An Artificial Neural Network Approach to the Analytic Continuation Problem

Romain Fournier,[1] Lei Wang,[2] Oleg V. Yazyev,[1, *] and QuanSheng Wu[1, †]

[1]*Institute of Physics, Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland*
[2]*Institute of Physics, Chinese Academy of Sciences, Beijing 100190, China*
(Dated: October 3, 2018)

Inverse problems are encountered in many domains of physics, with analytic continuation of the imaginary Green's function into the real frequency domain being a particularly important example. However, the analytic continuation problem is ill-defined and currently no analytic transformation for solving it is known. We present a general framework for building an artificial neural network that efficiently solves this task with a supervised learning approach, provided that the forward problem is sufficiently stable for creating a large training dataset of physically relevant examples. By comparing with the commonly used maximum entropy approach, we show that our method can reach the same level of accuracy for low-noise input data, while performing significantly better when the noise strength increases. We also demonstrate that adding an unsupervised denoising step significantly improves the accuracy of the maximum entropy predictions for noisy inputs. The computational cost of the proposed neural network appoach is reduced by almost three orders of magnitude compared to the maximum entropy method.

Numerical simulations are playing an extensive role in a growing number of scientific disciplines. Most commonly, numerical simulations approximate a process or a field on a discretized map, taking as input an equation describing the model as well as initial and boundary conditions [1]. Problems falling under this definition are known as direct or forward problems. However, in a number of situations it is required to reconstruct an approximation of the input data or the model that generated it given the observable data. One particularly important example, known as the Fredholm integral equation of the first kind, takes the following form:

$$g(s,t) = k \circ f := \int_a^b k(t,s)f(s)ds, \qquad (1)$$

where $g(s,t)$ is the available quantity, $f(s)$ is the quantity of interest and $k(t,s)$ is the kernel. Such problem definitions are known as inverse problems and are mostly ill-posed [2]. Noise affecting the data may lead to arbitrarily large errors in the solutions. Formally speaking, one is interested in operators $k\circ$ that are ill-conditioned or degenerated. Therefore, the formal inversion $f = k^{-1} \circ g$ is not a stable operation [1]. One popular approach for solving such problems is to construct regularizing algorithms that converge to a "pseudo-solution" [3]. In this work, we will focus on a particularly important case of the analytic continuation problem in quantum many-body physics. The analytic continuation problem seeks to recover the electron single-particle spectral density in the real frequency domain $A(\omega)$ from the fermionic Green's function $G(\tau)$ in imaginary time domain. These quantities are related by the following equation:

$$G(\tau) = - \int \frac{e^{-\omega\tau}}{1 + e^{-\hbar\omega\beta}} A(\omega)d\omega. \qquad (2)$$

High frequencies of $A(\omega)$ are exponentially damped and their contribution to $G(\tau)$ is negligible. Therefore, very different spectral density functions can correspond to very similar Green's functions due to the ill-defined character of the problem. The analytic continuation is particularly important in quantum Monte Carlo methods that seek to compute $G(\tau)$, and whose results need to be compared to the observed $A(\omega)$ [4].

There exist several techniques for performing the analytic continuation (see Refs. 5–10). Among them, the so-called maximum entropy (MaxEnt) approach is the most commonly used. This method initiates by generating random distributions $A^i(\omega)$ and computes their corresponding $G^i(\tau)$ using Eq. 2. $G(\tau)$ and $G^i(\tau)$ are then used to minimize $Q = \frac{1}{2}\chi^2 - \alpha S$, where $\chi^2[A^i] = \sum_{m,n}(G^i(\tau_m) - G(\tau_n))^2\sqrt{C_{mn}^{-1}}$, with C_{mn} being the covariance matrix, S an entropy term that regularizes the problem by preventing the overfitting of the data, and α is a temperature-like parameter that weights the relative importance of χ^2 over S. This last parameter is usually fixed by inferring the probability distribution $P(\alpha|G)$ using Bayes' rule in a first step, and then by either fixing α at the value that maximizes this distribution or by averaging different predictions with the weights provided by the distribution [11]. There is no *a priori* way to know which of these two approaches will give the best results in a given case.

