A BENCHMARK ANALYSIS OF GRAPH AND NON-GRAPH METHODS FOR CAENORHABDITIS ELEGANS NEURON CLASSIFICATION

Jingqi Lu[†] *Keqi Han*[†] *Yun Wang*[§] *Lu Mi*[‡*] *Carl Yang*[†]

[†] Department of Computer Science, Emory University, USA
[§] Department of Biomedical Informatics, Emory University, USA
[‡] College of AI, Tsinghua University, China

ABSTRACT

This study establishes a benchmark for *Caenorhabditis elegans* neuron classification, comparing four graph methods (GCN, GraphSAGE, GAT, GraphTransformer) against four non-graph methods (Logistic Regression, MLP, LOLCAT, NeuPRINT). Using the functional connectome, we classified Sensory, Interneuron, and Motor neurons based on Spatial, Connection, and Neuronal Activity features. Results show that attention-based GNNs significantly outperform baselines on the Spatial and Connection features. The Neuronal Activity features yielded poor performance, likely due to the low temporal resolution of the underlying neuronal activity data. Our benchmark validates the use of GNNs and highlights that Spatial and Connection features are key predictors for *Caenorhabditis elegans* neuron classes. Code is available at: https://github.com/JingqiLuu/neuronclf-gnn-benchmark.

***Index Terms*—** Neuron Classification, Graph Neural Networks, Benchmark

1. INTRODUCTION

Neuron classification is a fundamental problem in neuroscience and is essential for understanding the organization and function of neural circuits [1]. In *Caenorhabditis elegans* (*C. elegans*), neurons are categorized into three functional classes: Sensory, Interneurons, and Motor [1, 2]. Owing to its compact and fully mapped nervous system, *C. elegans* serves as an ideal platform for studying this problem. Recently, Randi et al. [2] constructed a neural signal propagation atlas that captures functional signal dynamics not predictable from anatomy alone. This atlas provides both functional connectome information and neuronal activity data, enabling data-driven neuron classification and deeper insights into information flow in neural circuits. [3].

Traditionally, neuron identification relied on static features such as morphology or molecular markers [4, 5, 6]. With the advent of large-scale neural recording technologies, researchers have increasingly explored classification methods based on neuronal activity time series [7, 8, 9]. Recent deep learning models, such as LOLCAT [7] and NeuPRINT [8], have shown promise in classifying neurons using individual neuronal activity. In parallel, modeling the brain as a network has driven the rapid adoption of Graph Neural Networks (GNNs) in macroscopic brain network analysis, especially for tasks like disease diagnosis or biomarker identification [10, 11, 12]. Despite these advances, the integration of activity-based representations and graph-based learning frameworks for neuron classification at the microscale, especially in the *C. elegans* functional connectome, remains largely unexplored.

This study establishes a benchmark for neuron classification in *C. elegans*. We systematically evaluate graph methods against non-graph methods (e.g., LOLCAT [7], NeuPRINT [8]) using the functional connectome [2] and three feature types: Spatial, Connection, and Neuronal Activity. This study aims to (1) assess the relative performance of graph-based and non-graph-based approaches, (2) identify the most informative feature type for neuron classification, and (3) compare the efficacy of different GNN architectures.

2. METHOD

2.1. Dataset

The data for this study are drawn from the openly accessible *C. elegans* functional connectome published by Randi et al. [2]. This connectome is modeled as a directed weighted graph $G = (V, E, W)$. The node set V consists of the 300 neurons in the *C. elegans* nervous system. The edge set E contains 8,703 directed edges, representing functional signal propagation, while W is the corresponding weighted adjacency matrix where W_{ij} denotes the functional connection strength from neuron i to j. For the neuron classification task, we adopt the dataset's intrinsic functional categorization as ground truth labels, which classifies neurons into three distinct classes: Sensory (90 neurons), Interneuron (93 neurons), and Motor (117 neurons) [2]. The dataset also provides the raw data for feature construction (detailed in Section 2.2), including the neuronal activity data for each neuron (2Hz sampling rate) and their spatial coordinates.

* LM's work was done when she was a full-time employee at Georgia Institute of Technology, USA.

