RESHAPING RESERVOIRS: HEBBIAN PLASTICITY FOR IMPROVED DATA SEPARABILITY

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper introduces Hebbian Architecture Generation (HAG), a method grounded in Hebbian plasticity principles, designed to optimize the structure of Reservoir Computing networks. HAG adapts the synaptic weights in Recurrent Neural Networks by dynamically forming connections between neurons that exhibit high Pearson correlation. Unlike conventional reservoir computing models that rely on static, randomly initialized connectivity matrices, HAG tailors the reservoir architecture to specific tasks by autonomously optimizing network properties such as signal decorrelation and singular value spread. This task-specific adaptability enhances the linear separability of input data, as supported by Cover's theorem, which posits that increasing the dimensionality of the feature space improves pattern recognition. Experimental results show that HAG outperforms traditional Echo State Networks across various predictive modeling and pattern recognition benchmarks. By aligning with biological principles of structural plasticity, HAG addresses limitations of static reservoir architectures, offering a biologically plausible and highly adaptable alternative for improved performance in dynamic learning environments.

1 INTRODUCTION

In this paper, we introduce Hebbian Architecture Generation (HAG), a novel approach that dynamically adjusts the synaptic weights in Recurrent Neural Networks (RNNs) to improve the quality of their representations. HAG is founded on Hebbian theory (Attneave et al., 1950), the principle that synaptic connections between co-activating neurons strengthen over time, encapsulated by the maxim "neurons that fire together wire together." Leveraging this idea of topology informed by correlations, HAG dynamically constructs reservoir connectivity to better suit the task at hand, addressing several core challenges in Reservoir Computing (RC).

RC, particularly through Echo State Networks (ESNs) (Jaeger, 2001), provides a framework for transforming input signals into high-dimensional dynamic states, as dictated by Cover's theorem. Cover's theorem posits that increasing the dimensionality of a feature space enhances the likelihood that complex data patterns become linearly separable, which is crucial for efficient learning in RC. However, traditional ESNs, as highlighted by Jaeger (2005), face notable limitations in reservoir suitability, unsupervised adaptation, or alignment with biological principles.

HAG addresses these limitations in several ways:

1. **Task-specific:** beginning with a blank connectivity matrix, HAG forms dynamic synaptic connections, which inherently tailors the reservoir to the demands of specific tasks, moving beyond the random, static connectivity found in traditional ESNs.
2. **Unsupervised adaptation:** It autonomously optimizes network properties such as singular value spread and signal decorrelation by leveraging high Pearson correlations between neurons, thereby enhancing the linearity of the feature space.
3. **Performance predictors:** By focusing on measurable properties such as signal decorrelation and feature space expansion, we provide specific criteria for assessing and enhancing reservoir performance. This approach moves beyond the tautological notion that a reservoir is suitable if it yields accurate models.

4. **Biological insights:** Reflecting the principles of Hebbian and structural plasticity, HAG mimics the adaptability of biological neural networks by forming new connections and reorganizing correlated input features to optimize feature extraction.

Our experimental results demonstrate that HAG not only addresses those key challenges, but also improves performance across multiple benchmarks compared to traditional RC approaches.

This paper first presents the necessary background (section 2) for echo state networks. We then introduce the HAG algorithm, the motivation behind this rule, and the main elements of our experiments in Section 3. Section 4 then covers the significant performance improvements on several benchmarks for two versions of the HAG algorithm compared to traditional RC. Finally, in section 5 we show that our algorithm projects input data into a higher-dimensional space which leads to better performance according to Cover's theorem.

2 BACKGROUND

2.1 ECHO STATE NETWORKS

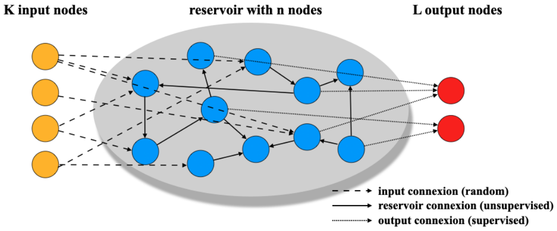


Figure 1: Schematic architecture of ESNs

In ESNs, the challenge of capturing nonlinear relationships in data is addressed by transforming the time series input into a high-dimensional space using a dynamical system (the reservoir) which in the case of ESNs is an artificial neural network of recurrently connected nodes. In this new feature space, the complex, nonlinear relationships may become linear, enabling the use of simple linear models, such as ridge regression, to accurately estimate the target variable. Notably, only the linear readout layer is trained, while the reservoir remains fixed, reducing computational complexity and enhancing training efficiency.

An illustration of the ESN framework (Lukoševičius & Jaeger, 2009) (Jaeger, 2001) is given in Fig. 1. It normally consists of a randomly initiated RNN of n neurons (the *reservoir*), and a trained *readout layer* that creates the outputs as a linear combination of the reservoir neuron states.

The evolution of the vector states $\mathbf{x}$ of the n neurons is determined by the interactions between the connection matrix $\mathbf{W}$, the input matrix $\mathbf{W_{in}}$, the input $u[t]$ and the activation function σ that we fix as the hyperbolic tangent function, combined in the following equation:

$$\mathbf{x}[t+1] = \sigma(\mathbf{W} \times \mathbf{x}[t] + \mathbf{W_{in}} \times u[t] + \mathbf{b}) \tag{1}$$

The purpose of the bias term $\mathbf{b}$ is to enrich the dynamics of the network, but is kept small through bias scaling to avoid dominating the system. $\mathbf{b}$ is a $1 \times n$ vector and dimension of the matrices are $n \times n$ for $\mathbf{W}$ and $1 \times n$ for $\mathbf{W_{in}}$. In ESNs, these weights are typically randomly initialized from a chosen distribution, and modulated by input scaling (s_{in}) and bias scaling (s_{b}) that are two hyperparameters of the systems. However, as we explain below, mean-HAG uses a different weight initialization scheme where these weights evolve during operation according to an unsupervised rule.

Once $\mathbf{W}$, $\mathbf{W_{in}}$ and $\mathbf{b}$ are fixed, the desired output is obtained through the following equation :

$$y[t+1] = \mathbf{W_{out}} \times \mathbf{x}[t+1] + \mathbf{b_{out}} \tag{2}$$

where $\mathbf{W_{out}}$, $\mathbf{b_{out}}$ are learned using ridge regression with regularization parameter λ to prevent overfitting by controlling the size of the output such that the generated train output, $y[t]$, optimally approximates a desired target output, $y^{target}[t]$ (Lukoševičius & Jaeger, 2009).

3 METHODS

While Lukoševičius & Jaeger (2009) gives an overview of other unsupervised methods used to improve RC, a detailed overview of the effect of biological rules on the dynamics of RC is given in Morales et al. (2021). Additionally, the self-organizing recurrent network (SORN) introduced by Lazar (2009) focuses on spiking neurons and leverages multiple forms of plasticity to adapt a form of spiking neural network.

Our approach differs from previously explored techniques on three points. First, our algorithm takes inspiration from structural plasticity and starts from a blank connectivity matrix, and is then able to create connections between neurons that are not connected. Second, our algorithm is based on the Pearson correlation, which allows us to recombine frequently correlated input features, transforming the data into a new feature space with reduced correlation. In this transformed space, the input features are recombined into a higher-dimensional space where patterns become as linearly independent as possible. Third, our reservoir uses solely excitatory connections, meaning that every weight of the connection matrix is positive.

3.1 HAG ALGORITHM

We start from a blank echo state network with no connections except from the input connectivity. The HAG algorithm dynamically adjusts synaptic weights w_{ij} based on either variance or average of the activity of the reservoir's neurons. Those two types of homeostatic mechanisms lead to two corresponding algorithms:

1. **Mean Homeostasis Function (mean-HAG)**: Adjusts synaptic weights based on deviations from a target mean activity rate.
2. **Variance Homeostasis Function (variance-HAG)**: Adjusts synaptic weights by comparing the standard deviation of neuron states to a target standard deviation.

A more detailed explanation on the HAG algorithm is provided in Appendix B with pseudo-code in B.1.

