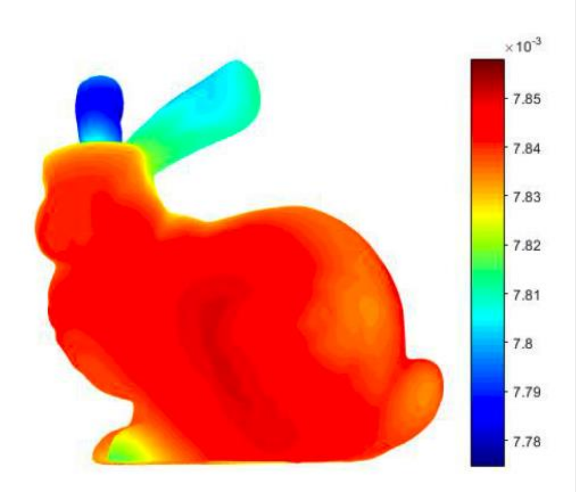
PINNs Tutorial

- Introduction to PINNs
  - What are PINNs?
  - Considerations when designing PINNs
- Google Colab Exercises
  - PINNs to estimate cerebral blood flow
  - PINNs to model cardiac electrophysiology

- Recent developments in PINNs
  - Problems with PINNs
  - How to address them

Some Recent Developments in Physics-Informed Neural Networks



Annie Ching-En Chiu & Marta Varela

ching-en.chiu18@imperial.ac.uk

marta.varela@imperial.ac.uk

IMPERIAL

Problems we have discussed

- Spectral bias
  - Can use Fourier inputs
- Vanishing/exploding gradients
  - Input and output normalisation
- Difficult to choose loss term weights and learning rate
  - Learning rate annealing

Some more PINN problems…

A. Difficulty converging
   – Often require morose problem-specific hyperparameter tuning
B. Problems finding solutions over long time ranges
C. Difficult to adapt to complicated geometries
D. Physics and data learning are not separable
   – Need to retrain the entire PINN when new data is available
   – Difficult to combine data from different experiments
E. Can learn model parameters, but not model formulation

Some more PINN problems…

A. Difficulty converging
   - Often require morose problem-specific hyperparameter tuning

B. Problems finding solutions over long time ranges

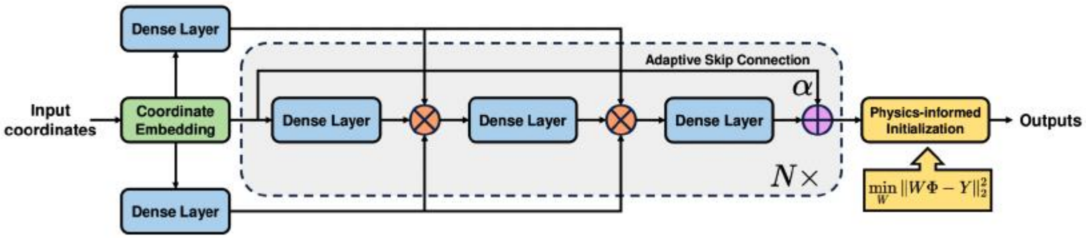
C. Difficult to adapt to complicated geometries

D. Physics and data learning are not separable
   - Need to retrain the entire PINN when new data is available
   - Difficult to combine data from different experiments

E. Can learn model parameters, but not model formulation

Converging is difficult

- Why?
  - Variance of the derivatives is very low, especially for deep NNs.
  - Because we use a residual formulation, there is an initial bias towards the trivial solution in homogeneous DEs.
- How to address this?
  - Include an adaptive skip-connection (PirateNets[1])



  - α, the skip connection weights, are learnable and initialised to 0. The effective depth of the NN increases during training.

