need a deterministic control of gradient sensitivity, that is a bound on $\partial_\theta \ell$ in (18). This motivates the following result.

3.2 Clipping bias bound

We now turn to the structural property of quantum models that makes their training under DP-SGD more favorable. Unless indicated otherwise by a subscript the norms are operator norms and we will use $(\cdot)_+ = \max\{0, \cdot\}$ for better readability.

Theorem 2 (Deterministic clipping-bias bound) *Consider the variational classifier of Definition 1 with K class observables $\{O_k\}_{k=1}^K$, parameterised unitary $U(\theta) = \prod_{j=1}^p e^{-i\theta_j H_j}$, and cross-entropy loss at temperature τ. Set*

$$G \;:=\; 2 \max_{1\le k\le K} \|O_k\| \left(\sum_{j=1}^{p} \|H_j\|^2 \right)^{1/2}$$

Then for every data point (x_i, y_i) in the dataset the per-example loss gradient $g_i(\theta) = \partial_\theta \ell(f(x_i,\theta), y_i)$ satisfies

$$\|g_i(\theta)\|_2 \;\le\; \frac{2G}{\tau} \tag{19}$$

In particular, for every clipping threshold $C > 0$, the clipping bias $b_C(\theta)$ defined in (13) satisfies

$$\|b_C(\theta)\|_2 \;\le\; \left(\frac{2G}{\tau} - C \right)_+ \tag{20}$$

Proof is postponed to Appendix A. Theorem 2 shows that quantum models may retain more utility under DP-SGD because they admit deterministic structural control of clipping bias. The bound is deterministic, model-level, and computable directly from the circuit description. No assumption on empirical gradient tails is needed. The bound is also uniform over the encoded input: the proof uses only that the encoded datum enters as a quantum state, so the same bound holds for amplitude, angle, basis, or data-reuploading encodings, provided the trainable parameters appear only in the variational unitary layers. It is likewise independent of the gradient computation method, as long as it is exact, since this structural bound only depends on the gradient itself.

The theorem provides a direct route from the model structure to the actual DP-SGD update used in training: the object clipped by the optimizer is the per-example loss gradient $g_i(\theta)$, and its distortion is controlled explicitly by the circuit generators and observable norms. In particular, for $C = 2G/\tau$ clipping bias vanishes, so once a deterministic gradient bound is available, clipping can be tuned structurally rather than only heuristically. Smaller clipping thresholds can then be used without inducing a large clipping bias, and this directly lowers the scale σC of the Gaussian noise added by DP-SGD. In practice one wants observables O_k and generators H_j to be local or low-weight, and circuits to remain sufficiently shallow, so that the norms entering (22) do not scale unfavorably. If one instead uses highly nonlocal observables, large-support generators, or overly deep circuits, the worst-case constant can become loose enough to be uninformative for clipping calibration even though the theorem remains correct. Thus, locality is not needed for validity, but it is important for tightness and therefore for practical usefulness in private training.

Notably, this locality constraint also overlaps with the trainable regime for the models considered here. As discussed previously, models with non-local generators or observables suffer from barren plateaus and in this case the main problem is no longer the clipping bias but the vanishing gradient signal. In the trainable setting, the model also remains classically simulable, which is however not a problem by itself, as the comparison criterion is the accuracy loss under clipped (and therefore private) training, rather than expressivity or computational efficiency. This is related in spirit

to Lipschitz-constrained architectures, which trade clipping bias for bias in model space by inducing gradient control as an architectural constraint built into the model class [16]. For quantum models the control is inherited from unitary evolution together with bounded expectation values.

We can now circle back to the original motivation. Theorem 1 shows that, after propagation through the loss, shot noise and hardware noise can appear as an additive Gaussian perturbation on gradient coordinates. However, this is not enough to replace the calibrated Gaussian mechanism of DP-SGD. The first obstacle is that the variance terms σ_ξ^2, $\sigma_{\zeta,j}^2$, and therefore the total variance v, are data dependent. Differential privacy requires noise calibration that is controlled independently of the particular training example, whereas the shot-noise variance can collapse near the extreme expectation values ± 1. Introducing additional depolarizing noise can prevent the variance from vanishing at the extremes, but at the price of extra gradient bias and potentially worse trainability. The second obstacle is that in its most general form the bound in Theorem 2 can be too loose to yield competitive privacy budgets when inserted into a Gaussian-mechanism analysis. Trying to lower the budget by decreasing the number of shots also increases noise and hurts trainability or expressivity [25, 33, 34], and can invalidate the assumption needed for Gaussian approximation.

For these reasons, using native quantum noise alone to derive record-level privacy guarantees leads to bounds that are too loose for practical training scenarios. Quantum noise can still act as an additional source of randomness and may therefore help against empirical privacy attacks, but the formal privacy guarantee should continue to come from classical DP-SGD.

4 Numerical simulations

The setup for our numerical simulations is directly affected by the analytical insights from the previous sections. Since we are interested in comparing quantum and classical models, it is important to set up experiments that allow for a fair comparison while taking into account the metrics we have discussed. The experiments are organized to test the clipping bias explanation by asking two questions: First, at a fixed clipping threshold, do quantum models exhibit lower clipping frequency and reduced clipping bias than matched classical baselines? Second, under actual private training with the same DP-SGD mechanism, do they retain more utility?

Due to the differences between quantum and classical models, one needs to define equivalence between them to enable a fair comparison.

Definition 4 (Equivalence between quantum and classical models (informal)) Quantum and classical models are equivalent if they contain approximately the same number of parameters p and, under the same optimizer, fine-tuned to each model separately, achieve similar accuracy on the same task.

Note that this differs from most approaches in the literature in that accuracy is not a consequence of model dimension. Rather, since we are interested in the performance drop of the models when going from non-private to private settings, we design them

to have equivalent performance at the same number of parameters when privacy-enhancing techniques are not employed. This practical definition of equivalence allows for comparison of models in real use-case settings.

The important nuance is that the number of parameters can affect the convergence rates so it has to stay the same between classical and quantum models. Since DP-SGD privatizes a gradient vector in $\mathbb{R}^p$, the injected Gaussian perturbation is also p-dimensional; hence, while the per-coordinate variance is fixed by (σ, C, B), the total noise scales as $\mathbb{E}[\|\xi_t\|_2^2] = p\,\sigma^2 C^2/|B|^2$, so $\mathbb{E}[\|\xi_t\|_2] \approx (\sigma C/|B|)\sqrt{p}$, degrading the update MSE (12) as p grows. Moreover, under a fixed privacy budget (ε, δ) and sampling rate $q = |B|/N$, composition limits the number of steps T; larger models often require larger T to train, which forces either larger σ (more noise per step) or early stopping. The parameter count also affects DP training through clipping bias. Clipping acts on the full per-example gradient vector, and the clipping probability $\Pr[\|g_i\|_2 > C]$ typically increases with dimension at a fixed threshold C, since if coordinates have comparable scale then $\|g_i\|_2$ tends to grow like $\sqrt{p}$. Hence larger p can lead to a higher fraction of clipped gradients and a larger bias in the direction and magnitude of the update, unless C is increased.

