

Article

Probabilistic Modeling with Matrix Product States

James Stokes ^{1,*} and John Terilla ²

¹ Flatiron Institute, New York, NY 10010, USA

² Tunnel, New York, NY 10001, USA; john@tunnel.tech

* Correspondence: jstokes@flatironinstitute.org

Received: 13 November 2019; Accepted: 12 December 2019; Published: 17 December 2019



Abstract: Inspired by the possibility that generative models based on quantum circuits can provide a useful inductive bias for sequence modeling tasks, we propose an efficient training algorithm for a subset of classically simulable quantum circuit models. The gradient-free algorithm, presented as a sequence of exactly solvable effective models, is a modification of the density matrix renormalization group procedure adapted for learning a probability distribution. The conclusion that circuit-based models offer a useful inductive bias for classical datasets is supported by experimental results on the parity learning problem.

Keywords: machine learning; density matrix renormalization group; quantum information

1. Introduction

The possibility of exponential speedups for certain linear algebra operations has inspired a wave of research into quantum algorithms for machine learning purposes [1]. Many of these exponential speedups hinge on assumptions of fault tolerant quantum devices and efficient data preparation, which are unlikely to be realized in the near future. Focus has thus shifted to hybrid quantum-classical algorithms which involve optimizing the parameters of a variational quantum circuit to prepare a desired quantum state and have the potential to be implemented on near-term intermediate scale quantum devices [2].

Hybrid quantum-classical algorithms have been found to solve difficult eigenvalue problems [3] and to perform hard combinatorial optimization [4]. A number of recent works consider unsupervised learning within the hybrid quantum-classical framework [5–9].

In the context of machine learning, as emphasized in [2], it is less clear if variational hybrid quantum-classical algorithms offer advantages over existing purely classical algorithms. Density estimation, which attempts to learn a probability distribution from training data, has been suggested as an area to look for advantages [7] because a quantum advantage has been identified in the ability of quantum circuits to sample from certain probability distributions that are hard to sample classically [10]. In high-dimensional density estimation relevant to machine learning, expressive power is only part of the story and indeed algorithms in high-dimensional regime rely crucially on their inductive bias. Do the highly expressive probability distributions implied by quantum circuits offer a useful inductive bias for modeling high-dimensional classical data? We address this question in this paper.

We work within the confines of a classically tractable subset of quantum states modeled by tensor networks, which may be thought of as those states that can be prepared by shallow quantum circuits. Even more narrowly, we restrict to matrix product states akin to one-dimensional shallow circuits. Mathematically, tensor networks are a graphical calculus for describing interrelated matrix factorizations for which there exist polylogarithmic algorithms for a restricted set of linear algebra computations. We propose an unsupervised training algorithm for a generative model inspired by the

density matrix renormalization group (DMRG) procedure. The training dynamics take place on the unit sphere of a Hilbert space, where in contrast to many variational methods, a state is modified in a sequence of deterministic steps that do not involve gradients. The efficient access to certain vector operations afforded by the tensor network ansatz allows us to implement our algorithm in a purely classical fashion.

We experimentally probe the inductive bias of the model by training on the dataset P_{20} consisting of bitstrings of length 20 having an even number of 1 bits. The algorithm rapidly learns the uniform distribution on P_{20} to high precision, indicating that the tensor network quantum circuit model provides a useful inductive bias for this classical dataset and the resulting trained model is small, only 336 parameters. The P_{20} dataset can be frustrating to learn for other models, such as restricted Boltzmann machines (RBMs) trained with gradient-based methods. The difficulty of training RBMs to learn parity with contrastive divergence and related training algorithms is noted in [11]. The difficulty for other gradient based deep-learning methods on parity problems has been studied in [12]. To put the work in this paper in context, we note that generative modeling using tensor networks has been considered for several datasets for which classical neural models trained with gradient based methods are successful [13,14]. We also note that shallow quantum circuits have already been successful for a related supervised parity classification problem [15].

In an effort to improve accessibility, we avoid the language of quantum-many body physics and quantum information and explain the algorithm and results in terms of elementary linear algebra and statistics. While this means some motivational material is omitted, we believe it sharpens the exposition. One exception is the visual language of tensor networks where the benefits of simplifying tensor contractions outweigh the costs of using elementary, but cumbersome, notation. We refer readers unfamiliar with tensor network notation to [16–19] or to the many other surveys.

The organization of the paper is as follows. In Section 2 we state the optimization problem at the population level and propose a finite-sample estimator. In Sections 3 and 4 we describe an abstract discrete-time dynamical system evolving on the unit sphere of Hilbert space which optimizes our empirical objective by exactly solving an effective problem in a sequence of isometrically embedded Hilbert subspaces. In Section 5 we provide a concrete realization of this dynamical system for a class of tensor networks called matrix product states. Section 6 outlines experiments demonstrating that the proposed iterative solver successfully learns the parity language using limited data.

2. The Problem Formulation

Recall that a unit vector ψ in a finite-dimensional Hilbert space \mathcal{H} defines a probability distribution P_ψ on any orthonormal basis by setting the probability of each basis vector e to be

$$P_\psi(e) := |\langle \psi, e \rangle|^2. \quad (1)$$

We refer to the probability distribution P_ψ in Equation (1) as the *Born distribution* induced by ψ .

Let π be a probability distribution on a finite set \mathcal{X} and fix a field of scalars, either \mathbb{R} or \mathbb{C} . Let \mathcal{H} be the free vector space on the set \mathcal{X} . Use $|x\rangle$ to denote the vector in \mathcal{H} corresponding to the element $x \in \mathcal{X}$. The space \mathcal{H} has a natural inner product defined by declaring the vectors $\{|x\rangle : x \in \mathcal{X}\}$ to be an orthonormal basis.

Define a unit vector $\psi_\pi \in \mathcal{H}$ by

$$\psi_\pi := \sum_{x \in \mathcal{X}} \sqrt{\pi(x)} |x\rangle. \quad (2)$$

Notice that ψ_π realizes π as a Born distribution:

$$\pi(x) = P_{\psi_\pi}(|x\rangle) \text{ for all } x \in \mathcal{X}. \quad (3)$$

The formula for ψ_π as written in Equation (2) involves perfect knowledge of π and unrestricted access to the Hilbert space \mathcal{H} . This paper is concerned with situations when knowledge about π is limited to a finite number of training examples, and ψ is restricted to some tractable subset \mathcal{M} of the unit sphere.

