USING PHYSICS INFORMED NEURAL OPERATORS FOR SPDC
PROJECT SUBMISSION

Dor Hay Shacham, Nativ Maor, Ben Halperin
Technion Israel Institute of Technology, Department of Computer Science

ABSTRACT

Spontaneous parametric down conversion (SPDC) in quantum optics is an invaluable resource for the realisation of high-dimensional qudits with spatial modes of light. Especially important is the solving of the Inverse SPDC problem, which allows for the generation of a desired qudit state given the initial state of the system. The inverse problem requires solving the wave equations for the propagation of field operators through an interaction medium, the focus of our work will be on this part, aka the "forward problem". Earlier work has introduced a physically constrained and differentiable model, which solves the wave equations via the Split Step Fourier method. Here, we improve upon this by introducing a solver based on a Physically Informed Neural Operator (PINO) Network. By learning over a wide array of pump configurations, we generalise the network to solve unseen pump profiles that may arise in the lab. Training the network to produce physical observables such as coincidence rates and density matrices helps ensure that the solutions we receive are indeed physically sound. As an improvement to the solving of the forward SPDC problem, our work can help further progress in various areas in quantum optics, as well as quantum information processing and cryptography - by introducing a new way of solving the dynamic equations.

1 INTRODUCTION

BACKGROUND

Spontaneous Parametric Down-Conversion (SPDC) is a quantum phenomenon in nonlinear optics that occurs when a single photon of higher energy spontaneously splits into two lower-energy photons. In SPDC, a high-energy (pump) photon interacts with a nonlinear crystal. Due to the nonlinear properties of the crystal, the high-energy photon is transformed into two lower-energy photons through a process known as "down-conversion." The resulting photons are called "signal" and "idler" photons. SPDC has several important applications in quantum optics and quantum information like Quantum Entanglement, Quantum Imaging, Quantum Communication, and more.

As shown in [11] the SPDC process, the first and second-order correlation between the photons and the density matrix (All of which from now on will be referenced as **"The Observables"**) can be simulated by a numeric solution of the equation governor of the dynamic of SPDC.

ISSUES WITH COMPLEXITY

Solving the SPDC dynamics involves solving the mentioned coupled PDE 1, which is nonlinear and can't be solved analytically. A traditional way to solve PDEs of such structure is by a pseudo-spectral numerical method called "split step method".

The split-step method relies on splitting the domain of solution to small steps while also separating the PDE operator into linear and nonlinear parts. Explained briefly, the method approximates a small step of propagation by the nonlinear operator, translate the solution to the Fourier domain with Fourier transform and propagates that solution by another small step dictated by the Linear

operator in the Fourier domain, finally returning to the spatial domain with inverse Fourier transform resulting with an approximation of the solution by that small step.

The method thus involves discretization of the spatial domain and involves back and forth Fourier transforms which can be computationally heavy. The time complexity of such process is $O(m \cdot n \cdot log(n))$ as each Fourier transform is $O(n \cdot log(n))$ using FFT and it is done m times. When n is the number of data elements in a cut perpendicular to the direction of propagation of the field in the crystal and m is the number of slices in the direction of propagation.

ACCELERATION VIA DEEP LEARNING

Our approach to the problem of solving the dynamics relies on learning the solution using a supervised deep learning method based on the FNO/PINO architecture. These architectures have the capability of approximating general operators - specifically in our case the operator which returns the solution of the PDE given the initial conditions and parameters of the problem specified by the boundary conditions as initial conditions and the nonlinear crystal and laser pump as parameters. In this work we have experimented with this method and tried to learn such an operator dictated by a specific simple crystal, learning from many pumps with the goal of it being able to generalise to unseen pumps - with the goal in mind to get a good approximation of the observables and correlations between the solution fields.

The time complexity of the inference part of this approach depends on the depth of the network, the number of layers of each kind and other hyper-parameters. That being said, the costly parts are the Fourier transforms which occurs in the Fourier layers - and their cost is comparable to the the cost of a single iteration of the Split-Step method. As the networks trained typically consists of less Fourier layers than the number of slices of the direction of propagation, we expect solving by our method will be much quicker. than in the split-step method. Our benchmark experiments with the methods did show an enhancement in time, as will be discussed later on in the results section.

CONTRIBUTIONS

- A fast framework for solving SPDC - proved to be faster in the tested cases.
- Extensive experimental validation, done by comparing the network against the classically simulated ground truth, shows decent accuracy for obtaining the objective in solving the SPDC dynamics with the ability of generalization to unseen pumps.

