Introduction Least squares, logistic regression, and the support vector machine (SVM) are widely used techniques in statistical modeling and machine learning to unravel patterns, make predictions, and derive meaningful insights from data. While least squares is used in regression data fitting [1–8], logistic regression is specifically designed for binary classification problems [9–14]. The SVM classifies vectors in a feature space into one of two sets, given training data from the sets [15–19].

Online learning algorithms have gained much attention in recent decades, in both the academic and industrial sectors [20–26]. In this framework, the following sequence of events takes place in every time step $t \in [T]$ for some fixed $T \in \mathbb{Z}_+$: 1) The learner receives an unlabelled example $x^{(t)}$; 2) The learner makes a prediction $\hat{y}^{(t)}$ based on an existing weight vector $w^{(t)}$; 3) The learner receives the true label $y^{(t)}$ and suffers a loss $L(w^{(t)}, x^{(t)}, y^{(t)})$ that is convex in $w^{(t)}$; 4) The learner updates the weight vector according to some update rule. The *regret* of an online algorithm is defined as the difference between the total loss incurred by using a certain sequence of strategies and the total loss incurred by using the best fixed strategy in hindsight. Specifically[1], $Regret := \frac{1}{T}\sum_{t=1}^{T} L\left(w^{(t)}, x^{(t)}, y^{(t)}\right) - \min_{u\in\mathbb{R}^d} \frac{1}{T}\sum_{t=1}^{T} L\left(u, x^{(t)}, y^{(t)}\right)$, where u is some fixed strategy. Seeing the importance of sparse solutions in the era of big data and the strength of online algorithms [27–29], Ref. [30] developed a method to obtain sparsity via truncated gradient descent, showing a near-optimal online regret bound of $O(1/\sqrt{T})$. In their work, the following assumptions are made.

Assumption 1. *For every $t \in [T]$,*

(i) The loss function $L\left(w^{(t)}, x^{(t)}, y^{(t)}\right)$ is convex in $w^{(t)}$ for all $x^{(t)}, y^{(t)}$.

(ii) There exist constants $A, B \in \mathbb{R}_{>0}$ such that $\left\|\nabla_{w^{(t)}} L\left(w^{(t)}, x^{(t)}, y^{(t)}\right)\right\|_2^2 \leqslant A \cdot L\left(w^{(t)}, x^{(t)}, y^{(t)}\right) + B$ for all $x^{(t)}, y^{(t)}$.

(iii) $\sup_{x^{(t)}} \|x^{(t)}\|_2 \leqslant C$, , for some constant $C \in \mathbb{R}_+$.

Ref. [30] pointed out some common loss functions of linear prediction problems with corresponding choices of parameters A and B (which are not necessarily unique), under the assumption that $\sup_{x^{(t)}} \|x^{(t)}\|_2 \leqslant C$. Among them are

- Logistic: $\ln\left(1 + \exp\left(-w^{(t)T} x^{(t)} \cdot y(t)\right)\right)$; $A = 0, B = C^2, y^{(t)} \in \{\pm 1\}$ for all $t \in [T]$.
- SVM (hinge loss): $\max\left\{0, 1 - w^{(t)T} x^{(t)} \cdot y(t)\right\}$; $A = 0, B = C^2, y^{(t)} \in \{\pm 1\}$ for all $t \in [T]$.
- Least squares (square loss): $\left(w^{(t)T} x^{(t)} - y^{(t)}\right)^2$; $A = 4C^2, B = 0,\ y^{(t)} \in \mathbb{R}$ for all $t \in [T]$.

Our contribution In this work, we present the first quantum online algorithm that outputs a sparse solution for least squares, logistic regression and the SVM. Our work is based on Ref. [30], and hence the guarantees of our algorithm hold under the same assumptions as [30]. Our quantum algorithm uses subroutines such as quantum norm and inner product estimation and quantum state preparation, which are based on the recent technique of non-destructive unbiased quantum amplitude amplification [31–33]. We show that our quantum algorithm maintains the $O(1/\sqrt{T})$ regret bound of Ref. [30] while achieving a quadratic speedup in the dimension d of the problem. This speedup is noticeable when $T \geqslant \frac{d}{\log^2 1/\delta}$. Our algorithms also take advantage of unitaries that perform efficient arithmetic computation to compute every entry of the weight vector at any time step. This allows us to save on the space/memory of the algorithm for storing the weight vectors, which is $O(d)$ in Ref. [30]. We summarize our results in the Table 1.