2.2. Feature Construction

Based on best practices in brain network analysis [10] and research on neuronal activity patterns [7, 8], we constructed and evaluated the following three types of features:

- **Spatial Feature**: The feature for each neuron is a 3-dimensional vector $f_{\text{spatial}} = (x, y, z)$, representing its standardized coordinates in physical space.
- **Connection Feature**: Each neuron's feature $f_{\text{conn}} = W_{i,:}$ is its row vector in the adjacency matrix W, capturing its functional connectivity with all other neurons [10].
- **Neuronal Activity Feature**: This feature is constructed following LOLCAT [7]. The continuous neuronal activity data for each neuron ($X_i(t)$) is first segmented into non-overlapping windows ($W_{i,k}$) of duration T_w. Within each window, we detect events via thresholding to calculate Interspike Intervals (ISIs). For each window $W_{i,k}$, these ISIs are summarized into an empirical distribution vector $\mathbf{d}_{i,k} \in \mathbb{R}^{D_{ISI}}$, where the value at index $\tau - 1$ represents the count of ISIs equal to τ time steps (for $1 \leq \tau \leq D_{ISI}$). The final feature for neuron i is a sequence $\mathcal{S}_i = (\mathbf{d}_{i,1}, ..., \mathbf{d}_{i,N_k})$. Sequences are padded to a uniform maximum length L_{max} for batch processing, resulting in a feature tensor $F_{ISI} \in \mathbb{R}^{N \times L_{max} \times D_{ISI}}$, where N denotes the number of neurons.

2.3. Model Architectures

Two main categories of models were compared: graph models and non-graph models [7, 8] that do not explicitly use the graph adjacency matrix. For each model-feature combination, optimal hyperparameters were determined via Bayesian optimization (detailed in Section 2.4). The resulting key parameters are summarized in their respective descriptions.

Graph methods. Four mainstream GNN architectures based on the message-passing paradigm were evaluated. These models iteratively update node representation by aggregating neighborhood features. In each layer l, node v_i's representation h_i^l is updated via message passing and updating.

In the message passing step, each node v_i receives messages from all its neighbors. This process can be formally described by:

$$m_i^l = \sum_{j \in N_i} M^l(h_i^{l-1}, h_j^{l-1}, w_{ij}), \tag{1}$$

where N_i denotes the neighbor set of v_i, h_j^{l-1} is the feature representation of a neighboring node j from the previous layer, w_{ij} represents the weight of the edge from node j to i, and M^l is the message function at layer l.

In the update step, the aggregated message m_i^l is used to update the v_i's representation from the previous layer:

$$h_i^l = U^l(h_i^{l-1}, m_i^l), \tag{2}$$

where U^l is the update function at layer l, which is typically a learnable linear transform with a non-linear activation. The GNN models, which differ in their definitions of the message (M^l) and update (U^l) functions, include:

- **Graph Convolutional Network (GCN)** [13]: A classic GNN model that performs convolution by aggregating the weighted features of neighboring nodes. Optimized models used 2-6 layers and 30 to 160 hidden dimensions.
- **GraphSAGE** [14]: Employs neighborhood sampling and with various aggregation functions, demonstrating good scalability for large-scale graphs. Optimized models used 2-7 layers with hidden dimensions between 60 and 260.
- **Graph Attention Network (GAT)** [15]: Uses a multi-head attention mechanism to assign different importance weights to each neighbor for more flexible information aggregation. Optimized models used 2-5 layers, 40-190 hidden dimensions, and 1 to 8 attention heads.
- **GraphTransformer** [16]: Extends the Transformer architecture to graph-structured data by incorporating self-attention mechanisms that capture long-range dependencies and structural relationships between nodes. Optimized models used 2-4 layers, hidden dimensions between 50 and 200, and 1 to 8 attention heads.

