Article

Quantum Optical Experiments Modeled by Long Short-Term Memory

Thomas Adler [1,*], **Manuel Erhard** [2,†], **Mario Krenn** [3,4,‡], **Johannes Brandstetter** [1,§], **Johannes Kofler** [1] **and Sepp Hochreiter** [1,5]

1   ELLIS Unit Linz, LIT AI Lab, Institute for Machine Learning, Johannes Kepler University, 4040 Linz, Austria; brandstetter@ml.jku.at (J.B.); kofler@ml.jku.at (J.K.); hochreit@ml.jku.at (S.H.)
2   Institute for Quantum Optics and Quantum Information, Austrian Academy of Sciences & Vienna Center for Quantum Science & Technology, University of Vienna, 1090 Vienna, Austria; manuel.erhard@univie.ac.at
3   Department of Chemistry, University of Toronto & Vector Institute for Artificial Intelligence, Toronto, ON M5G 1M1, Canada; mario.krenn@univie.ac.at
4   Department of Computer Science, University of Toronto & Vector Institute for Artificial Intelligence, Toronto, ON M5G 1M1, Canada
5   Institute of Advanced Research in Artificial Intelligence (IARAI), Landstraßer Hauptstraße 5, 1030 Vienna, Austria
*   Correspondence: adler@ml.jku.at
†   Current address: Quantum Technology Laboratories GmbH, Wohllebengasse 4/4, 1040 Vienna, Austria.
‡   Current address: Max Planck Institute for the Science of Light, 91058 Erlangen, Germany.
§   Current address: Faculty of Science, Informatics Institute, University of Amsterdam, 1090 GH Amsterdam, The Netherlands.

Abstract: We demonstrate how machine learning is able to model experiments in quantum physics. Quantum entanglement is a cornerstone for upcoming quantum technologies, such as quantum computation and quantum cryptography. Of particular interest are complex quantum states with more than two particles and a large number of entangled quantum levels. Given such a multiparticle high-dimensional quantum state, it is usually impossible to reconstruct an experimental setup that produces it. To search for interesting experiments, one thus has to randomly create millions of setups on a computer and calculate the respective output states. In this work, we show that machine learning models can provide significant improvement over random search. We demonstrate that a long short-term memory (LSTM) neural network can successfully learn to model quantum experiments by correctly predicting output state characteristics for given setups without the necessity of computing the states themselves. This approach not only allows for faster search, but is also an essential step towards the automated design of multiparticle high-dimensional quantum experiments using generative machine learning models.

Keywords: quantum optics; multipartite high-dimensional entanglement; supervised machine learning; long short-term memory

1. Introduction

In the past decade, artificial neural networks have been applied to a plethora of scientific disciplines, commercial applications, and every-day tasks with outstanding performance in, e.g., medical diagnosis, self-driving, and board games [1,2]. In contrast to standard feedforward neural networks, long short-term memory (LSTM) [3,4] architectures have recurrent connections, which allow them to process sequential data such as text and speech [5].

Such sequence-processing capabilities can be particularly useful for designing complex quantum experiments, since the final state of quantum particles depends on the sequence of elements, i.e., the experimental setup, these particles pass through. For instance, in quantum optical experiments, photons may traverse a sequence of wave plates, beam

splitters, and holographic plates. High-dimensional quantum states are important for multiparticle and multisetting violations of local realist models, as well as for applications in emerging quantum technologies such as quantum communication and error correction in quantum computers [6,7].

Already for three photons and only a few quantum levels, it becomes, in general, infeasible for humans to determine the required setup for a desired final quantum state, which makes automated design procedures for this inverse problem necessary. With the algorithm MELVIN [8], it was shown how to automate this process. MELVIN uses a toolbox of optical elements, randomly generates sequences of these elements, calculates the resulting quantum state, and then checks whether the state is interesting, i.e., maximally entangled and involving many quantum levels. Furthermore, it has learning capability that speeds up the discoery of more complex systems. The setups proposed by MELVIN have been realized in laboratory experiments [9,10]. Numerous variations have been developed since then. For example, genetic algorithms [11,12] coupled with neural networks [13], reinforcement-learning-based search [14], gradient-descent of a continuous experimental space [15,16] or efficient human-interpretable representations [17] and unsupervised deep generative models [18]. See a recent review about these developments [19].

