Comparative Study of Physics-Informed Neural Network Architectures for Steel Fatigue Strength Prediction under Data Scarcity

Yancai Sun[1,2,3,4, a], Wenzhong Deng[2,3,4, b], Yan He[1, *]

[1] College of Electromechanical Engineering, Qingdao University of Science and Technology, Qingdao 266061, China

[2] Guangxi Key Laboratory of Special Engineering Equipment and Control, Guilin University of Aerospace Technology, Guilin 541004, China

[3] University Engineering Research Center of Non-standard Intelligent Equipment and Process Control Technology, Guangxi, Guilin 541004, China

[4] Guangxi Region Precious Metal Materials Advanced Process Research Center, Guilin 541004, China

[*]Corresponding authors: heyan@qust.edu.cn

E-mail: [a]2024025@guat.edu.cn, [b]dwzh69@163.com

Abstract

The effect of architecture choice on physics-informed learning under data scarcity remains unclear for steel fatigue prediction. This study presents a comparative evaluation of four physics-informed neural network (PINN) architectures on the NIMS steel fatigue dataset with 437 samples: a loss-level baseline MLP-PINN, Residual Physics Network (ResPhys-Net), Multi-Fidelity PINN (MF-PINN), and Physics-Aware Attention PINN (PA-PINN). The models are assessed by 5-fold cross-validation and by repeated small-sample experiments with training fractions from 10% to 100%. In the full-data setting, PA-PINN achieves the highest mean R^2 of 0.985, followed closely by ResPhys-Net (0.985) and MF-PINN (0.984). Under the 10% training-data setting, ResPhys-Net and MF-PINN retain mean R^2 values of 0.884 and 0.879, respectively, compared with 0.828 for XGBoost and 0.815 for Random Forest. A complementary accuracy–stability analysis further indicates that residual and multi-fidelity architectures are less sensitive to repeated subsampling than attention-based fusion. These results suggest that architecture preference depends on the data regime: attention-based fusion is slightly advantageous with richer data, whereas residual and multi-fidelity designs show more stable behavior when the available fatigue data are very limited.

Keywords

Physics-informed neural networks, fatigue strength prediction, small-sample learning, residual physics network, multi-fidelity learning.

1. Introduction

Fatigue failure accounts for approximately 80–90% of all structural failures in engineering practice [1], making fatigue strength prediction a critical task in materials design and structural safety assessment. For steel components, the rotating bending fatigue strength at 10^7 cycles serves as a fundamental design parameter. However, fatigue testing is inherently expensive-a single test may require days to complete, and comprehensive characterization across different

compositions and heat treatment conditions demands extensive experimental campaigns, often resulting in limited datasets.

Traditional approaches rely on empirical correlations between composition, processing parameters, and mechanical properties. The Murakami relation ($\sigma f \approx 1.6 \cdot HV$) connects Vickers hardness to fatigue strength, while the Maynier equations estimate hardness from chemical composition and heat treatment history [7]. Although these physics-based models provide interpretable predictions, they are limited in accuracy due to oversimplified assumptions about complex metallurgical interactions such as carbide precipitation, grain boundary effects, and multi-phase transformations.

In recent years, data-driven machine learning (ML) methods have been widely used for materials property prediction. Agrawal et al. [2] demonstrated $R^2 > 0.95$ using ensemble methods on the NIMS steel fatigue dataset. Subsequent studies applied gradient boosting [8], Gaussian process regression, support vector regression, and deep neural networks to this task [9]. However, purely data-driven models still face limitations in small-sample scenarios: without sufficient training data to capture the underlying physical relationships, these models tend to overfit noise, produce physically inconsistent predictions, and exhibit poor extrapolation behavior.

Physics-informed neural networks (PINNs) [3] provide a framework for integrating domain knowledge into neural network training. By incorporating prior knowledge as soft constraints in the loss function, PINNs can regularize the learning process, particularly when data is limited. Recent advances in PINN architectures include PINNsFormer [4] for capturing temporal dependencies through Transformer-based architectures, PirateNets [5] for residual adaptive training that mitigates spectral bias, and composite multi-fidelity networks [6] that learn correlations between physics-based and data-driven predictions. Most PINN studies, however, focus on solving partial differential equations in fluid dynamics and heat transfer. The application of physics-informed learning to discrete materials property prediction-where the available knowledge consists of empirical metallurgical priors rather than PDE residuals-remains relatively underexplored. In the present work, "physics-informed" refers to empirical metallurgy-guided priors and regularization terms rather than exact governing laws. Furthermore, architecture-level physics integration has not been systematically compared for this steel-fatigue task domain.