Every T time steps, we calculate for each neuron a growth indicator:

$$\Delta z_i = \frac{1}{\beta}(s_i - \rho) \tag{3}$$

where s_i represents either the average i-th neuron's activity $\langle x_i \rangle_T$ over period T for mean-HAG, or the i-th neuron's standard deviation $\sigma_{x_i,T}$ over period T for variance-HAG. We denote ρ_r as the activity target value and β_r as the rate spread (for mean-HAG), and ρ_v as the variance target and β_v as the variance spread (for variance-HAG).

If $\Delta z_i < -1$, the neuron needs to increase its activity. In this case, one connection weight is increased by δw. The creation of new connections is restricted to neurons that have been identified as requiring additional connections. To choose which connection to increase, for every neuron that has not yet achieved homeostasis, we compute pairwise Pearson correlation coefficients (Pearson, 1895) with every other neuron that is also not at homeostasis. We establish a connection with the highest correlated neuron. Detailed description can be found in Appendix B.2.

If $\Delta z_i > 1$, the neuron needs to decrease its activity. In this case, one connection weight is decreased by δw. Unlike the creation of new connections, the pruning of connections is performed randomly, independently of the state of the neuron's partners, regardless of whether they also need to decrease their activity or not.

The network is said to be at homeostasis if, for each neuron i, the absolute value of Δz_i is less than 1 (i.e. s_i is between $\rho - \beta$ and $\rho + \beta$). At this homeostasis the network maintains a desired level of variance or average in neuronal activity as seen in Figure 2.

We impose a limit on the degree of the node, γ. By design, the RNN will only have positive connections, which ensures predictable increases in neuronal activity. This configuration restricts the network to excitatory connections, which might limit the computational properties of the ESN.

3.1.1 ENSURING CONVERGENCE

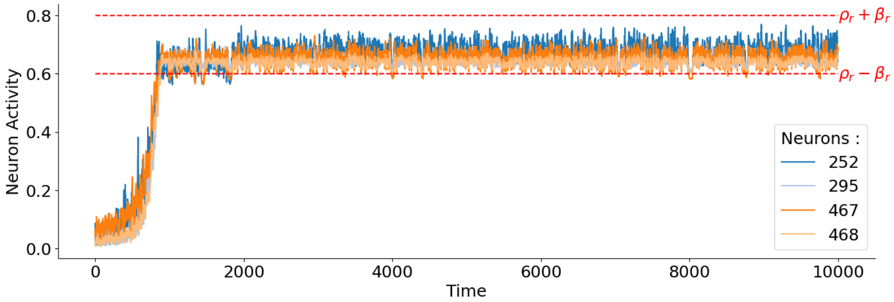


Figure 2: Different neurons' activities during training with mean-HAG algorithm. As new connections are added every T period the average of neuron states $\langle x_i \rangle_T$ clearly converges within the homeostatic range of activity

The mean-HAG algorithm dynamically adjusts neural connectivity based on deviations from target mean activity rates. This ensures convergence during the learning phases as seen in Figure 2.

In contrast, the variance-HAG algorithm aims to increase variability, which could inadvertently amplify the overall signal strength and lead to potential saturation. To counteract this, we employ a homeostatic plasticity mechanism, when a neuron's activity level exceeds a saturation threshold θ_{sat}, the algorithm scales the weights by η_{sat}. This adjustment is crucial for maintaining network balance, avoiding disruptive saturation, and ensuring the smooth convergence of the algorithm.

3.1.2 MOTIVATION

Reservoir Computing (RC) leverages the concept of transforming input data into a higher-dimensional space where complex patterns become more linearly separable. This idea is rooted in Cover's theorem (Cover, 1965), which suggests that projecting data into a high-dimensional non-linear space increases the likelihood of linear separability. Traditional Echo State Networks (ESNs) achieve this transformation using randomly instantiated reservoirs. However, random reservoirs may not optimally exploit the structure of the input data, potentially limiting their effectiveness in separating complex patterns.

Our Hebbian Architecture Generation (HAG) algorithm is motivated by the need to enhance this transformation process. By dynamically adjusting the reservoir's connectivity based on neuron activations, HAG aims to design a network structure that recombines the inputs into a larger manifold compared to a randomly instantiated reservoir. This recombination effectively increases the dimensionality and richness of the feature space, improving the reservoir's capacity to represent and separate complex input patterns.

The HAG algorithm leverages Hebbian principles to form connections between highly correlated neurons. By doing so, it tailors the reservoir to the specific characteristics of the input data, enhancing the linear separability of the transformed data. This targeted adaptation aligns with the insights provided by Cover's theorem but goes beyond by actively constructing a reservoir that better suits the task at hand. Such an approach draws inspiration from biological systems, where plasticity mechanisms dynamically shape neural networks' states in response to input. Similar mechanisms have been shown to drive the transitions between dynamic states in biological networks in response to changes in input strength, mediated by homeostatic plasticity (Zierenberg et al., 2018). These findings underscore the importance of adaptive connectivity in optimizing dynamic representations.

3.2 DATASETS

ESNs have employed a diversity of benchmarks and datasets, as extensively documented in Sun et al. (2020). Our study employs a diverse array of datasets to assess the capabilities of our algorithm.

We utilized `ReservoirPy` (Trouvain et al., 2020), a library updated with contemporary advancements, featuring a modular architecture for assembling ESNs and a suite of standard algorithms for training the readout layer.

3.2.1 TASKS

The training of the ESN system occurs in two phases. Initially, for classification tasks such as speech recognition, the reservoir processes input signals into high-dimensional representations, capturing the dynamic state at the end of each input sequence. This state, representing the reservoir's response to the input, serves as the feature vector for training the readout layer using ridge regression to determine the optimal output weights W_{out} .

For prediction tasks aimed at forecasting future values from past inputs, we adopt a sequence-to-sequence approach. The reservoir updates its state at each time step, using the entire sequence of states to train the readout layer. Here, W_{out} is optimized to minimize the difference between the predicted outputs and the actual targets at each time step.

3.2.2 CLASSIFICATION DATASETS

CatsDogs The **CatsDogs** dataset is the auditory counterpart to the classic image classification task, containing WAV audio files—164 for cats (1,323 seconds) and 113 for dogs (598 seconds)—recorded at 16 kHz.

SpokenArabicDigits The **Spoken Arabic Digits** dataset contains recordings of 88 individuals pronouncing Arabic digits 0–9, with ten pronunciations per digit per speaker. It is commonly used for testing speech recognition algorithms due to the phonetic diversity of Arabic numerals. (Mouldi Bedda, 2008)

Japanese Vowels The **Japanese Vowels** dataset includes recordings of nine male speakers pronouncing sequences of Japanese vowels. It is frequently utilized in research on linguistic characteristics and speaker identification technologies. (Mineichi Kudo, 1999)

FSDD The **Free Spoken Digit Dataset** (FSDD) is an open collection of English audio recordings of spoken digits 0–9 by multiple speakers. Designed for experimenting with speech processing techniques like classification and clustering, it provides a straightforward entry point into digital speech processing. (Jackson et al., 2018)

SPEECHCOMMANDS The **SPEECHCOMMANDS** dataset comprises over 105,000 audio files of short commands like "Yes," "No," "Up," and "Down," spoken by various speakers. Widely used for training and benchmarking models in voice user interfaces. (Warden, 2018)

3.2.3 PREDICTION DATASETS

MackeyGlass Derived from a differential equation, the **MackeyGlass** dataset is noted for its use in modeling nonlinear dynamics and chaos, making it a challenging dataset for time series prediction models. (Mackey & Glass, 1977)

Lorenz The **Lorenz** dataset is based on the Lorenz attractor, a set of chaotic differential equations used extensively in predicting nonlinear system behaviors and atmospheric studies. (Lorenz, 1963)

Sunspot (SILSO) The **Sunspot** dataset from SILSO includes smoothed monthly mean sunspot numbers from 1749 to 2020, reflecting solar activity and serving as a proxy for the Sun's magnetic field strength. Its complexity makes it a significant test case for forecasting models in time series analysis and solar studies. (SILSO World Data Center, 2024)

3.3 PREPROCESSING

We adopt distinct preprocessing methods tailored to classification and prediction tasks, each leveraging the properties of audio signals to prepare data for analysis and model training.

