PINN Training via NTK: Insights and Limitations

1093084

Abstract

This paper investigates training dynamics of physics-informed neural networks (PINNs) to address prevalent challenges in optimisation of the related loss function. We review the results of Neural Tangent Kernel (NTK) theory, which provides an explantation for failures in PINN training and motivates preconditioning techniques to improve the loss-landscape. Noting that certain results of NTK theory break down in the case of nonlinear PDEs, we consider second-order methods as a means for overcoming optimisation challenges in this regime. Through numerical experiments, we validate NTK based models and show their effectiveness in both the linear and nonlinear regime. Further numerical results demonstrate that preconditioning techniques and second-order methods outperform traditional optimisers, highlighting their potential for robust PINN training.

1 Introduction

Despite their potential within SciML [1], physics-informed neural networks (PINNs) frequently encounter training difficulties, even in scenarios of relatively low complexity [2], [3]. A physics informed regularization makes the loss-landscape ill-conditioned [3] [4], especially when the underlying PDE exhibits numerical stiffness [2]. Wang et. al. [5] developed a theoretical understanding by analysing PINNs in the infinite width limit and proposed a way to improve the conditioning of training dynamics, using a so called neural tangent kernel (NTK). Building on this theory, recent research develops a theory of conditioning for PINNs and proposes to precondition the PINN-loss [6]. By contrast, [7] shows that certain pivotal results of NTK theory break down in the nonlinear regime and argues that second-order methods must thus be used. In this work, we set out to understand the training pathologies of PINNs through the lens of NTK theory and examine the effectiveness of NTK based approaches in contrast with second-order methods. Our primary questions are: (a) How can we understand training pathologies of PINNs theoretically through NTK theory? (b) Are the theoretical results in this domain valid in a practical setting? (c) Is NTK theory useful in the nonlinear regime or must second-order methods be employed here?

2 Preliminaries

Consider a boundary value problem (BVP) given by

$$\begin{cases} \mathcal{F}[u](\boldsymbol{x}) = f(\boldsymbol{x}), & \boldsymbol{x} \in \Omega \\ u(\boldsymbol{x}) = g(\boldsymbol{x}), & \boldsymbol{x} \in \partial\Omega \end{cases} \tag{$\mathcal{P}$}$$

where $\mathcal{F}$ is some differential operator, u is the unknown function and Ω is its spatio-temporal domain. A PINN approximates the solution to ($\mathcal{P}$) via a neural network. Let $\boldsymbol{\theta}$ be a vector of parameters of the network and $u_{\boldsymbol{\theta}}$ the corresponding network. Given boundary data $\{\boldsymbol{x}_b^i, g(\boldsymbol{x}_b^i)\}_{i=1}^{N_b}$ and collocation points $\{\boldsymbol{x}_r^i, f(\boldsymbol{x}_r^i)\}_{i=1}^{N_r}$ we introduce the boundary loss and residual loss, respectively:

$$\mathcal{L}_b(\boldsymbol{\theta}) := \frac{1}{2}\sum_{i=1}^{N_b} |u_{\boldsymbol{\theta}}(\boldsymbol{x}_b^i) - g(\boldsymbol{x}_b^i)|^2, \;\; \mathcal{L}_r(\boldsymbol{\theta}) := \frac{1}{2}\sum_{i=1}^{N_r} |\mathcal{N}[u_{\boldsymbol{\theta}}](\boldsymbol{x}_r^i) - f(\boldsymbol{x}_r^i)|^2.$$

The goal of a vanilla PINN is to minimise the constituent loss $\mathcal{L} = \mathcal{L}_b + \mathcal{L}_r$.

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

3 Theory of the Neural Tangent Kernel

Consider the solution $\boldsymbol{\theta}(t)$ of the gradient-flow system $\frac{d\boldsymbol{\theta}}{dt} = -\nabla\mathcal{L}(\boldsymbol{\theta})$, corresponding to gradient descent (GD) with infinitesimal step-size. In order to develop some theory around the training dynamics of PINNs, we make a central definition and present two results from [5].

