Journal of Computational Science

Title: Deep Rayleigh quotient iteration (DRQI) for high-precision numerical solutions of eigenvalue problems for differential operators

Dear Editors and Reviewers:

Thank you for your letter and for the reviewers' comments concerning our manuscript entitled "*Deep Rayleigh quotient iteration (DRQI) for high-precision numerical solutions of eigenvalue problems for differential operators*". Those comments are all valuable and helpful for revising and improving our paper. We have studied the comments carefully and revised our manuscript accordingly. The response to the reviewers' comments is below in Blue (also see the new version of the manuscript).

Reviewer 1:

Q1 Inconsistent results with the provided code

R1:

Many thanks to Reviewers for their thorough and rigorous evaluation of our work, including rerunning our code and providing detailed diagnostics. Their insightful comments are very helpful to improve our studies and to access the robust results.

The fixed random seeds are employed in the update studies to address reproducibility concerns, and we have re-conducted and re-run all experiments. The statistical results over multiple seeds and the range of variation to reflect the sensitivity of each method to initialization are reported in the revised manuscript. Reviewers and researchers may check the log files and reproduce the related results by following the recorded parameters in the updated Github repository at https://github.com/LokimuKH19/DRQI . In general, a more cautious stance on comparing algorithmic superiority are adopted in the reviesed manuscript, focusing on the robustness and statistical performance across multiple trials.

Q1.1 1D SGD DRQI with 10,000 iterations, learning rate 0.01, $\omega = 0.77$ - My computer, lambda error is 4.6×10^{-4}, reported in the article 2.8×10^{-6}. DRM under the same configuration (10,000 iterations, SGD with learning rate of 0.01) gives 9.3×10^{-6}, reported in article as 5.8×10^{-5}.

R1.1:

Thanks to the reviewers for the rigorous and scientific work on our studies. By double checking, we confirmed that the previous version was based on a non-deterministic run and did not reflect a fixed random seed, which led to variation. , we updated the procedure as mentioned above with the reproductable ones in the revised manuscript.

The Fig. 2 and Fig. 5 in the revied manuscript present the update results using the consistent settings, which match the observed performance. Additional reproducible runs and logs can be found in the updated GitHub repository.

Q1.2 In 1D, using the Adam optimizer, DRM yielded an error of 2.6×10^{-6}, the article reports 2.1×10^{-5}. IPMNN with Adam gave 5.97×10^{-7} on my computer, whilst the reported value in the article is 3.3×10^{-6}. The DRQI seems highly sensitive to initialisation, I ran it several times and at various points the final error gave 0.155 with

Adam's learning rate of 0.001, ω = 0.5 and 10,000 iterations, whilst the paper reports 7 orders of magnitude lower. The lowest I observed was of the order of 10^{-7}-10^{-6}.

R1.2:

Thank you again for carefully verifying the results using the Adam optimizer. We have thoroughly reviewed our experiment logs and found that, due to a documentation oversight, the learning rate used for the DRQI configuration was incorrectly listed as 0.001 in Table 1, while the actual setting was 0.01. The mistake has been corrected this in the revised manuscript.

As shown in the updated Fig 5, DRQI indeed demonstrates sensitivity to initialization at smaller learning rates. However, DRQI consistently reaches absolute eigenvalue errors below 10^{-7} when using the appropriate learning rate (e.g., 0.01), confirming its convergence under the settings reported. These results are now statistically validated across multiple seeds and reproducible via the code and logs available in the repository.

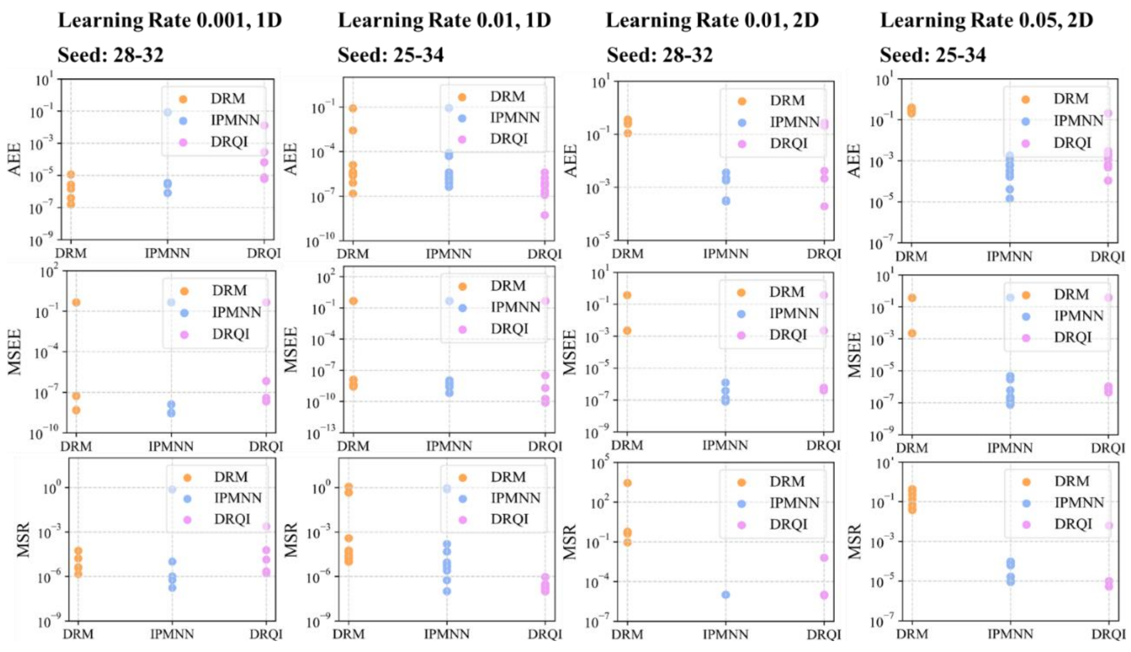


Fig 5. The error distribution of 1D and 2D eigenvalue problem (21), with the Adam optimizer

Q1.3 In the 2D cases with Adam, the DRM seemed to not be behaving correctly at all, suggesting an implementation issue with the eigenvalue reaching a lower value than its theoretical minimum. This would most likely be because the code appears to reuse the same points across iterations leading to a significant overfitting issue, which should not be done in the DRM as the structure of its loss function makes it highly susceptible with non-random integration. This would also appear to be the case, to a lesser extent, in the IPMNN method. By changing the sampling to quasi-Monte Carlo, the 2D implementation of DRM with the Adam optimizer improved by two orders of magnitude, and by slighting adjusting the learning rate, the results were marginally better results than the DRQI in the same experiment. Similarly, DRM with quasi MC in 3D produced a final error around half of the reported value DRQI in the same experiment (cited as 8.9×10^{-4} for DRQI), whilst on my computer, DRQI only obtained errors of 5×10^{-3}.

R1.3:

As mentioned above , the loss structure in DRM makes it highly sensitive to non-random sampling, which can lead to artificially low eigenvalue approximations due to overfitting the fixed training data.We further verified that the approach switching from fixed Latin Hypercube Sampling (LHS) to quasi-Monte Carlo (QMC) sampling indeed alleviates this issue in 2D cases. This adjustment improved the DRM's performance significantly and led to a better match with the theoretical eigenvalue in some cases. The related log file have been uploaded in the Github link. Related expreiments have been reported in Section 3.1.1.3.