Fixing the number of parameters and the optimization routine, including the number of iterations, allows us to concentrate on a few tunable hyperparameters. When switching to DP training, it is important to ensure that the additional hyperparameters involved in DP-SGD are properly tuned across the models.

The clipping-only study shows that under these equivalence conditions, quantum models preserve bounded gradients and reach higher test accuracy with smaller clipping thresholds. This in turn affects the amount of necessary noise in privacy preserving simulations, allowing for better accuracy under a fixed privacy budget, and therefore a more optimal privacy-utility tradeoff. Finally, we employ private training in a quantum image loading and classification pipeline, showing how it can outperform an equivalent convolutional neural network. All the implementation details and results discussed in this section can be found in the companion Github repository [35].

4.1 Effect of Clipping

In order to understand the detrimental effect of clipping, we first consider experiments where privacy is not enforced. This means that the Gaussian noise in DP-SGD is not added, which allows us to isolate the effects of clipping. Such a setting has no practical meaning of its own, but it enables a cleaner numerical study of clipping effects. For the same reason we choose a synthetic tabular classification task with $N = 2048$ samples, $K = 4$ classes and 16 features. Having strict control over the classification dataset lets us avoid outliers that could bias the results. The dataset is not perfectly separable, as some of the labels are randomly flipped to ensure non-triviality of the task.

As a classical model we take a neural network with several fully connected layers. To maximize the performance of the classifier under private training, we choose tanh functions as the activation for each layer. As discussed in [13], the family of tempered sigmoid activations, and in particular tanh, tends to outperform unbounded activations such as ReLU in private training context. The quantum classifier ansatz is built using amplitude encoding and L layers of $R_Z R_Y R_Z$ followed by a CNOT ring,

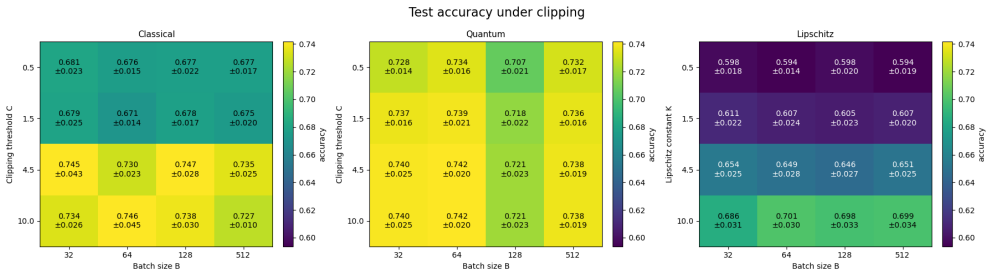

Fig. 2: Final achieved test accuracy depending on the clipping threshold C and the batch size $|B|$. For each configuration the learning rate is fine-tuned to ensure stable training with the highest achievable accuracy in 100 training epochs. With appropriate tuning of the learning rate, batch size does not noticeably impact the accuracy. However, the bias floor causes the classical models with larger gradients to struggle under smaller clipping threshold C.

with Pauli-Z expectation read-outs on the first K qubits similar to Figure 1. We also include a comparison with classical Lipschitz neural networks, structured similarly to classical neural networks, but utilizing gradient norm preserving (GNP) layers instead with Lipschitz constant matching the clipping threshold C. As discussed previously, Lipschitz neural networks avoid clipping bias altogether, but they induce bias in model space, limiting the expressivity of the model.

We sweep across several values of the clipping threshold C and the batch size $|B|$, and for every configuration we perform 10 experiments for 100 training epochs with distinct random seeds on parameter initialization, reporting mean test accuracy $\pm$ one std. across the seeds. The results are shown in Figure 2. As can be seen, the batch size does not significantly affect training since the optimizer is tuned for each configuration separately. Moreover, the models achieve similar accuracy at a large clipping threshold of $C = 10$, which corresponds to unclipped or minimally clipped training (for Lipschitz models this corresponds to less restrictive models with more expressivity). However, the classical models clearly struggle for clipping thresholds $C \leq 1.5$, indicating detrimental effects of clipping bias.

To better understand this phenomenon, we dive deeper into the training configuration where classical and quantum models diverge, with $C = 1.5$ and $|B| = 64$ for quantum and classical models only (since Lipschitz networks do not suffer from clipping bias). We benchmark four models: classical, quantum, clipped classical and clipped quantum. After each epoch, in addition to model accuracy and distance from the unclipped loss, we compute a diagnostic snapshot of the gradient distribution (for clipped models we use the pre-clipping distribution at the end of every epoch). Based on the complete per-example gradient distribution, we compute and report several metrics. First, we analyse the true population gradient $\|g\| = \|\partial_\theta \mathcal{L}\|$, which shrinks monotonically as expected, indicating convergence in Figure 3. The only exception is the clipped classical model, for which the population gradient decreases initially but eventually plateaus. At the same time, the per-example gradient does not necessarily decrease. Looking at the mean and maximum per-example gradient norms $\mathbb{E}_i \|g_i\|$

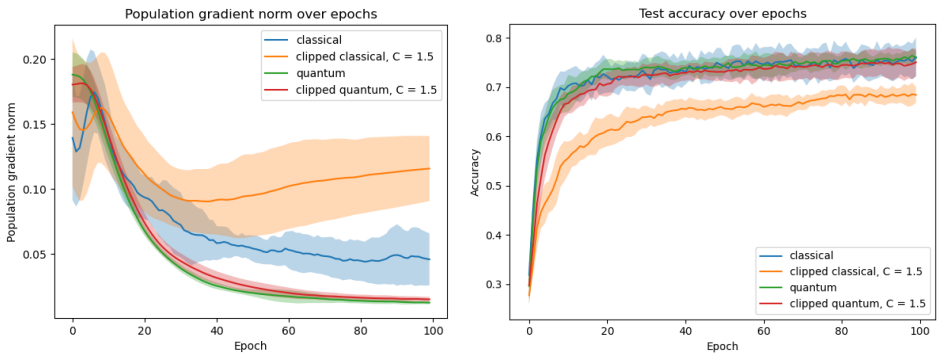

Fig. 3: Evolution of population gradient and test accuracy for classifiers on synthetic data

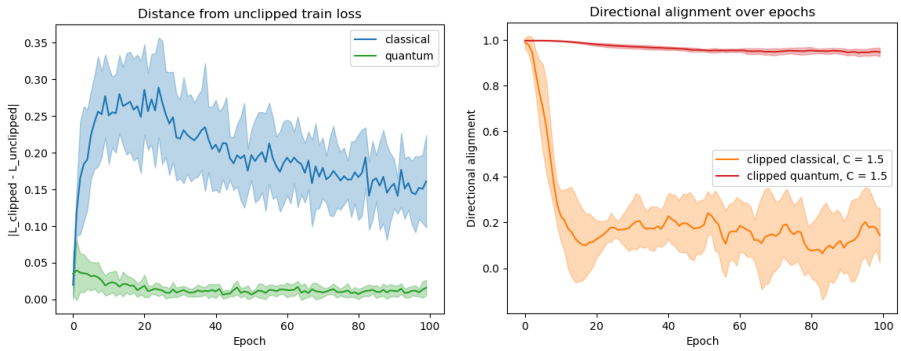

Fig. 4: Evolution of distance from the unclipped loss $|\mathcal{L}_{\text{clipped}} - \mathcal{L}_{\text{unclipped}}|$ and directional alignment for classifiers on synthetic data

and $\max_i \|g_i\|$ in Figure 5, it becomes clear that a clipping threshold of $C = 1.5$ is optimal for quantum models on average and worst-case, but it is too small for classical ones, which results in a large clipping probability $p_{\text{clip}} = \Pr[\|g_i\| > C]$ (Figure 6) and increased clipping bias preventing convergence to the optimum. Importantly, clipping here significantly affects the directional alignment in Figure 4, which prevents the optimizer from converging. Together these results indicate that quantum model performs better due to the reduced clipping bias induced by bounded gradients.

