Machine Learning: Science and Technology

PAPER • OPEN ACCESS

Encrypted machine learning of molecular quantum properties

MACHINE
LEARNING
Science and Technology

PAPER

Encrypted machine learning of molecular quantum properties

Jan Weinreich[1,2], Guido Falk von Rudorff[3] and O Anatole von Lilienfeld[4,5,6,*]

[1] University of Vienna, Faculty of Physics,Kolingasse 14-16, AT-1090 Wien, Austria
[2] University of Vienna, Vienna Doctoral School in Physics, Boltzmanngasse 5, 1090 Vienna, Austria
[3] University Kassel, Department of Chemistry, Heinrich-Plett-Str.40, 34132 Kassel, Germany
[4] Vector Institute for Artificial Intelligence, Toronto, ON M5S 1M1, Canada
[5] Departments of Chemistry, Materials Science and Engineering, and Physics, University of Toronto, St. George Campus, Toronto, ON, Canada
[6] Machine Learning Group, Technische Universität Berlin and Institute for the Foundations of Learning and Data, 10587 Berlin, Germany
[*] Author to whom any correspondence should be addressed.

E-mail: anatole.vonlilienfeld@utoronto.ca

Keywords: encrypted machine learning, molecular properties, cooperation in adversarial setting, scarcity of experimental data

Supplementary material for this article is available online

CrossMark

OPEN ACCESS

RECEIVED
22 December 2022

REVISED
8 March 2023

ACCEPTED FOR PUBLICATION
30 March 2023

PUBLISHED
27 April 2023

Abstract

Large machine learning (ML) models with improved predictions have become widely available in the chemical sciences. Unfortunately, these models do not protect the privacy necessary within commercial settings, prohibiting the use of potentially extremely valuable data by others. Encrypting the prediction process can solve this problem by double-blind model evaluation and prohibits the extraction of training or query data. However, contemporary ML models based on fully homomorphic encryption or federated learning are either too expensive for practical use or have to trade higher speed for weaker security. We have implemented secure and computationally feasible encrypted ML models using oblivious transfer enabling and secure predictions of molecular quantum properties across chemical compound space. However, we find that encrypted predictions using kernel ridge regression models are a million times more expensive than without encryption. This demonstrates a dire need for a compact ML model architecture, including molecular representation and kernel matrix size, that minimizes model evaluation costs.

1. Introduction

The global amount of information has grown exponentially over time. Seagate, a large data storage company, projects it to reach 181 zettabytes by 2025 [1]. Countless machine learning (ML) applications are based on this wealth of data, as reflected in a rapidly growing number of ML publications [2]. Still, especially sensitive data is not publicly accessible, preventing innovative ML innovations in these fields. The main issue is that the evaluation of ML models is not double-blind: the user querying the ML model can gather information about the training set and discloses all information about the query. The holder of the database can accumulate huge amounts of querying user data posing a threat if the server is under attack from a third party. This is especially relevant considering the fast-growing use of cloud computing [1] and number of cyber-attacks. End-to-end encryption cannot solve this issue as data is usually processed in unencrypted form.

Decisions based on knowledge derived from protected data without revealing any data would warrant immediate benefits: potentially relevant fields include modeling health data or sharing predictions evaluated on protected databases. Furthermore, double-blind ML evaluation may reduce customers' hesitations to send sensitive medical data to the cloud, allowing, for instance, more personalized health recommendations—without giving away private information. From the viewpoint of chemistry and medical sciences, a potential application is commercial data from pharmaceutical companies, since a considerable amount is invested in various screening approaches (*in vivo* and *in vitro*). While the collected data sets are relevant for developing new pharmaceuticals, they are generally not published. Currently, nondisclosure agreements are the only way for the chemical industry to provide academia with protected data. However,

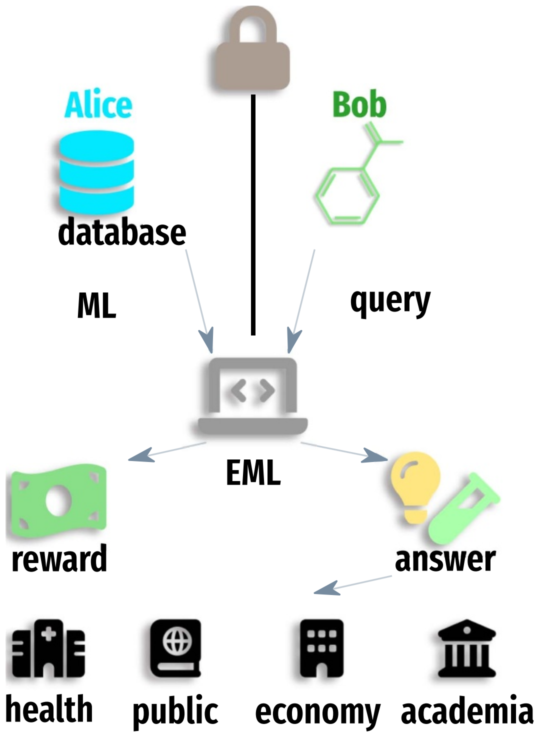

Figure 1. Bob the user **B**, working in health, industry or academic, or public sector, wants to predict properties of query molecules: first Alice **A**, holder of a large database, and **B** agree on a molecular representation and the encrypted machine learning (EML) protocol. Next **A** trains EML on her local node with unencrypted calculations. Subsequently, both parties jointly compute the query prediction by exchanging encrypted information without disclosure. Finally, the prediction is revealed to **B** while **A** will never see the query data. **B** Vice versa cannot extract any information about the hidden reference data owned by **A**. All protocol steps are privacy-preserving and both parties effectively form an encrypted double-blind ML Oracle.

this comes with legal and economic risks as well as bureaucratic barriers. To give an idea of the value of privacy: a ballpark estimate of post-approval R&D costs of a new drug ranges between $1.8 and $2.8 billion where much of these costs are clinical trials [3–5]. In particular, toxicological assays are critical for the development of drug candidates [6, 7], new nanomaterials [8], or pesticides, but can take many years and millions of dollars [9, 10] to complete. These time and cost constraints are a substantial bottleneck for the innovation of new substances.

