Tensorization of neural networks for improved privacy and interpretability

José Ramón Pareja Monturiol[1,2⋆], **Alejandro Pozas-Kerstjens**[3], **David Pérez-García**[1,2]

[1]Departamento de Análisis Matemático, Universidad Complutense de Madrid, 28040 Madrid, Spain
[2]Instituto de Ciencias Matemáticas (CSIC-UAM-UC3M-UCM), 28049 Madrid, Spain
[3]Department of Applied Physics, University of Geneva, 1211 Geneva, Switzerland

⋆ joserapa@ucm.es

Abstract

We present an algorithm for constructing tensor-train representations of functions. The method only requires black-box access to the target function and a small set of sample points defining the domain of interest. Thus, it is particularly well suited for tensorizing machine learning models, where the domain of interest is naturally defined by the training dataset. We show that this approach can be used to enhance the privacy and interpretability of neural network models. Specifically, we apply our decomposition to (i) obfuscate neural networks whose parameters encode patterns tied to the training data distribution, and (ii) estimate topological phases of matter that are easily accessible from the tensor train representation. Additionally, we show that this tensorization can serve as an efficient initialization method for optimizing tensor trains in general settings, and that, for model compression, our algorithm achieves a superior trade-off between memory and time complexity compared to conventional tensorization methods of neural networks.

1 Introduction

Despite their successes, neural networks (NNs) have a significant drawback that limits their use in many applications: they function as *black boxes*, with no direct means of understanding their inner workings. Interpreting the output of a NN typically requires *post hoc* techniques to explain its decisions [1, 2]. An alternative approach is to develop models that are inherently interpretable [3, 4]. However, in practice, improvements in interpretability tend to be linked to limitations in expressivity.

Another significant concern with NNs is privacy [5, 6]. The lack of interpretability makes it difficult to determine whether sensitive information has been memorized. Current techniques aim to mitigate this risk by training models through reinforcement learning with human feedback, or using the framework of differential privacy [7–10]. These solutions are computationally expensive and negatively impacts model performance, including accuracy and fairness [11]. Moreover, these methods primarily aim to protect against *membership inference attacks*, which reveal the presence of specific data points in a dataset [12]. However, they offer limited protection against *property inference attacks*, which aim to extract properties of the entire training dataset [13].

Among the possible solutions to these issues are Tensor Network (TN) models. Tensor networks provide efficient representations of high-dimensional tensors with low-rank structure. Originating in condensed matter theory, they were developed as tractable representations of quantum many-body states, facilitating the study of entanglement, symmetries, and phase transitions [14–16]. Due to their practical efficiency, TNs have become a fundamental tool to simulate large quantum systems [17–22]. Recently, TNs have been proposed as machine learning models for broader tasks. Initial studies in

Machine Learning and the Physical Sciences Workshop, NeurIPS 2025.

this area [23, 24] implemented 1D models, known as Tensor Trains (TTs) [25] or Matrix Product States (MPS) in the physics literature [26]. Subsequent research introduced the use of more complex tensor networks [27–30].

TNs have been shown to outperform other deep and classical algorithms on tabular data in anomaly detection tasks [31]. Furthermore, from their successful application in representing complex quantum systems, TNs are known to be effective in modeling probability distributions, especially when the network layout aligns with the data structure (e.g., using 2D PEPS for image modeling [30]). This structural alignment enables TNs to access critical information directly, such as the correlations within the data (i.e., the entanglement structure when studying quantum systems), or efficient computation of marginal and conditional distributions.

Motivated by this, we aim to employ low-rank tensor decompositions to transform the entire NN black-box into a single TN, thereby enhancing interpretability and privacy in the reconstructed model. To achieve this, we draw on the ideas of cross interpolation [32] and of sketching [33] for approximating probability distributions with TTs. The resulting algorithm, Tensor Train via Recursive Sketching from Samples (TT-RSS), is better suited for decomposing NNs when provided with a set of training samples that define the subregion of interest within the model's domain. Using it, we achieve promising results in approximating classification models, and we demonstrate that the resulting tensorized models are more private and interpretable than the original ones. Moreover, we discuss how this tensorization method can serve as a general approach to randomly initialize TTs, and we argue that decomposing NNs in this manner can effectively compress models, achieving a better trade-off between memory and time efficiency compared to alternative methods [34–37]. These two are pressing problems in the area of machine learning based on TNs. An implementation of TT-RSS is available in the open-source Python package TensorKrowch [38].

