Supplementary Information for

Can Machine Learning Predict Extreme Events in Complex Systems?

Di Qi and Andrew J. Majda

Department of Mathematics and Center for Atmosphere and Ocean Science, Courant Institute of Mathematical Sciences, New York University, New York, NY 10012

This PDF file includes:

Supporting Information Text

A. The truncated KdV model with statistical phase transition

Here we provide more details about the derivation and properties of the truncated *Korteweg-de Vries* (tKdV) equations for surface water wave turbulence, and the statistical phase transition in the Gibbs invariant measures used as the test model in the *main text*.

A.1. Mathematical formulation of the truncated KdV equation as a Hamiltonian system. The classic KdV equation (1) can be written in the standard form as

$$u_t + uu_x + u_{xxx} = 0, \quad x \in [-\pi L_0, \pi L_0]\,. \tag{S1}$$

The state variable $u(x,t)$ for the leading-order surface wave disturbance is defined on a periodic geometry of length $2\pi L_0$. The KdV equation [S1] can be also recognized as a *Hamiltonian system* by the form

$$\dot{u} = \{u, \mathcal{H}\} = \mathcal{J}\frac{\delta \mathcal{H}}{\delta u}, \quad \mathcal{J} = -\partial_x,\ \mathcal{H} = \int_{-\pi L_0}^{\pi L_0} \left(\frac{1}{6}u^3 - \frac{1}{2}u_x^2\right) dx. \tag{S2}$$

The Poisson bracket is defined by the symplectic operator $\mathcal{J}$

$$\{\mathcal{F}, \mathcal{G}\} = \int_{-\pi L_0}^{\pi L_0} \frac{\delta \mathcal{F}}{\delta u}\mathcal{J}\frac{\delta \mathcal{G}}{\delta u} dx,$$

which forms a skew-symmetric and bilinear form satisfying the Jacobi identity, $\{\{\mathcal{F},\mathcal{G}\},\mathcal{H}\} + \{\{\mathcal{H},\mathcal{F}\},\mathcal{G}\} + \{\{\mathcal{G},\mathcal{H}\},\mathcal{F}\} = 0$, acting on functionals $\mathcal{F}(u)$ and $\mathcal{G}(u)$. The evolution of any functional $\mathcal{F}(u)$ obeys the dynamical equation

$$\mathcal{F}_t = \{\mathcal{F}, \mathcal{H}\}\,.$$

Immediately, we have the conservation of the Hamiltonian in [S2], $\mathcal{H}_t = \{\mathcal{H}, \mathcal{H}\} = 0$. Besides the Hamiltonian $\mathcal{H}$, the momentum $\mathcal{M}$ and energy $\mathcal{E}$ are another two important conserved quantities in the KdV equation defined as

$$\mathcal{M}(u) = \int_{-\pi L_0}^{\pi L_0} u dx, \quad \mathcal{E}(u) = \frac{1}{2}\int_{-\pi L_0}^{\pi L_0} u^2 dx.$$

In modeling shallow water waves using the KdV equation, it is convenient to adopt a normalized version of the equation [S1]. The state variable u is normalized with zero mean and unit energy with the change of variables

$$t = \tilde{t},\ x = L_0\tilde{x} + M_0\tilde{t}, \quad u = E_0^{1/2}L_0^{-1/2}\tilde{u} + M_0,$$

where $M_0 = \int_{-\pi L_0}^{\pi L_0} u dx$ is the conserved total momentum and $E_0 = \frac{1}{2}\int_{-\pi L_0}^{\pi L_0} u^2 dx - \pi L_0 M_0^2$ is the conserved total energy from the original system [S1], and L_0 defines the characteristic length scale of the traveling water waves. The additional shift in time $M_0 t$ in the new coordinate creates the Doppler shift from the non-zero mean momentum M_0. In this way, the total momentum is normalized to zero $\mathcal{M}(\tilde{u}) = 0$ without loss of generality due to the Galilean invariance. The total energy in the normalized state $\tilde{u}$ is rescaled to unity, $\mathcal{E}(\tilde{u}) = 1$, conserved during the evolution, while E_0 characterizes the total energy injected in the system. For simplicity in representation, we use the normalized state variables and neglect the 'tildes' in the notations.

