Investigating the Ability of PINNs To Solve Burgers' PDE Near Finite-Time BlowUp

Dibyakanti Kumar
Barclays
Pune, India
dibyakantikumar1999@gmail.com

Dr. Anirbit Mukherjee
Department of Computer Science
University of Manchester
Manchester, UK
anirbit.mukherjee@manchester.ac.uk

Abstract

Physics Informed Neural Networks (PINNs) have been achieving ever newer feats of solving complicated PDEs numerically while offering an attractive trade-off between accuracy and speed of inference. A particularly challenging aspect of PDEs is that there exist simple PDEs which can evolve into singular solutions in finite time starting from smooth initial conditions. In recent times some striking experiments have suggested that PINNs might be good at even detecting such finite-time blow-ups. In this work, we embark on a program to investigate this stability of PINNs from a rigorous theoretical viewpoint. Firstly, we derive generalization bounds for PINNs for Burgers' PDE, in arbitrary dimensions, under conditions that allow for a finite-time blow-up. Then we demonstrate via experiments that our bounds are significantly correlated to the ℓ_2-distance of the neurally found surrogate from the true blow-up solution, when computed on sequences of PDEs that are getting increasingly close to a blow-up.

1 Introduction

Partial Differential Equations (PDEs) are used for modeling a large variety of physical processes from fluid dynamics to bacterial growth to quantum behaviour at the atomic scale. Quite recently, deep learning has emerged as a competitive way to solve PDEs numerically. For this work, we focus on the PINN formalism from Raissi et al. (2019). Many studies have demonstrated the success of this setup in simulating complex dynamical systems like the Navier-Stokes PDE (Arthurs & King, 2021; Wang et al., 2020; Eivazi et al., 2022) and many more.

Work of Mishra & Molinaro (2022); De Ryck et al. (2022) has provided the first-of-its-kind bounds on the generalization error of PINNs for approximating various standard PDEs, including the Navier-Stokes' PDE. Such bounds strongly motivate why minimization of the PDE residual at collocation points can be a meaningful way to solve the corresponding PDEs. However, the findings and analysis in Krishnapriyan et al. (2021); Wang et al. (2021a) point out that the training dynamics of PINNs can be unstable and failure cases can be found among even simple PDE setups.

An interesting possibility with various differential equations representing the time dynamics of some system is that their solution might have a finite-time blow-up. Blow-up is a phenomena where the solution becomes infinite at some points as t approaches a certain time $T < \infty$, while the solution is well-defined for all $0 < t < T$ i.e. $\sup_{x \in D} |\boldsymbol{u}(\boldsymbol{x}, t)| \to \infty \quad as \quad t \to T^-$. One can see simple

Machine Learning and the Physical Sciences Workshop, NeurIPS 2023.

examples of this fascinating phenomenon, for example, for the following ODE $\frac{du}{dt} = u^2,\ u(0) = u_0,\ u_0 > 0$ it's easy to see that it's solution blows-up at $t = \frac{1}{u_0}$. Wintner's theorem (Wintner, 1945) provides a sufficient condition for a very generic class of ODEs for the existence of a well-defined solution for them over the entire time-domain, in other words, the non-existence of a finite-time blowup. Further developments about obtaining such guarantees for ODEs can be seen in Cooke (1955) and Pazy (1983) (Theorem 3.3).

There is a long-standing quest in numerical methods of PDE solving to be able to determine the occurrence, location and nature of finite time blow-ups (Stuart & Floater, 1990). The seminal works Fujita (1966, 1969) pioneered systematic research into finite-time blow-ups of PDEs, focusing on classic semi-linear examples like the Frank-Kamenetsky equation which model combustion.

To the best of our knowledge, the behaviour of PINNs in the proximity of finite-time blow-up has not received adequate attention in prior work until very recent experimental studies like Wang et al. (2022d). We note that there are multiple real-world phenomena whose PDE models have finite-time blow-ups and these singularities are known to correspond to practically relevant processes – such as in chemotaxis models (Tanaka, 2023) and thermal-runoff models (Herrero & Velázquez, 1993). In light of the recent rise of methods for PDE solving by neural nets, it raises a curiosity whether the new methods, in particular PINNs, can be used to reliably solve PDEs near such blow-ups. While a general answer to this is outside the scope of this work, *we derive theoretical risk bounds for PINNs which are amenable to be tested against certain analytically describable finite-time blow-ups. Additionally, we give experiments to demonstrate that our bounds retain non-trivial insight even when tested in the proximity of such singularities.* An informal summary of our results could be found in Appendix B.3.

In the program we undertake here, we show how well the theoretical risk bounds can be obtained for PINNs and how much they correlate to their experimentally observed values - in certain blow-up situations. We focus on reduced models of fluid dynamics, i.e Burgers' PDE in one and two spatial dimensions. The choice of our test case is motivated by the fact that these PDE setups have analytic solutions with blow-up – as is necessary to do a controlled study of PINNs facing such a situation.

Related Works Firstly we note that the population risk bound for PINNs proven in Hu et al. (2022b) applies to linear second order PDE and hence it cannot be applied to our study since Burgers' PDE is not a linear PDE. Mishra & Molinaro (2022) derived generalization bounds for PINNs, that explicitly depend on the trained neural net. The ambit of their analysis includes the $1 + 1$-Burgers' PDE. However for testing against analytic blow-up solutions, we need such bounds at zero viscosity unlike what is considered therein, and most critically, unlike Mishra & Molinaro (2022) we keep track of the prediction error at the spatial boundary of the computational domain with respect to non-trivial functional constraints. De Ryck et al. (2022) derived a generalization bound for Navier-Stokes PDE. We note that, in contrast to the approach presented in De Ryck et al. (2022), our method does not rely on the assumption of periodicity in boundary conditions or divergencelessness of the true solution. This allows us to apply our bound to analytic cases of finite-time blow-ups for the $d + 1$-Burgers' PDE.

Notation In the subsequent section we use $d + 1$ to represent dimensions, here d is the spatial dimension and 1 represent temporal dimension. ∇ represents the differential operator i.e. $(\frac{\partial}{\partial x_1}, \ldots, \frac{\partial}{\partial x_d})$. For any real function u on a domain D, $\|u(x)\|_{L^\infty(D)}$ represents $\sup_{x \in D} |u(x)|$.

2 Main Results

2.1 Generalization Bounds for the $(d + 1)$-dimensional Burgers' PDE

The PDE that we consider is as follows,

$$\partial_t \boldsymbol{u} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = 0,\ \boldsymbol{u}(t = t_0) = \boldsymbol{u}_{t_0} \tag{1}$$

Here $\boldsymbol{u} : D \times [t_0, T] \to \mathbb{R}^d$ is the fluid velocity and $\boldsymbol{u}_{t_0} : D \to \mathbb{R}^d$ is the initial velocity. Then corresponding to a surrogate solution $\boldsymbol{u}_\theta$ we define the residuals as,

$$\mathcal{R}_{\text{pde}} \coloneqq \partial_t \boldsymbol{u}_\theta + (\boldsymbol{u}_\theta \cdot \nabla)\boldsymbol{u}_\theta,\ \mathcal{R}_{\text{t}} \coloneqq \boldsymbol{u}_\theta(t = t_0) - \boldsymbol{u}(t = t_0) \tag{2}$$

In the following theorem we consider $t_0 = \frac{-1}{\sqrt{2}} + \delta$ and $T = \delta$ for some $\delta > 0$. Here the spatial domain is represented by $D \subset \mathbb{R}^d$ and Ω represents the whole domain $D \times [t_0, T]$.

Theorem 2.1. *Let $d \in \mathbb{N}$ and $\boldsymbol{u} \in C^1(D \times [t_0, T])$ be the unique solution of the (d+1)-dimensional Burgers' equation given in (1). Then for any C^1 surrogate solution to equation 1, say $\boldsymbol{u}_\theta$, the L^2-risk with respect to the true solution is bounded as,*

$$\log\left(\int_\Omega \|\boldsymbol{u}(\boldsymbol{x},t) - \boldsymbol{u}_\theta(\boldsymbol{x},t)\|_2^2 \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t\right) \le \log\left(\frac{C_1 C_2}{4}\right) + \frac{C_1}{\sqrt{2}} \tag{3}$$

where, $C_1 = d^2\|\nabla \boldsymbol{u}_\theta\|_{L^\infty(\Omega)} + 1 + d^2\|\nabla \boldsymbol{u}\|_{L^\infty(\Omega)}$ *and* $C_2 = \int_D \|\mathcal{R}_t\|_2^2 \,\mathrm{d}\boldsymbol{x} + \int_\Omega \|\mathcal{R}_{pde}\|_2^2 \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t + d^2\|\nabla \boldsymbol{u}_\theta\|_{L^\infty(\Omega)} \int_\Omega \|\boldsymbol{u}_\theta\|_2^2 \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t + d^2\|\nabla \boldsymbol{u}\|_{L^\infty(\Omega)} \int_\Omega \|\boldsymbol{u}\|_2^2 \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t$

Proof of the above can be seen in Appendix A.1. We note that the bound presented in (3) does not make any assumptions about the existence of a blow-up in the solution and its applicable to all solutions that have continuous first derivatives however large, as would be true for the situations very close to blow-up as we would consider. Also, we note that the bound in De Ryck et al. (2022) makes assumptions which (even if set to zero pressure) prevent it from being directly applicable to the setups which can capture analytic solutions arbitrarily close to finite-time blow-up. We will also demonstrate, in following sections, that the bounds obtained in one dimension - are "stable" in a precise sense as will be explained after the following theorem. Further discussion on Theorem 2.1 is in Appendix G.

2.2 Generalization Bounds for a Finite-Time Blow-Up Scenario with (1+1)-dimensional Burgers' PDE

For $u : [-1, 1] \times [t_0, T] \to \mathbb{R}$ being at least once continuously differentiable in each of its variables we consider a Burgers's PDE as follows on the space domain being $[-1, 1]$ and the two limits of the time domain being specified as $t_0 = -1 + \delta$ and $T = \delta$ for any $\delta > 0$,

$$u_t + uu_x = 0,\; u(x, -1+\delta) = \frac{x}{-2+\delta},\; u(-1,t) = \frac{1}{1-t}\;;\; u(1,t) = \frac{1}{t-1} \tag{4}$$

We note that in the setup for Burger's PDE being solved by neural nets that was analyzed in the pioneering work in Mishra & Molinaro (2022), the same amount of information was assumed to be known i.e the PDE, an initial condition and boundary conditions at the spatial boundaries. However in here, the values we choose for the above constraints are non-trivial and designed to cater to a known solution for this PDE i.e $u = \frac{x}{t-1}$ which blows up at $t = 1$.

For any C^1 surrogate solution to the above, say u_θ its residuals can be written as,

$$\mathcal{R}_{int,\theta}(x,t) \coloneqq \partial_t(u_\theta(x,t)) + \partial_x \frac{u_\theta^2(x,t)}{2},\; \mathcal{R}_{tb,\theta}(x) \coloneqq u_\theta(x,-1+\delta) - \frac{x}{-2+\delta} \tag{5}$$

$$(\mathcal{R}_{sb,-1,\theta}(t), \mathcal{R}_{sb,1,\theta}(t)) \coloneqq \left(u_\theta(-1,t) - \frac{1}{1-t},\; u_\theta(1,t) - \frac{1}{t-1}\right) \tag{6}$$

Theorem 2.2. *Let $u \in C^1((-1+\delta, \delta) \times (-1, 1))$ be the unique solution of the one dimensional Burgers' PDE in equation 4. Then for any surrogate solution for the same PDE, say $u^* \coloneqq u_{\theta^*}$ its L^2−risk w.r.t the true solution is bounded as,* $\left(\int_{-1+\delta}^{\delta}\int_{-1}^{1} |u(x,t) - u_\theta(x,t)|^2\, dxdt\right)$

$$\begin{aligned}\le \left[1 + Ce^C\right]\Bigg[\int_{-1}^{1} \mathcal{R}_{tb,\theta^*}(x)dx + 2C_{2b}\left(\int_{-1+\delta}^{\delta} \mathcal{R}_{sb,-1,\theta^*}^2(t)dt + \int_{-1+\delta}^{\delta} \mathcal{R}_{sb,1,\theta^*}^2(t)dt\right) \\ + 2C_{1b}\left(\left(\int_{-1+\delta}^{\delta} \mathcal{R}_{sb,-1,\theta^*}^2(t)dt\right)^{\frac{1}{2}} + \left(\int_{-1+\delta}^{\delta} \mathcal{R}_{sb,1,\theta^*}^2(t)dt\right)^{\frac{1}{2}}\right) + \int_{-1+\delta}^{\delta}\int_{-1}^{1} \mathcal{R}_{int,\theta^*}^2(x,t)dxdt\Bigg]\end{aligned} \tag{7}$$

where $C = 1 + 2C_{u_x}$, *with* $C_{u_x} = \|u_x\|_{L^\infty((-1+\delta,\delta)\times(-1,1))} = \left\|\frac{1}{t-1}\right\|_{L^\infty([-1+\delta,\delta])} = \frac{1}{1-\delta}$ *and* $C_{1b} = \|u(1,t)\|_{L^\infty([-1+\delta,\delta])}^2 = \frac{1}{(1-\delta)^2}$; $C_{2b} = \|u_{\theta^*}(1,t)\|_{L^\infty([-1+\delta,\delta])} + \frac{3}{2}\left(\frac{1}{1-\delta}\right)$

Proof in Appendix A.3. Note that the RHS of equation 7 is evaluatable without exactly knowing the eact true solution – the constants in equation 7 only requires some knowledge of the supremum value of u at the spatial boundaries and the behaviour of the first order partial derivative of u.

