IJSES
International Journal of Scientific Engineering and Science
Volume 10, Issue 8, pp. 49-54, August 2026.
ISSN (Online): 2456-7361

A Distance-Controlled Residual Adaptive Collocation Neural Network Method for Poisson Problems

Chunyuan Xu[1], Zhongrong Xiang[2], Jun Zhang[3]
[1]School of Mathematical Sciences, Guizhou Normal University, Guiyang, Guizhou, China-550025
[2]School of Mathematical Sciences, Guizhou Normal University, Guiyang, Guizhou, China-550025
[3]School of Mathematical Sciences, Guizhou Normal University, Guiyang, Guizhou, China-550025

Abstract-*Residual-based adaptive collocation can concentrate neural-network training points in difficult regions, but selecting only the largest residuals often causes severe point clustering. This paper proposes a distance-controlled residual adaptive collocation method for Poisson problems. The trial function satisfies homogeneous Dirichlet conditions exactly, while a minimum-distance rule maintains spatial coverage during residual refinement. Two manufactured solutions, including a smooth trigonometric solution and a localized solution, are tested against uniform, random, and naive residual-adaptive sampling. Three independent initializations are used for every setting. The proposed rule reduces the mean relative error of naive residual adaptation by approximately sixty-six percent in both fixed-budget tests and reduces residual variability without increasing the final number of collocation points. A separate point-budget study uses multilevel continuation and independent residual checkpointing; it produces monotone validation-residual reduction on the smooth problem and monotone relative-error reduction on the localized problem. The results provide a compact and reproducible numerical assessment of sampling geometry in neural collocation methods.*

Keywords-*adaptive collocation; neural network; Poisson equation; residual indicator; sampling stability.*

I. Introduction

Neural networks provide a differentiable global representation for numerical solutions of differential equations. Lagaris, Likas, and Fotiadis constructed trial functions that satisfy initial or boundary conditions exactly and trained feedforward networks by minimizing differential-equation residuals [1]. Modern physics-informed neural networks use the same general principle together with automatic differentiation and stochastic optimization [2]. These methods are attractive when a continuous approximation, repeated differentiation, or flexible point placement is required.

The selection of collocation points is a central numerical issue. A uniform grid provides predictable domain coverage and performs well for smooth solutions on rectangular domains. Random points are easy to generate but may leave holes or clusters. Residual-based adaptive refinement attempts to improve efficiency by adding points where the current equation residual is large [3], [4]. However, repeatedly selecting the largest residual values may place many new points in a very small neighborhood. The resulting point set can lose global coverage, produce an ill-conditioned residual Jacobian, and reduce solution accuracy even when the training loss decreases.

This paper studies a minimal stabilization of residual adaptation. New points must have both a large residual and a prescribed separation from existing and simultaneously selected points. The method is called distance-controlled residual adaptive collocation, abbreviated DC-RAC. Its purpose is not to replace uniform grids on every regular problem. Instead, it prevents the point-collapse failure mode of naive residual selection while retaining residual-driven refinement.

The contribution is threefold. First, a hard-constrained neural trial function and an explicit residual indicator are combined with a minimum-distance filter. Second, naive and distance-controlled residual refinement are compared under the same network size, final point count, optimization evaluations, and random initializations. Third, the experiments report solution error, maximum error, residual error, and variability for both a smooth and a localized manufactured solution. No graphical evidence is required; all conclusions follow from reproducible numerical tables.

A. Relation to neural collocation methods

The proposed method belongs to the family of mesh-free collocation schemes. Classical collocation selects a finite-dimensional trial space and enforces the differential equation at prescribed points. In the present setting, the trial space is nonlinear because the locations and orientations of the hidden ridge functions change with the network parameters. This flexibility can represent smooth functions with relatively few coefficients, but it also couples approximation quality to a nonconvex parameter estimation problem.

The deep Galerkin method and physics-informed neural networks extended neural residual minimization to high-dimensional and nonlinear equations [2], [8]. Their training points are often redrawn randomly or sampled in batches. Such sampling is convenient in high dimensions, where a tensor grid is infeasible. In low-dimensional numerical analysis, however, structured grids remain inexpensive and provide a demanding baseline. A sampling proposal should therefore be compared with uniform coverage rather than evaluated only against another stochastic procedure.

