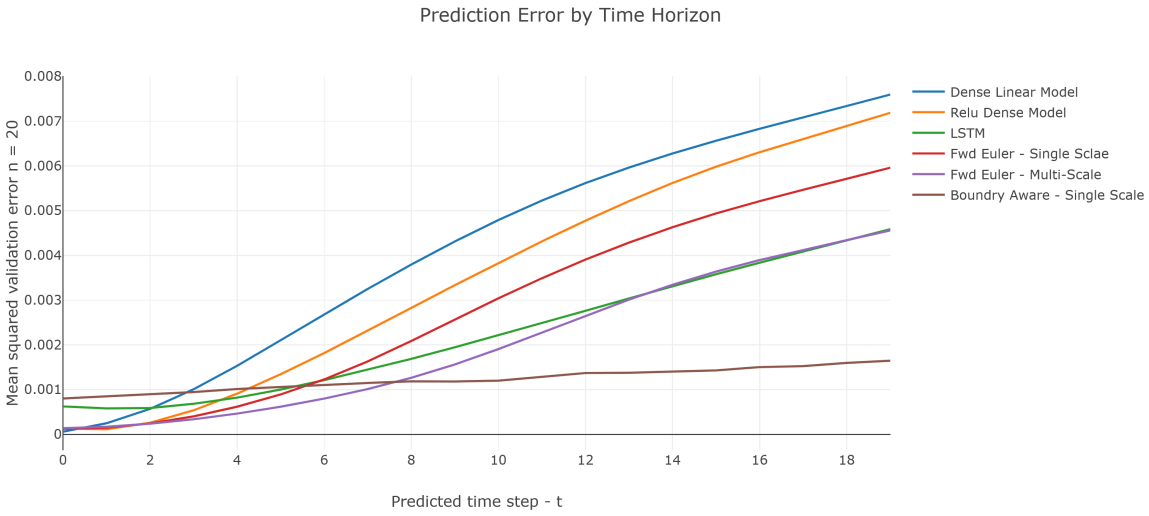

Figure 1: Mean squared error by predicted time-step. Note the strong consistency in accuracy over time offered by the boundary-aware architecture.

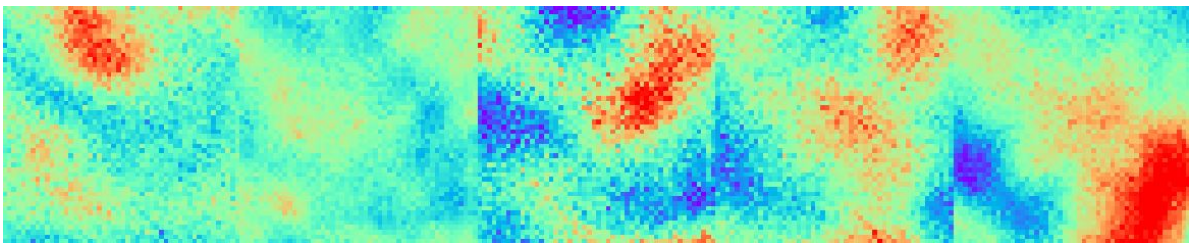

Figure 2: Visualization of prediction error $\hat{y} - y$ at $t = 20$ for baseline linear model. While early time-step accuracy is high, prediction degrades quickly as shown in figure 1.

1 Executive summary

1.1 Previous work

Previous work significantly improved performance of discovered architectures, using deep, multi-scale convolutions to learn complex dynamics efficiently in terms of network parameters. Additionally we developed a curriculum learning style approach, varying the prediction horizon leveraging temporal consistency of turbulent dynamics to further increases in network performance and robustness during training. Finally, we established a benchmark metric, based on relative accuracy as compared to a linear baseline, dubbed *RBA*, where perfect prediction represents a score of 1, and 0 indicates baseline mean-squared error.