Most importantly, Theorem 2.2 shows that despite the setting here being of proximity to finite-time blow-up, the naturally motivated PINN risk in this case [1] is "(L_2, L_2, L_2, L_2)-stable"[2] in the precise sense as defined in Wang et al. (2022a). In Appendix A.4 we apply quadrature rules on (7) and show a version of the above bound which makes the sample size dependency of the bound more explicit.

3 Experiments

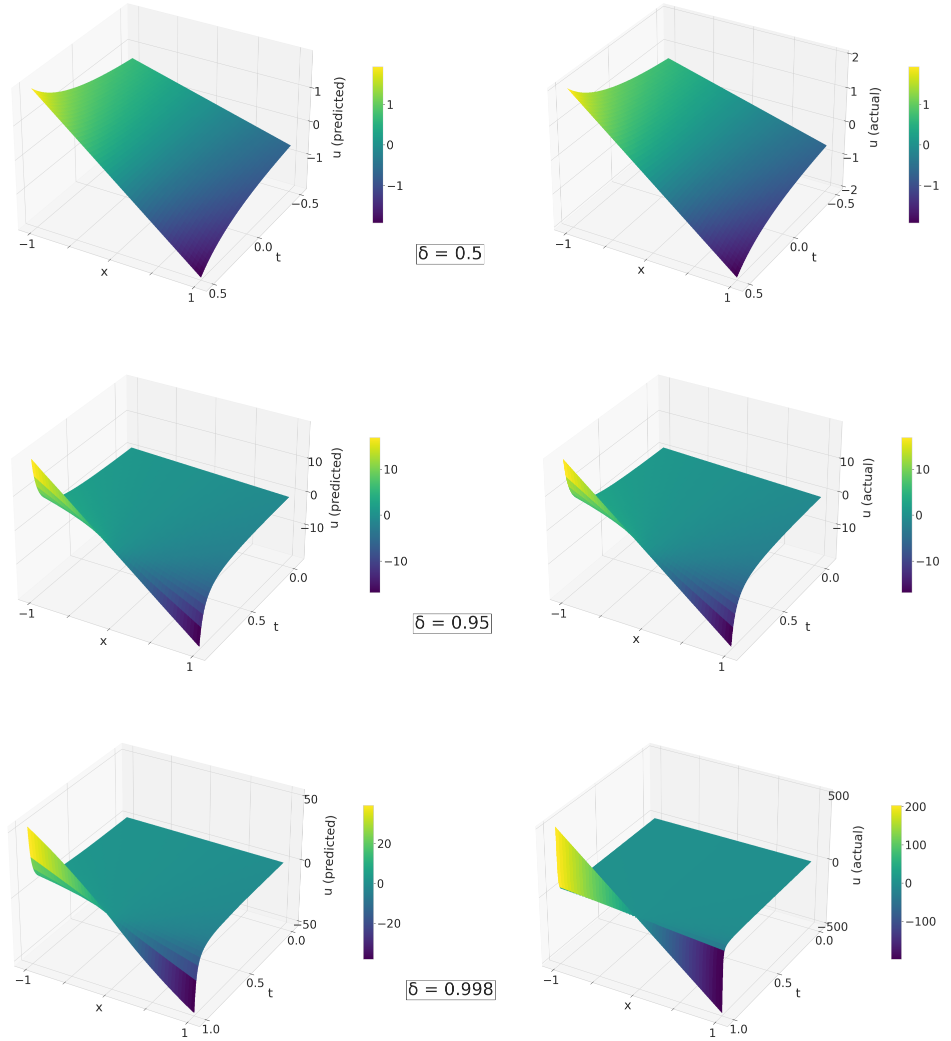

Figure 1: A demonstration of the visual resemblance between the neurally derived solution for equation 4 (left) and the true solution (right) at different values of the δ parameter getting close to the PDE with blow-up at $\delta = 1$. A PINN with a width of 300 and a depth of 6 was trained to generate the plots on the left.

Our experiments are designed to demonstrate the efficacy of the generalization error bounds discussed in Section 2 in the vicinity of finite-time blow-ups happening in our use cases. Towards motivating the novelty of our setup we give a brief overview of how demonstrations of deep-learning generalization bounds have been done in the recent past in Appendix D.

[1] PINN risk is defined as $\mathbb{E}[|\mathcal{R}_{int,\theta}(x,t)|^2] + \mathbb{E}[|\mathcal{R}_{tb,\theta}|^2] + \mathbb{E}[|\mathcal{R}_{sb,-1,\theta}|^2] + \mathbb{E}[|\mathcal{R}_{sb,1,\theta}|^2]$

[2] Suppose Z_1, Z_2, Z_3, Z_4 are four Banach spaces, a PDE defined by (41) is Z_1, Z_2, Z_3, Z_4-stable, if $\|\boldsymbol{u}_\theta(x,t) - \boldsymbol{u}(x,t)\|_{Z4} = \mathcal{O}(\|\partial_t \boldsymbol{u}_\theta + \mathcal{N}_{\boldsymbol{x}}[\boldsymbol{u}_\theta(\boldsymbol{x},t)]\|_{Z_1} + \|\boldsymbol{u}_\theta(\boldsymbol{x},0) - h(\boldsymbol{x})\|_{Z_2} + \|\boldsymbol{u}_\theta(\boldsymbol{x},t) - g(\boldsymbol{x},t)\|_{Z_3})$ as $\|\partial_t \boldsymbol{u}_\theta + \mathcal{N}_{\boldsymbol{x}}[\boldsymbol{u}_\theta(\boldsymbol{x},t)]\|_{Z_1}, \|\boldsymbol{u}_\theta(\boldsymbol{x},0) - h(\boldsymbol{x})\|_{Z_2}, \|\boldsymbol{u}_\theta(\boldsymbol{x},t) - g(\boldsymbol{x},t)\|_{Z_3} \to 0$ for any $\boldsymbol{u}_\theta$

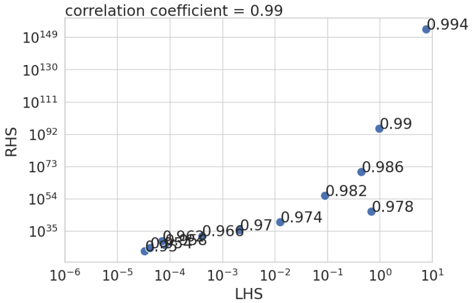

Figure 2: Demonstration of the presence of high correlation between the LHS and the RHS of (7) in Theorem 2.2 over PDE setups increasingly close to the singularity for a PINN of width 300. Each point is labeled with the value of δ.

The Finite-Time Blow-Up of (1+1)-dimensional Burgers PDE from Section 2.2 The neural networks we use here has a depth of 6 layers, and we experiment at a width of 300 neurons. For training, we use full-batch Adam optimizer for $100,000$ iterations and a learning rate of 10^{-4}. We subsequently select the model with the lowest training error for further analysis. In Figure 1 the plots have been shown for the predicted and actual solution, for neural nets solving equation 4 at different values of the δ parameter and it is clear that the visual resemblance of the neurally derived solution persists even quite close to the blowup at $\delta = 1$. In Figure 2 we see that the LHS and the RHS of (7) measured on the trained models is such that the correlation is very high (~ 1) over multiple values of the proximity parameter – up to being very close to the blow-up point. In Figure 5 in the appendix we show the plot for 30 neurons width, we note that the correlation increases with the width of the neural net, a desirable phenomenon that our bound does capture – albeit implicitly. In Figure 4 in the appendix, we illustrate that the upper-bound derived in Theorem 2.2 does indeed fall over a reasonable range of widths at a fixed δ. The mean and the standard deviations plotted therein are obtained over six iterations of the experiment at different random seeds.

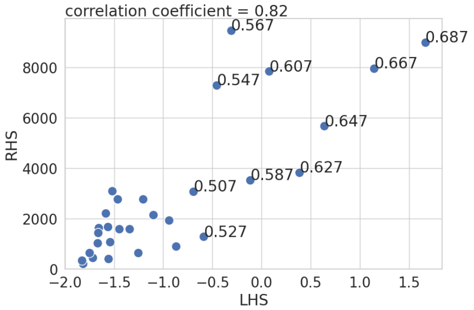

Figure 3: This plot shows the behaviour of LHS and RHS of (3) in Theorem 2.1, for a PINN of width 100 trained at different values of the δ parameter that quantifies proximity to the blow-up point. Only points with $\delta > \frac{1}{2}$ are marked with the value for clarity.

Testing Against a (2+1)-dimensional Exact Burgers' Solution with Finite-Time Blow-Up From (Biazar & Aminikhah, 2009) we know that there is an exact finite-time blow-up solution for Burgers' PDE in (1) for the case of $d = 2$, $u_1 = \frac{x_1+x_2-2x_1t}{1-2t^2}$, $u_2 = \frac{x_1-x_2-2x_2t}{1-2t^2}$ where u_i denotes the i^{th} component of the velocity being solved for. Note that at $t = 0$, both the above velocities are smooth while they eventually develop singularities at $t = \frac{1}{\sqrt{2}}$ - as is the expected hallmark of non-trivial finite-time blow-up solutions of PDEs. Also note that this singularity is more difficult to solve for since it is blowing up as $\mathcal{O}(\frac{1}{t^2})$ as compared to the $\mathcal{O}(\frac{1}{t})$ blow-up in the previous section in one dimension. We set ourselves to the task of solving for this on a sequence of computational domains $x_1, x_2 \in [0,1]$ and $t \in [-\frac{1}{\sqrt{2}} + \delta, \delta]$ where $\delta \in [0, \frac{1}{\sqrt{2}})$. Hence we have a sequence of PDEs to solve for – parameterized by δ and larger δs getting close to the blow-up. Further details of this setup are given in Appendix F. In figure 3 we see the true risk and the derived bound in Theorem 2.1 for depth 6 neural nets obtained by training on the above loss. In Figure 6 in the appendix we show the plot for 30 neurons width, the experiments show that the insight from the previous demonstration continues to hold and more vividly so. Here, for the experiments at low width (30) the correlation stays around 0.50 until only $\delta = 0.307$, and beyond that it decreases rapidly. However, for experiments at width 100 the correlation remains close to 0.80 for δ much closer to the blow-up at $\frac{1}{\sqrt{2}}$.

4 Concluding Discussions

In this work we have taken some of the first-of-its kind steps to initiate research into understanding the ability of neural nets to solve PDEs at the edge of finite-time blow-up. Our work suggests a number of exciting directions of future research. Firstly, more sophisticated modifications to the PINN formalism could be found to solve PDEs specifically near finite-time blow-ups.

Secondly, we note that it remains an open question to establish if there is any PINN risk for the $d+1$-dimensional Burgers, for $d > 1$, that is stable by the condition stated in Wang et al. (2022a), as was shown to be true in our $1+1$-dimensional Burgers in Theorem 2.2.

