Transfer Learning with Physics-Informed Neural Networks for Efficient Simulation of Branched Flows

Raphaël Pellegrin, Blake Bullwinkel, Marios Mattheakis, Pavlos Protopapas
Institute for Applied Computational Science (IACS), Harvard University

Introduction

Physics-Informed Neural Networks (PINNs) offer a promising approach to solving differential equations. We adopt a recently developed transfer learning approach for PINNs and introduce a multi-head model to efficiently obtain accurate solutions to nonlinear systems of DEs. We apply the method to simulate stochastic branched flows, a universal phenomenon in random wave dynamics. We compare the results achieved by feed forward and GAN-based PINNs on two physically relevant transfer learning tasks (initial condition TL and potential TL).

PINNs Background

The standard unsupervised neural network approach was introduced in [1] and can be used to solve differential equations of the form

$$\mathcal{L}\left(u(t, \mathbf{x})\right) = 0, \tag{1}$$

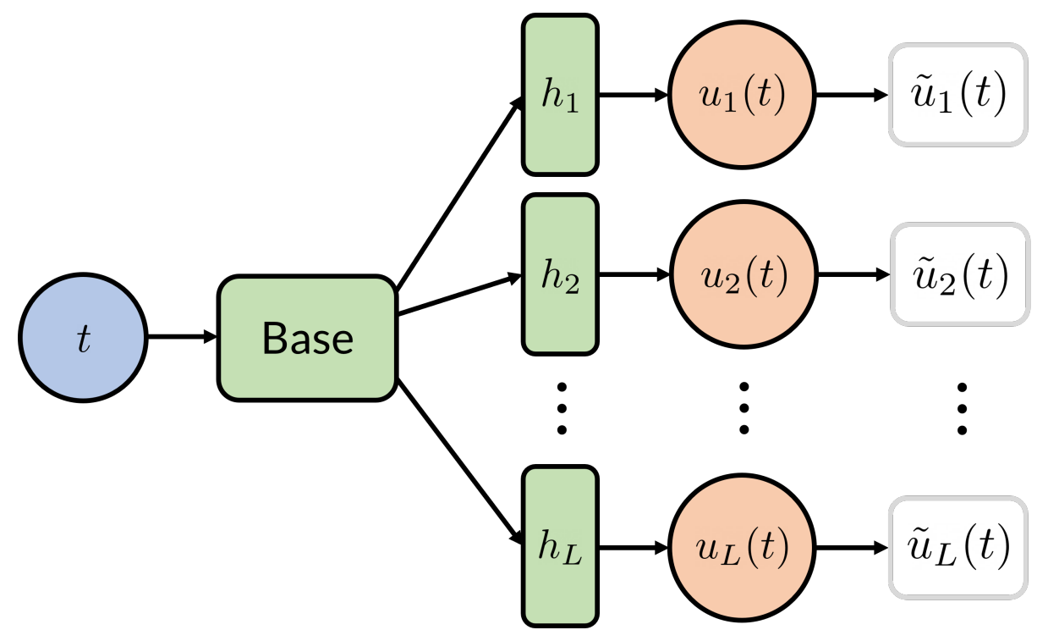
where $\mathbf{x} = (x_1, \dots, x_n)$, $u : \mathbb{R}^{n+1} \to \mathbb{R}$ and $\mathcal{L}$ is a differential operator. During training, we sample M training points $(t, \mathbf{x})$ from the domain of the equation D and use this vector as input to a FFNN, which outputs the neural solution $u_\theta(t, \mathbf{x})$. We re-parametrise this output into $\tilde{u}_\theta(t, \mathbf{x})$ to satisfy initial and boundary conditions exactly. Using automatic differentiation, we compute the derivatives of this output with respect to each of the independent variables and build the loss:

$$\frac{1}{M} \sum_{(t,\mathbf{x}) \in D} \mathcal{L}\left(\tilde{u}_\theta(t, \mathbf{x})\right)^2 . \tag{2}$$

Transfer Learning for PINNs

In [2], Desai et al. show that one-shot transfer learning can be used to obtain accurate solutions to linear systems of ODEs and PDEs, thereby eliminating the need to train the network from scratch for a new linear system. We build upon this work by proposing a method that can be applied to non-linear systems. This method consists of two phases. First, we train a base neural network with multiple output heads, solving the system for a range of different configurations (e.g., initial conditions or potentials). Second, we freeze the weights of the base network and fine-tune new linear heads on a new task.

