Residual-Guided Randomized Neural Networks

Mushir Akhtar
Department of Mathematics
Indian Institute of Technology Indore
phd2101241004@iiti.ac.in

M. Tanveer*
Department of Mathematics
Indian Institute of Technology Indore
mtanveer@iiti.ac.in

Mohd. Arshad
Department of Mathematics
Indian Institute of Technology Indore
arshad@iiti.ac.in

***Abstract*—Randomized neural networks enable fast and analytically tractable training by fixing the input to hidden layer parameters at random and learning the output weights in closed form; however, their performance critically depends on a single uninformed draw of hidden units. This one shot and task uninformed feature construction often leads to redundant representations and suboptimal utilization of model capacity. To address this limitation, we propose a simple and broadly applicable residual guided procedure that greedily constructs the hidden layer using a closed form residual decrease criterion. At each stage, we (i) generate a pool of random candidate units, (ii) score each candidate by the exact reduction it induces in the ridge regularized objective, (iii) select the top k units, and (iv) refit the readout in closed form using the standard design with direct input links. This procedure yields a progressive training process with a guaranteed monotonic decrease of the training objective. The method is model agnostic: only the candidate generation is architecture specific, while the scoring selection refitting loop is shared across models. Extensive experiments on 71 benchmark datasets from the UCI repository, covering both binary and multiclass classification tasks, demonstrate that the proposed residual-guided models consistently outperform their baseline counterparts in terms of accuracy, stability, and overall ranking performance. Code and Supplementary are available at https://github.com/mtanveer1/RG-RNN.**

***Index Terms*—Randomized neural networks, residual guided learning, closed form training, random feature construction, greedy incremental learning.**

I. Introduction

RANDOMIZED neural networks (RaNNs) have attracted significant attention as efficient alternatives to deep learning models, particularly in scenarios involving limited data, strict computational constraints, or real-time requirements [1, 2, 3]. Unlike deep architectures trained via iterative backpropagation, RaNNs fix the input-to-hidden layer parameters at random and learn only the output-layer weights, typically using closed-form. This decoupling eliminates gradient-based optimization, substantially reduces training complexity, and yields analytically tractable learning procedures while retaining universal approximation capability under mild conditions [4, 5].

Within this paradigm, several representative RaNN architectures have been proposed, differing primarily in how random features are constructed and incorporated into the predictive model. The Random Vector Functional Link (RVFL) network [1, 6] is a widely adopted formulation that combines nonlinear random hidden features with direct input-to-output links, enabling the preservation of linear input structure alongside nonlinear representations. The Extreme Learning Machine (ELM) [7] can be viewed as a special case of RVFL in which the direct input connections are removed, further simplifying the architecture while maintaining fast closed-form training. To enhance representational diversity without increasing depth, the Broad Learning System (BLS) [8, 9] was proposed as an alternative that expands the network horizontally by incrementally adding feature and enhancement nodes, enabling efficient incremental learning without retraining from scratch.

Building upon these foundational architectures, extensive research has focused on improving RVFL, ELM, and BLS models in terms of stability, robustness, and generalization [6, 10]. Representative efforts include architectural refinements [11, 12], regularization strategies [13, 14], ensemble formulations [15, 16], and incremental learning mechanisms [17, 18], many of which report enhanced empirical performance across diverse learning scenarios. Recently, copula-aligned weight initialization (CAWI) has been proposed to incorporate dependency-aware copula modeling into weight initialization, aligning randomized weights with the intrinsic structure of the input data and thereby improving representation quality and learning performance [19].

Despite this progress, most existing extensions share a common structural limitation: hidden representations are still generated through fixed, randomly sampled feature mappings. Consequently, increasing the number of hidden nodes or feature groups often introduces redundancy, leading to inefficient utilization of model capacity and limited adaptability to the data distribution. In particular, indiscriminate expansion with additional random features provides no guarantee that newly introduced representations contribute meaningfully to reducing the supervised residual error.

These observations motivate the need for randomized learning frameworks that retain closed-form training while enabling principled and data-driven feature expansion. In this work, we propose a residual-guided randomized neural network framework that incrementally constructs hidden representations by explicitly leveraging the current residual of the model. At each stage, randomly generated candidate features are evaluated based on their closed-form contribution to residual reduction, and only the most effective features are retained. This residual-guided strategy enables adaptive and efficient feature construction while preserving the computational efficiency and analytical tractability of randomized neural networks. Importantly, the

*Corresponding author.

proposed framework is model-agnostic and can be seamlessly integrated into a broad class of RaNNs, including RVFL, ELM, BLS, and their variants, without modifying their fundamental training procedures.