Some more PINN problems…

A. Difficulty converging
   – Often require morose problem-specific hyperparameter tuning
B. Problems finding solutions over long time ranges
C. Difficult to adapt to complicated geometries
D. Physics and data learning are not separable
   – Need to retrain the entire PINN when new data is available
   – Difficult to combine data from different experiments
E. Can learn model parameters, but not model formulation

Issues with Long Time Intervals

- PINNs trains across all time points simultaneously. This makes them susceptible to degenerating into local minima.
- Forcing sequential (causal) training mitigates this[1, 2]:
  - The weight for time i is inversely exponentially dependent on the cumulative loss for past times.
  - L(i) will not be minimised until the past time points have.

$$\mathcal{L}_r(\boldsymbol{\theta}) = \frac{1}{N_t}\sum_{i=1}^{N_t} \exp\left(-\epsilon \sum_{k=1}^{i-1} \mathcal{L}_r(t_k, \boldsymbol{\theta})\right) \mathcal{L}_r(t_i, \boldsymbol{\theta}).$$

- Another strategy involves dividing the time domain into small intervals and training PINNs at different time scales and sequentially[3].
  - The solution at the end of interval i will be the initial condition for i+1.

Some more PINN problems…

A. Difficulty converging
   - Often require morose problem-specific hyperparameter tuning
B. Problems finding solutions over long time ranges
C. Difficult to adapt to complicated geometries
D. Physics and data learning are not separable
   - Need to retrain the entire PINN when new data is available
   - Difficult to combine data from different experiments
E. Can learn model parameters, but not model formulation

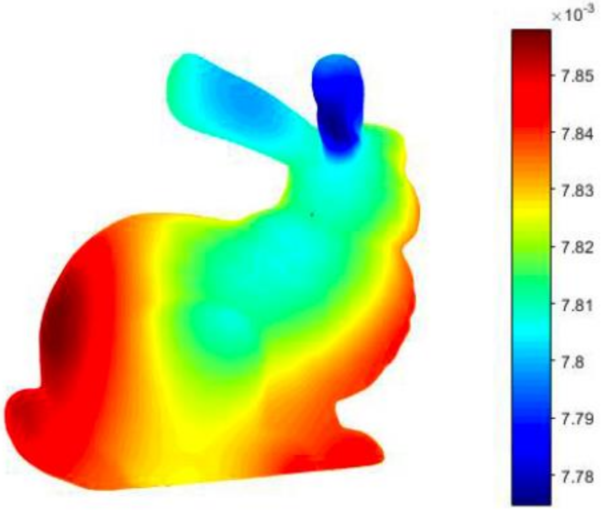
Complex Geometries

- Spatial inputs are typically inputted in Cartesian coordinates, but derivatives are wrt geometry-specific metric.
- AutoDiff cannot be used in these curved manifolds. Other options: diffusion maps, radial-basis function-generated finite differences, generalised moving least squares.[4]
- Alternatively, reformulate the PDE,

e.g. $\Delta_\Gamma u + u = f$ on manifold G as

$$\begin{cases} \Delta u + u = f \\ \langle \nabla u, \boldsymbol{n} \rangle = 0 \end{cases}$$

an extra geometric loss function term



Some more PINN problems…

A. Difficulty converging
   – Often require morose problem-specific hyperparameter tuning
B. Problems finding solutions over long time ranges
C. Difficult to adapt to complicated geometries
D. Physics and data learning are not separable
   – Need to retrain the entire PINN when new data is available
   – Difficult to combine data from different experiments
E. Can learn model parameters, but not model formulation