2 PRIOR WORK

As will be further elaborated below, PINO, the main architecture we used, is based on a separate paper [4].
It, in turn, is based on the combination of two separate works: The idea of using existing physical data to enhance the efficiency of neural nonlinear PDE solvers as described by [6]
and the Fourier Neural Operator architecture by [3]. Similar work has been done on a more general scale to allow the solving of differential equations.
[1] present a specialized neural network intended to solve ODEs that can adapt to different ODE types.
Another related and interesting application of neural network for solving complicated systems is the FermiNet described in [5] that approximates the wave function of many electron chemical systems.

3 METHOD

3.1 SPDC

In [11] it is shown that the coupled PDE describing the dynamic of SPDC is given by equations 1

$$
\begin{aligned}
i\frac{\partial E_i^{out}}{\partial \zeta} &= -\frac{\nabla_\perp^2}{2k_i}E_i^{out} + \kappa_i e^{-i\Delta k\zeta}\left(E_s^{vac}\right)^* \\
i\frac{\partial E_i^{vac}}{\partial \zeta} &= -\frac{\nabla_\perp^2}{2k_i}E_i^{vac} + \kappa_i e^{-i\Delta k\zeta}\left(E_s^{out}\right)^* \\
i\frac{\partial E_s^{out}}{\partial \zeta} &= -\frac{\nabla_\perp^2}{2k_s}E_s^{out} + \kappa_s e^{-i\Delta k\zeta}\left(E_i^{vac}\right)^* \\
i\frac{\partial E_s^{vac}}{\partial \zeta} &= -\frac{\nabla_\perp^2}{2k_s}E_s^{vac} + \kappa_s e^{-i\Delta k\zeta}\left(E_i^{out}\right)^*
\end{aligned}
\tag{1}
$$

Where k_j ($j = p, s, i$ for the pump, signal, and idler respectively) is the wave number; $\Delta k = k_p - k_s - k_i$ is the phase mismatch; c is the speed of light; $\zeta = z$ is the coordinate along the propagation direction and $\boldsymbol{r} = (x, y)$ is the position on the transverse plane; $\nabla_\perp^2$ is the transverse Laplacian operator; $\varepsilon_p(\boldsymbol{r})$ is the pump field envelope; $\chi^{(2)}(\boldsymbol{r}, \zeta)$ is the second-order susceptibility; $\kappa_j(\boldsymbol{r}, \zeta) = \frac{\omega_j^2}{c^2 k_j}\chi^{(2)}(\boldsymbol{r}, \zeta)\,\varepsilon_p(\boldsymbol{r})$ is the nonlinear-coupling coefficient; E_j^{out} ($j = i, s$ for the idler and signal) are the output field of the generated photon and E_j^{vac} ($j = i, s$ for the idler and signal) are the vacuum field fluctuations.

It was also shown that by generating Gaussian noise (in order to emulate the physical vacuum field uncertainty) it is possible (using a classical solver) to solve equations 1, and from the solution to simulate the density distribution, the coincidence rate, and the density matrix (i.e. the observables) [11].

In [8] SPDC-inv was introduced - a physically constrained differentiable model that solves the inverse problem; given a quantum state, what is the pump/crystal that will produce this state. This model uses the methods of simulating the observables mentioned earlier.

In our work, we used parts of the SPDC-inv model in order to generate data using the classical solver and simulating the observables.

3.2 PINO

In this work we applied a novel deep learning architecture called Physical Informed Neural Operator (PINO) to generate a time efficient solver for the coupled PDEs describing the process of SPDC 1.

The PINO model is based on an architecture called Fourier Neural Operator (FNO). FNO is a special case of Neural Operators (NO), which is a generalisation of neural networks with the purpose of approximating general operators (maps between infinite dimensional function spaces) by the combination of integral operators and nonlinear activation functions. NO architecture is invariant to discretization of the function spaces mapped, it does so by sharing the model parameters for different discretizations of the underlying function spaces. Similarly to the universal approximation theory for neural networks, there is a universal approximation theory for neural operators [2].

An emergent strength of NO is the capability to learn solutions for families of PDEs, allowing to learn solutions of the PDEs for varying parameters and initial conditions and their discretizations. This contrasts a weakness of the traditional methods of solving PDEs in numerical ways - as they lack the capability to generalise for different initial conditions, parameters, and their discretization and require an intensive computation for each case. Thus a PDE solver based on NO can theoretically be utilized to enable a quick evaluation for various cases, and be cheaper in computation time and resources compared to traditional numerical solvers upon evaluating many cases.