Non-Graph methods. We also evaluated four models that do not explicitly use the graph adjacency matrix:

- **Multi-Layer Perceptron (MLP)**: A standard feed-forward network. Optimized models used 2 to 6 layers with hidden dimensions varying between approximately 150 and 460.
- **Logistic Regression**: A multinomial linear classification model. Optimized models used regularization strength that C ranging from 10^{-4} to 10^{1} and L1 penalty type.
- **LOLCAT** [7]: A supervised attention-based model designed to classify a neuron using only its isolated neuronal activity data. It operates by (1) dividing the full neuronal activity data into short "snippets", (2) computing the ISI distribution for each snippet, and (3) using a multi-head attention mechanism to learn which snippets are most informative for classification. Optimized models used 16-64 embedding/hidden dimensions and 4 attention heads.
- **NeuPRINT** [8]: A self-supervised framework learning a time-invariant representation for each neuron. It uses a Transformer model to predict a neuron's future activity based on its own past activity and permutation-invariant statistics of surrounding population activity. We use this framework to generate representation vectors (64-dim), which are then fed into an MLP classifier.

2.4. Implementation Details

All model evaluations were performed using a stratified 5-fold Cross-Validation (CV) scheme for robustness. The dataset was split into five folds, with each fold serving once as the test set while the remaining four were used for training. Hyperparameters for each model-feature combination were optimized using the Optuna framework, an open-source tool designed for automating optimization processes [17]. We em-

Table 1: Performance Comparison of Graph and Non-graph Methods for Neuron Classification Across Three Feature Sets. For each metric (column), the highest score is in **bold** and the second-highest score is underlined.

Type	Model	Spatial Feature		Neuronal Activity Feature		Connection Feature	
		Accuracy	F1	Accuracy	F1	Accuracy	F1
Non-graph method	Logistic Regression	$56.33_{\pm 4.52}$	$54.17_{\pm 4.38}$	$36.59_{\pm 6.20}$	$17.86_{\pm 6.11}$	$54.67_{\pm 5.62}$	$52.49_{\pm 6.56}$
	MLP	$56.67_{\pm 2.98}$	$55.52_{\pm 3.69}$	$40.00_{\pm 5.25}$	$31.81_{\pm 4.93}$	$54.67_{\pm 1.94}$	$53.00_{\pm 1.75}$
	LOLCAT	$57.00_{\pm 2.67}$	$55.86_{\pm 2.31}$	$39.00_{\pm 5.83}$	$25.11_{\pm 8.84}$	$54.00_{\pm 2.26}$	$52.78_{\pm 2.28}$
	NeuPRINT	$55.33_{\pm 9.33}$	$50.87_{\pm 10.61}$	$45.57_{\pm 8.69}$	$37.19_{\pm 14.18}$	$50.67_{\pm 4.90}$	$42.51_{\pm 13.08}$
Graph method	GCN	$60.00_{\pm 6.65}$	$54.38_{\pm 8.16}$	$45.00_{\pm 8.50}$	$40.53_{\pm 9.83}$	$58.05_{\pm 7.14}$	$44.87_{\pm 12.29}$
	GraphSAGE	$57.07_{\pm 4.52}$	$44.09_{\pm 5.54}$	$\mathbf{50.00_{\pm 10.21}}$	$\mathbf{47.34_{\pm 10.11}}$	$60.49_{\pm 4.73}$	$52.37_{\pm 6.07}$
	GAT	$61.46_{\pm 2.84}$	$57.12_{\pm 3.62}$	$47.50_{\pm 10.07}$	$46.03_{\pm 10.54}$	$61.46_{\pm 9.56}$	$55.08_{\pm 12.94}$
	GraphTransformer	$\mathbf{62.44_{\pm 3.96}}$	$\mathbf{57.12_{\pm 4.46}}$	$49.17_{\pm 9.65}$	$45.87_{\pm 11.18}$	$\mathbf{62.44_{\pm 6.28}}$	$\mathbf{58.73_{\pm 5.77}}$

ployed Bayesian optimization with the Tree-structured Parzen Estimator (TPE) algorithm [18] as the sampler to efficiently search the hyperparameter space. Each model-feature combination ran 30 optimization trials. The optimization objective was to maximize the mean 5-fold CV validation accuracy. The hyperparameter search space included key parameters such as learning rate (log-uniform between 10^{-5} and 10^{-2}), weight decay (log-uniform), hidden layer dimensions (uniform between 32 and 512), number of network layers (integer between 2 and 7), and dropout rate (uniform between 0.0 and 0.8). Model-specific parameters (e.g., attention heads for GAT) were also included in the search.

3. RESULTS

We evaluated all models across the three feature sets. The reported results represent the mean accuracy and F1 score (± standard deviation) obtained from 5-fold cross-validation. Experimental results are summarized in Table 1.