At the population level, the problem to be solved is to find the closest approximation ψ_* to ψ_π within \mathcal{M} ,

$$\psi_* := \arg \min_{\psi \in \mathcal{M}} \|\psi - \psi_\pi\|.$$

We assume access to a sequence $(X_i)_{i=1}^n$ of samples drawn independently from π , giving rise to the associated empirical distribution

$$\hat{\pi}(x) := \frac{1}{n} \sum_{i=1}^n \delta_{X_i}(x). \tag{4}$$

It is natural to define the following estimator whose Born distribution coincides with the empirical distribution

$$\psi_{\hat{\pi}} = \sum_{x \in \mathcal{X}} \sqrt{\hat{\pi}(x)} |x\rangle. \tag{5}$$

We are thus led to consider the following optimization problem.

Problem 1. Given a sequence $\{X_i\}_{i=1}^n$ of i.i.d. samples drawn from π and a subset $\mathcal{M} \subseteq \{\psi \in \mathcal{H} : \|\psi\| = 1\}$ of the unit sphere in \mathcal{H} , find

$$\hat{\psi} := \arg \min_{\psi \in \mathcal{M}} \|\psi - \psi_{\hat{\pi}}\|.$$

Our proposal differs from existing literature on Born Machines which have employed log-likelihood objective functions minimized by gradient descent (see [20] for a review). As we will see, the choice of loss function as the l_2 norm allows analytical updates with guaranteed improvement. This should be contrasted with the log-likelihood objective for which no such guarantee exists and gradient descent may diverge if the learning rate is not chosen appropriately.

Although the problem formulation contains no explicit regularization term, regularization is achieved implicitly by controlling the complexity of the model class \mathcal{M} . In the experiments section, the model hypothesis class is defined by a small integer hyperparameter called bond-dimension. We solve the problem for several choices of bond-dimension using a held-out test set to measure overfitting and generalization. In the case where \mathcal{X} consists of strings, the associated Hilbert space \mathcal{H} has a dimension that is exponential in the string length. The model hypothesis class \mathcal{M} should be chosen so that the induced Born distribution $P_{\hat{\psi}}$ offers a useful inductive bias for modeling high-dimensional probability distributions over the space of sequences. We note, as an aside, that the plug-in estimator $\|\psi - \psi_{\hat{\pi}}\|$ is a biased estimator of the population objective $\|\psi - \psi_\pi\|$.

3. Outline of Our Approach to Solving the Problem

We present an algorithm that, given a fixed realization of data $(x_1, \dots, x_n) \in \mathcal{X}^n$ and an initial state $\psi_0 \in \mathcal{M}$, produces a deterministic sequence $\{\psi_t\}_{t \geq 0}$ of unit vectors in \mathcal{M} . The algorithm is a variation of the density matrix renormalization group (DMRG) procedure which we call *exact single-site DMRG* in which each step produces a vector closer to $\psi_{\hat{\pi}}$. The sequence is defined inductively as follows: given ψ_t , the inductive step defines a subspace \mathcal{H}_{t+1} of \mathcal{H} , which also contains ψ_t . Then ψ_{t+1} is defined to be the vector in \mathcal{H}_{t+1} closest to $\psi_{\hat{\pi}}$. Inspired by ideas from the Renormalization Group

we provide an analytic formula for ψ_{t+1} . The fact that the distance to the target vector $\psi_{\hat{\pi}}$ decreases after each iteration follows as a simple consequence of the following facts

$$\psi_t \in \mathcal{H}_{t+1} \quad \text{and} \quad \psi_{t+1} = \arg \min_{\{\psi \in \mathcal{H}_{t+1}: \|\psi\|=1\}} \|\psi_{\hat{\pi}} - \psi\|. \tag{6}$$

See Figure 1.

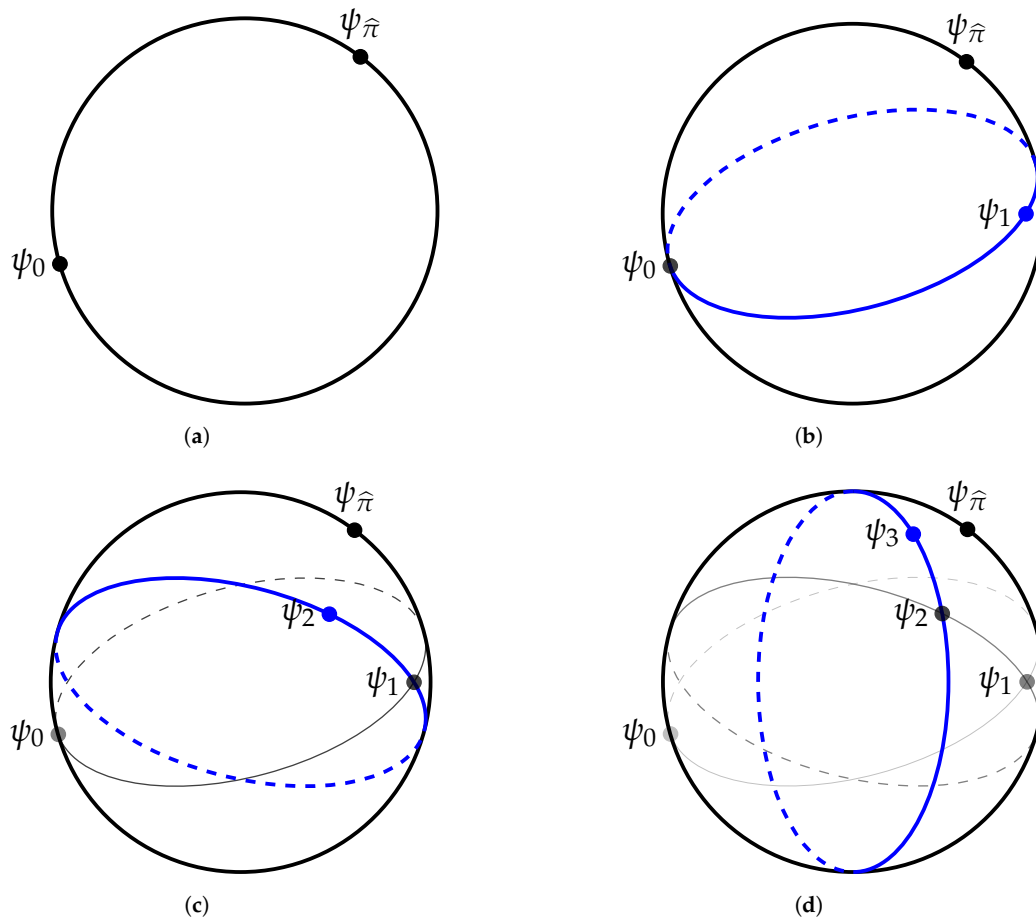
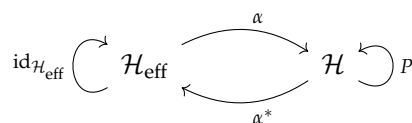


