Forum Terratec
AI in scientific computing
Deep Learning for physical Processes

Patrick GALLINARI
Professor
&
Distinguished Research Scientist

patrick.gallinari@lip6.fr, p.gallinari@criteo.com

SORBONNE UNIVERSITÉ

criteo AI Lab

Presentation outline

- Deep Learning « recent » breakthroughs
  - Virtual vs real physical world
- Deep Learning for dynamical process – the interplay of machine learning and numerical simulation - illustrations
  - Learning PDE with NNs
  - Solving PDEs with NNs
  - Combining Physic models and NNs

Deep Learning « recent » breakthroughs

- Deep Learning SOTA for different application domains
    - Vision

Image classification and object detection (YOLO, Redmon 2016)

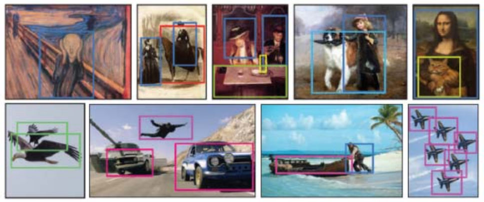
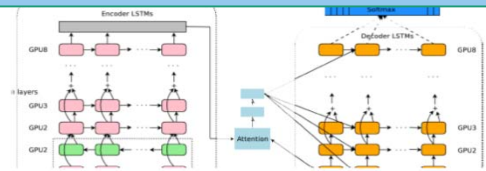


Figure 6: Qualitative Results. YOLO running on artwork and natural images. It is mostly accurate although it does think one person in an image is an airplane.

Scene segmentation (Segnet, Badrinarayanan 2017)

- Natural Language Processing

Google Neural Machine Translation 2016 – (Wu et al 2016) Over 100 languages



- Games

Figure 1: Screen shots from five Atari 2600 Games: (*Left-to-right*) Pong, Breakout, Space Invaders, Seaquest, Beam Rider

What about the real – physical world? Challenges

- Machine Learning will induce major changes in many industrial domains
    - Automatic vehicles, Healthcare, Security, etc
- What about domains where process models are based on engineering frameworks developed over many years?
    - How to develop the interplay between engineering culture and data intensive methods?
        - Aeronautics, Energy, Manufacturing,Transport,etc.
        - Climate, Geophysics, Universe, etc
        - IRT SystemX@Saclay hosts projects on this very topic
            - e.g. program IA2 - *Intelligence Artificielle et Ingénierie Augmentée*

Focus: Deep Learning for Dynamical Processes - Coupling Deep Learning and Partial Differential Equations

Deep Learning and Dynamical Processes
Motivation

- Modeling complex physical systems: 2 paradigms
  - Background knowledge on the physical phenomenon
    - Differential equations for dynamic systems/ Numerical Analysis
  - Data coming from observations or simulation
    - Agnostic machine learning
- Challenges for numerical analysis
  - Reduce the cost of simulations through reduced models exploiting the approximation power of DNNs – model reduction
  - Fast development of models when data are available
  - Solve problems difficult for classical methods e.g. high dimensions, complex dynamics
  - Integrate data from observations into the modeling process

Challenges for machine learning

- Learn complex dynamics from scratch
- Can statistics learn physical principles or discover underlying physical laws?
- Incorporate physical knowledge into statistical models
- Benefits from theory and methods from numerical analysis

Differential equations and NNs: recent trends and contributions from different communities

- From ODE to NNs
  - Similarity of Deep NNs and ODE numerical schemes
  - ODE numerical schemes for designing and training NNs
  - Training NNs with ODE solvers
  - Analyzing properties of NNs, e.g. stability
- From NNs to PDEs (in red examples used for the talk)
  - **Learning PDEs from data**
  - **Solving PDE with NNs - Reduced models**
  - Dealing with partially observed data
  - **Combining physic models and NNs**
- Interest from several application domains

Learning Differential Equations from Data (Rudy et al. 2017) - Sparse regression

- Collect observations
- Consider a library of terms, e.g. system state, derivatives, …
- Use sparse linear regression to fit observations on library terms