To investigate the turbulent dynamics in different scales generated from the KdV equation, usually a Galerkin projection $\mathcal{P}_\Lambda$ is applied to the state variable u with a high wavenumber truncation up to Λ

$$u_\Lambda(x,t) \equiv \mathcal{P}_\Lambda u = \sum_{|k|\le\Lambda} \hat{u}_k(t)\, e^{ikx}, \tag{S3}$$

with in total $J = 2\Lambda + 1$ grid points. The Galerkin truncated state variable u_Λ is normalized with zero mean $\mathcal{M}_\Lambda = \hat{u}_0 = 0$ and unit energy $\mathcal{E}_\Lambda = 2\pi\sum_{k=1}^{\Lambda}|\hat{u}_k|^2 = 1$. Therefore, the water wave motion is described by the truncated KdV equation (tKdV) by projecting the continuous equation [S1] to the truncated subspace with water depth D_0 dependence (1, 2)

$$\frac{\partial u_\Lambda}{\partial t} + \frac{D_0^{-3/2}}{2}E_0^{1/2}L_0^{-3/2}\frac{\partial}{\partial x}\mathcal{P}_\Lambda(u_\Lambda)^2 + D_0^{1/2}L_0^{-3}\frac{\partial^3 u_\Lambda}{\partial x^3} = 0, \quad x \in [-\pi, \pi]\,. \tag{S4}$$

The tKdV model [S4] is non-dimensionalized in the periodic domain $[-\pi, \pi]$ with the three model parameters (E_0, L_0, D_0). The additional projection in front of the quadratic term u_Λ^2 is used to remove the aliasing modes that go beyond the range $|k| > \Lambda$. The conserved Hamiltonian is discretized accordingly in the finite dimensional subspace decomposed into the difference of two components containing cubic and quadratic terms

$$\mathcal{H}_\Lambda = D_0^{-3/2}E_0^{1/2}L_0^{-3/2}H_3(u_\Lambda) - D_0^{1/2}L_0^{-3}H_2(u_\Lambda)\,, \qquad H_3(u) = \frac{1}{6}\int_{-\pi}^{\pi} u^3 dx,\ H_2(u) = \frac{1}{2}\int_{-\pi}^{\pi} u_x^2 dx. \tag{S5}$$

Above, the cubic term H_3 describes the skewness of the state, while the quadratic term H_2 characterizes the *slopes of the surface waves*, u_x. The advantage of adopting the normalized formulation [S4] with Hamiltonian [S5] is that it enables us to

easily control the different cases with changing statistics from a unified model setup. The amplitudes of the characterizing model parameters (E_0, L_0, D_0) can be discovered from a scale analysis from the experimental data. For the direct numerical simulations in the *main text*, we pick model parameter values as $E_0 = 100, L_0 = 6, D_0 = 0.24$ and $J = 32$ following the derivation in (2) from a detailed scale analysis.

It can be shown that the conserved quantities above are still conserved in this truncated system. Especially, the Hamiltonian structure of the previous continuous equation is maintained in this semi-discrete tKdV equation. The truncated equation [S4] stays as a Hamiltonian system with the corresponding discrete Hamiltonian $\mathcal{H}_\Lambda$. Furthermore, the truncated system [S4] satisfies the Liouville property (3, 4), thus equilibrium statistical mechanics can be constructed based on the conserved quantities.

A.2. Equilibrium statistical mechanics for the Gibbs invariant measures. For a better characterization of the turbulent solutions of the tKdV model, we introduce a statistical description of state u captured by ensemble simulations. First, the equilibrium probability distribution can be quantified by an invariant statistical measure. The equilibrium invariant measure is dictated by the conservation laws in the tKdV equation. There exist two important conserved functionals, the total energy $\mathcal{E}_\Lambda$ and the Hamiltonian $\mathcal{H}_\Lambda$, in the tKdV equation [S4]. The choice is to pick a mixed Gibbs measure with microcanonical ensemble in the quadratic energy $\mathcal{E}_\Lambda$ and canonical ensemble in the Hamiltonian $\mathcal{H}_\Lambda$ (3, 4). The invariant Gibbs measure is then defined based on canonical Hamiltonian fixed on the isosurface with constant energy (normalized to unit)

$$\mathcal{G}_\theta\left(u_\Lambda; E\right) = C_\theta \exp\left(-\theta \mathcal{H}_\Lambda\right) \delta\left(\mathcal{E}_\Lambda - 1\right),$$