Figure 1. A bird’s eye view of the training dynamics of exact single-site DMRG on the unit sphere. (a) The initial vector ψ_0 and the vector $\psi_{\hat{\pi}}$ lie in the unit sphere of \mathcal{H} . (b) The vector ψ_0 is used to define the subspace \mathcal{H}_1 . The unit vectors in \mathcal{H}_1 define a lower dimensional sphere in \mathcal{H} (in blue). The vector ψ_1 is the vector in that sphere that is closest to $\psi_{\hat{\pi}}$. (c) The vector ψ_1 is used to define the subspace \mathcal{H}_2 . The unit sphere in \mathcal{H}_2 (in blue) contains ψ_1 but does not contain ψ_0 . The vector ψ_2 is the unit vector in \mathcal{H}_2 closest to $\psi_{\hat{\pi}}$. (d) The vector ψ_2 is used to define the subspace \mathcal{H}_3 . The vector ψ_3 is the unit vector in \mathcal{H}_3 closest to $\psi_{\hat{\pi}}$. And so on.

4. Effective Versions of the Problem

Each proposal subspace \mathcal{H}_t mentioned in the previous section will be defined as the image of an “effective” space. We begin with a general description of an effective space.

Let $\alpha : \mathcal{H}_{\text{eff}} \rightarrow \mathcal{H}$ be an isometric embedding of a Hilbert space \mathcal{H}_{eff} into \mathcal{H} . We refer to \mathcal{H}_{eff} as the effective Hilbert space. The isometry α and its adjoint map α^* are summarized by the following diagram,



The composition $\alpha^* \alpha = \text{id}_{\mathcal{H}_{\text{eff}}}$ is the identity on \mathcal{H}_{eff} . The composition in the other order $\alpha \alpha^*$ is an orthogonal projection onto $\alpha(\mathcal{H}_{\text{eff}})$ which is a subspace of \mathcal{H} isometrically isomorphic to \mathcal{H}_{eff} . Call this orthogonal projection P

$$P := \alpha \alpha^*. \tag{7}$$

The effective version of the problem formulated in Section 2 is to find the unit vector $\psi \in \alpha(\mathcal{H}_{\text{eff}})$ in the image of the effective Hilbert space that is closest to $\psi_{\hat{\pi}}$. This effective problem is solved exactly in two simple steps. The first step is orthogonal projection: $P(\psi_{\hat{\pi}})$ is the vector in $\alpha(\mathcal{H}_{\text{eff}})$ closest to $\psi_{\hat{\pi}}$. The second step is to normalize $P(\psi_{\hat{\pi}})$, which may not be a unit vector, to obtain the unit vector in $\alpha(\mathcal{H}_{\text{eff}})$ closest to $\psi_{\hat{\pi}}$.

Therefore, the analytic solution to the effective problem is $P(\psi_{\hat{\pi}}) / \|P(\psi_{\hat{\pi}})\|$ where

$$P(\psi_{\hat{\pi}}) = \alpha \alpha^* (\psi_{\hat{\pi}}) \tag{8}$$

$$= \alpha \alpha^* \left(\sum_{x \in \mathcal{X}} \sqrt{\hat{\pi}(x)} |x\rangle \right) \tag{9}$$

$$= \alpha \left(\sum_{x \in \mathcal{X}} \sqrt{\hat{\pi}(x)} \alpha^*(|x\rangle) \right). \tag{10}$$

In the exact single-site DMRG algorithm, the space $\alpha(\mathcal{H}_{\text{eff}})$ is contained within our model hypothesis class \mathcal{M} . We also offer a multi-site DMRG algorithm in the Appendix A. In this multi-site algorithm, the analytic solution to the effective problem in $\alpha(\mathcal{H}_{\text{eff}})$ does not lie in \mathcal{M} so the solution to the effective problem needs to undergo an additional “model repair” step.

Before going on to the details of the algorithm, it might be helpful to look more closely at the solution to the effective problem. For each training example x_i , call the vector $\alpha^*(|x_i\rangle) \in \mathcal{H}_{\text{eff}}$ an *effective data point*. Then, the argument of α in (10) becomes the weighted sum of effective data

$$\sum_{x \in \mathcal{X}} \sqrt{\hat{\pi}(x)} \alpha^*(|x\rangle). \tag{11}$$

The effective data are not necessarily mutually orthogonal and so the vector in (11) will not be a unit vector. One may normalize to obtain a unit vector in \mathcal{H}_{eff} and then apply α to obtain the analytic solution to the effective problem. Normalizing in \mathcal{H}_{eff} and then applying α is the same as applying α and then normalizing in \mathcal{H} since α is an isometry.

5. The Exact Single-Site DMRG Algorithm

Now specialize to the case that π is a probability distribution on a set \mathcal{X} of sequences. Suppose that $\mathcal{X} = A^N$ consists of sequences of length N in fixed alphabet $A = \{e_1, \dots, e_d\}$. The Hilbert space \mathcal{H} , defined as the free Hilbert space on \mathcal{X} , has a natural tensor product structure $V^{\otimes N}$ where V is the free Hilbert space on the alphabet A . We refer to V as the *site space*. So in this situation, the vectors $\{|e_1\rangle, \dots, |e_d\rangle\}$ are an orthonormal basis for the d -dimensional site space V and the vectors

$$|e_{i_1} e_{i_2} \dots e_{i_N}\rangle := |e_{i_1}\rangle \otimes |e_{i_2}\rangle \otimes \dots \otimes |e_{i_N}\rangle \tag{12}$$

are an orthonormal basis for the d^N dimensional space $\mathcal{H} = V^{\otimes N}$. We choose as model hypothesis class the subset $\mathcal{M} \subseteq \mathcal{H}$ consisting of normalized elements in \mathcal{H} that have a low rank matrix product state (MPS) factorization. Vectors in this model hypothesis class have efficient representations, even in

cases where the Hilbert space \mathcal{H} is of exponentially high dimension. For simplicity of presentation, we consider matrix product states with a single fixed bond space W , although everything that follows could be adapted to work with tensor networks without loops having arbitrary bond spaces.