A standard deep neural network is often written as:

$$
u = (K_n \circ \sigma_n \circ K_{n-1} \circ \sigma_{n-1} \circ ... \circ K_1 \circ \sigma_1 \circ K_0)v \tag{2}
$$

When v is the input layer, u is the output layer, K_i are linear or convolution layers and σ_i are non linear activation functions. A NO is similar in structure but the maps $K_i : v_i \mapsto v_{i+1}$ (for $i \in [1, n]$) - general integral operators, are described as

$$
v_{i+1}(x) = \int k_i(x, y)v_i(y)dy + W_i v_i(x) \tag{3}
$$

When k_i are general kernel functions w and W_i are linear transformations. (explanation rephrased from Fourier Neural Operators — neuraloperator 0.2.1 documentation)

FNO is a specific kind of NO that uses convolutions as the global integral operators and utilises Fourier transform properties to its advantage in computing these convolutions. The property used is convolution in the spatial domain being equivalent to point-wise multiplication in the Fourier domain.

The network is built by a projection to higher dimension (by network P) followed by the composition of multiple layers called "Fourier layers", followed by a projection to the target dimension by network Q.

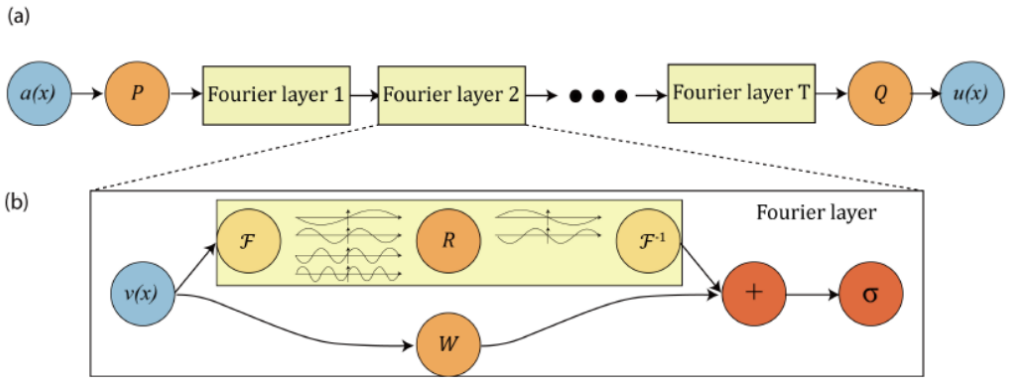


Figure 1: figure and methods taken from [3]

As can be seen in the above figure, the Fourier layer is composed of two parts, Top and Bottom.

Top part: Projecting the output of the previous layer into Fourier basis ($\mathcal{F}$ represents a Fourier transform) and applying a parameterized linear operator on the low modes (R), after that projects back to the original basis by an inverse Fourier transform ($\mathcal{F}^{-1}$ represents an inverse Fourier transform) - effectively doing a general convolution with the function r which is the inverse Fourier transform of R. Bottom part: Local linear transform W.

FNO architecture has shown its capability to successfully model a variety of PDEs including Darcy flow, Navier-Stokes equation and Burgers equation [3].

Physics Informed Neural Operators are based on the FNO architecture, with a twist. While training the network, instead of solely computing the loss function by comparing the output of the network and the ground truth, It uses our awareness of the underlying physics - manifested in knowing the physical properties of the process such as conservation laws and dynamics, thus knowing the governing equations in the form of PDEs. In this case, with the knowledge of the PDEs, PINO calculates the loss by adding a component that encapsulates the agreement of the learned solution to the PDE and the initial condition in the form of Mean Squared Error (MSE). Mathematically the general loss function can be written as:

$$Loss_{total} = w_{FNO} \cdot Loss_{FNO} + w_{PDE} \cdot Loss_{PDE} + w_{I.C} \cdot Loss_{I.C} \tag{4}$$

$Loss_{FNO}$ is the loss of the FNO described by the L_2 distance between the solution $u_{solution}$ and the ground truth u_{grt}:

$$Loss_{FNO} = ||u_{solution} - u_{grt}||_2 \tag{5}$$

$Loss_{I.C}$ is the loss of the Initial conditions described by the L_2 distance between the solution $u_{solution}$ and the ground truth u_{grt} on the boundary of the domain solved marked by " $|_\partial$":

$$Loss_{I.C} = || \ u_{solution}|_\partial - u_{grt}|_\partial \ ||_2 \tag{6}$$

$Loss_{PDE}$ is the loss of the PDE described by the L_2 norm of the $\Lambda(u_{solution})$ when Λ is the PDE operator in its homogeneous form:

$$Loss_{PDE} = ||\Lambda(u_{solution})||_2 \tag{7}$$

The ws are the weights of the respective losses.