2 Tensor trains via recursive sketching from samples: main idea

In this work we are interested in decomposing n-dimensional continuous functions $f : X_1 \times \cdots \times X_n \to \mathbb{R}$, with $X_1, \ldots, X_n \subset \mathbb{R}$ into TT form. We choose scalar functions for simplicity, since the extension to vector-valued functions is straightforward. A function f admits a TT representation with ranks $r_1, \ldots, r_{n-1}$ if there exist functions, also referred to as *cores*, $G_1 : X_1 \times [r_1] \to \mathbb{R}$, $G_k : [r_{k-1}] \times X_k \times [r_k] \to \mathbb{R}$ for all $k \in \{2, \ldots, n-1\}$, and $G_n : [r_{n-1}] \times X_n \to \mathbb{R}$, where $[r_k] = \{1, \ldots, r_k\}$, such that

$$f(x_1, \ldots, x_n) = G_1(x_1, \alpha_1) G_2(\alpha_1, x_2, \alpha_2) \cdots G_{n-1}(\alpha_{n-2}, x_{n-1}, \alpha_{n-1}) G_n(\alpha_{n-1}, x_n) \tag{1}$$

for all $(x_1, \ldots, x_n) \in X_1 \times \cdots \times X_n$. We use Einstein's convention to implicitly sum over the repeated indices $\{\alpha_1, \ldots, \alpha_{n-1}\}$.

When the function f is discrete, the cores are discrete tensors. Otherwise, the cores are continuous functions, expressed as linear combinations of a small set of embedding functions such that $G_k(\alpha_{k-1}, x_k, \alpha_k) = \breve{G}_k(\alpha_{k-1}, i_k, \alpha_k)\phi_k(i_k, x_k)$. For simplicity, below we consider a discrete function $f : [d_1] \times \cdots \times [d_n] \to \mathbb{R}$ and assume it admits a TT representation with ranks $r_1, \ldots, r_{n-1}$. Additionally, we are given a set of N pivots, $\mathbf{x} = \{(\mathbf{x}_1^i, \ldots, \mathbf{x}_n^i) : \mathbf{x}_j^i \in [d_j]\}_{i \in [N]}$, which represent points defining a region of interest within the domain of f (e.g., training points from the model).

The algorithm is based on the observation that f can be treated as a low-rank matrix. This means that there exists a decomposition of each unfolding matrix of f of the form

$$f(x_{1:k}, x_{k+1:n}) = \Phi_k(x_{1:k}, \alpha_k)\Psi_k(\alpha_k, x_{k+1:n}), \tag{2}$$

where $\alpha_k \in [r_k]$, $x_{a:b} = (x_a, x_{a+1}, \ldots, x_b)$, and Φ_k and Ψ_k are matrices whose column and row spaces, respectively, span the column and row spaces of f. Since we assume that f admits a TT representation, one possible decomposition of its k-th unfolding matrix is obtained by contracting the first k and the last $n-k$ cores, respectively. Therefore, if we can efficiently obtain a set of vectors spanning the column space of the unfolding matrices of f, we can define

$$\Phi_k(x_{1:k}, \alpha_k) = \Phi_{k-1}(x_{1:k-1}, \alpha_{k-1}) G_k(\alpha_{k-1}, x_k, \alpha_k), \tag{3}$$

from which each core can be determined. This result is formalized and proved in Ref. [33]. To implement this idea efficiently, we construct the column spaces of f by restricting to a subset of columns indexed by the elements in $\{\mathbf{x}_{k+1:n}^i\}_{i \in [N_{k+1:n}]}$ and then applying a random projection.

This is achieved by defining projections $T_{k+1} : [d_{k+1}] \times \cdots \times [d_n] \times [N_{k+1:n}] \to \mathbb{R}$, such that $\Phi_k(x_{1:k}, \alpha_k) = f(x_{1:k}, x_{k+1:n})\, T_{k+1}(x_{k+1:n}, \alpha_k)$ for $k \in [n-1]$. Moreover, to reduce the number of equations to solve, we define $S_{k-1} : [N_{1:k-1}] \times [d_1] \times \cdots \times [d_{k-1}] \to \mathbb{R}$ to restrict to the rows indexed by the elements in $\{\mathbf{x}^i_{1:k}\}_{i \in [N_{1:k}]}$. Using both S and T, the equations to be solved are

$$S_{k-1}(\beta_{k-1}, x_{1:k-1})\Phi_{k-1}(x_{1:k-1}, \alpha_{k-1})G_k(\alpha_{k-1}, x_k, \alpha_k) = S_{k-1}(\beta_{k-1}, x_{1:k-1})\Phi_k(x_{1:k}, \alpha_k). \tag{4}$$

This way, we define sketches such that applying them entails evaluating f at a polynomial number of relevant points (e.g., elements from the training set when decomposing NNs) to construct the tensor $f(\mathbf{x}_{1:k-1}, [d_k], \mathbf{x}_{k+1:n})$, making the method efficient for any type of function. In contrast, TT-RS, the previous approach from Ref. [33], uses random sketches for S and T, incurring a computational cost that scales exponentially with n.