The use of statistical methods for solving Fredholm integral equations has emerged recently. Those include the regularization of the problem and the acceleration of existing methods through dimensionality reduction of the Green's functions [12, 13]. Dahm and Keller pointed out that a similar context of the Fredholm inverse integral equation of the 2nd kind maps onto a reinforcement learning framework [14]. More recently, several works have shown that a machine learning approach is suitable for solving inverse problems [15–17]. Among them, Arsenault *et al.* directly applied the kernel ridge regressions to the analytic continuation problem. In their work,

a dataset of physically relevant pairs of spectral density and Green's functions was generated and the kernel-ridge regression was used to train a model that approximates inversion of Eq. 2. The predictions were then projected onto a basis of functions that satisfies the problem constraints to ensure the physical relevance of the result. This data-driven approach distills the prior knowledge into the simulated training data, instead of placing it as prior knowledge in a default spectrum as in the traditional MaxEnt method. This allows more flexibility in the regularization of the dataset.

In this Letter, we propose an artificial neural network (ANN) approach to solving the analytic continuation problem. We show that the conventional regularization techniques of deep learning can efficiently solve inverse problems. In particular, a final softmax layer ensures the positivity and normalization of the spectrum function $\hat{A}(\omega)$. In that sense, our framework directly integrates the projection step employed in Ref. 17. We show that by applying standard training techniques, we are able to reach accuracy comparable with the MaxEnt approach while keeping lower standard deviations in the error distribution. Furthermore, our approach is more robust against input data that contains noise. We also show that the use of an unsupervised denoising preprocessing step greatly improves the robustness of the MaxEnt method against noisy inputs. Finally, we demonstrate that a trained neural network solves the analytic continuation problem at a much lower computational cost.

By their nature, supervised learning algorithms require a training dataset. Data can be collected from experimental measurements or simulated according to a theoretical model. In our work, we use the latter approach. We stress here that, apart the final softmax layer that ensures the output to have the shape of a density function, the choice of the training data is the main source of regularization of the problem. In this work, we choose to simulate spectral density functions that always has a quasiparticle peak close to $\omega = 0$ as often encountered when considering correlated metals [18, 19]. As in Ref. 17, the model spectral densities $A(\omega)$ are defined as a sum of uncorrelated Gaussian distributions:

$$A_j(\omega) = \frac{1}{N_R} \sum_{i=0}^{R_j} \exp\left(-\frac{(\omega - \mu_i)^2}{2\sigma_i^2}\right), \qquad (3)$$

where the frequencies $\omega \in [-\Omega_0, \Omega_0]$, the maximum frequency $\Omega_0 = 15$ eV, the centers of the peaks $\mu_j \in [-6, 6]$ eV, the number of Gaussian distributions $R_j \in 1, 2, ..., 21$ and their broadening $\sigma_j \in [0.1, 1]$ eV. The prefactor $1/N_R$ is used to normalize the spectrum. Parameters R_j, μ_j and σ_j are uniformly sampled over the above-mentioned ranges, with the exception of μ_0 that is constrained to be located close to the origin ($\mu_0 \in [-0.5, 0.5]$ eV). These spectral functions are then used to compute their corresponding Green's functions according to Eq. 2. A representative example of a $(G(\tau), A(\omega))$ pair is given in Fig. 1a,b. This approach can be extended
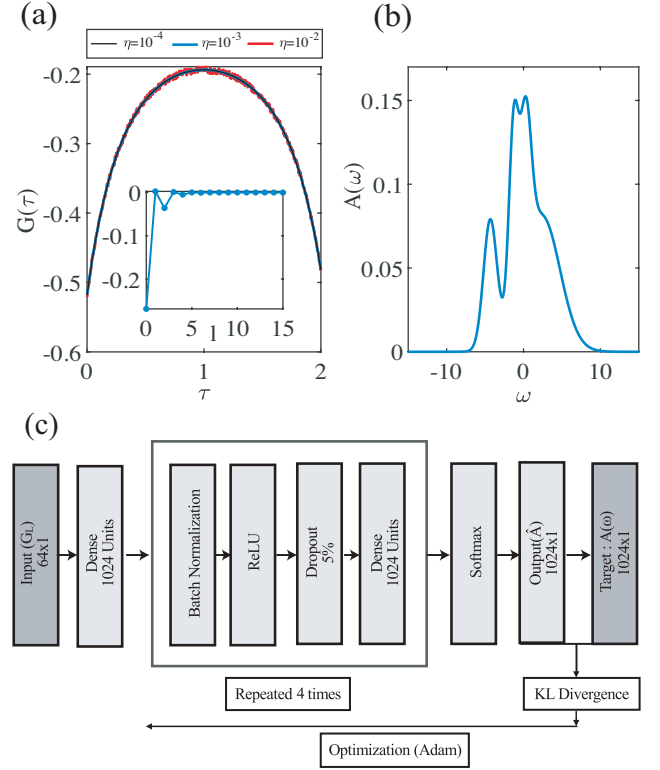