It was demonstrated in previous works that PINO architecture is able to compete with state-the-art approximation methods, including the regular FNO, both in accuracy and time, solving multiple PDEs including coupled PDEs such as the nonlinear shallow water equations [7].

In this work we incorporated the idea of PINO architecture being able to approximate the solution of the equations describing the SPDC dynamics, using simulated data from a classical solver as the ground truth (grt), to train a network which will be able to predict the observables given initial conditions of the SPDC problem. While being able to generalise to various discretizations of the fields, different parameters and initial conditions than those which the network was trained on, thus effectively creating a quick and flexible solver.

3.3 Implementation Details

Data

In the training of the SPDC-PINO network, ground truth was established using a classical numerical solution based on the split-step method, detailed in the spdc-inv papers. The data generation process utilized code from the spdc-inv repository [10],[9].

To create the data, we employed equations 1, initializing $E_{i,s}^{out}(\zeta = 0) \equiv 0$ as a boundary condition. The split-step Fourier method was then applied to numerically solve the equations for the four fields at each crystal point. Later the data was rearranged to expose the network to pump and field values at $\zeta = 0$, predicting E_i^{out} and E_s^{out} throughout the crystal. To mimic real-world scenarios, where lower modes of the pump have higher probability, the pump profile used in the generated data were randomly generated linear combinations of low pump modes, from the Laguerre-Gauss profile, i.e.

$$\begin{aligned} \varepsilon_p(\boldsymbol{r}) &= \sum_{n=1}^{M} A_n e^{i\varphi_n} P_n(\boldsymbol{r}) \\ A_n &\in U(0,1], \varphi_n \in U(0,2\pi] \end{aligned} \tag{8}$$

Here, $\varepsilon_p(\boldsymbol{r})$ is the pump profile, A_n is a uniformly distributed amplitude, φ_n is a uniformly distributed phase, $P_n(\boldsymbol{r})$ is the n'th LG pure beam mode, and M is the max mode of LG used for pump construction. The pump beam is re-normalized for consistent power.

Various crystal types were considered during training, with a preference for a fixed, poled crystal. To evaluate network performance, tests focused on common experimental pumps—four low LG pure beam modes ($p = 0, l = 0, 1, 2, 3, 4$), each solved with 500 vacuum states. Notably, although the network trained on a variety of modes, it was not exposed beforehand to the pure pump modes used in the tests.

Measure of distance

In order to evaluate the distance between the predicted and the ground truth observables we used 3 known measures:

1. MSE - Mean Square Error; given by the form:

$$MSE = \frac{1}{N}\sum_i |P_i - G_i|^2 \tag{9}$$

   Where G is the ground truth solution; P is the prediction; i is an indexing of the elements of P and G and N is the total amount of elements.

2. Total Variation distance - A common measure between probability distributions. It is given by:

$$TVD = \frac{1}{2}\sum_i |P_i - G_i| \tag{10}$$

   Where G and P are two probability distributions and i is an indexing of P and G.

3. Trace distance - A common measure between density matrices. It is given by:

$$T(\rho,\sigma) = \frac{1}{2}\|\rho-\sigma\|_1 = \frac{1}{2}Tr\left[\sqrt{(\rho-\sigma)^\dagger(\rho-\sigma)}\right] = \frac{1}{2}\sum_i \sqrt{\lambda_i} \tag{11}$$

Where ρ, σ are density matrices; λ_i are the eigenvalues of $A = (\rho-\sigma)^\dagger(\rho-\sigma)$. Since A is hermitian and PSD λ_i bound to be real and non-negative thus $\sqrt{\lambda_i}$ also bountd to be real and non-negative.

TRAINING

Our goal in this work was to create an efficient data based solver for the SPDC dynamics using the architecture of PINO - a solver whose solution is evaluated based on the comparison between the observables described by the network output and the observables of the ground truth.

The PINO network structure enables the approximation of an arbitrary operator - in our case the operator we try to approximate is getting: the fields at the origin ($\zeta = 0$) of the non-linear crystal - which includes 3 fields - ($\varepsilon_p, E_i^{vac}, E_s^{vac}$) and their spatial grid. The network is trained with the goal of approximating the fields through the crystal described by the SPDC dynamics, and especially at the end of the crystal - (E_i^{out}, E_s^{out}).