Definition 1. *The neural tangent kernel (NTK) $\boldsymbol{K}(t)$ at time t of a PINN for $(\mathcal{P})$ is given by*

$$\boldsymbol{K}(t) := \begin{bmatrix} \boldsymbol{J}_u(\boldsymbol{\theta}(t)) \\ \boldsymbol{J}_r(\boldsymbol{\theta}(t)) \end{bmatrix} \left[\boldsymbol{J}_u^\top(\boldsymbol{\theta}(t)) \; \boldsymbol{J}_r^\top(\boldsymbol{\theta}(t)) \right], \; [\boldsymbol{J}_u(\boldsymbol{\theta})]_{ij} := \frac{du_{\boldsymbol{\theta}}}{d\theta_j}(\boldsymbol{x}_b^i), \;\; [\boldsymbol{J}_r(\boldsymbol{\theta})]_{ij} := \frac{d\mathcal{F}[u_{\boldsymbol{\theta}}]}{d\theta_j}(\boldsymbol{x}_r^i).$$

Moreover, we define $\boldsymbol{K}_{uu}(t) := \boldsymbol{J}_u(\boldsymbol{\theta}(t))\boldsymbol{J}_u^\top(\boldsymbol{\theta}(t))$ and $\boldsymbol{K}_{rr}(t) := \boldsymbol{J}_r(\boldsymbol{\theta}(t))\boldsymbol{J}_r^\top(\boldsymbol{\theta}(t))$.

In the sequel we consider a width-m neural network, which, with standard notation, takes the form

$$u_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{\sqrt{m}}\boldsymbol{W}^{(1)}(\sigma(\boldsymbol{W}^{(0)}\boldsymbol{x} + \boldsymbol{b}^{(0)})) + \boldsymbol{b}^{(1)}. \tag{1}$$

Lemma 1. *With $\boldsymbol{\theta}(t)$ as above, the corresponding PINN satisfies*

$$\begin{bmatrix} \frac{du_{\boldsymbol{\theta}(t)}(\boldsymbol{x}_b)}{dt} \\ \frac{d\mathcal{F}[u_{\boldsymbol{\theta}(t)}](\boldsymbol{x}_r)}{dt} \end{bmatrix} = -\boldsymbol{K}(t) \begin{bmatrix} u_{\boldsymbol{\theta}(t)}(\boldsymbol{x}_b) - g(\boldsymbol{x}_b) \\ \mathcal{F}[u_{\boldsymbol{\theta}(t)}](\boldsymbol{x}_r) - f(\boldsymbol{x}_r) \end{bmatrix}.$$

Remark 1. *The result holds for all differential operators and any neural network architecture, and shows that NTKs play a pivotal role in the training dynamics of PINNs.*

Lemma 2. *Under some reasonable assumptions found in [5], for the PINN corresponding to the 1-D Poisson equation, if $\boldsymbol{\theta}(0) \sim \mathcal{N}(0, \boldsymbol{I})$, then, in the infinite width limit, $\boldsymbol{K}(t)$ is deterministic at initialisation and constant throughout training, hence we get that*

$$\boldsymbol{Q} \begin{bmatrix} u_{\boldsymbol{\theta}(t)}(\boldsymbol{x}_b) - g(\boldsymbol{x}_b) \\ \mathcal{F}[u_{\boldsymbol{\theta}(t)}](\boldsymbol{x}_r) - f(\boldsymbol{x}_r) \end{bmatrix} \approx e^{-\boldsymbol{\Lambda}t}\boldsymbol{Q} \begin{bmatrix} g(\boldsymbol{x}_b) \\ f(\boldsymbol{x}_r) \end{bmatrix}, \tag{2}$$

where $\boldsymbol{K}(0) = \boldsymbol{Q}^\top\boldsymbol{\Lambda}\boldsymbol{Q}$ is the eigendecomposition of the NTK.

