Graph-affiliated Unsupervised Segmentation Assisted Simple Neural Network

***Abstract*—High-throughput 16S rRNA-seq data, complemented by detailed cultivation information, constitutes a critical resource in bacterial research with promising implications for biomedical applications such as fecal microbiota transplantation. However, the inherent high dimensionality and substantial costs associated with 16S rRNA-seq data constrain its full utility. In this paper, we introduce a novel approach—graph-affiliated unsupervised segmentation-assisted simple neural network (GASNN)—designed to analyze 16sRNA-seq data efficiently. In a proof-of-concept application involving the prediction of cultivation media temperature, the GASNN model achieved significant performance enhancements over a traditional simple neural network (SNN). Further experiments across various tasks consistently demonstrated that GASNN improves the performance of SNN models. Nevertheless, a notable limitation of the proposed approach is that its benefits may diminish as the network architecture deepens, thereby impeding its ability to reveal the intrinsic manifold structure of the data.**

***Index Terms*—16S rRNA-seq, bacterial cultivation, unsupervised segmentation, manifold learning**

I. Introduction

THE technique of 16S ribosomal RNA (16S rRNA) sequencing is a cornerstone technique in microbial ecology and molecular microbiology, offering critical insights into the structure and function of microbial communities. The 16S rRNA gene is highly conserved across bacterial species, making it an ideal molecular marker for the identification and phylogenetic analysis of bacteria. Its universal presence in all bacteria allows for the classification of microbes in a given sample without the need for culturing, which can be a time-consuming and selective process. Consequently, 16S rRNA sequencing has become a powerful tool in microbial diversity studies and an essential component of metagenomic approaches to understanding microbial ecosystems [8], [19].

The application of 16S rRNA sequencing extends far beyond simple bacterial identification. It is now a foundational method for microbial community profiling, allowing researchers to uncover the diversity of bacterial species in various environmental niches, including soil, water, and the human gut. The ability to sequence and catalog these communities at high throughput has enabled significant advances in microbial ecology, including the discovery of novel species and the characterization of complex microbial interactions [2]. In clinical settings, 16S rRNA sequencing has also facilitated the analysis of the human microbiome, revealing its association with health conditions and providing insights into the therapeutic potential of microbiome-based interventions, such as fecal microbiota transplantation (FMT) [2].

Despite the successes of 16S rRNA sequencing in bacterial identification and community profiling, challenges remain in its application for more complex tasks, such as predicting cultivation conditions. While species-level classification from 16S rRNA sequences has achieved remarkable accuracy, as high as 90% in some studies, predicting cultivation parameters like media temperature based solely on 16S rRNA data is an area that remains underexplored [9]. Nevertheless, the growing availability of databases such as DSMZ and publicly released datasets from NIH provide a rich source of cultivation media information that, when integrated with 16S rRNA sequencing data, holds the potential to enable novel predictive models in microbial cultivation [19]. Such advancements could have significant implications for optimizing bacterial growth conditions and improving the scalability of microbial applications, particularly in fields like biotechnology and personalized medicine.

In this study, we explore the application of machine learning techniques for predicting cultivation media temperature from 16S rRNA sequencing data. To this end, we begin by investigating traditional methods, including Random Forest and Multi-Layer Perceptron (MLP) models. In our initial regression experiments, we observe that a Random Forest model yields an R^2 value of approximately 0.55, indicating moderate predictive accuracy for media temperature. Similarly, a MLP with three hidden layers (3LP) achieves a similar R^2 of around 0.5, which is not satisfactory for practical applications.

However, when we introduce a novel approach—combining the feature data with multiple clustering configurations, each optimized at different resolutions—we observe a remarkable improvement. These multiple configurations are obtained through a process involving k-Nearest Neighbors (kNN), stochastic graph t-SNE (SGtSNE), and multi-configuration optimized clustering (BlueRed) [16]. When this series of transformations is added as an unsupervised layer to the 3LP model, the regression task's performance dramatically improves, achieving an R^2 of approximately 0.85. This significant enhancement suggests that the combination of feature data and the diverse clustering configurations allows the model to capture the intrinsic manifold structure of the data, leading to more accurate predictions of media temperature at an early stage of the neural network's processing pipeline.