FIG. 1. (a) Example of Green's function $G(\tau)$ containing noise of three different amplitudes η and (b) its corresponding spectral density function $A(\omega)$. The Legendre representation G_l of the Green's function (without noise) is shown in panel (a) inset. Only the first 16 coefficients are displayed for clarity. (c) Schematic diagram of the architecture of the proposed ANN model.

to other Green's function and self-energy problems by adapting Eq. 3 [20].

Working in high-dimensional spaces leads to numerous problems commonly referred to as the "curse of dimensionality" [21]. In the absence of simplifying assumptions, the amount of data necessary to approximate a function to a given accuracy grows exponentially with the number of dimensions [22]. This is due to the fact that for a given amount of data, the parameter space becomes more sparse when the dimensionality increases, which is also reflected in the term "empty space phenomenon". Therefore, finding a more compact representation of the data facilitates the learning process of the model. In previous works, the orthogonal basis of Legendre polynomials was successfully used to improve results involving Green's functions [13, 23]. Applying this idea, we compute the inputs G_l from $G(\tau)$ using the following definition:

$$G(\tau) = \sum_{l \geq 0} (2l + 1) G_l P_l\left(2\frac{\tau}{\beta} - 1\right), \qquad (4)$$

where $P_l(x)$ are the Legendre polynomials. The truncation of polynomial expansions of a function can give rise

to the Gibbs oscillations [24, 25]. In our case, the norm of the coefficients decreases rapidly, therefore only the first 64 coefficients of the expansion are retained (Fig. 1b, inset).

We randomly generate one million $(G(\tau), A(\omega))$ pairs using this procedure and split them into a test set including 500 pairs [26] and a training set containing the rest of data. Since in practice we deal with quantum Monte Carlo data that is noisy by nature, we also generated test sets in which the Green's functions were corrupted by errors ϵ_i [Fig. 1(a)] :

$$\tilde{G}(\tau_i) = G(\tau_i) + \epsilon_i \tag{5}$$

where ϵ_i are independent random variables normally distributed with standard deviation η.

The predicted spectral density functions $\hat{A}(\omega)$ are compared with the ones present in the dataset $A(\omega)$ using two different measures. Namely, we employ the mean absolute error (MAE) as the most intuitive way to assess the difference

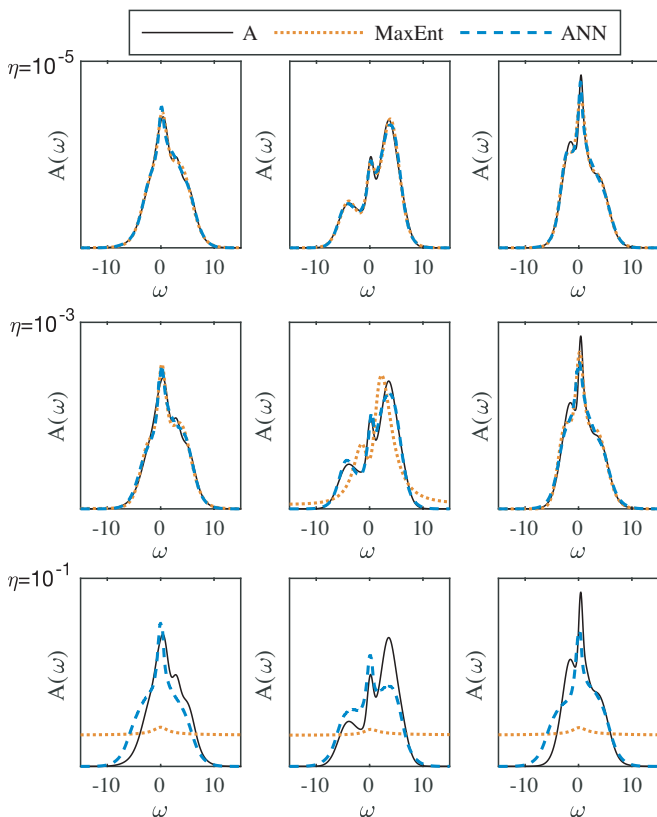$$\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |A(\omega_i) - \hat{A}(\omega_i)| \tag{6}$$