Remark 2. *Since $\boldsymbol{K}(0) = \begin{bmatrix} \boldsymbol{K}_{uu}(0) & * \\ * & \boldsymbol{K}_{rr}(0) \end{bmatrix}$, Lemma 2 shows that the convergence rates of $\mathcal{L}_b$ and $\mathcal{L}_r$ are determined by the scale of the eigenvalues of $\boldsymbol{K}_{uu}(0)$ and $\boldsymbol{K}_{rr}(0)$, respectively.*

It is empirically observed in [5] that the eigenvalues of $\boldsymbol{K}_{rr}(t)$ have much greater magnitude than those of $\boldsymbol{K}_{uu}(t)$ (in the setting of Lemma 2). The residual loss will thus converge faster than the boundary loss, so the network does not optimise both constituents of $\mathcal{L}$ equally. The investigation of the NTK thus provides a natural solution for making the training more well-conditioned: weight the respective loss constituents to balance the magnitudes of eigenvalues. It is important to note that the above analysis does not apply to all PINNs, as their NTK might not be constant. Nevertheless Lemma 1 holds in general, so [5] uses the constant NTK scenario as a heuristic for the training dynamics of a general PINN. Indeed, the discrepancy between the eigenvalues of $\boldsymbol{K}_{uu}(t)$ and $\boldsymbol{K}_{rr}(t)$ is presented as a fundamental theoretical explanation for failure in PINN training and this is used to motivate a learning algorithm for more effective training.

Algorithm 1 PINN-NTK

1: **Input:** Initialisation $\boldsymbol{\theta}_0$, # training iterations S, learning rate η
2: **for** $n = 0, \dots, S-1$ **do**
3: Set $\lambda_b = \frac{\mathrm{Tr}(\boldsymbol{K}(n))}{\mathrm{Tr}(\boldsymbol{K}_{uu}(n))}$, $\lambda_r = \frac{\mathrm{Tr}(\boldsymbol{K}(n))}{\mathrm{Tr}(\boldsymbol{K}_{rr}(n))}$ and update via $\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n - \eta\nabla_{\boldsymbol{\theta}}[\lambda_b\mathcal{L}_b + \lambda_r\mathcal{L}_r](\boldsymbol{\theta}_n)$.
4: **end for**
5: **return** $\boldsymbol{\theta}_S$

We saw above, that learning dynamics of PINNs can be understood in terms of the corresponding NTK, and that a discrepancy between eigenvalues may be at the root of training-pathologies. Note however that the analysis in [5] gives no theoretical insight into this discrepancy, and why it occurs. Taking inspiration from classic numerical analysis [6] introduces a theory of conditioning for PINNs.

Definition 2. *The relative condition number for solving* $(\mathcal{P})$ *with a PINN is*

$$\text{cond}(\mathcal{P}) := \lim_{\epsilon \to 0^+} \sup_{\substack{0 < ||\delta f||_{L^2(\Omega)} \leq \epsilon \\ \boldsymbol{\theta} \in \Theta}} \frac{||\delta u||_{L^2(\Omega)} / ||u||_{L^2(\Omega)}}{||\delta f||_{L^2(\Omega)} / ||f||_{L^2(\Omega)}}$$

where $\delta u = u_{\boldsymbol{\theta}} - u$, $\delta f = \mathcal{F}[u_{\boldsymbol{\theta}}] - f$, *and* Θ *is the parameter space, provided that* $0 \notin \{||u||, ||f||\}$.

Theorem 1. *Let* $\boldsymbol{\theta}(t)$ *be a solution to* $\frac{d\boldsymbol{\theta}}{dt} = -\nabla \mathcal{L}_r(\boldsymbol{\theta})$. *Under the assumptions of Lemma 1 and some further assumptions found in [6], if* $\text{cond}(\mathcal{P}) < \infty$, *there exist* $\xi > \sup_t \mathcal{L}(\boldsymbol{\theta}(t))$ *and* $\alpha : (0, \xi) \to \mathbb{R}$ *with* $\lim_{\epsilon \to 0^+} \alpha(\epsilon) = 0$ *such that*

$$Tr(\boldsymbol{K}_{rr}(t))/N_r \gtrapprox \frac{||f||^2_{L^2(\Omega)} / [||u||^2_{L^2(\Omega)} |\Omega|]}{\text{cond}(\mathcal{P})^2 + \alpha(\mathcal{L}_r(\boldsymbol{\theta}(t)))} ||\nabla_{\boldsymbol{\theta}} u_{\boldsymbol{\theta}(t)}||^2_{L^2(\Omega)}.$$