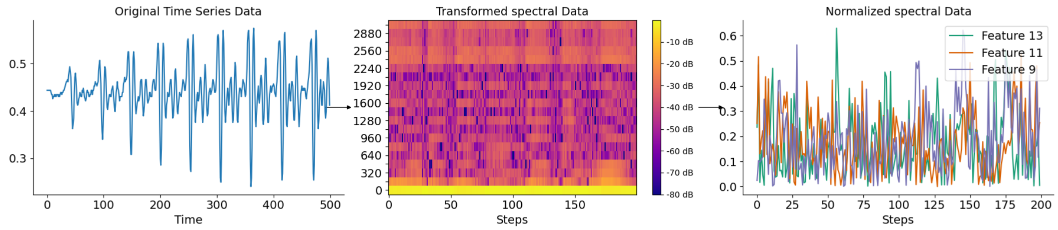


Figure 3: Preprocessing pipeline for classification datasets (here for FSDD) from raw time series (left) to spectrogram (middle) and normalized features (right).

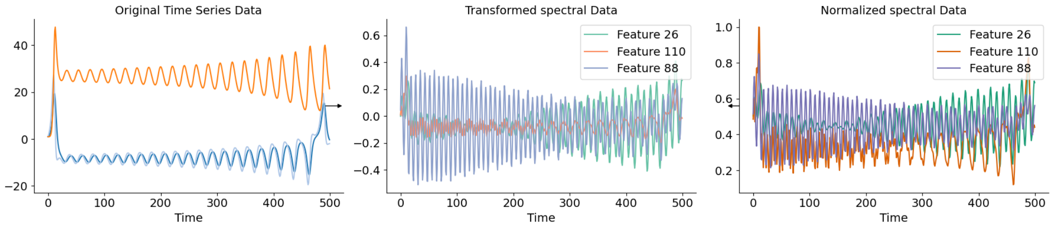


Figure 4: Preprocessing pipeline for prediction datasets (here for Lorenz) from raw time series (left) to features after filtering around main frequencies (middle) and normalized features (right). Here the number of time steps in the time series are preserved to perform prediction tasks

In the classification tasks, we preprocess the audio signals by converting them into spectral representations using *Mel-Frequency Cepstral Coefficients* (MFCCs), computed with the `librosa` library (McFee et al., 2015). MFCCs are widely used in speech and audio processing to capture the timbral aspects of sound by modeling the human auditory system's response. They provide a compact representation of the spectral properties of audio signals, making them suitable for classification tasks. This process is illustrated in Figure 3.

Conversely, for prediction tasks, the preprocessing utilizes band-pass filters to isolate specific frequency bands from the signal. This approach starts with the identification of peak frequencies. Frequency bands are then defined around these peaks by calculating the bandwidth as half the distance to adjacent peak frequencies. This method effectively segments the time series into parts that contain relevant information for predictive modeling. The filtered signals obtained from this process constitute the feature set used for forecasting future events or states from past time series. Compared to the method for classification tasks, this method preserves the number of time steps in the time series, which is essential to perform prediction tasks. This process is illustrated in Figure 4.

For each dataset, the network size is set to just above 500 units by duplicating the multivariate input a specific number of times. For example, the SpokenArabicDigits dataset uses a network size of 507, achieved by duplicating its 13-dimensional spectrogram 39 times. Similarly, the Japanese Vowels dataset has a network size of 504, with a 12-dimensional spectrogram duplicated 42 times.

3.4 HYPERPARAMETER SEARCH

To ensure robust model evaluation, we employed cross-validation strategies appropriate for each task type. For classification tasks without predefined groups, we used Stratified K-Fold cross-validation with shuffling to maintain class distribution across folds. When group-based classification was necessary, Stratified Group K-Fold cross-validation was applied to preserve both class distribution and group integrity, preventing data leakage. For time series prediction tasks, Time Series Split cross-validation was utilized to respect temporal ordering and prevent future data leakage.

Hyperparameter tuning was performed using `optuna` (Akiba et al., 2019), leveraging the Tree-structured Parzen Estimator (TPE) sampler over 400 trials per dataset and algorithm variant. The TPE sampler efficiently explores the hyperparameter space by focusing on promising regions, making it suitable for our optimization tasks (Bergstra et al., 2011).

Table 1: Mean cross-validation classification accuracy averaged over 3-folds for 400 trials.

	E-ESN	**ESN**	**IP-ESN**	**mean-HAG**	**variance-HAG**
CatsDogs	61.6%	61.5%	**83.0%**	71.4%	76.8%
Japanese Vowels	95.2%	95.9%	**98.5%**	96.3%	94.4%
SpokenArabicDigits	56.1%	76.6%	71.4%	77.0%	**82.8%**
FSDD	28.0%	20.0%	**72.9%**	51.4%	44.6%
SPEECHCOMMANDS	5.8%	5.91%	7.1%	32.6%	**34.0%**

Table 2: Cross-validation NRMSE averaged over 3-folds for 400 trials

	E-ESN	**ESN**	**IP-ESN**	**mean-HAG**	**variance-HAG**
MackeyGlass	0.070	0.068	**0.055**	0.066	0.063
Lorenz	0.20	0.14	0.147	0.21	**0.14**
Sunspot	**0.0155**	0.0202	0.0223	0.0195	0.0198

The hyperparameters optimized included input scaling, bias scaling, ridge regression coefficient, and algorithm-specific parameters such as rate target and variance target. Table 5 summarizes the hyperparameter ranges.

4 RESULTS

We evaluated the performance of the HAG algorithms (mean-HAG and variance-HAG) against baseline models, including traditional ESNs, ESNs with only positive weights (E-ESNs), and Intrinsic Plasticity ESNs (IP-ESNs) as described in Schrauwen et al. (2008). The results highlight the robust improvements offered by the HAG algorithms, particularly in classification tasks. Details on the hyperparameter search results can be found in C.2.

4.1 CROSS VALIDATION PERFORMANCES

Tables 1 and 2 present the mean cross-validation classification accuracies and normalized root mean square errors (NRMSEs), respectively, averaged over 3-folds for 400 trials per model and dataset.

In classification tasks, the HAG algorithms consistently outperformed E-ESN and standard ESNs. For example, on the SPEECHCOMMANDS dataset, variance-HAG achieved a notable accuracy of 34.0%, significantly outperforming the ESN (5.9%) and E-ESN (5.8%). The IP-ESN model demonstrated strong performance on the CatsDogs, Japanese Vowels datasets and FSDD, achieving the highest accuracies of 83.0%, 98.5% and 72.9%, respectively. However, on datasets such as SpokenArabicDigits and SPEECHCOMMANDS, HAG algorithms surpassed IP-ESN, showing greater adaptability to diverse datasets.

In prediction tasks, variance-HAG achieved competitive performance, matching the standard ESN on the Lorenz dataset (NRMSE of 0.140). However, the IP-ESN model achieved the lowest NRMSE on the MackeyGlass dataset (0.055), highlighting its strength in this type of task. On the Sunspot dataset, E-ESN achieved the best NRMSE of 0.0155, but both HAG algorithms performed comparably, demonstrating their versatility across different prediction challenges.

4.2 TEST PERFORMANCES

Tables 3 and 4 summarize the models' test performance using the best hyperparameters identified during cross-validation.

On the test datasets, the HAG algorithms maintained strong performance. On the SpokenArabicDigits dataset, mean-HAG achieved a remarkable accuracy of 95.4%, significantly outperforming IP-ESN (57.8%). On the SPEECHCOMMANDS dataset, variance-HAG achieved the highest accuracy

Table 3: Test classification accuracy over 8 trials on the test dataset using the whole training dataset

	E-ESN	**ESN**	**IP-ESN**	**mean-HAG**	**variance-HAG**
CatsDogs	60.9%	60.4%	55.41%	68.8%	**69.1%**
Japanese Vowels	95.8%	96.7%	97.0%	**98.4%**	94.19%
SpokenArabicDigits	63.6%	74.8%	57.84%	**95.4%**	95.2%
FSDD	23.5%	25.3%	32.75%	**46.3%**	44.0%
SPEECHCOMMANDS	6.11%	6.94%	6.35%	17.3%	**27.4%**

Table 4: Mean NRMSE over 8 trials on the test dataset using the whole training dataset

	E-ESN	**ESN**	**IP-ESN**	**mean-HAG**	**variance-HAG**
MackeyGlass	**0.0518**	0.0533	0.0734	0.0667	0.0663
Lorenz	0.174	0.187	0.174	0.173	**0.159**
Sunspot	0.0695	0.0376	0.0510	**0.0342**	0.0544

of 27.4%, a substantial improvement over all baselines. IP-ESN excelled on the CatsDogs, Japanese Vowels and FSDD dataset during cross-validation but underperformed on the test set compared to the HAG algorithms, suggesting potential overfitting.