1.1. Related Work

The integration of physics into ML for materials science has received increasing attention. In the fatigue domain, Chen et al. [9] proposed a general PINN framework incorporating Basquin and Walker models for metallic fatigue life prediction, achieving improved extrapolation to unseen loading conditions. Li et al. [8] developed a PINN framework for very high cycle fatigue of additively manufactured titanium alloys using fatigue indicator parameters derived from crystal plasticity simulations. For multi-fidelity learning, Meng and Karniadakis [6] demonstrated that a composite neural network can effectively learn correlations between low-fidelity physics models and high-fidelity experimental data. In the broader context of scientific machine learning, PINNsFormer [4] introduced Transformer-based temporal encoding to PINNs, while PirateNets [5] proposed residual adaptive training to address spectral bias in standard PINNs.

However, these studies predominantly employ standard MLP architectures with physics incorporated only at the loss function level. The architectural design space—including residual connections, multi-fidelity branching, and attention mechanisms for physics integration at the structural level-remains largely unexplored for materials property prediction tasks. To the best of our knowledge, these alternative integration strategies have not been compared

systematically for this steel-fatigue task under data-scarce conditions that are common in experimental materials science.

1.2. Contributions

This study makes three contributions. First, it provides a controlled comparison of four PINN architectures for steel fatigue strength prediction on a shared dataset and protocol. Second, it evaluates these models in two practically relevant regimes: full-data 5-fold cross-validation and repeated small-sample training from 10% to 100% of the available data. Third, it examines how the relative rankings change across data regimes, with PA-PINN leading slightly in full-data evaluation and ResPhys-Net and MF-PINN showing more stable low-data behavior.

The remainder of this paper is organized as follows. Section II describes the dataset, the four metallurgical priors, and the proposed architectures. Section III presents the full-data and small-sample results, the feature augmentation and ablation controls, and a discussion of the findings. Section IV concludes with a summary and directions for future work.