4.2 Private training

Naturally, we also consider more practical settings where privacy is enforced with noise. Extending the results from the previous section, we perform a further analysis for models trained under a fixed privacy budget. Again sweeping across the clipping threshold and privacy budget values and reporting mean $\pm$ one standard deviation over 10 random initializations, we see a similar pattern here: quantum models perform better in general with smaller clipping thresholds, as shown in Figure 7. The reason is

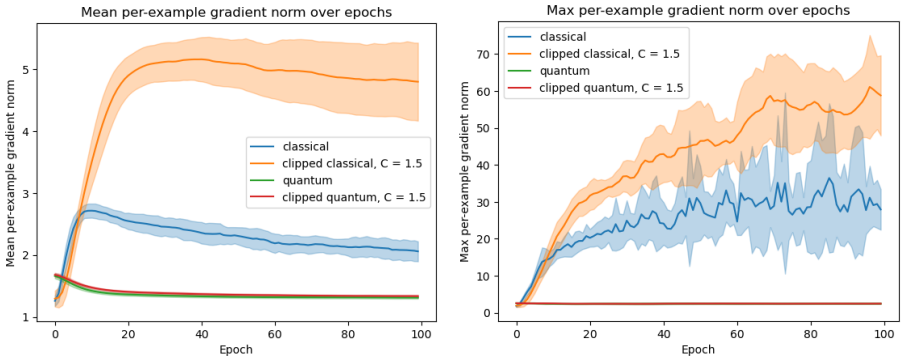

Fig. 5: Evolution of mean and maximum gradient norm for classifiers on synthetic data

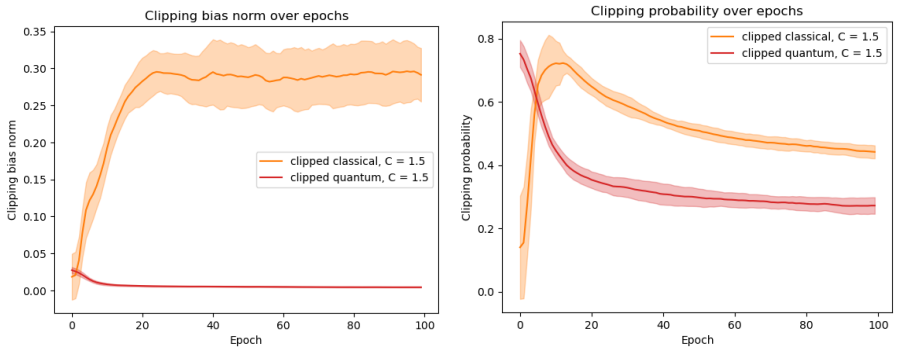

Fig. 6: Evolution of clipping bias norm and clipping probability for classifiers on synthetic data

that small C also reduces the amount of noise σC needed at every iteration to comply with a fixed privacy budget. We take a deeper look at this phenomenon for $C = 1.5$ and $\varepsilon = 10$ and compute the most relevant quantity that captures the effects of both clipping and Gaussian noise – MSE (12). We plot its evolution throughout training alongside test accuracy in Figure 8.

Finally, we apply these results to an image classification pipeline on a real dataset. To test the performance under private training on complex images, we take images from the Honda Scenes dataset [36], which is a labeled dataset designed for dynamic scene classification. This dataset features time-based annotations detailing various aspects such as road locations, surrounding environment, weather conditions, and the state of the road surface. We sample 2000 images and classify between four classes corresponding to weather conditions and time of day: clear day, clear evening, snowy day, and snowy evening. We scale down the dimensions of images to 64×96 and convert them to gray-scale for simplicity. The classical network is a convolutional neural network with max-pooling and convolution layers, along with GroupNorm layers.

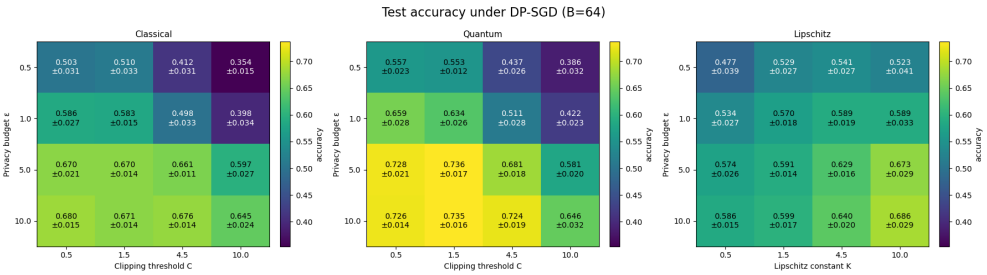

Fig. 7: Final achieved test accuracy depending on the clipping threshold C and the privacy budget ε for batch size $|B| = 64$. For each configuration the learning rate is fine-tuned to ensure stable training with the highest achievable accuracy in 100 training epochs. Smaller clipping thresholds reduce the added noise scale but induce clipping bias, and the impact of this double effect is visible in the figure. The optimal results are achieved when the clipping threshold tightly bounds the gradient without clipping too much, so that the scale of the added noise is reduced to a minimum. For the quantum model $C = 1.5$ ensures the best results, beating the best classical equivalent by around 5% increase in accuracy. As expected, the accuracy drops regardless of the clipping threshold under tighter privacy budgets when more calibrated noise is being added. Note that the colorscheme matches that of Figure 2, but the colorscale range differs to better highlight the differences.

Quantum image classifiers are based on a previously proposed loading and classification pipeline described in [37], where there is no classical feature extraction before the quantum processing. As a first step in this pipeline quantum representations of images are learned using a technique known as hierarchical learning, which first learns coarse grained representations of images with a lower resolution, gradually scaling them up and learning more fine grained representations. Once trained, the loader parameters are fixed and loaders are then merged with the variational classifier, which outputs Pauli-Z expectation values corresponding to image classes. The variational ansatz has L variational layers of the same form as the one used for synthetic dataset. A schematic representation of the training pipeline is shown on Figure 9.