To assess the generalizability of this approach, we apply the graph-affiliated unsupervised segmentation-assisted simple neural network (GASNN) to other well-established benchmarking tasks. In the MNIST dataset, the introduction of the unsupervised layer leads to an impressive clustering result, which is directly linked to the improvement in classification

J. Wang, S. Xu are with the Division of Natural and Applied Sciences, Duke Kunshan University, Kunshan, China e-mail: shixin.xu@dukekunshan.edu.cn
R. Guo is with Boston University, Boston, MA, USA
Manuscript received ???, 2025; revised ???, 2025.

accuracy [4]. Although the results on the ImageNet dataset are less visually interpretable, we observe the robustness of the model, contributing to improved performance [3]. This suggests that the ability of GASNN to incorporate unsupervised graph-based features may have broad applicability, not only for microbial cultivation predictions but also for other complex, high-dimensional tasks.

II. Methodology

A. Feature Data Preprocessing

Our dataset was acquired from two primary sources—DSMZ and NIH—and includes extensive information on bacterial strains, cultivation conditions, and associated metadata such as temperature and 16S rRNA sequences. While the database itself contains multiple tables (e.g., **STRAINS**, **MEDIA**, **SOLUTIONS**, **INGREDIENTS**, **STEPS**, and **GAS**) with fields covering everything from taxonomic classification to specific cultivation protocols, we focus on two critical attributes for this study: the 16S rRNA gene sequences and the cultivation media temperature. The data was obtained through publicly available records provided by DSMZ [5] and NIH [14].

In total, we identified approximately 65,000 records where both 16S rRNA data and cultivation media temperature values were available. Within these records, the temperature values typically range from psychrophilic to thermophilic conditions, and were normalized to maintain consistency in subsequent modeling steps. The 16S rRNA sequences are strings of varying lengths (primarily between 500 and 1,500 nucleotides), reflecting the heterogeneity of bacterial strains in the database.

To transform the raw 16S rRNA sequences into a feature vector suitable for machine learning models, we employed a k-mer-based approach [6], [11], which is commonly used in genomic analyses. The k-mer method quantifies the frequency of all possible substrings (k-mers) of length k in a sequence. For a given sequence $S = \{s_1, s_2, \ldots, s_N\}$, where each $s_i \in \{A, T, C, G\}$ for nucleotide sequences, the frequency of each unique k-mer m is counted as:

$$f_m = \text{count}(m, S),$$

where $\text{count}(m, S)$ is the number of times the k-mer m appears in the sequence S. The feature vector for each sequence is composed of the counts of each possible k-mer in the sequence. The k-mer frequency matrix for the entire dataset is constructed by stacking the feature vectors of all individual sequences.

This transformation results in a high-dimensional feature matrix, where each sequence is represented by a vector of k-mer frequencies. This approach, while effective for capturing sequence composition, can sometimes lead to a loss of positional information, which is an inherent tradeoff when using k-mer based methods.

Before proceeding to model training, we performed several quality control steps. First, we removed any rows containing missing or corrupted values. We then discarded any features that exhibited zero variance, as they provide no discriminatory power for the models. After these cleaning steps, we were left with 9,862 samples, each described by 4,216 features. Recognizing that many remaining features were still redundant or sparsely informative, we further applied a univariate feature selection strategy to retain only the top 1,000 features for training purposes. This step not only reduces computational overhead but also helps mitigate the risk of overfitting.

To ensure all features were on a comparable scale, we employed a standard scaling process [15] that transforms each feature by subtracting the mean and dividing by the standard deviation:

$$x_{\text{scaled}} = \frac{x - \mu}{\sigma},$$

where x is the original feature value, μ is the mean of that feature across all samples, and σ is its standard deviation. Finally, we split our dataset into a training set (80%) and a test set (20%) to facilitate performance assessment and minimize the risk of overfitting. This comprehensive preprocessing pipeline—covering data cleaning, dimensionality reduction, and normalization—ensured that the subsequent modeling steps were built on a robust and consistent feature representation of the 16S rRNA sequences and their corresponding temperature values.