The training process is similar to other supervised learning training processes. The output of the network is compared with the ground truth by a loss function and an optimization process is done based upon the result. The loss function of the PINO architecture includes few components: data loss- an MSE based loss between the output and ground truth, PDE loss - a loss which evaluates how closely the output satisfies the PDE describing the dynamics, and IC Loss - similar to PDE loss but specifically to the origin of the crystal at $\zeta = 0$.

The components are linearly added with differing weights between them- the choice of the weights is done as part of the hyper-parameters tuning process.

While training we used a randomly sub-sampled grid of the initial condition and the ground truth solution. Due to the grid-invariance of the PINO model, we were able to train the network on the smaller sub-sampled grid and test it on the full-resolution ground truth. This random sub-sampling process made the training more efficient and prevented over-fitting the data.

We trained the network with various hyper-parameter combinations (ratios between the losses, ADAM optimization parameters like learning rate, grid discretization choices, batch sizes, etc) and different normalizations of the data loss.

As a result of various amount of experiments, exploring the properties of the results and how closely they resemble an expected physical result and how stable the learning process was, we narrowed down the various choices of normalization heuristics and hyper-parameters changed such that the hyper-parameters we tuned were the learning rate and the weight between the PDE loss and data loss.

For the tuning process, we first took the basic configuration of PINO [4], changed the learning rate and the relative weight between the PDE loss to the data loss, and tested the output and the prediction of the observables.

Maybe somethings about the hardware used

4 RESULTS

The network which resulted with the best results was trained on a data set with 180 samples, which contained 18 different random pumps; each pump solved with 10 different random vacuum modes. Each sample was a solution with resolution $121 \times 121 \times 10$ ($X \times Y \times Z$) but was sub-sampled randomly to a resolution of $30 \times 30 \times 10$. during the training, The loss weight to the PDE loss was set to 0 (i.e. the loss was only data loss).

The network was also validated along the training on a dataset of 20 samples and the checkpoint that was taken for the tests was of epoch 505 where the validation loss was lowest. The training loss and the validation loss as a function of the epoch (on a log scale) are presented in figure 2.

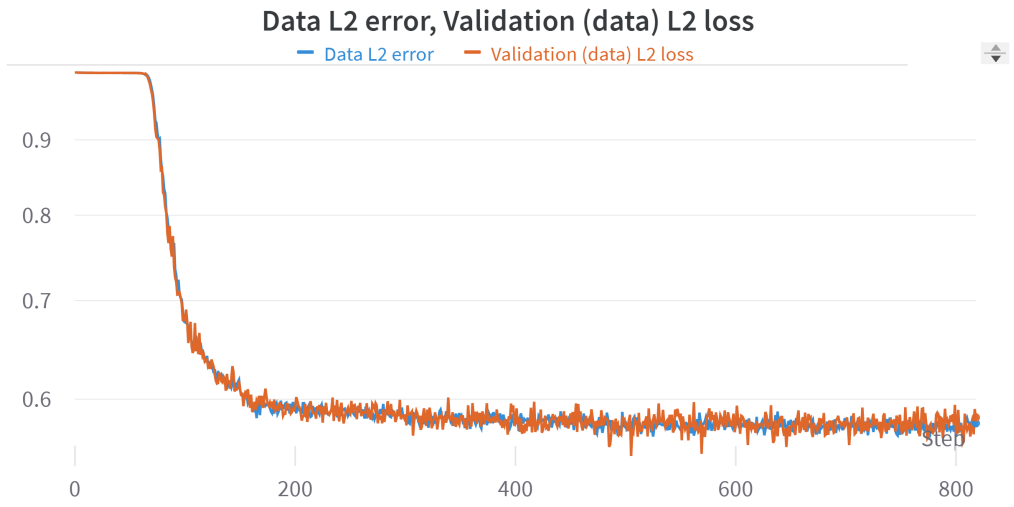


Figure 2: In blue the data loss as a function of the epoch and in orange the validation loss as a function of the epoch (In log scale)

Quantitative Results

The network was tested on predicting the density distribution, coincidence rate, and density matrix of the solution of pumps consisting of the first 4 LG beams (p=0,l=0,1,2,3,4) each solved with 500 vacuum states (modes which the network was not trained on).

In tables 1 and 2 there are quantitative comparisons by different measures of distance (according to formulas 9,10) between the PINO solutions and the split-step solutions (i.e. the ground truth).