References

Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 254–263. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/arora18b.html.

Christopher J. Arthurs and Andrew P. King. Active training of physics-informed neural networks to aggregate and interpolate parametric solutions to the navier-stokes equations. *Journal of Computational Physics*, 438:110364, 2021. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2021.110364. URL https://www.sciencedirect.com/science/article/pii/S002199912100259X.

Atilim Gunes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *Journal of Marchine Learning Research*, 18:1–43, 2018.

Jafar Biazar and Hossein Aminikhah. Exact and numerical solutions for non-linear burger's equation by vim. *Mathematical and Computer Modelling*, 49(7-8):1394–1400, 2009.

David Broomhead and David Lowe. Radial basis functions, multi-variable functional interpolation and adaptive networks. *ROYAL SIGNALS AND RADAR ESTABLISHMENT MALVERN (UNITED KINGDOM)*, RSRE-MEMO-4148, 03 1988.

Jian Cheng Wong, Chinchun Ooi, Abhishek Gupta, and Yew-Soon Ong. Learning in sinusoidal spaces with physics-informed neural networks. *IEEE Transactions on Artificial Intelligence*, pp. 1–15, 2022. doi: 10.1109/TAI.2022.3192362.

K. L. Cooke. A non-local existence theorem for systems of ordinary differential equations. *Rendiconti del Circolo Matematico di Palermo*, 4(3):301–308, Sep 1955. ISSN 1973-4409. doi: 10.1007/BF02854201. URL https://doi.org/10.1007/BF02854201.

Arka Daw, Jie Bu, Sifan Wang, Paris Perdikaris, and Anuj Karpatne. Rethinking the importance of sampling in physics-informed neural networks. *arXiv preprint arXiv:2207.02338*, 2022.

Arka Daw, Jie Bu, Sifan Wang, Paris Perdikaris, and Anuj Karpatne. Mitigating propagation failures in physics-informed neural networks using retain-resample-release (R3) sampling. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 7264–7302. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/daw23a.html.

Tim De Ryck, Ameya D Jagtap, and Siddhartha Mishra. Error estimates for physics informed neural networks approximating the navier-stokes equations. *arXiv preprint arXiv:2203.09346*, 2022.

Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.

Weinan E, Jiequn Han, and Arnulf Jentzen. Algorithms for solving high dimensional PDEs: from nonlinear monte carlo to machine learning. *Nonlinearity*, 35(1):278–310, dec 2021. doi: 10.1088/1361-6544/ac337f. URL https://doi.org/10.1088%2F1361-6544%2Fac337f.

Hamidreza Eivazi, Mojtaba Tahani, Philipp Schlatter, and Ricardo Vinuesa. Physics-informed neural networks for solving reynolds-averaged navier–stokes equations. *Physics of Fluids*, 34(7): 075117, jul 2022. doi: 10.1063/5.0095270. URL https://doi.org/10.1063%2F5.0095270.

N. Benjamin Erichson, Michael Muehlebach, and Michael W. Mahoney. Physics-informed autoencoders for lyapunov-stable fluid flow prediction, 2019.

Hiroshi Fujita. On the blowing up of solutions of the cauchy problem for $u_1 + \delta u + u^{1+\alpha}$. 1966. URL https://api.semanticscholar.org/CorpusID:118871869.

Hiroshi Fujita. On the nonlinear equations $\Delta u + e^u = 0$ and $\partial v/\partial t = \Delta v + e^v$. *Bulletin of the American Mathematical Society*, 75(1):132 – 135, 1969.

Miguel A Herrero and JJL Velázquez. Plane structures in thermal runaway. *Israel Journal of Mathematics*, 81:321–341, 1993.

Zheyuan Hu, Ameya D. Jagtap, George Em Karniadakis, and Kenji Kawaguchi. When do extended physics-informed neural networks (XPINNs) improve generalization? *SIAM Journal on Scientific Computing*, 44(5):A3158–A3182, sep 2022a. doi: 10.1137/21m1447039. URL https://doi.org/10.1137%2F21m1447039.

Zheyuan Hu, Ameya D. Jagtap, George Em Karniadakis, and Kenji Kawaguchi. When do extended physics-informed neural networks (xpinns) improve generalization? *SIAM Journal on Scientific Computing*, 44(5):A3158–A3182, 2022b. doi: 10.1137/21M1447039. URL https://doi.org/10.1137/21M1447039.

Antony Jameson, Luigi Martinelli, and J Vassberg. Using computational fluid dynamics for aerodynamics–a critical assessment. In *Proceedings of ICAS*, pp. 2002–1, 2002.

Eurika Kaiser, J. Nathan Kutz, and Steven L. Brunton. Data-driven discovery of koopman eigenfunctions for control, 2021.

Karniadakis, G.E., and Kevrekidis. Physics-informed machine learning, 2021.

Aditi Krishnapriyan, Amir Gholami, Shandian Zhe, Robert Kirby, and Michael W Mahoney. Characterizing possible failure modes in physics-informed neural networks. *Advances in Neural Information Processing Systems*, 34:26548–26560, 2021.

I.E. Lagaris, A. Likas, and D.I. Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks*, 9(5):987–1000, 1998. doi: 10.1109/72.712178. URL https://doi.org/10.1109%2F72.712178.

Zaharaddeen Lawal, Hayati Yassin, Daphne Teck Ching Lai, and Azam Idris. Physics-informed neural network (pinn) evolution and beyond: A systematic literature review and bibliometric analysis. *Big Data and Cognitive Computing*, 11 2022. doi: 10.3390/bdcc6040140.

Zongyi Li, Miguel Liu-Schiaffini, Nikola Kovachki, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Learning dissipative dynamics in chaotic systems, 2022.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, mar 2021. doi: 10.1038/s42256-021-00302-5. URL https://doi.org/10.1038%2Fs42256-021-00302-5.

Lu Lu, Xuhui Meng, Shengze Cai, Zhiping Mao, Somdatta Goswami, Zhongqiang Zhang, and George Em Karniadakis. A comprehensive and fair comparison of two neural operators (with practical extensions) based on fair data. *Computer Methods in Applied Mechanics and Engineering*, 393:114778, 2022. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2022.114778. URL https://www.sciencedirect.com/science/article/pii/S0045782522001207.

Siddhartha Mishra and Roberto Molinaro. Estimates on the generalization error of physics-informed neural networks for approximating PDEs. *IMA Journal of Numerical Analysis*, 43(1):1–43, 01 2022. ISSN 0272-4979. doi: 10.1093/imanum/drab093. URL https://doi.org/10.1093/imanum/drab093.

Anirbit Mukherjee. *A study of the mathematics of deep learning*. PhD thesis, The Johns Hopkins University, 2020.

Ramchandran Muthukumar and Jeremias Sulam. Sparsity-aware generalization theory for deep neural networks. In Gergely Neu and Lorenzo Rosasco (eds.), *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pp. 5311–5342. PMLR, 12–15 Jul 2023. URL https://proceedings.mlr.press/v195/muthukumar23a.html.

Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*, 2017.

Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. Towards understanding the role of over-parametrization in generalization of neural networks. *arXiv preprint arXiv:1805.12076*, 2018.

A. Pazy. *Some Nonlinear Evolution Equations*, pp. 183–205. Springer New York, New York, NY, 1983. ISBN 978-1-4612-5561-1. doi: 10.1007/978-1-4612-5561-1_6. URL https://doi.org/10.1007/978-1-4612-5561-1_6.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pp. 5301–5310. PMLR, 2019.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

Bogdan Raonić, Roberto Molinaro, Tobias Rohner, Siddhartha Mishra, and Emmanuel de Bezenac. Convolutional neural operators. *arXiv preprint arXiv:2302.01178*, 2023.

Franz M Rohrhofer, Stefan Posch, Clemens Gößnitzer, and Bernhard C Geiger. On the role of fixed points of dynamical systems in training physics-informed neural networks. *arXiv preprint arXiv:2203.13648*, 2022.

Cristopher Salvi, Maud Lemercier, and Andris Gerasimovics. Neural stochastic pdes: Resolution-invariant learning of continuous spatiotemporal dynamics, 2022.

Justin Sirignano and Konstantinos Spiliopoulos. Dgm: A deep learning algorithm for solving partial differential equations. *Journal of computational physics*, 375:1339–1364, 2018.

A. M. Stuart and M. S. Floater. On the computation of blow-up. *European Journal of Applied Mathematics*, 1(1):47–71, 1990. doi: 10.1017/S095679250000005X.

Yuya Tanaka. Finite-time blow-up in a two species chemotaxis-competition model with degenerate diffusion. *arXiv preprint arXiv:2304.13421*, 2023.

Nils Wandel, Michael Weinmann, and Reinhard Klein. Teaching the incompressible navier–stokes equations to fast neural surrogate models in three dimensions. *Physics of Fluids*, 33(4):047117, apr 2021. doi: 10.1063/5.0047428. URL https://doi.org/10.1063%2F5.0047428.

Chuwei Wang, Shanda Li, Di He, and Liwei Wang. Is l^2 physics informed loss always suitable for training physics informed neural network? *Advances in Neural Information Processing Systems*, 35:8278–8290, 2022a.

Rui Wang, Karthik Kashinath, Mustafa Mustafa, Adrian Albert, and Rose Yu. Towards physics-informed deep learning for turbulent flow prediction, 2020.

Sifan Wang, Yujun Teng, and Paris Perdikaris. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43(5): A3055–A3081, 2021a. doi: 10.1137/20M1318043. URL https://doi.org/10.1137/20M1318043.

Sifan Wang, Hanwen Wang, and Paris Perdikaris. Learning the solution operator of parametric partial differential equations with physics-informed deeponets. *Science Advances*, 7(40):eabi8605, 2021b. doi: 10.1126/sciadv.abi8605. URL https://www.science.org/doi/abs/10.1126/sciadv.abi8605.

Sifan Wang, Shyam Sankaran, and Paris Perdikaris. Respecting causality is all you need for training physics-informed neural networks. *arXiv preprint arXiv:2203.07404*, 2022b.

Sifan Wang, Xinling Yu, and Paris Perdikaris. When and why pinns fail to train: A neural tangent kernel perspective. *Journal of Computational Physics*, 449:110768, 2022c. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2021.110768. URL https://www.sciencedirect.com/science/article/pii/S002199912100663X.

Yongji Wang, Ching-Yao Lai, Javier Gómez-Serrano, and Tristan Buckmaster. Asymptotic self-similar blow up profile for 3-d euler via physics-informed neural networks, 2022d.

Aurel Wintner. The non-local existence problem of ordinary differential equations. *American Journal of Mathematics*, 67(2):277–284, 1945. ISSN 00029327, 10806377. URL http://www.jstor.org/stable/2371729.

Bing Yu et al. The deep ritz method: a deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, 6(1):1–12, 2018.