B. Graph-Affiliated Unsupervised Segmentation

We employ a graph-affiliated unsupervised segmentation approach to enhance the feature data representation. The steps involved in this process are k-Nearest Neighbors (kNN) graph construction, stochastic graph t-SNE (SGtSNE) embedding, and modularity-based clustering with a resolution parameter. Specifically, we adopt the Leiden algorithm for community detection and draw on a multi-configuration methodology inspired by the unpublished "BlueRed" framework developed by Xiaobai Sun et al. (2025) at Duke University. By leveraging these techniques together, we obtain an enriched set of cluster labels at multiple resolution scales, thereby augmenting the original feature data for subsequent machine learning tasks.

1) Step 1: k-Nearest Neighbors (kNN): We begin by constructing a k-nearest neighbors (kNN) graph from the 16S rRNA feature data, where each data point is connected to its k nearest neighbors based on a specified distance metric. We use kNN graphs for non-uniformly distributed data and latent-manifold structures, where a modest integer value for k is sufficient to capture the variance, maintain local connectivity, and allow information to propagate along the geodesic path. This approach avoids overly dense or disconnected graphs that might arise from other types of graph construction, such as rNN graphs.

2) Step 2: Stochastic Graph t-SNE (SGtSNE): Next, we apply stochastic graph t-SNE (SGtSNE), a variation of t-SNE that operates directly on graph-based data. This method is especially useful for preserving local similarities and capturing the manifold structure of the data. The conversion from distance $d(x, y)$ to the edge weight $w(x, y)$ in SGtSNE is done as follows:

$$w(x, y) = \frac{1}{\lambda} \exp\left(-\frac{d^2(x, y)}{2\sigma_x^2}\right), \quad x, y \in X, (x, y) \in E(G)$$

where λ is a normalization parameter that adjusts the weight range. The non-linear scaling with σ_x is chosen to be adaptive and is determined by the following sparse equation:

$$\sum_{y:(x,y)\in E} \exp\left(-\frac{d^2(x,y)}{2\sigma_x^2}\right) = \lambda, \quad x \in X$$

This ensures that the weights are appropriately scaled. After these transformations, the adjacency matrix is column-stochastic. SGtSNE provides a lower-dimensional embedding that preserves the neighborhood structure of the data, allowing for more effective clustering in the next steps.

3) Step 3: Leiden Algorithm with a Resolution Parameter: We then perform community detection using the Leiden algorithm [1], [18], which introduces a resolution parameter γ. The clustering quality is measured by a modularity function that balances internal connectivity within communities against external connectivity. In a Hamiltonian formulation, the corresponding energy function $H(\mathbf{S})$ can be expressed as:

$$H(\mathbf{S}) = \sum_{i,j\in V} A_{ij}\,\delta(S_i, S_i) - \gamma \sum_{i,j\in V} A_{ij}\,\delta(S_i, S_j),$$

where A_{ij} is the weight of the edge between nodes i and j, $\delta(S_i, S_j)$ is the Kronecker delta function (1 if i and j belong to the same community, 0 otherwise), and γ is the resolution parameter controlling the granularity of the detected communities.

4) γ-Transformation and Multi-Configuration Methodology: Inspired by the "BlueRed" framework (Xiaobai Sun et al., 2025, unpublished), we consider multiple configurations of the Leiden algorithm by varying γ systematically, rather than relying on a single fixed resolution. To handle the resolution problem more smoothly, we map $\gamma \in [0, \infty)$ to a bounded parameter $\theta \in [0, 1]$ using a sigmoid transformation:

$$\theta(\gamma) = \frac{1}{1 + e^{-\alpha(\gamma-\beta)}},$$

where α and β are parameters that control the shape of the transformation. This mapping helps avoid unbounded parameter sweeps and provides a more stable range for identifying meaningful clusters.

Moreover, to prioritize finer community structures during optimization, we may employ a descending triangular weighting function $w(\theta)$ defined by:

$$w(\theta) = \begin{cases} 1 - 2\theta & \text{if } 0 \le \theta \le 0.5, \\ 2\theta - 1 & \text{if } 0.5 < \theta \le 1. \end{cases}$$

The combined approach yields multiple stable clustering solutions by scanning different values of θ. These solutions collectively form a multi-configuration result set, enabling deeper insights into the data's hierarchical structure.