Additionally, the enforcement of the early stopping criterion based on the Mean Square Residual (same as the PDE loss function used in PINNs) helps to mitigate the overfitting tendency to some extent. Specifically, the training process is terminated when both the residual loss and its change across iterations fall below certain thresholds. This approach serves as a soft regularization mechanism and prevents the optimizer from excessively fitting the sampled data while deviating from the true solution structure.

Q1.4 In 2D with Adam at 5,000 iterations, I found both the errors of DRQI and IPMNN to be around 3×10^{-4}, being marginally worse in the DRQI than the author's reported figure and two orders of magnitude better in IPMNN.

R1.4:

The discrepancy in DRQI's error may stem from the differences of the initialization and random seed, thus we re-conducted the experiments under fixed seeds and carefully documented the performance variation across runs, the detailed information are unpdate in Section 3.1.1 of the reviesed manuscript. These results from Fig. 3 and Fig. 5 demonstrate that the eigenvalue errors for both IPMNN and DRQI reached on the order of 10^{-3} in 2D using the Adam optimizer and 0.01 as the initial learning rate, and triggered the early stop criterion with less than 500 iterations. When the initial learning rate comes to 0.01, the IPMNN failed to satisfy the early stop condition during training but DRQI managed to do so. In addition, there remains observable variability depending on initialization. More detialed information can be found in the updated repository.

Q1.5 In the 3D problems, the DRQI had strange behaviour, with the Adam optimizer sending the DRM method's loss to minus infinity (suggesting a severe implementation or integration problem) whilst the DRQI sent the loss to 0 and then didn't move.

R1.5:

Thank you for highlighting the peculiar behavior observed in the 3D experiments. After careful investigation, we confirmed that earlier versions of our code for DRM could exhibit divergence (loss approaching minus infinity or infinity) when using the Adam optimizer combined with LHS and an aggressive learning rate. This instability, which we reported in the revised Fig. 4 and recorded in the repository, was found to be closely related to the sampling strategy employed.

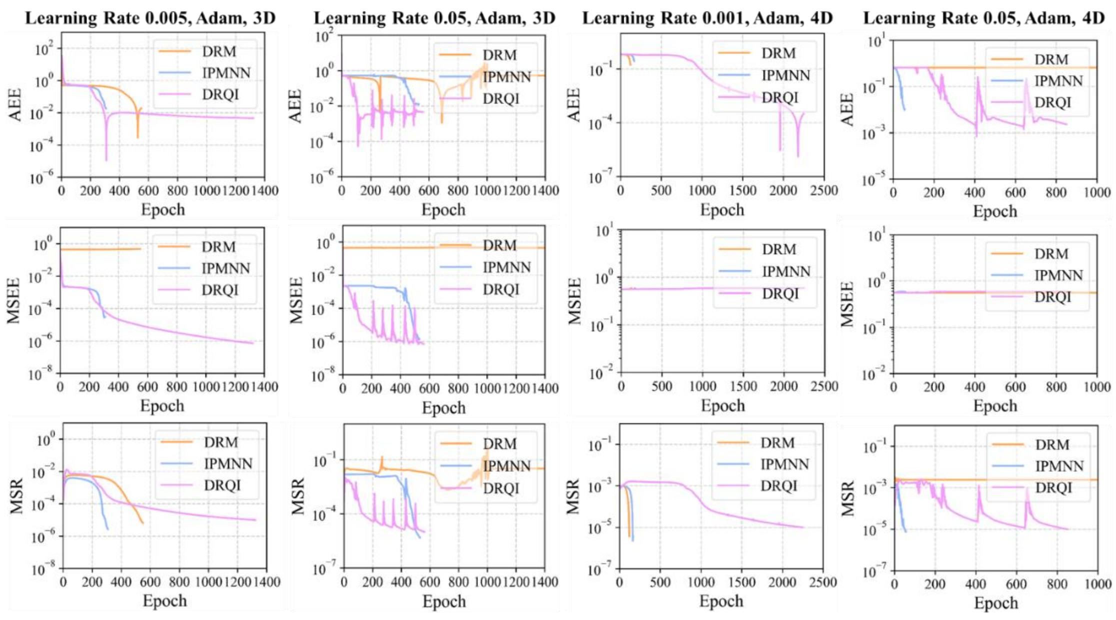


Fig 4. The convergence curves of each model for solving 3D and 4D eigenvalue problem (21), with random seeds 39 for 3D, 30 for 4D

A detailed analysis of this phenomenon and our attempt to address it using QMC sampling is presented in Section 3.1.1, see in the revised Fig.10:

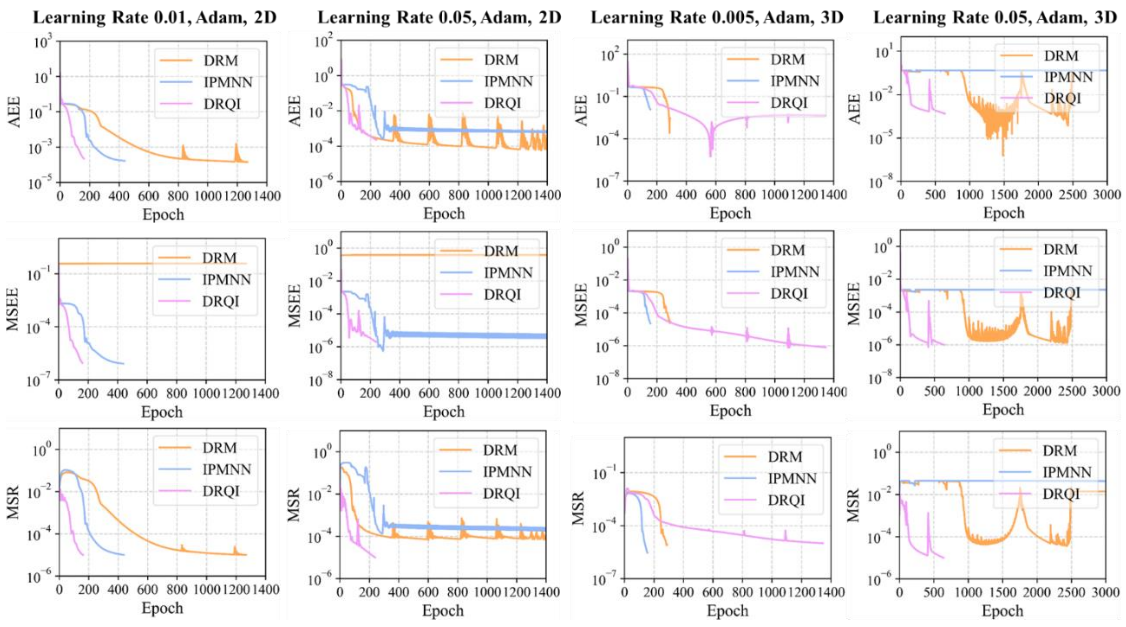


Fig.10 The convergence curves of each model for solving eigenvalue problem (21) by using QMC

sampling, with random seeds 29 and 41 for 2D and 3D results respectively

2 General comments

Q2.1 I find the article in general to be quite poorly written, with ambiguous wording, uncommon terminology and grammatical errors throughout. The most significant issue with respect to the language is that I find the descriptions of the methodology to be incredibly difficult to understand, and only upon reading the ideas presented as an algorithm did I begin to capture the method.

R2.1:

We thank the reviewer for the constructive feedback regarding the clarity and presentation of our methodology. In response, we have significantly revised the language throughout the manuscript, with particular emphasis on Section 2.3. Specifically, we have:

- Rewritten the key steps of the method in a clearer and more concise form;
- Reorganized the explanations around DRQI to improve readability and logical flow;
- Standardized terminology and removed ambiguous or uncommon phrasing;
- Corrected grammatical issues to enhance overall readability.