with θ the *inverse temperature*. Summarizing the expressions for the truncated variables, the invariant Gibbs measure for the tKdV model [S4] about the normalized state variable u_Λ with unit energy can be rewritten in the following explicit form

$$\mathcal{G}_\theta\left(u_\Lambda\right) = C_\theta \exp\left(-\theta\left\{h_3 \int_{-\pi}^{\pi} u_\Lambda^3 dx - h_2 \int_{-\pi}^{\pi} \left(\partial_x u_\Lambda\right)^2 dx\right\}\right) \delta\left(\frac{1}{2}\int_{-\pi}^{\pi} u_\Lambda^2 dx - 1\right), \quad \text{[S6]}$$

with the coefficients $h_3 = \frac{1}{6} E_0^{1/2} L_0^{-3/2} D_0^{-3/2}$ and $h_2 = \frac{1}{2} L_0^{-3} D_0^{1/2}$ depending on the model parameters. A constant mean state will not alter the final invariant measure with a Doppler shift in the solution. The expectation of any functional $F(u)$ can be computed based on the above invariant measures [S6] using proper sampling strategies

$$\langle F \rangle_{\mathcal{G}_\theta} \equiv \int F(u)\, \mathcal{G}_\theta(u)\, du.$$

The invariant measures $\mathcal{G}_\theta$ predict the equilibrium PDFs of the system and can be sampled to serve as the initial ensemble for direct numerical simulations of the tKdV equation to generate different final model statistics. The Gibbs invariant measure can be sampled effectively using a proper Markov chain Monte Carlo scheme (2).

The distinct statistics generated from the tKdV model can be controlled by the the inverse temperature parameter θ. It is found (2, 5) that the negative temperature regime, $\theta < 0$, gives the correct energy spectra and PDFs consistent with the experiments. The Gibbs invariant measure [S6] transfers from near-Gaussian to highly skewed distribution as the amplitude of the inverse temperature θ increases. Three typical statistical regimes with near-Gaussian statistics ($\theta = -0.1$), mildly skewed PDF ($\theta = -0.25$), and strongly skewed PDF ($\theta = -0.5$) are used as test regimes in the *main text*.

Besides, the autocorrelation functions characterizes the mixing properties of the turbulent system. It is usually useful to consider both the autocorrelations at the physical grid points as well as in the spectral modes. The autocorrelation functions for the physical grids $\mathcal{R}_{ij}$ or between two spectral modes $\hat{\mathcal{R}}_{kl}$ can be computed correspondingly as

$$\mathcal{R}_{ij}(\tau; t) = \langle u_i(t+\tau)\, u_j(t)\rangle, \quad \hat{\mathcal{R}}_{kl}(\tau; t) = \langle \hat{u}_k(t+\tau)\, \hat{u}_l^*(t)\rangle, \quad \text{[S7]}$$

where $\langle\cdot\rangle$ can be viewed as the statistical average in ensemble members. With the homogeneous statistics for translation invariance, the formulas [S7] for the autocorrelation functions can be simplified as $\mathcal{R}_{ij}(\tau; t) \equiv \mathcal{R}(\tau)$ and $\hat{\mathcal{R}}_{kl}(\tau; t) \equiv \hat{\mathcal{R}}_k(\tau)\, \delta_{kl}$ independent of the starting time t for stationary processes. Accordingly, the decorrelation time is defined as the time integration of the autocorrelation functions

$$T_{\text{decorr}} = \int_0^\infty \mathcal{R}(\tau)\, d\tau, \quad \hat{T}_{\text{decorr},k} = \int_0^\infty \hat{\mathcal{R}}_k(\tau)\, d\tau. \quad \text{[S8]}$$

They are used to characterize the mixing time scale in the physical grids as well as the mixing rates of the Fourier modes for different scales. The larger scales often get correlated for longer time than the smaller scale modes.

B. Details on the convolutional neural network architecture for extreme event prediction

In the following we summarize the detailed implementation of the deep convolutional neural network used in the *main text* for learning turbulent dynamics and predicting extreme events.