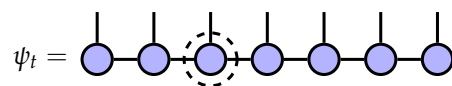
The exact single-site DMRG algorithm begins with an initial vector $\psi_0 \in \mathcal{M}$ and produces ψ_1, ψ_2, \dots inductively by solving an effective problem in the subspace

$$\mathcal{H}_{t+1} := \alpha_{t+1}(\mathcal{H}_{\text{eff},t+1}) \tag{13}$$

which we now describe. Let us drop the subscript $t + 1$ from the isometry α_{t+1} and the effective Hilbert space $\mathcal{H}_{\text{eff},t+1}$ in the relevant effective problem—just be aware that the embedding

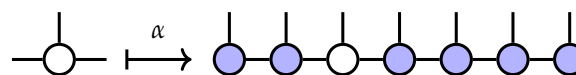
$$\alpha : \mathcal{H}_{\text{eff}} \rightarrow \mathcal{H} \tag{14}$$

will change from step to step. The map α is defined using an MPS factorization of ψ_t in mixed canonical form relative to a fixed site which varies at each step according to a predetermined schedule. For the purposes of illustration, the third site is the fixed site in the pictures below.



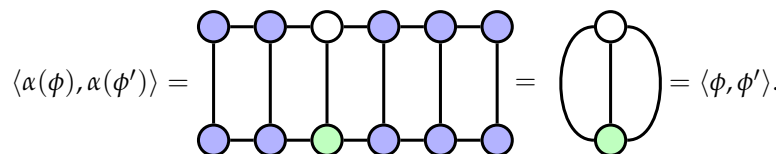
$$\psi_t = \text{MPS chain with 7 blue tensors, 3rd site fixed} \tag{15}$$

The effective space is $\mathcal{H}_{\text{eff}} = W \otimes V \otimes W$ and the isometric embedding $\alpha : W \otimes V \otimes W \rightarrow V^{\otimes N}$ is defined for any $\phi \in W \otimes V \otimes W$ by replacing the tensor at the fixed site of ψ_t with ϕ :



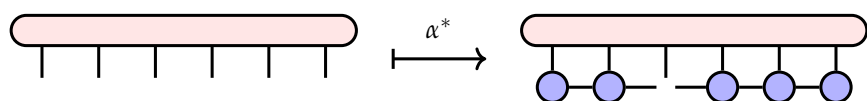
$$\text{White tensor} \xrightarrow{\alpha} \text{White tensor in blue chain} \tag{16}$$

To see that α is an isometry, use the gauge condition that the MPS factorization of ψ_t is in mixed canonical form relative to the fixed site, as illustrated below:



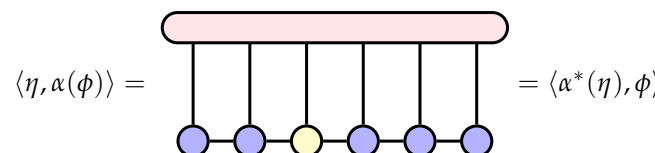
$$\langle \alpha(\phi), \alpha(\phi') \rangle = \text{MPS contraction} = \text{Loop} = \langle \phi, \phi' \rangle. \tag{17}$$

The adjoint map $\alpha^* : V^{\otimes N} \rightarrow W \otimes V \otimes W$ has a clean pictorial depiction as well.



$$\text{Red bar} \xrightarrow{\alpha^*} \text{Red bar over blue chain} \tag{18}$$

To see that α^* as pictured above is, in fact, the adjoint of α , note that for any $\eta \in \mathcal{H}$ and any $\phi \in \mathcal{H}_{\text{eff}}$, both $\langle \eta, \alpha(\phi) \rangle$ and $\langle \alpha^*(\eta), \phi \rangle$ result in the same tensor contraction:



$$\langle \eta, \alpha(\phi) \rangle = \text{Red bar over blue chain with yellow tensor} = \langle \alpha^*(\eta), \phi \rangle \tag{19}$$

In the picture above, begin with the blue tensors. Contracting with the yellow tensor gives $\alpha(\phi)$ and then contracting with the red tensor gives $\langle \eta, \alpha(\phi) \rangle$. On the other hand, first contracting with the red tensor yields $\alpha^*(\eta)$ resulting in $\langle \alpha^*(\eta), \phi \rangle$ after contracting with the yellow tensor.

Now, Equation (10) describes an analytic solution for the vector in $\mathcal{H}_{t+1} := \alpha(W \otimes V \otimes W)$ closest to $\psi_{\hat{\pi}}$. Namely, $\alpha(\phi/\|\phi\|)$ where

$$\phi = \sum_{x \in \mathcal{X}} \sqrt{\hat{\pi}(x)} \alpha^*(|x\rangle). \tag{20}$$

For each sample $|x_i\rangle = |e_{i_1} e_{i_2} \dots e_{i_N}\rangle$, the effective data point $\alpha^*(|x_i\rangle) \in V \otimes W \otimes V$ is given by the contraction

$$\alpha^*(|x_i\rangle) = \begin{array}{cccccccc} & e_{i_1} & e_{i_2} & e_{i_3} & e_{i_4} & e_{i_5} & e_{i_6} & e_{i_7} \\ & | & | & | & | & | & | & | \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ & | & | & | & | & | & | & | \\ & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \end{array} = \begin{array}{c} | \\ \bullet \\ | \end{array} \tag{21}$$

Once the effective form $\alpha^*(|x\rangle)$ of each distinct training example $|x\rangle$ has been computed, weighted by $\sqrt{\hat{\pi}(x)}$, summed, and normalized, one obtains an expression for the unit vector $\phi/\|\phi\| \in W \otimes V \otimes W$, depicted as follows,

$$\frac{\phi}{\|\phi\|} = \begin{array}{c} | \\ \bullet \\ | \end{array} \tag{22}$$