We hope these revisions make the proposed method more accessible and easier to follow.

Q2.2 Their method and discussion lacks any appropriate functional analytic framework. They discuss differential operators defined on a Hilbert space, but never specify the Hilbert spaces themselves. Furthermore, they claim that their mappings are from the Hilbert space to itself, which is in disagreement with the strong formulations that they employ. In particular, there is no suggestion as to the norm in which their approximated eigenvectors should be approaching the exact ones. As they work within a strong formulation, should this be an H2-type norm?

R2.2:

The revised manuscript now clarifies the functional analytic setting in Section 2.1. Specifically, we consider a linear differential operator $\mathcal{L}$ defined on a connected domain $\Omega \subset \mathbb{R}^d$with Lipschitz boundary, and formulate the eigenvalue problem under the strong form as

$$\begin{cases} \mathcal{L}u = \lambda u, \boldsymbol{x} \in \Omega \\ Bu = 0, \boldsymbol{x} \in \partial\Omega \end{cases} \tag{1}$$

Here, the Hilbert space is taken to be $H = L^2(\Omega)$, and the domain of the operator is $\mathcal{D}(\mathcal{L}) \subset H^2(\Omega)$, such that all differential operators and boundary conditions are well defined. Since the eigenvalue problem is treated in strong form, we ensure that the neural network ansatz for the eigenfunction lies within $\mathcal{D}(\mathcal{L})$, which is consistent with requiring square-integrable second derivatives. This is automatically satisfied by the use of smooth neural network approximators (e.g., feedforward networks with tanh activation functions) and the fact that all gradients are computed via automatic differentiation. Thus, the approximated eigenfunctions reside within $H^2(\Omega)$ and the formulation is mathematically well-posed.

Additionally, all reported quantities (loss functions, eigenvalue errors, residuals) are computed using the standard L^2-norm, and this has been clarified in the revised text. We acknowledge the reviewer's insight regarding possible alternative norms (e.g., H^2-type), but given the structure of the operator and the convergence behavior we observe, the L^2-based metrics remain a reasonable and practical choice in this setting.

To cleairfy, we made these modifications to the revised Section 2.1.

"The eigenvalue problem is posed in its strong formulation, where the operator $\mathcal{L}$ involves second-order partial derivatives, as shown in Equation (1)."

"Accordingly, $u(\boldsymbol{x})$ is assumed to lie in the Sobolev space $H^2(\Omega) \cap \mathcal{V}$, where $\mathcal{V}$ encodes the boundary constraint (e.g., $\mathcal{V} = H_0^1(\Omega)$ in case of homogeneous Dirichlet boundary conditions represented by B). The operator $\mathcal{L}$ is thus regarded as a mapping from $H^2(\Omega) \cap \mathcal{V}$ into $L^2(\Omega)$. Unless otherwise specified, all residuals and error measures in this work are computed in the standard L^2-norm."

Q2.3 The authors state that their model can deal with "arbitrary boundary condition[s]" in section 2.1, but their approach doesn't seem able to handle, for example, Neumann conditions.

R2.3:

The sentence stating that our framework is capable of handling "arbitrary boundary conditions" refers to the established use of hard-constraint strategies in PINN literature, where boundary conditions are incorporated directly into the neural network ansatz via function multiplication.

Specifically, a widely adopted approach is to construct the solution in the form:

$$u(\boldsymbol{x}, \boldsymbol{\theta}) = A(\boldsymbol{x})\mathcal{N}_{\boldsymbol{\theta}}(\boldsymbol{x}) + b(\boldsymbol{x}) \tag{4}$$

where $A(\boldsymbol{x})$ vanishes only on the boundary (i.e., $A(\boldsymbol{x}) = 0$ for $\boldsymbol{x} \in \partial\Omega$), and $b(\boldsymbol{x})$ is any function satisfying the boundary condition, as shown in Equation (5). (We added "$A(\boldsymbol{x}) \neq 0, \boldsymbol{x} \in \Omega$" to avoid the non-trivial solution.)

This formulation ensures that the resulting solution $u(\boldsymbol{x}, \boldsymbol{\theta})$ automatically satisfies the prescribed boundary condition, regardless of the specific form of $\mathcal{N}_{\boldsymbol{\theta}}(\boldsymbol{x})$. For Dirichlet boundary conditions, this is straightforward. For **Neumann boundary conditions**, a similar approach applies, where $b(\boldsymbol{x})$ is chosen to satisfy the boundary flux and $A(\boldsymbol{x})$

is constructed so that the derivative vanishes on the boundary.

For example, consider the 1D Neumann problem with $\partial_x u(0) = \partial_x u(1) = 0$. A hard-constraint construction could be:

$$u(x,\theta) = x^2(x-1)^2\mathcal{N}_{\boldsymbol{\theta}}(x) + \text{constant}$$

or

$$u(x,\theta) = \sin^2 \pi x \cdot \mathcal{N}_{\boldsymbol{\theta}}(x) + \text{constant}$$

or other variants where the architecture ensures the Neumann conditions are embedded into the form of the network and its derivatives.

This type of boundary-aware neural ansatz has been studied in detail in:

- Lu, L. et al. Physics-informed neural networks with hard constraints for inverse design. SIAM J. Sci. Comput., 43(6), B1105–B1132, 2021.
- Liu, S. et al. A unified hard-constraint framework for solving geometrically complex PDEs. NeurIPS, 35, 2022.

To clarify this point, we added the above references in Section 2.1.

"…the network's output is subjected to a common hard constraint method [25, 26] to satisfy boundary conditions automatically the boundary conditions when using the network to approximate the eigenfunction. In particular, the guessed function at each iteration step is expressed as shown in Equation (4)."

Q2.4 Their Algorithm 2 expresses the loss in terms of the current and previous iteration of u when the relax ation factor ω is used, as both iterations depend on their respective values of the trainable parameters, it should be made clear exactly what derivative with respect to θ is being taken.

R2.4:

In the revised Equation (18), the semi-implicit loss function is now defined as:

$$l(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x}\in\mathcal{X}}\{\mathcal{L}u_k(\boldsymbol{x};\boldsymbol{\theta}) - (\lambda+1)[\omega u_k(\boldsymbol{x};\boldsymbol{\theta}) + (1-\omega)u_{k-1}(\boldsymbol{x})]\}^2 \tag{18}$$

where $u_k(\boldsymbol{x},\boldsymbol{\theta})$ denotes the current eigenfunction approximation parameterized by the trainable parameters $\boldsymbol{\theta}$, while $u_k(\boldsymbol{x},\boldsymbol{\theta})$ is treated as fixed and detached from the computation graph. Consequently, the gradient $\nabla_{\boldsymbol{\theta}} l(\boldsymbol{\theta})$ is only taken with respect to $u_k(\boldsymbol{x};\boldsymbol{\theta})$, and the term $u_{k-1}(\boldsymbol{x})$ serves as a constant reference in the forward pass.

We have clarified this point in the revised manuscript to ensure that the gradient flow is unambiguously defined. First, the parameters of each u is added to Equation (18). Second, the following description is supplied to the description of the equation:

"The gradient $\nabla_{\boldsymbol{\theta}} l(\boldsymbol{\theta})$ is computed solely with respect to u_k, and u_{k-1} is held constant."