A Proofs for the Main Theorems

A.1 Proof of Theorem 2.1

Let $\boldsymbol{u}$ be the actual solution of the (d+1)-dimensional Burgers' PDE and $\boldsymbol{u}_\theta$ the predicted solution by the PINN with parameters θ. Let's define

$$\begin{aligned} f(\boldsymbol{u}) &\coloneqq \frac{\|\boldsymbol{u}\|_2^2}{2} \\ \hat{\boldsymbol{u}} &\coloneqq \boldsymbol{u}_\theta - \boldsymbol{u} \end{aligned}$$

Then we can write the (d+1)-dimenional Burgers' and it's residual as

$$\partial_t \boldsymbol{u} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = 0 \tag{8}$$

$$\mathcal{R}_{\text{pde}} \coloneqq \partial_t \boldsymbol{u}_\theta + (\boldsymbol{u}_\theta \cdot \nabla)\boldsymbol{u}_\theta \tag{9}$$

Now multiplying (9) with $\boldsymbol{u}_\theta$ on both sides we get

$$\boldsymbol{u}_\theta \cdot \mathcal{R}_{\text{pde}} = \partial_t f(\boldsymbol{u}_\theta) + \boldsymbol{u}_\theta \cdot \nabla f(\boldsymbol{u}_\theta) \tag{10}$$

Similarly, multiplying both sides of (8) with $\boldsymbol{u}$ we get

$$\partial_t f(\boldsymbol{u}) + \boldsymbol{u} \cdot \nabla f(\boldsymbol{u}) = 0 \tag{11}$$

Some calculation on (8) and (9) yields

$$\begin{aligned} &\boldsymbol{u} \cdot [(\partial_t \boldsymbol{u}_\theta + (\boldsymbol{u}_\theta \cdot \nabla)\boldsymbol{u}_\theta - \mathcal{R}_{\text{pde}}) - (\partial_t \boldsymbol{u} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u})] = -\hat{\boldsymbol{u}} \cdot [\partial_t \boldsymbol{u} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u}] \\ \implies &\partial_t(\boldsymbol{u} \cdot \hat{\boldsymbol{u}}) + \boldsymbol{u} \cdot ((\boldsymbol{u}_\theta \cdot \nabla)\boldsymbol{u}_\theta) - \boldsymbol{u} \cdot ((\boldsymbol{u} \cdot \nabla)\boldsymbol{u}) - \boldsymbol{u} \cdot \mathcal{R}_{\text{pde}} = -\hat{\boldsymbol{u}} \cdot ((\boldsymbol{u} \cdot \nabla)\boldsymbol{u}) \\ \implies &\partial_t(\boldsymbol{u} \cdot \hat{\boldsymbol{u}}) + \boldsymbol{u} \cdot ((\boldsymbol{u}_\theta \cdot \nabla)\boldsymbol{u}_\theta) - \boldsymbol{u} \cdot \nabla f(\boldsymbol{u}) - \boldsymbol{u} \cdot \mathcal{R}_{\text{pde}} = -\hat{\boldsymbol{u}} \cdot ((\boldsymbol{u} \cdot \nabla)\boldsymbol{u}) \end{aligned} \tag{12}$$

Let

$$S \coloneqq \frac{1}{2}\hat{\boldsymbol{u}} \cdot \hat{\boldsymbol{u}} \tag{13}$$

$$\begin{aligned} \implies \partial_t S &= \partial_t f(\boldsymbol{u}_\theta) - \partial_t f(\boldsymbol{u}) - \partial_t(\boldsymbol{u} \cdot \hat{\boldsymbol{u}}) \\ &= [\boldsymbol{u}_\theta \cdot \mathcal{R}_{\text{pde}} - \boldsymbol{u}_\theta \cdot \nabla f(\boldsymbol{u}_\theta)] + [\boldsymbol{u} \cdot \nabla f(\boldsymbol{u})] \\ &\quad - [-\boldsymbol{u} \cdot ((\boldsymbol{u}_\theta \cdot \nabla)\boldsymbol{u}_\theta) + \boldsymbol{u} \cdot \nabla f(\boldsymbol{u}) + \boldsymbol{u} \cdot \mathcal{R}_{\text{pde}} - \hat{\boldsymbol{u}} \cdot ((\boldsymbol{u} \cdot \nabla)\boldsymbol{u})] \\ &= \hat{\boldsymbol{u}} \cdot \mathcal{R}_{\text{pde}} - \boldsymbol{u}_\theta \cdot \nabla f(\boldsymbol{u}_\theta) + \boldsymbol{u} \cdot ((\boldsymbol{u}_\theta \cdot \nabla)\boldsymbol{u}_\theta) + \hat{\boldsymbol{u}} \cdot ((\boldsymbol{u} \cdot \nabla)\boldsymbol{u}) \\ &= \hat{\boldsymbol{u}} \cdot \mathcal{R}_{\text{pde}} + \hat{\boldsymbol{u}} \cdot ((\boldsymbol{u} \cdot \nabla)\boldsymbol{u} - (\boldsymbol{u}_\theta \cdot \nabla)\boldsymbol{u}_\theta) \end{aligned}$$

here we represent the spatial domain $[0,1] \times [0,1]$ by D and use Ω to represent the $D \times [-\frac{1}{\sqrt{2}} + \delta, \delta]$. We then define

$$\mathcal{T} \coloneqq \hat{\boldsymbol{u}} \cdot ((\boldsymbol{u} \cdot \nabla)\boldsymbol{u}) \tag{14}$$

$$\tilde{H} \coloneqq \hat{\boldsymbol{u}} \cdot ((\boldsymbol{u}_\theta \cdot \nabla)\boldsymbol{u}_\theta) \tag{15}$$

And this leads to,

$$\partial_t S + \tilde{H} = \hat{\boldsymbol{u}} \cdot \mathcal{R}_{\text{pde}} + \mathcal{T} \tag{16}$$

Thus we have the inequalities,

$$\begin{aligned} \int_D \partial_t \|\hat{\boldsymbol{u}}\|_2^2 \, \mathrm{d}\boldsymbol{x} \leq &\int_D \|\hat{\boldsymbol{u}}\|_2^2 \, \mathrm{d}\boldsymbol{x} + \int_D \|\mathcal{R}_{pde}\|_2^2 \, \mathrm{d}\boldsymbol{x} \\ &+ 2\int_D \hat{\boldsymbol{u}} \cdot ((\boldsymbol{u} \cdot \nabla)\boldsymbol{u}) \, \mathrm{d}\boldsymbol{x} - 2\int_D \hat{\boldsymbol{u}} \cdot ((\boldsymbol{u}_\theta \cdot \nabla)\boldsymbol{u}_\theta) \, \mathrm{d}\boldsymbol{x} \end{aligned}$$

[Using lemma A.1 we get]

$$\begin{aligned} &\leq \int_D \|\hat{\boldsymbol{u}}\|_2^2 \, \mathrm{d}\boldsymbol{x} + \int_D \|\mathcal{R}_{pde}\|_2^2 \, \mathrm{d}\boldsymbol{x} \\ &+ 2d^2 \|\nabla \boldsymbol{u}\|_{L^\infty(\Omega)} \int_D \|\boldsymbol{u}\|_2 \|\hat{\boldsymbol{u}}\|_2 \, \mathrm{d}\boldsymbol{x} + 2d^2 \|\nabla \boldsymbol{u}_\theta\|_{L^\infty(\Omega)} \int_D \|\boldsymbol{u}_\theta\|_2 \|\hat{\boldsymbol{u}}\|_2 \, \mathrm{d}\boldsymbol{x} \\ &\leq \int_D \|\hat{\boldsymbol{u}}\|_2^2 \, \mathrm{d}\boldsymbol{x} + \int_D \|\mathcal{R}_{pde}\|_2^2 \, \mathrm{d}\boldsymbol{x} \end{aligned}$$

$$
\begin{aligned}
&+ d^2 \|\nabla \boldsymbol{u}\|_{L^\infty(\Omega)} \int_D \left[\|\boldsymbol{u}\|_2^2 + \|\hat{\boldsymbol{u}}\|_2^2\right] \mathrm{d}\boldsymbol{x} + d^2 \|\nabla \boldsymbol{u}_\theta\|_{L^\infty(\Omega)} \int_D \left[\|\boldsymbol{u}_\theta\|_2^2 + \|\hat{\boldsymbol{u}}\|_2^2\right] \mathrm{d}\boldsymbol{x} \\
&\le \left[1 + d^2 \|\nabla \boldsymbol{u}\|_{L^\infty(\Omega)} + d^2 \|\nabla \boldsymbol{u}_\theta\|_{L^\infty(\Omega)}\right] \int_D \|\hat{\boldsymbol{u}}\|_2^2 \,\mathrm{d}\boldsymbol{x} \\
&+ \int_D \|\mathcal{R}_{pde}\|_2^2 \,\mathrm{d}\boldsymbol{x} + d^2 \|\nabla \boldsymbol{u}\|_{L^\infty(\Omega)} \int_D \|\boldsymbol{u}\|_2^2 \,\mathrm{d}\boldsymbol{x} + d^2 \|\nabla \boldsymbol{u}_\theta\|_{L^\infty(\Omega)} \int_D \|\boldsymbol{u}_\theta\|_2^2 \,\mathrm{d}\boldsymbol{x} \\
&\le C_1 \int_D \|\hat{\boldsymbol{u}}\|_2^2 \,\mathrm{d}\boldsymbol{x} + \int_D \|\mathcal{R}_{pde}\|_2^2 \,\mathrm{d}\boldsymbol{x} \\
&+ d^2 \|\nabla \boldsymbol{u}\|_{L^\infty(\Omega)} \int_D \|\boldsymbol{u}\|_2^2 \,\mathrm{d}\boldsymbol{x} + d^2 \|\nabla \boldsymbol{u}_\theta\|_{L^\infty(\Omega)} \int_D \|\boldsymbol{u}_\theta\|_2^2 \,\mathrm{d}\boldsymbol{x} \qquad (17)
\end{aligned}
$$

where

$$
\begin{aligned}
C_1 &= d^2 \|\nabla \boldsymbol{u}_\theta\|_{L^\infty(\Omega)} \\
&\quad + 1 + d^2 \|\nabla \boldsymbol{u}\|_{L^\infty(\Omega)}
\end{aligned}
$$

Lets define the domain $D \times [-\frac{1}{\sqrt{2}} + \delta, \tilde{\delta}]$ by $\tilde{\Omega}$ where $\tilde{\delta} \in [-\frac{1}{\sqrt{2}} + \delta, \delta)$.

Integrating over $\tilde{\Omega}$ we get

$$
\begin{aligned}
\int_{\tilde{\Omega}} \partial_t \|\hat{\boldsymbol{u}}\|_2^2 \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t &\le C_1 \int_{\tilde{\Omega}} \|\hat{\boldsymbol{u}}\|_2^2 \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t + \int_{\tilde{\Omega}} \|\mathcal{R}_{pde}\|_2^2 \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t \\
&+ d^2 \|\nabla \boldsymbol{u}\|_{L^\infty(\Omega)} \int_{\tilde{\Omega}} \|\boldsymbol{u}\|_2^2 \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t + d^2 \|\nabla \boldsymbol{u}_\theta\|_{L^\infty(\Omega)} \int_{\tilde{\Omega}} \|\boldsymbol{u}_\theta\|_2^2 \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t \\
&\le \int_D \|\mathcal{R}_t\|_2^2 \,\mathrm{d}\boldsymbol{x} + C_1 \int_{\tilde{\Omega}} \|\hat{\boldsymbol{u}}\|_2^2 \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t + \int_{\tilde{\Omega}} \|\mathcal{R}_{pde}\|_2^2 \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t \\
&+ d^2 \|\nabla \boldsymbol{u}\|_{L^\infty(\Omega)} \int_{\tilde{\Omega}} \|\boldsymbol{u}\|_2^2 \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t + d^2 \|\nabla \boldsymbol{u}_\theta\|_{L^\infty(\Omega)} \int_{\tilde{\Omega}} \|\boldsymbol{u}_\theta\|_2^2 \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t \\
&\le \int_D \|\mathcal{R}_t\|_2^2 \,\mathrm{d}\boldsymbol{x} + C_1 \int_{\Omega} \|\hat{\boldsymbol{u}}\|_2^2 \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t + \int_{\Omega} \|\mathcal{R}_{pde}\|_2^2 \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t \\
&+ d^2 \|\nabla \boldsymbol{u}\|_{L^\infty(\Omega)} \int_{\Omega} \|\boldsymbol{u}\|_2^2 \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t + d^2 \|\nabla \boldsymbol{u}_\theta\|_{L^\infty(\Omega)} \int_{\Omega} \|\boldsymbol{u}_\theta\|_2^2 \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t \\
&\le C_1 \int_{\tilde{\Omega}} \|\hat{\boldsymbol{u}}\|_2^2 \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t + C_2 \qquad (18)
\end{aligned}
$$

where,

$$
\begin{aligned}
C_2 = &\int_D \|\mathcal{R}_t\|_2^2 \,\mathrm{d}\boldsymbol{x} + \int_{\tilde{\Omega}} \|\mathcal{R}_{pde}\|_2^2 \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t + d^2 \|\nabla \boldsymbol{u}_\theta\|_{L^\infty(\Omega)} \int_{\tilde{\Omega}} \|\boldsymbol{u}_\theta\|_2^2 \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t \\
&+ d^2 \|\nabla \boldsymbol{u}\|_{L^\infty(\Omega)} \int_{\tilde{\Omega}} \|\boldsymbol{u}\|_2^2 \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t
\end{aligned}
$$