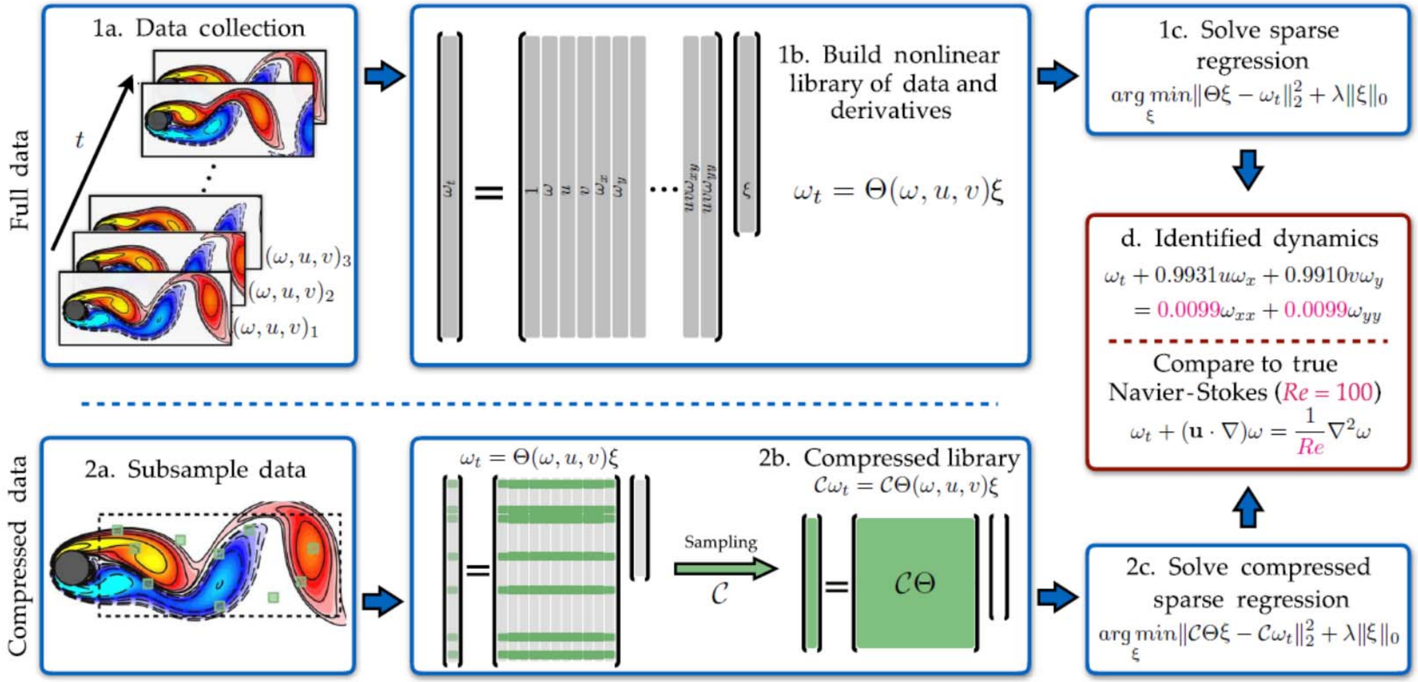


Fig. 1. Steps in the PDE functional identification of nonlinear dynamics (PDE-FIND) algorithm, applied to infer the Navier-Stokes equations from data. (1a) Data are collected as snapshots of a solution to a PDE. (1b) Numerical derivatives are taken, and data are compiled into a large matrix $\mathbf{\Theta}$, incorporating candidate terms for the PDE. (1c) Sparse regressions are used to identify active terms in the PDE. (2a) For large data sets, sparse sampling may be used to reduce the size of the problem. (2b) Subsampling the data set is equivalent to taking a subset of rows from the linear system in Eq. 2. (2c) An identical sparse regression problem is formed but with fewer rows. (d) Active terms in ξ are synthesized into a PDE.

Learning Differential Equations from Data
PDE-NET (Long et al. 2018) PDE-NET 2.0 (Long et al. 2019)

- Inspired from classical NN architectures, e.g. ResNet
  - Implement the F dynamics with a specific NN

δt −block

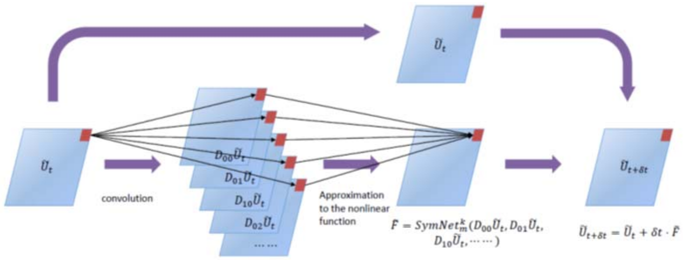


Figure 1: The schematic diagram of a δt-block.

