

1. Executive summary (1-2 pages).

In the previous period we leveraged deep neural networks to predict the dynamics of turbulent vorticity simulations. We found over a single time step even networks with low-dimensional states were able to accurately predict the vorticity at a future frame. However, simple auto-encoder style networks alone were not able to accurately predict vorticity over a significant period.

In this period we were successful in accurately predicting mid-length turbulent vorticity dynamics by leveraging recurrent artificial neural network layers and an encoder - decoder scheme to generate an embedding for a given patch and predict the future vorticity. Extending these results to long horizon trajectories allowed for estimation of the maximum prediction horizon for a given model, however more investigation must be done to correlate network confidence with predictive accuracy.

1.1. Hypothesis 1. Turbulent vorticity can be modeled as a function of a latent state and transition function.

1.1.1. *What was to be tested? What was the expected outcome prior to testing?* We will exploit the time-dependency of turbulent flow to discover a model that is able to encode the latent state of a given patch of turbulent flow and evaluate the transition function to generate new vorticity predictions by updating the latent state.

This formulation iterates through the input to produce an encoding of the known history, and then sequentially generates future vorticity using a separate decoder network. This formulation has many advantages over learning a single state to future state mapping by forcing the transition function for the latent state to remain constant across all patches and time steps.

As a result, the learned model should learn faster and result in a more robust and accurate network.

1.1.2. *High-level summary of main results.* Using a recurrent encoder decoder model dramatically increased prediction performance over 20 future time-steps. By increasing the size of the latent state for prediction, the prediction accuracy further improved to match that of previous models only predicting a single time step of vorticity.

Additionally because the network is learning patch dynamics, evaluating can easily be extended to arbitrary sequence lengths allowing prediction of long sequences of vorticity.

1.1.3. *Discuss relevance to project and DARPA concerns.* This demonstrates a viable dynamics model for medium-horizon vorticity data can be discovered through deep recurrent neural networks. An impressive result given the complexity and dynamic nature of vorticity data.

1.2. Hypothesis 2. Long horizon prediction of turbulent vorticity data is intractable.

1.2.1. *What was to be tested? What was the expected outcome prior to testing?*

When predicting chaotic motion, small perturbations result in dramatic changes to the flow at a given time step. Given finite resolution, prediction becomes impossible beyond a certain horizon and predictions must diverge. We experiment with training models to estimate 5 minutes of a future vorticity to provide an impossible prediction task for the network. This formulation is expected to perform poorly as there exists a large amount of unexplainable error in such a long turbulent trajectory.

1.2.2. *High-level summary of main results.* Networks learned short-term predictions well, however as expected were unable to maintain prediction accuracy over the entire trajectory resulting in a mean average error of 0.1, significantly better than random however not to the $1e-4$ level provided

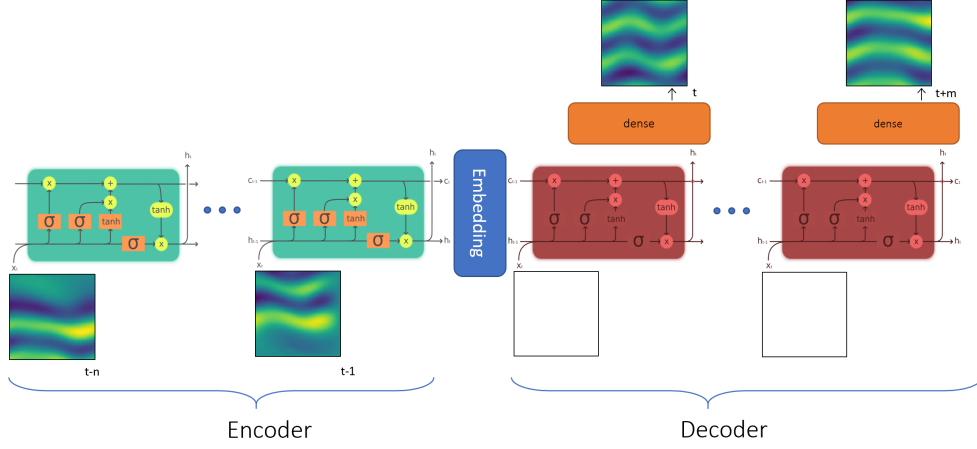


Figure 1: Illustration of encoder/decoder formulation 2 using separate long short term memory (LSTM) recurrent layers for the encoder and decoder with an additional linear activation layer (dense) to map the LSTM output to a vorticity image patch. For experiments in figure 3, $n = 20$ and $m = 20$

by previous medium-length prediction tasks. The gradients of the network were significantly noisier during training as a result of the irreducible error in the training set trajectories, which indicates naively extending the prediction horizon decreases the ability for the network to predict even short term features.