Applying Gronwall's inequality on (18) we get

$$
\int_D \left\|\hat{\boldsymbol{u}}(\boldsymbol{x}, \tilde{\delta})\right\|_2^2 \mathrm{d}\boldsymbol{x} \le C_2 + \int_{-\frac{1}{\sqrt{2}}+\delta}^{\tilde{\delta}} C_2 C_1 e^{\int_t^\delta C_1 \,\mathrm{d}s} \,\mathrm{d}t \le C_2 \left[1 + \int_{-\frac{1}{\sqrt{2}}+\delta}^{\tilde{\delta}} C_1 e^{\frac{C_1}{\sqrt{2}}} \,\mathrm{d}t\right] \qquad (19)
$$

Integrating (19) over $\mathrm{d}\tilde{\delta}$ we get

$$
\begin{aligned}
\int_\Omega \left\|\hat{\boldsymbol{u}}(\boldsymbol{x}, \tilde{\delta})\right\|_2^2 \mathrm{d}\boldsymbol{x}\,\mathrm{d}\tilde{\delta} &\le C_2 \int_{\frac{-1}{\sqrt{2}}+\delta}^{\delta} \left[1 + \int_{-\frac{1}{\sqrt{2}}+\delta}^{\tilde{\delta}} C_1 e^{\frac{C_1}{\sqrt{2}}} \,\mathrm{d}t\right] \mathrm{d}\tilde{\delta} \\
&\le C_2 \left[\frac{-1}{\sqrt{2}} + \int_{\frac{-1}{\sqrt{2}}+\delta}^{\delta} \int_{-\frac{1}{\sqrt{2}}+\delta}^{\tilde{\delta}} C_1 e^{\frac{C_1}{\sqrt{2}}} \,\mathrm{d}t\,\mathrm{d}\tilde{\delta}\right] \\
&\le C_2 \left[\frac{-1}{\sqrt{2}} + \frac{C_1}{4} e^{\frac{C_1}{\sqrt{2}}}\right] \\
\implies \log\left(\int_\Omega \left\|\hat{\boldsymbol{u}}(\boldsymbol{x}, \tilde{\delta})\right\|_2^2 \mathrm{d}\boldsymbol{x}\,\mathrm{d}\tilde{\delta}\right) &\le \log\left(\frac{C_1 C_2}{4}\right) + \frac{C_1}{\sqrt{2}} \qquad (20)
\end{aligned}
$$

A.2 Useful Lemmas

Lemma A.1.

$$\begin{aligned}
\int_D p \cdot ((q \cdot \nabla) r) dx &= \int_D \left[\sum_{i=1}^{d} p_i \left(q \cdot \nabla r_i \right) \right] dx \\
&\leq \int_D \left[\sum_{i=1}^{d} \|p\|_2 \left(\|q\|_2 \|\nabla r_i\|_2 \right) \right] dx \\
&\leq \int_D \|p\|_2 \|q\|_2 \left[\sum_{i=1}^{d} \|\nabla r_i\|_2 \right] dx \\
&\leq \int_D \|p\|_2 \|q\|_2 \left[\sum_{i=1}^{d} d \|\nabla r\|_{L^\infty(\Omega)} \right] dx \\
&\leq d^2 \|\nabla r\|_{L^\infty(\Omega)} \int_D \|p\|_2 \|q\|_2 dx
\end{aligned}$$

A.3 Proof of Theorem 2.2

Proof. We define

$$f(u)=\frac{u^2}{2}$$

which means the first equation in (4) can be written as

$$u_t+f(u)_x=0 \tag{21}$$

Then we define the entropy flux function as

$$\mathcal{Q}(u)=\int_a^u sf'(s)ds \quad \text{for any } a\in\mathbb{R}$$

Let $\hat{u}=u^*-u$. From the first equation in (5) we get

$$\partial_t\left(\frac{(u^*)^2}{2}\right)+\partial_x\mathcal{Q}(u^*)=u^*\mathcal{R}_{int,\theta*} \tag{22}$$

and from (21) we obtain

$$\partial_t\left(\frac{u^2}{2}\right)+\partial_x\mathcal{Q}(u)=0 \tag{23}$$

Some calculation on (21) and the first equation in (5) yields

$$\partial_t(u\hat{u})+\partial_x(u(f(u^*)-f(u)))=[f(u^*)-f(u)-\hat{u}f'(u)]u_x+u\mathcal{R}_{int,\theta^*} \tag{24}$$

Subtracting (24) and (23) from (22) we get

$$\partial_t S(u,u^*)+\partial_x H(u,u^*)=\hat{u}\mathcal{R}_{int,\theta^*}+T_1 \tag{25}$$

with,

$$\begin{aligned}S(u,u^*)&\coloneqq\frac{(u^*)^2}{2}-\frac{u^2}{2}-\hat{u}u=\frac{1}{2}\hat{u}^2,\\ H(u,u^*)&\coloneqq\mathcal{Q}(u^*)-\mathcal{Q}(u)-u(f(u^*)-f(u)),\\ T_1&=-\left[f(u^*)-f(u)-f'(u)\hat{u}\right]u_x\end{aligned}$$

As flux f is smooth, we can apply Taylor expansion[3] on T_1 and expand $f(u^*)$ at u,

$$\begin{aligned}T_1&=-\left[\cancel{f(u)}+\cancel{f'(u)\hat{u}}+\frac{f''(u+\gamma(u^*-u))}{2}(u^*-u)^2-\cancel{f(u)}-\cancel{f'(u)\hat{u}}\right]u_x\\ &\quad[\text{ where }\gamma\in(0,1)]\\ &=-\frac{1}{2}\cancelto{1}{f''(u+\gamma(u^*-u))}\,\hat{u}^2u_x\\ &=-\frac{1}{2}\hat{u}^2u_x\end{aligned}$$

Hence it can be reasonably bounded by

$$|T_1|\le\|u_x\|_{L^\infty}\hat{u}^2 \tag{26}$$

[3] with the Lagrange form of the remainder

where C_{u_x} is given by $C_{u_x} = \|u_x\|_{L^\infty}$. Next, we integrate (25) over the domain (-1,1)

$$
\begin{aligned}
&\int_{-1}^{1} \partial_t S(u,u^*)dx = -\int_{-1}^{1} \partial_x H(u,u^*)dx + \int_{-1}^{1} \hat{u}\mathcal{R}_{int,\theta^*}dx + \int_{-1}^{1} T_1 dx \\
\Rightarrow\ &\frac{\mathrm{d}}{\mathrm{d}t}\int_{-1}^{1} \frac{\hat{u}^2(x,t)}{2}dx \le H(u(-1,t),u^*(-1,t)) - H(u(1,t),u^*(1,t)) \\
&\qquad + C_{u_x}\int_{-1}^{1} \hat{u}^2(x,t)dx + \int_{-1}^{1} \hat{u}(x,t)\mathcal{R}_{int,\theta^*}(x,t)dx \\
\Rightarrow\ &\frac{\mathrm{d}}{\mathrm{d}t}\int_{-1}^{1} \hat{u}^2(x,t)dx \le 2H(u(-1,t),u^*(-1,t)) - 2H(u(1,t),u^*(1,t)) \\
&\qquad + 2C_{u_x}\int_{-1}^{1} \hat{u}^2(x,t)dx + \int_{-1}^{1} (\mathcal{R}^2_{int,\theta^*}(x,t) + \hat{u}^2(x,t))dx \\
\Rightarrow\ &\frac{\mathrm{d}}{\mathrm{d}t}\int_{-1}^{1} \hat{u}^2(x,t)dx \le 2H(u(-1,t),u^*(-1,t)) - 2H(u(1,t),u^*(1,t)) \\
&\qquad + C\int_{-1}^{1} \hat{u}^2(x,t)dx + \int_{-1}^{1} \mathcal{R}^2_{int,\theta^*}(x,t)dx
\end{aligned}
\tag{27}
$$

where $C = 1 + 2C_{u_x}$. We can estimate $H(u(1,t),u^*(1,t))$ using (4)

$$
\begin{aligned}
H(u(1,t),u^*(1,t)) &= \mathcal{Q}(u^*(1,t)) - \mathcal{Q}(u(1,t)) - u(1,t)(f(u^*(1,t)) - f(u(1,t))) \\
&= \mathcal{Q}'(\gamma_1\mathcal{R}_{sb,1,\theta^*}(t))\mathcal{R}_{sb,1,\theta^*}(t) - \frac{u(1,t)}{2}\left[\hat{u}(1,t)\left[u^*(1,t) + u(1,t)\right]\right] \\
&\left[\text{ for some } \gamma_1 \in (0,1) \text{ by the mean-value theorem}\right] \\
&= \gamma_1 f'(\gamma_1\mathcal{R}_{sb,1,\theta^*})\mathcal{R}^2_{sb,1,\theta^*}(t) - \frac{u(1,t)}{2}\left[\mathcal{R}_{sb,1,\theta^*} + 2u(1,t)\right]\mathcal{R}_{sb,1,\theta^*} \\
&= \gamma_1\left[f'(\gamma_1\mathcal{R}_{sb,1,\theta^*}) - \frac{u(1,t)}{2}\right]\mathcal{R}^2_{sb,1,\theta^*}(t) - u^2(1,t)\mathcal{R}_{sb,1,\theta^*} \\
&\le C_{2b}\mathcal{R}^2_{sb,1,\theta^*}(t) + u^2(1,t)|\mathcal{R}_{sb,1,\theta^*}| \\
&\text{with } C_{2b} = C_{2b}(\|f'\|_\infty, \|u\|_{C^1([-1,1]\times[-1+\delta,\delta])})
\end{aligned}
$$

Similarly we can estimate

$$
H(u(-1,t),u^*(-1,t)) \le C_{2b}\mathcal{R}^2_{sb,-1,\theta^*}(t) + u^2(-1,t)|\mathcal{R}_{sb,-1,\theta^*}|
$$

Now, we can integrate (27) over the time interval $[-1+\delta,\bar{\delta}]$ for any $\bar{\delta} \in [-1+\delta,\delta]$ and use the above inequalities along with (5)

$$
\begin{aligned}
&\int_{-1+\delta}^{\bar{\delta}} \frac{\mathrm{d}}{\mathrm{d}t}\int_{-1}^{1} \hat{u}^2(x,t)dxdt \le \int_{-1+\delta}^{\bar{\delta}} (2H(u(-1,t),u^*(-1,t)) - 2H(u(1,t),u^*(1,t)))dt \\
&\qquad + \int_{-1+\delta}^{\bar{\delta}} C\int_{-1}^{1} \hat{u}^2(x,t)dxdt + \int_{-1+\delta}^{\bar{\delta}}\int_{-1}^{1} \mathcal{R}^2_{int,\theta^*}(x,t)dxdt \\
\Rightarrow\ &\int_{-1}^{1} \hat{u}^2(x,\bar{\delta})dx \le \int_{-1}^{1} \hat{u}^2(x,-1+\delta)dx + 2C_{2b}\left[\int_{-1+\delta}^{\delta} \mathcal{R}^2_{sb,-1,\theta^*}(t)dt + \int_{-1+\delta}^{\delta} \mathcal{R}^2_{sb,1,\theta^*}(t)dt\right] \\
&\qquad + 2\left[\int_{-1+\delta}^{\delta} u^2(-1,t)|\mathcal{R}_{sb,-1,\theta^*}|dt + \int_{-1+\delta}^{\delta} u^2(1,t)|\mathcal{R}_{sb,1,\theta^*}|dt\right] \\
&\qquad + C\int_{-1+\delta}^{\bar{\delta}}\int_{-1}^{1} \hat{u}^2(x,t)dxdt + \int_{-1+\delta}^{\delta}\int_{-1}^{1} \mathcal{R}^2_{int,\theta^*}(x,t)dxdt
\end{aligned}
$$

$$
\Rightarrow\ \int_{-1}^{1} \hat{u}^2(x,\bar{\delta})dx \le \int_{-1}^{1} \mathcal{R}_{tb,\theta^*}(x)dx + 2C_{2b}\left[\int_{-1+\delta}^{\delta} \mathcal{R}^2_{sb,-1,\theta^*}(t)dt + \int_{-1+\delta}^{\delta} \mathcal{R}^2_{sb,1,\theta^*}(t)dt\right]
$$