The main contributions of this work are summarized as follows:

1) We propose a general residual-guided learning framework for randomized neural networks that enables adaptive, data-driven hidden feature expansion while retaining closed-form output weight learning.
2) The proposed framework is model-agnostic and can be seamlessly integrated into a broad class of randomized neural networks, including RVFL, ELM, BLS, and their variants, without altering their core architectures or training paradigms.
3) We provide a theoretical monotonicity guarantee, establishing that the ridge-regularized training objective is non-increasing with the progressive addition of residual-guided hidden features, which is further validated through empirical analysis.
4) Extensive experimental evaluations on 71 benchmark datasets covering both binary and multiclass classification tasks demonstrate that the proposed residual-guided models consistently outperform their baseline counterparts.

The remainder of this paper is organized as follows. Section II presents the preliminaries. Section III provides the motivation and problem statement. Section IV introduces the proposed framework. Section V reports experimental results. Finally, Section VI concludes the paper.

II. Preliminaries

This section establishes the notation and reviews the Random Vector Functional-Link (RVFL) network, one of the standard randomized neural network.

A. Notation

Let $\mathbf{U} \in \mathbb{R}^{n\times p}$ denote the input data matrix with n samples and p features, and let $\mathbf{T} \in \mathbb{R}^{n\times q}$ denote the corresponding target matrix, where q is the output dimension (for classification, $\mathbf{T}$ is typically one-hot encoded). Randomized hidden-layer activations are collected in $\mathbf{G} \in \mathbb{R}^{n\times h}$, generated via random affine transformations of $\mathbf{U}$ followed by an element-wise nonlinearity $\phi(\cdot)$. Each hidden unit is parameterized by a random weight vector $\mathbf{w} \in \mathbb{R}^p$ and a bias $\beta \in \mathbb{R}$, yielding the random weight matrix $\mathbf{W} \in \mathbb{R}^{p\times h}$. The augmented feature matrix is defined as $\mathbf{A} = [\,\mathbf{U}\ \mathbf{G}\,] \in \mathbb{R}^{n\times(p+h)}$. The output-layer parameter matrix is denoted by $\mathbf{\Omega} \in \mathbb{R}^{(p+h)\times q}$; when an explicit bias term is included, $\mathbf{A}$ is augmented with a column of ones. The transpose operator is denoted by $(\cdot)^\top$, $\mathbf{e}_n \in \mathbb{R}^n$ denotes the vector of ones, and $\mathbf{I}_k$ denotes the $k \times k$ identity matrix.

B. Random Vector Functional-Link Network (RVFL)

RVFL is a shallow feedforward architecture consisting of input, hidden, and output layers, distinguished by the presence of direct links from the input layer to the output layer. These direct links enable RVFL to jointly exploit linear structure from the raw inputs and nonlinear structure from randomized hidden features, often improving stability and generalization. The standard RVFL architecture is illustrated in Fig. 1.

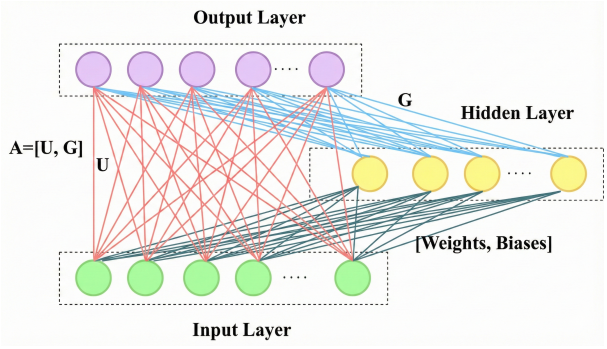

Fig. 1: Architecture of a standard RVFL network.

In RVFL, the hidden-layer parameters are randomly sampled once and kept fixed throughout training. Given $\mathbf{W} \in \mathbb{R}^{p\times h}$ and bias $\beta \in \mathbb{R}$, the hidden-layer activations for input matrix $\mathbf{U} \in \mathbb{R}^{n\times p}$ are computed as

$$\mathbf{G} = \phi\big(\mathbf{U}\mathbf{W} + \mathbf{e}_n\beta\big) \in \mathbb{R}^{n\times h}.$$