Architecture



DEQGAN

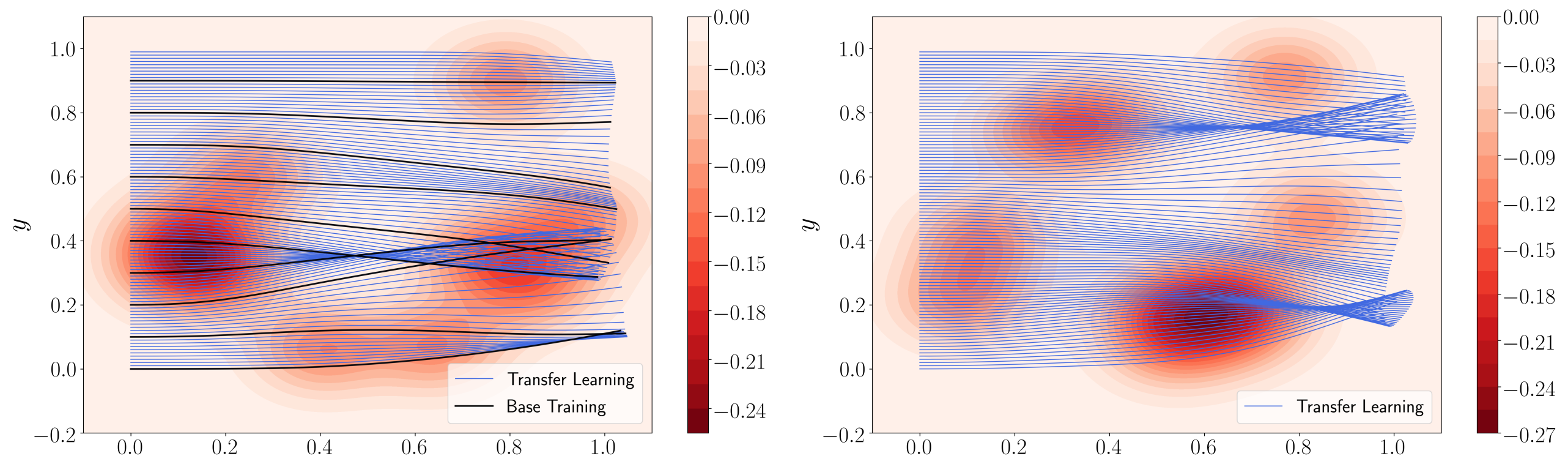
Bullwinkel et al. [2] noted that there is no theoretical reason to use the L_2 norm of the residuals (as in (2)) over any other loss function and proposed DEQGAN, which extends the FFNN method for PINNs to GANs and can be thought of as "learning the loss function." Rather than computing a loss over the equation residuals, DEQGAN labels these vectors "fake" data samples and zero-centered Gaussian noise as "real" data samples. As the discriminator gets better at classifying these samples, the generator $\tilde{u}_\theta$ is forced to propose solutions such that the equation residuals are increasingly indistinguishable from a vector of zeros, thereby approximating the solution to the differential equation.

Modeling Branched Flows

Stochastic branched flow is a universal wave phenomenon that occurs when waves propagate in random environments. Branching has been observed in tsunami waves, electromagnetic waves in gravitational fields and electronic flows in graphene [3]. We can model a two dimensional branched flow by considering a particle with position $\mathbf{x} = (x, y)$ and velocity $\mathbf{p} = (p_x, p_y)$, both functions of time t, traveling through a weak random potential $V(x, y)$. With the Hamiltonian $H(\mathbf{x}, \mathbf{p}) = ||\mathbf{p}||_2^2/2 + V(\mathbf{x})$ we obtain Hamilton's equations, given by the following system of ODEs

$$\begin{cases} \dot{x}(t) = p_x(t) \\ \dot{y}(t) = p_y(t) \\ \dot{p_x}(t) = -\dfrac{\partial V(x(t), y(t))}{\partial x} \\ \dot{p_y}(t) = -\dfrac{\partial V(x(t), y(t))}{\partial y}, \end{cases} \tag{3}$$

ICs TL and Potential TL



Particle trajectories through two weak potentials. The trajectories are generated by FFNN models. On the left plot, black lines correspond to the trajectories for 11 evenly spaced initial conditions that are used for base training on a first potential. Blue lines show the solutions obtained via transfer learning after freezing the base for 100 evenly spaced initial conditions on the same potential (left plot) and on a second potential (right plot). The color bars indicate the value of the potential.