Inspired by these advances, we investigate how LSTM networks can learn quantum optical setups and predict the characteristics of the resulting quantum states, a task whose level of humanly perceived difficulty distinctly goes beyond that of other deep learning tasks, like object recognition or text generation. We train the neural networks using millions of setups generated by MELVIN. The huge amount of data make deep learning approaches the first choice. We use cluster cross validation [20,21] to evaluate the models.

2. Methods

2.1. Target Values

Let us consider a quantum optical experiment using three photons with orbital angular momentum (OAM) [22,23]. The OAM of a photon is characterized by an integer whose size and sign represent the shape and handedness of the photon wavefront, respectively. For instance, after a series of optical elements, a three particle quantum state may have the following form:

$$|\Psi\rangle = \frac{1}{2}\left(|0,0,0\rangle + |1,0,1\rangle + |2,1,0\rangle + |3,1,1\rangle\right). \tag{1}$$

This state represents a physical situation, in which there is a 1/4 chance (modulus square of the amplitude value 1/2) that all three photons have OAM value 0 (first term), and a 1/4 chance that photons 1 and 3 have OAM value 1, while photon 2 has OAM value 0 (second term), and so on, for the two remaining terms.

We are generally interested in two main characteristics of the quantum states: (1) Are they maximally entangled? (2) Are they high-dimensional? The dimensionality of a state is represented by its Schmidt rank vector (SRV) [24,25]. State $|\Psi\rangle$ is indeed maximally entangled because all terms on the right hand side have the same amplitude value. Its SRV is (4,2,2), as the first photon is four-dimensionally entangled with the other two photons, whereas photons two and three are both only two-dimensionally entangled with the rest.

A setup is labeled "positive" ($y_E = 1$) if its output state is maximally entangled and if the setup obeys some further restrictions, e.g., behaves well under multi-pair emission, and otherwise labeled "negative" ($y_E = 0$). The target label capturing the state dimensionality is the SRV $y_{SRV} = (n, m, k)^\top$. We train LSTM networks to directly predict these state characteristics (entanglement and SRV) from a given experimental setup without actually predicting the quantum state itself.

2.2. Loss Function

For learning classification, we use the binary cross-entropy (BCE) loss function in combination with logistic sigmoid output activation, which dates back to Good [26] and is explained, e.g., by Bishop [27], as the negative log-likelihood of a Bernoulli distribution. For

regression, it is always possible to reorder the photon labels such that the SRV has entries in non-increasing order. An SRV label is thus represented by 3-tuple $y_{\text{SRV}} = (n, m, k)^\top$ which satisfies $n \geq m \geq k$. Hence, we will subsequently refer to the SRV's first entry n as the "leading Schmidt rank".

With slight abuse of notation, we model $n \sim \mathcal{P}(\lambda)$ as a Poisson-distributed random variable and $m \sim \mathcal{B}(n, p), k \sim \mathcal{B}(m, q)$ as Binomials with ranges $m \in \{1, \ldots n\}$ and $k \in \{1, \ldots, m\}$ and success probabilities p and q, respectively. The resulting log-likelihood objective (omitting all terms not depending on λ, p, q) for a data point x with label $(n, m, k)^\top$ is

$$
\begin{aligned}
\ell(\hat{\lambda}, \hat{p}, \hat{q} \mid x) = {} & n \log \hat{\lambda} - \hat{\lambda} + m \log \hat{p} + (n - m) \log(1 - \hat{p}) \\
& + k \log \hat{q} + (m - k) \log(1 - \hat{q})
\end{aligned}
\tag{2}
$$

where $\hat{\lambda}, \hat{p}, \hat{q}$ are the network predictions (i.e., functions of x) for the distribution parameters of n, m, k respectively. The Schmidt rank value predictions are $\hat{n} = \hat{\lambda}, \hat{m} = \hat{p}\hat{\lambda}, \hat{k} = \hat{p}\hat{q}\hat{\lambda}$. To see this, we need to consider the marginals of the joint probability mass function

$$
f(n, m, k) = \frac{\lambda^n e^{-\lambda}}{n!} \binom{n}{m} p^m (1 - p)^{n-m} \binom{m}{k} q^k (1 - q)^{m-k}.
\tag{3}
$$