An additional aspect is that the emergence of ML has made modern research more dependent on access to high-quality datasets. Compiling new impactful datasets is only possible with the required funding to perform lab experiments. Public data often originates from several sources, resulting in inconsistencies [11]. Access to predictions based on secret but single-origin high-quality measurements could help mitigate these problems.

Driven by this vision, a growing number of institutions is considering using ML models that allow hiding training data, as can be seen in recent projects such as the *melloddy* initiative [12]. Multiple computational approaches for privacy-preserving ML have been developed; a popular example is federated learning [13–19] where several data sets from different data holders are used to compute a local gradient for subsequent updates of a global model. In particular, Zhu *et al* [20] have implemented federated learning for molecular properties. Despite many advantages, if not properly addressed, federated learning can show several security risks. This is particularly the case when participants are allowed to deviate from the predefined ML protocol (in a *malicious* adversary setting). When training a federated learning model, each potentially *malicious* participant can send false data on purpose [21] to prevent learning of the global model [22–24]. Furthermore, in an iterative procedure, any participant could compare the last global model with the previous state. This may allow probing where the update of other data holders had the greatest impact to detect points that likely exist in the other data sets. In certain scenarios federated learning models allow unnoticed extraction of training data [25].

In this study we achieve double-blind ML prediction of molecular quantum properties by competitive cooperation, a.k.a. *coopetition*: two competitive parties that do not trust each other cooperate in exchanging encrypted pieces of information to evaluate the ML model as illustrated in figure 1.

The encrypted ML (EML) protocol ensures that the data holder maintains access control to the model at all times. More specifically, we considered a two-party setting with Alice **A** the data holder and Bob **B** the user querying the ML model. We will keep this color highlighting consistently throughout the following. Neither **A** nor **B** reveal private data when querying the oracle.

The encryption algorithm MASCOT [26] used in this work is safe against a dishonest majority of attackers and based on a so-called oblivious transfer protocol. Oblivious transfer achieves privacy by transferring an oblivious amount of information between parties which makes it impossible to recover the original secret information. Specifically, as a solution to the double-blind evaluation we have implemented an oblivious transfer ML oracle based on kernel ridge regression [27, 28] a supervised learning method. Each query consists of a vector with input features and results in a single encrypted prediction of a scalar value.

To summarize the key properties and the threat model of the algorithm: no central server is needed since oblivious transfer removes the need for a central entity as required in federated learning. Only the party that owns the data has access to model weights.

The ML oracle can only be trained and evaluated using a single training set split: before training the model we split the data once into test and training data and train only on this training set resulting in a single ML model and a single encrypted prediction for each query. Using multiple training data splits is not possible since any information on the model variance will also leak information about the proximity of training points. To be more accurate predictions for molecules close to the training points will have a much smaller statistical variance than those far from any training point. As a result, including variance of the predictions would provide a systematic attack strategy. The same argument holds for all models that provide a statistical estimate for the uncertainty [29–31] as entry points for model inversion attacks.

The amount of data that can be recovered from a single prediction depends on what an adversary already knows about the individual whose privacy is at risk.

The use of additional publicly known reference data and repeated queries of the encrypted model could enable the attacker to quantify the bias encoded in the encrypted data (see the example given in section 3.2). Reconstruction attacks on training data have had limited success and researchers have focused mainly on membership inference [32], which can be used as a basis for reconstruction attacks. It is also possible to extract memorized private information from deployed language models [33]. Regarding our approach, we assume that there is a secure communication channel between the two parties. Given the security of the oblivious transfer protocol, the data owner cannot learn anything about the query. While we have not found an example of such an attack in the literature the querying party, might send designed queries in an attempt to reconstruct the decision boundary of the kernel-ridge regression algorithm based only on the predicted values. However, we cannot rule out that it is possible to construct an attack in this manner.

Testing encrypted predictions for molecular properties reveals that the results of unencrypted calculations are exactly reproduced. We find compact ML representations superior in terms of cost per prediction and show higher numerical stability.

2. Methods

2.1. Oblivious transfer versus fully homomorphic encryption

Ideas of encrypted calculations of arbitrary functions were first hypothesized in the late 70s [38]. The archetypal problem solved in this context was Yao's Millionaires' problem: two wealthy individuals with money amount x_1 and x_2 want to know if $x_1 > x_2$ is true or false without revealing exact amounts [34, 35] x_1 and x_2.

The first algorithm for a fully homomorphic encryption [39] scheme was presented in 2009 allowing fully encrypted addition and multiplication and evaluation of any real function f. To explain what is meant by computation on encrypted data, we illustrate the addition of numbers $x_3 = x_1 + x_2$. The addition is performed with encrypted E representations or ciphertexts c_1 and c_2 of numbers with a public key p_k. The first ciphertext is $c_1 = \mathrm{E}(p_k, x_1)$ and the second $c_2 = \mathrm{E}(p_k, x_2)$. The decryption D of the addition using the secret key s_k results in the correct number as follows,

$$x_3 = \mathrm{D}(s_k, c_3) = \mathrm{D}(s_k, c_1 + c_2) \tag{1}$$

$$= \mathrm{D}(s_k, c_1) + \mathrm{D}(s_k, c_2) = x_1 + x_2\ . \tag{2}$$

Such encrypted calculations are called fully homomorphic encryption, *fully* because any function can be evaluated, and *homomorphic* meaning *same shape* because fully homomorphic encryption conserves relations between numbers in the encrypted space. A benefit of fully homomorphic encryption is it does not require communication between parties that own the private data instead encryption computations are

