Learning latent field dynamics of PDEs

Dmitrii Kochkov[*]; Alvaro Sanchez-Gonzalez[**]; Jamie Smith[*]; Tobias Pfaff[**]; Peter Battaglia[**]; Michael P. Brenner[*]

* Google Research

**DeepMind

Abstract

We present a new approach to learning surrogate models for simulation of complex physical systems described by nonlinear partial differential equations. It aims to capture three features of PDEs: locality, time continuity and formation of elementary patterns in the solution by learning a local, low dimensional latent representation and corresponding time evolution. We show that our method achieves top performance and competitive inference speed compared to baseline methods while operating with a 4-times more compact representation. Since the models learn local representations of the solution, they generalize to different system sizes that feature qualitatively different behavior without retraining.

Motivation

Mathematical models based on PDEs are among the most successful models of natural phenomena.
We consider systems described by PDEs of the form:

$$\frac{\partial \mathbf{u}(x,t)}{\partial t} = f(\mathbf{u}, \nabla\mathbf{u}, \ldots, \nabla^n\mathbf{u})$$

where $\mathbf{u}(x,t)$ represents the configuration of the system at position x and time t

Traditional numerical methods simulate the dynamics via iterative updates, computing the evolving state at every time step. For this process to be accurate, smallest features of the solution must be fully resolved.

This requirement leads to 2 challenges:

- Computational degrees of freedom far exceed the dimension of the solution manifold
- Fine discretizations affect the time step that can be taken at each iteration without introducing large errors

Both result in an increased computational cost.

Here we formulate a class of ML models that alleviate some of these obstacles by using learned representations of the solution manifold that incorporate the following priors of PDEs:

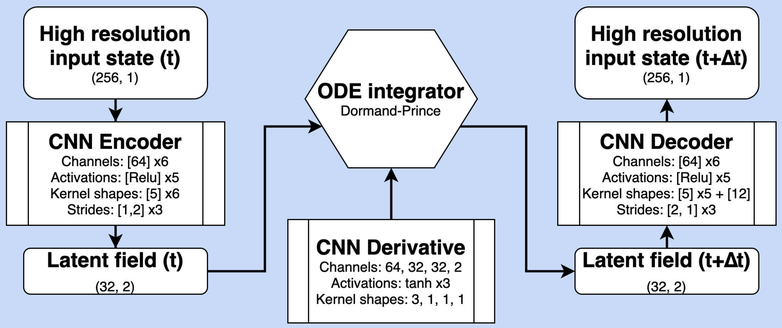
1. Locality of laws of nature
2. Time continuity
3. Existence of a low dimensional solution manifold

Our model efficiently parametrizes solution manifold that is easier to evolve in time, while locality enables generalization to different simulation settings.

Model architecture

We call our models Latent Field Model (LFM). They consists of three components: an encoder E, a decoder D, and a derivative model F. Together, they define the relationship between the physical states of the system $u(t)$, corresponding latent states $z(t)$ and their time derivatives $\dot{z}(t)$

1. Encoder:
   a. Maps the input state $u(t)$ to a low dimensional latent field representation $z(t)$
   b. Learns efficient parametrization of the solution manifold; preserves spatial structure
2. Time derivative model:
   a. Predicts time derivative in the latent space $\dot{z}(t)$ for the given configuration $z(t)$
   b. Learned dynamics is local in space and continuous in time
3. Decoder:
   a. Reconstructs the high resolution state $u(t)$ from the latent field representation $z(t)$
   b. Enables real space inference and training



Training and inference

During training we learn low-dimensional representation of the solution manifold.

For a perfect model we would expect 3 correspondences:

1. Physical state $u(t)$ and latent field $z(t)$ are bijective with Encoder/Decoder as maps
2. Encoded true derivative $\dot{u}(t)$ should correspond to the predicted derivative of latents
3. Decoded latent derivative should exactly match the true derivative $\dot{u}(t)$

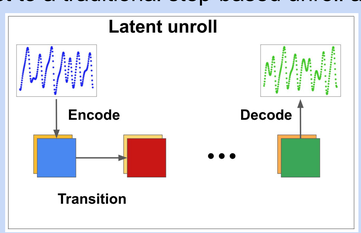
$$\mathcal{D}(\mathcal{E}(\mathbf{u}))(t) = \mathbf{u}(t) \quad J_{\mathcal{E}}(\mathbf{u})\dot{\mathbf{u}}(t) = \dot{\mathbf{z}}(t) \quad J_{\mathcal{D}}(\mathbf{z})\dot{\mathbf{z}}(t) = \dot{\mathbf{u}}(t)$$