Multiple steps

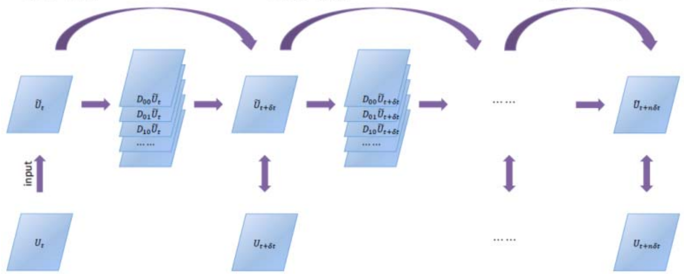


Figure 2: The schematic diagram of the PDE-Net 2.0.

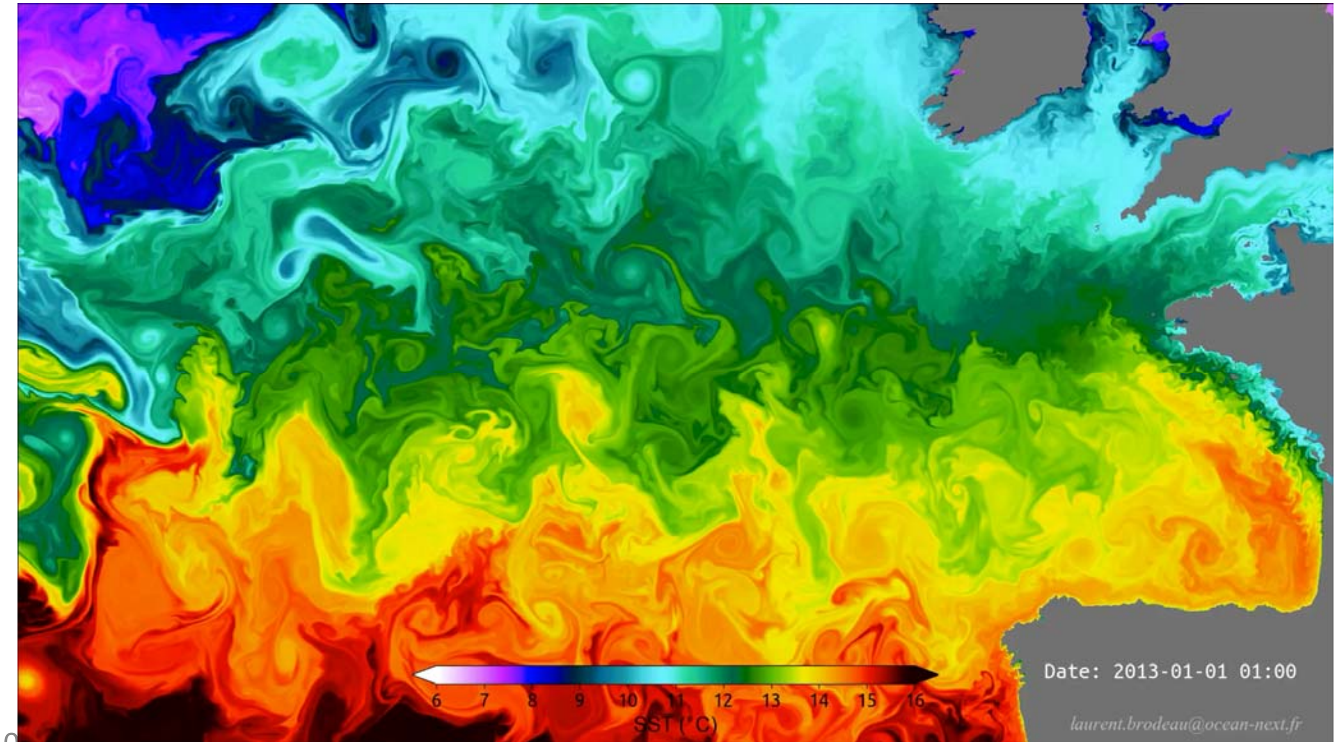
- Each block implements
  - Spatial component - Convolution filters
    - Learned convolutional filters approximate spatial differential operators
  - Temporal component - Skip connections (Euler like)
    - $x_{t+1} = x_t + f(x_t)$
  - Learns cross features between differential operators - learns the explicit form of the PDE