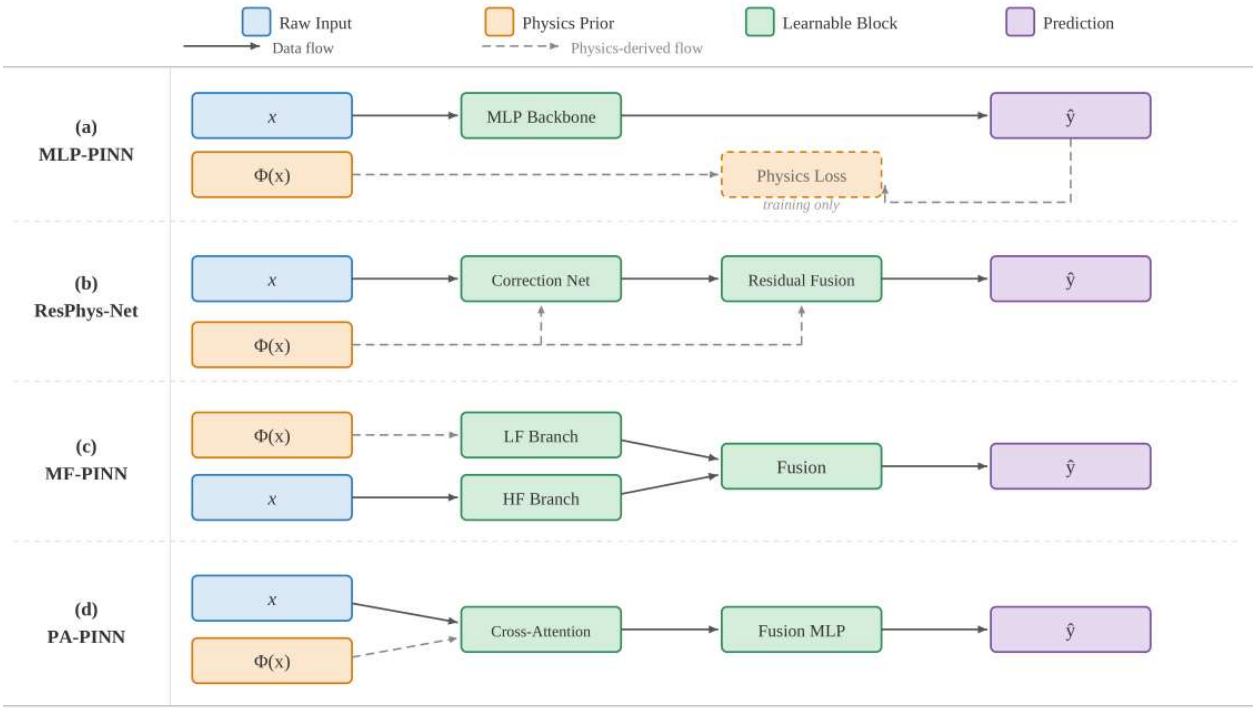

Fig. 1 Comparison of the four PINN architectures. Solid arrows: data flow; dashed arrows: physics-derived flow.

2. Methodology

2.1. Dataset Description

The NIMS (National Institute for Materials Science, Japan) steel fatigue dataset [2] comprises 437 steel samples with 25 input features and one target variable. The features include: (a) 9 chemical composition elements (C, Si, Mn, P, S, Ni, Cr, Cu, Mo in wt.%); (b) 4 upstream processing parameters (reduction ratio, inclusion area proportions); and (c) 12 heat treatment conditions (normalizing, through-hardening, carburization, diffusion, quenching, and tempering parameters). The target is rotating bending fatigue strength at 10^7 cycles, ranging from 225 to 1190 MPa (mean: 553, std: 187 MPa). All features are standardized using StandardScaler before training.

2.2. Physics-Based Domain Knowledge

Four metallurgical relations are encoded as empirical metallurgical priors. They are treated here as weak regularizers rather than exact governing laws, because their validity may vary across compositions, heat-treatment routes, and microstructural states:

(i) Hardness–Fatigue Relation. The Murakami approximation σf ≈ 1.6·HV [1], where Vickers hardness HV is estimated via a simplified Maynier equation [7]:

$$\mathrm{HV} \approx 293 + 614C + 45\mathrm{Si} + 36\mathrm{Mn} + 29\mathrm{Ni} + 58\mathrm{Cr} + 84\mathrm{Mo} - 0.4 \cdot \max(\mathrm{TT} - 200, 0) \quad (1)$$

(ii) Carbon Equivalence Ordering. The IIW carbon equivalent formula CE = C + Mn/6 + (Cr+Mo)/5 + (Ni+Cu)/15 provides a scalar measure of hardenability. Higher CE generally corresponds to higher fatigue strength, enforced via pairwise ordering constraints with 200 randomly sampled pairs per batch.

(iii) Tempering Temperature Ordering. Higher tempering temperature reduces hardness through precipitate coarsening, leading to lower fatigue strength. This is enforced as a pairwise monotonicity constraint.

(iv) Physical Range Bounds. Fatigue strength of engineering steels is bounded within [100, 1500] MPa, enforced via a ReLU penalty.

2.3. Proposed PINN Architectures

We propose four architectures representing a progression from loss-level to architecture-level physics integration, as illustrated in Fig. 1.

1) MLP-PINN (Baseline)

A standard multi-layer perceptron with physics-augmented loss: L = Ldata + λ·Lphysics, where Ldata = MSE(ŷ, y) and Lphysics is the weighted sum of the four physics constraint losses described above. This represents the conventional approach of loss-level physics integration.

2) ResPhys-Net (Residual Physics Network)

The main architectural feature is embedding the physics-based estimate as a residual skip connection:

$$\hat{y} = \sigma(\alpha) \cdot \Phi(\boldsymbol{x}) + \big(1 - \sigma(\alpha)\big) \cdot f_\theta([\boldsymbol{x};\, \varphi(\boldsymbol{x})]) \quad (2)$$

where Φ(x) is the Murakami–Maynier physics estimate from (1), φ(x) ∈ $\mathbb{R}^4$ denotes physics-derived features (CE, HV/500, σf^{est}/1000, TT/700), fθ is a correction network that takes the concatenation of standardized raw features and physics features as input, and α is a learnable scalar. Instead of learning the full input-output mapping, the network learns a residual correction to the physics estimate, analogous to residual learning in ResNets.