B.1. A densely connected mixed-scale neural network for imaging processing. On the update in each single layer, the input data is arranged as the tensor $\mathbf{x} \in \mathbb{R}^{J\times N\times C}$, where $J \times N$ is the input 2-dimensional model data with J spatial grid points and N time steps. C is the number of channels starting with a single channel $C = 1$ in the first layer from the input data, and in later layers in the densely connected network case it contains a combination all the previous layer data in history. The output data $\mathbf{y} \in \mathbb{R}^{J\times N\times 1}$ is set to have the same tensor dimension size with a single channel output ($C' = 1$) for the prediction.

Specifically, each layer updates the input data use the general operator as

$$\mathbf{y} = \sigma\left(g_h\left(\mathbf{x}\right) + b\right), \quad g_h = \sum_{i=1}^{C} h^i * x^i,$$

where $h * x$ is the convolution operator with the filter kernel h, b is a constant parameter for the model bias, and σ is the nonlinear operator. Usually in a training data set from an ensemble of solutions with size M, the neural network updates each ensemble member separately and uses the final output to update the cost function. The parameters in the convolution operator g_h and the bias b change for different layers and for different output channels. The rectified linear unit (ReLU) function is taken as the nonlinear operator

$$\sigma\left(x\right) = \max\left\{x, 0\right\}.$$

With the i-th channel $\mathbf{x}^i \in \mathbb{R}^{J\times N}$ from the input data, the model parameters in one network layer includes:

- The convolutional filter kernel h^i for the i-th channel data covering the dimensions in space and time. It usually covers a small window with size $w_1 \times w_2$, where w_1 is determined by the correlation in the spatial direction and w_2 is determined by the temporal correction;
- The bias b_i for each output layer as a scalar parameter to be trained in the network. The bias is added before the nonlinear operation σ;

In a *feedforward deep neutral network*, the input data in the l-th layer $\mathbf{x}_l$ is only feed to the next $(l + 1)$-th layer through the convolution operator for the output in the m-th channel

$$\mathbf{x}_{l+1}^m = \sigma\left(g_{lm}\left(\mathbf{x}_l\right) + b_{lm}\right), \quad g_{lm}\left(\mathbf{x}_l\right) = \sum_{i=0}^{C_l} h_{lm}^i * \mathbf{x}_l^i. \qquad [S9]$$

Above σ is the nonlinear operator such as the ReLU. The convolution operator g_{lm} sends the input data $\mathbf{x}_l$ in the l-th layer to the next layer on the m-th channel. b_{lm} adds the bias to each channel, and h_{lm}^i is the convolution kernel in a small size. The feedforward deep network may require a larger number of layers to work and more model parameters to train. It may also require proper downscaling and upscaling going through the layers (then the size of the data changes through the network layers), while these downscaling and upscaling may not be an feasible approach for simulating the dynamical model time integration steps.

A densely connected and mix-scale structure. *A mixed-scale dense neural network* (MS-D Net) (6) mixes different scales within each layer using a dilated convolution, and densely connects all the feature maps. First, the dilated filter $h_{lm}^i\left(s_{lm}\right)$ convolves the grids with a distance as multiples of length s. The first layer starts with a non-dilated filter with size $w_1 \times w_2$ (usually a 3×3 filter in practice). Then the dilation is increased by 1 (that is, fill zeros in the convolution kernel h) at each following layer until it reaches the maximum dilation S (set as $S = 5$ in the present test). The dilated convolutions are desgined to capture additional features within different distances in the images. It is used to assimilate the multiscale schemes in the PDE discretization. Large scale information is first extracted in the early layers, while the deeper layers improve the smaller scales. The mixed-scale structure can avoid the use of downscaling and upscaling operations that are usually necessary in the direct feedforward networks. Thus different (spatial and temporal) scales are included with the convolution filter kernels in different lengths.

Next, the dense connection includes information in all the previous layers $n = 0, \cdots, l$ to update the output data in the next layer $n = l + 1$

$$\mathbf{x}_{l+1}^m = \sigma\left(g_{lm}\left(\mathbf{x}_0, \cdots, \mathbf{x}_l\right) + b_{lm}\right), \quad g_{lm}\left(\mathbf{x}_0, \cdots, \mathbf{x}_l\right) = \sum_{n=0}^{l}\sum_{i=0}^{C_n} h_{nm}^i\left(s_{nm}\right) * \mathbf{x}_n^i. \qquad [S10]$$