Solving PDEs with NNs – Model Reduction
example: Deep Galerkin Method (Sirignano et al. 2018)

- Objective
  - Solve large dimensional PDEs
  - The form of the PDE is known, solve it when classical methods fail
- Principle
  - Consider a parabolic equation with d spatial dimensions
    - $$\begin{cases} \frac{\partial u(t,x)}{\partial t} + Lu(t,x) = 0, \quad (t,x) \in [0,T] \mathrm{x} \Omega \\ u(t=0,x) = u_0(x) \\ u(t,x) = g(t,x) \quad x \in \partial\Omega \end{cases}$$
  - Approximate u with a Neural Network $f(t,x;\theta)$
  - Define an appropriate loss function
    - $J(f) = \left\| \frac{\partial f(t,x;\theta)}{\partial t} + Lf(t,x;\theta) \right\|^2 + \ \ldots.$
  - Solve $J(f)$ by sampling in the spatio temporal domain + gradient descent
  - f represent the solution over the entire spatio-tempral domain

Combining PDEs and NNs
Deep Learning for Physical Processes: Incorporating Prior Scientific Knowledge (de Bezenac et al. 2018)

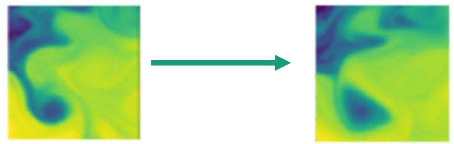
- Example: Sea Surface Temperature Prediction - SST (< 1 meter deep) on Atlantic ocean
    - Data: satellite imagery (IR)
    - Use cases: Weather prediction, anomaly detection, component of climate models
-



Combining PDEs and NNs - Deep Learning for Physical Processes: Incorporating Prior Scientific Knowledge (de Bezenac et al. 2018)

- Describe transport of I through **advection** and **diffusion**

$$\frac{\partial I}{\partial t} + (w.\nabla)I = D\nabla^2 I$$



  - I: quantity of interest (Temperature Image)
  - $w = \frac{\Delta x}{\Delta t}$ motion vector, D diffusion coefficient

- There exists a closed form solution
  - $I_{t+\Delta t}(x) = (k * I_{\mathrm{t}})(x - w(x))$
  - $I_{t+\Delta t}(x)$can be obtained from I_{t} through a convolution with kernel k (pdf of a Normal distribution: $k(x - w, y) = N(y|x - w, 2D\Delta t)$)

- If we knew the motion vector w and the diffusion coefficient D we could calculate $I_{t+\Delta t}(x)$ from I_{t}
  - **w and D unknown**
  - **-> Learn w and D**

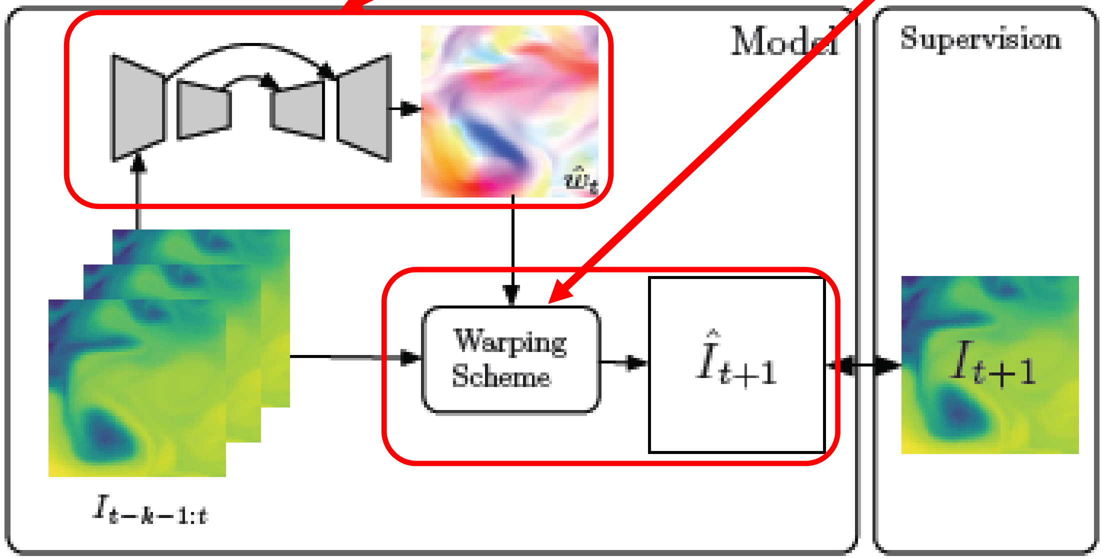
Combining PDEs and NNs - Deep Learning for Physical Processes: Incorporating Prior Scientific Knowledge (de Bezenac et al. 2018)