To obtain the marginal distribution of m, we can first sum over all possible k, which is easy. To sum out n, we first observe that $\binom{n}{m} = 0$ for $n < m$, i.e., the first m terms are zero, and we may write

$$
f(m) = \sum_{n=0}^{\infty} f(n, m) = \sum_{n=0}^{\infty} f(m + n, m)
\tag{4}
$$

capturing only non-zero terms. It follows that

$$
\begin{aligned}
f(m) &= \sum_{n=0}^{\infty} \frac{\lambda^{n+m} e^{-\lambda}}{(n+m)!} \binom{n+m}{m} p^m (1 - p)^n \\
&= e^{-\lambda} p^m \lambda^m \sum_{n=0}^{\infty} \frac{\lambda^n (1 - p)^n}{(n+m)!} \binom{n+m}{m} \\
&= \frac{e^{-\lambda} p^m \lambda^m}{m!} \sum_{n=0}^{\infty} \frac{\lambda^n (1 - p)^n}{n!} = \frac{e^{-p\lambda} (p\lambda)^m}{m!},
\end{aligned}
\tag{5}
$$

which is $\mathcal{P}(p\lambda)$-distributed. Using the same argument for k, we get that the marginal of k is $\mathcal{P}(pq\lambda)$-distributed. The estimates $\hat{n}, \hat{m}, \hat{k}$ are obtained by taking the means of their respective marginals.

2.3. Network Architecture

The used sequence processing model is depicted in Figure 1. We train two networks, one for entanglement classification (target y_{E}), and one for SRV regression (target y_{SRV}). The reason why we avoid multitask learning in this context is that we do not want to incorporate correlations between entanglement and SRV into our models. For instance, the SRV (6,6,6) was only observed in non-maximally entangled samples so far, which is a perfect correlation. This would cause a multitask network to automatically label such a sample as negative only because of its SRV. By training separate networks, we lower the risk of incorporating such correlations.
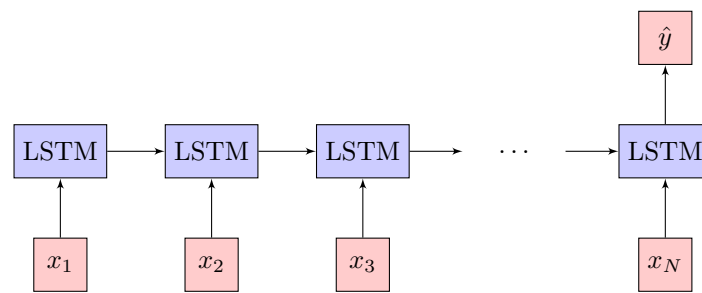
Figure 1. Sequence processing model for a many-to-one mapping. The target value $\hat{y}$ can be either an estimate for y_E (entanglement classification) or y_SRV (SRV regression).

A setup of N elements is being fed into a network by its sequence of individual optical components $x = (x_1, x_2, ..., x_N)^\top$, where, in our data, N ranges from 6 to 15. We use an LSTM with 2048 hidden units and a component embedding space with 64 dimensions. The component embedding technique is similar to word embeddings [28].

3. Experiments

3.1. Dataset

The dataset produced by MELVIN consists of 7,853,853 different setups, of which 1,638,233 samples are labeled positive. Each setup consists of a sequence x of optical elements, and the two target values y_E and y_SRV. We are interested in whether the trained model is able to extrapolate to unseen SRVs. Therefore, we cluster the data by leading Schmidt rank n. Figure 2 shows the the number of positive and negative samples in the data set for each n.
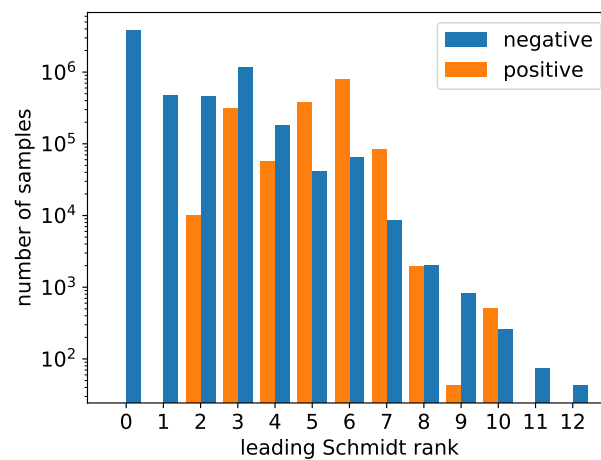

Figure 2. Negative and positive samples in the data set as a function of the leading Schmidt rank n.