3.1. Comparing Features

Spatial and Connection Features. The Spatial and Connection features are effective for distinguishing the three major classes of *C. elegans* neurons, with most models achieving accuracies around 55%-62%. This indicates that a neuron's functional class is closely related to both its connection patterns within the functional connectome and its spatial layout.

The effectiveness of Spatial feature stems from its strong class information: As shown in Fig.1c and Fig.1d, the *C. elegans* nervous system is highly structured. Sensory neurons (red) and interneurons (green) are predominantly located in the head region, while motor neurons (blue) are primarily distributed along the ventral nerve cord.

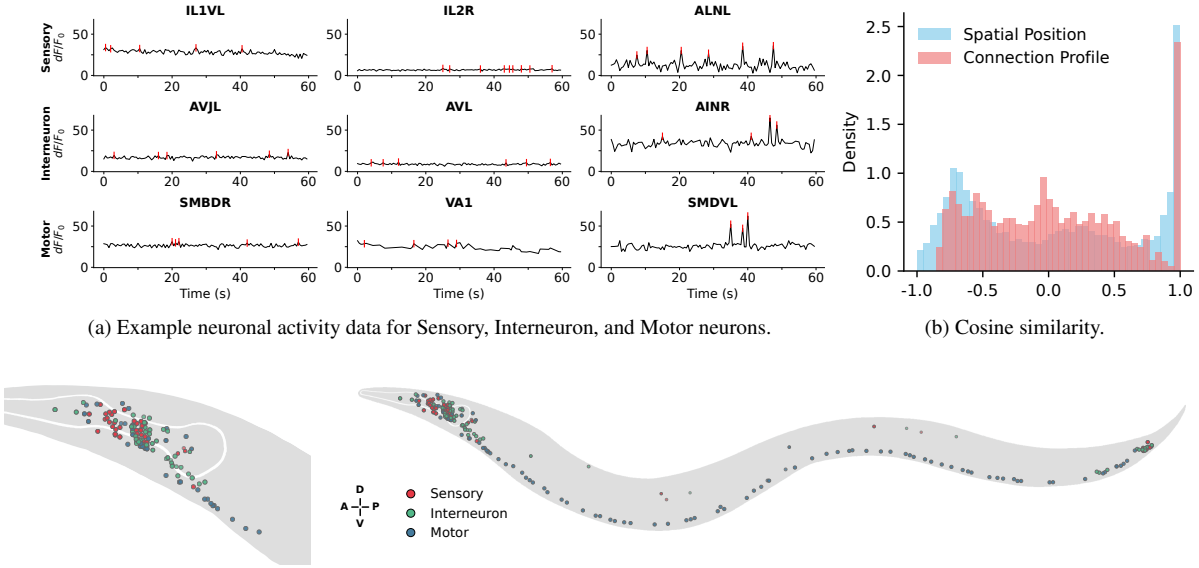
Spatial-Connectivity Feature Correlation. An interesting finding is that the Spatial and Connection feature, although conceptually different, ultimately yielded very similar classification performance. To investigate this, we analyzed the intrinsic correlation between these two feature sets. This analysis involved constructing N×N similarity matrices for both feature sets using Cosine similarity. We then computed the Pearson correlation between the Spatial and Connection Feature similarity matrices. As illustrated in Fig. 1b, the similarity distributions of the two features exhibit similar morphologies. We found a significant positive correlation: the Pearson correlation coefficient for the matrices derived from Pearson similarity was 0.324 ($p < 0.001$). This indicates that neurons that are physically close in space also tend to have similar functional connectivity patterns. This moderate coupling explains their comparable classification outcomes.

Neuronal Activity Feature. In contrast, the Neuronal Activity feature yielded poor performance on the neuron classification task, all models generally achieve accuracies below 50%. This suggests ISI are not distinct for these classes, or the low-sampling-rate (2Hz) data fails to capture them.