3) MF-PINN (Multi-Fidelity PINN)

Inspired by the composite neural network framework of Meng and Karniadakis [6], MF-PINN employs a dual-branch architecture that models both linear and nonlinear correlations between physics-based and data-driven predictions:

$$\hat{y} = \sigma(\alpha) \cdot y_{\mathrm{LF}} + \big(1 - \sigma(\alpha)\big) \cdot f_{\mathrm{NL}}([\boldsymbol{x};\, y_{\mathrm{LF}}]) \quad (3)$$

The low-fidelity (LF) branch is a small network (4 → 32 → 16 → 1) that maps physics features to a physics-informed initial estimate. The high-fidelity (HF) branch takes raw features

concatenated with yLF as input and learns the nonlinear correction. An auxiliary loss encourages the LF branch to approximate the Murakami–Maynier estimate, providing a physics-based "warm start" for the entire prediction pipeline. This dual-branch decomposition is motivated by the observation that low-fidelity physics models, although individually inaccurate, can stabilize learning trajectories when combined with data-driven corrections.

4) PA-PINN (Physics-Aware Attention PINN)

PA-PINN introduces cross-attention between physics-derived and raw features. The key idea is to let the physics context dynamically gate which raw features receive more weight during prediction, rather than treating all features equally:

$$\boldsymbol{h} = \mathrm{CrossAttn}(\mathrm{Q} = \varphi(\boldsymbol{x}), \mathrm{KV} = [\boldsymbol{x};\ \varphi(\boldsymbol{x})]) \tag{4}$$

where φ(x) serves as the query in multi-head cross-attention (default: 2 heads, d_model = 32), learning to dynamically weight raw input features based on the physics context. The attended representation is concatenated with a raw feature encoding and passed through a fusion MLP for final prediction. Unlike PINNsFormer [4] which uses attention for temporal dependencies, PA-PINN applies attention at the feature level to adaptively fuse physics knowledge with data-driven features.

Table 1. Summary of the Compared Architectures

Model	Physics integration	Expected benefit
MLP-PINN	Physics only in loss	Simple baseline with soft regularization
ResPhys-Net	Residual skip from Φ(x)	Stable correction around a physics prior
MF-PINN	Low-/high-fidelity branches	More stable behavior with limited training data
PA-PINN	Cross-attention on φ(x)	Physics-conditioned feature fusion

Table 1 summarizes the four architectures and their expected benefits. The progression from MLP-PINN to PA-PINN represents increasing structural coupling between physics knowledge and the neural network. All four share the same physics loss terms, allowing the comparison to isolate the effect of architectural design choices on prediction accuracy and stability. Note that MLP-PINN and ResPhys-Net have comparable parameter counts (4.5k and 51.9k), whereas PA-PINN (25.2k) introduces attention overhead. The computational implications are examined in Section III-E.

2.4. Physics Constraint Implementation

All four architectures share the same physics loss Lphysics, computed in the original (non-standardized) scale to preserve the units of the empirical prior terms. In this study, these terms are used as weak empirical regularizers rather than as truth-enforcing constraints. The hardness constraint uses a Huber-like formulation with ±150 MPa tolerance: Lhard = mean(ReLU($|\hat{y} - \sigma f^{est}| - 150$)^2) / 500^2. The ordering constraints (CE and TT) use pairwise comparison: Lorder = mean(ReLU(−Δfeat·Δpred)) for pairs with |Δfeat| > τ. An adaptive warmup schedule linearly increases λ from 0 to its target value over the first 100 epochs, allowing the network to first learn basic data patterns before the regularization terms are fully enforced.

2.5. Training Configuration

All PINN models are trained with AdamW optimizer, cosine annealing with warm restarts (T_0 = 100, Tmult = 2), gradient clipping (max norm 1.0), and a maximum of 800 epochs with batch size 64. During training, model checkpoints are selected according to validation R^2. Bayesian hyperparameter optimization via Optuna [10] (10 trials, inner 3-fold CV) is applied to each

PINN architecture and to the conventional baselines using the same tuning budget. In the 5-fold evaluation, the inner tuning is performed on the training portion of each outer fold only. Conventional baselines include ANN, Random Forest (RF), XGBoost, SVR, and GPR. The repeated small-sample experiments use a fixed 20% test split and training fractions of 10%, 20%, 30%, 50%, and 100%, repeated ten times per fraction, with all models sharing the same split and subsampled training indices within each repeat.

