Reply to Reviewer 37N7

We thank the referee for the detailed review and helpful suggestions for improving the manuscript.

The paper has been revised in response to reviewer suggestions and to correct minor errors. Significant changes are highlighted in blue. Further, our response to the questions and suggestions are detailed below.

Comment 1. The main result states that the NTK change by the order ϵ^{-2}. But I am a little confused by what this statement means. Does it mean that before and after training, the NTK will change by order ϵ^{-2}? Or it simply means that before training, the NTK will change because of the change in the loss function of order ϵ^{-1}?

Reply This was not well explained in the text, and the confusion you point out may be linked to the question raised in Comment 2. below. The analysis in the paper is inspired by asymptotic homogenization theory, where one studies a *sequence of problems*, indexed by ϵ, to try and understand their limiting behavior as ϵ vanishes.

So, for a *fixed* ϵ (say ϵ_1), the NTK matrix simply changes as a function of time t according to its definition in Eq. (14) (here we are assuming the network parameters evolve according to a continuous time gradient flow, as in Eq. (11); see the additional discussion on this point below). Suppose then that one measures the Frobenius norm at time T to be some number C:

$$\|K^{\epsilon_1}(T)\|_F = C.$$

If one then considers again the same problem but with $\epsilon_2 = \epsilon_1/2$. Then the main result simply says that one can expect

$$\|K^{\epsilon_2}(T)\|_F \approx 4C.$$

The punchline is that problems with large scale separation can be expected to have larger NTK matrices than those with moderate or mild scale separation.

Comment 2. The argument that NTK changes as ϵ^{-2} "implies that... training the neural network with gradient descent based methods to achieve an accurate approximation of the solution to the PDE becomes increasingly difficult" is tenuous. I do not see why the claim is implied, the authors need to explain this in detail. After all, I think this claim cannot be supported by the theory of the paper. NTK is only relevant to the case when the learning rate is vanishingly small and is not relevant when the learning rate is large. In fact, this work might imply that it is better to use a large learning rate, a point that is not discussed in the main text.

Reply Thank you for raising an issue with our claim that "training...becomes increasingly difficult", as this critical argument was not properly developed in the previous version of the paper. We have updated the text to include a discussion of this point in some detail; please see the discussion in the new Section 3.1, as well as the additional numerical experiments in Sections 4.3 and 4.4 in support of our main claim.

The theory developed in the paper builds on the previous analyses of PINN training from Wang et al. (SIAM J. Sci. Comput., 2021), Wang et al. (Comput. Meth. Appl. Mech. Engr., 2021) and Wang et al. (J. Comput. Phys., 2022) in which PINN training was analyzed from the lens of a vanishingly small learning rate. This is natural to consider, since the standard optimizers used for PINNs (such as Adam and L-BFGS) are only guaranteed to find local, rather than global minimizers. A relatively large learning rate risks missing out on local optima and can be unstable from the point of classical numerical analysis; see section 4.2 of Wang et al. (SIAM J. Sci. Comput.) for a nice discussion of this latter point.

Comment 3. It seems that the problem of training identified in this paper already has a solution. The authors should discuss the relevant work of https://arxiv.org/abs/2006.08195. The problem with PINN is due to the fact that the assumed model activation function is monotonic (Eq. 18), whereas learning the right solution requires a periodic function. It seems to the audience that the problem simply comes from the fact that the inductive bias of the model does not match the problem. Including a discussion of the result in https://arxiv.org/abs/2006.08195 and performing numerical experiments to either validate or disprove what I state above will greatly improve the manuscript.

Reply Thank you for this suggestion to improve the manuscript. We included a comment in the conclusions section that highlights the possibility of selecting alternative, problem-specific activation functions that can improve the network's inductive biases.

The theory in the paper characterizes PINN performance for divergence-form elliptic problems with general Dirichlet conditions; the examples in section 4.2 were chosen to illustrate poor performance (for equations of the form (28)) even on relatively simple, one-dimensional problems. For simplicity the boundary conditions were taken to be homogeneous Dirichlet, which indeed implies in this case that the solution is periodic (it equals zero at both endpoints).

Hence, the activation function proposed in Ziyin et al. (NeurIPS 2020) may indeed improve the PINN performance equations of the form (28) ("Darcy" form) for this particular case; they may also further improve the results for the regression problem and Poisson problem, although we note that reasonable approximations for these cases were already achieved with the monotonic, hyperbolic tangent activation function.

Nevertheless, one would hope that PINN approximation of Darcy form equations can succeed in more general situations where one doesn't have the relative simplicity of periodic boundary conditions. Also, in practice one can encounter multiscale equations of the form (3) where the microscale is not periodic (i.e. the coefficient $a^\epsilon(x)$ is not a periodic function); in this case alternative activation functions to the ones proposed in Ziyin et al. would be needed. For this reason we focus here on the most commonly used PINN activation functions.

Finally, we also note that the activation functions proposed in Ziyin et al. appear to be designed for extrapolation of temporally varying functions, which is fundamentally different that the boundary value problems considered here.