3.2. Workflow

All samples with $n \geq 9$ are moved to a special extrapolation set consisting of only 1754 setups (gray cell in Table 1). The remainder of the data, i.e., all samples with $n < 9$, are then divided into a training set and a conventional test set, with 20 % of the data drawn at random (iid). This workflow is shown in Figure 3.

Table 1. Cluster cross validation folds (0–8) and extrapolation set (9–12) characterized by leading Schmidt rank n. Samples with $n = 0$ and samples with $n = 1$ are combined and then split into two folds (0,1) at random.

0,1 0,1	2	3	4	5	6	7	8	9–12

The test set is used to estimate the conventional generalization error, while the extrapolation set is used to shed light on the ability of the learned model to perform on higher Schmidt rank numbers. If the model extrapolates successfully, we can hope to find experimental setups that lead to new interesting quantum states.

Cluster cross validation (CCV) [20,21] is an evaluation method similar to standard cross validation [29]. Cross validation randomly partitions the training data into several folds, and then uses all but one fold for training and the remaining fold for validation. This is done several times such that every fold plays the role of the validation set exactly once. Instead of partitioning the folds iid, CCV groups them according to a clustering method. Thus, CCV removes similarities between training and validation set and simulates situations in which the withheld folds have not been obtained yet, thereby allowing us to investigate the ability of the network to discover these withheld setups. We use CCV with nine folds (white cells in Table 1). Seven of these folds correspond to the leading Schmidt ranks $2, \ldots, 8$. The samples with $n = 1$ (not entangled) and $n = 0$ (not even a valid three-photon state) are negative by definition. These samples represent special cases of $y_E = 0$ setups, and it is not necessary to generalize to these cases without training on them. Therefore, the 4,300,268 samples with $n < 2$ are divided into two folds at random, such that the model will always see some of these special samples while training.
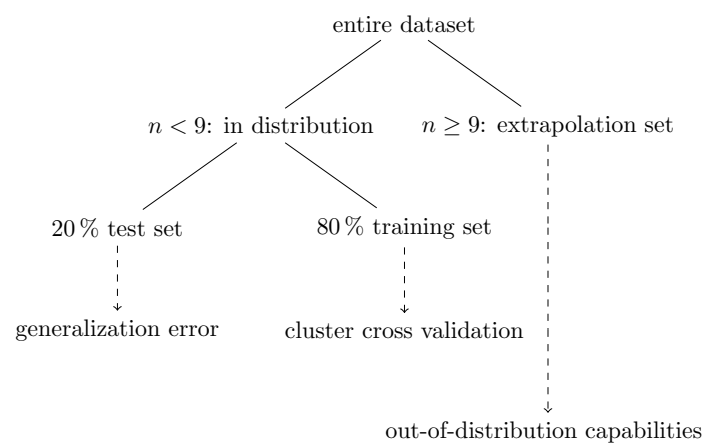

Figure 3. Workflow. We split the entire data by their leading Schmidt rank n. All samples with $n \geq 9$ constitute the extrapolation set, which we use to explore the out-of-distribution capabilities of our model. For the remaining samples (i.e., $n < 9$), we make a random test split at a ratio of $1/4$. The test set is used to estimate the conventional generalization error of our model. We use the training set to perform cluster cross validation.

3.3. Results

Let us examine if the LSTM network has learned something about quantum physics. A good model will identify positive setups correctly, while discarding as many negative setups as possible. This behavior is reflected in the metrics true positive rate $\text{TPR} = \text{TP}/(\text{TP} + \text{FN})$ and *true negative rate* $\text{TNR} = \text{TN}/(\text{TN} + \text{FP})$, with TP, TN, FP, FN the true positives, true negatives, false positives, false negatives, respectively. A metric that quantifies the success rate within the positive predictions is the precision (or positive predictive value), defined as $\text{PPV} = \text{TP}/(\text{TP} + \text{FP})$.