In prediction tasks, variance-HAG achieved the lowest NRMSE on the Lorenz dataset (0.159), while mean-HAG performed best on the Sunspot dataset (0.0342). IP-ESN anderperformed on the test datasets.

The results demonstrate that the HAG algorithms, particularly variance-HAG, consistently outperform baseline models, including IP-ESN, in classification tasks. Their ability to dynamically adjust synaptic weights based on Hebbian principles allows for improved representation of input data, enhancing accuracy and generalization. While IP-ESN achieved strong performance on some datasets, its inconsistency across tasks highlights the advantage of HAG's tailored reservoir dynamics, particularly on the most complex task of SPEECHCOMMANDS. In prediction tasks, HAG algorithms showed competitive performance but did not consistently outperform E-ESN or IP-ESN. This suggests that further optimization of the HAG approach is needed for time-series forecasting applications.

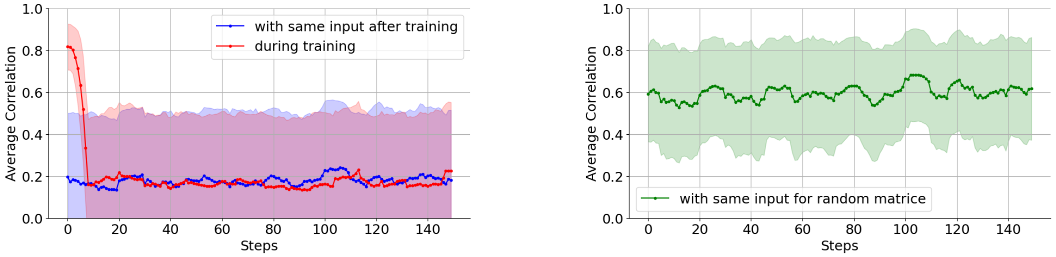
5 DISCUSSION

We hypothesize that our algorithm enhances the dynamics of ESNs by transforming redundant inputs into more informative representations. Leveraging Hebbian learning principles, our approach projects input data into a new feature space with reduced feature correlation, enriching the pool of features and increasing the likelihood of discovering linearly separable representations. This transformation is expected to improve the network's ability to perform complex tasks by creating a richer set of features for downstream processing.

Aligned with our hypothesis based on Cover's theorem, that increasing the dimensionality of neural states simplifies problem-solving, we evaluate the dynamic richness of our reservoir, which is specifically designed to increase neural state dimensionality. Following methodologies by Gallicchio & Micheli (2022), we assess dynamic richness using Pearson correlation and Cumulative Explained Variance (CEV). Detailed analysis, Pearson correlations and CEVs values for every dataset/function combination are presented in Appendix D

5.1 PEARSON CORRELATION

To elucidate the operational dynamics of our reservoir, we assess the correlation among neural states using Pearson correlation (Pearson, 1895), which measures the linear relationship between neuron activation states. This metric helps understand inter-neuronal connectivity and synchrony, directly impacting the network's ability to process complex data patterns. By computing Pearson correlation coefficients, we quantify initial levels of synchrony and track their evolution over time, offering insights into the network's dynamic restructuring in response to varying inputs. Detailed explanation about this calculation can be found in Appendix A.2



(a) during the formation of the connections for a set of good performing hyperparameters

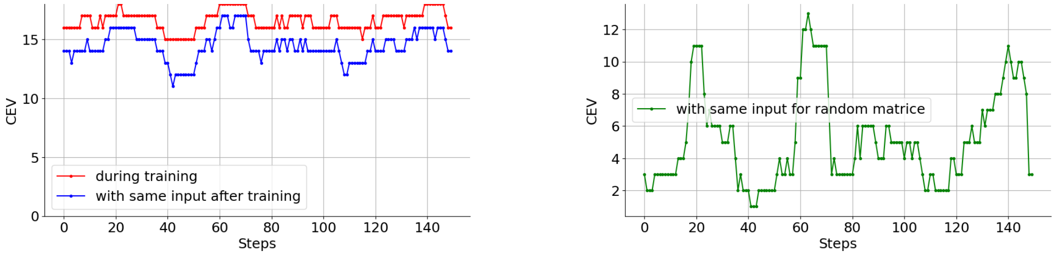
(b) for a random reservoir instantiated with best performing hyperparameters

Figure 5: Comparison of the evolution of the average Pearson correlation between neural states between the mean-HAG-designed and randomly instantiated reservoirs on the same input data (FSDD dataset). Lower Pearson correlation indicates better conditioning of the learning problem.

Figure 5 illustrates the correlation evolution within a high-performing HAG network compared to a randomly initialized reservoir. Initially, the correlation metric is near unity, indicating highly synchronized neuron states, a typical characteristic of freshly initialized networks where states are scaled input vectors with a bias term, resulting in almost identical states. Over time, a pronounced decline in mean correlation is evident, aligning with the HAG method's objective to refine the connectivity matrix by dynamically linking neurons based on correlation levels. This fosters a more diverse and functionally rich neural dynamic, enhancing the reservoir's computational capabilities. The comparison supports our hypothesis that HAG networks better represent complex input data than traditional ESNs.

It should be noted that similar patterns are observed across different graphs, highlighting the underlying coherence in input data characteristics. The same pattern of Pearson correlation can be observed in the inputs, that then influence neural state variations.

5.2 CUMULATIVE EXPLAINED VARIANCE



(a) during the formation of the connections for a set of good performing hyperparameters

(b) for a random reservoir instantiated with best performing hyperparameters

Figure 6: Comparison of evolution of moving cumulative explained variance (with $\theta = 0.9$) on the same input data (here with the SPEECHCOMMANDS datasets). Higher CEV indicates better conditioning of the learning problem.

To quantify the reservoir's dynamic behavior, we perform Principal Component Analysis (PCA) on the reservoir states. This approach allows us to analyze how the variance in the reservoir's state space is distributed among different principal components, providing insight into the complexity and richness of the dynamics.

We denote $\mathbf{H}$ as the data matrix formed by collecting the reservoir states, where each column corresponds to a state vector at a particular time step. Performing PCA on $\mathbf{H}$ yields singular values $\sigma_1, \sigma_2, \ldots, \sigma_n$, arranged in decreasing order. These singular values are directly related to the variance explained by each principal component.

The proportion of variance explained by the j-th principal component is calculated as:

$$R_j = \frac{\sigma_j^2}{\sum_{k=1}^{n} \sigma_k^2} \tag{4}$$

The cumulative explained variance up to the d-th principal component is then given by:

$$C_d = \sum_{j=1}^{d} R_j = \frac{\sum_{j=1}^{d} \sigma_j^2}{\sum_{k=1}^{n} \sigma_k^2} \tag{5}$$

This cumulative measure indicates the total proportion of variance captured by the first d principal components. To assess the effective dimensionality of the reservoir's state space, we determine the minimum number of principal components required to reach a predetermined threshold θ of cumulative explained variance:

$$D = \arg\min_{d} \left(C_d \geq \theta\right) \tag{6}$$

A higher value of D suggests that more principal components are needed to capture the same amount of variance, indicating a richer and more complex dynamic structure within the reservoir.

Figure 6 presents the cumulative explained variance curves for both the HAG-designed reservoir and the randomly instantiated reservoir on SPEECHCOMMANDS dataset. The HAG-designed reservoir exhibits a more gradual increase in cumulative explained variance, requiring more principal components to reach the threshold θ. This implies a higher effective dimensionality compared to the random reservoir, supporting our hypothesis that the HAG algorithm enhances the richness of the reservoir's dynamics.