The full algorithm is shown in the pseudo-code below

Algorithm 1 Tensor Train via Recursive Sketching from Samples

Require: Discrete function $f : [d_1] \times \cdots \times [d_n] \to \mathbb{R}$.
Require: Sketch samples $\mathbf{x} = \{(\mathbf{x}^i_1, \ldots, \mathbf{x}^i_n) : \mathbf{x}^i_j \in [d_j]\}_{i=1}^N$.
Require: Target ranks $r_1, \ldots, r_{n-1}$.
$s_1, \ldots, s_{n-1}, T_2, \ldots, T_n \leftarrow$ SKETCHFORMING($\mathbf{x}$).
$\widetilde{\Phi}_1, \ldots, \widetilde{\Phi}_d \leftarrow$ SKETCHING($f, s_1, \ldots, s_{n-1}, T_2, \ldots, T_n$).
$B_1, \ldots, B_n \leftarrow$ TRIMMING($\widetilde{\Phi}_1, \ldots, \widetilde{\Phi}_n, r_1, \ldots, r_{n-1}$).
$A_1, \ldots, A_{n-1} \leftarrow$ SYSTEMFORMING($B_1, \ldots, B_{n-1}, s_1, \ldots, s_{n-1}$).
Solve via least-squares for the unknowns $G_1 : [d_1] \times [r_1] \to \mathbb{R}$, $G_k : [r_{k-1}] \times [d_k] \times [r_k] \to \mathbb{R}$ for all $k \in \{2, \ldots, n-1\}$, and $G_n : [r_{n-1}] \times [d_n] \to \mathbb{R}$:

$$\begin{aligned} G_1(x_1, \alpha_1) &= B_1(x_1, \alpha_1), \\ A_{k-1}(\beta_{k-1}, \alpha_{k-1})G_k(\alpha_{k-1}, x_k, \alpha_k) &= B_k(\beta_{k-1}, x_k, \alpha_k), \quad k \in \{2, \ldots, n-1\}, \\ A_{n-1}(\beta_{n-1}, \alpha_{n-1})G_n(\alpha_{n-1}, x_n) &= B_n(\beta_{n-1}, x_n). \end{aligned}$$

return $G_1, \ldots, G_n$

The subroutines SKETCHFORMING, SKETCHING, TRIMMING, and SYSTEMFORMING correspond, respectively, to the steps of creating the sketch functions T_{k+1} and S_{k-1} to f, forming the coefficient matrices $\widetilde{\Phi}_k$, finding the proper tensor ranks via singular value decomposition (SVD), and forming the system of equations that is eventually solved. All of them are described in detail in the long version of the work [39].

3 Applications

Now we detail several applications of TT-RSS to fulfill different purposes. All experiments described below, and additional ones, can be found in https://github.com/joserapa98/tensorization-nns.

Privacy Recently, a privacy vulnerability known as *property inference* has been observed that might affect machine learning models trained via gradient descent methods [13]. It was shown that during the learning process, the distribution of the parameters begins to exhibit biases associated with biases in certain features of the dataset. More significantly, this effect was observed even when the biases appeared in features that were not relevant to the learning task. To address this issue, it was proposed to replace NN models with TNs, whose gauge freedom is well-characterized, since his freedom allows the parameters of the models to be redefined independently of the learning process, effectively removing hidden patterns.

Here we study property inference vulnerabilities in a much more realistic scenario than that in Ref. [13]. We use the CommonVoice dataset [40], training models to classify voices by gender (male or female), and attack them to infer if the training dataset contained a majority of points with "England English" or "Canadian English" accent. Each voice sample is represented as a vector in $[0, 1]^{500}$, and the irrelevant information (the accent) cannot be easily extracted from the features available, in contrast with Ref. [13].