The output layer operates on the augmented representation $\mathbf{A} = [\,\mathbf{U}\ \mathbf{G}\,] \in \mathbb{R}^{n\times(p+h)}$ and learns the readout matrix $\mathbf{\Omega}$ by solving the ridge-regularized least-squares problem

$$\mathbf{\Omega}^\star = \arg\min_{\mathbf{\Omega}}\ \|\mathbf{A}\mathbf{\Omega} - \mathbf{T}\|_F^2 + \lambda\|\mathbf{\Omega}\|_F^2, \qquad \lambda > 0. \tag{1}$$

This problem admits a closed-form solution, which can be expressed in either primal or dual form depending on the relative dimensions of $\mathbf{A}$,

$$\mathbf{\Omega}^\star = \begin{cases} (\mathbf{A}^\top\mathbf{A} + \lambda\mathbf{I}_{p+h})^{-1}\mathbf{A}^\top\mathbf{T}, & \text{if } (p+h) \le n, \\ \mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top + \lambda\mathbf{I}_n)^{-1}\mathbf{T}, & \text{otherwise.} \end{cases} \tag{2}$$

III. Motivation & Problem Statement

RaNNs are attractive since they eliminate backpropagation while enhancing linear models with nonlinear features. However, their performance critically depends on a *single, task-uninformed draw* of randomized hidden features. This section clarifies why such a design is fragile and formulates the precise optimization problem addressed in this work.

Note: For clarity, we present the subsequent discussion in the RVFL setting, a representative RaNN architecture, and note that the proposed framework extends naturally to other RaNNs.

Setting: Let $\mathbf{U} \in \mathbb{R}^{n\times p}$ denote the input data matrix and $\mathbf{T} \in \mathbb{R}^{n\times q}$ the corresponding target matrix. A randomized mapping generates hidden-layer activations $\mathbf{G} \in \mathbb{R}^{n\times h}$, which are concatenated with the inputs to form the augmented matrix $\mathbf{A} = [\,\mathbf{U}\ \ \mathbf{G}\,] \in \mathbb{R}^{n\times(p+h)}$. The output layer uses a linear

readout $\mathbf{\Omega} \in \mathbb{R}^{(p+h)\times q}$, learned via ridge-regularized least squares,

$$\mathcal{J}(\mathbf{G}) \triangleq \min_{\mathbf{\Omega}\in\mathbb{R}^{(p+h)\times q}} \left\| [\,\mathbf{U}\;\; \mathbf{G}\,]\,\mathbf{\Omega} - \mathbf{T} \right\|_F^2 + \lambda \|\mathbf{\Omega}\|_F^2. \quad (3)$$

For any fixed $\mathbf{G}$, the minimizer of (3) is available in closed form. Consequently, the *effective design problem* is to construct $\mathbf{G}$ so as to minimize $\mathcal{J}(\mathbf{G})$ under a prescribed hidden-unit budget.

A. Core Limitations

In the standard RVFL formulation, $\mathbf{G}$ is generated once by sampling a random affine map and applying a nonlinearity elementwise. This one-shot, label-uninformed construction leads to several practical shortcomings:

1) **Feature misalignment and inefficiency.** The randomly generated columns of $\mathbf{G}$ are not tailored to the supervised task and need not align with discriminative directions in the data. Consequently, many hidden units may exhibit weak correlation with the *supervised residual* after fitting the linear readout, leading to inefficient use of model capacity.
2) **Lack of adaptive feature selection.** Once $\mathbf{G}$ is fixed, ridge regularization can only uniformly shrink the output weights; it cannot selectively suppress or discard uninformative or noisy hidden features. This restriction yields a suboptimal bias-variance trade-off and can degrade generalization when h is constrained.
3) **Sensitivity to random initialization and lack of progressive guarantees.** Performance is highly sensitive to the random seed, as the entire representation depends on a single draw of $\mathbf{G}$. Moreover, there is no principled mechanism for progressive model building: intermediate subsets of hidden units are not guaranteed to yield monotonic improvement in the training objective or predictive performance as additional hidden units are added.

B. Problem Statement

Let $\mathcal{S}$ denote the set of admissible hidden features generated by the randomized mapping,

$$\mathcal{S} = \{\mathbf{g} \in \mathbb{R}^n \mid \mathbf{g} = \phi(\mathbf{U}\mathbf{w} + \mathbf{e}_n\beta),\ (\mathbf{w},\beta) \sim \mathcal{P}\}, \quad (4)$$

where $\mathcal{P}$ denotes a prescribed sampling distribution over $\mathbb{R}^p \times \mathbb{R}$.