6 CONCLUSION

We presented a new algorithm generating a connectivity matrix by identifying and connecting highly correlated neural nodes within a reservoir. This algorithm contrasts with traditional random matrix instantiation by creating a connectivity pattern that enhances decorrelation of reservoir states. By aligning the connectivity within the reservoir to the intrinsic correlations of the system, the proposed approach not only supports a new theoretical parallel with Hebbian plasticity but also demonstrates practical superiority over conventional methods.

The dynamic adaptability of HAG not only addresses the limitations of static reservoirs in Echo State Networks but also showcases the practical application of biologically-inspired algorithms in improving computational efficiency and task-specific performance. The autonomy in synaptic adjustment provided by HAG points towards architectures that are more aligned with biological neural processes, potentially leading to more robust and adaptable neural network models.

Our results underscore the utility of biologically-inspired design principles in computational models, emphasizing the potential of Hebbian learning rules not just as a theoretical construct but as a practical tool in machine learning. By enhancing the linear separability of input data through dynamic structural adaptations, HAG represents a significant step forward in the evolution of neural architecture search techniques, offering a flexible, efficient, and biologically relevant method to the challenges of modern AI research.

Future research should delve deeper into understanding the mechanisms that enable HAG to perform well with complex datasets. Additionally, exploring the scalability of HAG in larger and more intricate systems will further validate its effectiveness and contribute to the development of more intelligent and adaptable neural network models.

REFERENCES

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *The 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2623–2631, 2019.

Fred Attneave, M. B., and D. O. Hebb. The organization of behavior; a neuropsychological theory. *The American Journal of Psychology*, 63(4):633, October 1950. ISSN 0002-9556. doi: 10.2307/1418888.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11, pp. 2546–2554, Red Hook, NY, USA, 2011. Curran Associates Inc. ISBN 9781618395993.

Thomas M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, EC-14(3):326–334, June 1965. ISSN 0367-7508. doi: 10.1109/pgec.1965.264137.

Claudio Gallicchio and Alessio Micheli. Architectural richness in deep reservoir computing. *Neural Computing and Applications*, 35(34):24525–24542, January 2022. ISSN 1433-3058. doi: 10.1007/s00521-021-06760-7.

Zohar Jackson, César Souza, Jason Flaks, Yuxin Pan, Hereman Nicolas, and Adhish Thite. Jakobovski/free-spoken-digit-dataset: v1.0.8, 2018.

H. Jaeger. Short term memory in echo state networks. *GMD Forschungszentrum Informationstechnik*, 2001. doi: 10.24406/PUBLICA-FHG-291107.

H. Jaeger. Reservoir riddles: suggestions for echo state network research (extended abstract). In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, IJCNN-05. IEEE, 2005. doi: 10.1109/ijcnn.2005.1556090.

Andreea Lazar. SORN: a self-organizing recurrent neural network. *Frontiers in Computational Neuroscience*, 3, 2009. doi: 10.3389/neuro.10.023.2009.

Edward N. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20(2):130–141, March 1963. ISSN 1520-0469. doi: 10.1175/1520-0469(1963)020⟨0130:dnf⟩2.0.co;2.

Mantas Lukoševičius and Herbert Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009. doi: 10.1016/j.cosrev.2009.03.005.

Michael C. Mackey and Leon Glass. Oscillation and chaos in physiological control systems. *Science*, 197(4300):287–289, 1977. doi: 10.1126/science.267326.

Brian McFee, Colin Raffel, Dawen Liang, Daniel Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th Python in Science Conference*, SciPy, pp. 18–24. SciPy, 2015. doi: 10.25080/majora-7b98e3ed-003.

Jun Toyama Mineichi Kudo. Japanese vowels, 1999.

Guillermo B. Morales, Claudio R. Mirasso, and Miguel C. Soriano. Unveiling the role of plasticity rules in reservoir computing. *Neurocomputing*, 2021. doi: 10.1016/j.neucom.2020.05.127.

Nacereddine Hammami Mouldi Bedda. Spoken arabic digit, 2008.

Karl Pearson. Vii. note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58(347–352):240–242, December 1895. ISSN 2053-9126. doi: 10.1098/rspl.1895.0041.

Benjamin Schrauwen, Marion Wardermann, David Verstraeten, Jochen J. Steil, and Dirk Stroobandt. Improving reservoirs using intrinsic plasticity. *Neurocomputing*, 71(7–9):1159–1171, March 2008. ISSN 0925-2312. doi: 10.1016/j.neucom.2007.12.020.

SILSO World Data Center. The International Sunspot Number. *International Sunspot Number Monthly Bulletin and online catalogue*, 06 2024.

Chenxi Sun, Moxian Song, Shenda Hong, and Hongyan Li. A review of designs and applications of echo state networks. *CoRR*, 2020.

Nathan Trouvain, Luca Pedrelli, Thanh Trung Dinh, and Xavier Hinaut. ReservoirPy: An efficient and user-friendly library to design echo state networks. In *Artificial Neural Networks and Machine Learning – ICANN 2020*, pp. 494–505. Springer International Publishing, 2020. doi: 10.1007/978-3-030-61616-8_40. URL `https://doi.org/10.1007/978-3-030-61616-8_40`.

Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition, 2018.

Johannes Zierenberg, Jens Wilting, and Viola Priesemann. Homeostatic plasticity and external input shape neural network dynamics. *Physical Review X*, 8(3):031018, July 2018. ISSN 2160-3308. doi: 10.1103/physrevx.8.031018.

A PEARSON CORRELATION

To dynamically form connections in the reservoir, we leverage the Pearson correlation coefficient (Pearson, 1895) to identify neurons that exhibit strong linear relationships in their activation patterns. This process is performed exclusively on neurons that have not yet achieved homeostasis, as defined by the growth indicator Δz.

A.1 PEARSON CORRELATION COEFFICIENT

For two neurons i and j, the Pearson correlation coefficient, r_{ij}, measures the linear relationship between their respective activation states, $x_i[t]$ and $x_j[t]$, over a time period T. It is defined as:

$$r_{ij} = \frac{\sum_{t=1}^{T}(x_i[t] - \bar{x}_i)(x_j[t] - \bar{x}_j)}{\sqrt{\sum_{t=1}^{T}(x_i[t] - \bar{x}_i)^2}\sqrt{\sum_{t=1}^{T}(x_j[t] - \bar{x}_j)^2}}, \tag{7}$$

where:

- $x_i[t]$ and $x_j[t]$ are the activation states of neurons i and j at time t,
- $\bar{x}_i = \frac{1}{T}\sum_{t=1}^{T} x_i[t]$ is the mean activation state of neuron i over the period T,
- $\bar{x}_j = \frac{1}{T}\sum_{t=1}^{T} x_j[t]$ is the mean activation state of neuron j over the same period.

The coefficient r_{ij} ranges from -1 (perfect negative correlation) to 1 (perfect positive correlation), with 0 indicating no linear relationship.

A.2 PEARSON AS A MEASURE OF RICHNESS

Figure 5 illustrates the average Pearson correlation between the activations of all neurons in the reservoir within each time window. Using a sliding window approach, the reservoir states are segmented into overlapping windows of a fixed size. For each window, the pairwise Pearson correlation coefficients, r_{ij}, are calculated for all neuron pairs (i, j), excluding self-correlations ($i = j$).

The average Pearson correlation for a given time window is defined as:

$$\mu_{\text{window}} = \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} r_{ij}, \tag{8}$$

where:

- N is the number of neurons in the reservoir,
- r_{ij} is the Pearson correlation coefficient between neurons i and j over the time window.

This metric, μ_{window}, represents the mean correlation between neuron activations in the window. By plotting μ_{window} for successive time windows, Figure 5 shows how the average correlation evolves during the pretraining phase.

Lower average correlations over time indicate reduced synchronization among neurons, reflecting the reservoir's increasing ability to generate a diverse and decorrelated feature space. This is a desirable property in reservoirs, as it enhances their capacity to separate input patterns in a high-dimensional space, aligning with the goals of effective reservoir design.