FIG. 2. Comparison of the starting spectral functions $A(\omega)$ (solid lines) with the predicted $\hat{A}(\omega)$ calculated using the MaxEnt approach and the proposed ANN model (dashed lines). Different noise levels η are applied to the Green's functions of the three examples of spectral density functions.

and the Kullback-Leibler divergence (KLD) [27]

$$\text{KLD} = \sum_{i=1}^{N} A(\omega_i) \ln \left[\frac{A(\omega_i)}{\hat{A}(\omega_i)} \right], \tag{7}$$

which results in a better convergence of the model. Our aim is to build an ANN that is able to yield accurate predictions on unseen data. In deep learning terms, we speak of having sufficient representative and generalization capacity. The training of our model is improved until it performs better on average than the MaxEnt method on test data containing a low noise level of $\eta = 10^{-5}$. This ensures that ANN performs well on real data. In our simulations MaxEnt reaches an average MAE of 0.0038 for such data.

The step-by-step design of our ANN starting from the simplest 2-layer model is briefly described in the Supplementary Material document [26]. The ultimate architecture of our model, shown in Fig. 1(c), consists of an input layer connected to a dense layer, followed by four repetitions of a sequence of layers. The first layer in this sequence is a batch normalization layer that improves the training by normalizing the data [28]. This layer is followed by a linear rectifier unit (ReLU), which adds non-linearities into the model by taking the maximum value between 0 and each unit. Then follows a dropout unit that helps to avoid overfitting issues by randomly dropping connections between units during the training process [29]. Finally, a fully connected dense layer containing 1024 units learns the relation between the input and the output. After four repetitions of this sequence, the output is computed using a Softmax layer[30], which ensures its similarity to a probability density function. The training is performed on a dataset of one million samples using the Adam algorithm [31] as an optimizer and the KLD as a loss function. We reduce the learning rate, that is the speed at which the parameters are updated, each time the validation loss reaches a plateau during the training. Details concerning the training process can be found in the Supplementary Material [26].

We have also investigated two approaches to improving the robustness of our results against noisy input data. The first approach relies on preprocessing data using unsupervised learning. Inspired by denoising algorithms in image recognition, we adopted the principal component analysis (PCA) approach for this purpose [32, 33]. As a first step, we compute the principal components of the Legendre coefficients present in our database to form a new basis whose vectors are aligned along the direction of maximum variance in the dataset. The noisy samples are then expressed in this basis, and all but the first n components are set to zero. In our case, $n = 3$ components are retained, which corresponds to directions covering 99.8% of the variance of the training dataset. The data is then transformed back to the original space. This filtering approach keeps the important features of the sample while reducing the noise it contains. Examples of the filtered data can be found in the Supplementary Material docu-
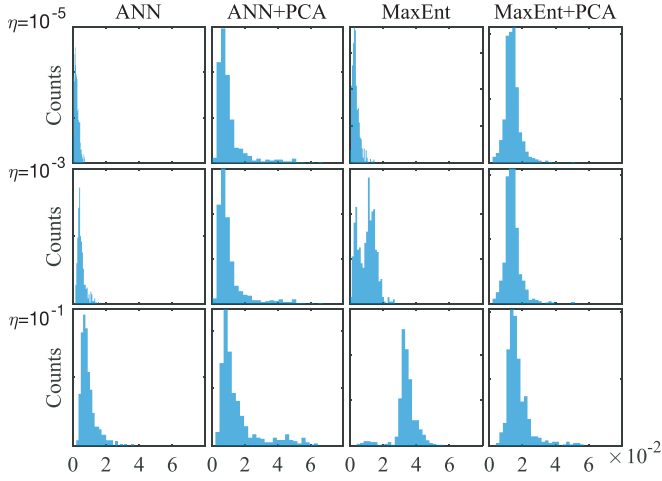
FIG. 3. Distributions of the MAE for the different methods and for increasing levels of noise η. Adding a PCA step makes the model more robust with respect to noise.