1.2.3. Discuss relevance to project and DARPA concerns. By demonstrating the difficulty of training long trajectories directly, we demonstrate the need for additional methods, such as curriculum learning, in order to learn vorticity transition functions that are able to predict beyond medium time horizons.

2. Planned activities and milestones for next reporting period.

1. Discovering the maximum time horizon for prediction. Training a series of networks with increasing prediction horizon until performance drops significantly should demonstrate the maximum time horizon for accurate vorticity prediction.
2. Add prediction accuracy estimate to network. By adding an accuracy estimation element to the network, lower penalty can be given to states where the network is not confident in its estimation. This could reveal properties of vorticity trajectories that prove difficult for deep networks to predict, as well as increase the accuracy of exiting training models.

3. Technical Formulation

Let $P_t \in \mathbb{R}^{360 \times 279}$ be the turbulent flow at time $t \in \{1, 2, \dots, 1000\}$ and define a window $w_{(r,t)}$ of P_t in terms of a 50px by 50px region $r \in \{(u, v) \mid u \in \{0, \dots, 309\}, v \in \{0, \dots, 228\}\}$ For a particular region and time $(r, t) \in (\mathbb{R}^2, \mathbb{R})$ the input x is a series of windows in the past, given by:

$$x = \{w_{(r,i)} \mid t - 20 \leq i \leq t - 1\} \in \mathbb{R}^{50 \times 50 \times 20}$$

and our target y is a single window of the form:

$$y = \{w_{(r,i)} \mid i = t\} \in \mathbb{R}^{50 \times 50}$$

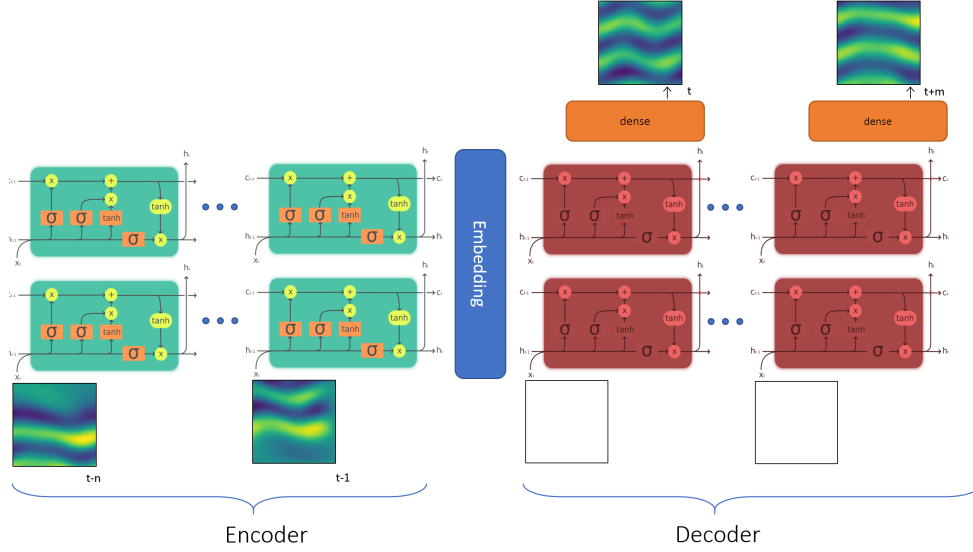


Figure 2: Illustration of encoder/decoder formulation 1, 3, and 4. This fomulation uses two stacked long short term memory (LSTM) recurrent layers for each the encoder and decoder with an additional linear activation layer (dense) to map the LSTM output to a vorticity image patch. Here $n = 20$ and $m = 20, 200$