	Signal - Density distribution	Idler - Density distribution	Coincidence rate
l=0	0.0328	0.0482	0.0997
l=1	0.0284	0.0310	0.0608
l=2	0.0302	0.0281	0.0764
l=3	0.0294	0.0296	0.1034
l=4	0.0312	0.0326	0.1226

Table 1: The difference between the prediction and the ground truth for the Signal, Idler-density distribution and the coincidence rate with **Total variation distance**

	Signal - Density distribution	Idler - Density distribution	Coincidence rate
l=0	$7.36 \cdot 10^{-11}$	$1.55 \cdot 10^{-10}$	$9.11 \cdot 10^{-5}$
l=1	$3.44 \cdot 10^{-11}$	$3.98 \cdot 10^{-11}$	$2.48 \cdot 10^{-5}$
l=2	$3.23 \cdot 10^{-11}$	$2.75 \cdot 10^{-11}$	$3.98 \cdot 10^{-5}$
l=3	$3.19 \cdot 10^{-11}$	$3.27 \cdot 10^{-11}$	$6.39 \cdot 10^{-5}$
l=4	$4.03 \cdot 10^{-11}$	$4.59 \cdot 10^{-11}$	$1.15 \cdot 10^{-4}$

Table 2: The difference between the prediction and the ground truth for the Signal, Idler-density distribution and the coincidence rate with **MSE**

From the results, it arises that the prediction of the density distribution is the most accurate from the observables, with a MSE difference of less than $2 \cdot 10^{-10}$ for the signal and the idler, or a difference

of less than 5% in the total variation distance. The coincidence rate MSE difference is in order of magnitude greater yet the total variation distance for all the predictions is lower than 12%.

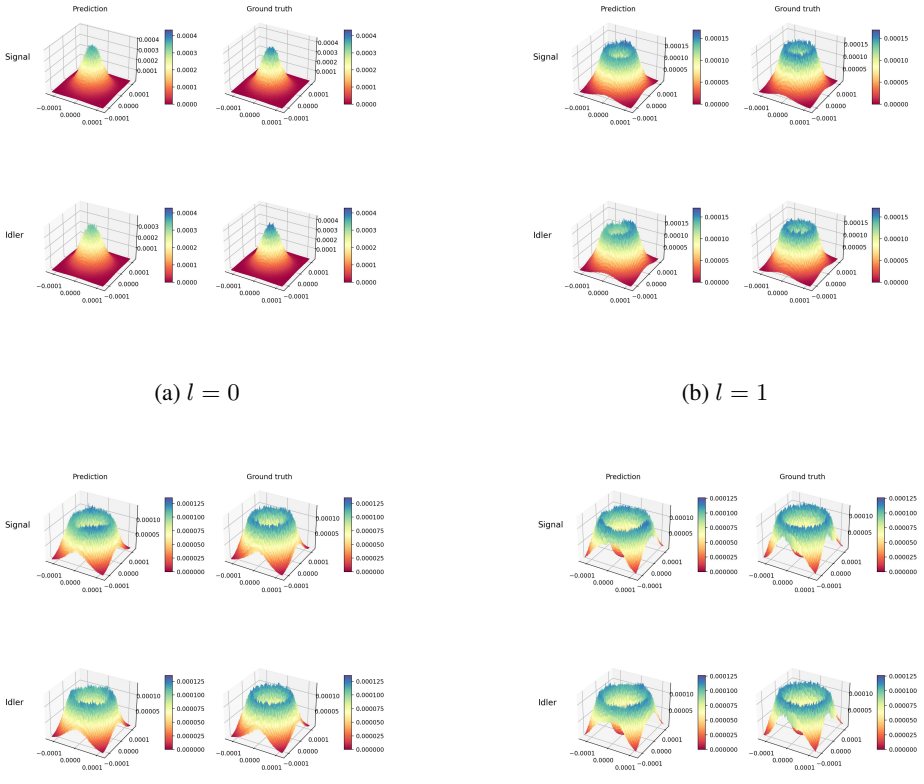
EFFICIENCY TEST

In order to test the difference in efficiency of the two methods, since the network was not written to utilize parallel computing power, as a proof of concept the parallelism of the classical solver was disabled and the test had been solving 1000 samples sequentially. The two solvers were run on a GPU 'NVIDIA RTX A4000' with 16 GB RAM.