$$+ 2C_{1b}\left[\int_{-1+\delta}^{\delta} |\mathcal{R}_{sb,-1,\theta^*}| dt + \int_{-1+\delta}^{\delta} |\mathcal{R}_{sb,1,\theta^*}| dt\right]$$

$$+ C\int_{-1+\delta}^{\bar{\delta}}\int_{-1}^{1} \hat{u}^2(x,t)dxdt + \int_{-1+\delta}^{\delta}\int_{-1}^{1} \mathcal{R}^2_{int,\theta^*}(x,t)dxdt$$

where $C_{1b} = C_{1b}(\|u(1,t)\|_{L^\infty})$

$$\leq \int_{-1}^{1} \mathcal{R}_{tb,\theta^*}(x)dx + 2\bar{C}_{2b}\left[\int_{-1+\delta}^{\delta} \mathcal{R}^2_{sb,-1,\theta^*}(t)dt + \int_{-1+\delta}^{\delta} \mathcal{R}^2_{sb,1,\theta^*}(t)dt\right]$$

$$+ 2C_{1b}(\delta - (\delta - 1))^{\frac{1}{2}}\left[\left(\int_{-1+\delta}^{\delta} \mathcal{R}^2_{sb,-1,\theta^*} dt\right)^{\frac{1}{2}} + \left(\int_{-1+\delta}^{\delta} \mathcal{R}^2_{sb,1,\theta^*} dt\right)^{\frac{1}{2}}\right]$$

$$+ C\int_{-1+\delta}^{\bar{\delta}}\int_{-1}^{1} \hat{u}^2(x,t)dxdt + \int_{-1+\delta}^{\delta}\int_{-1}^{1} \mathcal{R}^2_{int,\theta^*}(x,t)dxdt$$

by using Holder's inequality

$$\leq C_T + C\int_{-1+\delta}^{\bar{\delta}}\int_{-1}^{1} \hat{u}^2(x,t)dxdt$$

$$\text{where } C_T = \int_{-1}^{1} \mathcal{R}_{tb,\theta^*}(x)\,\mathrm{d}x + 2C_{2b}\left[\int_{-1+\delta}^{\delta} \mathcal{R}^2_{sb,-1,\theta^*}(t)\,\mathrm{d}t + \int_{-1+\delta}^{\delta} \mathcal{R}^2_{sb,1,\theta^*}(t)\,\mathrm{d}t\right]$$

$$+ \int_{-1+\delta}^{\delta}\int_{-1}^{1} \mathcal{R}^2_{int,\theta^*}(x,t)\,\mathrm{d}x\,\mathrm{d}t + 2C_{1b}\left[\left(\int_{-1+\delta}^{\delta} \mathcal{R}^2_{sb,-1,\theta^*}\,\mathrm{d}t\right)^{\frac{1}{2}} + \left(\int_{-1+\delta}^{\delta} \mathcal{R}^2_{sb,1,\theta^*}\,\mathrm{d}t\right)^{\frac{1}{2}}\right] \quad (28)$$

Using integral form of Grönwall's inequality on (28)

$$\int_{-1}^{1} \hat{u}^2(x,\bar{\delta})dx \leq C_T + \int_{-1+\delta}^{\bar{\delta}} C_T C e^{\int_t^{\delta} C ds} dt \leq \left[1 + \int_{-1+\delta}^{\bar{\delta}} C e^C dt\right] C_T \quad (29)$$

Integrating (29) over $\bar{\delta}$ together with the definition of generalization error (35) we get

$$\int_{-1+\delta}^{\delta}\int_{-1}^{1} \hat{u}^2(x,\bar{\delta})dxd\bar{\delta} \leq C_T \int_{-1+\delta}^{\delta}\left[1 + \int_{-1+\delta}^{\bar{\delta}} C e^C dt\right] d\bar{\delta}$$

$$\mathcal{E}_G^2 \leq \left[1 + Ce^C\right] C_T \quad (30)$$

□

A.4 Making The Data Dependence Explicit in the bounds for $1 + 1$ Burgers' PDE

A.4.1 Quadrature Rule

Let' say we have a mapping $g : \mathbb{D} \to \mathbb{R}^m$ such that $g \in Z^* \subset L^p(\mathbb{D}, \mathbb{R}^m)$ and $\mathbb{D} \subset \mathbb{R}^{\bar{d}}$. Let's say we have an integral that we want to approximate

$$\bar{g} \coloneqq \int_{\mathbb{D}} g(y)\, dy \quad (31)$$

where dy denotes the $\bar{d}$-dimensional Lebesgue measure. To approximate this integral by the quadrature rule we need (i) the quadrature points $y_i \in \mathbb{D}$ for $1 \leq i \leq N$ for some $N \in \mathbb{N}$ and (ii) weights w_i with $w_i \in \mathbb{R}_+$. Then we can approximate (31) by the quadrature

$$\bar{g}_N \ \coloneqq \ \sum_{i=1}^{N} w_i g(y_i) \quad (32)$$

Then the error of this approximation is bounded by

$$|\ \bar{g} - \bar{g}_N\ | \leq C_{quad}(\|g\|_{Z^*}, \bar{d})\ N^{-\alpha}\ , \text{ for some } \alpha > 0 \quad (33)$$

These quadrature weights, quadrature points and α vary with $\bar{d}$'s range.

A.4.2 Applying Quadrature rule on Theorem 2.2

The loss function can then be written as

$$
\begin{aligned}
\mathcal{L}(\theta) \;=\; \mathcal{E}_T^2 \;\coloneqq\; & \frac{1}{N_{tb}} \underbrace{\sum_{n=1}^{N_{tb}} w_n^{tb} |\mathcal{R}_{tb,\theta^*}(x_n)|^2}_{(\mathcal{E}_T^{tb})^2} + \frac{1}{N_{sb}} \underbrace{\sum_{n=1}^{N_{sb}} w_n^{sb} |\mathcal{R}_{sb,-1,\theta^*}(t_{n,\delta})|^2}_{(\mathcal{E}_T^{sb,-1})^2} \\
& + \frac{1}{N_{sb}} \underbrace{\sum_{n=1}^{N_{sb}} w_n^{sb} |\mathcal{R}_{sb,1,\theta^*}(t_{n,\delta})|^2}_{(\mathcal{E}_T^{sb,1})^2} + \frac{\lambda}{N_{int}} \underbrace{\sum_{n=1}^{N_{int}} w_n^{int} |\mathcal{R}_{int,\theta^*}(x_n, t_{n,\delta})|^2}_{(\mathcal{E}_T^{int})^2}
\end{aligned} \tag{34}
$$

In our experiments we choose all w-s to be equal to 1 and train our model on that. We define the L^2-risk of u_θ with respect to the true solution u of equation

$$
\mathcal{E}_G(u_\theta) \coloneqq \left(\int_{-1+\delta}^{\delta} \int_{-1}^{1} |u(x,t) - u_\theta(x,t)|^2 \, dxdt \right)^{\frac{1}{2}} \tag{35}
$$

Theorem A.2. *Let $u \in C^k((-1+\delta,\delta) \times (-1,1))$ be the unique solution of the viscous scalar conservation law for any $k \geq 1$. Let $u^* = u_{\theta^*}$ be the PINN, then the generalization error (35) is bounded by*

$$
\begin{aligned}
\mathcal{E}_G^2(u^*) \leq \left(1 + Ce^C\right) \Bigg[& \sum_{n=1}^{N_{tb}} w_n^{tb} |\mathcal{R}_{tb,\theta^*}(x_n)|^2 + \sum_{n=1}^{N_{int}} w_n^{int} |\mathcal{R}_{int,\theta^*}(x_n, t_{n,\delta})|^2 \\
& + 2C_{2b} \left(\sum_{n=1}^{N_{sb}} w_n^{sb} |\mathcal{R}_{sb,-1,\theta^*}(t_{n,\delta})|^2 + \sum_{n=1}^{N_{sb}} w_n^{sb} |\mathcal{R}_{sb,1,\theta^*}(t_{n,\delta})|^2 \right) + 2C_{1b} \left(\mathcal{E}_T^{sb,-1} + \mathcal{E}_T^{sb,1} \right) \\
& + \frac{C_{quad}^{tb}}{N_{tb}^{\alpha_{tb}}} + \frac{C_{quad}^{int}}{N_{int}^{\alpha_{int}}} + 2C_{2b} \left(\frac{\left(C_{quad}^{sb,-1} + C_{quad}^{sb,1} \right)}{N_{sb}^{\alpha_{sb}}} \right) + 2C_{1b} \left(\frac{\left(C_{quad}^{sb,-1} + C_{quad}^{sb,1} \right)}{N_{sb}^{\frac{\alpha_{sb}}{2}}} \right) \Bigg]
\end{aligned} \tag{36}
$$

where $C = 1 + 2C_{u_x}$, with

$$
\begin{aligned}
C_{u_x} &= \|u_x\|_{L^\infty} = \left\| \frac{1}{t-1} \right\|_{L^\infty([-1+\delta,\delta])} \\
C_{1b} &= \|u(1,t)\|_{L^\infty}^2 = \left\| \frac{1}{1-t} \right\|_{L^\infty([-1+\delta,\delta])}^2 \\
C_{2b} &= \|u_{\theta^*}(1,t)\|_{L^\infty([-1+\delta,\delta])} + \frac{3}{2} \left\| \frac{1}{t-1} \right\|_{L^\infty([-1+\delta,\delta])}
\end{aligned} \tag{37}
$$

and $C_{quad}^{tb} = C_{quad}^{tb}\left(\left\| \mathcal{R}_{tb,\theta^*}^2 \right\|_{C_k} \right)$, $C_{quad}^{int} = C_{quad}^{int}\left(\left\| \mathcal{R}_{int,\theta^*}^2 \right\|_C^{k-2} \right)$, $C_{quad}^{sb,-1} = C_{quad}^{sb,-1}\left(\|\mathcal{R}_{sb,-1,\theta^*}\|_{C^k} \right)$, $C_{quad}^{sb,1} = C_{quad}^{sb,1}\left(\left\| \mathcal{R}_{sb,1,\theta^*}^2 \right\|_{C^k} \right)$ *are constants of the quadrature bound.*

Proof. In equation (30) within the proof of Theorem 2.2 we see that

$$
\int_{-1+\delta}^{\delta} \int_{-1}^{1} \hat{u}^2(x,\bar{\delta}) dx d\bar{\delta} \leq C_T \int_{-1+\delta}^{\delta} \left[1 + \int_{-1+\delta}^{\bar{\delta}} Ce^C dt \right] d\bar{\delta}
$$

$$
\mathcal{E}_G^2 \leq \left[1 + Ce^C \right] C_T
$$

where

$$
\begin{aligned}
C_T = & \underbrace{\int_{-1}^{1} \mathcal{R}_{tb,\theta^*}(x) dx}_{1} + \underbrace{2C_{2b} \left[\int_{-1+\delta}^{\delta} \mathcal{R}_{sb,-1,\theta^*}^2(t) dt + \int_{-1+\delta}^{\delta} \mathcal{R}_{sb,1,\theta^*}^2(t) dt \right]}_{3} \\
& + \underbrace{\int_{-1+\delta}^{\delta} \int_{-1}^{1} \mathcal{R}_{int,\theta^*}^2(x,t) dx dt}_{2} + \underbrace{2C_{1b} \left[\left(\int_{-1+\delta}^{\delta} \mathcal{R}_{sb,-1,\theta^*}^2 dt \right)^{\frac{1}{2}} + \left(\int_{-1+\delta}^{\delta} \mathcal{R}_{sb,1,\theta^*}^2 dt \right)^{\frac{1}{2}} \right]}_{4}
\end{aligned} \tag{38}
$$