We attribute this poor performance primarily to the temporal resolution limitations of the dataset used in this study. As illustrated by the example raw neuronal activity data in Fig. 1a, the signals are smooth and lack the sharp transients typical of high-temporal-resolution recordings. Event detection performed on these smooth signals (indicated by red lines in the figure) reveals that many neurons produce a sparse number of events within the observation window. This low sampling rate limits the ability to accurately resolve fast neural events and may lead to the loss of high-frequency information; as noted in the LOLCAT study, a lower sampling rate impacts the quality of time-based features[7]. This signal smoothness and event sparsity directly impact models reliant on temporal patterns: it might obscure the local temporal patterns that LOLCAT seeks, as many windows contain too few events to form a meaningful ISI distribution.

3.2. Comparing Methods

Graph Methods. Attention-based GNNs achieved the highest performance on the two most informative features. When using the Spatial feature, GraphTransformer achieves the highest accuracy (62.44%) and a leading F1 score (57.12%),



(a) Example neuronal activity data for Sensory, Interneuron, and Motor neurons.

(b) Cosine similarity.

(c) Head region distribution.

(d) Spatial distribution of neuron classes.

Fig. 1: Visualization of *C. elegans* neuron data features and class distributions. (a) Example neuronal activity data for representative Sensory, Interneuron, and Motor neurons. (b) Normalized probability density distributions of cosine similarity, comparing the Spatial and Connection features across all neuron pairs. (c) Magnified view of the spatial distribution of neuron classes in the head region. (d) Spatial distribution of the three neuron classes along the entire body.

with GAT also showing strong performance (61.46%). When using the Connection Feature, GraphTransformer (62.44%) and GAT (61.46%) again perform best, significantly outperforming all non-graph models. This highlights the advantage of attention-based GNNs in capturing complex functional connectivity patterns, as they can adaptively focus on the most informative connections for neuron classification.

An important finding is how the graph structure's contribution varied by feature. On Spatial and Connection features, the performance gain of GNNs over the strongest non-graph baselines was moderate. This is likely because, as established in Section 3.1.2, these features are already intrinsically correlated with the graph topology; neurons physically close tend to have similar connections. Thus, explicitly adding the graph structure provides limited new information. Conversely, on the Neuronal Activity feature, the graph structure provided a more significant improvement. GraphSAGE unexpectedly achieved the highest accuracy (50.00%), outperforming all non-graph methods, including activity-specific models like NeuPRINT (45.57%) and LOLCAT (39.00%). This suggests that when the node feature itself is weak and noisy (as shown in Section 3.1.3), aggregating information from neighbors via message passing provides crucial contextual information that aids classification. However, even with this improvement, the ISI feature's top performance remains significantly worse than that of the other two features.

Non-Graph Methods. LOLCAT (57.00%) achieved competitive scores on Spatial Feature. On the Neuronal Activity Feature, models designed for this data type showed moderate performance such as NeuPRINT (45.57%). The poor performance of LOLCAT(39.00%), a model designed for ISI distributions, reinforces our conclusion from Section 3.1.3: the low-resolution data and resulting event sparsity likely prevent the model from learning informative temporal patterns.

4. CONCLUSION, LIMITATION AND FUTURE WORKS

This study benchmarked graph and non-graph methods for *C. elegans* neuron classification using Spatial, Connection, and Neuronal Activity features. We found that GNNs, particularly GraphTransformer and GAT, significantly outperformed non-graph baselines on the Spatial and Connection features. These two features were found to be highly informative and significantly correlated. Conversely, the Neuronal Activity feature yielded poor performance across all models, which may be attributed to the data's low temporal resolution.

Our study has several limitations. First, our models were trained on a static functional connectome, whereas brain activity is inherently dynamic. Second, although we found Spatial and Connection features to be correlated, we did not explore their combination. Future work could explore dynamic GNN models to capture the temporal evolution properties of the network. Additionally, future research could investigate feature fusion (e.g., combining Spatial and Connection features) to examine potential gains.

5. COMPLIANCE WITH ETHICAL STANDARDS

This research study was conducted retrospectively using animal subject (C. elegans) data made available in open access by Randi et al. [2]. Ethical approval was not required as confirmed by the licenses attached with the open access datasets.

6. ACKNOWLEDGMENTS

This research was partially supported by the internal funds and GPU servers provided by the Computer Science Department of Emory University, the US National Science Foundation under Award Numbers 2312502 and 2319449. LM's work was done when she was a full-time employee at Georgia Institute of Technology in the USA and was not supported by the grants and resources above.