For each withheld CCV fold n, we characterize a setup to be "interesting" when it fulfills the following two criteria: (i) It is classified positive ($\hat{y}_E > \tau$) with τ the classification threshold of the sigmoid output activation. (ii) The SRV prediction $\hat{y}_{\text{SRV}} = (\hat{n}, \hat{m}, \hat{k})^\top$ is such that there exists a $y_{\text{SRV}} = (n, m, k)^\top$ with $\|y_{\text{SRV}} - \hat{y}_{\text{SRV}}\|_2 < r$. We call r the SRV radius. We denote samples which are classified as interesting (uninteresting) and indeed positive (negative) as true positives (negatives). Furthermore, we denote samples which are classified as interesting (uninteresting) and indeed negative (positive) as false positives (false negatives).

We employ stochastic gradient descent for training the LSTM network with momentum 0.5 and batch size 128. We sample mini-batches in such a way that positive and negative samples appear equally often in training. For balanced SRV regression, the leading Schmidt rank vector number n is misused as class label. The models were trained using early stopping after 40,000 weight update steps for the entanglement classification network and 14,000 update steps for the SRV regression network. Hyperparameter search was performed in advance on a data set similar to the training set.

Figure 4 shows the TNR, TPR, and rediscovery ratio for sigmoid threshold $\tau = 0.5$ and SRV radius $r = 3$. The rediscovery ratio is defined as the number of distinct SRVs, for which at least 20% of the samples are rediscovered by our method, i.e., identified as interesting, divided by the number of distinct SRVs in the respective cluster. The TNR for fold (0,1) is 0.9996, and the precision on the extrapolation set 9–12 is 0.659. Error bars in Figure 4 and later in the text are 95 % binomial proportion confidence intervals. Model performance depends heavily on parameters τ and r. Figure 5 shows the "beyond distribution" results for a variety of sigmoid thresholds and SRV radii.
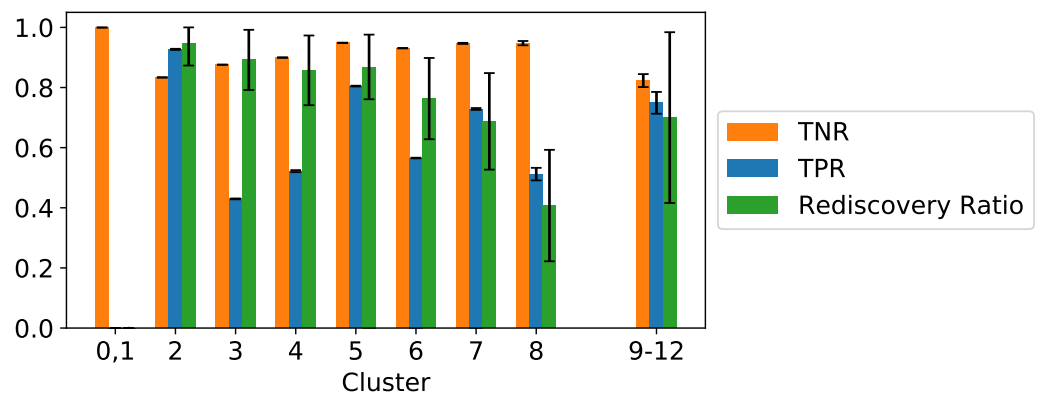

Figure 4. True negative rate (TNR), true positive rate (TPR), rediscovery ratio of the LSTM network using cluster cross validation for different folds 0–8. True negative rates are high for all validation folds. All metrics are good for the extrapolation set 9–12, demonstrating that the models perform well on data beyond the training set distribution, covering only leading Schmidt rank numbers 0–8. Error bars represent 95 % binomial proportion confidence intervals.

Table 2. Conventional in-distribution training and test errors. The test set consists of 20 % of the data. Performance on predicting the entanglement is measured using the BCE loss, TNR, and TPR. Performance on predicting the SRV is measured using the SRV loss according to Equation (2), SRV accuracy, and the mean distance between true SRV and predicted SRV.