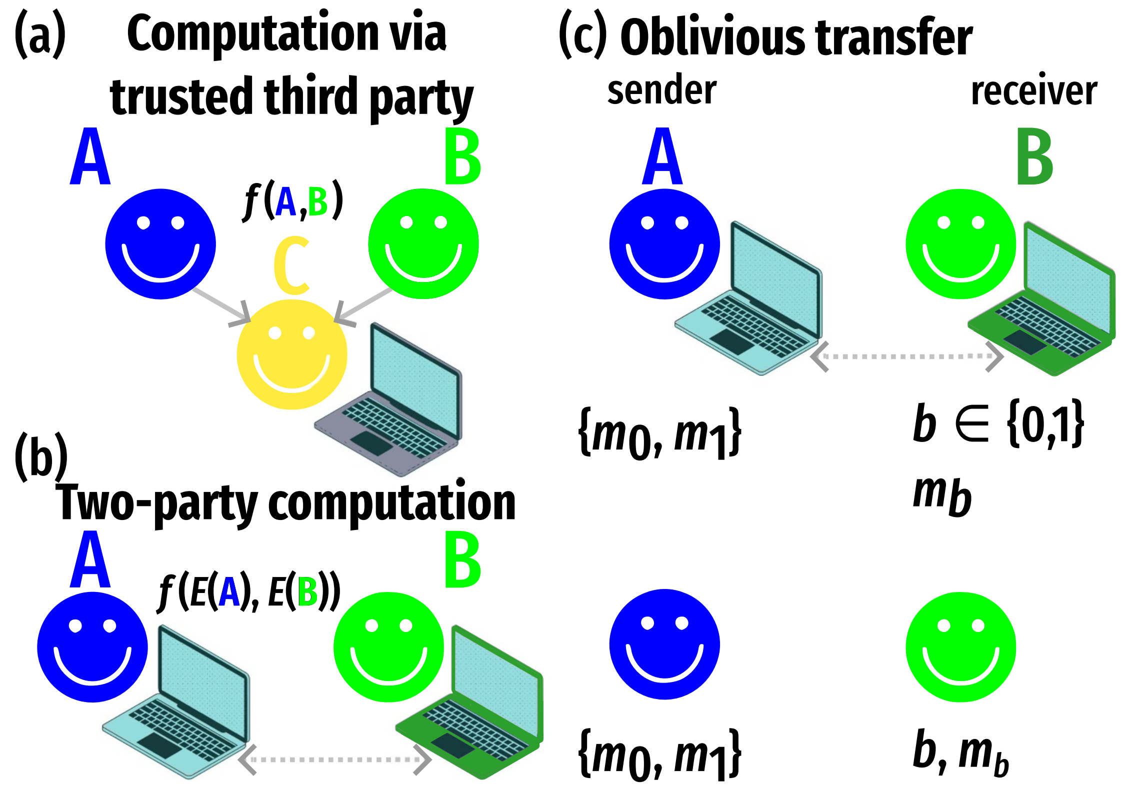

Figure 2. Parties Alice and Bob with private data **A** and **B** respectively compute the value of a publicly known function f that depends on their private inputs. Two scenarios are considered, (a) Alice and Bob submit all their data to a trusted third party Carol **C** who subsequently performs the required calculations. In the second case (b), two-party computation [34, 35] allows removing **C**. Instead, Alice and Bob exchange encrypted E information packages via oblivious transfer [36] to evaluate the function f. (c): Illustration of the originally proposed oblivious transfer protocol [37] for exchanging private information. The oblivious transfer protocol allows transmitting messages m_b to a receiver without asking the sender for the selected message (top). This is equivalent to the receiver obtaining m_b without interacting with the sender (bottom).

performed *offline*. This may also be viewed as a disadvantage since parties cannot query discovery-based requests where ad hoc access to results is necessary. A downside of fully homomorphic encryption is that computations are quite expensive. Furthermore, a central server is needed to perform the calculations only after receiving all encrypted information at once.

An alternative method for privacy-preserving function evaluation is multi-party computation [34, 35, 40] via oblivious transfer [36, 37]. In case of two parties, multi-party computation reduces to two-party computation (s. figure 2(b)) with one sender **A** and a receiver **B**. While the details of the MASCOT protocol are described elsewhere [26] we motivate the simplest possible oblivious transfer operation to build an understanding of more complex protocols for addition, multiplication, and eventually evaluation of real functions f. To perform an encrypted calculation of the public function f parties (B) and (A) exchange chunks of data without disclosing their private inputs. The most elemental oblivious transfer protocol that allows such a process is explained in figure 2(c): the sender **A** with a pair of private message bits m_0 and m_1 sends an *oblivious* amount of information packages $\{m_0, m_1\}$ to the receiver. The receiver **B** has a private input selection bit b of either 0 or 1 determining the message received that is m_b. If the choice of **B** was $b = 0$ then m_0, is received and vice versa. The sender does not know which bit of information was obtained by the receiver. The receiver does not learn anything about message m_{1-b} even if the experiment is repeated multiple times—no additional information is retrieved by performing the protocol. The sender only knows that one of the two transferred oblivious messages was received.

This is for instance achieved by Rivest–Shamir–Adleman (RSA) encryption [41]:

1. In addition to the two messages, **A** generates an RSA key pair consisting of the modulus N and public and private exponents e and d, respectively.
2. Furthermore, **A** generates two random numbers, x_0 and x_1, and sends them along with e and N to **B**.
3. **B** then selects b and a random number x_b.
4. **B** generates another random number k and computes $v = (x_b + k^e)$. This value is sent back to **A**.

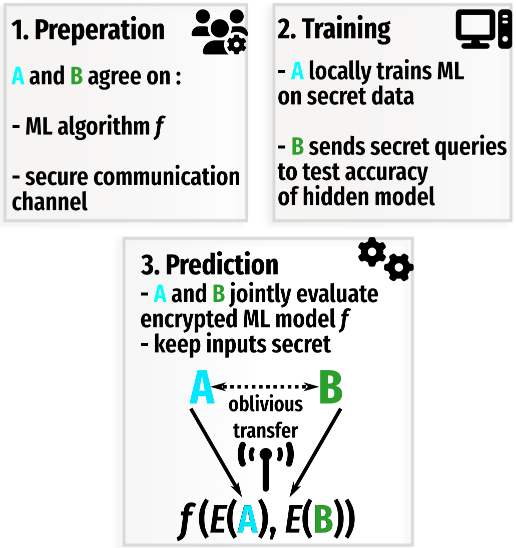
1. Preperation
A and B agree on :
- ML algorithm f
- secure communication channel
2. Training
- A locally trains ML on secret data
- B sends secret queries to test accuracy of hidden model
3. Prediction
- A and B jointly evaluate encrypted ML model f
- keep inputs secret
A
B
oblivious transfer
f(E(A), E(B))

Figure 3. Three-stage process of encrypted machine learning (EML) predictions for two parties, the data holder **A** providing training data and **B** the user submitting the query: preparation (1), training (2) and evaluation (3). The dashed arrow indicates the encrypted exchange via oblivious transfer for evaluation of f on the private inputs.