7. REFERENCES

[1] J. G. White, E. Southgate, J. N. Thomson, and S. Brenner, "The structure of the nervous system of the nematode *Caenorhabditis elegans*," *Phil. Trans. R. Soc. Lond. B*, vol. 314, no. 1165, pp. 1–340, 1986.

[2] F. Randi, A. K. Sharma, S. Dvali, and A. M. Leifer, "Neural signal propagation atlas of *Caenorhabditis elegans*," *Nature*, vol. 623, no. 7986, pp. 406–414, Nov. 2023.

[3] M. Chalfie, J. E. Sulston, J. G. White, E. Southgate, J. N. Thomson, and S. Brenner, "The neural circuit for touch sensitivity in *Caenorhabditis elegans*," *J. Neurosci.*, vol. 5, no. 4, pp. 956–964, Apr. 1985.

[4] N. W. Gouwens et al., "Integrated Morphoelectric and Transcriptomic Classification of Cortical GABAergic Cells," *Cell*, vol. 183, no. 4, pp. 935–953.e19, Nov. 2020.

[5] E. Yemini, A. Lin, A. Nejatbakhsh, E. Varol, R. Sun, G. E. Mena, A. D. T. Samuel, L. Paninski, V. Venkatachalam, and O. Hobert, "NeuroPAL: A Multicolor Atlas for Whole-Brain Neuronal Identification in *C. elegans*," *Cell*, vol. 184, no. 1, pp. 272–288.e11, Jan. 2021.

[6] S. R. Taylor, G. Santpere, A. Weinreb, A. Barrett, M. B. Reilly, C. Xu, et al., "Molecular topography of an entire nervous system," *Cell*, vol. 184, no. 16, pp. 4329–4347.e23, Aug. 2021.

[7] A. Schneider et al., "Transcriptomic cell type structures in vivo neuronal activity across multiple timescales," *Cell Rep.*, vol. 42, no. 4, pp. 112318, Apr. 2023.

[8] L. Mi, T. Le, T. He, E. Shlizerman, and U. Sümbül, "Learning Time-Invariant Representations for Individual Neurons from Population Dynamics," *Adv. Neural Inf. Process. Syst.*, vol. 36, pp. 46007–46026, Dec. 2023.

[9] H. Yu, H. Lyu, E. Y. Xu, C. Windolf, E. K. Lee, F. Yang, A. M. Shelton, S. Olsen, S. Minavi, O. Winter, International Brain Laboratory, E. L. Dyer, C. Chandrasekaran, N. A. Steinmetz, L. Paninski, and C. Hurwitz, "In vivo cell-type and brain region classification via multimodal contrastive learning," bioRxiv, Apr. 2025, bioRxiv 2024.11.05.622159.

[10] H. Cui et al., "BrainGB: A Benchmark for Brain Network Analysis With Graph Neural Networks," *IEEE Trans. Med. Imaging.*, vol. 42, no. 2, pp. 493–506, Feb. 2023.

[11] B.-H. Kim, J. Choi, E. Yun, K. Kim, X. Li, and J. Lee, "Learning Dynamic Brain Connectome with Graph Transformers for Psychiatric Diagnosis Classification," in *Proc. 2024 IEEE Int. Symp. Biomed. Imaging (ISBI)*, 2024, pp. 1–5.

[12] Keqi Han, Yao Su, Lifang He, Liang Zhan, Sergey Plis, Vince Calhoun, and Carl Yang, "Rethinking functional brain connectome analysis: Do graph deep learning models help?," *arXiv preprint arXiv:2501.17207*, 2025.

[13] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," arXiv preprint arXiv:1609.02907, 2017.

[14] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs," *Adv. Neural Inf. Process. Syst.*, vol. 30, pp. 1146–1155, 2017.

[15] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, et al., "Graph attention networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.

[16] C. Ying, T. Cai, S. Luo, S. Zheng, G. Ke, D. He, Y. Shen, and T.-Y. Liu, "Do Transformers Really Perform Badly for Graph Representation?," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 28877–28888, 2021.

[17] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, 2019, pp. 2623–2631, ACM.

[18] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Adv. Neural Inf. Process. Syst.* 2011, vol. 24, pp. 2559–2567, Curran Associates, Inc.