	Training	Test
BCE loss	10.2	10.4
TNR	$0.9271 \pm 2.4 \times 10^{-4}$	$0.9261 \pm 3.8 \times 10^{-4}$
TPR	$0.9469 \pm 4.1 \times 10^{-4}$	$0.9427 \pm 6.5 \times 10^{-4}$
SRV loss	2.247	2.24
SRV accuracy	0.9382	0.938
SRV mean distance	1.3943	1.4

Finally, we evaluate the conventional in-distribution performance in Table 2. These figures are consistent with a clean training procedure.
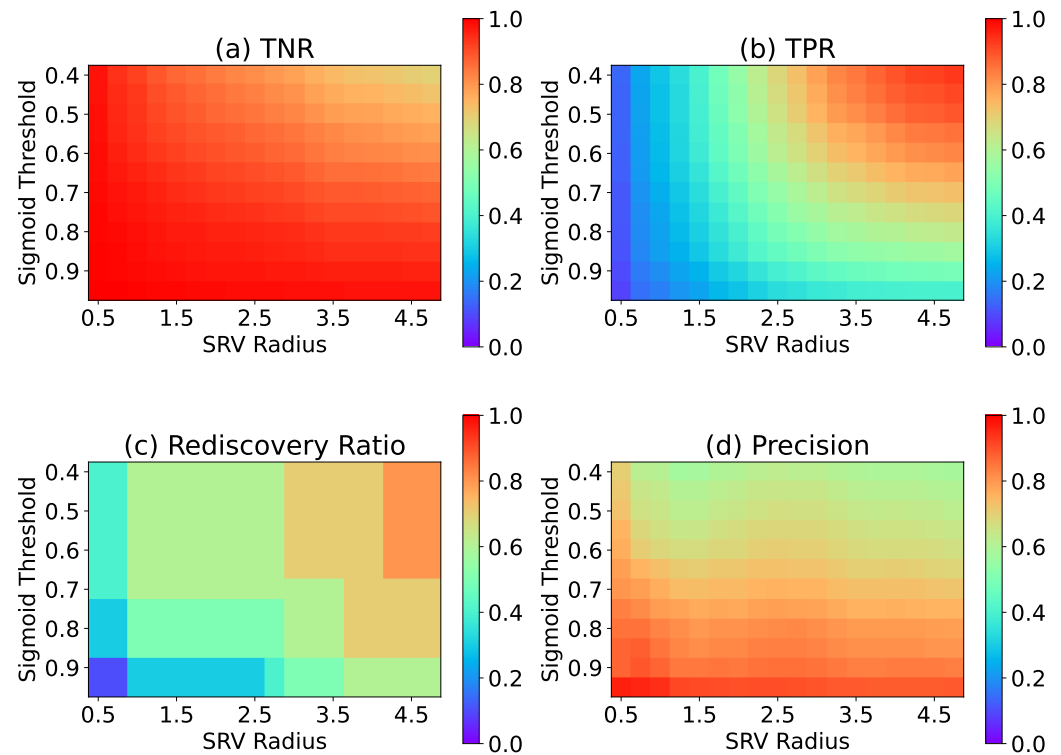
Figure 5. (**a**) True negative rate, (**b**) true positive rate, (**c**) rediscovery ratio, and (**d**) precision for the extrapolation set 9–12 for varying sigmoid threshold τ and SRV radius r. For too restrictive parameter choices ($\tau \to 1$ and $r \to 0.5$), the TNR and precision approach the value 1, while TPR and rediscovery ratio approach 0, such that no interesting new setups would be identified. For too loose choices (small τ, large r), too few negative samples would be rejected, such that the advantage over random search becomes negligible, reflected in smaller precision values. Hence, there is a trade-off between rediscovery ratio (diversity of discoveries) and precision (speed of discoveries). For a large variety of τ and r, the models perform satisfyingly well, allowing a decent compromise between TNR and TPR. This is also reflected by a value of 0.64 for the mean average precision, where the mean is taken over $r = 0.5$ to $r = 7$ with a step size of 0.1, and the average precision is over $\tau = 1$ to $\tau = 0$, with a step size of 0.01 for each value of r.

4. Outlook

Our experiments demonstrate that an LSTM-based neural network can be trained to model certain properties of complex quantum systems. Our approach is not limited to entanglement and Schmidt rank, but may be generalized to employ other objective functions, such as multiparticle transformations, interference and fidelity qualities, and so on.

Another possible next step to expand our approach towards the goal of automated design of multiparticle high-dimensional quantum experiments is the exploitation of generative models. Here, we consider generative adversarial networks (GANs) [30] and beam search [31] as possible approaches.

Generating sequences such as text in adversarial settings has been done using 1D CNNs [32] and LSTMs [33,34]. The LSTM-based approaches employ ideas from reinforcement learning to alleviate the problem of propagating gradients through the softmax outputs of the network. Since our data are, in structure, similar to text, these approaches are directly applicable to our setting.