Given a hidden-unit budget $h \in \mathbb{N}$ and a regularization parameter $\lambda > 0$, the objective is to select a collection of h hidden features from $\mathcal{S}$ that minimizes the ridge-regularized training loss. Formally, the design problem is

$$\min_{\mathbf{G}=[\,\mathbf{g}_1\ \cdots\ \mathbf{g}_h\,]} \mathcal{J}(\mathbf{G}) \quad \text{s.t.} \quad \mathbf{g}_j \in \mathcal{S}\ \ \forall j,\ \ \mathbf{G} \in \mathbb{R}^{n\times h}. \quad (5)$$

Equivalently, (5) seeks a size-r multiset of hidden features drawn from $\mathcal{S}$ such that, when concatenated with the direct input links $\mathbf{U}$, the resulting augmented representation minimizes the ridge objective in (3). This formulation defines a *combinatorial feature selection problem* over a large and typically continuous set of admissible features, rendering exact optimization computationally intractable.

Desired Properties: A practical alternative to one-shot RVFL should:

1) preserve *closed-form* training at the output layer (no backpropagation);
2) *adapt* hidden-unit selection to the supervised signal by prioritizing features that most reduce the current objective;
3) admit an *anytime* training procedure with a *monotonic decrease* of the objective as hidden nodes are added.

Our Approach: We address (5) via a *stagewise, residual-guided* construction. At each iteration, a manageable candidate pool is sampled from $\mathcal{S}$. Each candidate is scored by its *closed-form reduction* in (3) relative to the current model, the most effective units are selected, and the readout $\mathbf{\Omega}$ is refit exactly on the updated augmented matrix. This procedure directly targets the limitations of standard RVFL while retaining its hallmark efficiency and closed-form training.

IV. Method

In this section, we propose a residual-guided construction for randomized neural networks that retains the key advantages of fixed randomized hidden mappings and closed-form training while overcoming the limitations of one-shot hidden-layer sampling. The core idea is to replace the single, task-uninformed draw of hidden features with a supervised, stagewise selection procedure guided by the current residual error.

A. Training Objective

Given an input matrix $\mathbf{U} \in \mathbb{R}^{n\times p}$ and a hidden-feature matrix $\mathbf{G} \in \mathbb{R}^{n\times h}$, RVFL forms the augmented representation $\mathbf{A} = [\,\mathbf{U}\;\; \mathbf{G}\,]$ and learns a linear readout $\mathbf{\Omega} \in \mathbb{R}^{(p+h)\times q}$ via ridge-regularized least squares,

$$\mathbf{\Omega}^\star(\mathbf{A},\lambda) = \arg\min_{\mathbf{\Omega}} \|\mathbf{A}\mathbf{\Omega} - \mathbf{T}\|_F^2 + \lambda\|\mathbf{\Omega}\|_F^2, \qquad \lambda > 0. \quad (6)$$

The resulting training objective value is

$$\mathcal{J}(\mathbf{G}) = \min_{\mathbf{\Omega}} \|\,[\,\mathbf{U}\;\; \mathbf{G}\,]\mathbf{\Omega} - \mathbf{T}\|_F^2 + \lambda\|\mathbf{\Omega}\|_F^2. \quad (7)$$

Since the minimizer in (6) is available in closed form, learning in RVFL reduces to constructing a hidden-feature matrix $\mathbf{G}$, under a prescribed hidden-unit budget h, that yields a small value of $\mathcal{J}(\mathbf{G})$. This observation motivates a direct focus on how hidden features are generated and selected.

B. Residual-Guided Stagewise Construction

Rather than generating all h hidden features in a single step, the proposed method constructs the hidden-feature matrix $\mathbf{G}$ incrementally in a stagewise manner.

Let $\mathbf{G}_t \in \mathbb{R}^{n\times h_t}$ denote the hidden-feature matrix after stage t, with $h_0 = 0$, and define the corresponding augmented representation as $\mathbf{A}_t = [\,\mathbf{U}\;\; \mathbf{G}_t\,]$. At each stage, the current model is first fitted exactly by solving the ridge regression

problem, yielding the optimal readout and the associated residual

$$\mathbf{\Omega}_t = \mathbf{\Omega}^\star(\mathbf{A}_t, \lambda), \qquad \mathbf{R}_t = \mathbf{A}_t\mathbf{\Omega}_t - \mathbf{T}. \tag{8}$$

The residual $\mathbf{R}_t$ represents the portion of the supervised signal that is not yet captured by the current representation. Since the training objective $\mathcal{J}(\mathbf{G}_t)$ depends directly on this residual, reducing $\|\mathbf{R}_t\|_F$ leads to a decrease in the objective value.