[1] Strictly speaking, this is the per-step regret as we normalize by T. While the conventional regret is the unnormalized version, we nevertheless call this the regret in this paper.

Table 1: Summary of results. In this work, d is the dimension of the weight vectors, $C, D, g_{\max}$ are constants, δ is the failure probability, u^* is the best fixed wieght vector in hindsight.

Problem	Runtime		Regret	
	Ref. [30]	Our work	Ref. [30]	Our work
Least squares	$O(Td)$	$O\left(T^{3/2}\sqrt{d}\log\frac{1}{\delta}\right)$	$\frac{C^2\|u^*\|_2^2}{2\sqrt{T}}$	$\frac{C^2\left(CD+g_{\max}+\|u^*\|_2^2\right)}{\sqrt{T}}$
Logistic regression	$O(Td)$	$O\left(T^{3/2}\sqrt{d}\log\frac{1}{\delta}\right)$	$\frac{C^2\|u^*\|_2^2+1}{2\sqrt{T}}$	$\frac{1+C^2\left(2+g_{\max}+\|u^*\|_2^2\right)}{2\sqrt{T}}$
SVM	$O(Td)$	$O\left(T^{3/2}\sqrt{d}\log\frac{1}{\delta}\right)$	$\frac{C^2\|u^*\|_2^2+1}{2\sqrt{T}}$	$\frac{2+C^2\left(g_{\max}+\|u^*\|_2^2\right)}{2\sqrt{T}}$

Data input and quantum subroutines We assume quantum access to the entries of the unlabelled examples. The online nature of the problem is given by the fact that we obtain these example oracles at different times.

Data Input 1 (Online example oracle). *Let $x^{(1)}, \cdots, x^{(T)} \in \mathbb{R}^d$ be unlabelled examples. Define the unitary $U_{x^{(t)}}$ operating on $O(\log d)$ qubits such that for all $j \in [d]$ and $t \in [T]$, $U_{x^{(t)}}|j\rangle|\bar{0}\rangle = |j\rangle|x_j^{(t)}\rangle$. At time $t \in [T]$, assume access to $U_{x^{(1)}}, \cdots, U_{x^{(t)}}$.*

We define between and comparison oracles which will be used in the truncation unitary.

Definition 1 (Between and comparison oracles). *Let $a, b, x, y \in \mathbb{R}$ and $B \in \{0,1\}$. We say that we have access to a between oracle $\mathcal{O}_{btw,a,b}$ and a comparison oracle $\mathcal{O}_{comp}$ if we have access to unitaries $U_{btw,a,b}$ and U_{comp} that performs*

$$U_{btw,a,b}|x\rangle|0\rangle \to \begin{cases} |x\rangle|1\rangle, & \text{if } a \leqslant x \leqslant b \\ |x\rangle|0\rangle, & \text{otherwise} \end{cases}, \quad U_{comp}|B\rangle|x\rangle|0\rangle = \begin{cases} |B\rangle|x\rangle|\max(x,0)\rangle, & \text{if } B=1 \\ |B\rangle|x\rangle|\min(x,0)\rangle, & \text{if } B=0 \end{cases}.$$

Lemma 1 (Truncation unitary). *Let $\theta, \alpha \in \mathbb{R}_{>0}$. Assuming access to oracles in Def 1, there exists a unitary $U_{\mathcal{T},\alpha,\theta}$ operator that does the following operation up to sufficient accuracy in constant time.*

$$U_{\mathcal{T},\alpha,\theta} : |x\rangle \to \begin{cases} |\max\{x-\alpha, 0\}\rangle, & \text{if } 0 < x \leqslant \theta \\ |\min\{x+\alpha, 0\}\rangle, & \text{if } -\theta \leqslant x < 0 \\ |x\rangle, & \text{otherwise} \end{cases} \tag{1}$$

The lemma below makes use of the truncation unitary to allow for efficient computation of the weight vector at every time step for all three problems of interest.