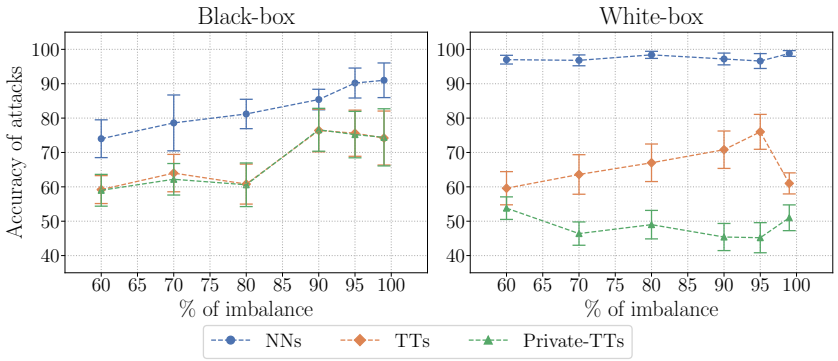

Figure 1: Accuracy of logistic regression models trained to classify whether a model was trained on a dataset with a majority of datapoints with the "England English" value for the feature `accent` or a majority of datapoints with the "Canadian English" value. For each percentage of imbalance in the training datasets of the models–in the horizontal axis–, a dataset is formed from 500 models trained on 20 different datasets (10 datasets per percentage, with 25 models per dataset). Then, an attack classifier is trained on each of these datasets, and used to predict the majority class in the training dataset of unseen models. The accuracy of such attacks is depicted in the vertical axis. For each configuration, attacks are evaluated via 10-fold validation, and mean accuracies are displayed with error bars at $\pm 0.5\sigma$, where σ denotes the standard deviation. While tensorization reduces the exposed information due to only approximating the original model, it is notably the gauge-based process which erases all information about the accent from the model parameters.

We first use fully connected NNs consisting of one 50-neuron hidden layer and a 2-neuron output layer. Between them, we use dropout and ReLU activation functions. The models are trained to minimize the negative log-likelihood using Adam. For tensorization, we select as pivots points that are not part of the training or validation sets, and we set $d = 2$, $r = 5$, and $N = 100$.

Tensorized models initially exhibit reduced accuracies, typically around 70%-75%, in contrast with the 80%-82% of the NNs. Thus, in a second step, we re-train the models for 10 epochs using the pivots. This process improves accuracy to approximately 78%-80%. This makes an acceptable tradeoff for enhanced privacy. Then, we obfuscate the TT models' parameters, by multiplying each core by independent, random, orthogonal matrices and their transposes between contractions. Since the added matrices are orthogonal, they cancel out with their transposes in adjacent contractions, preserving the TT's overall behavior.

As attacks, we use logistic regressors trained on 250 NNs for each of various percentages of imbalance between accents, for which we provide the corresponding labels. We consider two scenarios: (i) *black-box* access, where the input to the regressors is classifications made by the model on 100 predetermined samples equally divided between woman or man, and "England English" and "Canadian English" accents (none of them being training points for any of the models), and (ii) *white-box* access, where the input to the regressors is the collection of parameters of the model. The results of the attacks are depicted in Figure 1.

Interpretability In recent years, several approaches have emerged to approximate quantum states via neural network representations [41–45]. However, these representations often lack the capacity to analyze properties of the states directly [46–48]. The methodology we propose leverages pre-trained neural-network quantum states (NNQS) to derive a TN representation, enabling the direct study of such properties.

We illustrate this approach by estimating the symmetry-protected topological (SPT) phase of the Affleck-Kennedy-Lieb-Tasaki (AKLT) state [49]. We reconstruct a TT representation of the state via TT-RSS using only pivots (i.e., black-box information about the state) and use this representation to compute the order parameter defined in Ref. [50]. We reconstruct perfectly the AKLT state from just 14 pivots, even for states up to 500 sites. Notably, the order parameter reaches its expected value, namely -1, with as few as 6 pivots. This demonstrates that our method can estimate order parameters of systems with hundreds of sites in just a few seconds, in sharp contrast to the alternatives [46–48].

Initialization and compression As a side effect of tensorizing machine learning models via TT-RSS for privacy and interpretability, we observe two additional phenomena that can be independently exploited and address relevant problems in machine learning based on tensor networks. On the one hand, if the TT obtained through TT-RSS exhibits lower accuracies compared to the original NN models, these TTs could still serve as starting points for further optimization. This suggests that TT-RSS could be utilized as an initialization mechanism for TT machine learning models. On the other, TT-RSS allows us to leverage the efficiency of tensor networks. By tensorizing NN models, we can produce TT models that use fewer parameters, leading to reduced memory and computing costs.