5. Using v, **A** computes two possible keys as $k_0 = (v - x_0)^d \bmod N$ and $k_1 = (v - x_1)^d \bmod N$. Since **A** did not know which key was the correct one, both secret messages are combined with both keys ($m'_0 = m_0 + k_0$ and $m'_1 = m_1 + k_1$) and sent to **B**.
6. Upon receiving these messages, **B** computes $m_b = m'_b - k$, where b was the bit initially selected. Since **B** did not know the private exponent d he is unable to compute $k_{1-b} = (v - x_{1-b})^d \bmod N$ or to determine m_{1-b}.

Without this protocol, the receiver could have just asked for a specific bit but that would reveal information. During the oblivious transfer evaluation the roles of the receiver and sender are frequently interchanged, but at no point has any party enough information to reconstruct intermediate results. Remarkably, this rather unintuitive way of exchanging information allows privacy-preserving evaluation of any real function [42].

The key advantage of two-party computation via oblivious transfer and in particular of the protocol called malicious arithmetic secure computation with oblivious transfer [26] (MASCOT) is the small computational cost compared to fully homomorphic encryption and other implementations of multi-party computation. MASCOT provides security against a dishonest majority of attackers with *malicious* intent. As for all multi-party computation algorithms, continuous communication between all involved parties is needed, so the transfer of data is the main computing bottleneck. In MASCOT, floating-point numbers are translated into a finite integer representation. To avoid overflow errors the numerical precision $\mathcal{P}$ (s. detailed explanation of $\mathcal{P}$ in SI. section C) can be increased to allow representing larger numbers with better resolution.

2.2. Encrypted ML of molecular properties

2.2.1. Encrypted kernel ridge regression

Alice **A** holds secret training data and collaborates with Bob **B** the user by providing EML predictions for queries. **B** should not be able to learn anything about the training set, **A** should not learn anything about the query of **B**. Only the prediction is sent to **B** while the calculations cannot be inspected or manipulated by either party. We address this problem by encrypting the ML predictions using the MASCOT protocol discussed in the previous section. All following mathematical expressions are colored according to access to the respective data before, during, or after the encrypted prediction.

Setting up the ML oracle can be separated into three steps shown in figure 3: first, both parties agree on the same mathematical form to represent the data. In the case of movie preferences, this could be a vector containing location and age. For cloud-based services, it could be private data such as heart pressure, blood sugar, or pulse. For secret new drug-like molecules, we use molecular representation vectors such as the Coulomb matrix [43] (CM), or the FCHL19 [44, 45] that require three-dimensional nuclear coordinates and charges. Note that FCHL19 is a local representation that allows comparing atomic environments between different molecules with each other. However for demonstration purposes and because it allows direct timing benchmark comparisons we will treat FCHL19 as a flattened global representation vector like the CM.

Secondly, both parties agree on an ML protocol f, here kernel ridge regression [27, 28]. Kernel ridge regression is a supervised learning method in which for each prediction the features of the query instance are compared against all training instances and weighted by regression coefficients. Next, **A** trains a hidden ML model on a local machine. **A** locally computes the input representation vectors $\mathbf{X}_i$ that can correspond to any set of labels that show good correlation with the quantity of interest y. The kernel ridge regression weights $\boldsymbol{\alpha}$ are obtained by solving a system of equations,

$$\boldsymbol{\alpha} = [\mathbf{K} + \lambda \cdot \mathbb{I}]^{-1} \boldsymbol{y}. \tag{3}$$

All quantities in the upper equation are known to **A** notably the values $\boldsymbol{y}$ of hidden data. The elements of the training kernel matrix **K** are computed with Gaussian functions,

$$(\mathbf{K})_{i,j} = \exp\left[-\frac{||\mathbf{X}_i - \mathbf{X}_j||_2^2}{2\sigma^2}\right], \tag{4}$$

where the elements i, j are contained in the hidden training set and $||.||_2$ denotes the Euclidean norm.

The hyperparameter λ is kept private. The protocol could be adapted such that only **A** has access to σ. In this case, σ would be part of the private input of **A** to the MASCOT protocol. Here σ is considered part of the kernel function to be shared before the protocol begins. This does not reveal information about the training points since σ is only an indicator of the chemical diversity of the training set. For instance, for the Gaussian kernel, an empirical estimate for the kernel width is $\sigma = d_{\max}/\sqrt{2\log 2}$ where $d_{\max}$ is the largest distance matrix element of the training data. However, there are numerous ways of selecting training sets with the same optimized σ and distinct training molecules, since many molecules can have the same distance $d_{\max}$ in chemical space.

Note that models with Laplacian kernel functions reach a higher accuracy for atomization energies [46]—at no additional costs. Here, the deliberate choice of the Gaussian kernel function was motivated by the high computational costs of the Laplacian kernel when using the encryption protocol in comparison to Gaussian kernels. Within the specific encryption implementation, Gaussian kernel functions benefit from optimized dot products procedures resulting in seven times acceleration compared to the Laplacian kernel. For completeness, we have added the code for the encrypted Laplacian kernel to the repository (supplementary data).

Next, **B** calculates the representation vector $\mathbf{X}_Q$ of the query entities on another local machine. Afterward, the training weights $\mathrm{E}(\boldsymbol{\alpha})$, input representation vectors $\mathrm{E}(\mathbf{X}_i)$ and the query representations $\mathrm{E}(\mathbf{X}_Q)$ are encrypted (recall that E is encryption and D decryption). This process takes place during a prepossess phase after establishing a secure communication channel between the two parties. In practice, we perform all calculations using a virtual network on a single machine.