ment [26]. The second method of improving the robustness of the results only works for ANN and consists in adding a gaussian noise layer at the beginning of the network during the training phase. This layer adds random gaussian noise to the input data. Networks trained using this extra layer typically achieve better results than MaxEnt for levels of noise larger than $\eta = 10^{-4}$ by learning only from 25000 input data realizations.
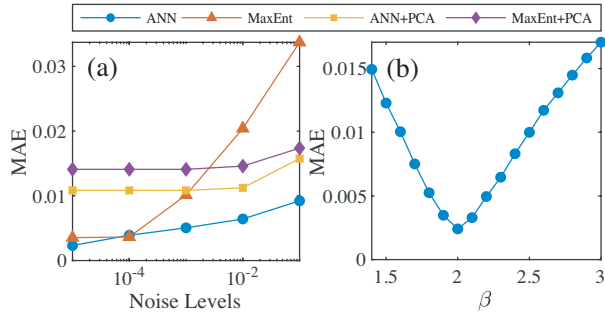

FIG. 4. (a) Comparison of the robustness against increasing levels of noise for the MaxEnt approach, the proposed ANN model and both models applied to three principal components of the Green's function. (b) MAE for the data generated at different temperatures β and predicted by a model trained for $\beta = 2$.

Figure 2 provides a qualitative comparison of the results of our ANN and the MaxEnt implementation of Levy et $al.$ [20] for three different noise levels, $\eta = 10^{-5}$, 10^{-3} and 10^{-1}, generated following the procedure described above. The level of noise was provided as parameter for MaxEnt and used to select the network for ANN, as explained above. In these examples, both methods predict $A(\omega)$ accurately for the lowest level of noise. However, at $\eta = 10^{-3}$ MaxEnt tends to suppress some peaks in the predicted spectral function $\hat{A}(\omega)$, while in

the case of ANN this tendency is much less pronounced. At the highest level of noise $\eta = 10^{-1}$, our ANN is able to correctly identify most peaks, whereas MaxEnt flattens the distribution (Fig. 2). Our ANN also showed better results compared to MaxEnt on the example of Legendre data from Ref. 20.

We will now analyze the errors committed by the discussed models and the impact of the PCA step for increasing levels of noise ($\eta = 10^{-5}$, 10^{-3} and 10^{-1}). Fig. 3 displays the MAE distributions of both the ANN and MaxEnt methods, used with and without the PCA. We first notice that ANN has two main advantages over the MaxEnt approach. Firstly, the mean MAE of ANN remains lower than the one of MaxEnt at all noise levels. Second, the MAE of ANN shows a smaller spread. Adding a preprocessing PCA step has the remarkable advantage of of reducing the dependance of the accuracy of results on the noise level. However, this filter enlarges the errors at the lowest levels of noise and creates small counts in the whole interval of errors. In case of ANN, there is no benefit of using the PCA methods, while results for high levels of noise are improved in the case of MaxEnt. These results suggest that PCA efficiently removes noise from the data, but also alters some important characteristics. When using PCA in practice, one should carefully study the tradeoff between the noise reduction and the accuracy of the model. Adding more principal components will improve the accuracy, but will also penalize the efficiency of noise reduction. Fig. 4a summarizes our observations. ANN outperforms MaxEnt and behaves even better with increasing the noise level, while the PCA filtering step offers better results at high noise levels.

It is worth discussing also the limitations of the proposed method. One drawback of our supervised learning approach is that the model is optimized for a particular inverse temperature (here, $\beta = 2$), whereas the MaxEnt method can provide it as a parameter. A direct approach for extending our model to arbitrary values of β would be to train a separate ANN for each temperature. Indeed, predictions based on our ANN remain stable in the vicinity of the optimal temperature, as shown in Fig. 4b. Another worth-trying idea would be to add β as an input parameter to the model and train it over a range of temperatures. Both of these approaches would increase the training time, but would have no consequences on the computation time of a prediction.

Finally, we would like to underline the computational efficiency of our approach compared to MaxEnt. ANN allows a direct mapping between Green's functions and the spectral densities. In that sense, we can define it as a direct way of solving the problem. In contrast, MaxEnt is an iterative method which requires generating trial functions until convergence is reached. In our computational setup, the time required for converting 500 $(G(\tau), A(\omega))$ pairs with $\eta = 10^{-5}$ is 5 seconds in the case of ANN (including the loading of the required libraries), while MaxEnt would need 51 minutes using the same setup.

To summarize, thanks to the stability of the forward problem, we have built an artificial neural network that solves the analytic continuation problem with an accuracy similar to that of the commonly used maximum entropy approach, but at a fraction of its computational cost. We have also shown that our ANN model performs better for noisy inputs and implemented a denoising filter based on the principal component analysis to improve the results. Adding more data and increasing the number of parameters can further improve the accuracy, although training must be performed with care to avoid overfitting issues. The great representative capacity of deep neural networks suggests that other inverse problem can be solved in a similar way, provided that a dataset can be built.