To train the network we randomly sample 50000 region, time pairs $S_i = \{(r, t) \mid i \in \{0, 1, \dots, 4999\}\}$ and withhold 500 for validation creating two datasets, $X_{val} = \{w_{(S_i)} \mid i < 500\}$ and $X_{train} = \{w_{(S_i)} \mid i \geq 500\}$. We then learn a convolutional network $f(x)$ to minimize the Huber loss, $L(y, f(x))$ given by:

$$L(y, f(x)) = \begin{cases} \max(0, 1 - y f(x))^2 & \text{for } y f(x) \geq -1, \\ -4y f(x) & \text{otherwise.} \end{cases}$$

A network $f(x)$ is composed of a decoder and encoder network. Recurrent layers, namely long short term memory units (LSTM) are stacked to encode and decode an embedding of the input window shown in figures 1 and 2. Additionally, some networks utilize a dense layer representing a matrix multiplication of the output of the LSTM layer. This maps the output of the recurrent module to a single target y . Training is conducted over 1×10^6 batches of 64 input - label sequences using the TensorFlow back-end and Adam optimizer.

4. Issues and concerns (new & ongoing).

None

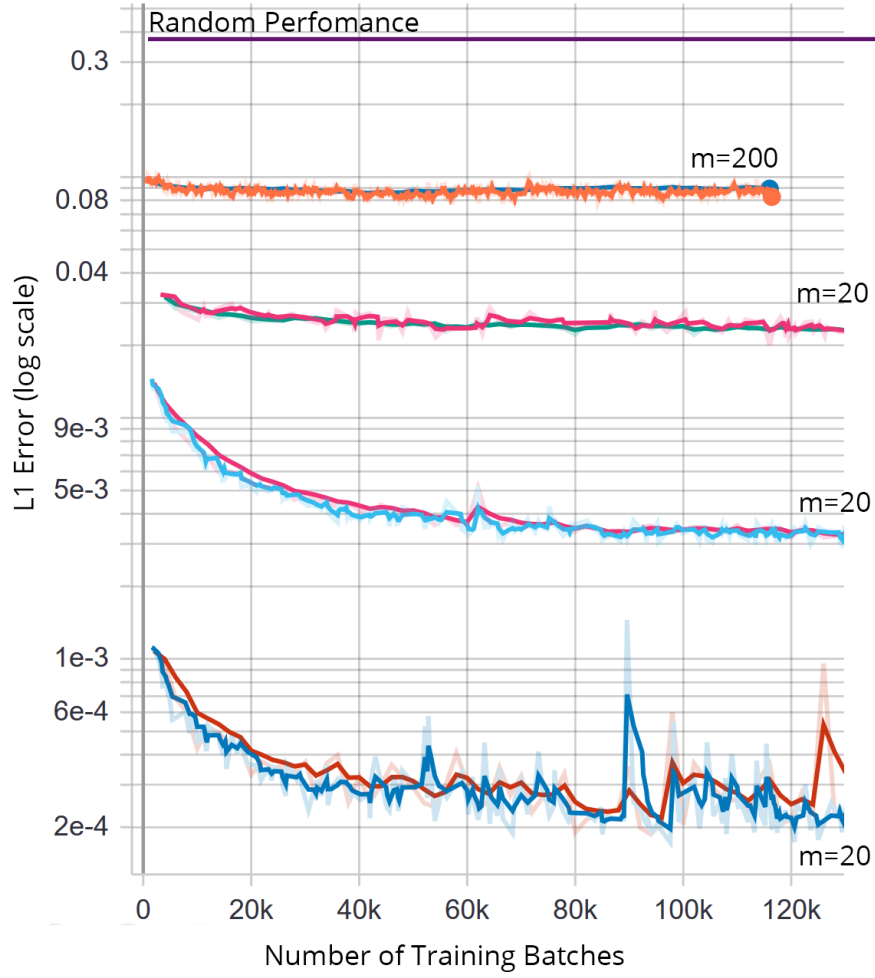


Figure 3: Comparison between four different networks and random performance, labeled top to bottom, random, network 1, 2, 3, and 4. Networks 2,3, and 4 are predict $m = 20$ time-steps of vorticity. Network 2 uses a single LSTM layer for encoding and decoding as shown in figure 1. Network 3 adds an additional LSTM layer during encoding and decoding, as illustrated in figure 2. Network 4 uses same formulation as network 3, but utilizes an increased embedding size. Network 1 uses the best parameters found when training network 4, but attempts to predict 200 time-steps of vorticity, performing noticeably worse than any network predicting only 20 time-steps.