The above theorem directly connects the magnitude of eigenvalues of $\boldsymbol{K}_{rr}(t)$ to the corresponding condition number. This gives a more precise theoretical explanation for training pathologies and a strategy for efficient training: lower the condition number. Based on this idea, using traditional numerical analysis, [6] discretises $(\mathcal{P})$ as a linear system: $\boldsymbol{A}\boldsymbol{u} = \boldsymbol{b}$, where $\text{cond}(\mathcal{P}) \approx ||\boldsymbol{A}^{-1}|| ||\boldsymbol{b}|| / ||\boldsymbol{u}||$. Taking $\boldsymbol{P} = \boldsymbol{L}\boldsymbol{U} \approx \boldsymbol{A}$ (by ILU-factorisation) yields $||\boldsymbol{A}^{-1}\boldsymbol{P}|| ||\boldsymbol{P}^{-1}\boldsymbol{b}|| / ||\boldsymbol{u}|| \approx 1$, so we can introduce a new well-conditioned loss $\mathcal{L}_{\text{PC}}(\boldsymbol{\theta}) = ||\boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{u}_{\boldsymbol{\theta}} - \boldsymbol{P}^{-1}\boldsymbol{b}||^2$, where $\boldsymbol{u}_{\boldsymbol{\theta}}$ is the discretisation of $u_{\boldsymbol{\theta}}$. Performing GD on $\mathcal{L}_{\text{PC}}$ is referred to as the preconditioned PINN algorithm (PCPINN).

Note that the above methods are motivated by an example where the NTK is deterministic and constant in the infinite width. However, [7] shows that this assumption fails in the nonlinear regime.

Theorem 2. *Under the assumptions of Lemma 2, and further justified assumptions found in [7], if* $\mathcal{F}$ *is nonlinear and* m *is taken to infinity, then* $\boldsymbol{K}(t)$ *is random and almost surely non-constant.*

Theorem 3. *If* $\mathcal{F}(u)$ *is a degree 2 polynomial in* $(u, \partial_x u, \partial_x^2 u, \dots)$, *then* $\nabla^2 \mathcal{L}_r$ *is not sparse.*

Remark 3. *According to [7] Theorem 3 indicates that* $\nabla^2 \mathcal{L}_r$ *cannot be disregarded throughout training, so that second-order methods are necessary in the nonlinear regime.*

Similar to Remark 3, [8] proposes to train PINNs by GD followed by damped Newton's method (DN), which is theoretically motivated by a convergence result.

Algorithm 2 Gradient-Damped Newton Descent (GDND)

1: **Input:** $\boldsymbol{\theta}_0,\ S_1,\ S_2,\ \eta_1,\ \eta_2, \gamma$.
2: GD on $\mathcal{L}$ for S_1 iterations, with initialisation $\boldsymbol{\theta}_0$ and learning rate η_1.
3: DN on $\mathcal{L}$ for S_2 iterations with initialisation $\boldsymbol{\theta}_{S_1}$, learning rate η_2 and damping parameter γ.
4: **return** $\boldsymbol{\theta}_{S_1+S_2}$