Here g_{lm} goes from the l-th layer to the next $(l + 1)$-th layer on the m-th channel. All the previous layer information $\{\mathbf{x}_0, \cdots, \mathbf{x}_l\}$ is used for the updating. In the *last layer* of the network, a fully connected layer is used to combine all the previous features together. It is equivalent to using a convolutional network with a filter kernel of size 1×1

$$\mathbf{y} = \sigma\left(\sum_{j,k} h_{jk} x_{jk} + b\right),$$

where x_{jk} is the j-th row and k-th column of the input data $\mathbf{x}$ and including all the previous layers. This is a linear combination of all the previous layer outputs.The mixed-scale dense neural network requires fewer feature maps and trainable parameters,

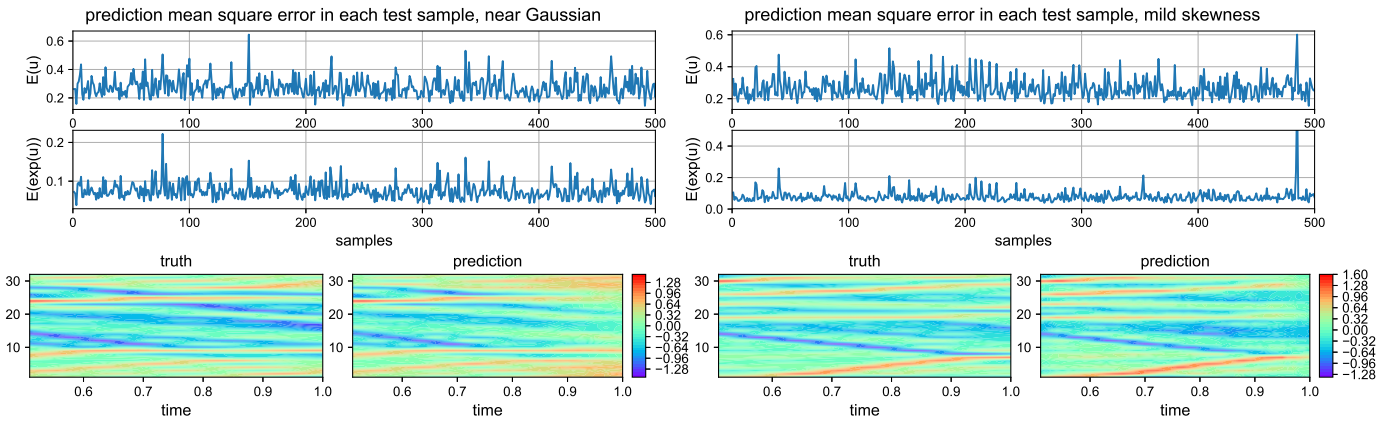

Fig. S1. Prediction using the same optimized neural network with $L = 80$ layers in regimes with near-Gaussian statistics (left) and mildly skewed statistics (right). The first row plots the relative square error for the state u and the scaled error for $\exp(u)$ among 500 test samples. The lower row shows one typical snapshot for each prediction case with near-Gaussian or medium skewness.

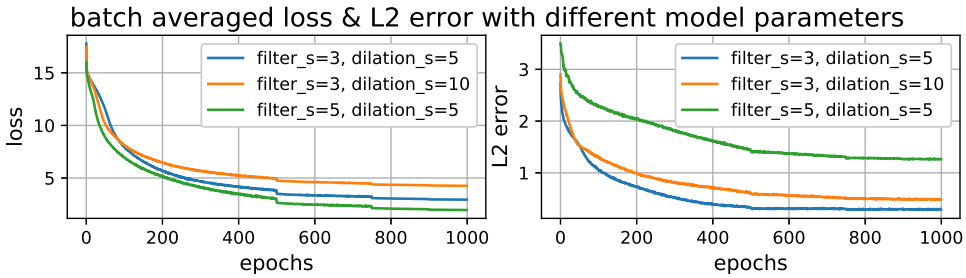

Fig. S2. Training neural networks with different convolution filter kernel sizes and maximum dilations.

so it is easier to handle compared with the direct feedforward network. For the mixed scale network structure, it is hopeful that it can first decompose the different scales (for large-scale information with a large dilation distance s). Then information at different scales communicates with each other through the dense network connection. As a result, more accurate prediction is expected since the dynamical model structure is better represented through the MS-D Net.

