Reply to Reviewer c8m4

We sincerely thank the referee for the thorough review and the myriad suggestions to revise and improve 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. Claim "We show that if the coefficient in the elliptic operator contains frequencies on the order of $1/\epsilon$, then the Frobenius norm of the neural tangent kernel matrix associated with the loss function grows as $1/\epsilon^2$" is not supported.

First, Theorem 3.2 is more restrictive, since it applies only to the periodic functions of the special form $a_\epsilon(x) = a(x/\epsilon)$.

Second, I am not sure that the claim as stated is accurate. For example, lets consider $a_\epsilon(x) = \epsilon\cos(2\pi x/\epsilon)$. For any fixed ϵ this term contains frequencies of order $1/\epsilon$, yet the neural tangent kernel is not going to contain terms of order since the absolute value of the derivative is bounded by 2π.

I suggest somewhat reformulating the sentence in the abstract or providing a stronger theoretical result.

Reply Thank you for pointing out this oversight in our formulation. Indeed, to make the claim more rigorous, one needs to stipulate that the coefficient a^ϵ is *uniformly elliptic* in ϵ. In one dimension, this means we require $\forall\epsilon > 0$ that

$$\lambda \le a^\epsilon(x) \le \Lambda \qquad \forall x \in \mathbb{R},$$

where λ and Λ do not depend on ϵ. Indeed, in the case $a^\epsilon(x) = \epsilon\cos(2\pi x/\epsilon)$, the elliptic problem described in Eq. (15) in the paper is not well posed as $\epsilon \to 0$.

This is guaranteed if we start with a coercive and bounded 1-periodic function $a(y)$ and then set $a^\epsilon(x) = a(x/\epsilon)$. We fixed the presentation in the abstract and in the NTK theory section (Section 3.1) to make this clear.

Comment 2. Theoretical result is too narrow. It is not clear whether it is applicable in a broader context.

As I understand, the authors want to study PINN for elliptic PDEs with oscillating coefficients. Yet the theoretical result applies only to the specific class of PDEs and specific loss. Here are several suggested directions that can make the article applicable to a broader set of problems:

(a) What if is such that the derivative is bounded? Is it going to affect the training or not? For example, we can consider a rescaled version of equation (28) from the article:

$$-\frac{d}{dx}\Big(\frac{\epsilon}{2.1+2\sin(x/\epsilon)}\frac{d}{dx}u^{\epsilon}(x)\Big)=f^{\epsilon}(x)\epsilon$$

Authors observed poor training with a version that is not rescaled. After the rescaling tangent kernel does not contain $1/\epsilon^2$. Will this simple rescaling improve training?

(b) Consider a non-conservative form elliptic equation

$$a_{\epsilon}(x)\frac{d^2}{dx^2}u(x)=f(x). \qquad (*)$$

Now tangent kernel again is well defined for $\epsilon\to 0$ as long as $a_\epsilon(x)$ is uniformly bounded. Is it right, that the training will be efficient for strongly oscillating a_ϵ?

(c) What if we use the variational form (e.g., as in the Deep Ritz method)? In this case, we do not need to consider the derivative of $a_\epsilon(x)$. Does it mean that the neural tangent kernel is well-defined? Are we going to see poor performance of PINN?

Reply (a) This is an excellent question, and we expanded our numerical experiments to characterize the results of rescaling in this way.

The rescaling is equivalent of course to multiplying the PDE residual portion of the PINN loss function by ϵ^2, and it is indeed quite sensible, as it cancels out the $1/\epsilon^2$ scaling the K_{uu} NTK matrix subblock.

In our expanded numerical results section, we observe that the Darcy PINNs without this rescaling simply do not satisfy the correct Dirichlet boundary conditions; the imbalance between the two different terms in the loss function (corresponding to the PDE residual and the boundary conditions) is too great. This imbalance is already an issue for Poisson-type equations, but the problem is exacerbated in the Darcy case by the $1/\epsilon^2$ NTK scaling.

As discussed extensively in Wang et al. (J. Comput. Phys., 2022), both the magnitude and the distribution of the eigenvalues of the K_{uu} and K_{bb} NTK matrix subblocks are closely connected to a PINNs performance.