- Prediction Model: Objective: predict I_{t+1} from past $I_t, I_{t-1}, \ldots$ 2 components:

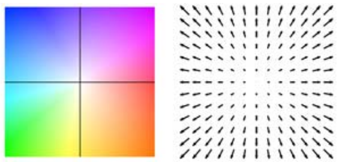
Convolution- Deconvolution NN for estimating motion vector w_t

Warping Scheme Implements discretized A-D solution

Past Images



Target image



Color: orientation
Intensity: flow intensity

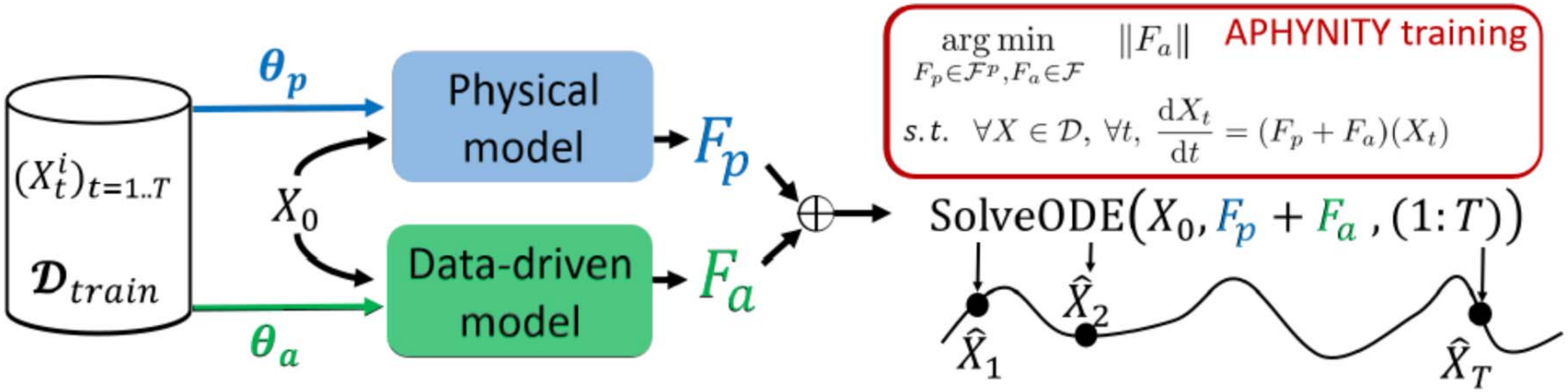
- End to End learning using only I_{t+1} supervision
- Stochastic gradient optimization

Combining PDEs and NNs
Aphynity: combine NN and differential solvers (Le Guen et al. 2020)

- Context
  - Data driven models are insufficient to predict complex physical dynamics, e.g. extrapolating is still an open problem
  - Physical models extrapolate well if they adequately describe the dynamics
    - But: backround may not be available or only partially, unknown external factors,…
- Objective
  - Hyp: Incomplete background knowledge is available, e.g. PDE
  - Provide a principled framework to make model based and data based framework cooperate
    - Identify correctly the physical parameters
    - The NN component should learn to describe the information that cannot be captured by the physics,