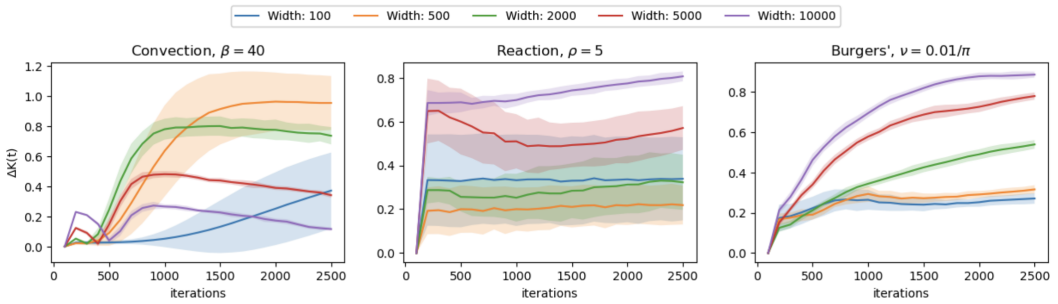
Theorem 4. *If* $\mathcal{L}$ *satisfies the so called PŁ***-condition in a certain ball around* $\boldsymbol{\theta}_0$, *there exists* S_1 *and* η_1 *such that* $\mathcal{L}(\boldsymbol{\theta}_{S_1+n}) \leq (2/3)^n \mathcal{L}(\boldsymbol{\theta}_{S_1})$, *provided* $\eta_2 = 5/6$ *and appropriate damping* γ.

Remark 4. *The PŁ***-condition holds with high probability for wide nets with least-squares loss [9].*

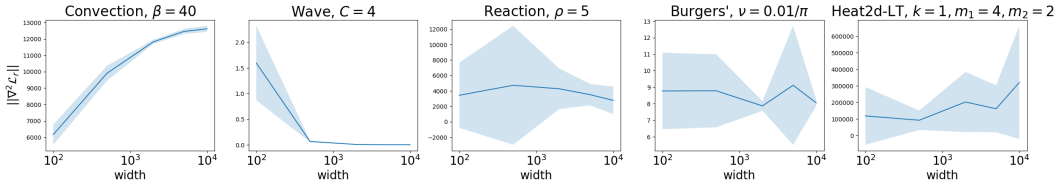
In practice, GDND is not used directly. Instead, we opt for a similar algorithm comprised of Adam followed by L-BFGS (a quasi-newton method), followed by NNCG (an adapted variant of DN) – all for maximal computational efficiency [8]. Broadly speaking, this method follows the same structure as GDND, so Theorem 4 is a relevant theoretical result that motivates Adam + L-BFGS + NNCG. For brevity we refer to Adam + L-BFGS + NNCG also as GDND.

4 Numerical Experiments

For the sake a consistent asymptotic behaviour in the theory, we have made some rather unorthodox assumptions on the considered neural network. More specifically, the normalisation factor $1/\sqrt{m}$ in (1) is usually omitted and $\boldsymbol{\theta}(0)$ is normally initialized using the Glorot scheme [10]. We wish to investigate whether the theoretical results above, hold in this more practical setting. Hence, in all experiments we use Glorot initialisation and a network without normalisation. We implement PINNs for solving several PDEs, both linear and nonlinear and test the claims of theorems 2 and 3.



(a)



(b)

Figure 1: (a) $\Delta \boldsymbol{K}(t) := ||\boldsymbol{K}(0) - \boldsymbol{K}(t)||_2 / ||\boldsymbol{K}(0)||_2$ throughout training for different widths and PDEs. (b) Spectral norm of residual loss Hessian versus network width, for one layer networks.

Furthermore, the claim in Remark 3 has a rather weak foundation and was not adequately tested in [7]. We therefore wish to investigate the effectiveness of NTK based approaches in the nonlinear regime, while taking the dynamics of the Hessian into consideration and comparing to second-order methods. The considered PDEs are displayed in Table 1. Throughout, we use the tanh activation function. To produce the results in Fig. 1a and Table 2 we adapted the code in [5], which implements Algorithm 1. For Fig. 1a we train PINNs using an Adam optimiser and calculate the NTK over 300 random sample-points (for numerical efficiency) at various iterations of the training process. We use an exponentially decaying learning rate, starting at 0.001 with decay rate 0.9, applied every 1000 iterations. For Table 2 the original results were obtained as the lowest error out of 10 independent trails and using the parameter values from [5] – we used a 3-layer net of width 500 and the learning rate scheme mentioned above. To obtain Fig. 1b we adapted the code in [8], which approximates the eigenvalues of $\nabla^2 \mathcal{L}_r$ at initialisation, using the techniques in [11] and [12]. The biggest of these eigenvalues in absolute values is the spectral norm of the Hessian matrix, as this is symmetric, since the residual loss is two times continuously differentiable when using tanh activation.