Each stage of the algorithm then proceeds by (i) generating a temporary pool of candidate random hidden features, (ii) evaluating each candidate according to its ability to reduce the current residual, and (iii) selecting a small subset of the most effective features to be permanently added to the model. After feature selection, the readout is refitted exactly using the augmented representation, and the process is repeated until the hidden-unit budget h is reached or a stopping criterion is satisfied. The individual steps of candidate generation, residual-based scoring, and selection are detailed below.

1) Candidate generation: At stage t, a temporary pool of M candidate hidden features is generated. Specifically, random affine parameters are drawn from the sampling distribution $\mathcal{P}$ and transformed using the activation function $\phi(\cdot)$,

$$\mathbf{g}^{(j)} = \phi(\mathbf{U}\mathbf{w}^{(j)} + \mathbf{e}_n\beta^{(j)}), \quad (\mathbf{w}^{(j)}, \beta^{(j)}) \sim \mathcal{P}, \quad j = 1, \ldots, M. \tag{9}$$

The resulting vectors $\{\mathbf{g}^{(j)}\}_{j=1}^M$ form a candidate pool that is used only at the current stage, from which a small number of informative features will be selected and permanently added to the model.

2) Residual-based scoring: Each candidate hidden feature is evaluated based on its ability to reduce the current training objective. For a candidate $\mathbf{g}$, we consider augmenting the current representation $\mathbf{A}_t$ with this feature and optimizing only its associated coefficient, while keeping the previously learned readout parameters fixed. Under this setting, the optimal coefficient and the resulting *exact decrease* in the ridge-regularized objective are given by

$$\mathbf{\Omega}_{\mathbf{g}}^\star = \frac{\mathbf{g}^\top\mathbf{R}_t}{\|\mathbf{g}\|_2^2 + \lambda}, \qquad \Delta(\mathbf{g}) = \frac{\|\mathbf{g}^\top\mathbf{R}_t\|_F^2}{\|\mathbf{g}\|_2^2 + \lambda}. \tag{10}$$

The score $\Delta(\mathbf{g})$ therefore quantifies the immediate reduction in the training objective that would be achieved by adding $\mathbf{g}$ alone, making it a supervised and scale-aware criterion for greedy feature selection.

3) Selection and refitting: After scoring all M candidates, we select the top-k features with the largest values of $\Delta(\mathbf{g})$ and append them to the current hidden-feature matrix,

$$\mathbf{G}_{t+1} = [\,\mathbf{G}_t \;\; \mathbf{G}_t^{\mathrm{sel}}\,], \qquad \mathbf{A}_{t+1} = [\,\mathbf{U} \;\; \mathbf{G}_{t+1}\,].$$

The linear readout is then refitted exactly on the updated representation by solving (6), yielding $\mathbf{\Omega}_{t+1}$. This stagewise process of candidate generation, residual-based scoring, selection, and refitting is repeated until the hidden-unit budget h is reached or a validation criterion terminates training.

The aforementioned steps complete the description of the proposed residual-guided stagewise construction. For clarity and completeness, the entire procedure is summarized in Algorithm 1.

Algorithm 1 Residual-Guided Stagewise Algorithm

Require: Data $\mathbf{U} \in \mathbb{R}^{n\times p}$, targets $\mathbf{T} \in \mathbb{R}^{n\times q}$; hidden nodes budget h; pool size M; block size k; regularization $\lambda > 0$; sampling distribution $\mathcal{P}$; activation $\phi(\cdot)$.
Ensure: Hidden features $\mathbf{G} \in \mathbb{R}^{n\times h'}$ with $h' \le h$ and readout $\mathbf{\Omega} \in \mathbb{R}^{(p+h')\times q}$.