Transfer Learning for PINNs

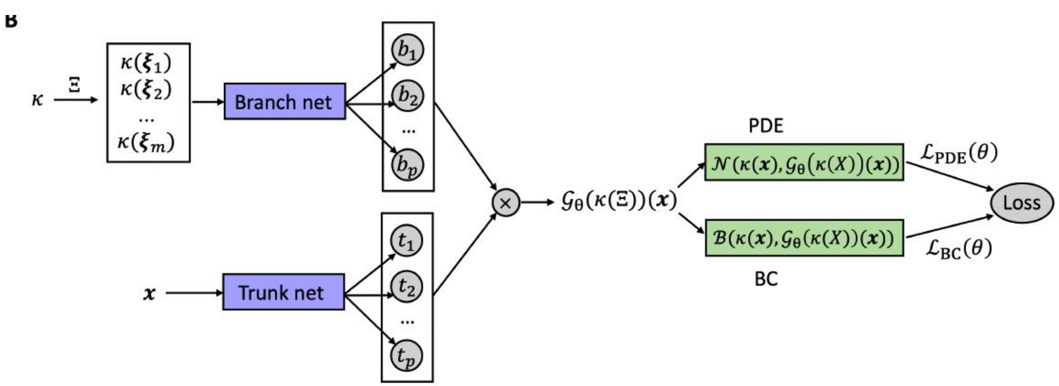
- One-Shot PINNs[6] separates the FCNN into:
  - non-linear hidden layers: learn the differential operator for an ODE/PDE family D_H
  - a final linear set of weights: linearly combine the solution modes W_{out}
- Bundle-trains D_H for a DE family using numerical solver solutions with different parameter values, forcing terms
- At inference, W_{out} can be calculated by performing matrix operations on D_H

$$W_{\text{out}} = (\hat{D}_H^T \hat{D}_H + \bar{D}_H^T \bar{D}_H)^{-1} (D_H^T f'(t) + \bar{D}_H^T u'_{\text{ic}}).$$

IMPERIAL

Learning the Operator

- Different paradigm: DeepONets[7]
  - Learns an operator between infinite-dimensional function spaces (e.g., integration)
  - Trained on numerical solutions of the operator[7] or using physics-based loss (PI-DeepONets[8])
- Example: $\mathcal{N}(\kappa, u) = -\mathrm{div}_g(\kappa \,\mathrm{grad}_g u) + cu - f$
  - Learn the solution u(x) for any k(ξ) subject to boundary conditions B
  - From data, identify k using an optimisation algorithm.

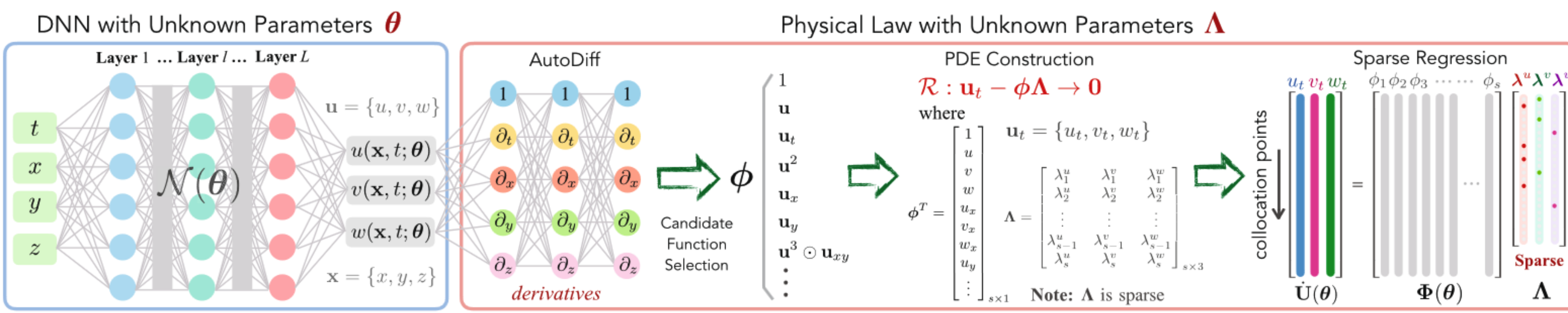


Some more PINN problems…

A. Difficulty converging
   – Often require morose problem-specific hyperparameter tuning
B. Problems finding solutions over long time ranges
C. Difficult to adapt to complicated geometries
D. Physics and data learning are not separable
   – Need to retrain the entire PINN when new data is available
   – Difficult to combine data from different experiments
E. Can learn model parameters, but not model formulation