The equivalence conditions are ensured for both models, with each having around 500 trainable parameters and both achieving the same accuracy under unconstrained training. As seen in Figure 10 under private training we see that the quantum classifier achieves higher accuracy and remains closer to the non-private unclipped optimization landscape, indicating the practical consequences of the bounded nature of its gradients in a private learning tasks applied to a real dataset.

5 Conclusion

In our work we analysed the propsects of privacy advantage in QML pipelines. Our noise propagation argument shows that shot noise and hardware noise can induce approximately Gaussian perturbations on loss gradients, but this resemblance is not enough for practical training-time differential privacy because the resulting variance

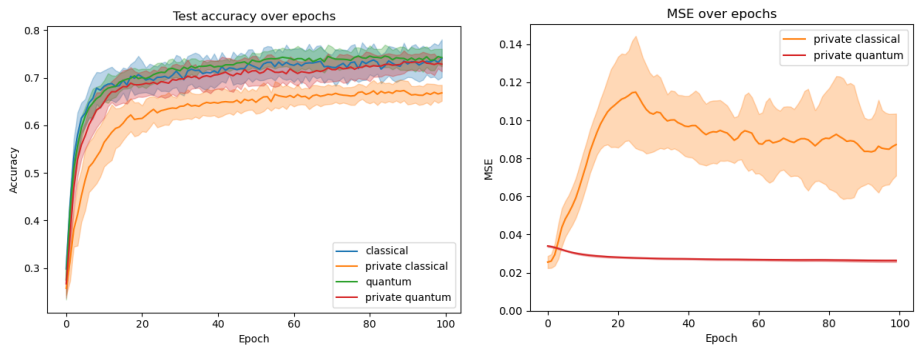

Fig. 8: Evolution of accuracy and MSE for private classifiers on synthetic data with clipping threshold $C = 1.5$ and practical privacy budget of $\varepsilon = 10$ and $\delta = 10^{-5}$. MSE aggregates the effects of both clipping and calibrated noise on gradient updates. As can be seen the private classical model underperforms due to the higher MSE of gradient updates.

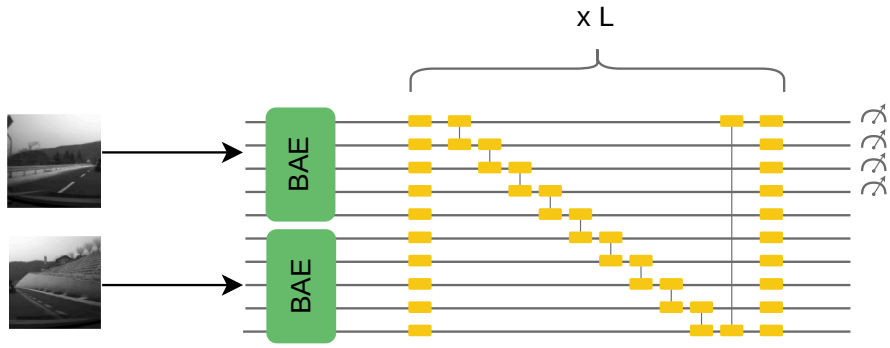

Fig. 9: Quantum image loading and classification model considered here. The block amplitude encoding (BAE) image loaders are trained in advance using hierarchical training to learn representations of image blocks as quantum states. These are then plugged into the pipeline for each individual image and the L-layer variational classifier outputs Z expectation values, which are used to train the model. Figure inspired by the ones found in [37].

is data dependent and the corresponding worst-case sensitivity control is too loose to serve as a competitive privacy mechanism on its own.

This observation motivated the structural result on clipping bias bound. For the class of local variational quantum models considered here, unitary evolution together with bounded observables yields deterministic control of per-example gradient norms. Through the softmax-cross-entropy link, this control transfers to the loss gradients actually used by DP-SGD, and therefore gives an explicit upper bound on clipping bias. In trainable regimes with sufficiently local observables and generators, this explains

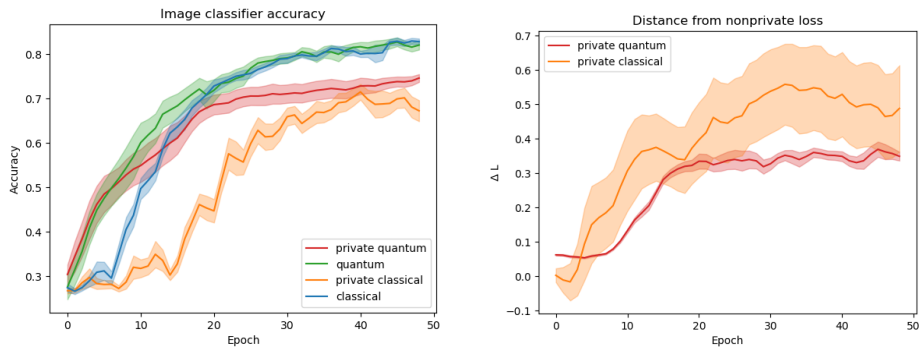

Fig. 10: Evolution of accuracy and distance from the non-private loss for image classifiers under private training with $\varepsilon = 10.0$, $\delta = 10^{-5}$ and $C = 1.5$. Under the clipping and noise constraints posed by the private optimizer, the quantum model achieves consistently higher classification accuracy and manages to stay closer to the unbiased optimisation landscape throughout training. The curves represent the average and standard deviation over 10 random initializations for the models.

why quantum models can admit smaller effective clipping distortion than matched classical baselines under the same privacy mechanism. Moreover, as verified by our numerical results, classical Lipschitz networks with structurally imposed bounded gradients struggle in regimes where quantum models do not. At the same time, reduced clipping bias in quantum models allows them to outperform classical alternatives on synthetic and real-world tasks under private training. Our model equivalence criteria suggest that private training is a particularly natural setting for comparing quantum models with classical ones.

The resulting picture is therefore twofold. Formal record-level privacy should still come from classical DP-SGD with clipping and calibrated Gaussian noise added after aggregation. Native quantum noise is not a replacement for that mechanism; however, reduced clipping bias makes quantum models better suited to private training using DP-SGD. This also suggests a broader quantum-inspired design principle to be explored further in future work: if one designs classical architectures with analogous norm-preserving parameterizations and bounded activation/output layers, then part of the same clipping-bias advantage transfers beyond quantum hardware.

While the quantum models considered here are completely classically simulable, rapid advances in quantum hardware and the approaching era of fault tolerant quantum computing (FTQC) will allow quantum-native variational learning algorithms to benefit from the privacy results presented here. In this regime local quantum variational models will stay trainable due to the absence of noise, while still retaining the desirable boundedness property.

6 Statements and Declarations

Funding: We acknowledge funding from Agence Nationale de la Recherche (Plan France 2030 through the project OQuLus ANR-22-PETQ-0013)

Conflicts of interest: We declare no conflicts of interest related to this work.
Data Availability: Data and code used to construct plots and tables in this work can be obtained from the companion Gihub repository [35].
Author Contributions: T.S. initiated the project, contributed to the theory and simulation results, wrote the simulation and data aggregation code, and edited the manuscript. F.G. and E.K. contributed to the idea, edited the manuscript, and supervised the work.