We have performed initial assessments of the capabilities of TT-RSS in these two fronts. We find that using as initialization for a TT model the tensorization of a simple model (such as a logistic regressor or a shallow NN) trained in the training dataset clearly outperforms the alternative available strategies (see Table 1). Regarding model compression, we have compared TT-RSS with other common techniques that tensorize NNs in a layer-wise fashion, finding *a priori* superior memory-time efficiency tradeoffs, albeit the results are currently limited to moderately small models.

Table 1: Training accuracies of TT models trained via the Adam optimizer (learning rate = 10^{-5}, weight decay = 10^{-10}) for the classification task used in the privacy experiments, initialized via TT-RSS applied on the NN models trained there, TT-RSS applied on the uniform distribution, stacks of Haar-random unitary matrices, and Gaussianly and later brought to SVD-based canonical form.

Training steps	TT-RSS	TT-RSS random	Unitaries	Canonical
0	80.40	51.88	45.07	48.93
100	87.01	54.20	81.93	50.29
200	89.45	55.35	82.57	50.22

4 Conclusions

In this work, we have presented a tensorization scheme that combines ideas from sketching and cross interpolation, resulting in an efficient method for function decomposition, TT-RSS. The key idea is having black-box access to the function, together with a small set of points (the pivots) that define a subregion of interest within the function's domain.

While applicable to general functions, this method is particularly well-suited for machine learning models, where the subregion of interest is defined by the training dataset. Using only a small number of these samples, we demonstrate that it is possible to reconstruct models in TT form for tasks such as image and audio classification.

We demonstrate that TT-RSS can be used to obfuscate NN models, enhancing privacy protection through the well-characterized gauge freedom of TTs. Additionally, we have found that TT-RSS leads to representations that are interpretable. Concretely, it is possible to estimate physical quantities (in our example, the SPT phase of the AKLT state) from them. Thus, TT-RSS (and extensions to higher-dimensional cases) offer a promising path towards extracting physical properties of systems represented by neural network quantum states [43].

Moreover, TT-RSS can serve as an initialization technique for training TT models. One can first train a simpler model, such as a linear or logistic model, to obtain a function that returns values within a controlled range across the domain. Then, applying TT-RSS allows the construction of a TT model with the desired embedding, providing stable results in the domain of interest. We also show indications that it can perform as a compression algorithm, reducing both the number of operations and the number of variables compared to the original models.

In conclusion, the approach presented here provides a promising framework for efficiently decomposing NN models into TN forms, offering improvements in privacy protection, model interpretability, and computational efficiency. The potential for extending these methods to higher-dimensional problems is particularly compelling, as it could address the computational challenges inherent in training TNs in such settings while retaining their advantages over standard NN models.

Acknowledgments

This work is supported by the Spanish Ministry of Science and Innovation MCIN/AEI/10.13039/5011 00011033 (CEX2023-001347-S, CEX2019-000904-S, CEX2019-000904-S-20-4, PID2020-1135 23GB-I00, PID2023-146758NB-I00), Comunidad de Madrid (QUITEMAD-CM P2018/TCS-4342, TEC-2024/ COM-84-QUITEMAD-CM), Universidad Complutense de Madrid (FEI-EU-22-06), the CSIC Quantum Technologies Platform PTI-001, the Swiss National Science Foundation (grant number 224561), and the Ministry for Digital Transformation and of Civil Service of the Spanish Government through the QUANTUM ENIA project call – Quantum Spain project, and by the European Union through the Recovery, Transformation and Resilience Plan – NextGenerationEU within the framework of the Digital Spain 2026 Agenda. This research was supported in part by Perimeter Institute for Theoretical Physics. Research at Perimeter Institute is supported by the Government of Canada through the Department of Innovation, Science, and Economic Development, and by the Province of Ontario through the Ministry of Colleges and Universities.

References

[1] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM Comput. Surv.*, 51(5):93, August 2018. ISSN 0360-0300. doi: 10.1145/3236009.

[2] Yu Zhang, Peter Tino, Ales Leonardis, and Ke Tang. A survey on neural network interpretability. *IEEE Trans. Emerg. Top. Comput. Intell.*, 5(5):726–742, October 2021. ISSN 2471-285X. doi: 10.1109/tetci.2021.3100641.