Finally, a stochastic gradient descent (SGD) method inside the batch data is used for the optimization of the model parameters. An adaptive learning rate optimization, *Adam*, is applied for determining the optimal learning rate for each iteration in the SGD. The learning rate is decayed by a factor $\gamma = 0.1$ once the number of epochs reaches some milestone. The Adam method is generally regarded as being fairly robust to the choice of hyperparameters.

The network is implemented using *PyTorch* and is performed on one or two NVIDIA P100 GPUs. The training of the neural network goes through 1000 epochs, while it is observed in most cases 200 epochs can already reduce the error in the loss function to a small value.

B.2. More results for model dependence on hyperparameters. Here we provide more results in the performance of the deep neural network in complementary to the main results shown in the *main text*. The neural network follows the standard structure described before. We pick the number of layers as $L = 80$ and a symmetric convolution kernel with size 3×3, and the same maximum dilation size in the two (spatial and temporal) directions is considered. In company with the prediction for the highly skewed regime in Fig. 4 of the *main text*, Figure S1 shows the additional predictions through the same optimized neural network among other statistical regimes with near-Gaussian and mildly skewed PDFs respectively. As expected, uniformly high accuracy is achieved again in the other two test regimes and the flow snapshots confirm the skill of the network to capture both the dominant traveling waves as well as the smaller scale turbulent structures in the flow field.

Next, the model dependence on the hyperparameters is investigated. Figure S2 first compares the evolution of training errors using different convolution filter kernel sizes and different maximum inflation sizes in the MS-D Net. Larger kernel sizes and bigger maximum inflations extend the multiscale connections to a wider range, though the values in far away points might not be closely correlated. It shows in the training results that too large dilation hampers the improvement in reducing the model error. Too large a convolution kernel shape also damages the model prediction skill in the error even though it may give a smaller loss function in the training process.

Finally, the different by using the combined relative entropy loss function with $\alpha = 1$ is compared with the single relative loss function $\alpha = 0$ only measuring the positive values in Figure S3. It shows that combining both the positive and negative extreme values through the two empirical partition functions helps to stabilize the convergence in the training process. Using a loss function only containing a positive value component, the training error becomes much less stable and slower to converge to

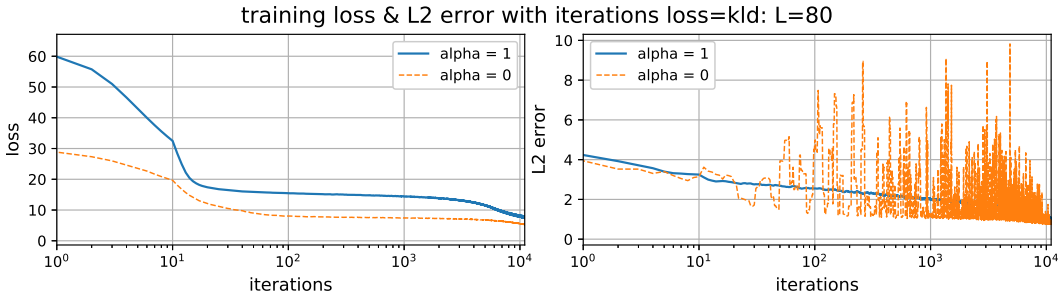

Fig. S3. Training neural networks with the combined loss function $\alpha = 1$ and the loss function only comparing the positive extreme values $\alpha = 0$ using the relation entropy.

the minimum error during the training process.

References

1. Johnson RS (1997) *A modern introduction to the mathematical theory of water waves.* (Cambridge university press) Vol. 19.
2. Majda AJ, Moore MNJ, Qi D (2019) Statistical dynamical model to predict extreme events and anomalous features in shallow water waves with abrupt depth change. *Proceedings of the National Academy of Sciences* 116(10):3982–3987.
3. Abramov RV, Kovačič G, Majda AJ (2003) Hamiltonian structure and statistically relevant conserved quantities for the truncated burgers-hopf equation. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* 56(1):1–46.
4. Majda AJ, Wang X (2006) *Nonlinear dynamics and statistical theories for basic geophysical flows.* (Cambridge University Press).
5. Bajars J, Frank J, Leimkuhler B (2013) Weakly coupled heat bath models for gibbs-like invariant states in nonlinear wave equations. *Nonlinearity* 26(7):1945.
6. Pelt DM, Sethian JA (2018) A mixed-scale dense convolutional neural network for image analysis. *Proceedings of the National Academy of Sciences* 115(2):254–259.