Q2.5 Judging by their code, fixed integration points are being used which can (and according to the experiments I did with their code, do) introduce significant overfitting problems. Yet there is no means of measuring overfitting (validation sets etc.). In, for example, figure 5, the DRM may have had an incredibly good solution at the 500th iteration when the eigenvalue error crosses zero, but as they made no attempt to identify the onset of overfitting, it is unknown if this is the case. In fact, the authors make a comment that there is no stopping criteria for DRM, but the most obvious answer is a validation set here. Somewhat related, the DRQI also crosses over zero and does have a stopping criterion, which appeared to have no use in stopping training at the correct point.

R2.5:

We thank the reviewer for the valuable observation regarding overfitting and stopping criteria in our work. Indeed, as noted, fixed integration points can introduce overfitting, and the lack of explicit validation sets in DRM poses challenges for timely stopping.

To address these concerns, the revised DRQI method employs a dual stopping criterion combining both an absolute PDE residual tolerance (ε_1) and a residual change tolerance (ε_2), as detailed in Algorithm 2. These contents has been added to Section 2.3:

"As for the stopping criteria, the main squared residual (MSR) $\mathcal{L}_{\text{eq}}^2 = \mathbb{E}_{x\in\mathcal{X}}[\mathcal{L}u(\boldsymbol{x}) - \lambda u(\boldsymbol{x})]^{\mathbf{2}}$ is employed to avoid the risk of overfitting to the target eigenvalue and aligns with the fundamental goal of solving the PDE accurately. Specifically, the stopping condition requires that the MSR not only falls below a threshold ε_1 but also that the improvement between consecutive iteration is sufficiently small (below ε_1). This design prevents premature stopping due to transient low residuals and avoids excessive training beyond convergence, thus mitigating overfitting despite the use of fixed sampling points with sufficient sample size."

Empirically, this mechanism proves effective: for example, in the 3D problem with learning rate 0.005 shown in the revised Fig. 4. In this particular negative example with a small learning rate, DRM and IPMNN converge relatively quickly and trigger the ε_1 stopping condition early. However, since the ε_2 condition is not immediately satisfied, training continues until ε_2 is also met, resulting in a staggered stopping process. On the other hand, DRQI converges more slowly, and when it finally reaches the ε_1 threshold, the ε_2 condition is also satisfied almost simultaneously, leading to a more immediate termination of training.

Q2.6 The authors make no attempt to quantify the error of the eigenvectors in their work. They report figures for errors of the eigenvalues and (in few cases) show diagrams of the error (with some, such as the errors in Fig. 3, on an incredibly inappropriate scale, meaning no detail can be seen). As their method is based on strong formulations, one would expect any good method to imply low H2 errors, or some similar norm.

R2.6:

In the revised manuscript, we have addressed this issue through a more comprehensive analysis of the eigenfunction approximation quality.

Specifically, we now include both the original and an improved version of the Mean Squared Eigenfunction Error (MSEE) to evaluate the discrepancy between the predicted and theoretical eigenfunctions and added the purpose of employing it:

"This metric captures the mean squared difference between the predicted and theoretical eigenfunctions. The squared form is used to maintain consistency with the MSE loss (18) employed during training and emphasizes larger deviations."

In addition, we introduce density-based diagnostics of eigenfunctions and plot them in the histograms, as shown in Fig 8 and other results obained by the similar method, which help visualize the structural fidelity of the learned eigenfunctions, particularly in higher dimensions where pointwise error visualization becomes intractable. These density plots reveal important distinctions between convergence types (Type-A vs. Type-B), and allow cross-validation of the MSEE values.

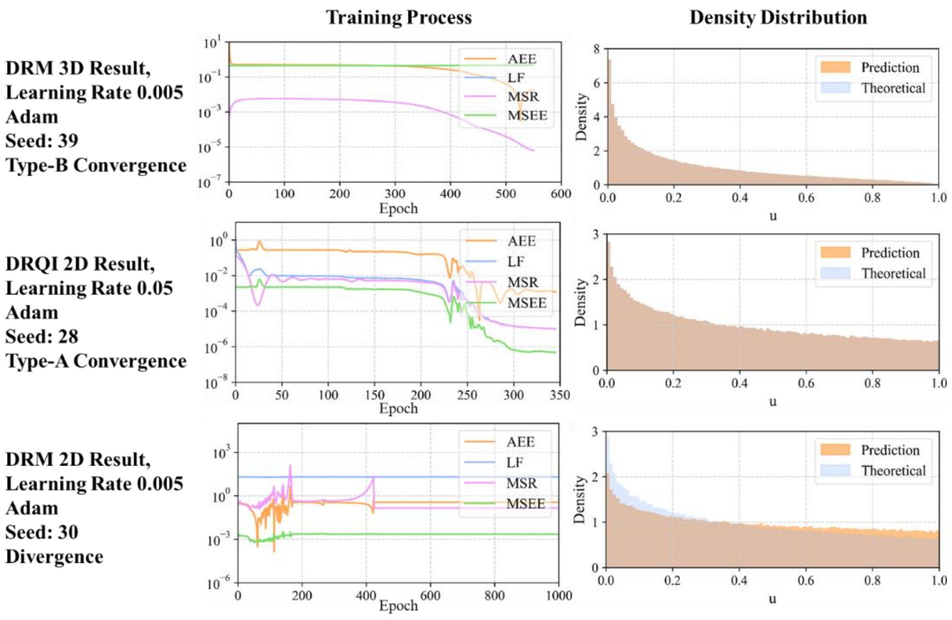


Fig 8. Characteristic density distributions under different convergence behaviors in multi-dimensional cases, where $N = 500000$ is employed in the density plots

Furthermore, the MSR which reflects the equation-level consistency of the predicted eigenfunction with the underlying PDE, also serves as an indirect but meaningful measure of eigenfunction quality. We also added the following description of MSR:

"This measures the degree to which the predicted eigenpair satisfies the original differential equation. Again, the squared form is used to align with the objective function minimized during training. In addition, this error serves as a central indicator for the quality of learned solutions, especially in the absence of analytical references in most of the practical engineering scenarios."

Q2.7 It is very unclear if the comparisons with other methods are fair. The most obvious question surrounds the learning rates- each method is based on very different loss functions with very different interpre tations, so there is no reason to believe that the optimal learning rate with the DRQI would be the same one with the other methods.

R2.7:

Thank you for raising this important point. Indeed, in the revised version of the manuscript, we have explicitly discussed the impact of different learning rates on the convergence behaviors of each method in Section 3.1. Given that DRQI, DRM, and IPMNN are based on distinct loss functions and optimization landscapes, we did not assume that a single learning rate would be universally optimal. Instead, we empirically explored how learning rates influence each method and reported the corresponding performance under multiple configurations. The variability in convergence due to learning rate selection is also analyzed in the accompanying figures (e.g., Fig. 2~4), where we observed that DRQI generally prefers a relatively larger learning rate in lower-dimensional settings, while DRM and IPMNN exhibit more stable behavior under smaller values. This comparative analysis was designed to be as fair and transparent as possible.

Q2.8 The entire argument of Section 3.3 seems to be that initializing near a non-minimal eigenvector will lead to convergence to this non-minimal eigenvector. This is a problem, because it suggests that the model is highly susceptible to initialization and may not converge to the minimal eigenvector when it is wanted. If this is not the case, then the higher order eigenvectors will be unstable, and obtaining them amounts to being lucky enough to stop training at the right time.