Subsequently, the following expression for encrypted kernel ridge regression prediction is evaluated,

$$\begin{aligned} f &= \mathrm{D}(\mathrm{E}(y)[\mathrm{E}(\mathbf{X}_Q)] \\ &= \mathrm{D}\left(\sum_i^N \mathrm{E}(\alpha_i)\ \mathrm{E}(k)[\mathrm{E}(\mathbf{X}_i), \mathrm{E}(\mathbf{X}_Q); \sigma]\right) . \end{aligned} \tag{5}$$

It is essential that the kernel values $\mathrm{E}(k)[\mathrm{E}(\mathbf{X}_i), \mathrm{E}(\mathbf{X}_Q)]$ are not known to either party. Otherwise, participants could probe kernel elements k by repeatedly querying the oracle to obtain the compound space spanned by the training molecules. For the same reason, the distances $d_{iQ} = ||\mathbf{X}_i - \mathbf{X}_Q||_2^2$ between the training set and the query molecule [47] are never disclosed. Next **B** may evaluate a few encrypted samples to validate the consistency of the hidden predictions. If found to be necessary, **A** can increase the training set size or data diversity in hope of improving the accuracy of the model. In the prediction phase, f is evaluated via oblivious transfer without disclosing $\mathrm{E}(\alpha_i), \mathrm{E}(\mathbf{X}_i), \mathrm{E}(\mathbf{X}_Q)$. Finally, the decrypted plaintext predictions are send to **B** while **A** could obtain a reward in form of a payment for the prediction provided. Effectively, both parties form an ML oracle with a true black-box character.

We use learning curves to quantify the error of the predictions w.r.t. the reference values measured as the mean absolute error (MAE) as a function of the size of the training set N. Learning curves are helpful to understand the efficiency of ML models and are generally found [27] to be linear on a log–log scale,

$$\log\left(\frac{\mathrm{MAE}}{\mathrm{unit}}\right) \approx I - S \cdot \log(N) , \tag{6}$$

where I is the initial error and S is the slope indicating the improvement of the model given more training data.

3. Results and discussion

3.1. Encrypted Kernel predictions: *malicious* security for computational chemistry

Now, we demonstrate encrypted ML predictions for fictitiously confidential chemical data. Predicting the stability of molecules is a key problem in computational chemistry and is well described by solving the Schrödinger equation and atomization energies, the energy contained in all bonds of a molecule. However, solving the Schrödinger equation comes at high computational costs: for instance, costs for solutions of a density functional theory calculation scale to the cubed power with the number of atoms. To give a very rough estimate, computing a molecular dataset with $\sim 20\,000$ molecules of the size of aspirin with coupled cluster singles and doubles [48] scaling with the seventh power of system size would consume 20 000 CPU hours [49]—even for a relatively small basis set such as def2-SVP [50].

Such high computational costs underline the value of high-level computational data. As a potential scenario, we consider company **A** providing encrypted ML predictions and an industrial customer **B** with interest in 20 secret molecules but without time, experience or access to software to perform calculations. We have computed learning curves of atomization energies using encrypted predictions with the QM9 database [51] of organic molecules with up to $N = 8192$ compounds. The resulting learning curves using the CM [43] and FCHL19 [44, 45] representations are shown in figure 4. The deviation from the unencrypted case amounts to numerical noise and cannot be identified visually in the learning curves. Hence, we find that EML accurately reproduces unencrypted predictions.

As expected, time t, as well as data traffic D per prediction, increases linearly with the number of training points N (s. figure 5). Furthermore, there is a striking difference between the FCHL19 [44, 45] representation with $L = 18720$ entries that takes more than twice as long (1 hour at $N = 128$) for a single prediction than the CM [43] ($L = 351$).

Coincidentally, we find that the costs for the CM for $\mathcal{P} = 42$ are almost identical to the costs of the same prediction when using the FCHL19 representation at $\mathcal{P} = 15$ which highlights the importance of selecting a compact representation.

Contrary to FCHL19 the CM representation contains no information about angles or local environments resulting in a larger MAE. Overall, data transfer D between parties is the main computational bottleneck for prediction [26] explaining the near-perfect correlation between D and t (s. figure 5).

Consequently, compact representations such as the CM reduce the prediction time by reducing D. The role of compact representations becomes more evident when studying QM9 learning curves (s. figure 4) for lower numerical precision settings corresponding to faster predictions. For high numerical precision ($\mathcal{P} = 42$) there is hardly any visible difference between the EML and kernel ridge regression learning curves (as in figure 4). At $\mathcal{P} = 15$, we find that the FCHL19 EML learning curve shows a dramatic deterioration for $N \geqslant 128$ while the CM learning curve only begins to deviate at $N > 2000$. Although compact representations include less chemical information, they allow for larger training set sizes, given the same target accuracy as well as high numerical stability. If using representation vectors such as FCHL19 cannot be avoided because predictions with high accuracy w.r.t. the test set is needed the numerical precision $\mathcal{P}$ can be increased to avoid numerical instabilities.

Fortunately, there exists an optimal $\mathcal{P}$ with minimal computational cost and sufficient numerical precision, since t increases only linearly with $\mathcal{P}$, while the numerical deviation decays exponentially. We show the scaling of the prediction time t with the numerical precision $\mathcal{P}$ in figure 6. All other parameters are kept constant. It is encouraging to observe the exponential decay of average numerical noise Δ with increasing fixed number representation precision $\mathcal{P}$ in figure 7. Increasing the precision will on the other hand increase computational costs nearly linearly, cf figure 6. However, we observe a sudden increase in the average prediction time for $\mathcal{P}$ values greater than 20 almost doubling the computational costs. This is due to specific implementation details of the MASCOT protocol.

3.2. Limitations and attack scenarios

The user **B** could query the oracle with points for which reference values are known. A small error for the predicted values would suggest that similar points exist in the hidden training set. This attack will probably not be a threat, as it may require more points as contained in the hidden training set. On the other hand, this procedure can reassure **B** that the hidden model provides reasonable predictions and that **A** has not deliberately added incorrect training points.

In a hypothetical scenario where **B** suspects that **A** stole some reference data **B** could proceed with a similar attack to confirm the suspicion: **B** can query the EML oracle with the data points in question. If the

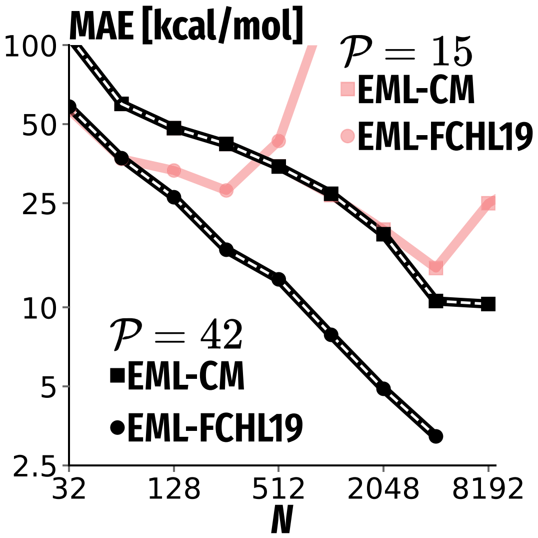

Figure 4. Prediction error for a random subset of QM9 [51] atomization energies as a function of training set size N: mean absolute error (MAE) from encrypted machine learning (EML). Results are shown for two different molecular representations, the Coulomb matrix [43] (CM) and FCHL19 [44, 45]. The numerical precision of fixed-point number representation of floating point numbers is set to either $\mathcal{P} = 15$ or 42. The dashed white lines superimposed with the $\mathcal{P} = 42$ EML results, show the learning curves computed with unencrypted python predictions for comparison.