3. Results and Discussion

3.1. Full-Data Performance (5-Fold Cross-Validation)

Table 2 and Fig. 2 summarize the 5-fold cross-validation results. PA-PINN has the highest mean R^2 (0.9850) and the lowest RMSE (22.49 MPa), with ResPhys-Net and MF-PINN following closely. However, the bootstrap intervals overlap strongly, and no pairwise Wilcoxon comparison among the leading methods remains significant after Bonferroni correction. This pattern suggests that several architectures reach a similar performance level when all 437 samples are used, and that architecture choice mainly affects mean ranking and stability.

Table 2. 5-Fold CV Results (mean ± std). RMSE and MAE in MPa.

Method	R^2	RMSE	MAE
PA-PINN	0.9850 ± 0.003	22.49 ± 2.50	16.32 ± 1.17
ResPhys-Net	0.9846 ± 0.005	22.53 ± 3.18	15.81 ± 1.70
MF-PINN	0.9843 ± 0.003	23.02 ± 2.54	15.67 ± 1.85
MLP-PINN	0.9832 ± 0.004	23.68 ± 2.86	16.25 ± 1.68
XGBoost	0.9824 ± 0.005	24.10 ± 2.59	15.88 ± 0.48
SVR	0.9801 ± 0.006	25.49 ± 4.18	17.95 ± 3.36
RF	0.9759 ± 0.008	28.14 ± 2.95	19.47 ± 1.35
ANN	0.9590 ± 0.022	35.70 ± 7.13	26.15 ± 5.38

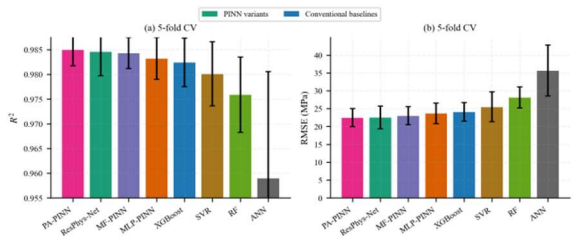

Fig. 2 5-fold cross-validation comparison for the stable methods. (a) Mean R^2 with standard deviation. (b) Mean RMSE. The four PINN variants cluster near the top, while XGBoost remains the best-performing conventional baseline.

3.2. Small-Sample Performance

Table 3 and Fig. 3 report the repeated small-sample results. The regime with 10% training data corresponds to roughly 35 training samples and is the most challenging setting. In this regime, ResPhys-Net obtains the highest mean R^2 (0.884), followed closely by MF-PINN (0.879) and MLP-PINN (0.878). All four PINN variants show higher repeated-run means than XGBoost (0.828) and RF (0.815), while ANN, SVR, and GPR become unstable. The 10% bootstrap intervals remain broad and overlapping, so the low-data comparison should be interpreted as a favorable average accuracy–stability trend rather than as statistically definitive superiority. Notably, the performance gap between PINN variants and XGBoost narrows as the training fraction increases beyond 50%, suggesting that architectural advantages are most valuable precisely when data is most scarce.

Table 3. R^2 Under Varying Training Data Availability (mean of 10 runs)

Method	10%	20%	30%	50%	100%
ResPhys-Net	.884	.950	.958	.982	.991
MF-PINN	.879	.951	.955	.981	.990
MLP-PINN	.878	.945	.953	.979	.990
PA-PINN	.846	.947	.937	.981	.990
XGBoost	.828	.929	.934	.977	.987
RF	.815	.922	.929	.974	.986

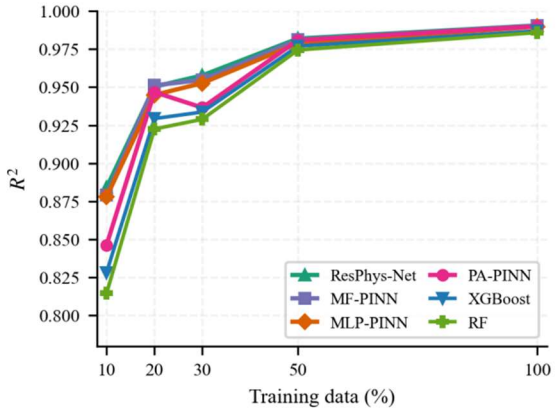

Fig. 3 Small-sample performance of the four PINN variants and the two leading conventional baselines. ResPhys-Net and MF-PINN maintain the highest mean performance at 10%–30% training data, while the gap decreases as more samples are added.