Lemma 2. *Let $\theta \in \mathbb{R}_{>0}$. For all $t \in [T]$, let there be given the set of unitaries $U_{x^{(t)}}$ as in Data Input 1, vectors $y = (y^{(1)}, \cdots, y^{(t)}), \tilde{y} = (\tilde{y}^{(1)}, \cdots, \tilde{y}^{(t)}) \in \mathbb{R}^t$ and a real number $\eta \in \mathbb{R}_{>0}$. Assuming access to a gravity sequence $(g^{(1)}, \cdots, g^{(T)})$ and a truncation oracle as in Lemma 1, there exists a unitary operators that perform the following operations for different cases:*

(i) Least squares : $|j\rangle|\bar{0}\rangle \to |j\rangle|\mathcal{T}\left(w_j^{(t)} + 2\eta\left(y^{(t)} - \tilde{y}^{(t)}\right)x_j^{(t)}, g^{(t)}\eta, \theta\right)\rangle$;

(ii) Logistic regression: $|j\rangle|\bar{0}\rangle \to |j\rangle|\mathcal{T}\left(w_j^{(t-1)} + 2\eta\frac{x_j^{(t)}y^{(t)}e^{-y^{(t)}\tilde{y}^{(t)}}}{1+e^{-y^{(t)}\tilde{y}^{(t)}}}, g^{(t)}\eta, \theta\right)\rangle$;

(iii) SVM: $|j\rangle|\bar{0}\rangle \to \begin{cases} |j\rangle|\mathcal{T}\left(w_j^{(t)} + \eta y^{(t)}x_j^{(t)}, g^{(t)}\eta, \theta\right)\rangle, & y^{(t)}\tilde{y}^{(t)} < 1 \\ |j\rangle|\mathcal{T}\left(w_j^{(t)}, g^{(t)}\eta, \theta\right)\rangle, & \text{otherwise} \end{cases}$

to sufficient numerical precision. This computation takes one query to the data input and requires $O(T + \log d)$ qubits and quantum gates.

We also use quantum state preparation, norm and inner product estimation subroutines from References [34–38] based on nondestructive unbiased amplitude estimation [31–33].

Main results We prove regret bounds for our quantum algorithm applied to least squares, logistic regression and the SVM.

Theorem 1. *Let $\epsilon, \delta \in (0,1)$ and $C, D, g_{\max} \in \mathbb{R}_+$. Let $u \in \mathbb{R}^d$ be any vector and for $t \in [T]$, let $\tilde{y}^{(t)}$ be an additive estimate of $\hat{y}^{(t)} = w^{(t)T}x^{(t)}$ of error ϵ_{IP} and $q_{w^{(t+1)}}(v) := \left\|v \cdot I\left(\left|w^{(t+1)}\right| \leqslant \theta\right)\right\|_1$ for a fixed vector $w^{(t+1)} \in \mathbb{R}^d$ and some vector $v \in \mathbb{R}^d$, with $\tilde{q}_{w^{(t+1)}}(v)$ being its eadditive stimate. With success probability $1-\delta$ and in time $O\left(T^{3/2}\sqrt{d}\log\frac{1}{\delta}\right)$, under Assumptions 1(iii),there exists a quantum algorithm for sparse online learning that achieves for*

(i) Least squares, a regret of

$$\frac{1}{T}\sum_{t=1}^{T}\left[\left(\tilde{y}^{(t)} - y^{(t)}\right)^2 - \left(u^T x^{(t)} - y^{(t)}\right)^2 + g^{(t)}Q^{(t+1)}\right] \leqslant \frac{C^2\left(CD + g_{\max} + \|u\|_2^2\right)}{\sqrt{T}},$$

where $\left|y^{(t)} - \hat{y}^{(t)}\right| \leqslant D$ for all $t \in [T]$;

(ii) Logistic regression, a regret of

$$\frac{1}{T}\sum_{t=1}^{T}\left[\ln\left(1 + e^{-y^{(t)}\tilde{y}^{(t)}}\right) - \ln\left(1 + e^{-y^{(t)}u^T x^{(t)}}\right) + g^{(t)}Q^{(t+1)}\right] \leqslant \frac{1 + C^2\left(2 + g_{\max} + \|u\|_2^2\right)}{2\sqrt{T}};$$

(iii) the SVM, a regret of

$$\frac{1}{T}\sum_{t=1}^{T}\left[\left(1 - y^{(t)}\tilde{y}^{(t)}\right)^+ - \left(1 - y^{(t)}u^T x^{(t)}\right)^+ + g^{(t)}Q^{(t+1)}\right] \leqslant \frac{2 + C^2\left(g_{\max} + \|u\|_2^2\right)}{2\sqrt{T}}$$

where $Q^{(t+1)} := \left(\tilde{q}^{(w+1)}\left(w^{(t+1)}\right) - q^{w^{((t+1)}}(u)\right)$.