Initialize $\mathbf{G}_0 \leftarrow [\,]$, $h_0 \leftarrow 0$, $t \leftarrow 0$.
while $h_t < h$ **and** stopping criterion not met **do**
  $\mathbf{A}_t \leftarrow [\,\mathbf{U} \;\; \mathbf{G}_t\,]$.
  $\mathbf{\Omega}_t \leftarrow \mathbf{\Omega}^\star(\mathbf{A}_t, \lambda)$ ▷ closed-form ridge solution
  $\mathbf{R}_t \leftarrow \mathbf{A}_t\mathbf{\Omega}_t - \mathbf{T}$.
  for $j = 1, \ldots, M$ **do** ▷ temporary candidate pool
    Sample $(\mathbf{w}^{(j)}, \beta^{(j)}) \sim \mathcal{P}$.
    $\mathbf{g}^{(j)} \leftarrow \phi(\mathbf{U}\mathbf{w}^{(j)} + \mathbf{e}_n\beta^{(j)}) \in \mathbb{R}^n$.
    $s^{(j)} \leftarrow \dfrac{\|\mathbf{g}^{(j)\top}\mathbf{R}_t\|_F^2}{\|\mathbf{g}^{(j)}\|_2^2 + \lambda}$ ▷ score $\Delta(\mathbf{g}^{(j)})$
  end for
  Let $\mathcal{I}$ be indices of the top-k scores $\{s^{(j)}\}_{j=1}^M$ (or fewer if $h_t + k > h$).
  $\mathbf{G}_t^{\mathrm{sel}} \leftarrow [\,\mathbf{g}^{(j)}\,]_{j\in\mathcal{I}}$.
  $\mathbf{G}_{t+1} \leftarrow [\,\mathbf{G}_t \;\; \mathbf{G}_t^{\mathrm{sel}}\,]$, $h_{t+1} \leftarrow h_t + |\mathcal{I}|$.
  $t \leftarrow t + 1$.
end while
$\mathbf{A} \leftarrow [\,\mathbf{U} \;\; \mathbf{G}_t\,]$, $\mathbf{\Omega} \leftarrow \mathbf{\Omega}^\star(\mathbf{A}, \lambda)$.
return $\mathbf{G}_t, \mathbf{\Omega}$.

C. Monotonicity Guarantee

In this subsection, we establish a key theoretical property of the proposed residual-guided stagewise method. Specifically, we show that the ridge-regularized training objective is non-increasing across stages, ensuring that the algorithm makes consistent progress as additional hidden features are incorporated.

Lemma IV.1. *Let $\mathcal{J}(\mathbf{G})$ be defined as in* (7)*. At each stage of the residual-guided procedure, the following inequality holds:*

$$\mathcal{J}(\mathbf{G}_{t+1}) \le \mathcal{J}(\mathbf{G}_t).$$

Proof. Let $\mathbf{A}_t = [\,\mathbf{U} \;\; \mathbf{G}_t\,]$ and $\mathbf{A}_{t+1} = [\,\mathbf{U} \;\; \mathbf{G}_{t+1}\,]$, where $\mathbf{G}_{t+1} = [\,\mathbf{G}_t \;\; \mathbf{G}_t^{\mathrm{sel}}\,]$. By construction, $\mathbf{A}_t$ is a column-submatrix of $\mathbf{A}_{t+1}$.

Let $\mathbf{\Omega}_t^\star$ denote an optimal solution attaining $\mathcal{J}(\mathbf{G}_t)$ in (7). We construct a coefficient matrix $\widetilde{\mathbf{\Omega}}$ for $\mathbf{A}_{t+1}$ by assigning $\mathbf{\Omega}_t^\star$ to the columns corresponding to $\mathbf{A}_t$ and setting the coefficients of the newly added columns to zero. Then,

$$\mathbf{A}_{t+1}\widetilde{\mathbf{\Omega}} = \mathbf{A}_t\mathbf{\Omega}_t^\star, \qquad \|\widetilde{\mathbf{\Omega}}\|_F = \|\mathbf{\Omega}_t^\star\|_F.$$

Consequently, $\widetilde{\mathbf{\Omega}}$ achieves the same objective value on $\mathbf{A}_{t+1}$ as $\mathbf{\Omega}_t^\star$ does on $\mathbf{A}_t$. Since $\mathcal{J}(\mathbf{G}_{t+1})$ is defined as the minimum of the ridge-regularized objective over all coefficient matrices associated with $\mathbf{A}_{t+1}$, it cannot exceed this value. Therefore,

$$\mathcal{J}(\mathbf{G}_{t+1}) \le \mathcal{J}(\mathbf{G}_t),$$

which completes the proof. □

D. Computational Complexity

At each stage, generating a pool of M candidate hidden features requires $O(npM)$ operations, corresponding to the evaluation of random affine mappings and activation functions. Scoring the candidates incurs an additional cost of $O(nMq)$ to compute the inner products $\mathbf{G}_t^{\text{pool}\top}\mathbf{R}_t$, along with $O(nM)$ operations to evaluate the associated ℓ_2 norms. Refitting the linear readout after stage t using a naive approach costs $O(n(p+h_t)^2)$ when using the primal formulation, or $O(n^2(p+h_t))$ when using the dual formulation. However, since only a small block of k hidden features is added at each stage, the readout update can be performed incrementally by exploiting the resulting low-rank modification of the augmented matrix. This reduces the refitting cost to $O\big(nk(p+h_t)\big)+O\big(k^2(p+h_t)\big)$. Let T denote the total number of stages. Since $T \approx \lceil h/k \rceil$, the overall computational cost of the proposed method grows approximately linearly with the hidden-unit budget h for fixed pool size M and block size k.