3.3. Structured Integration vs. Naive Feature Augmentation

To test whether the observed gains can be explained simply by appending a few physics-derived features, we also added the same feature set $\varphi(x)$ to several conventional baselines. Table 4 shows that naive augmentation is inconsistent: RF benefits slightly, SVR changes little, and XGBoost and ANN both degrade. In contrast, all four structured PINN variants remain above these augmented baselines in 5-fold CV. A standalone Murakami–Maynier surrogate preserves rank information ($r = 0.785$) but remains inaccurate ($R^2 = -1.27$), so the prior is informative but incomplete.

Table 4. Effect of Appending Physics Features to Conventional Baselines

Model	Base R^2	$+\varphi(x)$	ΔR^2
XGBoost	0.9824	0.9814	−0.0010
RF	0.9759	0.9780	+0.0021
SVR	0.9801	0.9800	−0.0001
ANN	0.9590	0.8907	−0.0683

3.4. Focused Ablation of ResPhys-Net

We further examined ResPhys-Net through a matched ablation using the same fixed holdout and repeated subsampling seeds as the main small-sample benchmark. Table 5 and Fig. 4 show that the largest drop occurs when ResPhys-Net is replaced with a plain MLP, especially at 10% data (0.887 to 0.835). By contrast, removing only the additional physics loss leaves the mean R^2 nearly unchanged, and removing the residual skip causes only a small decrease. Under the present data regime and constraint strength, this result suggests that the structured residual representation contributes more than the auxiliary penalty term itself. This observation is consistent across all data fractions: the skip connection provides a stable performance floor, while the physics loss contributes primarily to convergence speed rather than final accuracy.

Table 5. Focused Ablation of ResPhys-Net Under the Matched Small-Sample Protocol (R^2, mean of 10 runs)

Variant	10%	20%	30%	100%
Full model	.887	.953	.958	.990
w/o physics loss	.887	.953	.958	.990
w/o skip connection	.887	.951	.953	.989
Plain MLP	.835	.943	.947	.989

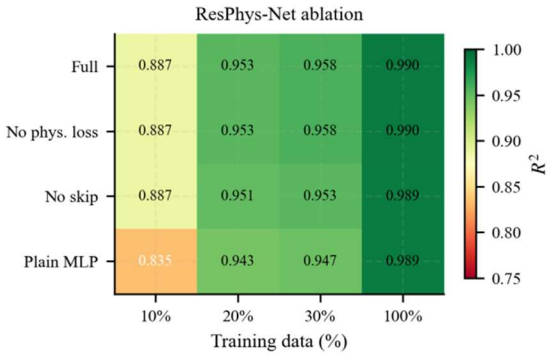

Fig. 4 Focused component ablation for ResPhys-Net under the matched small-sample protocol. The largest degradation appears when the model is replaced with a plain MLP, while removing the extra physics loss changes little.

3.5. Computational Cost

Beyond accuracy, practical deployment also depends on training and inference cost. Table 6 summarizes the relative runtime and model size measured in our current implementation. XGBoost remains the cheapest option to train, while the PINN models require about 9–13 s per run. Among the PINN variants, MLP-PINN is the lightest, whereas ResPhys-Net and PA-PINN trade additional parameters for stronger physics-guided structure. Even so, the inference times remain in the millisecond range, which is acceptable for offline fatigue screening and batch materials evaluation.

Table 6. Relative Computational Cost

Model	Train (s)	Infer. (ms)	Params
MLP-PINN	9.36	0.38	4.5k
ResPhys-Net	10.34	2.12	51.9k
MF-PINN	12.84	1.99	4.8k
PA-PINN	11.22	2.66	25.2k
XGBoost	0.88	0.10	—

3.6. Low-Data Accuracy–Stability Trade-off

Fig. 5 summarizes the average behavior over the 10%–30% regime, where model choice matters most. ResPhys-Net and MF-PINN are located toward the high-accuracy and low-variability corner relative to XGBoost and RF, whereas PA-PINN remains competitive in mean accuracy but shows larger variability across repeated subsampling. Similar limited-data benefits from structured physics guidance have also been noted in broader reviews of physics-informed machine learning and in other materials-property estimation tasks with scarce data [11,12,13].