Applying quadrature bounds on (38) we get,

$$\begin{aligned}
C_T \le\ & \underbrace{\sum_{n=1}^{N_{tb}} w_n^{tb} |\mathcal{R}_{tb,\theta^*}(x_n)|^2 + C_{quad}^{tb}(\|\mathcal{R}_{tb,\theta^*}\|_{C^k}) N_{tb}^{-\alpha_{tb}}}_{1} \\
& + \underbrace{\sum_{n=1}^{N_{int}} w_n^{int} |\mathcal{R}_{int,\theta^*}(x_n, t_{n,\delta})|^2 + C_{quad}^{int}(\|\mathcal{R}_{int,\theta^*}\|_{C^{k-2}}) N_{int}^{-\alpha_{int}}}_{2} \\
& + \underbrace{2C_{2b} \Bigg[\sum_{n=1}^{N_{sb}} w_n^{sb} |\mathcal{R}_{sb,-1,\theta^*}(t_{n,\delta})|^2 + \sum_{n=1}^{N_{sb}} w_n^{sb} |\mathcal{R}_{sb,1,\theta^*}(t_{n,\delta})|^2}_{3} \\
& + \underbrace{\left(C_{quad}^{sb}(\|\mathcal{R}_{sb,-1,\theta^*}\|_{C^k}) + C_{quad}^{sb}(\|\mathcal{R}_{sb,1,\theta^*}\|_{C^k}) \right) N_{sb}^{\alpha_{sb}} \Bigg]}_{3} \\
& + \underbrace{2C_{1b} \left[\left(\sum_{n=1}^{N_{sb}} w_n^{sb} |\mathcal{R}_{sb,-1,\theta^*}(t_{n,\delta})|^2 \right)^{\frac{1}{2}} + \left(\sum_{n=1}^{N_{sb}} w_n^{sb} |\mathcal{R}_{sb,1,\theta^*}(t_{n,\delta})|^2 \right)^{\frac{1}{2}} \right.}_{4} \\
& + \underbrace{\left. \left(C_{quad}^{sb}(\|\mathcal{R}_{sb,-1,\theta^*}\|_{C^k}) + C_{quad}^{sb}(\|\mathcal{R}_{sb,1,\theta^*}\|_{C^k}) \right)^{\frac{1}{2}} N_{sb}^{\frac{\alpha_{sb}}{2}} \right]}_{4}
\end{aligned} \tag{39}$$

Replacing the sums of residuals by training error we get

$$\begin{aligned}
\mathcal{E}_G^2 \le\ & \left(1 + Ce^C\right) \left[\left(\mathcal{E}_T^{tb}\right)^2 + \left(\mathcal{E}_T^{int}\right)^2 + 2C_{2b} \left(\left(\mathcal{E}_T^{sb,-1}\right)^2 + \left(\mathcal{E}_T^{sb,1}\right)^2 \right) + 2C_{1b} \left(\mathcal{E}_T^{sb,-1} + \mathcal{E}_T^{sb,1} \right) \right] \\
& + \left(1 + Ce^C\right) \Bigg[C_{quad}^{tb} N_{tb}^{-\alpha_{tb}} + C_{quad}^{int} N_{int}^{-\alpha_{int}} + 2C_{2b} \left(\left(C_{quad}^{sb,-1} + C_{quad}^{sb,1} \right) N_{sb}^{-\alpha_{sb}} \right) \\
& + 2C_{1b} \left(\left(C_{quad}^{sb,-1} + C_{quad}^{sb,1} \right) N_{sb}^{\frac{-\alpha_{sb}}{2}} \right) \Bigg]
\end{aligned} \tag{40}$$

□

B Physics Informed Neural Networks (PINNs)

B.1 Motivation behind PINNs

Partial Differential Equations (PDEs) are used for modeling a large variety of physical processes from fluid dynamics to bacterial growth to quantum behaviour at the atomic scale. But differential equations that can be solved in "closed form," that is, by means of a formula for the unknown function, are the exception rather than the rule. Even the biggest of industries still find it extremely expensive to implement the numerical PDE solvers – like airplane industries aiming to understand how wind turbulence pattern changes with changing aerofoil shapes, (Jameson et al., 2002) need to choose very fine discretizations which can often increase the run-times prohibitively.

In the recent past, deep learning has emerged as a competitive way to solve PDEs numerically. We note that the idea of using nets to solve PDEs dates back many decades Lagaris et al. (1998) (Broomhead & Lowe, 1988). In recent times this idea has gained significant momentum and "AI for Science" (Karniadakis et al., 2021) has emerged as a distinctive direction of research. Some of the methods at play for solving PDEs neurally (E et al., 2021) are the Physics Informed Neural Networks (PINNs) paradigm (Raissi et al., 2019) (Lawal et al., 2022), "Deep Ritz Method" (DRM, Yu et al. (2018)), "Deep Galerkin Method" (DGM), Sirignano & Spiliopoulos (2018) and many further variations that have been developed of these ideas, (Kaiser et al., 2021; Erichson et al., 2019; Wandel et al., 2021; Li et al., 2022; Salvi et al., 2022). An overarching principle that many of these implement is to try to constrain the loss function by using the residual of the PDE to be solved.

These different data-driven methods of solving the PDEs can broadly be classified into two kinds, **(1)** ones which train a single neural net to solve a specific PDE and **(2)** operator methods – which train multiple nets in tandem to be able to solve a family of PDEs in one shot. Lu et al. (2021, 2022); Wang et al. (2021b) The operator methods are particularly interesting when the underlying physics is not known and state-of-the-art approaches of this type can be seen in works like (Raonić et al., 2023).

For this work, we focus on the PINN formalism from Raissi et al. (2019). Many studies have demonstrated the success of this setup in simulating complex dynamical systems like the Navier-Stokes PDE (Arthurs & King, 2021; Wang et al., 2020; Eivazi et al., 2022), the Euler PDE (Wang et al., 2022d), descriptions of shallow water wave by the Korteweg-De Vries PDEs (Hu et al., 2022a) and many more.

Work in Mishra & Molinaro (2022); De Ryck et al. (2022) has provided the first-of-its-kind bounds on the generalization error of PINNs for approximating various standard PDEs, including the Navier-Stokes' PDE. Such bounds strongly motivate why minimization of the PDE residual at collocation points can be a meaningful way to solve the corresponding PDEs. However, the findings and analysis in Krishnapriyan et al. (2021); Wang et al. (2021a) point out that the training dynamics of PINNs can be unstable and failure cases can be found among even simple PDE setups. It has also been studied that when trivial solutions can exist for the PDE, the PINN training can get stuck at those solutions Rohrhofer et al. (2022); Cheng Wong et al. (2022). Work in Wang et al. (2022b) has shown that traditional ways of solving PINNs can violate causality.

B.2 A review of the framework of physics-informed neural networks

Consider the following specification of a PDE satisfied by an appropriately smooth function $\boldsymbol{u}(\boldsymbol{x}, t)$

$$\begin{aligned} &\boldsymbol{u}_t + \mathcal{N}_{\boldsymbol{x}}[\boldsymbol{u}] = 0, && \boldsymbol{x} \in D, t \in [0, T] \\ &\boldsymbol{u}(\boldsymbol{x}, 0) = h(\boldsymbol{x}), && \boldsymbol{x} \in D \\ &\boldsymbol{u}(\boldsymbol{x}, t) = g(\boldsymbol{x}, t), && \boldsymbol{t} \in [0, T], \boldsymbol{x} \in \partial D \end{aligned} \tag{41}$$

where $\boldsymbol{x}$ and t represent the space and time dimensions, subscripts denote the partial differentiation variables, $\mathcal{N}_{\boldsymbol{x}}[\boldsymbol{u}]$ is the nonlinear differential operator, D is a subset of $\mathbb{R}^d$ s.t it has a well-defined boundary ∂D. Following Raissi et al. (2019), we try to approximate $\boldsymbol{u}(\boldsymbol{x}, t)$ by a deep neural network $\boldsymbol{u}_\theta(\boldsymbol{x}, t)$, and then we can define the corresponding residuals as,

$$\mathcal{R}_{pde}(x, t) \coloneqq \partial_t \boldsymbol{u}_\theta + \mathcal{N}_{\boldsymbol{x}}[\boldsymbol{u}_\theta(\boldsymbol{x}, t)],\ \mathcal{R}_t(x) \coloneqq \boldsymbol{u}_\theta(\boldsymbol{x}, 0) - h(\boldsymbol{x}),\ \mathcal{R}_b(x, t) \coloneqq \boldsymbol{u}_\theta(\boldsymbol{x}, t) - g(\boldsymbol{x}, t)$$

Note that the partial derivative of the neural network ($\boldsymbol{u}_\theta$) can be easily calculated using auto-differentiation (Baydin et al., 2018). The neural net is then trained on a loss function

$$\mathcal{L}(\theta) \coloneqq \mathcal{L}_{pde}(\theta) + \mathcal{L}_t(\theta) + \mathcal{L}_b(\theta)$$

where $\mathcal{L}_{pde}$, $\mathcal{L}_t$ and $\mathcal{L}_b$ penalize for $\mathcal{R}_{pde}$, $\mathcal{R}_t$ and $\mathcal{R}_b$ respectively for being non-zero. Typically it would take the form

$$\mathcal{L}_{pde} = \frac{1}{N_{pde}} \sum_{i=1}^{N_{pde}} \left|\mathcal{R}_{pde}(x_r^i, t_r^i)\right|^2,\ \mathcal{L}_t = \frac{1}{N_t} \sum_{i=1}^{N_t} \left|\mathcal{R}_t(x_t^i)\right|^2,\ \mathcal{L}_b = \frac{1}{N_b} \sum_{i=1}^{N_b} \left|\mathcal{R}_b(x_b^i, t_b^i)\right|^2$$

where (x_r^i, t_r^i) denotes the collocation points, (x_t^i) are the points sampled on the spatial domain for the initial loss and (x_b^i, t_b^i) are the points sampled on the boundary for the boundary loss. The aim here is to train a neural net $\boldsymbol{u}_\theta$ such that $\mathcal{L}_\theta$ is as close to zero as possible.

B.3 Informal Summary of Our Results

At the very outset, we note that to the best of our knowledge there are no available off-the-shelf generalization bounds for any setup of PDE solving by neural nets where the assumptions being made include any known analytic solution with blow-up for the corresponding PDE. So, as a primary step we derive new risk bounds for Burgers' PDE in Theorem 2.1 and Theorem 2.2, where viscosity is set to zero and its boundary conditions are consistent with finite-time blow-up cases of Burgers' PDE that we eventually want to test on. Besides being designed to cater to blow-up situations, the bound in Theorem 2.2 is also "stable" in the sense of Wang et al. (2022a).

Our experiments reveal that for our test case with Burgers' PDE while the theoretical error bounds we derive are vacuous (as is routine for neural net generalization bounds), somewhat surprisingly they do maintain a non-trivial amount of correlation with the L^2-distance of the derived solution from the true solution.

C Failure modes of PINN

Notwithstanding the increasing examples of the success of PINNs, it is known that PINNs can at times fail to converge to the correct solution even for basic PDEs – as reflected in several recent studies on characterizing the "failure modes" of PINNs. Studies by Wang et al. (2021a), and more recently by Daw et al. (2023) have demonstrated that sometimes this failure can be attributed to problems associated with the loss function, specifically the uneven distribution of gradients across various components of the PINN loss. Wang et al. (2021a) attempt to address this issue by assigning specific weights to certain parts of the loss function. While Daw et al. (2022) developed a way to preferentially sample collocation points with high loss and subsequently use them for training. Krishnapriyan et al. (2021) observed a similar issue within the structure of the loss function. While not changing the PINN loss function, they introduced two techniques: "curriculum regularization" and "sequence-to-sequence learning" for PINNs to enhance their performance. In Wang et al. (2022c) PINNs have been analyzed from a neural tangent kernel perspective to suggest that PINNs suffer from "spectral-bias"(Rahaman et al., 2019) which makes it more susceptible to failure in the presence of "high frequency features" in the target function. They propose a method for improving training by assigning weights to individual components of the loss functions, aiming to mitigate the uneven convergence rates among the various loss elements. It has also been studied that when trivial solutions can exist for the PDE, the PINN training can get stuck at those solutions Rohrhofer et al. (2022); Cheng Wong et al. (2022). Work in Wang et al. (2022b) has shown that traditional ways of solving PINNs can violate causality.

D Review of previous generalization bound experiments

In the thought-provoking paper Dziugaite & Roy (2017) the authors had computed their bounds for 2-layer neural nets at various widths to show the non-vacuous nature of their bounds. But these bounds are not applicable to any single neural net but to an expected neural net sampled from a specified distribution. Inspired by these experiments, works like Neyshabur et al. (2017) and Mukherjee (2020) perform a de-randomized PAC-Bayes analysis on the generalization error of neural nets - which can be evaluated on any given net.