[3] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.*, 1:206–215, 2019. doi: 10.1038/s42256-019-0048-x.

[4] Chaofan Chen, Oscar Li, Chaofan Tao, Alina Jade Barnett, Jonathan Su, and Cynthia Rudin. This looks like that: deep learning for interpretable image recognition. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2019. Curran Associates Inc. URL `https://proceedings.neurips.cc/paper_files/paper/2019/hash/adf7ee2dcf142b0e11888e72b43fcb75-Abstract.html`.

[5] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8821–8831. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/ramesh21a.html`.

[6] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models . In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, Los Alamitos, CA, USA, June 2022. IEEE Computer Society. doi: 10.1109/CVPR52688.2022.01042.

[7] Cynthia Dwork. Differential privacy. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer Berlin Heidelberg, 2006. doi: 10.1007/11787006_1. `https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/dwork.pdf`.

[8] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284. Springer Berlin Heidelberg, 2006. doi: 10.1007/11681878_14. `https://iacr.org/archive/tcc2006/38760266/38760266.pdf`.

[9] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, 2014. ISSN 1551-305X. doi: 10.1561/0400000042. `https://www.cis.upenn.edu/~aaroth/Papers/privacybook.pdf`.

[10] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS'16, pages 308–318. ACM, October 2016. doi: 10.1145/2976749.2978318.

[11] Eugene Bagdasaryan and Vitaly Shmatikov. Differential privacy has disparate impact on model accuracy. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Adv. Neural Inf. Process. Syst.*, volume 32, Red Hook, NY, USA, 2019. Curran Associates Inc. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/fc0de4e0396fff257ea362983c2dda5a-Abstract.html.

[12] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, 2017. doi: 10.1109/SP.2017.41.

[13] Alejandro Pozas-Kerstjens, Senaida Hernández-Santana, José Ramón Pareja Monturiol, Marco Castrillón López, Giannicola Scarpa, Carlos E. González-Guillén, and David Pérez-García. Privacy-preserving machine learning with tensor networks. *Quantum*, 8:1425, July 2024. ISSN 2521-327X. doi: 10.22331/q-2024-07-25-1425.

[14] Román Orús. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Ann. Phys.*, 349:117–158, 2014. ISSN 0003-4916. doi: 10.1016/j.aop.2014.06.013. URL https://www.sciencedirect.com/science/article/pii/S0003491614001596.

[15] Jacob C. Bridgeman and Christopher T. Chubb. Hand-waving and interpretive dance: an introductory course on tensor networks. *J. Phys. A: Math. Theor.*, 50(22):223001, may 2017. doi: 10.1088/1751-8121/aa6dc3.

[16] J. Ignacio Cirac, David Pérez-García, Norbert Schuch, and Frank Verstraete. Matrix product states and projected entangled pair states: Concepts, symmetries, theorems. *Rev. Mod. Phys.*, 93:045003, Dec 2021. doi: 10.1103/RevModPhys.93.045003. URL https://link.aps.org/doi/10.1103/RevModPhys.93.045003.

[17] Yao-Yun Shi, Lu-Ming Duan, and Guifré Vidal. Classical simulation of quantum many-body systems with a tree tensor network. *Phys. Rev. A*, 74(2):022320, 2006. doi: 10.1103/PhysRevA.74.022320. URL https://link.aps.org/doi/10.1103/PhysRevA.74.022320.

[18] Luca Tagliacozzo, Glen Evenbly, and Guifré Vidal. Simulation of two-dimensional quantum systems using a tree tensor network that exploits the entropic area law. *Phys. Rev. B*, 80:235127, Dec 2009. doi: 10.1103/PhysRevB.80.235127. URL https://link.aps.org/doi/10.1103/PhysRevB.80.235127.

[19] Valentin Murg, Frank Verstraete, Örs Legeza, and Reinhard M. Noack. Simulating strongly correlated quantum systems with tree tensor networks. *Phys. Rev. B*, 82:205105, Nov 2010. doi: 10.1103/PhysRevB.82.205105. URL https://link.aps.org/doi/10.1103/PhysRevB.82.205105.

[20] Feng Pan, Keyang Chen, and Pan Zhang. Solving the sampling problem of the sycamore quantum circuits. *Phys. Rev. Lett.*, 129:090502, Aug 2022. doi: 10.1103/PhysRevLett.129.090502. URL https://link.aps.org/doi/10.1103/PhysRevLett.129.090502.