B DETAILS ON THE HAG ALGORITHM

B.1 PSEUDO-CODE

Algorithm 1: HAG Algorithm

```
Input: Reservoir weights $\mathbf{W}$, Input weights $\mathbf{W}_{\text{in}}$, Bias $\mathbf{b}$, Pretraining data $X_{\text{pretrain}}$,
Hyperparameters $(\rho, \beta, \delta w, \gamma, T_{\text{min}}, T_{\text{max}})$
Output: Adjusted reservoir weights $\mathbf{W}$
for each time increment $T_{current}$ sampled from logspace between $T_{min}$ and $T_{max}$ do
    for $t \leftarrow 1$ to $T_{current}$ do
        Update reservoir states:
        $\mathbf{x}[t+1] \leftarrow \sigma\left(\mathbf{W}\mathbf{x}[t] + \mathbf{W}_{\text{in}}\mathbf{u}[t] + \mathbf{b}\right)$
    end
    for each neuron $i$ do
        Compute activity measure $s_i$ (mean or variance over $T_{\text{current}}$)
        Compute growth indicator:
        $\Delta z_i \leftarrow \frac{1}{\beta}(s_i - \rho)$
        if $\Delta z_i < -1$ then
            Find neuron $j$ with highest Pearson correlation with neuron $i$
            Increase weight:
            $w_{ij} \leftarrow w_{ij} + \delta w$
            if degree(i) ¿ $\gamma$ then
                // Implement logic to maintain maximum degree
            end
        end
        if $\Delta z_i > +1$ then
            Randomly select a synapse $w_{ij}$ connected to neuron $i$ to decrease
            $w_{ij} \leftarrow \max(w_{ij} - \delta w, 0)$
                                                // Ensure non-negativity
        end
        if variance-HAG and $x_i$ exceeds saturation threshold $\theta_{sat}$ then
            for each outgoing synapse $w_{ij}$ of neuron $i$ do
                $w_{ij} \leftarrow w_{ij} \times \eta_{\text{sat}}$
            end
        end
    end
end
```

B.2 CONNECTION FORMATION BASED ON PEARSON CORRELATION

Identifying the Most Correlated Pair. To form new connections, we consider only neurons i and j for which the growth indicator Δz_i or Δz_j satisfies:

$$\Delta z_i > 1 \quad \text{or} \quad \Delta z_j > 1. \tag{9}$$

For this subset of neurons, we compute pairwise correlation coefficients r_{ij} for all pairs, with r_{ij} as defined in A.1. The pair (i^*, j^*) with the highest absolute correlation is selected:

$$(i^*, j^*) = \arg\max_{(i,j)} |r_{ij}|, \tag{10}$$

where the maximization is performed over all neuron pairs (i, j) that have not yet achieved homeostasis.

Establishing the Connection. Once the most highly correlated pair (i^*, j^*) is identified, a connection is established between these two neurons by incrementing the corresponding weight $w_{i^*j^*}$ in the connectivity matrix $\mathbf{W}$. The updated weight is given by:

$$w_{i^*j^*} \leftarrow w_{i^*j^*} + \delta w,$$

where $\delta w > 0$ is the weight increment parameter.

This mechanism ensures that connections are formed preferentially between neurons that exhibit high correlation, promoting the restructuring of the reservoir to enhance its dynamic representation of input data.

Table 5: Hyperparameter ranges, settings, and notation

CATEGORY	PARAMETER	SYMBOL	RANGE/SETTINGS
Shared Across Models	Input Scaling	s_{in}	0.01 to 0.2, step 0.005
	Bias Scaling	s_b	0 to 0.2, step 0.005
	Ridge Coefficient	λ	Logarithmic scale from 10^{-15} to 10^1
	Maximum degree	γ	Between 10 to 20
	Weight Increment	δw	0.001 to 0.1, step 0.001
mean-HAG	Rate Target	ρ_r	0.5 to 1, step 0.01
	Rate Spread	β_r	0.01 to 0.4, step 0.005
variance-HAG	Variance Target	ρ_v	0.001 to 0.02, step 0.001
	Variance Spread	β_v	0.001 to 0.02, step 0.001
	Saturation Threshold	θ_{sat}	0.8 to 0.98, step 0.02
	Saturation scaling	η_{sat}	0.8 to 0.98, step 0.02
	Connection Selection	-	”random” or ”pearson”
E-ESN/ESN	Connectivity	p	0 to 1, no step
	Spectral Radius	ρ_s	0.4 to 1.6, step 0.01
IP-ESN	Target distribution’s mean	μ	0 to 1, no step
	Target distribution’s variance	σ_{IP}	0 to 1, no step
Fixed Parameters	Activation Function	σ	Hyperbolic Tangent (tanh)
	W_{in} Connectivity	-	1 (Full Connectivity)
	Network Size	n	Just above 500 neurons
	Input Duplication	-	Equal duplication for each input variate

C DETAILS ON THE HYPERPARAMETER OPTIMIZATION

In this appendix, we present the detailed results of the hyperparameter optimization across the different datasets. The hyperparameters were optimized using Optuna over 400 trials per dataset.

The detailed hyperparameter settings serve as a reference for reproducing the results and offer insights into how different parameters impact the performance of the different algorithm.

Table 6: Optimized hyperparameters for the variance-HAG algorithm

	s_{in}	s_b	λ	ρ_v	β_v	θ_{sat}	η_{sat}	δw	γ
CatsDogs	0.030	0.000	10^0	0.017	0.007	0.86	0.88	0.100	11
FSDD	0.155	0.020	10^{-7}	0.011	0.001	0.82	0.94	0.087	14
Japanese Vowels	0.040	0.190	10^{-9}	0.003	0.013	0.90	0.86	0.075	18
Lorenz	0.065	0.170	10^{-9}	0.018	0.005	0.94	0.96	0.006	17
MackeyGlass	0.170	0.185	10^{-9}	0.019	0.007	0.80	0.94	0.076	10
SPEECHCOMMANDS	0.060	0.000	10^{-9}	0.014	0.003	0.92	0.80	0.030	20
SpokenArabicDigits	0.060	0.040	10^{-10}	0.014	0.001	0.82	0.96	0.057	16
Sunspot	0.180	0.135	10^{-9}	0.017	0.005	0.94	0.94	0.028	14

Table 7: Optimized hyperparameters for the mean-HAG algorithm

	s_{in}	s_b	λ	ρ_r	β_r	δw	γ
CatsDogs	0.120	0.100	10^0	0.51	0.39	0.021	12
FSDD	0.200	0.075	10^{-5}	0.56	0.39	0.050	10
Japanese Vowels	0.195	0.195	10^{-4}	0.66	0.21	0.094	13
Lorenz	0.190	0.170	10^{-9}	0.97	0.165	0.021	19
MackeyGlass	0.045	0.185	10^{-10}	0.94	0.295	0.043	10
SPEECHCOMMANDS	0.060	0.000	10^{-9}	0.58	0.34	0.002	16
SpokenArabicDigits	0.075	0.035	10^{-10}	0.53	0.38	0.025	11
Sunspot	0.050	0.105	10^{-9}	0.51	0.265	0.087	13

C.1 HYPERPARAMETER DEFINITIONS:

- **Input Scaling (s_{in}):** Scaling factor applied to the input weights $\mathbf{W}_{\text{in}}$.
- **Bias Scaling (s_b):** Scaling factor applied to the bias vector $\mathbf{b}$.
- **Ridge Coefficient (λ):** Regularization parameter in ridge regression, where $\lambda = 10^{\text{ridge exponent}}$.
- **Variance Target (ρ_v):** Target standard deviation of neuron states.
- **Rate Spread (β_r):** Spread parameter activity controlling the activity deviation from target ρ_r that is tolerated.
- **Rate Target (ρ_r):** Target activity of neuron states.
- **Variance Spread (β_v):** Spread parameter controlling the sensitivity to deviations from ρ_v.
- **Saturation Threshold (θ_{sat}):** Threshold beyond which intrinsic plasticity mechanisms reduce synaptic weights.
- **Saturation scaling (η_{sat}):** Factor by which synaptic weights are scaled when saturation occurs.
- **Weight Increment (δw):** Amount by which synaptic weights are increased during growth.
- **Maximum Degree (γ):** Maximum number of synaptic partners per neuron.
- **Mean of the target distribution for IP (μ):** Target mean for intrinsic plasticity normalization.
- **Variance of the target distribution for IP (σ_{IP}):** Target standard deviation for intrinsic plasticity normalization.