1.2 Important Results

In this period, we addressed spatially inconsistent boundary conditions by developing a new boundary-aware network architecture. This boundary-aware architecture further reduces mean squared error by 56% as compared to the previous best performing forward-Euler, multi-scale model, while

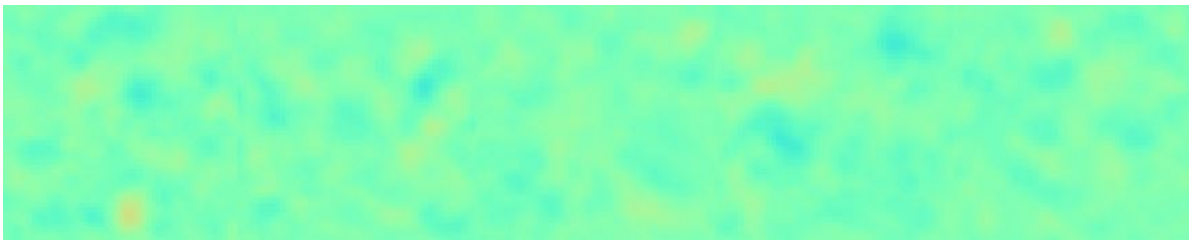

Figure 3: Visualization of prediction error $\hat{y} - y$ at $t = 20$ for the boundary-aware prediction model. Note that compared to previous approaches, boundary-aware architectures resolve low-frequency, long-term dynamics more accurately across the entire predicted trajectory as shown in figure 1.

Figure 4: Mean prediction error over a batch of patches for the forward-Euler single scale architecture. Note the artifacts along the edge of the region resulting from estimating boundary state.

maintaining an accurate prediction ($MSE_t < 0.002$) for the entire trajectory. While previous architectures have struggled with accumulation of error near the edges of predicted patches as shown in figure 4 and compounding systematic error in predicting long-term dynamics as seen in figure 2, the new architecture shows uniform error over the entire prediction region. This uniform error is illustrated by the comparatively gentle slope of the brown boundary aware architecture line in figure 1 and the uniform, low-magnitude, high-frequency error in figure 3.

1.3 Hypothesis 1 - Boundary-Aware Architectures

Previously, the forward Euler inspired architecture, illustrated in figure 6, leveraged convolutions at multiple scales to increase the fidelity of the dynamics model while maintaining the symmetry-constrained properties of a convolutional architecture. However, these architectures are sensitive

	MSE	RBA
Linear Baseline	0.004356	$\sim$
Fully Connected	0.003694	0.2374
LSTM	0.002326	0.3583
Foward Euler - Single Scale	0.002971	0.3857
Forward Euler - Multi Scale	0.002116	**0.5483**
Boundary Aware	**0.000922**	0.4351

Table 1: Comparison of network accuracy over 20 time-steps. Despite lower mean squared error, the boundary aware architecture has a lower RBA than multi-scale forward-Euler given its disadvantage over the baseline in early time-steps.

to changing boundary conditions because of the nature of convolutional layers incorporating local features which are not present on boundaries. This effect is compounded when convolution is repeatedly applied as is the case when rolling out predictions. When applying the dynamics across the entire flow, rather than discrete regions of turbulence, boundary conditions would be fixed at the physical borders, and therefore predictable and possible to estimate. However, given the patch formulation we see previously models unable to approximate boundary conditions which caused systematic error along borders as seen in figure 4.

To address this, we develop so called boundary-aware architectures that enforce boundary conditions during prediction roll-outs. This maintains the architectural assumption of spatially consistent dynamics, and grounds predictions by assuming true dynamics are known at patch edges. This architecture, illustrated in figure 7 dramatically increased prediction accuracy, especially over distant time-steps where compounding errors typically preclude accurate prediction.

1.3.1 What was to be tested? What was the expected outcome prior to testing? We tested constraining networks by padding the previous state $\hat{x}_t$ with ground truth boundary conditions x_t. We expected this to increase long-term prediction accuracy by eliminating compounding errors along boundaries, though by directly padding previous state with boundary conditions, this requires a different approach then previous network architectures as the dynamics are learned directly, rather than learning the dynamics of a hidden state,u_t.

1.3.2 High-level summary of main results. Using boundary-aware architectures, we were able to reduce MSE by 54% and decrease error-accumulation by a factor of 6. Further architecture improvements will likely lower MSE further given high relative MSE for early $t < 3$ time-steps.