In works such as Neyshabur et al. (2018) we see a bound based on Rademacher analysis of the generalization error and the experiments were performed for depth-2 nets at different widths to show the decreasing nature of their bound with increasing width – a very rare property to be true for uniform convergence based bounds. It is important to point out that the training data is kept fixed while changing the width of the neural net in Dziugaite & Roy (2017) and Neyshabur et al. (2018).

In Arora et al. (2018) the authors instantiated a way to do compression of nets and computed the bounds on a compressed version of the original net. More recently in Muthukumar & Sulam (2023) the authors incorporated the sparsity of a neural net alongside the PAC-Bayed analysis to get a better bound for the generalization error. In their experiments, they vary the data size while keeping the neural net fixed and fortuitously the bound becomes non-vacuous for a certain width of the net.

E Plotting the behaviour of RHS of equation 7 with varying width

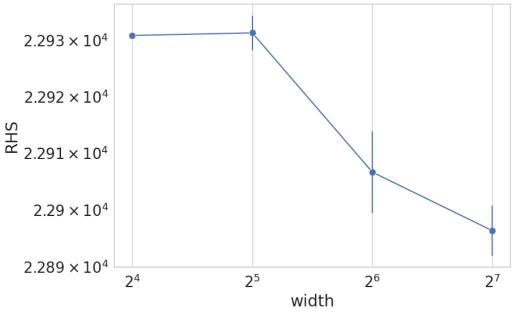

Figure 4: This plot tracks the RHS of equation (7) in Theorem 2.2 for training a depth 2 net at different widths towards solving equation 4 at $\delta = \frac{1}{2}$

F Details related to the training of PINN on $2 + 1$-Burgers' PDE

Let $g_{x_1,0}(x_2,t)$ and $g_{x_1,1}(x_2,t)$ be the boundary conditions for u_1 at $x_1 = 0$ & 1. Let $g_{x_2,0}(x_1,t)$ and $g_{x_2,1}(x_1,t)$ be the boundary conditions for u_2 at $x_2 = 0$ & 1 and u_{1,t_0} and u_{2,t_0} with $t_0 = -\frac{1}{\sqrt{2}}+\delta$ be the initial conditions for the two components of the velocity field. Hence the PDE we seek to solve is,

$$\left\{\ \boldsymbol{u}_t + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} = 0\right. \quad , \begin{cases} u_{1,t_0} = \frac{(1+\sqrt{2}-2\delta)x_1+x_2}{2\delta(\sqrt{2}-\delta)} \\ u_{2,t_0} = \frac{x_1-(1-\sqrt{2}+2\delta)x_2}{2\delta(\sqrt{2}-\delta)} \end{cases} , \begin{cases} g_{x_1,0}(x_2,t) \coloneqq u_1(x_1=0) = \frac{x_2}{1-2\cdot t^2} \\ g_{x_1,1}(x_2,t) \coloneqq u_1(x_1=1) = \frac{1+x_2-2\cdot t}{1-2\cdot t^2} \\ g_{x_2,0}(x_1,t) \coloneqq u_2(x_2=0) = \frac{x_1}{1-2\cdot t^2} \\ g_{x_2,1}(x_1,t) \coloneqq u_2(x_2=1) = \frac{x_1-1-2\cdot t}{1-2\cdot t^2} \end{cases} \tag{42}$$

Let $\mathcal{N} : \mathbb{R}^3 \to \mathbb{R}^2$ be the neural net to be trained, with output coordinates labeled as $(\mathcal{N}_{u_1}, \mathcal{N}_{u_2})$. Using this net we define the neural surrogates for solving the above PDE as,

$$u_{1,\theta} \coloneqq \mathcal{N}_{u_1}(x_1,x_2,t) \ \ u_{2,\theta} \coloneqq \mathcal{N}_{u_2}(x_1,x_2,t)$$

Correspondingly we define the PDE population risk, $\mathcal{R}_{pde}$ as,

$$\mathcal{R}_{pde} = \|\partial_t \boldsymbol{u}_\theta + \boldsymbol{u}_\theta \cdot \nabla \boldsymbol{u}_\theta\|^2_{[0,1]^2\times[-\frac{1}{\sqrt{2}}+\delta,\delta],\nu_1} \tag{43}$$

In the above $\boldsymbol{u}_\theta = (u_{1,\theta}, u_{2,\theta})$ and ν_1 is a measure on the whose space-time domain $[0,1]^2 \times [-\frac{1}{\sqrt{2}} + \delta, \delta]$. Corresponding to a measure ν_2 on $[0,1] \times [-\frac{1}{\sqrt{2}} + \delta, \delta]$ (first interval being space and the later being time), we define $\mathcal{R}_{s,0}$ and $\mathcal{R}_{s,1}$ corresponding to violation of the boundary conditions,

$$\begin{aligned} \mathcal{R}_{s,0} &= \|u_{1,\theta} - g_{x_1,0}(x_2,t)\|^2_{\{0\}\times[0,1]\times[-\frac{1}{\sqrt{2}}+\delta,\delta],\nu_2} + \|u_{2,\theta} - g_{x_2,0}(x_1,t)\|^2_{[0,1]\times\{0\}\times[-\frac{1}{\sqrt{2}}+\delta,\delta],\nu_2} \\ \mathcal{R}_{s,1} &= \|u_{1,\theta} - g_{x_1,1}(x_2,t)\|^2_{\{1\}\times[0,1]\times[-\frac{1}{\sqrt{2}}+\delta,\delta],\nu_2} + \|u_{2,\theta} - g_{x_2,1}(x_1,t)\|^2_{[0,1]\times\{1\}\times[-\frac{1}{\sqrt{2}}+\delta,\delta],\nu_2} \end{aligned} \tag{44}$$

For a choice of measure ν_3 on the spatial volume $[0,1]^2$ we define $\mathcal{R}_t$ corresponding to the violation of initial conditions $\boldsymbol{u}_{t_0} = (u_1(t_0), u_2(t_0))$,

$$\mathcal{R}_t = \|\boldsymbol{u}_\theta - \boldsymbol{u}_{t_0}\|^2_{[0,1]^2,t=t_0,\nu_3} \tag{45}$$

Thus the population risk we are looking to minimize is, $\mathcal{R} = \mathcal{R}_{pde} + \mathcal{R}_{s,0} + \mathcal{R}_{s,1} + \mathcal{R}_t$

We note that for the exact solution given above the constants in Theorem 2.1 evaluate to,

$$\begin{aligned} C_1 &= 2^2\|\nabla\boldsymbol{u}_\theta\|_{L^\infty(\Omega)} \\ &\quad + 1 + 2^2 \max_{t=\frac{1}{\sqrt{2}}+\delta,\delta}\left\{\left|\frac{1-2t}{1-2t^2}\right| + \left|\frac{1}{1-2t^2}\right|, \left|\frac{1}{1-2t^2}\right| + \left|\frac{1+2t}{1-2t^2}\right|\right\} \\ C_2 &= \int_D \|\mathcal{R}_t\|_2^2\,\mathrm{d}\boldsymbol{x} + \int_{\tilde{\Omega}} \|\mathcal{R}_{pde}\|_2^2\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t + 2^2\|\nabla\boldsymbol{u}_\theta\|_{L^\infty(\Omega)} \int_{\tilde{\Omega}} \|\boldsymbol{u}_\theta\|_2^2\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t \\ &\quad + 2^2 \max_{t=\frac{1}{\sqrt{2}}+\delta,\delta}\left\{\left|\frac{1-2t}{1-2t^2}\right| + \left|\frac{1}{1-2t^2}\right|, \left|\frac{1}{1-2t^2}\right| + \left|\frac{1+2t}{1-2t^2}\right|\right\} \int_{\tilde{\Omega}} \|\boldsymbol{u}\|_2^2\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t \\ &= \int_D \|\mathcal{R}_t\|_2^2\,\mathrm{d}\boldsymbol{x} + \int_{\tilde{\Omega}} \|\mathcal{R}_{pde}\|_2^2\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t + 2^2\|\nabla\boldsymbol{u}_\theta\|_{L^\infty(\Omega)} \int_{\tilde{\Omega}} \|\boldsymbol{u}_\theta\|_2^2\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t \\ &\quad + 2^2 \max_{t=\frac{1}{\sqrt{2}}+\delta,\delta}\left\{\left|\frac{1-2t}{1-2t^2}\right| + \left|\frac{1}{1-2t^2}\right|, \left|\frac{1}{1-2t^2}\right| + \left|\frac{1+2t}{1-2t^2}\right|\right\} \left[\frac{11t-7}{12(1-2t^2)} + \frac{5t+1}{12(1-2t^2)}\right]^{\delta}_{t=\frac{1}{\sqrt{2}}+\delta} \end{aligned}$$

G Discussion on the Generalization Bounds in Theorem 2.1

We note that the bounds in Section 2.1 are not dependent on the details of the loss function that might eventually be used in the training to obtained the $\boldsymbol{u}_\theta$. In that sense such a bound is more universal than usual generalization bounds which depend on the loss.

Also, note that the inequality proven in Theorem 2.1 bounds the distance of the true solution from a PINN solution in terms of (a) norms of the true solution and (b) various integrals of the found solution like its norms and unsupervised risks on the computation domain. Hence this is not like usual generalization bounds that get proven in deep-learning theory literature where the LHS is the population risk and RHS is upper-bounding that by a function that is entirely computable from knowing the training data and the trained net. Being in the setup of solving PDEs via nets lets us construct such new kinds of bounds which can exploit knowledge of the true PDE solution.

While Theorem 2.1 is applicable to Burgers' equations in any dimensions, it becomes computationally very expensive to compute the bound in higher dimensions. Therefore, in order to better our intuitive understanding, we separately analyze the case of $d = 1$, in the upcoming Section 2.2. Furthermore, the RHS of (3) only sees the errors at the initial time and in the space-time bulk. In general dimensions it is rather complicated to demonstrate that being able to measure the boundary risks of the surrogate solution can be leveraged to get stronger generalization bounds. But this can be transparently kept track of in the $d = 1$ case - as we will demonstrate now for a specific case with finite-time blow-up.

H Plotting the behaviour of LHS and RHS of equation (7) with varying δ in Theorem 2.2

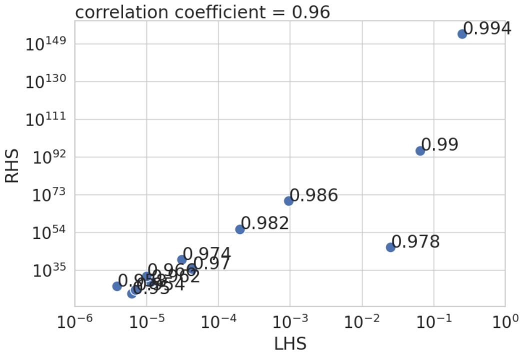

Figure 5: Demonstration of the presence of high correlation between the LHS (the true risk) and the RHS (and the derived bound) of equation (7) in Theorem 2.2 over PDE setups increasingly close to the singularity for a PINN of width 30. Each point is labeled with the value of δ.

I Plotting the behaviour of LHS and RHS of equation (3) with varying δ in Theorem 2.1

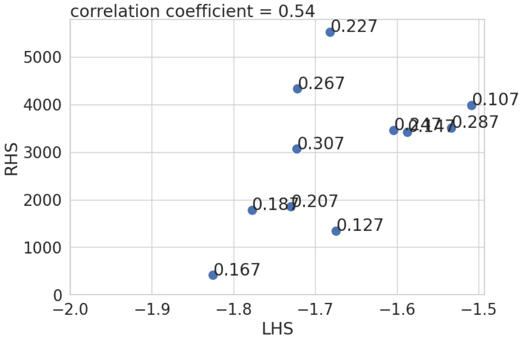

Figure 6: This plot shows the behaviour of LHS (the true risk) and RHS (and the derived bound) of (3) in Theorem 2.1, for a PINN of width 30 trained at different values of the δ parameter that quantifies proximity to the blow-up point. Only points with $\delta > \frac{1}{2}$ are marked with the value for clarity.