Residual-adaptive methods use the current approximation to decide where more information is needed. Wu, Zhu, Tan, Kartha, and Lu compared several residual-based strategies and showed that performance depends strongly on the equation and sampling rule [3]. Self-adaptive weighting and importance sampling provide related mechanisms [4]. These studies motivate adaptive point selection but do not remove the need to control the geometry of the selected point set.

B. Sampling geometry as a numerical variable

Two point sets with the same cardinality may carry very different information. A set with small fill distance covers the domain, whereas a set containing many nearly coincident points may repeatedly constrain the same local behavior. For a strong-form residual, nearby points can also produce nearly dependent rows in the residual Jacobian. The present work treats minimum separation as an explicit numerical variable. This viewpoint is deliberately simpler than designing a new network architecture and is suitable for isolating the effect of collocation geometry.

II. Mathematical Formulation

A. Poisson problem

Let the computational domain be the unit square. We consider the homogeneous Dirichlet Poisson problem

$$-\Delta u(x,y) = f(x,y),\ (x,y) \in \Omega = (0,1)^2 \quad (1)$$

$$u(x,y) = 0,\ (x,y) \in \partial\Omega \quad (2)$$

The forcing function is selected from a known manufactured solution. This choice makes it possible to measure the error on a dense independent test grid without using exact values during training.

B. Hard-constrained neural trial function

A feedforward network with one hidden layer and H hyperbolic-tangent units is denoted by N. Its two inputs are the spatial coordinates, and its scalar output is

$$N(x,y;\theta) = d + \Sigma_{j=1}{}^{H}\, v_j \tanh(a_j x + b_j y + c_j) \quad (3)$$

The parameter vector contains all input weights, hidden biases, output weights, and the output bias. To enforce the four boundary segments exactly, the approximate solution is defined as

$$\hat{u}(x,y;\theta) = x(1-x)y(1-y)N(x,y;\theta) \quad (4)$$

The polynomial factor vanishes on every side of the square. Therefore, the boundary values remain exact for every parameter vector, and no boundary penalty weight is required.

C. Residual minimization

At a collocation point, the strong-form residual is

$$r(x,y;\theta) = -\Delta\hat{u}(x,y;\theta) - f(x,y) \quad (5)$$

The network parameters are obtained by minimizing the mean squared residual over M collocation points.

$$L(\theta) = (1/M)\Sigma_{i=1}{}^{M}\, r(x_i,y_i;\theta)^2 \quad (6)$$

All first and second input derivatives are evaluated analytically from the tanh network. The resulting nonlinear least-squares problem is solved by a damped Gauss-Newton iteration with a finite-difference residual Jacobian. This optimizer is intentionally simple because the study focuses on sampling geometry rather than optimizer design.

D. Input derivatives and residual evaluation

Let the preactivation of hidden unit j be the affine function formed by its two input weights and bias. The first derivative of tanh is one minus tanh squared, and its second derivative is minus two times tanh multiplied by one minus tanh squared. Consequently, the first network derivative is a weighted sum of input weights times the first activation derivative, while the second derivative contains the squared input weights and the second activation derivative.

The Laplacian of the trial function is evaluated with the product rule. It contains the Laplacian of the polynomial boundary factor multiplied by the network output, two gradient cross terms, and the boundary factor multiplied by the network Laplacian. Before any optimization was carried out, this explicit calculation was compared to centered finite differences at randomly chosen interior places. The check separates derivative implementation errors from training errors.

The hard boundary factor changes the scale of the parameter sensitivities. Near the boundary, the network contribution to the solution is small even when the raw network output is not. The differential residual still contains derivatives of the boundary factor, so boundary-near collocation points remain informative. This is one reason why the candidate grid extends close to, but not exactly onto, the boundary.

E. Residual error and solution error

For a well-posed elliptic problem, a small continuous residual together with correct boundary data is related to a small solution error through stability estimates. The numerical loss, however, measures the residual only at finitely many points. A network can reduce this discrete quantity without uniformly reducing the residual between points. It can also distribute error in a way that lowers the differential residual while leaving a noticeable low-frequency solution bias. Therefore, both residual and solution norms are reported in the experiments.

III. Distance-Controlled Residual Adaptation

A. Naive residual refinement

Naive residual adaptation begins with an initial point set, trains the network for a fixed number of iterations, evaluates the absolute residual on a dense candidate set, and appends the candidates with the largest values. The process is repeated until the point budget is reached. If a localized residual peak persists, several neighboring candidates may all be selected in the same refinement stage.