1.3.3 Discuss relevance to project and DARPA concerns. This represents a significant improvement in long-term dynamics modeling for turbulent flow. The accumulation of compounding error can quickly cause predictions to diverge, however this approach has shown to reduce this compounding error accumulation by a factor of 6 enabling the simulation of long-term dynamics potentially 6x further into the future.

2 Achievements

2.1 Scientific Breakthroughs

None

2.2 Technology developments

None

2.3 Application results

Boundary aware architecture decreased compounding error accumulation by a factor of 6 suggesting that longer-term prediction experiments are necessary to test the limits of turbulent vorticity projection. Previously, error beyond 20 time-steps was too large for accurate ($MSE_t < 0.002$)

prediction, however current boundary -aware architectures easily maintain an accurate predicted vorticity through the entire 20 step trajectory.

2.4 Transitions achieved

None

3 Lessons Learned

Problems encountered/risks that occurred, and corresponding solutions/mitigations
None
Open Issues
None

4 Next Steps

Boundary-aware techniques show significant promise in extending the prediction horizon for turbulent patches. Additional architectures incorporating both multi-scale forward Euler inspired methods and boundary-aware techniques remain to be explored.
Boundary constraints eliminate network prediction of boundary conditions, which are varied given the spatial segmentation of the turbulent flow. In practice these boundary conditions may not be available for constraining predictions, therefore it is of interest to apply patch dynamics iteratively over the entire flow field using local predictions as estimated boundary conditions.
Short-term $(t < 3)$ predictions show larger than baseline mean squared error, suggesting that the network is not expressive enough to capture high-frequency features. Further experiments integrating additional network features from previous architectures are needed.

5 Technical details

To evaluate new models in a standard way, we establish a baseline performance using a fully connected linear network, and introduce a relative baseline accuracy metric, where σ represents the mean squared error of the model for a given time-step t.

$$RBA = \frac{1}{n}\sum_{t=1}^{n}\frac{(y_t - \hat{y}_{t\,baseline})^2 - (y_t - \hat{y}_{t\,model})^2}{(y_t - \hat{y}_{t\,baseline})^2}$$

5.1 Boundary-Aware architectures

Consider learning a PDE of the form $u_t = f(u)$. The simplest approximation of $u_t = f(u)$ is to discretize the time derivative u_t by $\frac{u_{n+1}-u_n}{\triangle t}$ and approximate the right hand side by $f(u_n)$. This leads to the discrete forward Euler scheme:

$$u_{n+1} = u_n + \triangle t f(u_n)$$

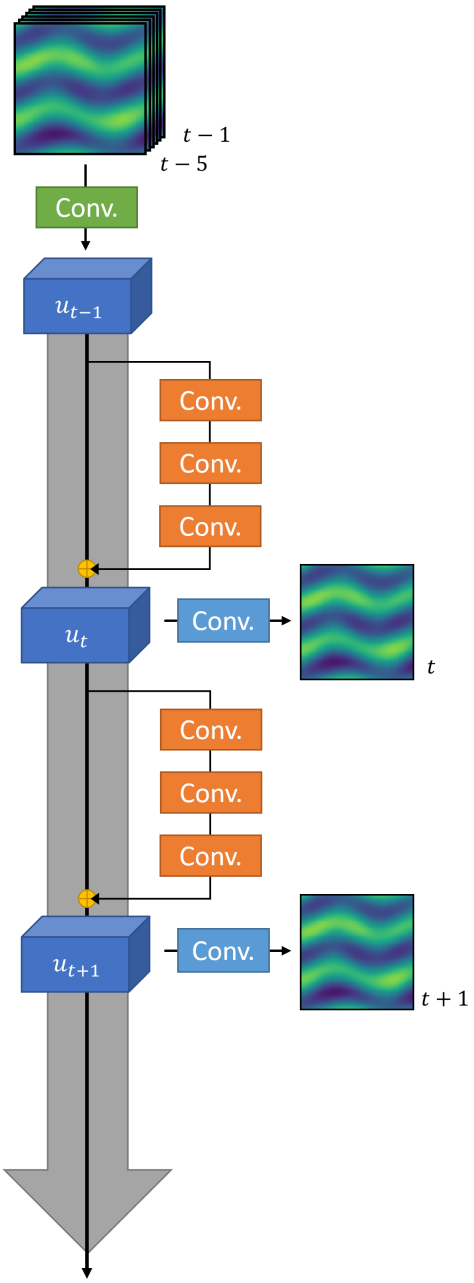

Figure 5: Illustration of the single-scale forward Euler architecture. The encoder $\phi(x)$ (shown in green) maps the input into the feature space. Three convolutional layers (shown in orange) are used to approximate the feature space dynamics $f(u_t)$ and summed with u_t (shown in yellow) to give u_{t+1} as outlined in equation 5.1. At each step, the feature vector u_t is decoded by $\theta(u_t)$ (as shown in light blue) predicting x_{t+1}.

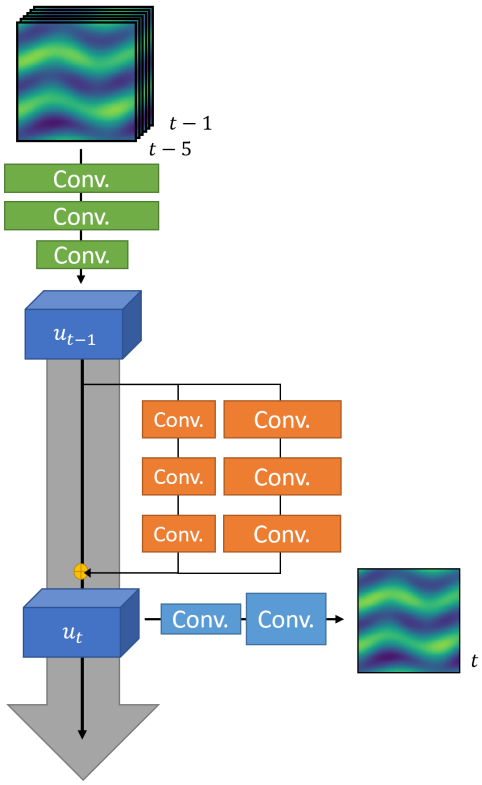

Figure 6: Illustration of the multi-scale forward Euler architecture. The dynamics $f(u_t)$ are enriched with filters at different scales (shown in orange). Additionally, the encoder $\phi(x)$ reduces the dimensionality of the input image by a factor of 4 in the encoding step $\phi(x)$ to increase the receptive field of the dynamics update calculation, $f(u_t)$ allowing for lower frequency features to be captured by the network.

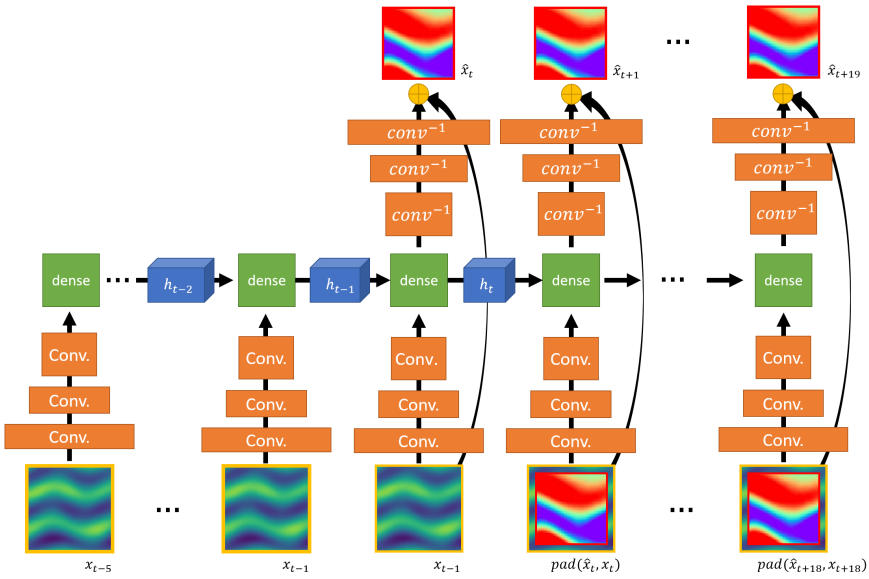

Figure 7: Illustration of the boundary-aware architecture modification. By including boundary conditions as padding of previous predicted flow, the dynamics are grounded by the surrounding vorticity resulting in significant reduction in error across the prediction trajectory. This architecture differs slightly from that of the forward-Euler based architectures in that instead of learning the dynamics of the feature space state representation $f(u_t)$, this mechanism of boundary constraining requires we learn the dynamics $f(x_t) = x_{t+1}$ directly.