Conclusion We proposed quantum online learning algorithms that output sparse solutions for least squares, logistic regression and the SVM. Our quantum algorithm achieves a quadratic speedup in the dimension of the problem as compared to its classical counterparts.Leveraging on unitaries that perform efficient arithmetic computation, we save on the space/memory of the algorithm.

We point out that our quantum algorithm having a run time that achieves quadratic improvement in the dimension d of the weight vector, its dependence on the number of time steps T increases. One natural question would be to ask if the trade-off between T and d can be avoided. Besides that, it would be interesting to explore how other variants of gradient descent such as mirror descent or stochastic gradient descent, combined with different "feature selection" techniques to obtain sparse solutions can contribute to an improvement in the regret bound. Considering that we have a unitary that computes entries of the weight vector that is updated via truncated gradient descent , one could consider potential applications of this unitary, for example in reinforcement learning [39]. On the other hand, one could explore possible applications of quantum algorithms in obtaining sparse solutions in the online learning setting as there has not been any work done in this regime. Instead of analyzing the (static) regret, one could consider studying the *dynamic* regret of the online algorithm which can be useful in scenarios where the optimal solution keeps changing in evolving environments [40–46].

References

[1] Per Christian Hansen, Victor Pereyra, and Godela Scherer. *Least squares data fitting with applications*. JHU Press, 2013.

[2] David Eberly. Least squares fitting of data. *Chapel Hill, NC: Magic Software*, pages 1–10, 2000.

[3] CA Cantrell. Review of methods for linear least-squares fitting of data and application to atmospheric chemistry problems. *Atmospheric Chemistry and Physics*, 8(17):5477–5487, 2008.

[4] Geoffrey S Watson. Linear least squares regression. *The Annals of Mathematical Statistics*, pages 1679–1699, 1967.

[5] Paul Geladi and Bruce R Kowalski. Partial least-squares regression: a tutorial. *Analytica chimica acta*, 185:1–17, 1986.

[6] Jean-Yves Audibert and Olivier Catoni. Robust linear least squares regression. *Annals of Statistics 2011*, 39(5), 2011.

[7] Odalric Maillard and Rémi Munos. Compressed least-squares regression. *Advances in neural information processing systems*, 22, 2009.

[8] Stephen Boyd and Lieven Vandenberghe. *Introduction to applied linear algebra: vectors, matrices, and least squares*. Cambridge university press, 2018.

[9] Michael P LaValley. Logistic regression. *Circulation*, 117(18):2395–2399, 2008.

[10] Todd G Nick and Kathleen M Campbell. Logistic regression. *Topics in biostatistics*, pages 273–301, 2007.

[11] Scott Menard. *Applied logistic regression analysis*. Number 106 in Quantitative Applications in the Social Sciences. Sage, 2002.

[12] Abhik Das. Logistic regression. In *Encyclopedia of Quality of Life and Well-Being Research*, pages 1–2. Springer, 2021.

[13] Sandro Sperandei. Understanding logistic regression analysis. *Biochemia medica*, 24(1):12–18, 2014.

[14] Jill C Stoltzfus. Logistic regression: a brief primer. *Academic emergency medicine*, 18(10):1099–1104, 2011.

[15] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995.

[16] Ovidiu Ivanciuc et al. Applications of support vector machines in chemistry. *Reviews in computational chemistry*, 23:291, 2007.

[17] Shujun Huang, Nianguang Cai, Pedro Penzuti Pacheco, Shavira Narrandes, Yang Wang, and Wayne Xu. Applications of support vector machine (svm) learning in cancer genomics. *Cancer genomics & proteomics*, 15(1):41–51, 2018.

[18] Zheng Rong Yang. Biological applications of support vector machines. *Briefings in bioinformatics*, 5(4):328–338, 2004.

[19] Francis EH Tay and Lijuan Cao. Application of support vector machines in financial time series forecasting. *omega*, 29(4):309–317, 2001.

[20] Matthew Hoffman, Francis Bach, and David Blei. Online learning for latent dirichlet allocation. *advances in neural information processing systems*, 23, 2010.

[21] L eon Bottou. Online learning and stochastic approximations. *Online learning in neural networks*, 17(9):142, 1998.

[22] Jyrki Kivinen, Alexander J Smola, and Robert C Williamson. Online learning with kernels. *IEEE transactions on signal processing*, 52(8):2165–2176, 2004.

[23] Ofer Dekel, Ran Gilad-Bachrach, Ohad Shamir, and Lin Xiao. Optimal distributed online prediction using mini-batches. *Journal of Machine Learning Research*, 13(1), 2012.