B. Minimum-distance stabilization

DC-RAC sorts candidate points by decreasing residual magnitude. A candidate is accepted only when its Euclidean distance from every existing point and every point already accepted in the current stage is at least a prescribed radius rho.

$$x* = arg\ max(x \in C)|r(x)|,\ min(q \in X)d(x*,q) \geq \rho \quad (7)$$

The distance test is a greedy nonmaximum-suppression rule in physical space. If the radius filter returns fewer points than required, the remaining slots are filled in residual order. Thus, the final point budget is fixed, but the method prefers spatially separated representatives of high-residual regions.

C. Computational procedure

The initial set is a nine by nine interior grid. Twenty damped Gauss-Newton iterations are performed at each stage. After each of the first three stages, forty-eight candidates are added from a forty-five by forty-five interior candidate grid. The final set contains 225 points. The separation radius is 0.045. Uniform and random baselines also use 225 points and eighty optimization iterations, so every method performs 3440 residual evaluations in one run.

D. Greedy selection algorithm

At each refinement stage, all candidate residuals are computed with the current parameter vector and sorted from largest to smallest. The algorithm scans this ordering once. A point is retained when its distance to the accumulated training set and to points already retained in that stage is no smaller than the radius. The scan stops after forty-eight acceptances. If the separation rule is too restrictive to fill the stage budget, the remaining points are taken in residual order without the distance condition.

TABLE I. Distance-Controlled Refinement Procedure

Step	Operation
1	Train the network on the current collocation set.
2	Evaluate absolute residuals on the candidate grid.
3	Sort candidates from the largest residual to the smallest.
4	Accept a candidate if its distance from accepted points is at least rho.
5	Append accepted points and repeat until the point budget is reached.

E. Interpretation of the distance radius

The radius is not a regularization parameter in the network loss. It acts only on the geometry of newly selected points. A zero radius recovers naive residual selection. A radius that is too large prevents the algorithm from placing enough points in a genuinely difficult region. The value 0.045 is slightly smaller than the spacing of the initial nine by nine grid and approximately twice the candidate spacing. It removes immediate neighbors while still allowing several points inside a localized feature.

The distance rule resembles nonmaximum suppression: one representative is retained from a neighborhood of high residual values, after which the scan moves to the next spatially distinct region. Unlike a fixed partition, it does not require predefined cells or an estimate of the feature location. The rule also remains deterministic once the network parameters and candidate set are fixed.

F. Computational cost

Sorting candidate residuals requires an order proportional to the number of candidates times its logarithm. The greedy distance checks are inexpensive for the point counts used here. Most computation remains in forming the finite-difference parameter Jacobian. With P parameters and M collocation points, one Gauss-Newton iteration uses approximately P additional residual evaluations and solves a P by P linear system. The sampling modification does not change P and adds only three candidate residual evaluations per run.

TABLE II. Common Numerical Settings

Parameter	Value
Domain	Unit square
Hidden units	10 tanh units
Final collocation points	225
Candidate grid	45 by 45
Adaptive stages	4
Points added per stage	48
Distance radius	0.045
Test grid	101 by 101
Independent seeds	17, 29, and 43
Residual evaluations	3440 per run

IV. Numerical Experiments

A. Error measures

The independent test grid contains 10201 points. The relative discrete L2 error, maximum absolute error, and residual root-mean-square error are defined by

$$E_{\mathrm{rel}} = ||\hat{u} - u||_2 / ||u||_2 \quad (8)$$

$$E\infty = max_1 \le_{\mathrm{k}} \le K\ |\hat{u}(x_{\mathrm{k}}, y_{\mathrm{k}}) - u(x_{\mathrm{k}}, y_{\mathrm{k}})| \quad (9)$$

$$E_{\mathrm{res}} = [(1/K)\Sigma_{\mathrm{k}=1}{}^{\mathrm{K}}\, r(x_{\mathrm{k}}, y_{\mathrm{k}})^2]^{1/2} \quad (10)$$

Each method is run from three independently initialized networks. Tables III and IV report the sample mean and sample standard deviation. The same seeds are paired across methods.