where $f(u_n)$ is approximated by a deep neural network. In the previous forward-Euler formulations , illustrated in figures 5 and 6, instead of learning the dynamics of the system directly, we use a learned feature representation u_n composed of linear convolutions of the input history. In contrast, for boundary-aware architectures, in order to effectively incorporate the boundary state, we learn directly the vorticity dynamics $f(x_t) = x_{t+1}$

In the newly introduced boundary-aware architecture shown in figure 7 we utilize a padded activation to learn region dynamics, predicting a 50x50 patch padded to 64x64 to incorporate local state. The down-sampling network uses a series of four convolutions, each with a stride of 2 and a filter-size of 4 doubling the number of channels at each stage. This results in a 8x8x128 vector which we concatenate with the previous hidden state h_t and use a dense, fully-connected layer outputting h_{t+1} and the activation used as input to the de-convolution stage. We then up-scale the modified activation using mirrored de-convolutions corresponding with the original down-sampling convolutions. Finally, we utilize a skip-layer as in our forward-Euler architecture, adding the learned dynamics $f(x_t)$ to the previous state, x_t giving us x_{t+1} This model is trained over 20 consecutive time-steps using 5 frames of history to generate the initial hidden state of 64 features using the MSE loss, $L_{mse} = \frac{1}{N}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2$. Training is conducted over 40000 batches using the Adam optimizer.

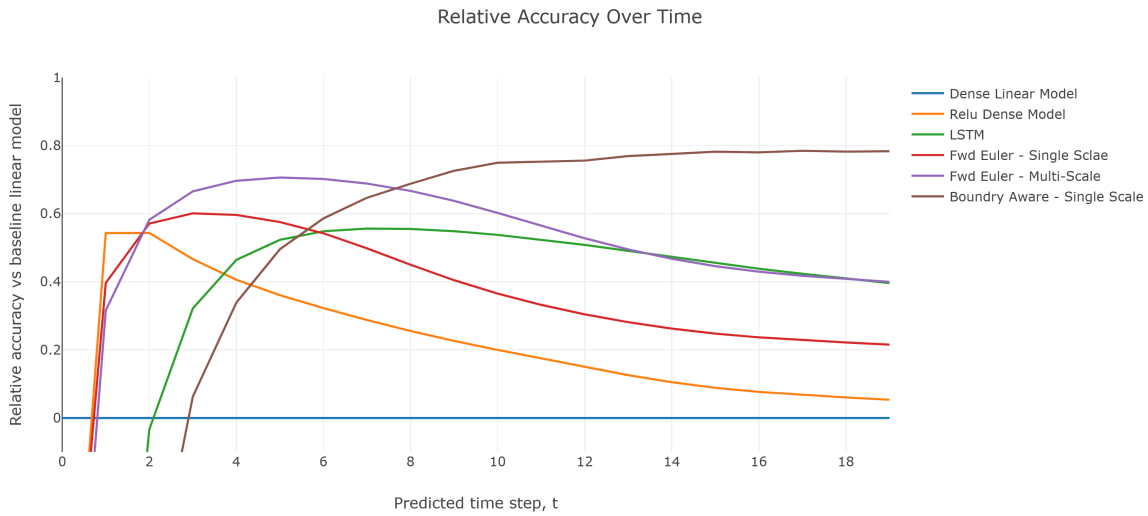

Figure 8: Accuracy of models relative to linear baseline (6.25 million parameters). Relative baseline accuracy, *RBA*, of 1.0 represents perfect prediction for a given time-step, negative values indicate worse than baseline accuracy.