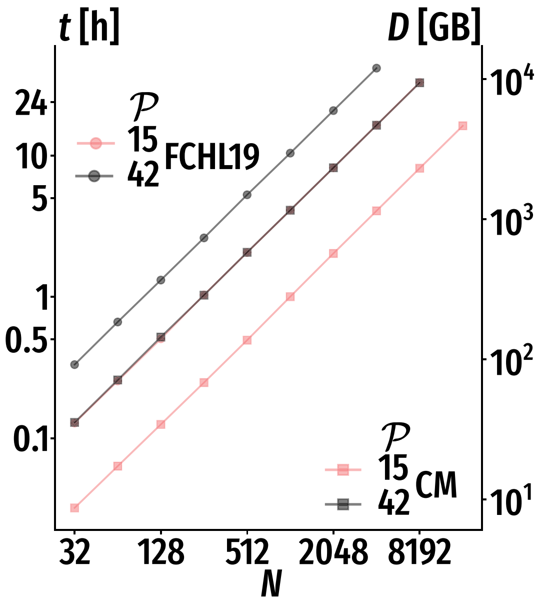

Figure 5. Encrypted machine learning (EML) prediction times averaged per molecule (left) and transferred data D (right) for Coulomb matrix [43] (CM) and FCHL19 [44, 45] representations. Results for two different numerical accuracy settings $\mathcal{P} = 15, 42$ are shown. The costs for CM and FCHL19 nearly perfectly overlap for $\mathcal{P} = 42$ and $\mathcal{P} = 15$ respectively.

predictions consistently agree with the original data this will support the suspicion that **A** included data from **B**. However, it is not conclusive evidence as **A** might add random noise to the training data to decrease the original bias of the reference values.

If **B** knew the scaling rule of the kernel ridge regression ML oracle and the time needed per prediction **B** might be able to guess the number of hidden training molecules. To address this issue **A** could artificially increase the training set by adding a random number of duplicate training points.

An inherent problem of neural networks trained with hidden data is that the loss function gradient vanishes for training set points. In addition, generative adversarial networks are used to reverse engineer points in the training set [25, 52, 53]. Although our approach guarantees safety, this comes with increased computational costs compared to unencrypted calculations. In turn, we find that *honest-but-curios* neural network predictions are orders of magnitude faster since the prediction speed does not depend on the number of training points (s. supplementary data section IV). However, the neural network protocol we have

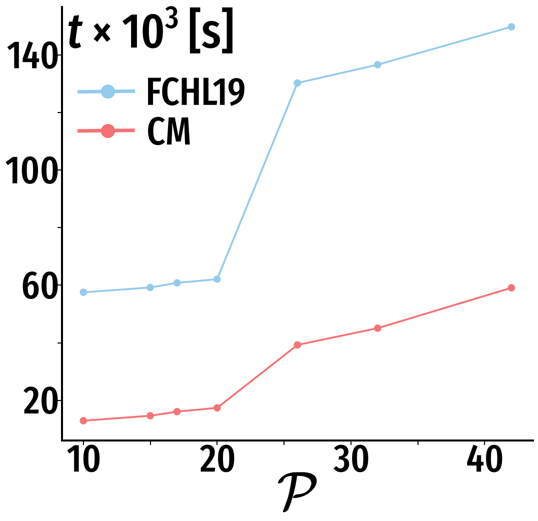

Figure 6. Per molecule prediction time t for various numerical precision settings $\mathcal{P}$ using the Coulomb matrix [43] (CM) and the FCHL19 [44, 45] representation at a training set size of $N = 4097$.

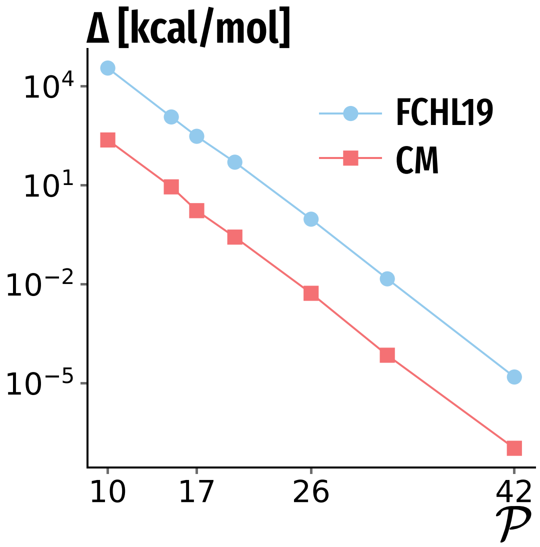

Figure 7. Average numerical error Δ between encrypted machine learning (EML) and plaintext python predictions for various accuracy settings $\mathcal{P}$ at training set size of $N = 4097$. Results for two different molecular representations, the Coulomb Matrix [43] (CM) and FCHL19 [44, 45] are shown.

considered in the supplementary data may not be safe against malicious attacks. Our MASCOT implementation of kernel ridge regression was the exact opposite in these two regards: Evaluation is relatively slow but secure. Overall, we find that encrypting ML predictions is a trade-off between security and computational efficiency.

4. Conclusion

The main advantage of the protocol is its safety against attackers with *malicious* intent, as it is impossible to extract molecular information, either from training or query instances, solely by evaluating encrypted predictions.