Although rescaling the loss function of course changes the magnitude of the eigenvalues of the K_{uu} subblock, indeed it does not change their distribution. As shown in Figure 3, there is a large discrepancy between the principal eigenvalue of the K_{uu} subblock and the others, which is a characteristic sign that the gradient descent dynamics (Eq. (11)) are stiff. In our new numerical tests of rescaling the loss function by ϵ^2, we still observe poor training behavior. While the Dirichlet boundary conditions are satisfied at a larger ϵ value (which is *not* the case if the loss function is not rescaled), this does not hold at smaller values of ϵ, which we attribute to the NTK matrix spectrum. Please see the new Section 4.4 for more details.

(b) This is a great point; in non-conservative form, the NTK matrix subblock K^{ϵ}_{uu} indeed does not scale as $1/\epsilon$ as it does in the conservative-form case. In the one-dimensional case, however, because $a_\epsilon(x)$ is coercive, we can simply recast $(*)$ as

$$\frac{d^2}{dx^2}u(x)=f(x)/a_\epsilon(x)=:g_\epsilon(x)$$

and we are left with a multiscale Poisson-type equation, as considered in the numerical examples and previously analyzed by Wang et al. (see references). In higher dimensions, however, we note this observation is no longer valid.

We modified the abstract and the text to emphasize that the paper develops theory for the conservative form case. Because equations of this form commonly arise due to physical conservation laws, we hope this is of sufficient interest to stand on its own.

(c) This is another excellent point; the focus of the present work is on PINNs, however, in alternative NN based approaches to solving PDEs based on energy-minimization, such as the Deep Ritz method, the derivative of $a_\epsilon(x)$ indeed does not appear. Li, Xu, & Zhang (Commun. Comp. Phys., 2020) applied the Deep Ritz method in combination with specialized activation functions to multiscale elliptic problems in divergence-form considered here (as well as to nonlinear problems), with mixed results.

To the best of the authors knowledge, a NTK theory has not been developed for the Deep Ritz method, but this is an interesting avenue for future research.

We added some discussion along these lines to the Conclusions section of the manuscript.

Comment 3. Experimental evidence is not sufficient.

The approach taken by the authors in Section 4.2 is quite elegant. Yet, the results are not convincing. I suggest repeating the same experiments (learning in L^2 setting, PINN+Poisson, PINN+Darcy) for a range of ϵ and showing errors for different methods.

Besides, I suggest adding at least rescaled version of equation (28) as explained in 2. above.

Reply In retrospect, it was foolish not to include a study for decreasing ϵ in the original submission. Thank you very much for the suggestion.

We indeed carried out a sequence of additional numerical experiments for the Darcy, Poisson, and L^2 regression cases and reported the results in the new Section 4.3. We also added a study as a function of ϵ of the Darcy PINN rescaled by ϵ^2, as suggested in Comment 2a; please refer to the new Section 4.4 for the results.

Comment 4. Minor suggestions.

(a) In equation (6), it is not clear right away that the index k refers to the coefficient with frequency k. Please consider clarifying this part.

(b) In Figure 1, a logarithmic scale in y seems more appropriate.

(c) Neural tangent kernel should be symmetric positive semidefinite. Yet in Lemma 3.1 $K_{ub} \neq K_{bu}^T$. It seems there is a typo in the coefficients.

(d) Page 6 "Although it does not hold for the ReLU function $\sigma(x) = \max(0, x)$, these are of course not suitable for NN-based solutions to PDEs with second order (or higher) differential operators." Perhaps, only for a strong form of PDE. Neural Networks with ReLU functions form a nonlinear approximation space of piecewise linear functions with variable positions breaking points. They cover the space of linear finite elements.

Reply (a) Clarified. Thanks!

(b) Figure 1 has indeed been switched to now have a log-scale on the y axis. The frequency principle is still apparent, but because things are visually crowded, the authors have a slight (but not strong) preference to revert to a non-log scale, however, this is but a minor point of contention.

(c) In general, while the K_{uu} and K_{bb} subblocks of the NTK matrix for PINNs are symmetric and positive semi-definite, the K_{ub} and K_{bu} subblocks are not square unless the collocation points in the domain interior N_c and boundary N_b are equal. Because of the scaling of the two different terms in the loss function (equation (8)), the K_{ub} subblock ought to be the scaled transpose of K_{bu}.

(d) We modified the sentence in the text to specify NN based solution methods, such as PINNs, that are based on the strong formulation of PDEs. Thanks for pointing this out.