Supplementary Materials

This is the supplementary materials for the paper *"Z. Long, Y. Lu, X. Ma and B. Dong, PDE-Net: Learning PDEs from Data, ICML 2018"*.

1. Data generation

We adopt periodic boundary condition for the first example, the convection-diffusion equation. Spectral method with 30×30 tensor-product basis functions has been used for spatial discretization in the computation domain $[0, 2\pi] \times [0, 2\pi]$. The temporal discretization is 4th order Runge-Kutta method with time step $\Delta t = 0.015$. Then the solution data are interpolated to a 200×200 mesh. To mimic real scenarios where we can only measure data with limited resolution and during data acquisition there might be minor shifts in the acquired data, the data that we actually feed to the network are of 50×50 resolution, which are downsampled from the originally generated data, and with a random 0∼3 grids shift in each direction. The initial value $u_0(x, y)$ is assumed to have the following form,

$$u_0(x,y) = \sum_{|k|,|l| \leq N} \lambda_{k,l} \cos(kx + ly) + \gamma_{k,l} \sin(kx + ly), \quad (1)$$

where $N = 9$, $\lambda_{k,l}, \gamma_{k,l} \sim \mathcal{N}(0, \frac{1}{50})$, and k and l are chosen randomly.

The data of the second example, the diffusion equation with nonlinear source, are generated by central difference scheme for spatial discretization on a 100×100 mesh in the computation domain $[0, 2\pi] \times [0, 2\pi]$, and then restricted to a 50×50 mesh. We do not conduct random shifts for this case due to the restriction from the Dirichlet boundary condition we adopt for the equation. The temporal discretization is forward Euler scheme with time step $\Delta t = 0.0009$. The initial value $u_0(x, y)$ is generated from

$$u_0(x,y) = u_0'(x,y) \frac{x(2\pi - x)y(2\pi - y)}{(2\pi)^4}, \quad (2)$$

where u_0' is obtained from (1) with maximum allowable frequency $N = 6$.

2. Details on layer-wise training

In the training process for PDE-Net, we use layer-wise training. We start with training the PDE-Net on the first δt-block with a batch of data, and then use the results of the first δt-block as the initialization and restart training the PDE-Net on the first two δt-blocks with another batch of data. Repeat this procedure until we complete all n δt-blocks. Note that all the parameters in each of the δt-block are shared across layers. In addition, we add a warm-up step before the training of the first δt-block. The warm-up step is to obtain a good initial guess of the parameters of the point-wise neural network that approximates F by using frozen filters.

Note that we have tried the more straightforward training where we train the entire network directly. However, the training is very slow and even fails sometimes. We think it is mainly due to the difficulty of choosing a suitable initial set of parameters.

3. Long-time dynamics prediction

In the main body of the paper, we have shown that a well-trained PDE-Net, even only with 3 δt-blocks, can predict the long-time dynamics of the equation perfectly (see Figure 4 and Figure 10 in paper). In fact, a well-trained deeper PDE-Net can predict longer dynamics. In Figure 1 and Figure 2, we show the errors between the true dynamics and the predicting dynamics with a well-trained PDE-Net having different δt-blocks (only warm-up, 1 δt-block, 3 δt-blocks, and 10-δt blocks) until $t = 5$ for the first example and $t = 2$ for the second example. The results indicate that a deep PDE-Net can even predict the dynamics of the equation with a tolerable error for a very long time.

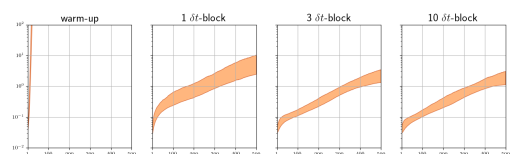

Figure 1. Long-time prediction for the PDE-Net with 7×7 filters for the first example. The horizontal axis indicates the prediction range from $1 \times \delta t$ to $500 \times \delta t$.

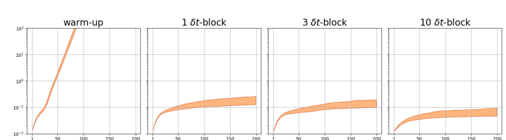

Figure 2. Long-time prediction for the PDE-Net with 7×7 filters for the second example. The horizontal axis indicates the prediction range from $1 \times \delta t$ to $200 \times \delta t$.

4. Discovery of hidden equation

In the main body of the paper, in order to have a more concise demonstration of the results, we only showed the coefficients $\{c_{ij} : i + j \leq 2\}$ learned by PDE-Net in Figure 5 of the main body of paper, and ignore the others. Here we show all of the coefficients $\{c_{ij} : i + j \leq 4\}$ in Figure 3 and Figure 4.