The protocol eliminates the need for a trusted third party or central server, as required by fully homomorphic encryption. Instead, it requires only a secure communication channel between the two parties. Since the protocol is online no transfer of all the encrypted data to a single server is needed, contrary to fully homomorphic encryption. This also allows live predictions for new query molecules. The latter aspect is important for Bayesian exploration of chemical space, e.g. in the context of self-driving laboratories

[54] that would require ad hoc predictions. We demonstrated that encrypted predictions of molecular properties based on EML are possible cf figure 4. EML can be adapted to various properties and chemistries with negligible adaptation of the encrypted kernel ridge regression protocol. Since EML does not require molecular representations as input, it may also be applied to pharmaceutical and private data from healthcare or finance.

Our implementation was only possible thanks to recent developments in multi-party computation protocols [26]. We note that added security comes at substantial additional computational costs with data transfer being the main computational bottleneck. Consequently, the compactness of the ML model, in the case of EML the kernel and the molecular representation play a crucial role. More specifically, we have demonstrated that long molecular representation vectors such as FCHL19 [44, 45] allow for more accurate predictions than the more compact CM [43].

As a result, users have to trade off the cost, accuracy, and security of the protocol. Our ball-park estimates indicate that a single molecular EML prediction is a million times more expensive than kernel ridge regression implemented in python code (s. figure 5). For instance, approximately 250 GB of network traffic is needed for a single prediction at a modest training set size of $N = 512$ using the extended-connectivity fingerprint [55] often used in cheminformatics. We believe that one of the first use cases of the encryption protocol could be very costly (in acquisition) and very valuable (highly confidential) data in the industry. For example, data obtained from multi-year toxicology studies on humans with large control groups. Before implementing the presented solution in day-to-day applications for cheaper data, however, the high computational costs of the encryption have to be addressed. Since there is a growing interest in maintaining privacy in ML we believe that the development of more compact ML models or the use of graphics processing units might be beneficial.

A goal of encrypted predictions is to enable decisions based on hidden data as if the knowledge leading to their actions was obtained by inspecting the secret data. However, since the prediction is encrypted, it is impossible to explain the actions that are solely based on the predictions. This lack of transparency may be problematic, as the model could have biases that cannot be explained by users unable to inspect the training data. It is an open question how encrypted predictions can be rationalized without inspecting the training set. One possible approach might be to understand the general behavior of the encrypted model without access to the underlying data, providing insight into the factors influencing the system's predictions.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: http://doi.org/10.5281/zenodo.7863192.

Acknowledgments

O A v L has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Grant Agreement No. 772834). J W acknowledges support from the Faculty of Physics and supervision by C Dellago at the University of Vienna. J W also acknowledges support from J G Brandenburg, in particular for proofreading the manuscript. O A v L has received support as the Ed Clark Chair of Advanced Materials and as a CIFAR AI chair.

Dataset

Initially, we pick a random set of 30 000 QM9 [51] molecules. Of those, we pick twenty test molecules and assign the rest to a training set. Hyperparameters were optimized with five-fold cross-validation for the largest training set size ($N = 8192$) shown in the learning curves using unencrypted calculations and the quantum machine learning code [56]. The small test set size is due to the high computational cost of the encrypted predictions.

We predict the atomization energies, which measure the total energy necessary to dissociate a molecular compound into individual atoms.

The encrypted oblivious transfer calculations were performed in a local network with Intel(R) Xeon(R) E5-2650 v4 @ 2.20GHz CPUs. The number values for the reported timings may differ depending on the hardware.

Third party material

In figures 1–3 we included icons and modified them with permission under the license https://fontawesome.com/license.

Conflict interest

We have no conflicting interests to declare.

ORCID iDs

Jan Weinreich https://orcid.org/0000-0002-9332-4543
O Anatole von Lilienfeld https://orcid.org/0000-0001-7419-0466