R2.8:

We appreciate the reviewer's insightful comment. As noted, the original Section 3.3 aimed to explore the convergence behavior of DRQI with respect to different initialization strategies. While our current implementation can target higher-order eigenfunctions via pre-training and adjusting the loss formulation, we acknowledge that we have not yet established a systematic or fully stable approach for ensuring convergence to a specific non-minimal mode. In contrast, as demonstrated in Sections 3.1 and 3.2, when initialized randomly, the model reliably converges to the minimal eigenvector, which is desirable in most practical scenarios. Given that the findings in Section 3.3 remain preliminary and somewhat tangential to the main focus of this work—which is the evaluation of algorithmic performance for the fundamental mode—we have decided to omit that section from the revised version to maintain clarity and coherence in the manuscript.

Should further progress be made in this direction, we will make the corresponding experimental results and implementation updates available in the project repository.

Q2.9 On page 23, the authors state that the phase of the solution is unimportant as the space is isotropic. This seems inconsistent with the fact that the potential induces anisotropy. Also, in the case when the potential function is zero, the minimal eigenvalue is zero, corresponding to constant functions, not cos(x).

R2.9:

We thank the reviewer for the valuable comment. While Section 3.3 has been removed from the revised manuscript for the sake of focus, we would like to clarify the point raised. In the corresponding numerical experiments, we considered a potential function of the form $v = 0.1\sum_{i=1}^{d}\cos x_i$ under periodic boundary conditions:

$$\begin{cases} -\Delta u + vu = \lambda u, \boldsymbol{x} \in [0,2\pi]^d \\ u(\boldsymbol{x}+2\pi\boldsymbol{q}) = u(\boldsymbol{x}) \\ \dfrac{\partial u(\boldsymbol{x}+2\pi\boldsymbol{q})}{\partial \boldsymbol{x}} = \dfrac{\partial u(\boldsymbol{x})}{\partial \boldsymbol{x}} \end{cases}$$

Where $\boldsymbol{q} \in \mathbb{Z}^d$. Although the potential v induces a form of weak anisotropy, its symmetric formulation ensures that all spatial directions are equally perturbed. As such, the resulting eigenfunctions exhibit a form of dimensional symmetry, and the observed phase shift does not alter the energetic content of the solution. Our original remark on "isotropy" was imprecise and has been revised to reflect that the system exhibits symmetric behavior across dimensions, rather than true isotropy.

Regarding the statement about the minimal eigenvalue being zero, we fully agree that constant functions solve the eigenvalue problem when $v = 0$, leading to a zero eigenvalue. And such constant solutions represent trivial eigenfunctions. While we understand the theoretical concern raised, our work focuses on practical settings where trivial solutions offer little engineering value. We therefore emphasize non-trivial modes throughout, as is both standard and necessary in applied contexts. In engineering contexts, these trivial modes are typically uninformative and carry no physical relevance. In all scenarios and including the removed Section 3.3 (as originally written), we focus exclusively on **non-trivial solutions**, which are more relevant in applied settings. The trival solution should be avoided in this context. For example, we observed that without additional constraints—such as the $A(\boldsymbol{x})$ introduced in Equation (23)—neural networks tend to converge to such trivial constant modes. To prevent this, we explicitly define and compute the minimal **non-trivial** eigenvalue and corresponding eigenfunction throughout our work, as also clarified in the revised statement in Section 2.1:

“Unless otherwise specified, the term "eigenvalue" and "eigenfunction" refers to the non-trival minimum modulus eigenvalue its corresponding eigenfunction of an operator.”

Q2.10 I struggle to see how 3.4 suggests that the DQRI is a good method- The solution obtained by spectral methods and that obtained by the DQRI are far from being in agreement, so at least one must be a poor approximation of the exact solution. If they have very similar estimations for the eigenvalue, then they are probably in fact both poor, as one would expect that being close to the minimal eigenvalue would impose some kind of stability, where one must be close to the corresponding eigenvector. Furthermore, I’m unclear as to why the reference value of 1.3957 is taken in this section, if their aim is to find the minimal eigenvalue.

R2.10

We sincerely appreciate the reviewer’s insightful question. This part of the original manuscript indeed intended to explore the potential applicability of the DRQI method beyond linear eigenvalue problems, particularly to nonlinear problems where spectral methods can still provide reference values.

However, after careful reconsideration, we acknowledge that this example (previously in Section 3.4) is not ideally aligned with the core focus of our work, which is on linear differential operators as introduced in Section 2.1. Furthermore, as the reviewer rightly points out, the comparison in that case was not conclusive due to the phase mismatch between the spectral and DRQI solutions, which dominated the discrepancy in the eigenfunction approximation. As such, we have decided to remove this section from the revised manuscript for clarity and focus.

Regarding the reference value of 1.3957, this was obtained using a spectral method implementation provided in our public GitHub repository, specifically under the folder: `To_Reviewer1 Q2.10`. When N is set to 32, this method yields the smallest modulus eigenvalue 1.3957 as the reference value.

We also included the MSR convergence curve for the DRQI solution of solving this problem. The MSR, as discussed in Section 3.1.2, serves as a robust indicator of the consistency between the approximate eigenvalue and the corresponding eigenfunction. Hence, its convergence provides evidence that DRQI is approximating both the eigenvalue and eigenfunction meaningfully, even in the absence of a known ground truth.

Additionally, we note that although both the loss function (LF) and the mean square residual (MSR) exhibit consistent convergence in this nonlinear eigenvalue problem, we did not observe the phenomenon—typical in linear problems—where the LF and MSR tend to coincide as training proceeds. This distinction can be seen in the convergence curves for various values of ω (e.g., $\omega = 0.1, 0.5, 0.9$, available in the supplementary folder, seed=30). The divergence suggests that the training remains in an early stage and has not fully entered the asymptotic regime. Therefore, we chose not to present these results in the main manuscript until the underlying mechanism is better understood.

Lastly, we would like to clarify that the original intention of including this example was exploratory in nature, with relevance to engineering applications where exact solutions are often unavailable, and the ability to achieve self-consistent approximations is of practical value.

We thank the reviewer again for prompting a clearer framing of the method’s scope.

Q2.11 The quality of the graphics is rather low, vector graphics should be used or high resolution rasterized graphics

R2.11:

Thank you for the valuable suggestion regarding the figure quality. We have updated our analysis tools in the Github repository to save images at 600 dpi resolution and imporved the quality of all figures. Additionally, you can manually adjust the resolution and format in the `save_figure(self)` function in `view.py` as needed to obtain higher-quality vector or raster graphics as you wish. Furthermore, as for the high resolution figure reported in the new manuscript, they can be accessed in the `high resolution figures` path in the repository.

Reviewer 2:

Q1: The manuscript compares the accuracy during the learning process with two other eigenvalue algorithms using neural networks. The Authors use the same network architecture and learning parameters and compare results obtained after a given number of learning epochs. Is the computational cost of a single step the same for all three methods? Would the results be different if the x-axis on the plots presented training time, instead of epoch number?

R1:

Thanks for the reviewer's insightful question.

1. Each of the three algorithms performs one forward propagation, several arithmetic operations, and one backward propagation per iteration step. Considering practical engineering scenarios where the neural network size is typically scaled up to improve fitting capacity, the dominant computational cost lies within the neural network operations themselves. The additional arithmetic computations outside the network are relatively negligible compared to the internal network computations. Therefore, the computational time per iteration step can be reasonably considered approximately equal across the three methods.

   We noticed that some of the runs recorded unusually high average training times per epoch (see in the QMC results). Upon investigation, this was due to the laptop being unintentionally operated under battery-saving mode without external power supply, which significantly throttled CPU/GPU performance.