References

[1] Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci. **9**(3–4), 211–407 (2014) https://doi.org/10.1561/0400000042

[2] Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 308–318 (2016). https://doi.org/10.1145/2976749.2978318

[3] Mironov, I.: Rényi differential privacy. 2017 IEEE 30th Computer Security Foundations Symposium (CSF), 263–275 (2017) https://doi.org/10.48550/arXiv.1702.07476 arXiv:1702.07476

[4] Wang, Y.-X., Balle, B., Kasiviswanathan, S.: Subsampled rényi differential privacy and analytical moments accountant. Journal of Privacy and Confidentiality **10**(2) (2021) https://doi.org/10.29012/jpc.723

[5] Chen, X., Wu, Z.S., Hong, M.: Understanding gradient clipping in private sgd: a geometric perspective. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS '20. Curran Associates Inc., Red Hook, NY, USA (2020)

[6] Koloskova, A., Hendrikx, H., Stich, S.U.: Revisiting gradient clipping: Stochastic bias and tight convergence guarantees. In: Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J. (eds.) Proceedings of the 40th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 202, pp. 17343–17363. PMLR, Honolulu, Hawaii, USA (2023). https://proceedings.mlr.press/v202/koloskova23a.html

[7] Pichapati, V., Suresh, A.T., Yu, F.X., Reddi, S.J., Kumar, S.: AdaCliP: Adaptive Clipping for Private SGD (2019). https://arxiv.org/abs/1908.07643

[8] Zaman, K., Marchisio, A., Hanif, M.A., Shafique, M.: A Survey on Quantum Machine Learning: Current Trends, Challenges, Opportunities, and the Road Ahead (2023) arXiv:2310.10315

[9] Chang, S.Y., Cerezo, M.: A Primer on Quantum Machine Learning (2025) arXiv:2511.15969

[10] Watkins, W.M., Chen, S.Y.-C., Yoo, S.: Quantum machine learning with differential privacy. Scientific Reports **13**, 2023 (2023) https://doi.org/10.1038/s41598-022-24082-z arXiv:2103.06232

[11] Rofougaran, R., Yoo, S., Tseng, H.-H., Chen, S.Y.-C.: Federated quantum

machine learning with differential privacy. In: ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 9811–9815 (2024). https://doi.org/10.1109/ICASSP48485.2024.10447155

[12] Li, Y., Zhao, Y., Zhang, X., Zhong, H., Pan, M., Zhang, C.: Differential privacy preserving quantum computing via measurement noise (2023) arXiv:2312.08210

[13] Papernot, N., Thakurta, A., Song, S., Chien, S., Erlingsson, Ú.: Tempered sigmoid activations for deep learning with differential privacy. Proceedings of the AAAI Conference on Artificial Intelligence **35**(10), 9312–9321 (2021) https://doi.org/10.1609/aaai.v35i10.17123

[14] Arjovsky, M., Shah, A., Bengio, Y.: Unitary evolution recurrent neural networks. In: Balcan, M.F., Weinberger, K.Q. (eds.) Proceedings of The 33rd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 48, pp. 1120–1128. PMLR, New York, New York, USA (2016). https://proceedings.mlr.press/v48/arjovsky16.html

[15] Mhammedi, Z., Hellicar, A., Rahman, A., Bailey, J.: Efficient orthogonal parametrisation of recurrent neural networks using householder reflections. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 2401–2409. PMLR, Sydney, Australia (2017). https://proceedings.mlr.press/v70/mhammedi17a.html

[16] Béthune, L., Massena, T., Boissin, T., Bellet, A., Mamalet, F., Prudent, Y., Friedrich, C., Serrurier, M., Vigouroux, D.: DP-SGD without clipping: The lipschitz neural network way. In: The Twelfth International Conference on Learning Representations (2024). https://openreview.net/forum?id=BEyEziZ4R6

[17] Hirche, C., Rouzé, C., França, D.S.: Quantum differential privacy: An information theory perspective. IEEE Transactions on Information Theory **69**(9), 5771–5787 (2023) https://doi.org/10.1109/TIT.2023.3272904

[18] Angrisani, A., Doosti, M., Kashefi, E.: A unifying framework for differentially private quantum algorithms (2023). https://arxiv.org/abs/2307.04733

[19] Du, Y., Hsieh, M.-H., Liu, T., Tao, D., Liu, N.: Quantum noise protects quantum classifiers against adversaries. Phys. Rev. Res. **3**, 023153 (2021) https://doi.org/10.1103/PhysRevResearch.3.023153

[20] Zhao, Y., Zhong, H., Zhang, X., Li, Y., Zhang, C., Pan, M.: Bridging quantum computing and differential privacy: Insights into quantum computing privacy. In: 2024 IEEE International Conference on Quantum Computing and Engineering (QCE), vol. 01, pp. 13–24 (2024). https://doi.org/10.1109/QCE60285.2024.00012

[21] Heredge, J., Kumar, N., Herman, D., Chakrabarti, S., Yalovetzky, R., Sureshbabu,

S.H., Li, C., Pistoia, M.: Characterizing privacy in quantum machine learning. npj Quantum Information **11** (2025) https://doi.org/10.1038/s41534-025-01022-z

[22] Su, J., He, R., Li, G., Qin, S., He, Z., Situ, H., Gao, F.: From membership-privacy leakage to quantum machine unlearning (2025) arXiv:2509.06086

[23] Rath, M., Date, H.: Quantum data encoding: a comparative analysis of classical-to-quantum mapping techniques and their impact on machine learning accuracy. EPJ Quantum Technology **11**(1), 72 (2024) https://doi.org/10.1140/epjqt/s40507-024-00285-3

[24] Pérez-Salinas, A., Cervera-Lierta, A., Gil-Fuster, E., Latorre, J.I.: Data re-uploading for a universal quantum classifier. Quantum **4**, 226 (2020) https://doi.org/10.22331/q-2020-02-06-226

[25] Larocca, M., Thanasilp, S., Wang, S., Sharma, K., Biamonte, J., Coles, P.J., Cincio, L., McClean, J.R., Holmes, Z., Cerezo, M.: Barren plateaus in variational quantum computing. Nature Reviews Physics **7**(4), 174–189 (2025) https://doi.org/10.1038/s42254-025-00813-9

[26] Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership inference attacks against machine learning models. In: IEEE Symposium on Security and Privacy, pp. 3–18 (2017). https://doi.org/10.1109/SP.2017.41

[27] Yeom, S., Giacomelli, I., Fredrikson, M., Jha, S.: Privacy risk in machine learning: Analyzing the connection to overfitting. 2018 IEEE Computer Security Foundations Symposium (CSF), 268–282 (2018) https://doi.org/10.1109/CSF.2018.00027 arXiv:1709.01604

[28] Carlini, N., Chien, S., Nasr, M., Song, S., Terzis, A., Tramèr, F.: Membership inference attacks from first principles. In: 2022 IEEE Symposium on Security and Privacy (SP), pp. 1897–1914 (2022). https://doi.org/10.1109/SP46214.2022.9833649