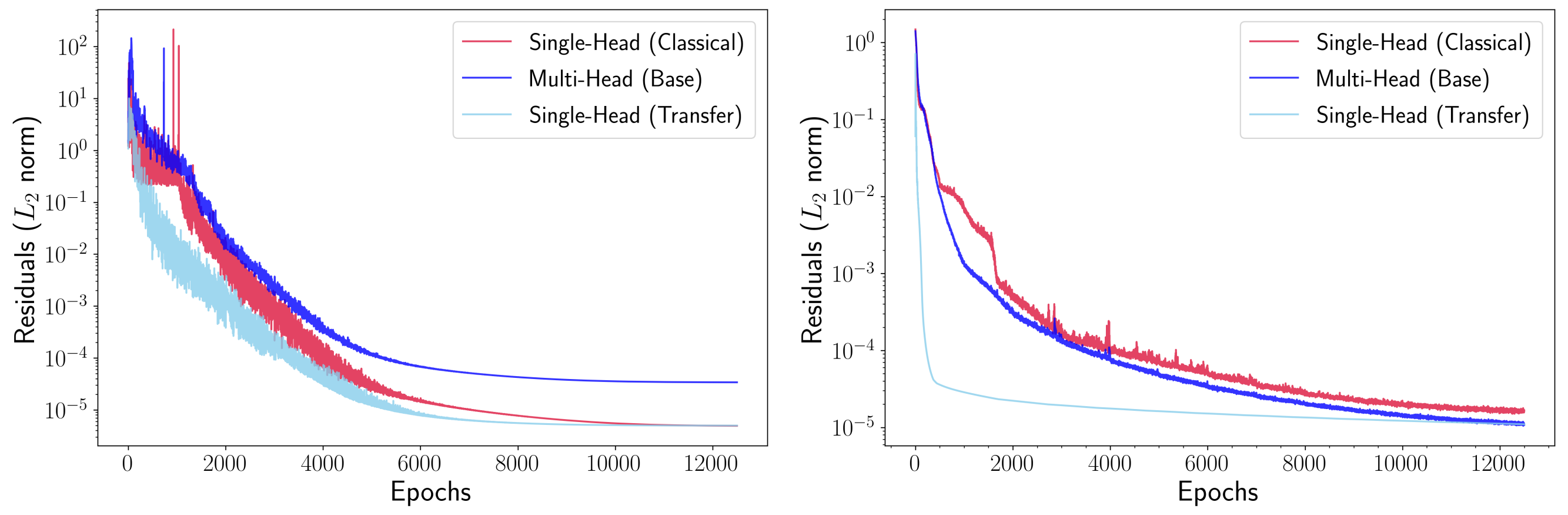
Details for Hamilton's Equations

For both the FFNN and DEQGAN models, we use networks with a base consisting of 5 hidden layers and 40 nodes. We then use L linear layers for the heads. Each head is responsible for the solution to one ray, i.e., one initial condition $\mathbf{z}_l(0) = (0, y_l(0), 1, 0)$, where $l = 1, \dots, L$, and has four outputs corresponding to x, y, p_x and p_y. For head l, we denote the outputs as $u_{l,1}$, $u_{l,2}$, $u_{l,3}$ and $u_{l,4}$. We use the initial value re-parameterization proposed by Mattheakis et al. in [4]

$$\tilde{u}_{l,i}(t) = \left[\mathbf{z}_l(0)\right]_i + \left(1 - e^{-t}\right) u_{l,i}(t), \tag{4}$$

$l = 1, \dots, L; i = 1, \dots, 4$ which forces the proposed solution to be exactly $\mathbf{z}_l(0)$ when $t = 0$ and decays this constraint exponentially in t.

Losses



Epochs vs. L_2 norm of the equation residuals for single-head (classical), multi-head (base training) and single-head (Initial Condition Transfer) runs using DEQGAN (left) and FFNN (right) models. The multi-head model losses are computed by averaging over the 11 heads.

References

[1] Lagaris, I. E., Likas, A., Papageorgiou, D. G. (1998). Neural network methods for boundary value problems defined in arbitrarily shaped domains. arXiv preprint cs/9812003.

[2] Bullwinkel, B., Randle, D., Protopapas, P., Sondak, D. (2022). Deqgan: Learning the loss function for pinns with generative adversarial networks. arXiv preprint arXiv:2209.07081.

[3] Mattheakis, M., Tsironis, G., Kaxiras, E. (2018). Emergence and dynamical properties of stochastic branching in the electronic flows of disordered dirac solids. EPL (Europhysics Letters), 122(2), 27003.

[4] Mattheakis, M., Protopapas, P., Sondak, D., Di Giovanni, M., Kaxiras, E. (2019). Physical symmetries embedded in neural networks. arXiv preprint arXiv:1904.08991.