C.2 OPTIMIZED HYPERPARAMETERS FOR EACH ALGORITHM

Tables 7, 6, 8, 9 and 10 summarize the optimal hyperparameters found for each dataset and algorithm. Key insights include:

Table 8: Optimized hyperparameters for E-ESN

	s_{in}	s_b	λ	p	ρ_s
CatsDogs	0.135	0.025	10^{-6}	0.842	0.49
FSDD	0.175	0.14	10^{-9}	0.0040	0.92
Japanese Vowels	0.190	0.005	10^{-9}	0.155	0.62
MackeyGlass	0.195	0.2	10^{-10}	0.0318	1.4
SPEECHCOMMANDS	0.040	0.05	10^{-12}	0.0038	1.02
SpokenArabicDigits	0.030	0.1	10^{-14}	0.0049	1.0
Sunspot	0.055	0.005	10^{-9}	0.0178	0.78

Table 9: Optimized hyperparameters for traditional ESN

	s_{in}	s_b	λ	p	ρ_s
CatsDogs	0.135	0.025	10^{-6}	0.842	0.49
FSDD	0.175	0.140	10^{-9}	0.004	0.92
Japanese Vowels	0.190	0.005	10^{-9}	0.155	0.62
MackeyGlass	0.130	0.045	10^{-10}	0.193	0.89
SPEECHCOMMANDS	0.135	0.005	10^{-11}	0.356	1.00
SpokenArabicDigits	0.035	0.005	10^{-14}	0.003	0.61
Sunspot	0.025	0.110	10^{-11}	0.014	0.46

1. **Variance-HAG and Mean-HAG:** Effective in dynamically tailoring the reservoir to the data, with significant reliance on ρ_v, ρ_r, and γ to optimize neuron activity and connectivity.
2. **Traditional ESN and E-ESN:** Spectral radius (ρ_s) and connectivity (p) are critical for matching the reservoir's memory and dynamic properties to the dataset.
3. **General Trends:** High input scaling (s_{in}) and low ridge coefficients (λ) are commonly effective across algorithms, reflecting the need for strong input signals and minimal regularization.

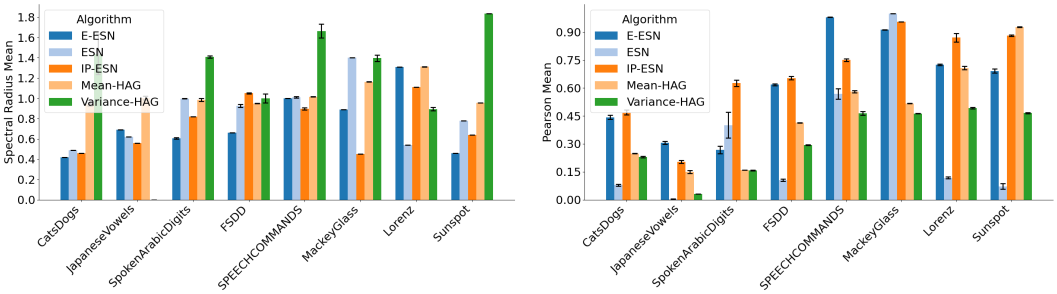
Table 10: Optimized hyperparameters for IP-ESN

	s_{in}	s_b	λ	p	ρ_s	μ	σ
CatsDogs	0.18	0.08	10^{-10}	0.462	0.46	0.733	0.142
FSDD	0.05	0.095	10^{-14}	0.005	1.05	0.298	0.087
Japanese Vowels	0.14	0.06	10^{-10}	0.327	0.56	0.886	0.719
Lorenz	0.09	0.06	10^{-9}	0.093	1.11	0.071	0.838
MackeyGlass	0.19	0.19	10^{-7}	0.048	0.45	0.158	0.363
SPEECHCOMMANDS	0.195	0.105	10^{-14}	0.005	0.89	0.097	0.062
SpokenArabicDigits	0.03	0.125	10^{-15}	0.015	0.82	0.318	0.582
Sunspot	0.01	0.145	10^{-11}	0.138	0.64	0.758	0.782

D ADDITIONAL RESULTS AND ANALYSIS

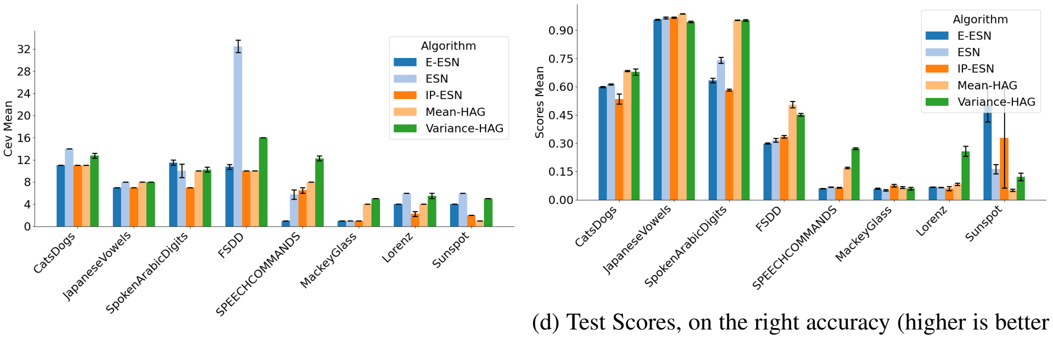
To further substantiate the effectiveness of our Hebbian Architecture Generation (HAG) method, we have expanded our experiments to include detailed results and additional metrics across our datasets and reservoir configurations. Specifically, we provide an in-depth analysis of the spectral radius, Pearson correlation coefficients among neuron activations, and the Cumulative Explained Variance (CEV) in the reservoir states. These metrics offer insights into the dynamical properties of the reservoirs and their impact on performance. Table 11 and 12summarizes the spectral radius, average Pearson correlation, and CEV for each combination of dataset and reservoir configuration. The spectral radius is measured from the connectivity matrix of the different networks obtained with various rules, while Pearson correlation and CEV are measured based on neurons' activity during the test set inference for each dataset.

D.1 RESULTS



(a) Spectral Radius

(b) Pearson Correlation



(c) Cumulative Explained Variance

(d) Test Scores, on the right accuracy (higher is better and on the left NRMSE (lower is better)

Figure 7: Detailed Results for Spectral Radius, Pearson Correlation, Cumulative Explained Variance, and Test Scores across different reservoir configurations and Datasets. Each subfigure illustrates the respective metric for various algorithms across the evaluated datasets.

D.1.1 SPECTRAL RADII

The spectral radius is an important parameter in Echo State Networks (ESNs), influencing the echo state property. Our results indicate that the HAG-based methods (*Mean-HAG* and *Variance-HAG*) generally have higher spectral radii compared to the traditional E-ESN, particularly in datasets like SPEECHCOMMANDS and CatsDogs (see Figure 7a). This suggests that the adaptive connectivity in HAG methods allows the reservoir to have a high spectral radius while performing extremly well, for instance the SPEECHCOMMANDS dataset.

D.1.2 PEARSON CORRELATION

The average Pearson correlation coefficient among neuron activations provides insight into the redundancy and diversity of the reservoir states. Lower correlation values imply a more decorrelated and thus more informative set of features for the readout layer.

In our experiments, the *Variance-HAG* method consistently achieved lower Pearson correlation coefficients compared to other methods across multiple datasets, as shown in Table 11 and Table 12 and illustrated in Figure 7b. For instance, in the SPEECHCOMMANDS dataset, *Variance-HAG* achieved a correlation of 0.471, significantly lower than the 0.980 observed in the random E-ESN. Similarly, in the MackeyGlass dataset, *Variance-HAG* achieved a correlation of 0.064, compared to 0.910 in the random E-ESN. This reduction in correlation indicates that *Variance-HAG* effectively decorrelates neuron activations, potentially leading to better generalization and performance.

D.1.3 CUMULATIVE EXPLAINED VARIANCE

The Cumulative Explained Variance (CEV) quantifies the dimensionality of the reservoir's projected feature space by identifying the number of principal components needed to explain a specified fraction of the variance. Higher CEV values indicate that more components are required to capture the system's dynamics, which can reflect richer dynamics or increased complexity.