Learning Model Terms

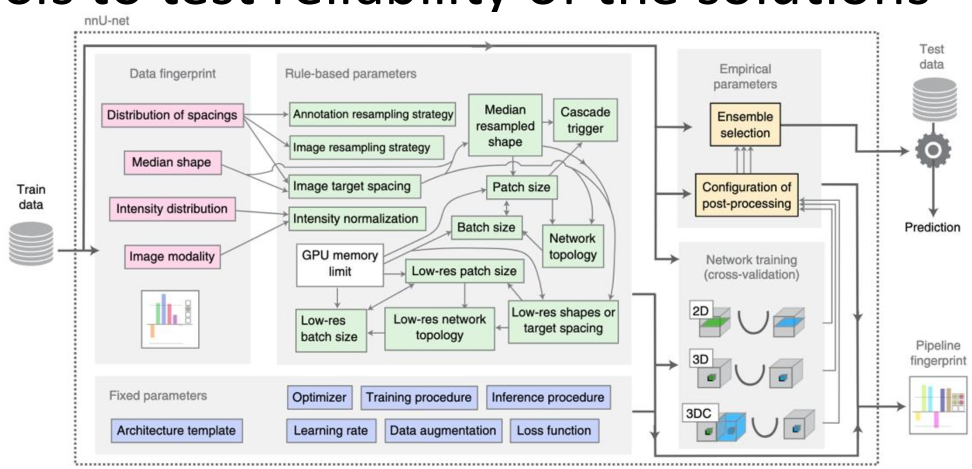
- Experimental data often suffers from systematic errors or does not follow mathematical model as closely as expected
- PINNs can learn equations from data
  - Physics-informed version of SinDy[9]



$$\underbrace{\mathcal{L}\left(\boldsymbol{\theta}, \boldsymbol{\Lambda}; \mathcal{D}_u, \mathcal{D}_c\right)}_{\text{total loss}} = \underbrace{\mathcal{L}_d\left(\boldsymbol{\theta}; \mathcal{D}_u\right)}_{\text{data loss}} + \alpha \underbrace{\mathcal{L}_p\left(\boldsymbol{\theta}, \boldsymbol{\Lambda}; \mathcal{D}_c\right)}_{\text{physics loss}} + \underbrace{\beta \|\boldsymbol{\Lambda}\|_0}_{\text{regularization}}$$

The Future of PINNs?

- Self-configuring PINNs ("nnU-Net[10] for PINNs")
  - Include best practices for data normalisation and hyperparameter tuning
  - Test alternative configurations to decide on optimal architecture
- Set of pre-trained models for most common equations in most common geometries ("HuggingFace[11] for PINNs")
- More flexible means of including heterogeneous data
- Tools to test reliability of the solutions



Automated (fixed, rule-based or empirical) configuration derived by distilling expert knowledge (more details in online methods)

References

1 Wang et al, PirateNets: Physics-informed Deep Learning with Residual Adaptive Networks, 2024, doi.org/10.48550/arXiv.2402.00326

2 Wang et al, Respecting causality for training physics-informed neural networks, 2024, doi.org/10.1016/j.cma.2024.116813

3 Meng et al, PPINN: Parareal physics-informed neural network for time-dependent PDEs, 2020, doi.org/10.1016/j.cma.2020.113250

4 Jiao et al, Solving forward and inverse PDE problems on unknown manifolds via physics-informed neural operators, 2024, doi.org/10.48550/arXiv.2407.05477

5 Fang et al, A Physics-Informed Neural Network Framework for PDEs on 3D Surfaces: Time Independent Problems, 2020, doi.org/10.1109/ACCESS.2019.2963390

6 Desai et al, One-Shot Transfer Learning of Physics-Informed Neural Networks, 2021, doi.org/10.48550/arXiv.2110.11286

7 Lu et al, Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators, 2021, doi.org/10.1038/s42256-021-00302-5

8 Wang et al, Learning the solution operator of parametric partial differential equations with physics-informed deeponets, 2021, doi.org/10.1126/sciadv.abi8605

9 Chen at al, Physics-informed learning of governing equations from scarce data, 2021, doi.org/10.1038/s41467-021-26434-1

10 Isensee et al, nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation, 2020, doi.org/10.1038/s41592-020-01008-z

11 https://huggingface.co/

IMPERIAL