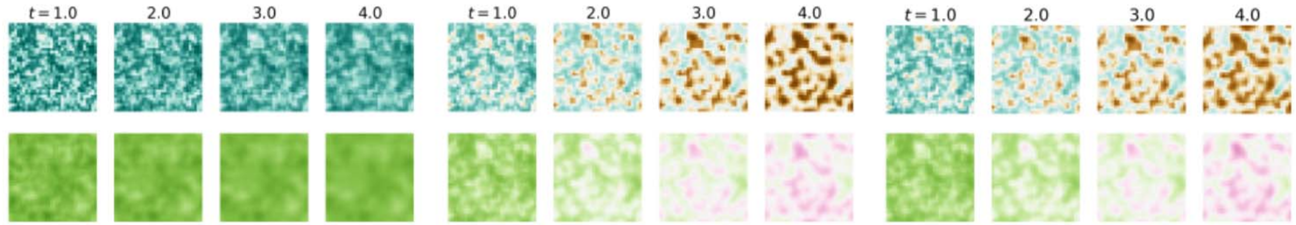
Aphynity: combine NN and differential solvers (Le Guen et al. 2020)

- We provide
  - A principled framework for the decomposition
  - Under mild hypothesis it comes with existence and unicity of the solution
    - Guaranties that the parameters of the physical system will be identified while approximating as best as possible the dynamics



Aphynity: combine NN and differential solvers (Le Guen et al. 2020) - Examples

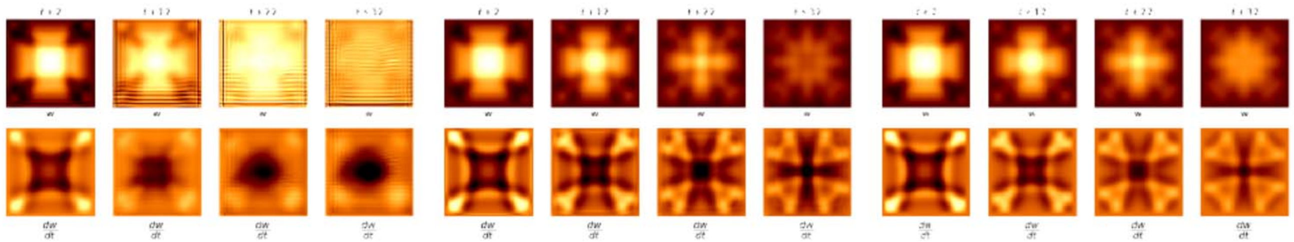
- Reaction Diffusion equations



(a) Param PDE (a, b), diffusion-only (b) APHYNITY Param PDE (a, b) (c) Ground truth simulation

Figure 2: Comparison of predictions of two components u (top) and v (bottom) of the reaction-diffusion system. Note that $t = 4$ is largely beyond the dataset horizon ($t = 2.5$).

- Damped wave equation



(a) Neural ODE (b) APHYNITY Param PDE (c) (c) Ground truth simulation

Figure 3: Comparison between the prediction of APHYNITY when c is estimated and Neural ODE for the damped wave equation. Note that $t + 32$, last column for (a, b, c) is already beyond the training time horizon ($t + 25$), showing the consistency of APHYNITY method.

Conclusion

- Several emerging topics at the crossroad of NNs and numerical modeling systems
- Open several perspectives both for statistical machine learning and for physical modeling

  - Cross fertilization of model based approaches and data driven approaches
  - New models for describing complex dynamics exploiting the large amounts of observation data
  - New perspectives for modeling/ training neural networks
    - e.g. as dynamical systems

References used in the presentation

- Ayed I., de Bezenac E., Pajot A., Brajard J. , Gallinari P. , (2019), Learning Dynamical Systems from Partial Observations, ICASSP 2020
- de Bezenac, E., Pajot, A., & Gallinari, P. (2018). Deep Learning For Physical Processes: Incorporating Prior Scientific Knowledge. In *ICLR*.
- Le Guen V., Yin Y., Dona J., Ayed I., de Bezenac E., Thome N., Gallinari P., Augmenting Physical Models with Deep Networks for Complex Dynamics Forecasting, https://arxiv.org/abs/2010.04456, 2020
- Long, Z., Lu, Y., & Dong, B. (2018). PDE-Net 2.0: Learning PDEs from Data with A Numeric-Symbolic Hybrid Deep Network. https://doi.org/arXiv:1812.04426
- Rudy, S. H., Brunton, S. L., Proctor, J. L., & Kutz, J. N. (2017). Data-driven discovery of partial differential equations. *SCIENCE ADVANCES*, *3* no. *4*(April).
- SIRIGNANO, J. AND SPILIOPOULOS, K. 2018. DGM: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics 375*, Dms 1550918, 1339–1364.