[29] Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R.G.L., Eichner, H., El Rouayheb, S., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P.B., Gruteser, M., Harchaoui, Z., He, C., He, L., Huo, Z., Hutchinson, B., Hsu, J., Jaggi, M., Javidi, T., Joshi, G., Khodak, M., Konečný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Qi, H., Ramage, D., Raskar, R., Song, D., Song, W., Stich, S.U., Sun, Z., Suresh, A.T., Tramèr, F., Vepakomma, P., Wang, J., Xiong, L., Xu, Z., Yang, Q., Yu, F.X., Yu, H., Zhao, S.: Advances and open problems in federated learning. Foundations and Trends in Machine Learning **14**(1–2), 1–210 (2021) https://doi.org/10.1561/2200000083

[30] Mahawaga Arachchige, P.C., Bertok, P., Khalil, I., Liu, D., Camtepe, S., Atiquzzaman, M.: Local differential privacy for deep learning. IEEE Internet of Things Journal **7**(7), 5827–5842 (2020) https://doi.org/10.1109/JIOT.2019.2952146

[31] Lecuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., Jana, S.: Certified robustness to adversarial examples with differential privacy. In: 2019 IEEE Symposium on Security and Privacy (SP), pp. 656–672 (2019). https://doi.org/10.1109/SP.2019.00044

[32] Wierichs, D., Izaac, J., Wang, C., Lin, C.Y.-Y.: General parameter-shift rules for quantum gradients. Quantum **6**, 677 (2022) https://doi.org/10.22331/q-2022-03-30-677

[33] Khanal, B., Rivas, P.: Evaluating the impact of noise on variational quantum circuits in nisq era devices. In: 2023 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE), pp. 1658–1664 (2023). https://doi.org/10.1109/CSCE60160.2023.00272

[34] Mele, A.A., Angrisani, A., Ghosh, S., Khatri, S., Eisert, J., Stilck França, D., Quek, Y.: Noise-induced shallow circuits and the absence of barren plateaus. Nature Physics **22**(5), 751–756 (2026) https://doi.org/10.1038/s41567-026-03245-z

[35] Sedrakyan, T.: private-qml: Code for "Private training in quantum machine learning". GitHub (2026). https://github.com/tigran-sedrakyan/private-qml

[36] Narayanan, A., Dwivedi, I., Dariush, B.: Dynamic Traffic Scene Classification with Space-Time Coherence (2019). https://arxiv.org/abs/1905.12708

[37] Gharibyan, H., Karapetyan, H., Sedrakyan, T., Subasic, P., Su, V.P., Tanin, R.H., Tepanyan, H.: Quantum image loading and classification: experiments on utility-scale quantum computers. Quantum Machine Intelligence **8**(1), 57 (2026) https://doi.org/10.1007/s42484-026-00388-3

A Proofs of Theorems

Theorem 1 (Quantum noise propagation to loss gradients) *Consider a K-class quantum classification model with logits*

$$f_k(x,\theta) = \mathrm{Tr}\big(O_k\,\rho(\theta)\big)$$

for traceless Pauli observables O_k, and cross-entropy loss ℓ. Under global depolarizing noise

$$\mathcal{E}_\lambda(\rho) = (1-\lambda)\rho + \lambda\,\mathbb{1}/d$$

and finite-shot estimation with shot count S, the induced perturbation of the loss gradient can be approximated as additive Gaussian noise:

$$\widehat{\partial_\theta \ell} \approx \partial_\theta \ell + \mathcal{N}(m, v), \tag{18}$$

where the mean m and variance v are computable via 21 from the logits, their gradients, the noise strength λ, and the shot count S.

Proof First, for simplicity, consider a model outputting an expectation value of a single observable. Let $f(\theta) = \mathrm{Tr}\big(O\,\rho(\theta)\big)$ be an ideal expectation-value objective and $\rho(\theta) = U(\theta)\rho(x)U(\theta)^\dagger$, where we omit the dependence of f on x. For a fixed parameter coordinate θ_j, consider the standard two-shift parameter-shift gradient without shot noise:

$$g_j(\theta) = \partial_{\theta_j} f(\theta) = \frac{1}{2}\Big(\mu_j^+(\theta) - \mu_j^-(\theta)\Big),$$

where

$$\mu_j^\pm(\theta) := f\Big(\theta \pm \frac{\pi}{2} e_j\Big)$$

are the shifted expectations and e_j is the j-th standard basis vector. Let the corresponding finite-shot estimators be

$$\widehat{\mu}_j^\pm(\theta) := \widehat{f}\Big(\theta \pm \frac{\pi}{2} e_j\Big), \qquad \widehat{g}_j(\theta) = \frac{1}{2}\Big(\widehat{\mu}_j^+(\theta) - \widehat{\mu}_j^-(\theta)\Big)$$

Since each O_k is a traceless Pauli operator, its single-shot eigenvalue outcomes take values in $\{\pm 1\}$. Let $X_{\pm,1}, \ldots, X_{\pm,S} \in \{\pm 1\}$ denote the single-shot outcomes for S shots. Then

$$\widehat{\mu}_j^\pm(\theta) = \frac{1}{S}\sum_{s=1}^{S} X_{\pm,s}, \qquad \mathbb{E}[X_{\pm,s}] = \mu_j^\pm(\theta),$$

and therefore

$$\mathrm{Var}\big[\widehat{\mu}_j^\pm(\theta)\big] = \frac{1}{S}\,\mathrm{Var}[X_{\pm,s}] = \frac{1}{S}\Big(1 - (\mu_j^\pm(\theta))^2\Big)$$

In particular,

$$\mathrm{Var}[\widehat{f}(\theta)] = \frac{1}{S}\Big(1 - f(\theta)^2\Big)$$

Writing $\widehat{f}(\theta) = f(\theta) + \xi(\theta)$, where $\xi(\theta)$ is the zero-mean shot-noise term, gives

$$\mathrm{Var}[\xi(\theta)] = \frac{1}{S}\Big(1 - f(\theta)^2\Big)$$

Similarly, writing $\widehat{g}_j(\theta) = g_j(\theta) + \zeta_j(\theta)$, and because the two shifted shot sets are independent, we obtain

$$\mathrm{Var}[\zeta_j(\theta)] = \frac{1}{4}\Big(\mathrm{Var}[\widehat{\mu}_j^+(\theta)] + \mathrm{Var}[\widehat{\mu}_j^-(\theta)]\Big) = \frac{1}{4S}\Big(2 - \big((\mu_j^+(\theta))^2 + (\mu_j^-(\theta))^2\big)\Big)$$

To model hardware noise, consider a CPTP channel $\mathcal{E}_\lambda$ applied to the ideal state before measurement, with noise strength λ. The noisy objective is

$$f_\lambda(\theta) = \mathrm{Tr}\big(O\,\mathcal{E}_\lambda(\rho(\theta))\big),$$

and the corresponding infinite-shot gradient coordinate can be written as

$$g_{j,\lambda}(\theta) = \partial_{\theta_j} f_\lambda(\theta) = g_j(\theta) + b_{j,\lambda}(\theta),$$

where $b_{j,\lambda}(\theta)$ is a generally nonzero bias term. For the global depolarizing channel

$$\mathcal{E}_\lambda(\rho) = (1-\lambda)\rho + \lambda\,\mathbb{1}/d,$$

one has

$$f_\lambda(\theta) = \mathrm{Tr}\big(O\,\mathcal{E}_\lambda(\rho(\theta))\big) = (1-\lambda)\,\mathrm{Tr}\big(O\,\rho(\theta)\big) + \lambda\,\mathrm{Tr}\Big(O\,\frac{\mathbb{1}}{d}\Big) = (1-\lambda)\,f(\theta)$$

for traceless Pauli observables, and therefore

$$g_{j,\lambda}(\theta) = (1-\lambda)\,g_j(\theta)$$

So the objective and the gradient coordinate are multiplicatively shrunk by the factor $1-\lambda$.