For beam search, there exist two different ideas, namely a discriminative approach and a generative approach. The discriminative approach incorporates the entire data set (positive and negative samples). The models trained for this work can be used for the discriminative approach, in that one constructs new sequences by maximizing the belief of the network that the outcome will be a positive setup. For the generative approach, the idea

is to train a model on the positive samples, only to learn their distribution via next element prediction. On inference, beam search can be used to approximate the most probable sequence given some initial condition [35]. Another option to generate new sequences is to sample from the softmax distribution of the network output at each sequence position, as has been used for text generation models [36,37].

In general, the automated design procedures of experiments have much broader applications beyond quantum optical setups, and can be of importance for many scientific disciplines other than physics.

5. Conclusions

We have shown that an LSTM-based neural network can be trained to successfully predict certain characteristics of high-dimensional multiparticle quantum states from the experimental setup, without any explicit knowledge of quantum mechanics. For humans, the difficulty of analyzing quantum optical experiments goes far beyond that of other deep learning problems like, e.g., image classification. The network performs well even on unseen data beyond the training distribution, proving its extrapolation capabilities. This paves the way for the automated design of complex quantum experiments using generative machine learning models.

References

1. Esteva, A.; Kuprel, B.; Novoa, R.A.; Ko, J.; Swetter, S.M.; Blau, H.M.; Thrun, S. Dermatologist-level classification of skin cancer with deep neural networks. *Nature* **2017**, *542*, 115–118. [CrossRef]
2. Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. Mastering the game of Go without human knowledge. *Nature* **2017**, *550*, 354–359. [CrossRef]
3. Hochreiter, S. Untersuchungen zu Dynamischen Neuronalen Netzen. Master's Thesis, Technical University of Munich, Munich, Germany, 1991.
4. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
5. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*; Curran Associates, Inc.: San Jose, CA, USA, 2014; pp. 3104–3112.
6. Shor, P.W. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A* **2000**, *52*, R2493-R2496. [CrossRef] [PubMed]
7. Kaszlikowski, D.; Gnacínski, P.; Zukowski, M.; Miklaszewski, W.; Zeilinger, A. Violations of local realism by two entangled N-dimensional systems are stronger than for two qubits. *Phys. Rev. Lett.* **2000**, *86*, 4418–4421. [CrossRef] [PubMed]
8. Krenn, M.; Malik, M.; Fickler, R.; Lapkiewicz, R.; Zeilinger, A. Automated Search for new Quantum Experiments. *Phys. Rev. Lett.* **2016**, *116*, 090405. [CrossRef] [PubMed]