V. Experimental Results

This section evaluates the proposed residual-guided framework by integrating it into three RaNN architectures, namely RVFL, ELM, and BLS, yielding the corresponding residual-guided models RG-RVFL, RG-ELM, and RG-BLS. The proposed models are compared with their baseline counterparts to assess the effectiveness of residual-guided construction. Performance is evaluated using classification accuracy, average rank, and standard deviation. Experiments are conducted on a total of 71 benchmark datasets, comprising both binary and multiclass classification problems, downloaded from the UCI repository [20]. The detailed experimental setup along with the hyperparametr configuration is provided in Section S.I of the supplementary file.

A. Performance Analysis

Table I summarizes the average classification accuracy, standard deviation, and average rank of the baseline and residual-guided models over 32 binary and 39 multiclass datasets. Overall, the results consistently demonstrate the effectiveness of the proposed residual-guided learning framework across all three randomized architectures, namely RVFL, ELM, and BLS.

For binary classification tasks, the residual-guided variants achieve clear and consistent accuracy gains over their corresponding baselines. In particular, RG-RVFL, RG-ELM, and RG-BLS improve the average accuracy from 80.92% to 82.92%, from 80.82% to 82.76%, and from 81.65% to 82.78%, respectively. Similar trends are observed for multiclass datasets, where the proposed models yield notable improvements, with average accuracy increasing from 73.33% to 75.94% for RVFL, from 73.34% to 75.35% for ELM, and from 74.96% to 76.20% for BLS. These results indicate that residual-guided construction consistently enhances the representational capacity of the underlying randomized learners. Beyond accuracy, the proposed framework also exhibits improved stability. As reflected by the lower average standard deviation values, all residual-guided models demonstrate reduced performance variability across datasets. For binary problems, the standard deviation decreases from 8.91 to 7.88 for RVFL, from 8.36 to 7.76 for ELM, and from 6.24 to 5.75 for BLS. A similar reduction is observed in multiclass settings, confirming that the residual-guided strategy not only improves mean performance but also yields more reliable and robust predictions across diverse data distributions. The superiority of the proposed approach is further reinforced by the average rank analysis. Across both binary and multiclass tasks, the residual-guided variants consistently achieve near-optimal average ranks, all close to one, indicating dominant performance among the compared methods. In contrast, the baseline models exhibit noticeably higher ranks, reflecting inferior relative performance. This consistent ranking advantage highlights the effectiveness of the proposed residual-guided mechanism independent of the specific randomized architecture employed.

TABLE I: Average accuracy (%), average standard deviation, and average rank of baseline and residual-guided models on 32 binary and 39 multiclass datasets.

Task	Metric	Method Comparison					
		RVFL	RG-RVFL†	ELM	RG-ELM†	BLS	RG-BLS†
Binary (32)	Accuracy ↑	80.92	**82.92**	80.82	**82.76**	81.65	**82.78**
	Std. Dev. ↓	8.91	**7.88**	8.36	**7.76**	6.24	**5.75**
	Rank ↓	1.97	**1.03**	1.88	**1.13**	1.91	**1.09**
Multiclass (39)	Accuracy ↑	73.33	**75.94**	73.34	**75.35**	74.96	**76.20**
	Std. Dev. ↓	10.43	**9.11**	10.12	**8.99**	7.98	**7.17**
	Rank ↓	1.96	**1.04**	1.94	**1.06**	1.86	**1.14**

† denotes the proposed counterpart.

Collectively, these results demonstrate that incorporating residual-guided learning into RVFL, ELM, and BLS models leads to systematic improvements in accuracy, stability, and overall ranking performance. Importantly, the gains are observed across both binary and multiclass classification tasks, underscoring the generality of the proposed framework. Detailed dataset-wise results for all evaluated models are provided in Tables S.I and S.II of the supplementary material.

To empirically validate the monotonicity property established in Section IV-C, we analyze the stagewise behavior of the proposed residual-guided framework. The detailed analysis is provided in Section S.II of the supplementary.