5) Overall Process and Benefits: By integrating these three steps—kNN graph construction, SGtSNE embedding, and modularity-based clustering with a resolution parameter (implemented via Leiden and further extended by the "BlueRed" multi-configuration concept)—we obtain a powerful feature augmentation process. This methodology enriches the subsequent machine learning models by uncovering latent manifold structure and providing multiple, complementary cluster assignments. As a result, the models become more robust and accurate in tasks such as predicting cultivation media temperature, benefiting from the enhanced representation and multi-resolution perspective offered by the proposed pipeline.

C. Model Architecture

In this section, we describe the core predictive model that builds upon the multi-configuration clustering outputs generated by the graph-affiliated unsupervised segmentation (Section II-B). Our primary objective is to predict cultivation media temperature from 16S rRNA-derived features, augmented by the multiple cluster assignments. To achieve this, we employ a simple multi-layer perceptron (MLP) comprising three fully connected layers (3LP). Figure 1 provides a high-level schematic of the complete pipeline, where the augmented feature data serve as input to the 3LP.

1) Input Augmentation with Multi-Configuration Clusters: Let $\mathbf{x} \in \mathbb{R}^d$ be the original feature vector (e.g., the k-mer counts or any preprocessed representation of the 16S rRNA data). From the graph-affiliated unsupervised segmentation, we obtain a set of multiple cluster assignments,

$$\{\omega_1, \omega_2, \ldots, \omega_{|\Omega|}\},$$

where ω_i corresponds to the i-th clustering configuration, and $|\Omega|$ is the total number of configurations. Each configuration ω_i assigns a cluster label to every sample. We treat these numeric cluster labels as additional features (after standardizing each configuration independently), resulting in an augmented feature vector

$$\mathbf{z} \;=\; \big(\mathbf{x}^\top,\, \omega_1,\, \omega_2,\, \ldots,\, \omega_{|\Omega|}\big)^\top \;\in\; \mathbb{R}^{\,d+|\Omega|}.$$

Although one might consider one-hot encoding of the cluster labels, our approach appends the numeric labels directly, relying on the MLP to learn appropriate transformations through its trainable weights. This design choice preserves potential hierarchical relationships among different configurations and simplifies the model input.

2) Three-Layer Perceptron (3LP) Design: We opt for a simple neural network with three fully connected layers (3LP), each followed by a Rectified Linear Unit (ReLU) activation [13] and a Dropout layer [17] to mitigate overfitting. Concretely, the layer dimensions are:

$$\text{(1) Input Layer: } d + |\Omega| \;\longrightarrow\; 256,$$

$$\text{(2) Hidden Layer: } 256 \;\longrightarrow\; 128,$$

$$\text{(3) Hidden Layer: } 128 \;\longrightarrow\; 64,$$

and finally an output neuron (or small output layer) for regression. Each fully connected layer can be written in the form:

$$\mathbf{h}^{(l)} \;=\; \mathrm{ReLU}\Big(\mathbf{W}^{(l)}\,\mathbf{h}^{(l-1)} \;+\; \mathbf{b}^{(l)}\Big),$$

where $\mathbf{h}^{(l-1)}$ denotes the output from the preceding layer (or the augmented input $\mathbf{z}$ for $l = 1$), $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ are trainable

weight matrices and bias vectors, respectively. After each ReLU activation, a Dropout function $\mathrm{Dropout}(p)$ randomly zeroes out a fraction p of the units to reduce co-adaptation of feature detectors [17]:

$$\mathbf{h}_{\text{drop}}^{(l)} \;=\; \mathbf{m} \odot \mathbf{h}^{(l)}, \quad \mathbf{m}_i \sim \mathrm{Bernoulli}(1-p),$$

where $\odot$ denotes element-wise multiplication and $\mathbf{m}$ is a random mask vector with entries drawn from the Bernoulli distribution.