Our results show that the HAG methods doesn't always exhibit higher CEV values. But it is important to notice that it does on the most complex dataset (SPEECHCOMMAND) where our algorithm shows the most impressive results.

Table 11: Spectral radius, Pearson Correlation, Cumulative Explained Variance (CEV), and test scores for the different reservoir configurations across prediction datasets' test data (average over 4 trials).

Dataset	**Algorithm**	**SR**	**Correlation**	**CEV**	**Test Scores**
MackeyGlass	E-ESN	0.89 ± 0.00	0.913 ± 0.001	1.0 ± 0.0	0.059 ± 0.004
	ESN	1.40 ± 0.00	1.00 ± 0.00	1.0 ± 0.0	0.051 ± 0.005
	IP-ESN	0.45 ± 0.00	0.955 ± 0.001	1.0 ± 0.0	0.076 ± 0.007
	Mean-HAG	1.16 ± 0.00	0.517 ± 0.001	4.0 ± 0.0	0.065 ± 0.005
	Variance-HAG	1.40 ± 0.03	0.463 ± 0.001	5.0 ± 0.0	0.060 ± 0.007
Lorenz	E-ESN	1.31 ± 0.00	0.725 ± 0.004	4.0 ± 0.0	0.068 ± 0.001
	ESN	0.54 ± 0.00	0.118 ± 0.005	6.0 ± 0.0	0.066 ± 0.001
	IP-ESN	1.11 ± 0.00	0.871 ± 0.023	2.3 ± 0.4	0.060 ± 0.011
	Mean-HAG	1.31 ± 0.00	0.707 ± 0.009	4.0 ± 0.0	0.083 ± 0.006
	Variance-HAG	0.90 ± 0.02	0.493 ± 0.004	5.5 ± 0.5	0.257 ± 0.027
Sunspot	E-ESN	0.46 ± 0.00	0.691 ± 0.011	4.0 ± 0.0	0.505 ± 0.092
	ESN	0.78 ± 0.00	0.072 ± 0.014	6.0 ± 0.0	0.163 ± 0.024
	IP-ESN	0.64 ± 0.00	0.882 ± 0.004	2.0 ± 0.0	0.610 ± 0.687
	Mean-HAG	0.96 ± 0.00	0.928 ± 0.002	1.0 ± 0.0	0.051 ± 0.005
	Variance-HAG	1.84 ± 0.00	0.466 ± 0.002	5.0 ± 0.0	0.122 ± 0.020

D.2 DISCUSSION

The analysis of these metrics supports the efficacy of the HAG methods in improving reservoir performance:

- **Enhanced Dynamics**: Higher spectral radii in HAG methods suggest a more powerful dynamic regime, allowing the reservoir to better model temporal dependencies.
- **Reduced Redundancy**: Lower Pearson correlations indicate that HAG methods produce more diverse neuron activations, reducing redundancy and providing richer information to the readout layer.
- **Enriched Feature Space**: Higher CEV values demonstrate that HAG methods generate a more informative and expansive feature space, facilitating better representation of input dynamics.

However, while HAG algorithms excel in classification tasks by enhancing feature separability, their performance in prediction tasks is more variable. As shown in Table 11 and Table 12, *Variance-HAG* does not always outperform traditional ESNs or other baseline models in prediction scenarios. This inconsistency suggests that the mechanisms driving connectivity adjustments in HAG may be more aligned with tasks requiring distinct feature separation rather than continuous temporal forecasting.

D.3 IMPLICATIONS FOR RESERVOIR DESIGN

Despite these shortcomings, the HAG methods offer valuable insights into how adaptive connectivity can enhance reservoir performance. The ability to dynamically tailor the reservoir to specific

Table 12: Spectral radius, Pearson Correlation, Cumulative Explained Variance (CEV), and test scores for the different reservoir configurations across classification datasets' test data (average over 4 trials).

Dataset	**Algorithm**	**SR**	**Correlation**	**CEV**	**Test Scores**
CatsDogs	E-ESN	0.42 ± 0.00	0.443 ± 0.011	11.0 ± 0.0	0.599 ± 0.003
	ESN	0.49 ± 0.00	0.078 ± 0.005	14.0 ± 0.0	0.613 ± 0.003
	IP-ESN	0.46 ± 0.00	0.469 ± 0.013	11.0 ± 0.0	0.535 ± 0.026
	Mean-HAG	0.94 ± 0.02	0.249 ± 0.001	11.0 ± 0.0	0.684 ± 0.003
	Variance-HAG	1.44 ± 0.20	0.230 ± 0.004	12.75 ± 0.43	0.678 ± 0.017
JapaneseVowels	E-ESN	0.69 ± 0.00	0.306 ± 0.008	7.0 ± 0.0	0.957 ± 0.002
	ESN	0.62 ± 0.00	0.004 ± 0.000	8.0 ± 0.0	0.966 ± 0.005
	IP-ESN	0.56 ± 0.00	0.203 ± 0.008	7.0 ± 0.0	0.968 ± 0.003
	Mean-HAG	1.00 ± 0.02	0.149 ± 0.007	8.0 ± 0.0	0.987 ± 0.001
	Variance-HAG	0.00 ± 0.00	0.031 ± 0.000	8.0 ± 0.0	0.945 ± 0.003
SpokenArabicDigits	E-ESN	0.61 ± 0.01	0.268 ± 0.020	11.5 ± 0.5	0.634 ± 0.012
	ESN	1.00 ± 0.00	0.401 ± 0.068	10.0 ± 1.22	0.740 ± 0.016
	IP-ESN	0.82 ± 0.00	0.626 ± 0.017	7.0 ± 0.0	0.582 ± 0.005
	Mean-HAG	0.99 ± 0.01	0.159 ± 0.000	10.0 ± 0.0	0.954 ± 0.002
	Variance-HAG	1.41 ± 0.01	0.157 ± 0.003	10.25 ± 0.43	0.952 ± 0.004
FSDD	E-ESN	0.66 ± 0.00	0.618 ± 0.005	10.75 ± 0.43	0.299 ± 0.004
	ESN	0.93 ± 0.01	0.105 ± 0.006	32.5 ± 1.12	0.316 ± 0.010
	IP-ESN	1.05 ± 0.01	0.653 ± 0.009	10.0 ± 0.0	0.335 ± 0.007
	Mean-HAG	0.95 ± 0.00	0.414 ± 0.001	10.0 ± 0.0	0.506 ± 0.017
	Variance-HAG	1.00 ± 0.04	0.293 ± 0.002	16.0 ± 0.0	0.452 ± 0.006
SPEECHCOMMANDS	E-ESN	1.00 ± 0.00	0.980 ± 0.001	1.0 ± 0.0	0.060 ± 0.001
	ESN	1.01 ± 0.01	0.569 ± 0.028	5.75 ± 0.83	0.068 ± 0.001
	IP-ESN	0.90 ± 0.01	0.750 ± 0.007	6.5 ± 0.5	0.065 ± 0.002
	Mean-HAG	1.02 ± 0.00	0.581 ± 0.005	8.0 ± 0.0	0.169 ± 0.004
	Variance-HAG	1.66 ± 0.07	0.464 ± 0.009	12.25 ± 0.43	0.272 ± 0.006

tasks by reducing neuron correlations and expanding the feature space underscores the potential of biologically inspired design principles in neural network architecture.

It is important to note that enriching the feature space (lower Pearson correlation and higher CEV) does not always translate into augmented performances across all tasks. This discrepancy may stem from the inherent differences between classification and prediction tasks. Classification tasks primarily benefit from enhanced feature separability, allowing for more accurate differentiation between classes. In contrast, prediction tasks rely heavily on the reservoir's ability to capture and retain temporal dependencies, which may not be directly enhanced by the structural adjustments made by HAG.

In conclusion, the Hebbian Architecture Generation (HAG) method presents a robust framework for enhancing reservoir computing, particularly in classification tasks, by leveraging biologically inspired adaptive connectivity. While its efficacy in prediction tasks is promising, it highlights the need for a nuanced approach that considers the distinct demands of different task types. The adaptability and biological plausibility of HAG not only address the limitations of static reservoir architectures but also pave the way for more versatile and efficient neural network models in diverse learning environments.