If shot noise is also present, then the shifted noisy expectations satisfy

$$\mu^{\pm}_{j,\lambda}(\theta) := f_\lambda\Big(\theta \pm \frac{\pi}{2}e_j\Big), \qquad \widehat{\mu}^{\pm}_{j,\lambda}(\theta) := \widehat{f}_\lambda\Big(\theta \pm \frac{\pi}{2}e_j\Big),$$

and the corresponding variance becomes

$$\mathrm{Var}\big[\widehat{\mu}^{\pm}_{j,\lambda}(\theta)\big] = \frac{1}{S}\Big(1 - (\mu^{\pm}_{j,\lambda}(\theta))^2\Big)$$

In particular,

$$\mathrm{Var}[\widehat{f}_\lambda(\theta)] = \frac{1}{S}\Big(1 - f_\lambda(\theta)^2\Big) = \frac{1}{S}\Big(1 - (1-\lambda)^2 f(\theta)^2\Big)$$

Writing

$$\widehat{f}_\lambda(\theta) = f(\theta) + \xi_\lambda(\theta), \qquad \widehat{g}_{j,\lambda}(\theta) = g_j(\theta) + \zeta_{j,\lambda}(\theta),$$

gives

$$\sigma^2_\xi := \mathrm{Var}[\xi_\lambda(\theta)] = \frac{1}{S}\Big(1 - (1-\lambda)^2 f(\theta)^2\Big),$$

$$\sigma^2_{\zeta,j} := \mathrm{Var}[\zeta_{j,\lambda}(\theta)] = \frac{1}{4S}\Big(2 - (1-\lambda)^2\big((\mu^+_j(\theta))^2 + (\mu^-_j(\theta))^2\big)\Big)$$

In contrast to the shot-noise-only case, these perturbations are not centered around zero relative to the ideal model. Indeed,

$$\mu^\xi := \mathbb{E}[\xi_\lambda(\theta)] = -\lambda f(\theta), \qquad \mu^\zeta_j := \mathbb{E}[\zeta_{j,\lambda}(\theta)] = -\lambda g_j(\theta)$$

For large S, the central limit theorem implies that the scalar perturbations are approximately Gaussian. We now apply this componentwise to the K-class logits. Write

$$f := (f_1(x,\theta), \ldots, f_K(x,\theta))^\top, \qquad \widehat{f} := (\widehat{f}_1(x,\theta), \ldots, \widehat{f}_K(x,\theta))^\top, \qquad \xi := \widehat{f} - f,$$

For a fixed parameter index j, let

$$g^{(j)} := (g_{j,1}, \ldots, g_{j,K})^\top, \qquad \widehat{g}^{(j)} := (\widehat{g}_{j,1}, \ldots, \widehat{g}_{j,K})^\top, \qquad \zeta^{(j)} := \widehat{g}^{(j)} - g^{(j)},$$

Let

$$p_y(f) := \mathrm{softmax}_\tau(f) = \mathrm{sm}(f) = \frac{e^{f_y/\tau}}{\sum_k e^{f_k/\tau}}, \qquad \widehat{p}_y(f) := \mathrm{sm}(\widehat{f}) = \mathrm{sm}(f+\xi)$$

Then the noisy estimator of the loss-gradient coordinate is

$$\widehat{\partial_{\theta_j}\ell} = \frac{1}{\tau}\left(\sum_{k=1}^{K} \widehat{p}_k\Big(g_{j,k}(x,\theta) + \zeta_{j,k}(x,\theta)\Big) - \Big(g_{j,y}(x,\theta) + \zeta_{j,y}(x,\theta)\Big)\right)$$

Setting $\xi = 0$ and $\zeta^{(j)} = 0$ recovers the noiseless gradient (17):

$$\partial_{\theta_j} \ell = \frac{1}{\tau} \left(\sum_{k=1}^{K} p_k \, g_{j,k}(x, \theta) - g_{j,y}(x, \theta) \right)$$

Since $\widehat{p} = \mathrm{sm}(f + \xi)$ is nonlinear in ξ, we use a first-order delta-method approximation and drop second-order products of the noise terms. Write

$$\frac{\partial p_k}{\partial f_r} = \frac{1}{\tau} p_k (\delta_{kr} - p_r), \qquad J_p(f) := \partial_f \, \mathrm{sm}(f) = \frac{1}{\tau} (\mathrm{diag}(p) - pp^\top)$$

Hence, linearizing the softmax around f gives

$$\widehat{p} \approx p + J_p(f)\xi, \qquad \delta p := \widehat{p} - p \approx J_p(f)\xi$$

Substituting $\widehat{p} = p + \delta p$ into the partial derivative estimator gives

$$\widehat{\partial_{\theta_j} \ell} = \frac{1}{\tau} \Big(\sum_{k=1}^{K} p_k \big(g_{j,k} + \zeta_{j,k}\big) + \sum_{k=1}^{K} \delta p_k \big(g_{j,k} + \zeta_{j,k}\big) - \big(g_{j,y} + \zeta_{j,y}\big) \Big)$$

Keeping only first-order terms in ξ and $\zeta^{(j)}$ yields

$$\widehat{\partial_{\theta_j} \ell} \approx \frac{1}{\tau} \Big(\sum_{k=1}^{K} p_k g_{j,k} - g_{j,y} \Big) + \frac{1}{\tau} \Big(\sum_{k=1}^{K} p_k \zeta_{j,k} - \zeta_{j,y} \Big) + \frac{1}{\tau} \sum_{k=1}^{K} \delta p_k \, g_{j,k}$$

Replacing $\delta p = J_p(f)\xi$ and defining

$$w^{(j)} := \tau J_p(f) g^{(j)}$$

gives

$$\widehat{\partial_{\theta_j} \ell} \approx \partial_{\theta_j} \ell + \underbrace{\frac{1}{\tau} \Big(p^\top \zeta^{(j)} - \zeta_{j,y} \Big)}_{\eta_j^{(\zeta)}} + \underbrace{\frac{1}{\tau^2} (w^{(j)})^\top \xi}_{\eta_j^{(\xi)}}$$