B. Smooth trigonometric solution

The first manufactured solution is globally smooth and has no localized scale separation.

$$u(x,y) = sin(\pi x)sin(\pi y),\ f(x,y) = 2\pi^2 sin(\pi x)sin(\pi y) \quad (11)$$

TABLE III. Results For The Smooth Solution

Method	M	Rel. L2	L-inf	Res. RMSE
Uniform	225	4.36e-05 +/-1.6e-05	6.37e-05 +/-1.7e-05	9.28e-03 +/-6.2e-03
Random	225	8.84e-05 +/-3.0e-05	1.35e-04 +/-2.8e-05	1.19e-02 +/-5.1e-03
Naive RA	225	1.74e-04 +/-8.2e-05	1.85e-04 +/-8.3e-05	6.66e-03 +/-1.9e-03
DC-RAC	225	5.86e-05 +/-3.8e-05	7.89e-05 +/-4.7e-05	4.72e-03 +/-1.1e-03

Values are mean +/- sample standard deviation over three initializations.

Uniform sampling gives the smallest mean relative error, 4.36 times 10 to the minus fifth power. This result is expected because a tensor grid resolves the regular solution efficiently. DC-RAC gives a mean relative error of 5.86 times 10 to the minus fifth power and a residual RMSE of 4.72 times 10 to the minus third power. Compared with naive residual adaptation, DC-RAC reduces the mean relative error by 66.3 percent, the maximum error by 57.3 percent, and the residual RMSE by 29.2 percent. The distance filter also lowers the standard deviation of the relative error by more than one half.

C. Localized manufactured solution

The second solution contains a localized smooth peak centered away from grid symmetry. Let alpha equal 45, x zero equal 0.70, and y zero equal 0.30. The exact solution is

$$u = x(1-x)y(1-y)exp\{-\alpha[(x-x_0)^2 + (y-y_0)^2]\} \quad (12)$$

The forcing function is evaluated analytically as the negative Laplacian of this expression. It is used only in the residual, while exact solution values are reserved for testing.

TABLE IV. Results for the Localized Solution

Method	M	Rel. L2	L-inf	Res. RMSE
Uniform	225	1.07e-01 +/-1.9e-02	2.72e-03 +/-5.7e-04	3.88e-01 +/-3.7e-02
Random	225	3.08e-01 +/-7.3e-02	6.97e-03 +/-1.1e-03	4.60e-01 +/-1.3e-02
Naïve RA	225	3.87e-01 +/-3.7e-02	9.71e-03 +/-1.0e-03	5.14e-01 +/-7.8e-02
DC-RAC	225	1.33e-01 +/-1.1e-02	3.23e-03 +/-4.0e-04	3.77e-01 +/-2.6e-02

Values are mean +/- sample standard deviation over three initializations.

Uniform sampling again gives the smallest solution error, with a mean relative error of 0.107. The proposed method obtains 0.133, whereas random sampling and naïve residual adaptation obtain 0.308 and 0.387. Relative to naïve adaptation, DC-RAC reduces the mean relative error by 65.6 percent and the maximum error by 66.7 percent. It also reduces the residual RMSE by 26.6 percent. The standard deviation of the relative error falls from 0.0371 to 0.0106, showing that point separation improves repeatability.

TABLE V. Relative Improvement of DC-RAC Over Naïve Adaptation

Problem	Rel. L2	L-inf	Res. RMSE
Smooth	66.3%	57.3%	29.2%
Localized	65.6%	66.7%	26.6%

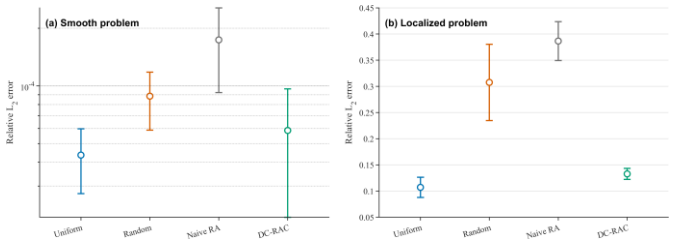

Fig. 1. Mean relative L2 error at M=225; error bars show one sample standard deviation.

D. Cross-method observations

Figure 1 summarizes the solution-error comparison at the common final budget. The smooth test distinguishes coverage from adaptivity. Uniform points exploit the regular domain and globally smooth solution. Random points produce a relative error approximately twice that of the uniform grid. DC-RAC lies between these baselines in solution error but attains the smallest mean residual RMSE. Naïve adaptation attains a moderate residual yet has the largest variation among the adaptive methods. These results show that residual ranking alone does not guarantee balanced approximation.