The classical solver finished in 368.86957 seconds, on average it took 0.36887 seconds for a sample. The PINO solver finished in 94.75268 seconds, on average it took 0.09475 seconds for a sample, almost a quarter of the classical solver time.

QUALITATIVE RESULTS

Figures 3 depicts the density distribution of the Signal and Idler for l=0,1,2,3

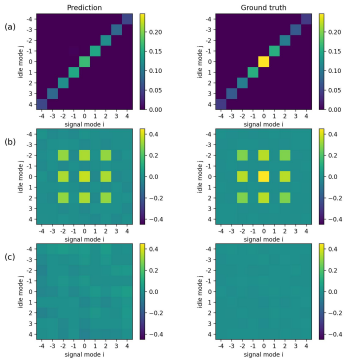


(a) $l = 0$ (b) $l = 1$

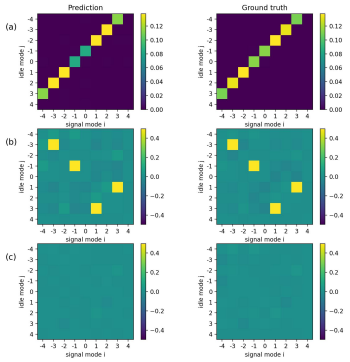
(c) $l = 2$ (d) $l = 3$

Figure 3: The prediction and the ground truth solutions of the Signal and Idler density distribution for a pump profile of p=0, l=0,1,2,3

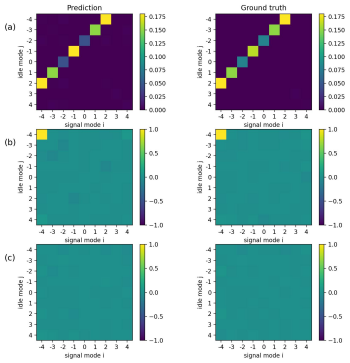
Figure 4 the prediction and the ground truth of the observables are presented for l=0,1,2,3.



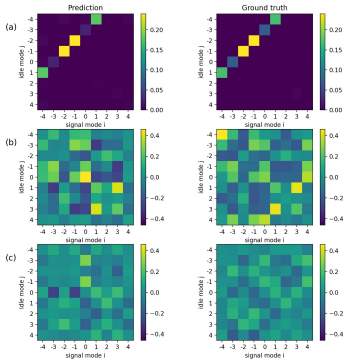
(a) $l = 0$



(b) $l = 1$



(c) $l = 2$



(d) $l = 3$

Figure 4: The prediction and the ground truth of the observables: (a) Coincidence rate, (b) real part of the density matrix, and (c) Imaginary part of the density matrix for a pump profile of p=0, l=0,1,2,3

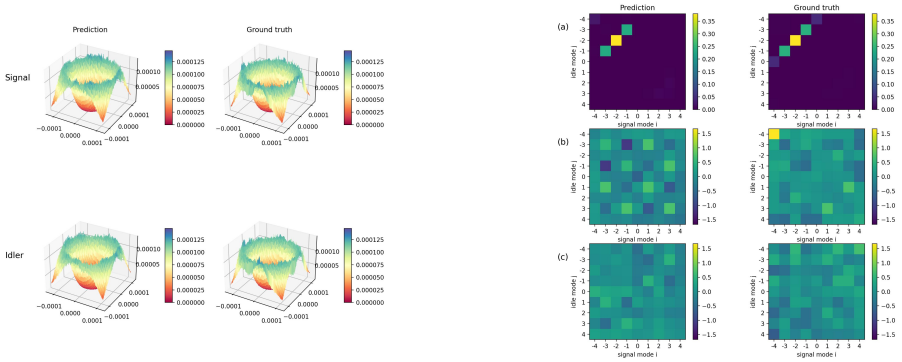
Failure Modes

As seen from the results predicting the higher modes proved to be a more difficult task, for example for l=4 the results are presented in figure 5.

Another unsuccessful direction was incorporating the PDE loss in the train. An example of such a case is presented in Figure 6. It seems that there is no process of minimization during the training which might indicate a further need of tuning the weights loss.