2. We conducted additional experiments to evaluate whether using training time instead of epoch number as the x-axis would affect the comparative results. The outcomes showed minimal differences, as the per-step computational costs were very similar. Due to this negligible impact, these results were not included in the main manuscript for brevity but are available in detail in our GitHub repository for transparency and reproducibility. (For 1D Laplace Case, 0.007~0.009s/epoch in average, could be fluctutated accordinng to the status of the hardwares.)

Q2: For PDEs on a domain with more than 2 dimensions, the manuscript contains plots of what it describes as the "probability density of the guessed function output by the model". As far as I know, this is not a standard terminology, and thus it should be properly explained. This concept and similar plots are used in the cited Inverse Power Method Neural Network paper, and it is clearly defined there - "density of a function u is defined as the probability density function of u(X), where X is a uniformly distributed random variable on Omega". I suggest adding this definition or a reference to the IPMNN paper at the first usage.

R2:

We appreciate the reviewer's helpful comment regarding the terminology and definition of the "density" used in our plots. In response, we have revised the manuscript to include a precise definition of this concept in Section 3.1.1.1. Specifically, we added following contents to clearify this issue:

"To further investigate the divergence between eigenvalue and eigenfunction convergence, especially in Type-B behavior, the statistical distribution of the neural network output $u(\boldsymbol{x})$ over the computational domain Ω is analyzed. Specifically, following the definition adopted in [22], let

$\boldsymbol{X} \sim \mathcal{U}(\Omega)$ be a random variable uniformly distributed over the domain $\Omega \subset \mathbb{R}^d$. The density of a function $u: \Omega \to \mathbb{R}$ is defined as the probability density function (PDF) of the scalar random variable $u(\boldsymbol{X}) \in \mathbb{R}$. Then the density can be represented as:"

$$\rho_u(z) = \frac{d}{dz}\mathbb{P}(u(\boldsymbol{X}) \leq z), \mathrm{z} \in \mathbb{R} \tag{23}$$

"In practice, this density is estimate by sampling i.i.d. points $\{\boldsymbol{x}_i\}_{i=1}^N \sim \mathcal{U}(\Omega)$, computing $u(\boldsymbol{x}_i)$ and applying a histogram normalized to represent a probability density function:"

$$\hat{\rho}_u(z) = \frac{1}{N\Delta z}\sum_{i=1}^{N} I_{[z_k, z_k+\Delta z)}\left(u(\boldsymbol{x}_i)\right), \text{where } I_{[z_k, z_k+\Delta z)}(u(\boldsymbol{x}_i)) = \begin{cases} 1, u(\boldsymbol{x}_i) \in [z_k, z_k+\Delta z) \\ 0, \text{else} \end{cases} \tag{24}$$

"Here I denotes the indicator function."

The corresponding implementation can be found in our public code repository, specifically under the `views.py` file, in the `elif mode == "Density"` section.

In this modification, a citation to the IPMNN reference has been added at the first usage of this term to ensure clarity for the reader.

Q3: That being said, if the usage of the term "density function" is as above, the shape of "Theoretical" curves on these density plots raises some concerns. Let us focus on the first plots, for the Laplacian with homogeneous Dirichlet BC (Fig. 4). First, the integral of these densities is clearly not equal to 1. Secondly, the theoretical curve is unexpectedly rugged. It has multiple local extrema with sharp corners, which is rather unexpected given the very smooth and unimodal nature of the eigenvalues, and very much unlike the curves in the IPMNN paper for the same problem.

R3:

Upon review, we agree that the original theoretical density plots exhibited undesirable artifacts such as non-smoothness and local sharp extrema. This issue stems primarily from the way the density plots were initially constructed: we sampled points from the same LHS dataset used in the training phase, which, while convenient, lacks uniform coverage of the domain—especially in higher dimensions—and can lead to sparse and irregular sampling patterns. These irregularities can distort the resulting empirical density, particularly when visualized via line plots.

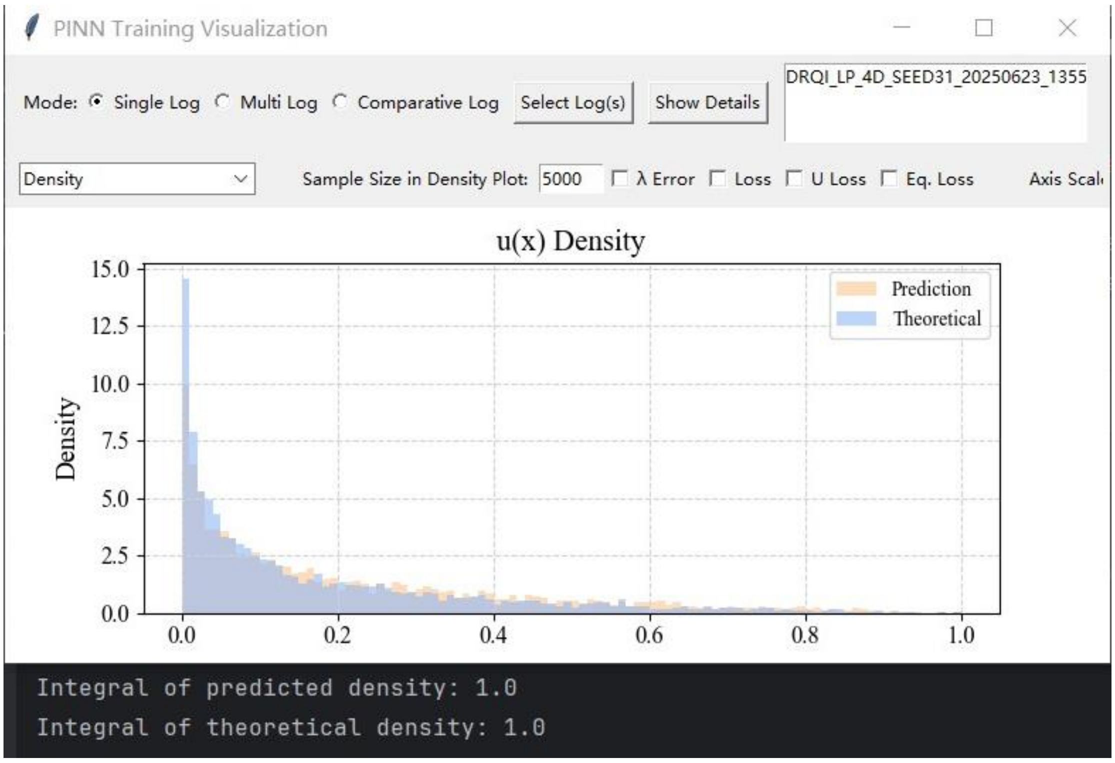
To address this, we have revised our density visualization strategy as follows:

1) Uniform Sampling with `torch.rand`: We now generate evaluation points using `torch.rand`, which draws i.i.d. samples from a uniform distribution over the domain. While `torch.rand` uses pseudo-random sampling and is not strictly stratified, it provides better uniformity in expectation compared to LHS for large sample sizes, and avoids the deterministic structure and clumping behavior that can affect low-dimensional LHS samples. This results in more reliable and unbiased empirical density estimates.

2) Histogram-based Visualization: We have also replaced the previous line-based plotting of density curves with histogram-based visualizations. Unlike line plots—which can suggest misleading continuity or artificial smoothness in sparsely sampled regions—histograms provide a more accurate depiction of the sampled distribution, especially in cases where the underlying probability density may exhibit compact support or multimodal structure. This makes them more appropriate for visualizing the empirical distribution of $u(\mathbf{X})$, particularly in high-dimensional settings.