Finally, apply the map α to get ψ_{t+1} :

$$\psi_{t+1} = \begin{array}{cccccccc} & | & | & | & | & | & | & | \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ & | & | & | & | & | & | & | \end{array} \tag{23}$$

To complete the description of the exact single-site DMRG algorithm, we need to choose a schedule in which to update the tensors. We use the following schedule, organized into back-and-forth sweeps, for the fixed site at each step

$$\underbrace{1, 2, 3, \dots, N-1, N, N-1, \dots, 3, 2, 1, 2, \dots, N-1, N, N-1, \dots, 2, 1, 2, \dots}_{\text{Sweep 1}} \quad \underbrace{\dots}_{\text{Sweep 2}} \tag{24}$$

A schedule that proceeds by moving the fixed site one position at a time allows us to take advantage of two efficiencies resulting in an algorithm that is linear in both the number of training examples n and the number of sites N . One efficiency is that most of the calculations of the effective data in Equation (21) used to compute ψ_{t+1} can be reused when computing ψ_{t+2} . The second efficiency is that when inserting the updated tensor in Equation (22), it can be done so that the resulting MPS factorization of ψ_{t+1} as pictured in Equation (23) will be in mixed canonical form relative to a site adjacent to the updated tensor, which avoids a costly gauge fixing step.

6. Experiments

This section considers the problem of unsupervised learning of probability distributions on bitstrings of fixed length (Code available online: <https://github.com/TunnelTechnologies/dmrg-exact>). The first problem we consider is the parity language P_N , which consists of bitstrings of length N containing an even number of 1 bits. The goal of this task is to learn the probability distribution p which assigns uniform mass to each bitstring in P_N and zero elsewhere. More explicitly,

$$p(x) = \frac{1}{|P_N|} \mathbb{I}_{P_N}(x) = \begin{cases} \frac{1}{|P_N|}, & x \in P_N \\ 0, & x \notin P_N \end{cases} \tag{25}$$

where $\mathbb{I}_{P_N} : \{0,1\}^N \rightarrow \{0,1\}$ denotes the indicator function of the subset $P_N \subset \{0,1\}^N$. The above unsupervised learning problem is harder than the parity classification problem considered in [12] because the training signal does not exploit data labels. Of the total $|P_N| = 2^{N-1}$ such bitstrings, we reserved random disjoint subsets of size 2% for training, cross-validation and testing purposes. A NLL of $N - 1$ corresponds to the entropy of the uniform distribution on P_N . If the model memorizes

the training set, it will assign to it a negative-log-likelihood (NLL) of $N - 1 + \log_2(0.02)$ corresponding to the entropy of the uniform distribution on the training data. A NLL of N corresponds to the entropy of the uniform distribution on all bitstrings of length N . The measure of generalization performance is the gap ϵ between the NLL of the training and testing data. We performed exact single-site DMRG over the real number field using the P_{20} dataset for different choices of bond dimension, which refers to the dimensionality of the bond space W in the effective Hilbert space $\mathcal{H}_{\text{eff}} = W \otimes V \otimes W$. Training was terminated according to an early stopping criterion as determined by distance between the MPS state and the state of the cross-validation sample. Since the bond dimension controls the complexity of the model class, and since matrix product states are universal approximators of functions on $\{0, 1\}^N$, we expect overfitting to occur for sufficiently large bond dimension. Indeed, the NLL as a function of bond dimension reported in Figure 2 displays the expected bias-variance tradeoff, with optimal model complexity occurring at bond dimension 3 with corresponding generalization gap $\epsilon = 0.0237$.

The second problem we consider is unsupervised learning of the divisible-by-7 language which consists of the binary representation of integers which are divisible by 7. The dataset was constructed using first 149797 such integers which lie in the range $[1, 2^{20}]$. We trained a length-20 MPS to learn the uniform distribution on the divisible-by-7 language as we did for P_{20} , except utilizing subsets of size 10% for training, testing and cross-validation. Figure 3 illustrates that the model trained on exact single site DMRG with a bond dimension of 8 learns the DIV7 dataset with nearly perfect accuracy, producing a model with a generalization gap of $\epsilon = 0.032$.

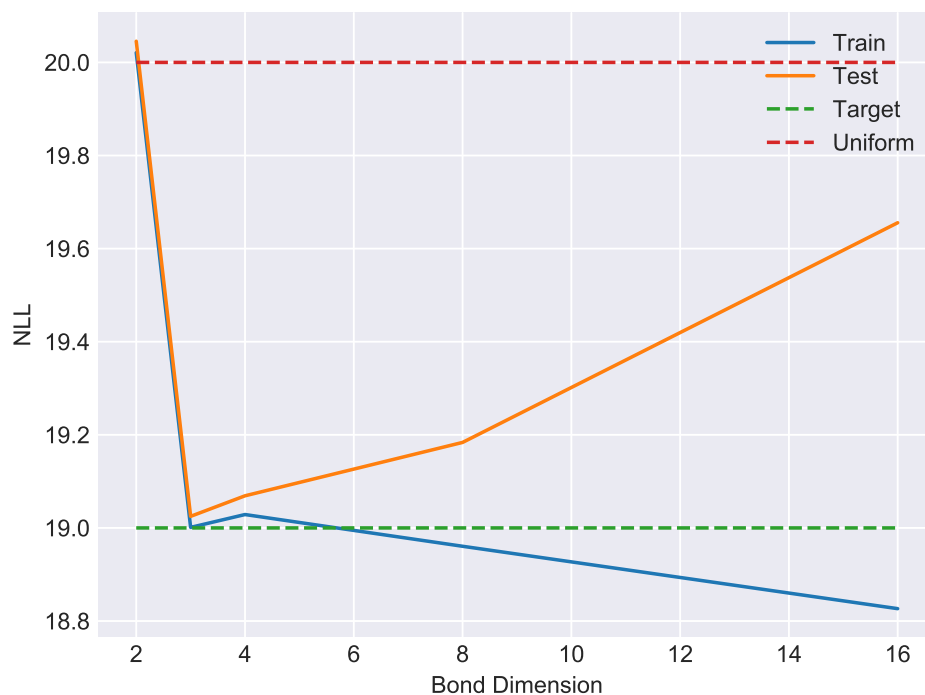


Figure 2. A representative bias-variance tradeoff curve showing negative log-likelihood (base 2) as a function of bond dimension for exact single-site DMRG on the P_{20} dataset. For bond dimension 3, the generalization gap is approximately $\epsilon = 0.0237$. For reference, the uniform distribution on bitstrings has NLL of 20. Memorizing the training data would yield a NLL of approximately 13.356.