Writing

$$a := p - e_y,$$

where $e_y \in \mathbb{R}^K$ is the y-th standard basis vector, one obtains

$$\eta_j^{(\zeta)} = \frac{1}{\tau} a^\top \zeta^{(j)}, \qquad \eta_j^{(\xi)} = \frac{1}{\tau^2} (w^{(j)})^\top \xi,$$

and therefore, with

$$\begin{aligned} \bar{\mu}_{\zeta,j} &:= \mathbb{E}[a^\top \zeta^{(j)}], & \bar{\mu}_{\xi,j} &:= \mathbb{E}[(w^{(j)})^\top \xi], \\ \bar{\sigma}^2_{\zeta,j} &:= \mathrm{Var}(a^\top \zeta^{(j)}), & \bar{\sigma}^2_{\xi,j} &:= \mathrm{Var}((w^{(j)})^\top \xi), \end{aligned}$$

assuming that ξ and $\zeta^{(j)}$ are obtained from independent circuit evaluations, the first-order perturbation is approximately Gaussian with

$$\begin{aligned} m &= \frac{1}{\tau} \bar{\mu}_{\zeta,j} + \frac{1}{\tau^2} \bar{\mu}_{\xi,j} \\ v &= \frac{1}{\tau^2} \bar{\sigma}^2_{\zeta,j} + \frac{1}{\tau^4} \bar{\sigma}^2_{\xi,j} \end{aligned} \tag{21}$$

This proves the approximation (18). □

Theorem 2 (Deterministic clipping-bias bound) *Consider the variational classifier of Definition 1 with K class observables $\{O_k\}_{k=1}^K$, parameterised unitary $U(\theta) = \prod_{j=1}^p e^{-i\theta_j H_j}$, and cross-entropy loss at temperature τ. Set*

$$G \;:=\; 2 \max_{1\le k\le K} \|O_k\| \left(\sum_{j=1}^p \|H_j\|^2\right)^{1/2}$$

Then for every data point (x_i, y_i) in the dataset the per-example loss gradient $g_i(\theta) = \partial_\theta \ell(f(x_i,\theta), y_i)$ satisfies

$$\|g_i(\theta)\|_2 \;\le\; \frac{2G}{\tau} \tag{19}$$

In particular, for every clipping threshold $C > 0$, the clipping bias $b_C(\theta)$ defined in (13) satisfies

$$\|b_C(\theta)\|_2 \;\le\; \left(\frac{2G}{\tau} - C\right)_+ \tag{20}$$

Proof Fix a data point (x_i, y_i) and write $\rho_0 := \rho(x_i)$ for the encoded state. We begin with the logit gradient bound. Fix a class k, and consider first the single-parameter case $U(\theta) = e^{-i\theta H}$ for a Hermitian generator H. Differentiating $\rho(\theta) = U\rho_0 U^\dagger$ gives

$$\partial_\theta \rho = (\partial_\theta U)\rho_0 U^\dagger + U\rho_0(\partial_\theta U^\dagger) = -iHU\rho_0 U^\dagger + iU\rho_0 U^\dagger H = -i[H,\rho]$$

Therefore, by cyclicity of the trace,

$$\partial_\theta f_k = \mathrm{Tr}(O_k\, \partial_\theta \rho) = -i\ \mathrm{Tr}(O_k[H,\rho]) = -i\ \mathrm{Tr}([O_k,H]\rho) = i\ \mathrm{Tr}(\rho[H,O_k]),$$

and hence

$$|\partial_\theta f_k| = \big|i\ \mathrm{Tr}(\rho[H,O_k])\big| \;\le\; \|\rho\|_1\, \|[H,O_k]\| \;\le\; 2\,\|H\|\,\|O_k\|,$$

using $\|\rho\|_1 = 1$. For the multi-parameter unitary, write

$$U(\theta) = A_j\, e^{-i\theta_j H_j}\, V_j$$

for each j, where A_j collects the gates to the left of $e^{-i\theta_j H_j}$ and V_j those to the right. Then

$$\partial_{\theta_j} U = A_j(-iH_j e^{-i\theta_j H_j})V_j = -i(A_j H_j A_j^\dagger)\, U,$$

similar to the single-parameter case, except with the evolved generator $H = A_j H_j A_j^\dagger$. Thus

$$\partial_{\theta_j} f_k = i\ \mathrm{Tr}(\rho[H,O_k]) = i\ \mathrm{Tr}\big(\tilde\rho_j[H_j,\, A_j^\dagger O_k A_j]\big),$$

where $\tilde\rho_j = e^{-i\theta_j H_j}\, V_j\, \rho_0\, V_j^\dagger\, e^{i\theta_j H_j}$ by cyclicity. As before,

$$|\partial_{\theta_j} f_k| \;\le\; \|[H,O_k]\| = \|[A_j H_j A_j^\dagger, O_k]\| = \|[H_j,\, A_j^\dagger O_k A_j]\| \;\le\; 2\|H_j\|\,\|O_k\|$$

Summing over coordinates yields

$$\|\partial_\theta f_k(x_i,\theta)\|_2^2 = \sum_{j=1}^p |\partial_{\theta_j} f_k|^2 \;\le\; 4\|O_k\|^2 \sum_{j=1}^p \|H_j\|^2 \;\le\; G^2, \tag{22}$$

For the loss gradient, stack the per-coordinate identities (17) for $j = 1,\dots,p$ into a single vector equation. Viewing each $\partial_\theta f_k(x_i,\theta)$ as a vector in $\mathbb{R}^p$, the per-example loss gradient is

$$g_i(\theta) = \partial_\theta \ell(f(x_i,\theta), y_i) = \frac{1}{\tau}\left(\sum_{k=1}^K p_k(x_i,\theta)\, \partial_\theta f_k(x_i,\theta) - \partial_\theta f_{y_i}(x_i,\theta)\right)$$

Taking the Euclidean norm and applying the triangle inequality,

$$\|g_i(\theta)\|_2 \;\le\; \frac{1}{\tau}\left(\sum_{k=1}^{K} p_k(x_i,\theta)\,\|\partial_\theta f_k(x_i,\theta)\|_2 + \|\partial_\theta f_{y_i}(x_i,\theta)\|_2\right) \;\le\; \frac{2G}{\tau},$$

where the last inequality uses $\sum_k p_k = 1$ together with the logit-gradient bound (22), which holds uniformly over k.

It remains to bound the clipping bias. By definition $\bar{g}_i(\theta) - g_i(\theta) = \Pi_C(g_i(\theta)) - g_i(\theta)$, and for any vector g one has $\|\Pi_C(g) - g\|_2 = (\|g\|_2 - C)_+$. Hence Jensen's inequality together with the uniform bound (19) gives

$$\|b_C(\theta)\|_2 = \big\|\,\mathbb{E}[\bar{g}_i(\theta) - g_i(\theta)]\big\|_2 \;\le\; \mathbb{E}\,\big(\|g_i(\theta)\|_2 - C\big)_+ \;\le\; \left(\frac{2G}{\tau} - C\right)_+,$$

which is (20). □