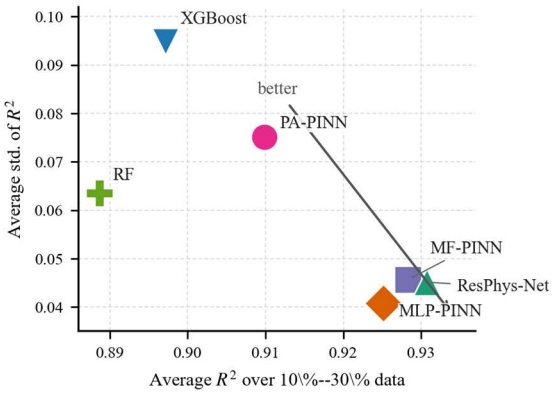

Fig. 5 Average accuracy–stability trade-off over the 10%–30% data regime. Higher values on the horizontal axis and lower values on the vertical axis correspond to higher mean accuracy with smaller run-to-run variability.

3.7. Discussion

The results suggest that the relative merits of these architectures depend on the amount of available training data. When all 437 samples are used, the differences among the leading PINN variants are small: PA-PINN reaches a mean R^2 of 0.9850, compared with 0.9846 for ResPhys-Net and 0.9843 for MF-PINN, and their bootstrap intervals overlap substantially. In this regime, architecture choice appears to influence mean ranking more than it changes the attainable performance level. By contrast, the separation becomes more visible in the 10%–30% regime, where ResPhys-Net and MF-PINN provide the highest repeated-run means. Their common characteristic is an explicit physics-guided prediction path, implemented either as a residual correction around a prior estimate or as a low-/high-fidelity decomposition. This reading is also in line with recent analyses showing that PINN performance depends strongly on the match between optimization, constraint balance, and problem representation [14,15].

The comparison with conventional baselines and auxiliary controls helps clarify where the gain is likely to come from. XGBoost remains the best-performing conventional baseline and approaches the PINN models once more than 50% of the training data are used, but its gap to the leading PINN architecture widens again at 10% data. At the same time, simply appending $\varphi(x)$ to a conventional learner does not produce a consistent benefit: the 5-fold R^2 change ranges from −0.0010 for XGBoost to +0.0021 for RF, and ANN drops by 0.0683 after augmentation. The matched ResPhys-Net ablation also indicates that, in the present setting, structured coupling contributes more than the auxiliary physics-loss term alone. For PA-PINN, a simpler gated-fusion control reaches $R^2 = 0.9845 \pm 0.0038$ in 5-fold CV, close to the full attention model (0.9851 ± 0.0031) and above the pure concatenation variant without attention (0.9835 ± 0.0043). Taken together, these results support the interpretation that the benefit of physics guidance in this task is associated more with structured conditioning than with attention alone or with direct feature stacking.

From a practical perspective, the low-data gain is not negligible relative to the added computational cost. At 10% training data, ResPhys-Net increases mean R^2 from 0.828 for XGBoost to 0.884 and reduces MAE from 44.0 to 39.6 MPa, while MF-PINN reaches a comparable R^2 of 0.879. This improvement is accompanied by a training-time increase from 0.88 s for XGBoost to about 10–13 s for the PINN models, whereas inference remains within 2–3 ms per sample. The present conclusions are nevertheless limited to the NIMS rotating-bending fatigue dataset and to the empirical physics relations encoded here. They should therefore be interpreted as evidence for this steel-fatigue benchmark rather than as a universal ranking of PINN architectures.

4. Conclusion

This study presents a comparative evaluation of four PINN architectures for steel fatigue strength prediction under data scarcity. Across the full-data 5-fold setting, PA-PINN attains the highest mean R^2 of 0.985, but the numerical differences among the leading PINN variants are small and their confidence intervals overlap. Under repeated low-data training, the contrast becomes clearer: ResPhys-Net and MF-PINN achieve the highest repeated-run means in the 10%–30% regime, reaching R^2 = 0.884 and 0.879 at 10% data, although the associated intervals remain broad.