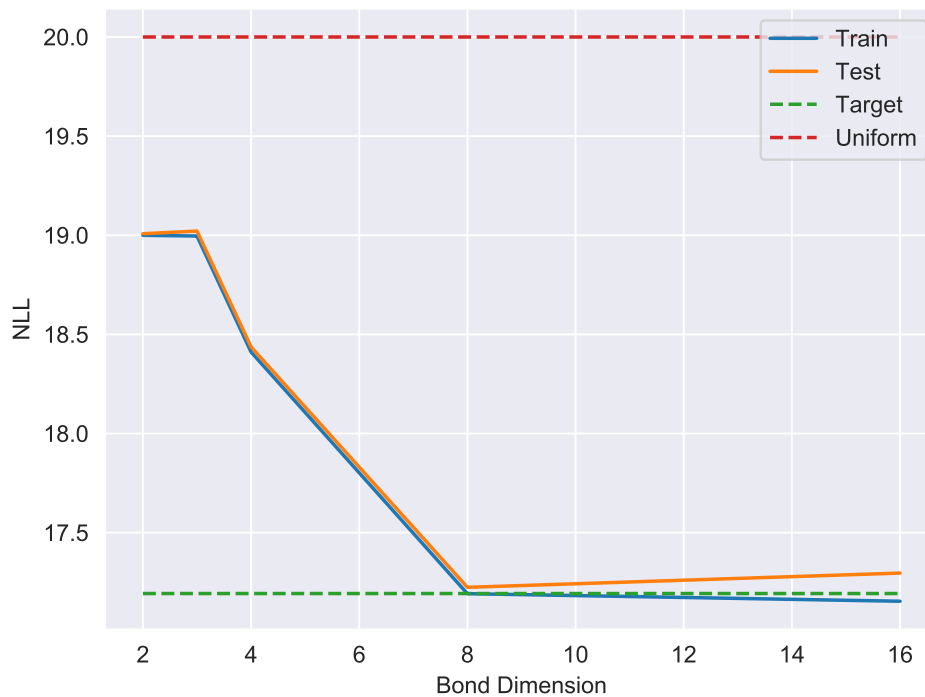


Figure 3. A representative bias-variance tradeoff curve showing negative log-likelihood (base 2) as a function of bond dimension for exact single-site DMRG on the div7 dataset. For bond dimension 8, the generalization gap is approximately $\epsilon = 0.032$. For reference, the uniform distribution on bitstrings has NLL of 20, the target distribution has a NLL of 17.192, and memorizing the training data would yield a NLL of approximately 13.87.

7. Discussion

A number of recent works have explored the parity dataset using restricted Boltzmann machines (RBMs) and found it to be difficult to learn, even in experiments that train using the entire dataset [11,21]. Recall that an RBM is a universal approximator of distributions on $\{0,1\}^N$, given sufficiently many hidden units. Ref. [21] proved that any probability distribution on $\{0,1\}^N$ can be approximated within ϵ in KL-divergence by an RBM with $m \geq 2^{(N-1)(1-\epsilon)+0.1}$ hidden units. For P_{20} this bound works out to be about 4×10^5 hidden nodes. It would be interesting to know whether it could be learned with significantly fewer.

It is not difficult to train a feedforward neural network to classify bitstrings by parity using labelled data, but we do not know if there are unsupervised generative neural models that do well learning P_N . Additionally, quantum circuits can be trained to classify labelled data [15]. It is reasonable to expect that recurrent models whose training involve conditional probabilities $\pi(x_1, \dots, x_k | x_{k+1}, \dots, x_N)$ might be frustrated by P_N since the conditional distributions contain no information: any bitstring of length less than N has the same number of completions in P_N as not in P_N .

The reader may be interested in [22,23] where quantum models are used to learn classical data. Those works considered quantum Boltzmann machines which were shown to learn the distribution more effectively than their classical counterparts using the same dataset. The complexity of classically simulating a QBM scales exponentially with the number of sites in contrast to the tensor network algorithms presented here, which scale linearly in the number of sites (for fixed bond dimension).

The main goal of this paper is to demonstrate the existence of classical datasets for which tensor network models trained via DMRG learn more effectively than generative neural models. It will be interesting to understand better how and why [24].

8. Conclusions and Outlook

The essence of DMRG in the Quantum Physics literature is to solve an eigenvalue problem in a high-dimensional Hilbert space \mathcal{H} by iteratively solving an *effective* eigenvalue problem in an isometrically embedded Hilbert subspace $\mathcal{H}_{\text{eff}} \subseteq \mathcal{H}$. In this paper we have shown how similar reasoning allows to solve a high-dimensional distribution estimation problem by iteratively solving a related linear algebra problem in effective Hilbert space. The proposed algorithm offers a number of advantages over existing gradient-based techniques including a guaranteed improvement theorem, and empirically performs well on tasks for which gradient-based methods are known to fail.

Author Contributions: Formal analysis, J.S. and J.T.; Software, J.S. and J.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Tunnel.