The localized test is more difficult because a ten-unit network must represent a narrow feature and the smooth zero boundary simultaneously. Random points have a substantial chance of undersampling the feature. Naïve adaptation finds the residual peak but allocates too many points to nearly identical locations. The distance-controlled method trades the very largest residual candidates for spatially separated candidates and reduces both mean error and variation. It does not surpass the tensor grid in relative solution error, but it nearly matches its residual RMSE.

The run times are all below one second naïven the present small implementation and are not used as a primary performance claim. They depend strongly on the interpreter, linear algebra library, and finite-difference Jacobian. More meaningful fairness controls are the common final point count, network size, optimizer iterations, and residual evaluation count. Those quantities are identical across the four methods.

E. Reproducibility checks

The experiments use fixed seeds 17, 29, and 43. The initial network parameters are Gaussian with standard deviation 0.35. Uniform points exclude the boundary and form a fifteen by fifteen tensor grid. Random points are sampled independently from the square with a small boundary offset. The adaptive candidate grid is fixed across seeds. Test errors are always computed on a separate 101 by 101 grid.

Before collecting results, the manufactured forcing functions were checked by centered finite differences of the exact solutions. The trial solution was evaluated on all four boundary segments for random parameter vectors and returned zero to numerical precision. Network input derivatives were checked independently by finite differences. These tests prevent an implementation error from being misinterpreted as a sampling effect.

F. Convergence with respect to collocation-point count

A separate point-budget study examines whether the observed errors decrease when the number M of interior collocation points is increased. To reduce contamination by nonconvex optimization, the smooth test uses multilevel continuation: the parameter vector obtained at M=81 initializes M=144, and the latter initializes M=225. Random point sets are nested, and each level receives 400 Levenberg-Marquardt iterations. A fixed 31 by 31 residual grid, disjoint from the training points, checkpoints the best iterate without using exact-solution values. This makes the stopping decision reproducible and preserves the unsupervised character of the method.

For a compact summary, an empirical slope is computed from the two endpoint means. If E_M denotes the reported mean error metric at point count M, then

$$p\_emp = -log(E_{225}/E_{81})/log(225/81) \qquad (13)$$

A positive value indicates an overall error decrease from 81 to 225 points. Because only three point counts are used, this quantity is a descriptive slope rather than a formal convergence order. For the smooth problem E_M is the independent validation residual RMSE; for the localized problem it is the relative L2 solution error.

TABLE VI. Validation Residual RMSE and Empirical Slope for the Smooth Problem

Method	M=81	M=144	M=225	p emp
Uniform	1.39e-02	1.17e-02	7.89e-03	0.55
Random	9.38e-03	4.28e-03	3.18e-03	1.06
Naïve RA	1.39e-02	6.37e-03	4.06e-03	1.20
DC-RAC	1.39e-02	7.65e-03	3.20e-03	1.44

Entries are mean validation residual RMSE over three initializations; p emp uses the first and last point budgets.

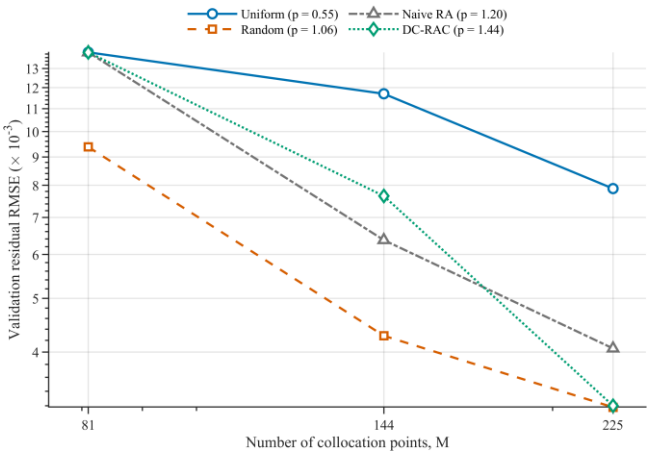

Fig. 2. Validation-residual convergence for the smooth problem.