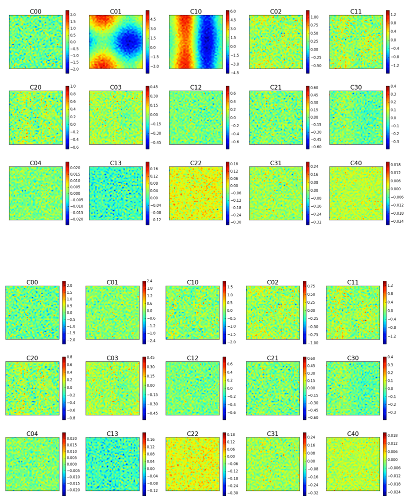

Figure 3. Upper: all of the learned coefficients by the PDE-Net with 6 δt-blocks and 5×5 filters. Lower: the errors between true and learned coefficients.

5. Generalization ability of PDE-Net

To further demonstrate how well the learned PDE-Net generalizes, we generate initial values following (1) with highest frequency equal to 12 for the first example, linear convection-diffusion equation. Note that the maximum allowable frequency in the training set is 9. The results of long-time prediction and the estimated dynamics are shown in Figure 5. Similarly, we generate initial values following (2) with highest frequency equal to 10 for the second example, diffusion equation with a nonlinear source. Note that the maximum allowable frequency in the training set is only 6. The results of long-time prediction and the estimated dynamics are shown in in Figure 6. Although oscillations are observed in the prediction, the estimated dynamic still captures the main pattern of the true dynamic.

6. Experiments on data with no noise

In the paper, all the experiments are based on noisy data, since the measurement error is inevitable in reality. The numerical results show that the PDE-Net is robust to noise. In fact, we also implement experiments to discover hidden

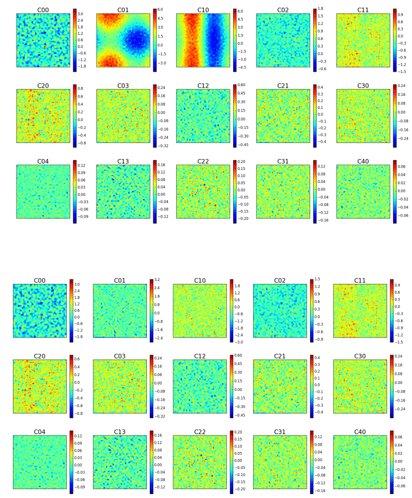

Figure 4. Upper: all of the learned coefficients by the PDE-Net with 6 δt-blocks and 7×7 filters. Lower: the errors between true and learned coefficients.

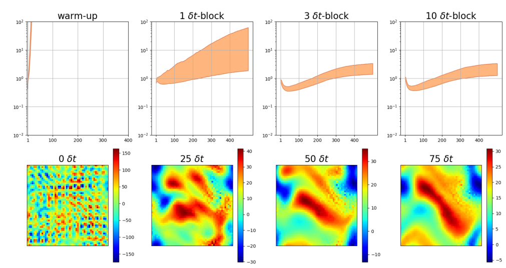

Figure 5. Testing with higher frequency initializations (linear convection-diffusion equation). First row: long-time prediction. Second row: estimated dynamics. Here, δt = 0.015.

equations by PDE-Net on noise-free data (data with no noise).

Considering the convection-diffusion equation example, we identify the coefficients by a 6-δt-block PDE-Net with 5×5 filters and 7×7 filters on the noise-free data set. The results are showed in Figure 7 and Figure 8.

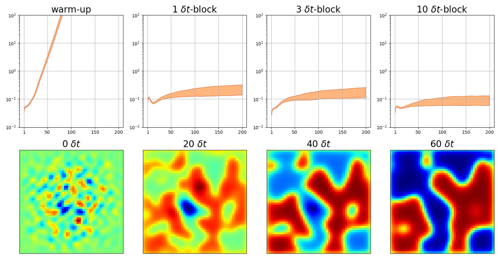

Figure 6. Testing with higher frequency initializations (diffusion equation with a nonlinear source). First row: long-time prediction. Second row: estimated dynamics.Here, $\delta t = 0.01$.

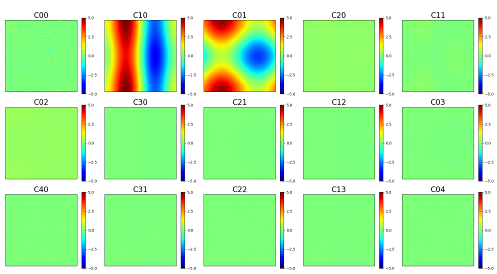

Figure 7. The learned coefficients by the PDE-Net with 6 δt-blocks and 5×5 filters on noise-free data.

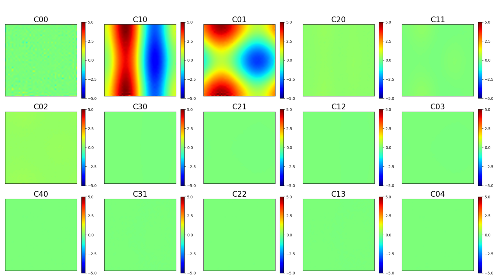

Figure 8. The learned coefficients by the PDE-Net with 6 δt-blocks and 7×7 filters on noise-free data.