Acknowledgments: The authors thank Tai-Danae Bradley, Giuseppe Carleo, Joseph Hirsh, Maxim Kontsevich, Jackie Shadlen, Miles Stoudenmire, and Yiannis Vlassopoulos for many helpful conversations.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Multi-Site DMRG

For completeness we now describe a related *multi-site DMRG* algorithm. The model class \mathcal{M} now consists of normalized vectors with matrix product factorizations, with possibly different bond spaces having dimension less than a fixed upper bound. The algorithm begins with an initial vector $\psi_0 \in \mathcal{M}$ and produces ψ_1, ψ_2, \dots inductively. The inductive step is similar in that we solve an effective problem in the image of an effective Hilbert space

$$\mathcal{H}_{t+1} := \alpha_{t+1}(\mathcal{H}_{\text{eff},t+1}) \tag{A1}$$

to find the unit vector in \mathcal{H}_{t+1} that is closest to the target state $\psi_{\hat{\pi}}$, which we now denote with a tilde:

$$\tilde{\psi}_{t+1} := \underset{\{\psi \in \mathcal{H}_{t+1}: \|\psi\|=1\}}{\text{arg min}} \|\psi_{\hat{\pi}} - \psi\|. \tag{A2}$$

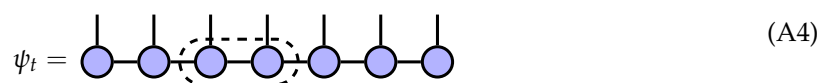
In multi-site DMRG, as opposed to single-site DMRG, the image of the effective space \mathcal{H}_{t+1} is not contained in the MPS model hypothesis class \mathcal{M} . So, the solution $\tilde{\psi}_{t+1}$ to the effective problem must undergo a “model repair” step

$$\tilde{\psi}_{t+1} \rightsquigarrow \psi_{t+1} \tag{A3}$$

to produce a vector $\psi_{t+1} \in \mathcal{M}$. In summary:

- Use ψ_t to define an isometric embedding $\alpha_{t+1} : \mathcal{H}_{\text{eff}} \rightarrow \mathcal{H}$ with $\psi_t \in \mathcal{H}_{t+1} := \alpha_{t+1}(\mathcal{H}_{\text{eff}})$.
- Let $\tilde{\psi}_{t+1}$ be the unit vector in \mathcal{H}_{t+1} closest to $\psi_{\hat{\pi}}$.
- Perform a model repair of $\tilde{\psi}_{t+1}$ to obtain a vector $\psi_{t+1} \in \mathcal{M}$. There are multiple ways to do the model repair.

In order to define the effective problem in the inductive step of multi-site DMRG, one uses an MPS factorization of ψ_t in mixed canonical gauge relative to an interval of r -sites. In the picture below, the interval consists of the two sites 3 and 4.



The effective Hilbert space $\mathcal{H}_{\text{eff}} = W_L \otimes V^{\otimes r} \otimes W_R$ where W_L and W_R are the bond spaces to the left and right of the fixed interval of sites, and r is the length of the chosen interval. The map $\alpha : W_L \otimes V^{\otimes r} \otimes W_R \rightarrow V^{\otimes n}$ is given by replacing the interval of sites and contracting

$$\begin{array}{c} \text{---} \text{---} \text{---} \end{array} \xrightarrow{\alpha} \begin{array}{c} \text{---} \text{---} \text{---} \text{---} \text{---} \end{array} \quad (\text{A5})$$

The map α and its adjoint α^* are described by, and have properties proved by, pictures completely analogous to those detailed for single-site DMRG in Section 5. The effective problem is also solved the same way. What is not the same is that the vector in $\mathcal{H}_{t+1} = \alpha(W_L \otimes V^{\otimes r} \otimes W_R)$ which solves the effective problem is *outside* of the model class \mathcal{M} and so one performs a model repair step $\tilde{\psi}_{t+1} \rightsquigarrow \psi_{t+1}$, pictured graphically in \mathcal{H}_{eff} by:

$$\begin{array}{c} \text{---} \text{---} \end{array} \rightsquigarrow \begin{array}{c} \text{---} \text{---} \end{array} \quad (\text{A6})$$

One way to perform the model repair is to choose

$$\psi_{t+1} := \underset{\psi \in \mathcal{M} \cap \mathcal{H}_{t+1}}{\text{arg min}} \|\psi - \tilde{\psi}_{t+1}\| \quad (\text{A7})$$

but the flexibility of the model repair step allows for other possibilities. One can use the model repair to implement a dynamic tradeoff between proximity to $\tilde{\psi}_{t+1}$ and other constraints of interest, such as bond dimension. Many of these implementations have good algorithms arising from singular value decompositions manageable in the effective Hilbert space. Let us denote such a model repair choice as ψ_{t+1}^{SVD} . Be aware that if ψ_{t+1}^{SVD} is the vector in $\mathcal{M} \cap \mathcal{H}_{t+1}$ nearest to $\tilde{\psi}_{t+1}$ as in Equation (A7), there is no guarantee that ψ_{t+1}^{SVD} will be nearer to $\psi_{\hat{\pi}}$ than the previous iterate. In fact, we have experimentally observed the sequence obtained by this kind of model repair to move away from $\psi_{\hat{\pi}}$. See Figure A1 for an illustration of this possibility.

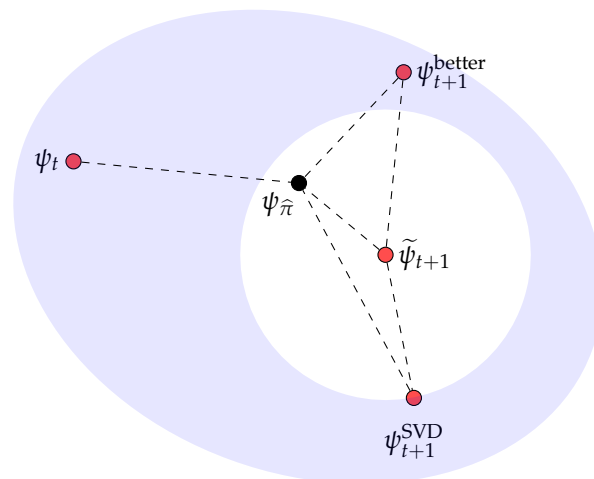


Figure A1. The shaded region represents the model class \mathcal{M} . The red points all lie in \mathcal{H}_{t+1} . The vector $\tilde{\psi}_{t+1}$ is defined to be the unit vector in \mathcal{H}_{t+1} closest to the target $\psi_{\hat{\pi}}$. Note that $\tilde{\psi}_{t+1}$ does not lie in \mathcal{M} . The vector ψ_{t+1}^{SVD} is defined to be the vector in $\mathcal{M} \cap \mathcal{H}_{t+1}$ closest to $\tilde{\psi}_{t+1}$. In this picture, $\|\psi_{t+1}^{\text{SVD}} - \psi_{\hat{\pi}}\| > \|\psi_t - \psi_{\hat{\pi}}\|$. There may be a point, such as the one labelled $\psi_{t+1}^{\text{better}}$, which lies in $\mathcal{M} \cap \mathcal{H}_{t+1}$ and is closer to $\psi_{\hat{\pi}}$ than ψ_{t+1}^{SVD} , notwithstanding the fact that it is further from $\tilde{\psi}_{t+1}$. This figure, to scale, depicts a scenario in which $\|\psi_t - \psi_{\hat{\pi}}\| = 0.09$, $\|\psi_{t+1}^{\text{SVD}} - \psi_{\hat{\pi}}\| = 0.10$, $\|\psi_{t+1}^{\text{better}} - \psi_{\hat{\pi}}\| = 0.07$, $\|\tilde{\psi}_{t+1} - \psi_{\hat{\pi}}\| = 0.06$, $\|\psi_{t+1}^{\text{SVD}} - \tilde{\psi}_{t+1}\| = 0.07$, and $\|\psi_{t+1}^{\text{better}} - \tilde{\psi}_{t+1}\| = 0.08$.