3) For the concern of density normalization raised by the reviewer, we have solved this problem in our new manuscript and code. This bug was caused by the uncorrect usage of density function and unnormalized theoretical solution and predition. In the revised version, we switched to histograms computed from uniformly sampled `torch.rand` points, and ensuring accurate normalization. For

verification, we now print the numerical integral of each density plot in the visualization module, you can check the information in the console when you plot the densities.


PINN Training Visualization
Mode: Single Log Multi Log Comparative Log
Select Log(s)
Show Details
DRQI_LP_4D_SEED31_20250623_1355
Density
Sample Size in Density Plot: 5000
λ Error
Loss
U Loss
Eq. Loss
Axis Scal
u(x) Density
Density
Prediction
Theoretical

```
Integral of predicted density: 1.0
Integral of theoretical density: 1.0
```

The updated results are shown in Fig. 7, 8, where both predicted and theoretical densities are visualized using this revised approach.

Q4: Some other minor issues are listed as annotations in the attached copy of the manuscript.

We appreciate the reviewer's careful reading and the helpful annotations. The manuscript has been revised accordingly to address all listed concerns. In particular, modifications have been made to:

- the abstract,
- the problem background and network constraint formulation in Section 2.1,
- the formal construction of algorithms in Section 2.3, and

- the definition of "density" in Section 3.1.1.2.

We thank the reviewer again for these constructive suggestions, which helped improve the clarity and rigor of the manuscript.

Reviewer 3:

Q1: All the benchmark results exposed in the abstract "Particularly, the DRQI reduces errors by 373.54 times and 3.11 times compared to the DRM and IPMNN, respectively, in the Laplace operator problem..." are inadequate there.

R1:

Thank you for the valuable comment. The previously included specific numerical comparisons in the abstract, such as the error reduction factors, were found to be too detailed and not appropriate for the abstract's concise nature. We have accordingly revised and streamlined the abstract to focus on the key methodological contributions and main findings, removing excessive numerical detail while retaining clarity and impact. The revised results exposed in the abstract turned into:

"…The algorithm employs a semi-implicit iteration scheme to steer the early-stage convergence path and enhance optimization flexibility, and the mean squared residual (MSR) of the eigenvalue problem itself as the stop criterion. The DRQI is validated by the comparative studies with the deep Ritz method (DRM) and the inverse power method neural network (IPMNN). The effectiveness of MSR as both a theoretical and numerical criterion for monitoring convergence and determining stopping conditions is demonstrated. Furthermore, DRQI exhibits fast and stable convergence to larger eigenvalues in low-dimensional settings, specifically in 1~3D problems with Dirichlet or periodic boundary conditions, showcasing its practical advantages…"

Q2: The extension of Rayleigh Quotient Iteration from matrix to operator settings lacks rigorous justification. The equivalence between operator discretization via sampling and matrix approximation is assumed without analysis.

R2:

Thank you for this insightful comment regarding the extension of the Rayleigh Quotient Iteration (RQI) from matrix to operator settings. Our approach is based on an asymptotic approximation perspective, where the essential idea of sampling and discretization is to approximate the continuous operator by a large-scale matrix, thus enabling the classical Rayleigh quotient iteration concept to be approximately realized within the neural network function space.

In our revised manuscript, particularly in Section 2.3 ("The Deep Rayleigh Quotient Iteration"), we imporved the expression and offered a detailed discussion that rigorously justifies this extension from a discretization and approximation perspective. Specifically, we adopt LHS sampling methods (in the expreiments we alos used QMC) to discretize the continuous operator eigenvalue problem on the domain, thereby transforming it into a finite-dimensional matrix eigenvalue problem. This is expressed formally in Equation (11), where the discrete operator matrix $\boldsymbol{L}$ is constructed from pointwise evaluations of the continuous differential operator $\mathcal{L}$ on the sampled points. Therefore, we added these discription to explain the discretization in a more rigorous sense:

"In Equation (11), the matrix $\boldsymbol{L}$ is an $m \times m$ is a finite-dimensional approximation of the continuous linear differential operator $\mathcal{L}$, constructed via pointwise sampling over a finite set $\mathcal{X} \subset \Omega$. As the number of sampling points increases and their distribution becomes dense in Ω, the discrete operator $\boldsymbol{L}$ provides a progressively better approximation to $\mathcal{L}$ in the sense of convergence in an appropriate operator norm. Therefore, the eigenvalue problem of the continuous operator $\mathcal{L}u = \lambda u$ can be discretized into a finite-dimensional eigenvalue problem $\boldsymbol{L}\boldsymbol{u} = \lambda \boldsymbol{u}$, where $\boldsymbol{u} \in \mathbb{R}^m$ is the evaluation of $u(\boldsymbol{x})$ at the sample points. In the k-th iteration epoch, the eigenvector $\boldsymbol{u}$ is defined as:"

$$\boldsymbol{Lu} = \boldsymbol{L}[u(\boldsymbol{x}_1), \dots, u(\boldsymbol{x}_m)]^{\mathrm{T}} = [\mathcal{L}u(\boldsymbol{x}_1), \dots, \mathcal{L}u(\boldsymbol{x}_m)]^{\mathrm{T}} \tag{11}$$

"where $u_k(\boldsymbol{x})$ is the eigenfunction at this time. Given sufficient smoothness of the eigenfunctions and a suitably dense and representative sampling of the domain, the spectrum of the discrete operator $\boldsymbol{L}$ converges to that of the continuous operator $\mathcal{L}$. This justifies the use of RQI on the discretized system to approximate the spectral properties of $\mathcal{L}$. Thus, during the iteration, according to the Equation (9):"

Furthermore, the theoretical analysis on the orders of magnitude in Section 3.1.1.2 demonstrates that, as the training progresses and the solution stabilizes, the DRQI loss function asymptotically converges to the MSR $\mathcal{L}_{\mathrm{eq}}^2 = \mathbb{E}_{\boldsymbol{x}\in\mathcal{X}}[\mathcal{L}u_k(\boldsymbol{x}) - \lambda u_k(\boldsymbol{x})]^{\mathbf{2}}$ up to a vanishing term of order $\mathcal{O}(1/m)$, where m is the number of sampling points. These contests has been added into the paper accordingly:

"To understand why the DRQI loss tends to converge toward the mean squared residual (MSR) in practice, consider the form of the DRQI loss function at iteration step k, as shown in Equation (18). As training progresses and the eigenfunction approximation stabilizes, it is expected that $u_k \approx u_{k-1}$, especially under a well-chosen relaxation factor ω. In the limit of convergence, the Equation (18) turns into"

$$l(\boldsymbol{\theta}) \approx \mathbb{E}_{\boldsymbol{x}\in\mathcal{X}}[\mathcal{L}u_k(\boldsymbol{x}) - (\lambda+1)u_k(\boldsymbol{x})]^2 = \mathbb{E}_{\boldsymbol{x}\in\mathcal{X}}\left[\left(\mathcal{L}u_k(\boldsymbol{x}) - \lambda u_k(\boldsymbol{x})\right) - u_k(\boldsymbol{x})\right]^2 \tag{26}$$

"This expression can be expanded as:"

$$l(\boldsymbol{\theta}) \approx \mathbb{E}_{\boldsymbol{x}\in\mathcal{X}}\left[\left(\mathcal{L}u_k(\boldsymbol{x}) - \lambda u_k(\boldsymbol{x})\right)^2 + u_k^2(\boldsymbol{x}) - 2u_k\left(\mathcal{L}u_k(\boldsymbol{x}) - \lambda u_k(\boldsymbol{x})\right)\right] \tag{27}$$