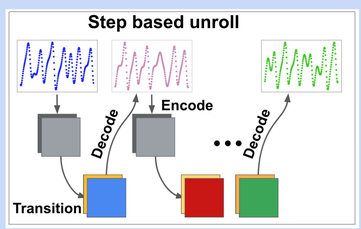
These three equations form the basis for our optimization procedure:

$$L = L_1 + L_2 + L_3$$

$$L_1 = \mathrm{MSE}(\mathbf{u}, \mathcal{D}(\mathcal{E}((\mathbf{u})))) \quad L_2 = \mathrm{MSE}(\dot{\mathbf{z}}, J_{\mathcal{E}}(\mathbf{u})\dot{\mathbf{u}}) \quad L_3 = \mathrm{MSE}(\dot{\mathbf{u}}, J_{\mathcal{D}}(\mathbf{z})\dot{\mathbf{z}})$$

During inference, we aim to perform the time evolution entirely in the latent space, in contrast to a traditional step-based unroll as shown below:





Results

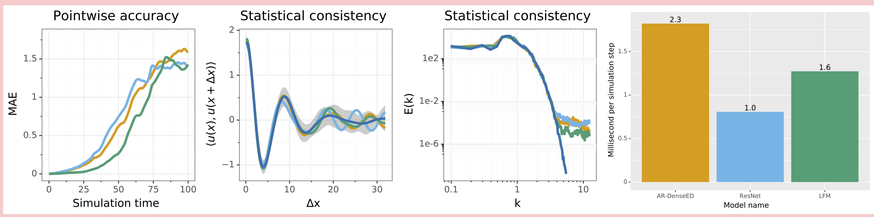
We train and evaluate the performance of latent field models and baselines on the Kuramoto-Sivashinsky equation: $\partial_t \mathbf{u} + \mathbf{u}\partial_x \mathbf{u} + \partial_x^2 \mathbf{u} + \partial_x^4 \mathbf{u} = 0$

This system is chaotic and features a low dimensional solution manifold;
All models are trained on a dataset of 90 trajectories in a developed chaotic regime

Evaluation measurements:
a. Pointwise accuracy tests short-term prediction performance
b-c. Statistical consistency tests preservation of internal structure at longer times
d. Performance compares the time it takes to obtain solution at a future time

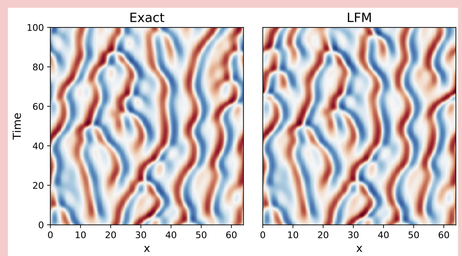
For pointwise accuracy we use (MAE), statistical consistency is probed using energy spectrum E(k) and velocity 2-point correlation

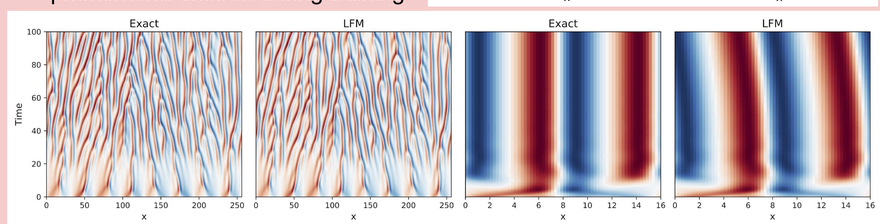
Our model achieves top accuracy, high statistical consistency and is fast to evaluate. We also find that learned dynamics is less *stiff* than the original equation: requiring ~100x fewer time-steps compared to direct integration with the same tolerance settings. It also uses 8x less memory than baselines operating at full resolution.



Generalization across system sizes:
Because our representation and learned dynamics are local, our model generalizes well to new system sizes.

- Our model correctly predicts the stationary-chaotic transition
- Correctly captures transient phenomena unseen during training





Future directions

- Explicit regularization of stiffness of equations in the latent space
- Learn representations that prserve invariant structures by construction?
- Stochastic formulation for higher level of coarsening
- Applications to more complex domains