9. Malik, M.; Erhard, M.; Huber, M.; Krenn, M.; Fickler, R.; Zeilinger, A. Multi-photon entanglement in high dimensions. *Nat. Photonics* **2016**, *10*, 248–252. [CrossRef]
10. Erhard, M.; Malik, M.; Krenn, M.; Zeilinger, A. Experimental GHZ entanglement beyond qubits. *Nat. Photonics* **2018**, *12*, 759–764. [CrossRef]
11. Knott, P. A search algorithm for quantum state engineering and metrology. *New J. Phys.* **2016**, *18*, 073033. [CrossRef]
12. Nichols, R.; Mineh, L.; Rubio, J.; Matthews, J.C.; Knott, P.A. Designing quantum experiments with a genetic algorithm. *Quantum Sci. Technol.* **2019**, *4*, 045012. [CrossRef]
13. O'Driscoll, L.; Nichols, R.; Knott, P.A. A hybrid machine learning algorithm for designing quantum experiments. *Quantum Mach. Intell.* **2019**, *1*, 5–15. [CrossRef]
14. Melnikov, A.A.; Poulsen Nautrup, H.; Krenn, M.; Dunjko, V.; Tiersch, M.; Zeilinger, A.; Briegel, H.J. Active learning machine learns to create new quantum experiments. *Proc. Natl. Acad. Sci. USA* **2018**, *115*, 1221–1226. [CrossRef] [PubMed]
15. Arrazola, J.M.; Bromley, T.R.; Izaac, J.; Myers, C.R.; Brádler, K.; Killoran, N. Machine learning method for state preparation and gate synthesis on photonic quantum computers. *Quantum Sci. Technol.* **2019**, *4*, 024004. [CrossRef]
16. Zhan, X.; Wang, K.; Xiao, L.; Bian, Z.; Zhang, Y.; Sanders, B.C.; Zhang, C.; Xue, P. Experimental quantum cloning in a pseudo-unitary system. *Phys. Rev. A* **2020**, *101*, 010302. [CrossRef]
17. Krenn, M.; Kottmann, J.S.; Tischler, N.; Aspuru-Guzik, A. Conceptual understanding through efficient automated design of quantum optical experiments. *Phys. Rev. X* **2021**, *11*, 031044. [CrossRef]
18. Flam-Shepherd, D.; Wu, T.; Gu, X.; Cervera-Lierta, A.; Krenn, M.; Aspuru-Guzik, A. Learning Interpretable Representations of Entanglement in Quantum Optics Experiments using Deep Generative Models. *arXiv* **2021**, arXiv:2109.02490.
19. Krenn, M.; Erhard, M.; Zeilinger, A. Computer-inspired quantum experiments. *Nat. Rev. Phys.* **2020**, *2*, 649–661. [CrossRef]
20. Mayr, A.; Klambauer, G.; Unterthiner, T.; Hochreiter, S. DeepTox: Toxicity Prediction using Deep Learning. *Front. Environ. Sci.* **2016**, *3*. [CrossRef]
21. Mayr, A.; Klambauer, G.; Unterthiner, T.; Steijaert, M.; Wegner, J.K.; Ceulemans, H.; Clevert, D.A.; Hochreiter, S. Large-scale comparison of machine learning methods for drug target prediction on ChEMBL. *Chem. Sci.* **2018**, *9*, 5441–5451. [CrossRef]
22. Yao, A.M.; Padgett, M.J. Orbital angular momentum: Origins, behavior and applications. *Adv. Opt. Photonics* **2011**, *3*, 161–204. [CrossRef]
23. Erhard, M.; Fickler, R.; Krenn, M.; Zeilinger, A. Twisted photons: new quantum perspectives in high dimensions. *Light. Sci. Appl.* **2018**, *7*, 17146. [CrossRef] [PubMed]
24. Huber, M.; de Vicente, J.I. Structure of multidimensional entanglement in multipartite systems. *Phys. Rev. Lett.* **2013**, *110*, 030501. [CrossRef] [PubMed]
25. Huber, M.; Perarnau-Llobet, M.; de Vicente, J.I. Entropy vector formalism and the structure of multidimensional entanglement in multipartite systems. *Phys. Rev. A* **2013**, *88*, 042328. [CrossRef]
26. Good, I.J. Rational Decisions. *J. R. Stat. Soc. Ser. B* **1952**, *14*, 107–114. [CrossRef]
27. Bishop, C.M. *Pattern Recognition and Machine Learning*, 5th ed.; Information Science and Statistics; Springer: Berlin/Heidelberg, Germany, 2007.
28. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
29. Hastie, T.; Tibshirani, R.; Friedman, J.H. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed.; Springer Series in Statistics; Springer: Berlin/Heidelberg, Germany, 2009. [CrossRef]
30. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 27*; Curran Associates, Inc.: San Jose, CA, USA, 2014; pp. 2672–2680.
31. Lowerre, B.T. The Harpy speech recognition system. Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 1976.
32. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A.C. Improved Training of Wasserstein GANs. In *Advances in Neural Information Processing Systems 30*; Curran Associates, Inc.: San Jose, CA, USA, 2017; pp. 5767–5777.
33. Yu, L.; Zhang, W.; Wang, J.; Yu, Y. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. *arXiv* **2016**, arXiv:1609.05473.
34. Fedus, W.; Goodfellow, I.; Dai, A.M. MaskGAN: Better Text Generation via Filling in the_____. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
35. Bengio, S.; Vinyals, O.; Jaitly, N.; Shazeer, N. Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks. In *Advances in Neural Information Processing Systems 28*; Curran Associates, Inc.: San Jose, CA, USA, 2015; pp. 1171–1179.
36. Graves, A. Generating sequences with recurrent neural networks. *arXiv* **2013**, arXiv:1308.0850.
37. Karpathy, A.; Li, F.-F. Deep visual-semantic alignments for generating image descriptions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–15 June 2015; pp. 3128–3137.