References

[1] Source: statista (available at: www.statista.com/topics/1464/big-data/#topicHeader__wrapper and www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf)
[2] Walsh I *et al* 2021 *Nat. Methods* **18** 1
[3] DiMasi J A, Grabowski H G and Hansen R W 2016 *J. Health Econ.* **47** 20
[4] Paul S M, Mytelka D S, Dunwiddie C T, Persinger C C, Munos B H, Lindborg S R and Schacht A L 2010 *Nat. Rev. Drug Discovery* **9** 203
[5] Avorn J 2015 *New Engl. J. Med.* **372** 1877
[6] Hartung T 2009 *Nature* **460** 208
[7] Morger A, Mathea M, Achenbach J, Wolf A, Buesen R, Schleifer K-J, Landsiedel R and Volkamer A 2020 *J. Cheminf.* **12** 24
[8] Choi J-Y, Ramachandran G and Kandlikar M 2009 *Environ. Sci. Technol.* **43** 3030
[9] Price P S, Hubbell B J, Hagiwara S, Paoli G M, Krewski D, Guiseppi-Elie A, Gwinn M R, Adkins N L and Thomas R S 2022 *Risk Anal.* **42** 707–29
[10] Krewski D *et al* 2020 *Arch. Toxicol.* **94** 1–58
[11] Tetko I V, Sushko Y, Novotarskyi S, Patiny L, Kondratov I, Petrenko A E, Charochkina L and Asiri A M 2014 *J. Chem. Inf. Modeling* **54** 3320
[12] MELLODY 2022 (available at: www.melloddy.eu/y2announcement)
[13] Adnan M, Kalra S, Cresswell J C, Taylor G W and Tizhoosh H R 2022 *Sci. Rep.* **12** 1
[14] McMahan B, Moore E, Ramage D, Hampson S and Arcas B A y 2017 *Proc. 20th Int. Conf. on Artificial Intelligence and Statistics* (Proc. of Machine Learning Research) vol 54, ed A Singh and J Zhu (PMLR) (available at: https://proceedings.mlr.press/v54/mcmahan17a.html) pp 1273–82
[15] Ro J H, Suresh A T and Wu K 2020 FedJAX: federated learning simulation with JAX (available at: http://github.com/google/fedjax)
[16] Hard A, Rao K, Mathews R, Ramaswamy S, Beaufays F, Augenstein S, Eichner H, Kiddon C and Ramage D 2019 Federated learning for mobile keyboard prediction (arXiv:1811.03604 [cs.CL])
[17] Choquette-Choo C A, Dullerud N, Dziedzic A, Zhang Y, Jha S, Papernot N and Wang X 2021 Capc learning: confidential and private collaborative learning (arXiv:2102.05188 [cs.LG])
[18] Sav S, Bossuat J-P, Troncoso-Pastoriza J R, Claassen M and Hubaux J-P 2022 *Patterns* **3** 100487
[19] Aggarwal D, Zhou J and Jain A K 2021 *2021 IEEE Int. Joint Conf. on Biometrics (IJCB)* vol 1
[20] Zhu W, Luo J and White A D 2022 *Patterns* **3** 100521
[21] Shumailov I, Shumaylov Z, Kazhdan D, Zhao Y, Papernot N, Erdogdu M A and Anderson R 2021 Manipulating sgd with data ordering attacks (arXiv:2104.09667 [cs.LG])
[22] Fowl L, Geiping J, Reich S, Wen Y, Czaja W, Goldblum M and Goldstein T 2022 Decepticons: corrupted transformers breach privacy in federated learning for language models (arXiv:2201.12675)
[23] Wen Y, Geiping J, Fowl L, Goldblum M and Goldstein T 2022 Fishing for user data in large-batch federated learning via gradient magnification (arXiv:2202.00580)
[24] Possible risks of the mellody platform (available at: www.melloddy.eu/blog/it-security-of-the-melloddy-platform)
[25] Fowl L H, Geiping J, Czaja W, Goldblum M and Goldstein T 2022 *Int. Conf. on Learning Representations* (available at: https://openreview.net/forum?id=fwzUgo0FM9v)
[26] Keller M, Orsini E and Scholl P 2016 *Proc. 2016 ACM SIGSAC Conf. on Computer and Communications Security (CCS '16 Association for Computing Machinery (New York))* pp 830–42
[27] Vapnik V N 1998 *Statistical Learning Theory* (New York: Wiley)
[28] Schütt K T, Chmiela S, von Lilienfeld O A, Tkatchenko A, Tsuda K and Müller K-R 2020 *Machine Learning Meets Quantum Physics* (Berlin: Springer)
[29] Wang Q and Kurz D 2022 *Proc. IEEE/CVF Winter Conf. on Applications of Computer Vision (WACV)* pp 2909–17
[30] Fredrikson M, Jha S and Ristenpart T 2015 *Proc. 22nd ACM SIGSAC Conf. on Computer and Communications Security (CCS '15 Association for Computing Machinery (New York))* pp 1322–33
[31] Wang Q and Kurz D 2022 *2022 IEEE/CVF Winter Conf. on Applications of Computer Vision (WACV)* pp 3870–8
[32] Shokri R, Stronati M, Song C, and Shmatikov V 2016 Membership inference attacks against machine learning models (arXiv:1610.05820)
[33] Carlini N *et al* 2020 Extracting training data from large language models (arXiv:2012.07805)
[34] Yao A C-C 1982 *23rd Annual Symp. on Foundations of Computer Science (Sfcs 1982)* p 160
[35] Yao A C-C 1986 *27th Annual Symp. on Foundations of Computer Science (Sfcs 1986)* p 162
[36] Kilian J 1988 *Proc. 12th Annual ACM Symp. on Theory of Computing (STOC '88 Association for Computing Machinery (New York)* pp 20–31
[37] Rabin M O 2005 How To Exchange Secrets with Oblivious Transfer *IACR Cryptology ePrint Archive* vol 2005 p 187

[38] Rivest R L, Adleman L and Dertouzos M L 1978 *Foundations of Secure Computation* (New York: Academic Press) p 169

[39] Gentry C 2009 *Proc. 41st Annual ACM Symp. on Theory of Computing (STOC '09 Association for Computing Machinery (New York)* pp 169–78

[40] Keller M 2020 Mp-spdz: a versatile framework for multi-party computation *Cryptology ePrint Archive* Report 2020/521 (available at: https://ia.cr/2020/521)

[41] Rivest R L, Shamir A and Adleman L M 1978 *Commun. ACM* **21** 120

[42] Schoenmakers B 2011 Oblivious transfer *Encyclopedia of Cryptography and Security*, ed H C A van Tilborg and S Jajodia (Boston, MA: Springer) pp 884–5

[43] Rupp M, Tkatchenko A, Müller K-R and von Lilienfeld O A 2012 *Phys. Rev. Lett.* **108** 058301

[44] Faber F A, Christensen A S, Huang B and von Lilienfeld O A 2018 *J. Chem. Phys.* **148** 241717

[45] Christensen A S, Bratholm L A, Faber F A and von Lilienfeld O A 2020 *J. Chem. Phys.* **152** 044107

[46] Hansen K, Montavon G, Biegler F, Fazli S, Rupp M, Scheffler M, von Lilienfeld O A, Tkatchenko A and Müller K-R 2013 *J. Chem. Theory Comp.* **9** 3404

[47] Fenner P and Pyzer-Knapp E O 2020 Privacy-preserving Gaussian process regression—a modular approach to the application of homomorphic encryption (arXiv:2001.10893)

[48] David Sherrill C and Schaefer H F 1999 *Advances in Quantum Chemistry* (New York: Academic) pp 143–269

[49] Heinen S, Schwilk M, von Rudorff G F and von Lilienfeld O A 2020 *Mach. Learn.: Sci. Technol.* **1** 025002

[50] Schäfer A, Huber C and Ahlrichs R 1994 *J. Chem. Phys.* **100** 5829

[51] Ramakrishnan R, Dral P O, Rupp M and von Lilienfeld O A 2014 *Sci. Data* **1** 140022

[52] Haim N, Vardi G, Yehudai G, Shamir O and Irani M 2022 arXiv:2206.07758

[53] Zhang Y, Jia R, Pei H, Wang W, Li B and Song D 2020 *2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) (2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR))* (IEEE) pp 250–8

[54] Seifrid M, Pollice R, Aguilar-Granda A, Morgan Chan Z, Hotta K, Ser C T, Vestfrid J, Wu T C and Aspuru-Guzik A 2022 *Acc. Chem. Res.* **55** 2454

[55] Rogers D and Hahn M 2010 *J. Chem. Inf. Model.* **50** 742

[56] Christensen A, Faber F, Huang B, Bratholm L, Tkatchenko A, Müller K, and Lilienfeld O 2017 submitted (available at: https://github.com/qmlcode/qml)