The additional controls indicate that the metallurgical prior is informative but insufficient as a standalone predictor, and that its value is greater when it is embedded through structured model coupling than when it is appended directly as extra features. For this benchmark, physics-conditioned fusion, including the attention-based variant, is slightly advantageous with richer data, whereas residual and multi-fidelity designs provide more stable behavior when the training set is severely limited. The present findings are benchmark-specific and should not be interpreted as a universal ranking of PINN architectures. A natural next step is to test whether the same pattern persists on additional fatigue datasets and under broader variations in composition and heat-treatment regimes. Extending the evaluation to multi-output fatigue properties, incorporating microstructure-informed priors, and exploring transfer learning across alloy families are promising directions for strengthening the practical applicability of physics-informed fatigue models.

Several limitations should be noted. First, the metallurgical priors used here are simplified empirical relations that may not generalize to all steel grades or heat-treatment routes. Second, the dataset contains only rotating-bending fatigue data at a single life criterion (10^7 cycles), and the conclusions may differ for other loading modes such as axial or torsional fatigue. Third, hyperparameter sensitivity was not exhaustively explored due to computational constraints. Despite these limitations, the controlled comparison protocol and the consistency of the observed trends across multiple repeated runs provide a reliable basis for the conclusions drawn.

Acknowledgments

This work was supported by the Guangxi Natural Science Foundation (2025GXNSFAA069565) and the Shandong Province Key R&D Program (2025CXGC010506). The authors thank NIMS of Japan for providing the public steel-fatigue dataset.

References

[1] Murakami, Y. (2002). Metal fatigue: Effects of small defects and nonmetallic inclusions. Elsevier.

[2] Agrawal, A., Deshpande, P. D., Cecen, A., Basavarsu, G. P., Choudhary, A. N., & Kalidindi, S. R. (2014). Exploration of data science techniques to predict fatigue strength of steel from composition and processing parameters. Integrating Materials and Manufacturing Innovation, 3, 8.

[3] Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics, 378, 686–707.

[4] Zhao, Z., Ding, X., & Prakash, B. A. (2024). PINNsFormer: A transformer-based framework for physics-informed neural networks. In Proceedings of ICLR.

[5] Wang, S., Li, B., Chen, Y., & Perdikaris, P. (2024). PirateNets: Physics-informed deep learning with residual adaptive networks. Journal of Machine Learning Research, 25, 1–48.

[6] Meng, X., & Karniadakis, G. E. (2020). A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse PDE problems. Journal of Computational Physics, 401, 109020.

[7] Maynier, P., Jungmann, B., & Dollet, J. (1978). Creusot-Loire system for the prediction of the mechanical properties of low alloy steel products. In D. V. Doane & J. S. Kirkaldy (Eds.), Hardenability concepts with applications to steel (pp. 518–545). AIME.

[8] Li, H., Sun, G., Tian, Z., Huang, K., & Zhao, Z. (2024). A physics-informed neural network framework based on fatigue indicator parameters for very high cycle fatigue life prediction of an additively manufactured titanium alloy. Fatigue & Fracture of Engineering Materials & Structures, 47(9), 3171–3188.

[9] Zhou, S., Henrich, M., Wei, Z., Feng, F., Yang, B., & Münstermann, S. (2025). A general physics-informed neural network framework for fatigue life prediction of metallic materials. Engineering Fracture Mechanics, 322, 111136.

[10] Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In Proceedings of ACM SIGKDD (pp. 2623–2631).

[11] Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021). Physics-informed machine learning. Nature Reviews Physics, 3, 422–440.

[12] Guo, J., Zhu, H., Yang, Y., & Guo, C. (2025). Advances in physics-informed neural networks for solving complex partial differential equations and their engineering applications: A systematic review. Engineering Applications of Artificial Intelligence, 161, 112044.

[13] Jo, J., Jeong, Y., Kim, J., & Yoo, J. (2024). Thermal conductivity estimation using physics-informed neural networks with limited data. Engineering Applications of Artificial Intelligence, 137, 109079.

[14] Wang, S., Yu, X., & Perdikaris, P. (2022). When and why PINNs fail to train: A neural tangent kernel perspective. Journal of Computational Physics, 449, 110768.

[15] Krishnapriyan, A. S., Gholami, A., Zhe, S., Kirby, R. M., & Mahoney, M. W. (2021). Characterizing possible failure modes in physics-informed neural networks. In Advances in Neural Information Processing Systems (Vol. 34, pp. 26548–26560).