[24] David P Helmbold, Robert E Schapire, Yoram Singer, and Manfred K Warmuth. On-line portfolio selection using multiplicative updates. *Mathematical Finance*, 8(4):325–347, 1998.

[25] Oren Anava, Elad Hazan, Shie Mannor, and Ohad Shamir. Online learning for time series prediction. In *Conference on learning theory*, pages 172–184. PMLR, 2013.

[26] Steven CH Hoi, Doyen Sahoo, Jing Lu, and Peilin Zhao. Online learning: A comprehensive survey. *Neurocomputing*, 459:249–289, 2021.

[27] Jingwei Liang and Clarice Poon. Screening for sparse online learning. *arXiv preprint arXiv:2101.06982*, 2021.

[28] Dayong Wang, Pengcheng Wu, Peilin Zhao, and Steven CH Hoi. A framework of sparse online learning and its applications. *arXiv preprint arXiv:1507.07146*, 2015.

[29] Fan Lin, Xiuze Zhou, and Wenhua Zeng. Sparse online learning for collaborative filtering. *International Journal of Computers Communications & Control*, 11(2):248–258, 2016.

[30] John Langford, Lihong Li, and Tong Zhang. Sparse online learning via truncated gradient. *Journal of Machine Learning Research*, 10(3), 2009.

[31] Aram W Harrow and Annie Y Wei. Adaptive quantum simulated annealing for bayesian inference and estimating partition functions. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 193–212. SIAM, 2020.

[32] Patrick Rall and Bryce Fuller. Amplitude estimation from quantum signal processing. *Quantum*, 7:937, 2023.

[33] Arjan Cornelissen and Yassine Hamoudi. A sublinear-time quantum algorithm for approximating partition functions. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1245–1264. SIAM, 2023.

[34] Joran van Apeldoorn and András Gilyén. Quantum algorithms for zero-sum games. *arXiv preprint arXiv:1904.03180*, 2019.

[35] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 305:53–74, 2002.

[36] Yassine Hamoudi, Patrick Rebentrost, Ansis Rosmanis, and Miklos Santha. Quantum and classical algorithms for approximate submodular function minimization. *arXiv preprint arXiv:1907.05378*, 2019.

[37] Tongyang Li, Shouvanik Chakrabarti, and Xiaodi Wu. Sublinear quantum algorithms for training linear and kernel-based classifiers. In *International Conference on Machine Learning*, pages 3815–3824. PMLR, 2019.

[38] Patrick Rebentrost, Yassine Hamoudi, Maharshi Ray, Xin Wang, Siyi Yang, and Miklos Santha. Quantum algorithms for hedging and the learning of ising models. *Physical Review A*, 103(1):012418, 2021.

[39] Sridhar Mahadevan and Bo Liu. Sparse q-learning with mirror descent. *arXiv preprint arXiv:1210.4893*, 2012.

[40] Omar Besbes, Yonatan Gur, and Assaf Zeevi. Non-stationary stochastic optimization. *Operations research*, 63(5):1227–1244, 2015.

[41] Ali Jadbabaie, Alexander Rakhlin, Shahin Shahrampour, and Karthik Sridharan. Online optimization: Competing with dynamic comparators. In *Artificial Intelligence and Statistics*, pages 398–406. PMLR, 2015.

[42] Aryan Mokhtari, Shahin Shahrampour, Ali Jadbabaie, and Alejandro Ribeiro. Online optimization in dynamic environments: Improved regret rates for strongly convex problems. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 7195–7201. IEEE, 2016.

[43] Tianbao Yang, Lijun Zhang, Rong Jin, and Jinfeng Yi. Tracking slowly moving clairvoyant: Optimal dynamic regret of online learning with true and noisy gradient. In *International Conference on Machine Learning*, pages 449–457. PMLR, 2016.

[44] Lijun Zhang, Tianbao Yang, Jinfeng Yi, Rong Jin, and Zhi-Hua Zhou. Improved dynamic regret for non-degenerate functions. *Advances in Neural Information Processing Systems*, 30, 2017.

[45] Lijun Zhang, Tianbao Yang, Zhi-Hua Zhou, et al. Dynamic regret of strongly adaptive methods. In *International conference on machine learning*, pages 5882–5891. PMLR, 2018.

[46] Peng Zhao, Yu-Jie Zhang, Lijun Zhang, and Zhi-Hua Zhou. Dynamic regret of convex and smooth functions. *Advances in Neural Information Processing Systems*, 33:12510–12520, 2020.