5 Conclusions

Our main objective was solving the wave equations for the propagation of field operators through a crystal in solving the forward SPDC problem in a faster way than with the Split Step method, without sacrificing accuracy or generality in a critical manner. Comparing our results and those from the Split Step method, we have managed to achieve, for the more probable cases, a good approximation of the observables while using only about a quarter of the time. Being an improvement, our work might also be able to improve the solving of related problems with similar hamiltonians, such as superfluids and superconductors. It is important to note that this is still a work in progress. While we had success in generalizing to pumps of lower modes, we had some difficulty in predicting pumps of higher modes. While they are less probable, this is still an issue that should be solved. We have currently trained the model on a fixed crystal. An interesting future lead would be to train the model



(a) (b)

Figure 5: On the left are the predictions and the ground truth solutions of the Signal and Idler density distribution. On the right the prediction and the ground truth of the observables: (a) Coincidence rate, (b) real part of the density matrix, and (c) Imaginary part of the density matrix for a pump profile of p=0, l=4

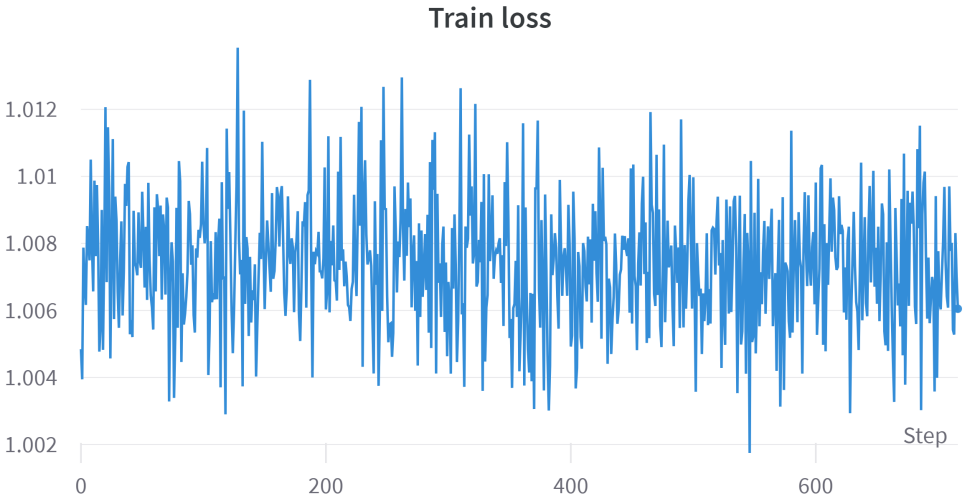


Figure 6: The total (Data + PDE) loss as a function of the epoch

such that it could generalize to different crystals, as it does with pumps. Another interesting venue for additional work is the generalization of the training process as a whole, in order to create a deep learning framework capable of solving different types of equations.

REFERENCES

[1] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[2] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces. *arXiv preprint arXiv:2108.08481*, 2021.

[3] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differ-

ential equations. *arXiv preprint arXiv:2010.08895*, 2020.

[4] Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. *arXiv preprint arXiv:2111.03794*, 2021.

[5] David Pfau, James S Spencer, Alexander GDG Matthews, and W Matthew C Foulkes. Ab initio solution of the many-electron schrödinger equation with deep neural networks. *Physical Review Research*, 2(3):033429, 2020.

[6] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*, 2017.

[7] Shawn G Rosofsky, Hani Al Majed, and EA Huerta. Applications of physics informed neural operators. *Machine Learning: Science and Technology*, 4(2):025022, 2023.

[8] Eyal Rozenberg, Aviv Karnieli, Ofir Yesharim, Joshua Foley-Comer, Sivan Trajtenberg-Mills, Daniel Freedman, Alex M Bronstein, and Ady Arie. Spdcinv: Inverse quantum-optical design of high-dimensional qudits. *arXiv preprint arXiv:2112.05934*, 2021.

[9] Eyal Rozenberg, Aviv Karnieli, Ofir Yesharim, Joshua Foley-Comer, Sivan Trajtenberg-Mills, Daniel Freedman, Alex M. Bronstein, and Ady Arie. Inverse design of spontaneous parametric downconversion for generation of high-dimensional qudits. *Optica*, 9(6):602–615, Jun 2022.

[10] Eyal Rozenberg, Aviv Karnieli, Ofir Yesharim, Sivan Trajtenberg-Mills, Daniel Freedman, Alex M. Bronstein, and Ady Arie. Inverse design of quantum holograms in three-dimensional nonlinear photonic crystals. In *Conference on Lasers and Electro-Optics*, page FM1N.7. Optica Publishing Group, 2021.

[11] Sivan Trajtenberg-Mills, Aviv Karnieli, Noa Voloch-Bloch, Eli Megidish, Hagai S Eisenberg, and Ady Arie. Simulating correlations of structured spontaneously down-converted photon pairs. *Laser & Photonics Reviews*, 14(3):1900321, 2020.