3) Regression Output and Training Objective: For media temperature prediction, the final (fourth) layer is a single linear neuron:

$$\hat{y} \;=\; \mathbf{w}^{\top}\, \mathbf{h}_{\text{drop}}^{(3)} + b,$$

where $\mathbf{h}_{\text{drop}}^{(3)}$ is the output of the third Dropout layer, $\mathbf{w}$ and b are the trainable parameters of the output layer, and $\hat{y}$ is the predicted temperature value. We train the model by minimizing the Mean Squared Error (MSE) loss:

$$\mathcal{L}_{\text{MSE}}(\hat{y}, y) \;=\; \frac{1}{N}\sum_{i=1}^{N}\big(\hat{y}_i - y_i\big)^2,$$

where y_i is the ground truth temperature for sample i, $\hat{y}_i$ is the model prediction, and N is the number of training samples. Optimization is performed using standard gradient-based methods (e.g., Adam [10]), and early stopping or additional regularization can be employed to prevent overfitting.

4) Overall Architecture and Benefits: Figure 1 shows a simplified depiction of our pipeline. The lower portion illustrates the graph-affiliated unsupervised segmentation, which generates multiple cluster configurations that are appended to the original feature vectors. The upper portion shows the three-layer perceptron, which processes the augmented input and outputs the predicted temperature. By providing the 3LP with cluster-based features at various resolutions, the model can capture latent manifold structures that might otherwise remain hidden in raw 16S rRNA-derived features. Despite its simplicity, this design already yields a significant improvement over baseline models (e.g., single-layer MLPs or Random Forests), demonstrating the efficacy of augmenting the original feature space with multi-configuration clustering information.

D. Evaluation Metrics

Our primary task is the regression-based prediction of cultivation media temperature from 16S rRNA data, and we therefore employ the coefficient of determination (R^2) as the principal metric for model evaluation. In addition to regression, we also investigate the performance of our method in a classification setting. In this context, we visualize the classification results using confusion matrix heatmaps and quantify accuracy with a series of standard metrics, including Overall Accuracy (OA), Average Accuracy (AA), and the Adjusted Rand Index (ARI).

1) Regression: Coefficient of Determination (R^2): For regression tasks (e.g., predicting media temperature), we use the coefficient of determination:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}\big(y_i - \hat{y}_i\big)^2}{\sum_{i=1}^{N}\big(y_i - \bar{y}\big)^2},$$

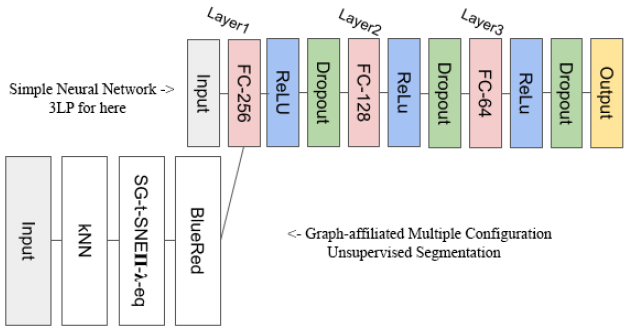

Fig. 1. Illustration of the proposed architecture. The *Simple Neural Network (3LP in this case)* is shown at the top, with three fully connected layers (256, 128, and 64 neurons, respectively), each followed by ReLU and Dropout. The *Graph-Affiliated Multiple-Configurations Unsupervised Segmentation* is depicted at the bottom, providing multiple clustering configurations that, once appended to the original feature vectors, serve as the input to the 3LP. This pipeline enriches the model's input representation and helps to capture latent manifold structures in the data.

where y_i is the ground truth for sample i, $\hat{y}_i$ is the corresponding model prediction, $\bar{y}$ is the sample mean of the target variable, and N is the total number of samples. An R^2 value of 1.0 indicates a perfect fit, whereas values close to 0 suggest that the model performs similarly to a simple mean-based predictor.