One might hope to improve the model repair step, say by pre-conditioning the singular value decomposition in a way that is knowledgeable about the target $\psi_{\hat{\pi}}$. For the experiments reported in this paper, single-site DMRG consistently outperformed multi-site DMRG for several choices of model repair step, and we include multi-site DMRG only for pedagogical reasons. The adaptability of the bond dimension afforded by the multi-site DMRG algorithm could provide benefits that outweigh the challenges of good model repair in some situations.

References

1. Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N.; Lloyd, S. Quantum machine learning. *Nature* **2017**, *549*, 195. [CrossRef]
2. Preskill, J. Quantum Computing in the NISQ era and beyond. *Quantum* **2018**, *2*, 79. [CrossRef]
3. Peruzzo, A.; McClean, J.; Shadbolt, P.; Yung, M.H.; Zhou, X.Q.; Love, P.J.; Aspuru-Guzik, A.; O'Brien, J.L. A variational eigenvalue solver on a photonic quantum processor. *Nat. Commun.* **2014**, *5*, 4213. [CrossRef]
4. Farhi, E.; Goldstone, J.; Gutmann, S. A quantum approximate optimization algorithm. *arXiv* **2014**, arXiv:1411.4028.
5. Huggins, W.; Patil, P.; Mitchell, B.; Whaley, K.B.; Stoudenmire, E.M. Towards quantum machine learning with tensor networks. *Quantum Sci. Technol.* **2019**, *4*, 024001. [CrossRef]
6. Liu, J.G.; Wang, L. Differentiable learning of quantum circuit Born machines. *Phys. Rev. A* **2018**, *98*, 062324. [CrossRef]
7. Benedetti, M.; Garcia-Pintos, D.; Perdomo, O.; Leyton-Ortega, V.; Nam, Y.; Perdomo-Ortiz, A. A generative modeling approach for benchmarking and training shallow quantum circuits. *npj Quantum Inf.* **2019**, *5*, 45. [CrossRef]
8. Du, Y.; Hsieh, M.H.; Liu, T.; Tao, D. The expressive power of parameterized quantum circuits. *arXiv* **2018**, arXiv:1810.11922.
9. Killoran, N.; Bromley, T.R.; Arrazola, J.M.; Schuld, M.; Quesada, N.; Lloyd, S. Continuous-variable quantum neural networks. *Phys. Rev. Res.* **2019**, *1*, 033063. [CrossRef]
10. Shepherd, D.; Bremner, M.J. Temporally unstructured quantum computation. *Proc. R. Soc. A Math. Phys. Eng. Sci.* **2015**, *465*. [CrossRef]
11. Romero, E.; Mazzanti Castrillejo, F.; Delgado, J.; Buchaca, D. Weighted Contrastive Divergence. *Neural Netw.* **2018**. [CrossRef] [PubMed]
12. Shalev-Shwartz, S.; Shamir, O.; Shammah, S. Failures of Gradient-Based Deep Learning. In Proceedings of the 34th International Conference on Machine Learning (ICML 2017), Sydney, Australia, 6–11 August 2017; pp. 3067–3075.
13. Han, Z.Y.; Wang, J.; Fan, H.; Wang, L.; Zhang, P. Unsupervised generative modeling using matrix product states. *Phys. Rev. X* **2018**, *8*, 031012. [CrossRef]
14. Cheng, S.; Wang, L.; Xiang, T.; Zhang, P. Tree tensor networks for generative modeling. *Phys. Rev. B* **2019**, *99*, 155131. [CrossRef]
15. Farhi, E.; Neven, H. Classification with quantum neural networks on near term processors. *arXiv* **2018**, arXiv:1802.06002.
16. Stoudenmire, E.M. The Tensor Network. 2019. Available online: <https://tensornetwork.org> (accessed on 13 February 2019).
17. Schollwöck, U. The density-matrix renormalization group in the age of matrix product states. *Ann. Phys.* **2011**, *326*, 96–192. [CrossRef]
18. Bridgeman, J.C.; Chubb, C.T. Hand-waving and interpretive dance: An introductory course on tensor networks. *J. Phys. A Math. Theor.* **2017**, *50*, 223001. [CrossRef]
19. Orús, R. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Ann. Phys.* **2014**, *349*, 117–158. [CrossRef]
20. Glasser, I.; Sweke, R.; Pancotti, N.; Eisert, J.; Cirac, J.I. Expressive power of tensor-network factorizations for probabilistic modeling, with applications from hidden Markov models to quantum machine learning. *arXiv* **2019**, arXiv:1907.03741.

21. Montúfar, G.F.; Rauh, J.; Ay, N. Expressive Power and Approximation Errors of Restricted Boltzmann Machines. In Proceedings of the 24th International Conference on Neural Information Processing Systems (NIPS'11), Granada, Spain, 12–15 December 2011; Curran Associates Inc.: Red Hook, NY, USA, 2011; pp. 415–423.
22. Amin, M.H.; Andriyash, E.; Rolfe, J.; Kulchytskyy, B.; Melko, R. Quantum boltzmann machine. *Phys. Rev. X* **2018**, *8*, 021050. [[CrossRef](#)]
23. Kappen, H.J. Learning quantum models from quantum or classical data. *arXiv* **2018**, arXiv:1803.11278.
24. Bradley, T.D.; Stoudenmire, E.M.; Terilla, J. Modeling Sequences with Quantum States: A Look Under the Hood. *arXiv* **2019**, arXiv:1910.07425.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).