VI. Conclusions

In this work, we introduced a residual-guided learning framework for randomized neural networks that addresses a fundamental limitation of existing RaNN architectures, namely the reliance on a single, task-uninformed draw of hidden features. By replacing one-shot random feature construction with a supervised, stagewise selection mechanism guided by the current residual, the proposed framework enables adaptive and efficient hidden-layer expansion while preserving the hallmark advantages of randomized learning, including closed-form training and computational efficiency. We establish a

monotonicity guarantee showing that the ridge-regularized training objective is non-increasing as residual-guided hidden features are progressively added. This property provides a foundation for progressive model construction and stands in contrast to conventional randomized networks, where intermediate representations offer no such guarantees. Extensive experimental results on 71 benchmark datasets from the UCI repository, spanning both binary and multiclass classification tasks, confirm the practical effectiveness of the proposed framework. The results highlight the generality of the framework and its ability to enhance diverse randomized models without altering their core training paradigms. In future, one can extend the proposed framework for regression, multilabel, and semi-supervised tasks.

REFERENCES

[1] Y.-H. Pao, G.-H. Park, and D. J. Sobajic, "Learning and generalization characteristics of the random vector functional-link net," *Neurocomputing*, vol. 6, no. 2, pp. 163–180, 1994.

[2] W. Cao *et al.*, "A review on neural networks with random weights," *Neurocomputing*, vol. 275, pp. 278–287, 2018.

[3] P. N. Suganthan and R. Katuwal, "On the origins of randomization-based feedforward neural networks," *Applied Soft Computing*, vol. 105, p. 107239, 2021.

[4] B. Igelnik and Y.-H. Pao, "Stochastic choice of basis functions in adaptive function approximation and the functional-link net," *IEEE Transactions on Neural Networks*, vol. 6, no. 6, pp. 1320–1329, 1995.

[5] D. Needell *et al.*, "Random vector functional link networks for function approximation on manifolds," *Frontiers in Applied Mathematics and Statistics*, vol. 10, p. 1284706, 2024.

[6] A. K. Malik *et al.*, "Random vector functional link network: recent developments, applications, and future directions," *Applied Soft Computing*, vol. 143, p. 110377, 2023.

[7] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.

[8] C. P. Chen and Z. Liu, "Broad learning system: An effective and efficient incremental learning system without the need for deep architecture," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 1, pp. 10–24, 2017.

[9] ——, "Broad learning system: A new learning paradigm and system without going deep," in *2017 32nd Youth Academic Annual Conference of Chinese Association of Automation (YAC)*. IEEE, 2017, pp. 1271–1276.

[10] X. Gong *et al.*, "Research review for broad learning system: Algorithms, theory, and applications," *IEEE Transactions on Cybernetics*, vol. 52, no. 9, pp. 8922–8950, 2022.

[11] S. Feng and C. P. Chen, "Fuzzy broad learning system: A novel neuro-fuzzy model for regression and classification," *IEEE transactions on cybernetics*, vol. 50, no. 2, pp. 414–424, 2018.

[12] A. K. Malik *et al.*, "Alzheimer's disease diagnosis via intuitionistic fuzzy random vector functional link network," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 4, pp. 4754–4765, 2024.

[13] J.-W. Jin and C. P. Chen, "Regularized robust broad learning system for uncertain data modeling," *Neurocomputing*, vol. 322, pp. 58–69, 2018.

[14] M. Akhtar *et al.*, "Towards robust and inversion-free randomized neural networks: The XG-RVFL framework," *Pattern Recognition*, vol. 172, 4 2026.

[15] Q. Shi *et al.*, "Random vector functional link neural network based ensemble deep learning," *Pattern Recognition*, vol. 117, 9 2021.

[16] M. Sajid, M. Tanveer, and P. N. Suganthan, "Ensemble deep random vector functional link neural network based on fuzzy inference system," *IEEE Transactions on Fuzzy Systems*, pp. 1–12, 2024. [Online]. Available: 10.1109/TFUZZ.2024.3411614

[17] C. Liu *et al.*, "Domain-incremental learning without forgetting based on random vector functional link networks," *Pattern Recognition*, vol. 151, p. 110430, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S003132032400181X

[18] L. Zhong *et al.*, "Robust incremental broad learning system for data streams of uncertain scale," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 4, pp. 7580–7593, 2025.

[19] M. Akhtar, M. Tanveer, and M. Arshad, "CAWI: Copula-aligned weight initialization for randomized neural networks," in *The 29th International Conference on Artificial Intelligence and Statistics*, 2026. [Online]. Available: https://openreview.net/forum?id=fwqydIjLn6

[20] D. Dua and C. Graff, "UCI machine learning repository," *URL http://archive. ics. uci. edu/ml*, vol. 7, no. 1, p. 62, 2017.