Name	Formula	Nonlinear	deg. 2 poly.	Parameter
Convection	$u_t + \beta u_x = 0$	×	×	$\beta = 40$
Wave	$u_{tt} - C u_{xx} = 0$	×	×	$C = 4$
Reaction	$u_t - \rho u(1-u) = 0$	✓	✓	$\rho = 5$
Burgers'	$u_t + u u_x = \nu u_{xx}$	✓	✓	$\nu = \pi/100$
Heat2d-LT	$u_t = \Delta u/1000 + 5\sin(ku^2) f(x,y,t)$ $f(x,y,t) = [1+2\sin(\pi t/4)]\sin(m_1 \pi x)\sin(m_2 \pi y)$	✓	×	$k=1$, $m_1=4, m_2=2$

Table 1: PDEs used for numerical testing. "deg. 2 poly." refers to the condition in Theorem 3. Further details are found in [6], [8] and [13].

In Fig. 1a we not only see that the contents of Theorem 2 hold in the practical setting, but also that the NTK remains constant throughout training for the convection equation. This indicates that Lemma 1 might hold more generally in the linear regime. In Fig. 1b we see that the Hessian does not necessarily vanish in the linear regime. Hence, if a non-vanishing gradient necessitates second-order information for training, this would sometimes also be needed in the linear regime. With reference to Table 1 the experiment in Fig. 1b validates the result of Theorem 3 in the practical setting.

From Table 2 we can discern multiple interesting patterns with regards to our previous discussions. The majority of the result are taken from previous works [13], [6], [8], but some result are also

L2RE-SCORES		1st order			Quasi-Newton	2nd order
		NTK based loss		Vanilla loss		
		PINN-NTK	**PCPINN**	**Adam**	**Adam+L-BFGS**	**GDND**
Linear	**Convection**	2.18+0	×	5.96e-2	4.19e-3	1.94e-3
	Wave	9.79e-2	1.28e-2	3.49e-1	5.52e-2	1.27e-2
Nonlinear	**Reaction**	1.44e-2	×	2.12e-2	1.92e-2	9.92e-3
	Burgers'	1.84e-2	1.42e-2	1.45e-2	1.33e-2	×
	Heat2d-LT	1.00e+0	2.11e-1	9.99e-1	1.00e+0	×

Table 2: L2RE-score for different methods when predicting various PDEs. "×" signifies that no result exists (to the best of the author's knowledge). A gray cell indicates best performance and a blue cell indicates an original result. Remaining results were taken from [6], [8] and [13].

original. Completing Table 2 was unfortunately outside the scope of this project, due to a lack of computational resources. A key observation here is that first-order methods perform well on the reaction equation and Burgers' equations – which are nonlinear – and NTK based methods perform especially well, despite the non-constancy of the NTK. This result runs completely contrary to the claim in Remark 3: a non-vanishing hessian does not seem to necessitate second-order methods. Furthermore, the NTK based approaches generally outperform Adam and Adam+L-BFGS, even in the nonlinear regime. This success cannot solely be attributed to the simplicity of PDEs, since we see that PCPINN achieves a substantial error-drop for the Heat2d-LT equation. Indeed, PCPINN outperforms other models by an order of magnitude on Heat2d-LT – which is notably a non-linear problem – indicating that this NTK based model is doing something non-trivial and effective. We see that GDND is also successful and achieves the lowest error on all PDEs where it is tested. Although these PDEs are not very complex (as is shown by the successfulness of Adam), this suggests that second-order methods are a promising tool for PINN training, and should not be overlooked. Finally, we note that PCPINN and GDND are the most successful models and that they achieve a very similar error on the one equation where they are both tested.