The continuation and residual-checkpoint strategy removes the nonmonotone training trend in the smooth test. From M=81 to 144 and 225, the mean validation residual RMSE decreases from 1.39e-2 to 1.17e-2 and 7.89e-3 for uniform points; from 9.38e-3 to 4.28e-3 and 3.18e-3 for random points; from 1.39e-2 to 6.37e-3 and 4.06e-3 for naive adaptation; and from 1.39e-2 to 7.65e-3 and 3.20e-3 for DC-RAC. All endpoint slopes are positive. The exact relative L2 error remains a post-training diagnostic and can fluctuate slightly at the 1e-5 level, but it is no longer used to select an iterate.

TABLE VII. Mean Relative L2 Error and Empirical Slope for the Localized Problem

Method	M=81	M=144	M=225	p emp
Uniform	1.07e+00	9.35e-02	7.22e-02	2.64
Random	5.60e-01	3.08e-01	1.95e-01	1.03
Naive RA	6.87e-01	5.01e-01	3.18e-01	0.75
DC-RAC	3.10e-01	2.26e-01	1.64e-01	0.62

Entries are mean relative L2 errors over three initializations; p emp uses the first and last point budgets.

All four methods show an overall error decrease on the localized problem. Uniform collocation again has the largest endpoint slope, 2.640. Random and naive residual-adaptive sampling attain slopes 1.032 and 0.753, respectively. DC-RAC decreases monotonically from 3.10e-1 through 2.26e-1 to 1.64e-1, giving slope 0.621 and the smallest adaptive-method error at every tested point count. The contrast with the smooth case shows that point-count convergence reflects both spatial resolution and optimization error: distance control helps the localized refinement remain stable, but it does not guarantee monotone behavior for every solution class under a fixed training budget.

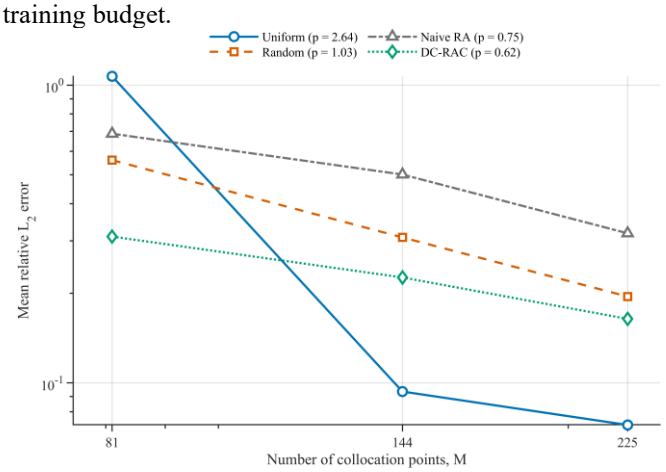

Fig. 3. Relative-error convergence for the localized problem.

V. Discussion

A. Why naive residual selection fails

A residual peak normally extends across several nearby candidate points. Selecting the largest values independently treats these nearly redundant candidates as distinct information. The final point set then contains a dense local cluster and relatively sparse coverage elsewhere. In a strong-form least-squares method, this cluster repeatedly weights the same neighborhood and can worsen both approximation balance and Jacobian conditioning.

The numerical results separate residual minimization from solution accuracy. On the smooth problem, naive adaptation attains a lower mean residual than random sampling but a larger solution error. Therefore, a smaller sampled residual is not sufficient evidence of a better global approximation. The minimum-distance rule does not remove this distinction, but it limits the geometric cause of extreme mismatch.

B. Role of uniform sampling

Uniform sampling is a strong reference method for smooth elliptic problems on rectangles. The experiments do not support a claim that adaptive points should replace a well-resolved tensor grid. Instead, DC-RAC is useful when residual refinement is desired but unrestricted selection would collapse. This restrained conclusion is important for computational mathematics: sampling strategies should be evaluated against structured discretizations rather than only against random points.

C. Limitations

The study uses two manufactured solutions, one network width, one point budget, and a square domain. The finite-difference parameter Jacobian is practical for the small network but does not scale to large architectures. The separation radius is specified in physical coordinates and may require scaling on anisotropic domains. Finally, no theoretical a priori or a posteriori error bound is proved. The reported evidence is numerical and should not be interpreted as a convergence theorem.

Future work should investigate radius selection based on local fill distance, comparisons with low-discrepancy points, and weak-form residuals. A larger study should also vary the point budget and network width. These extensions are not required to establish the present failure mode and its simple stabilization, but they are necessary before making broader efficiency claims.