"Recalling the definition of the Rayleigh quotient at iteration k:"

$$\lambda = \frac{\langle \mathcal{L}u_k, u_k \rangle}{\langle u_k, u_k \rangle} \Rightarrow \mathcal{L}u_k \approx \lambda u_k \tag{28}$$

“Therefore, when $k \to \infty$, $\lambda \to \lambda_{\text{th}}$ and $u \to u_{\text{th}}$, substituting into (27), noticing that $\mathbb{E}_{\boldsymbol{x}\in\mathcal{X}}[\mathcal{L}u_k(\boldsymbol{x}) - \lambda u_k(\boldsymbol{x})]^2 = \mathcal{L}_{\text{eq}}^2$, yields:”

$$l(\boldsymbol{\theta}) \approx \mathcal{L}_{\text{eq}}^2 + \mathbb{E}_{\boldsymbol{x}\in\mathcal{X}}[u_k^2(\boldsymbol{x})] \approx \mathcal{L}_{\text{eq}}^2 + \frac{1}{m}\|u_k\|^2, m \to \infty \tag{29}$$

“Since the u_k is L2 normalized at each iteration step, such that $\|u_k\|^2 \approx 1$, then:”

$$l(\boldsymbol{\theta}) - \mathcal{L}_{\text{eq}}^2 \approx \frac{1}{m} = o(1), m \to \infty \tag{30}$$

“This proves that the DRQI loss is asymptotically equivalent to the MSR up to a vanishing term of order $O(1/m)$. In practice, since the number of collocation points m is typically large (more than 1000 in this work), the difference is negligible and decays as a first-order infinitesimal. While the theoretical DRQI loss differs from the MSR, in practice it is observed that this difference is often much smaller than $1/m$. This is due to a combination of overfitting to collocation points, smoothness of the learned eigenfunction, and insufficient sampling of high-residual regions, all of which reduce the variance of residuals in the sample mean.”

Q3: The semi-implicit loss (Eq. 18) is introduced without theoretical support or ablation studies. The role and tuning of the relaxation factor ω remain unclear and ad hoc.

R3:

In response, a discussion has been added at the end of Section 3.2 to clarify the empirical and practical role of the relaxation factor. This addition analyzes the convergence behavior observed under different values of ωω and highlights its practical significance. In particular, the experimental results in Figure 17 show that while the convergence order may vary depending on the initialization (e.g., the random seed), the choice ω=0.8 consistently leads to more pronounced oscillations during early training. This suggests that ω serves primarily as a mechanism for early-stage guidance rather than a direct accelerator of convergence. These are added discussion helps clarify the rationale behind Eq. (18) and the tuning of ω in practice:

"To better understand the role of the relaxation factor during training, a comprehensive analysis of the structure of the loss function (18) should be conducted. In particular, the relaxation factor ω introduces an additional degree of freedom in the loss landscape. When $\omega = 1$, the DRQI loss reduces to a standard residual loss similar to DRM. However, for $\omega < 1$, the loss target includes a static component u_{k-1}, which acts as a regularization term that stabilizes the update by anchoring the output to the previous iteration."

"In practice, this blending has non-trivial interactions with adaptive optimizers such as Adam used in this section, which updates parameters according to:"

$$\theta_{k+1} = \theta_k - \eta \cdot \frac{\widehat{m}_k}{\sqrt{\widehat{v}_k} + \epsilon} \tag{34}$$

"where θ_k a component of the network parameters $\boldsymbol{\theta}$ at the k-th epoch, $\widehat{m}_k$ and $\widehat{v}_k$ are exponentially averaged estimates of the first and second moments of the gradient. When ω is large (e.g., $\omega = 0.8$), the learning target becomes more sensitive to the current model output ukuk, leading to higher variance in the residual term of the loss function and consequently in the gradient. This can inflate the second moment estimate $\widehat{v}_k$, resulting in inconsistent update magnitudes and visible oscillations in the loss curves, as observed in Fig 17. On the other hand, smaller ω values suppress this volatility, yielding smoother convergence curves."

"This explains why $\omega = 0.8$ consistently exhibits stronger oscillations. The behavior reflects early-stage instability rather than long-term convergence inefficiency."

Q4: The DRM implementation is modified (e.g., boundary conditions, $\gamma = 0$), potentially undercutting its accuracy. This makes the performance comparisons with DRQI questionable.

R4:

First, we would like to clarify that in our implementation, the Dirichlet boundary conditions are enforced via a hard constraint using the multiplier $A(\boldsymbol{x})$ (See in Equation (23) in our work), ensuring that the boundary conditions are strictly satisfied during training. This differs from the original DRM approach [15], which controlled the Dirichlet boundary condition softly by adding a regularization term on the boundary values

$$\int_{\partial\Omega} u^2(\boldsymbol{x}) d\boldsymbol{x}$$

to the loss function, for the eigenvalue problem of the Laplace operator with Dirichlet boundary conditions. While both methods seek to enforce Dirichlet boundary conditions, the use of the hard constraint via multiplier results in stricter boundary condition satisfaction.

Furthermore, the original DRM method, as proposed, does not address eigenvalue problems with periodic boundary conditions, which are also considered in our work to conduct the comparative study.

In fact, these modifications have improved DRM's performance compared to the original results reported in the literature. In the 1D Laplace operator eigenvalue problem with Dirichlet boundary conditions, the original DRM paper reported the error on the order of 10^{-2}, while our implementation achieves 10^{-6} level, demonstrating that the modified DRM implementation is more accurate. More reproducible test results and detailed logs supporting this implementation and performance improvement can be found in the logs folder of the attached GitHub repository.

To clarify the selection and rationale of these parameters and boundary condition treatments, we have added detailed explanations in Section 3.1.1 of the revised manuscript:

"For DRM, following the recommendations provided by the original authors [15], $\gamma = 100$ is used only in the 4D case, while $\gamma = 0$ is applied in all other settings."

Q5: Important experimental details—e.g., random seeds, optimizer settings, network initialization, training variance—are not reported. The stochastic nature of neural methods requires statistical validation, not just single-run results.

R5:

We have included a more detailed description of key experimental settings in the revised manuscript (see in Table 1, 2, 3, 4). Specifically, we now report the optimizer configurations (e.g., Adam with specified initial learning rates), sampling methods (e.g., LHS or QMC), and series of fixed random seeds for the results.

We acknowledge the inherent stochasticity of neural network training and agree that single-run results may not fully reflect robustness. Accordingly, we have added statistical validation for selected representative experiments by conducting several repeated runs under the same configuration, reporting the 3 metrics (see revised Fig. 5 and other point cloud charts) at the end of training. While we focused on a small number of runs for tractability, the observed variance was moderate and did not affect the overall conclusions of the paper.

We hope these additions improve the transparency and credibility of the experimental results.

Q6: The manuscript has some grammatical errors and verbose phrasing that obscure key ideas. A thorough edit is needed for clarity.

R6:

Thank you for your constructive feedback regarding the language and phrasing. We have made extensive efforts to improve the clarity and readability of the manuscript by revising the grammar and simplifying verbose expressions throughout the text. We hope these changes enhance the presentation of our key ideas. If the reviewers have any specific suggestions or examples, we would be grateful to receive them to further improve the manuscript.

Many THANKs to Editors and Reviewers, and hope that the revised manuscript will meet with your approval.

Best regards,

Liangxing Li