5 Conclusion

We conclude by addressing directly our initial questions, stated in the introduction. (a): We have seen that NTK theory begins to explain the failure modes of PINN training, by giving a relation between the training error of a PINN and the eigenvalues of its infinite-width NTK, according to equation (2). We have furthermore seen that these eigenvalues relate to the condition number of a PINN and controlling this number can thus enable effective training. The drawback of this theoretical development is that (2) only holds in a very specific and simple scenario, so in order to improve understanding, a more general analogue of this relation would be helpful. (b): Our first numerical experiment (Fig. 1) validates the theoretical result from theorem 2 and 3 even when a practical and common architecture and initialisation scheme is used in place of the idealised ones that we consider in theory. (c): Although the theoretical framework around NTK based approaches for PINN training is not completely robust, the usefulness of these methods seems to extend far beyond the simple scenario that motivates their use (see Lemma 2). Firstly, we have shown that PINN-NTK can outperform Adam and Adam+L-BFGS, even in the nonlinear regime (reaction equation). Secondly, PCPINN successfully modelled the Heat2d-LT equation, which had previously proven very challenging to get a handle on [6], [13]. Thus, based on the experiments considered here, nonlinear PDEs – and more specifically non-vanishing gradients – do not seem to necessitate the use of second-order methods. Alas, second-order methods seem a promising tool for training PINNs, but further testing is needed to understand when and how they can be used most effectively.

Since the usefulness of NTK theory seems to extend beyond the linear regime, more theoretical work is needed to better understand the role of the eigenvalues of $\boldsymbol{K}(t)$, when this is stochastic and non-constant. Such research could uncover more effective methods for PINN training, with a foundation that is theoretically sound, rather than heuristic. As indicate above, it would also be insightful to test GDND on more complex PDEs, to see if the method is as effective as indicated by the simpler situations. As it stands, both PCPINN and GDND show great promise, so it would be interesting to explore the prospect of combining the methods – training a PINN via GDND on $\mathcal{L}_{\text{PC}}$ – to see if they can complement each other to achieve a lower error than the individual methods.

References

[1] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*, 2017.

[2] 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, 2021.

[3] 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.

[4] Shamsulhaq Basir and Inanc Senocak. Critical investigation of failure modes in physics-informed neural networks. In *AIAA SciTech 2022 Forum*, page 2353, 2022.

[5] 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, 2022.

[6] Songming Liu, Chang Su, Jiachen Yao, Zhongkai Hao, Hang Su, Youjia Wu, and Jun Zhu. Preconditioning for physics-informed neural networks. *arXiv preprint arXiv:2402.00531*, 2024.

[7] Andrea Bonfanti, Giuseppe Bruno, and Cristina Cipriani. The challenges of the nonlinear regime for physics-informed neural networks. *arXiv preprint arXiv:2402.03864*, 2024.

[8] Pratik Rathore, Weimu Lei, Zachary Frangella, Lu Lu, and Madeleine Udell. Challenges in training pinns: A loss landscape perspective. *arXiv preprint arXiv:2402.01868*, 2024.

[9] Chaoyue Liu, Libin Zhu, and Mikhail Belkin. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *Applied and Computational Harmonic Analysis*, 59:85–116, 2022.

[10] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

[11] Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via hessian eigenvalue density. In *International Conference on Machine Learning*, pages 2232–2241. PMLR, 2019.

[12] Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W Mahoney. Pyhessian: Neural networks through the lens of the hessian. In *2020 IEEE international conference on big data (Big data)*, pages 581–590. IEEE, 2020.

[13] Zhongkai Hao, Jiachen Yao, Chang Su, Hang Su, Ziao Wang, Fanzhi Lu, Zeyu Xia, Yichi Zhang, Songming Liu, Lu Lu, et al. Pinnacle: A comprehensive benchmark of physics-informed neural networks for solving pdes. *arXiv preprint arXiv:2306.08827*, 2023.