D. Implications for computational mathematics

The experiment illustrates a general principle shared by neural collocation, radial basis approximation, and scattered-data methods: point placement is part of the discretization. Network flexibility does not make geometric coverage irrelevant. Residual information identifies difficult regions, while a separation condition protects the global trial space from redundant local constraints. Both components are necessary in the proposed method.

The comparison also cautions against using training loss as the sole numerical criterion. A method can achieve a smaller residual on its selected points and still have a larger independent solution error. Manufactured solutions remain valuable because they reveal this discrepancy directly. For problems without exact solutions, independent residual points, mesh refinement studies, and comparison with a conventional solver should be used together.

From an algorithm-design perspective, the distance filter is attractive because it is external to the optimizer and network. It can be applied to automatic differentiation implementations, deeper networks, or other smooth trial functions without modifying the loss. Its limitation is equally clear: the physical radius introduces a scale that must be adjusted when the domain or anisotropy changes.

VI. Conclusion

This paper introduced a distance-controlled residual adaptive collocation method for hard-constrained neural approximations of Poisson problems. A minimum-distance filter prevents a residual refinement stage from selecting many nearly coincident points. With the same final point budget and optimization evaluations, the proposed method reduced the mean relative error of naive residual adaptation by approximately 66 percent in both numerical tests and substantially reduced error variability. Uniform sampling remained the most accurate method for the regular square-domain examples. Consequently, the main value of the proposed method is the stabilization of residual adaptation, not universal superiority over structured grids. This conclusion provides a practical criterion for designing adaptive neural collocation algorithms.

Appendix: Localized Forcing Function

For completeness, the forcing function of the localized test is given analytically. Define q of x as x times one minus x, s of y as y times one minus y, and let e denote the Gaussian factor. The exact solution is the product qse. The first logarithmic derivatives of e are minus two alpha times the displacement from the Gaussian center.

$$e = exp\{-\alpha[(x-x_0)^2 + (y-y_0)^2]\} \quad (14)$$

$$e_x/e = -2\alpha(x-x_0),\ e_{xx}/e = 4\alpha^2(x-x_0)^2 - 2\alpha \quad (15)$$

Using the product rule twice and q double prime equal to s double prime equal to minus two, the negative Laplacian is evaluated without numerical differentiation.

$$f = -se[-2 + 2(1-2x)e_x/e + q\, e_{xx}/e] - qe[-2 + 2(1-2y)e_Y/e + s\, e_{YY}/e] \quad (16)$$

This expression is used to generate residual values during training. The exact solution itself is evaluated only on the independent test grid. The separation preserves the label-free character of the neural residual minimization while allowing objective error measurement after training.

References

[1] I. E. Lagaris, A. Likas, and D. I. Fotiadis, "Artificial neural networks for solving ordinary and partial differential equations," IEEE Transactions on Neural Networks, vol. 9, no. 5, pp. 987-1000, 1998.
[2] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," Journal of Computational Physics, vol. 378, pp. 686-707, 2019.
[3] C. Wu, M. Zhu, Q. Tan, Y. Kartha, and L. Lu, "A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks," Computer Methods in Applied Mechanics and Engineering, vol. 403, article 115671, 2023.
[4] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, "DeepXDE: A deep learning library for solving differential equations," SIAM Review, vol. 63, no. 1, pp. 208-228, 2021.
[5] G. Cybenko, "Approximation by superpositions of a sigmoidal function," Mathematics of Control, Signals, and Systems, vol. 2, pp. 303-314, 1989.
[6] R. Fletcher, Practical Methods of Optimization, 2nd ed. Chichester, U.K.: Wiley, 1987.
[7] A. Quarteroni, R. Sacco, and F. Saleri, Numerical Mathematics, 2nd ed. Berlin, Germany: Springer, 2007.
[8] J. Sirignano and K. Spiliopoulos, "DGM: A deep learning algorithm for solving partial differential equations," Journal of Computational Physics, vol. 375, pp. 1339-1364, 2018.
[9] A. D. Jagtap, E. Kharazmi, and G. E. Karniadakis, "Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems," Computer Methods in Applied Mechanics and Engineering, vol. 365, article 113028, 2020.
[10] S. Wang, Y. Teng, and P. Perdikaris, "Understanding and mitigating gradient flow pathologies in physics-informed neural networks," SIAM Journal on Scientific Computing, vol. 43, no. 5, pp. A3055-A3081, 2021.