[21] Changhun Oh, Minzhao Liu, Yuri Alexeev, Bill Fefferman, and Liang Jiang. Tensor network algorithm for simulating experimental Gaussian boson sampling. *Nat. Phys.*, 20:1461–1468, 2024. doi: 10.1038/s41567-024-02535-8.

[22] Joseph Tindall, Matthew Fishman, E. Miles Stoudenmire, and Dries Sels. Efficient tensor network simulation of IBM's Eagle kicked Ising experiment. *PRX Quantum*, 5:010308, Jan 2024. doi: 10.1103/PRXQuantum.5.010308. URL https://link.aps.org/doi/10.1103/PRXQuantum.5.010308.

[23] Edwin Stoudenmire and David J. Schwab. Supervised learning with tensor networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Adv. Neural Inf. Process. Syst.*, volume 29, pages 4799–4807. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/hash/5314b9674c86e3f9d1ba25ef9bb32895-Abstract.html.

[24] Alexander Novikov, Mikhail Trofimov, and Ivan V. Oseledets. Exponential machines. *Bull. Pol. Acad. Sci. Tech. Sci.*, 66(No 6 (Special Section on Deep Learning: Theory and Practice)):789–797, 2018. doi: 10.24425/bpas.2018.125926. URL https://journals.pan.pl/dlibra/publication/125926/edition/109868/content.

[25] Ivan Oseledets. Tensor-train decomposition. *SIAM J. Sci. Comput.*, 33(5):2295–2317, 2011. doi: 10.1137/090752286.

[26] David Pérez-García, Frank Verstraete, Michael M. Wolf, and J. Ignacio Cirac. Matrix product state representations. *Quantum Inf. Comput.*, 7(5):401—430, July 2007. ISSN 1533-7146. doi: 10.26421/QIC7.5-6-1.

[27] Ding Liu, Shi-Ju Ran, Peter Wittek, Cheng Peng, Raul Blázquez García, Gang Su, and Maciej Lewenstein. Machine learning by unitary tensor network of hierarchical tree structure. *New J. Phys.*, 21(7):073059, jul 2019. doi: 10.1088/1367-2630/ab31ef.

[28] Song Cheng, Lei Wang, Tao Xiang, and Pan Zhang. Tree tensor networks for generative modeling. *Phys. Rev. B*, 99:155131, Apr 2019. doi: 10.1103/PhysRevB.99.155131. URL https://link.aps.org/doi/10.1103/PhysRevB.99.155131.

[29] Justin A. Reyes and E. Miles Stoudenmire. Multi-scale tensor network architecture for machine learning. *Mach. Learn.: Sci. Technol.*, 2(3):035036, jul 2021. doi: 10.1088/2632-2153/abffe8.

[30] Tom Vieijra, Laurens Vanderstraeten, and Frank Verstraete. Generative modeling with projected entangled-pair states, 2022.

[31] Jinhui Wang, Chase Roberts, Guifré Vidal, and Stefan Leichenauer. Anomaly detection with tensor networks, 2020.

[32] Sergey Dolgov and Dmitry Savostyanov. Parallel cross interpolation for high-precision calculation of high-dimensional integrals. *Comput. Phys. Commun.*, 246:106869, 2020. ISSN 0010-4655. doi: 10.1016/j.cpc.2019.106869. URL https://www.sciencedirect.com/science/article/pii/S0010465519302565.

[33] YoonHaeng Hur, Jeremy G. Hoskins, Michael Lindsey, E.M. Stoudenmire, and Yuehaw Khoo. Generative modeling via tensor train sketching. *App. Comput. Harmon. Anal.*, 67:101575, 2023. ISSN 1063-5203. doi: 10.1016/j.acha.2023.101575. URL https://www.sciencedirect.com/science/article/pii/S1063520323000623.

[34] Alexander Novikov, Dmitrii Podoprikhin, Anton Osokin, and Dmitry P. Vetrov. Tensorizing neural networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Adv. Neural Inf. Process. Syst.*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/hash/6855456e2fe46a9d49d3d3af4f57443d-Abstract.html.

[35] Xindian Ma, Peng Zhang, Shuai Zhang, Nan Duan, Yuexian Hou, Ming Zhou, and Dawei Song. A tensorized transformer for language modeling. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Adv. Neural Inf. Process. Syst.*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/dc960c46c38bd16e953d97cdeefdbc68-Abstract.html.