Note: Another work using convolutional neural network to solve the analytic continuation problem [35] appeared when the present manuscript was in preparation.

[1] S. I. Kabanikhin, *Inverse and Ill-posed Problems: Theory and Applications* (Walter de Gruyter, Berlin, Boston, 2011).

[2] A. G. Ramm, *Inverse Problems: Mathematical and Analytical Techniques with Applications to Engineering* (Springer US, Boston, 2006).

[3] S. I. Kabanikhin, Journal of Inverse and Ill-posed Problems. **16**, 317 (2008).

[4] J. Gubernatis, N. Kawashima, and P. Werner, *Quantum Monte Carlo Methods: Algorithms for Lattice Models* (Cambridge University Press, Cambridge, 2016).

[5] M. Jarrell and J. Gubernatis, Physics Reports **269**, 133 (1996).

[6] A. W. Sandvik, Phys. Rev. B **57**, 10287 (1998).

[7] K. S. D. Beach, arXiv:cond-mat/0403055.

[8] O. F. Syljuåsen, Phys. Rev. B **78**, 174429 (2008).

[9] E. Klepfish, C. Creffield, and E. Pike, Nuclear Physics B-Proceedings Supplements **63**, 655 (1998).

[10] K. Beach, R. Gooding, and F. Marsiglio, Physical Review B **61**, 5147 (2000).

[11] S. Fuchs, T. Pruschke, and M. Jarrell, Phys. Rev. E **81**, 056701 (2010).

[12] J. Otsuki, M. Ohzeki, H. Shinaoka, and K. Yoshimi, Phys. Rev. E **95**, 061302 (2017).

[13] Q.-S. Wu, Y.-L. Wang, Z. Fang, and X. Dai, Chinese Physics Letters **30**, 090201 (2013).

[14] K. Dahm and A. Keller, in *ACM SIGGRAPH 2017 Talks*, SIGGRAPH '17 (ACM, New York, NY, USA, 2017) pp. 73:1–73:2.

[15] H. Li, J. Schwab, S. Antholzer, and M. Haltmeier, arXiv:1803.00092.

[16] M. T. McCann, K. H. Jin, and M. Unser, IEEE Signal Processing Magazine **34**, 85 (2017).

[17] L.-F. Arsenault, R. Neuberg, L. A. Hannah, and A. J. Millis, Inverse Problems **33**, 115007 (2017).

[18] F. Zhang and T. Rice, Physical Review B **37**, 3759 (1988).

[19] J. Kondo, Progress of theoretical physics **32**, 37 (1964).

[20] R. Levy, J. LeBlanc, and E. Gull, Computer Physics Communications **215**, 149 (2017).

[21] R. E. Bellman, *Adaptive Control Processes: A Guided Tour* (Princeton University Press, Princeton, 2015).

[22] J. A. Lee and M. Verleysen, *Nonlinear Dimensionality Reduction* (Springer Science & Business Media, New York, 2007).

[23] L. Huang, Chinese Physics B **25**, 117101 (2016).

[24] J. W. Gibbs, Nature **59**, 200 (1898).

[25] A. Weiße, G. Wellein, A. Alvermann, and H. Fehske, Rev. Mod. Phys. **78**, 275 (2006).

[26] The supplementary material document contains details of the design and training of the proposed ANN as well as the discussions of the effect of temperature variations, test dataset size and PCA filtering.

[27] S. Kullback and R. A. Leibler, The Annals of Mathematical Statistics **22**, 79 (1951).

[28] S. Loff and C. Szegedy, arXiv:1502.03167.

[29] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, Journal of Machine Learning Research **15**, 1929 (2014).

[30] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, Cambridge, 2016).

[31] D. P. Kingma and J. Ba, arXiv:1412.6980.

[32] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed., Springer Series in Statistics (Springer-Verlag, New York, 2002).

[33] L. Zhang, W. Dong, D. Zhang, and G. Shi, Pattern Recognition **43**, 1531 (2010).

[34] R. Fournier, "ACANN project repository," https://github.com/rmnfournier/ACANN (2018).

[35] H. Yoon, J.-H. Sim, and M. J. Han, arXiv:1806.03841v1.