[36] Ze-Feng Gao, Song Cheng, Rong-Qiang He, Z. Y. Xie, Hui-Hai Zhao, Zhong-Yi Lu, and Tao Xiang. Compressing deep neural networks by matrix product operators. *Phys. Rev. Res.*, 2:023300, Jun 2020. doi: 10.1103/PhysRevResearch.2.023300. URL https://link.aps.org/doi/10.1103/PhysRevResearch.2.023300.

[37] Viktoriia Chekalina, Georgiy Novikov, Julia Gusak, Alexander Panchenko, and Ivan Oseledets. Efficient GPT model pre-training using tensor train matrix representation. In Chu-Ren Huang, Yasunari Harada, Jong-Bok Kim, Si Chen, Yu-Yin Hsu, Emmanuele Chersoni, Pranav A, Winnie Huiheng Zeng, Bo Peng, Yuxi Li, and Junlin Li, editors, *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation*, pages 600–608, Hong Kong, China, December 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.paclic-1.60/.

[38] José Ramón Pareja Monturiol, David Pérez-García, and Alejandro Pozas-Kerstjens. TensorKrowch: Smooth integration of tensor networks in machine learning. *Quantum*, 8:1364, 2024. doi: 10.22331/q-2024-06-11-1364. https://github.com/joserapa98/tensorkrowch.

[39] José Ramón Pareja Monturiol, Alejandro Pozas-Kerstjens, and David Pérez-García. Tensorization of neural networks for improved privacy and interpretability, 2025. URL https://arxiv.org/abs/2501.06300.

[40] Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. Common voice: A massively-multilingual speech corpus. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4218–4222, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.520/.

[41] Yusuke Nomura, Andrew S. Darmawan, Youhei Yamaji, and Masatoshi Imada. Restricted Boltzmann machine learning for solving strongly correlated quantum systems. *Phys. Rev. B*, 96:205152, Nov 2017. doi: 10.1103/PhysRevB.96.205152. URL https://link.aps.org/doi/10.1103/PhysRevB.96.205152.

[42] Xun Gao and Lu-Ming Duan. Efficient representation of quantum many-body states with deep neural networks. *Nat. Commun.*, 8:662, September 2017. ISSN 2041-1723. doi: 10.1038/s41467-017-00705-2.

[43] Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, February 2017. ISSN 1095-9203. doi: 10.1126/science.aag2302.

[44] Mohamed Hibat-Allah, Martin Ganahl, Lauren E. Hayward, Roger G. Melko, and Juan Carrasquilla. Recurrent neural network wave functions. *Phys. Rev. Res.*, 2:023358, Jun 2020. doi: 10.1103/PhysRevResearch.2.023358. URL `https://link.aps.org/doi/10.1103/PhysRevResearch.2.023358`.

[45] David Fitzek, Yi Hong Teoh, Hin Pok Fung, Gebremedhin A. Dagnew, Ejaaz Merali, M. Schuyler Moss, Benjamin MacLellan, and Roger G. Melko. RydbergGPT, 2024.

[46] Juan Carrasquilla and Roger G. Melko. Machine learning phases of matter. *Nat. Phys.*, 13(5):431–434, February 2017. ISSN 1745-2481. doi: 10.1038/nphys4035.

[47] Evert P. L. van Nieuwenburg, Ye-Hua Liu, and Sebastian D. Huber. Learning phase transitions by confusion. *Nat. Phys.*, 13(5):435–439, February 2017. ISSN 1745-2481. doi: 10.1038/nphys4037.

[48] Mohamed Hibat-Allah, Roger G. Melko, and Juan Carrasquilla. Investigating topological order using recurrent neural networks. *Phys. Rev. B*, 108:075152, Aug 2023. doi: 10.1103/PhysRevB.108.075152. URL `https://link.aps.org/doi/10.1103/PhysRevB.108.075152`.

[49] Ian Affleck, Tom Kennedy, Elliott H. Lieb, and Hal Tasaki. Rigorous results on valence-bond ground states in antiferromagnets. *Phys. Rev. Lett.*, 59:799–802, Aug 1987. doi: 10.1103/PhysRevLett.59.799. URL `https://link.aps.org/doi/10.1103/PhysRevLett.59.799`.

[50] Jutho Haegeman, David Pérez-García, Ignacio Cirac, and Norbert Schuch. Order parameter for symmetry-protected phases in one dimension. *Phys. Rev. Lett.*, 109:050402, Jul 2012. doi: 10.1103/PhysRevLett.109.050402. URL `https://link.aps.org/doi/10.1103/PhysRevLett.109.050402`.