2) Classification: Confusion Matrix and Accuracy Metrics: To evaluate classification performance, we generate a confusion matrix heatmap that visualizes how the model's predictions compare with the true labels. Each row of the matrix corresponds to the actual class, and each column corresponds to the predicted class. From this matrix, we derive the following metrics:

- **Overall Accuracy (OA):** The proportion of correctly classified samples out of the total number of samples:

$$\mathrm{OA} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{1}\big(\hat{y}_i = y_i\big),$$

  where $\mathbf{1}(\cdot)$ is the indicator function that equals 1 when the condition is true and 0 otherwise.
- **Average Accuracy (AA):** The mean of the per-class accuracies, placing equal emphasis on each class, regardless of class size. For C classes, let acc_c be the accuracy for class c. Then,

$$\mathrm{AA} = \frac{1}{C}\sum_{c=1}^{C}\mathrm{acc}_c.$$

3) Adjusted Rand Index (ARI): The Adjusted Rand Index (ARI) [7] is a popular measure for assessing similarity between two clusterings (or label assignments). It corrects the Rand Index (RI) for chance grouping of elements. Let n_{ij} denote the number of elements that are assigned to cluster i in the ground truth and to cluster j in the predicted assignment. Let $a_i = \sum_j n_{ij}$ be the total number of elements in cluster i (ground truth), and $b_j = \sum_i n_{ij}$ the total number in cluster j (predicted). The ARI can be expressed as:

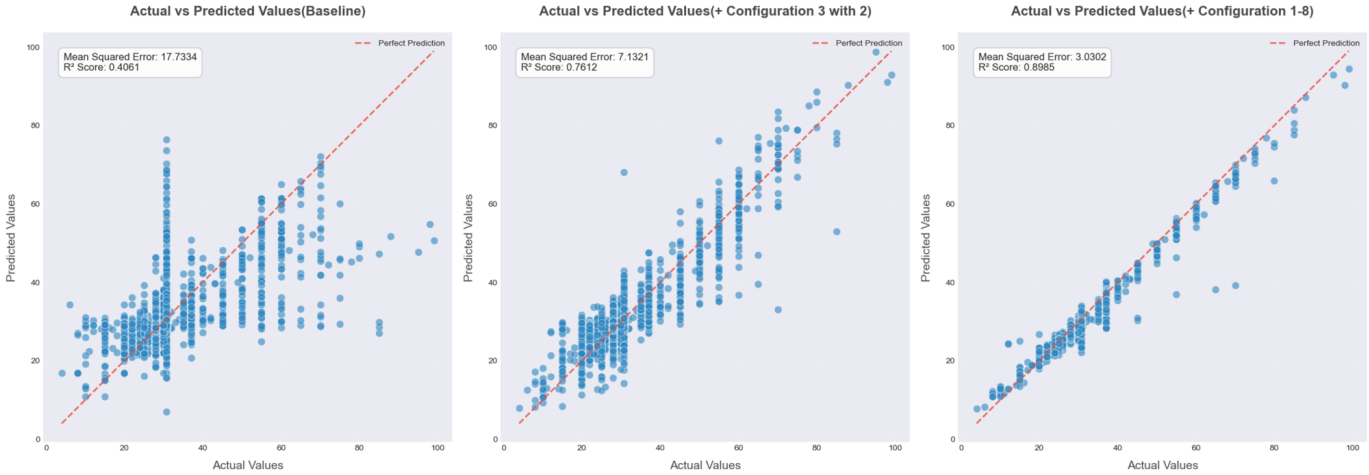

Fig. 2. Comparison of temperature prediction performance across models. From left to right: baseline Multi-Layer Perceptron (3LP), GASNN-RMS, and GASNN-combined. Each plot shows predicted versus true temperature values on the test set, with the diagonal line representing perfect predictions.

$$\text{ARI} = \frac{\sum_{i,j}\binom{n_{ij}}{2} - \frac{\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}}{\binom{N}{2}}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \frac{\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}}{\binom{N}{2}}},$$

where $\binom{n}{2} = \frac{n(n-1)}{2}$, N is the total number of elements (samples), and the denominator normalizes for both the range of the index and random labeling effects. The ARI ranges from -1 to 1, where 1 indicates a perfect match between predicted and true labels, and values near 0 indicate random labeling.

4) Interpretation and Comparison: The choice of metric depends on the nature of the task. For the media temperature prediction, we rely on R^2 to assess how well the model captures variance in the target variable. In potential classification tasks (e.g., if we transform temperature ranges into discrete categories or test on standard datasets like MNIST), we supplement our evaluation by examining the confusion matrix and reporting metrics such as OA, AA, and ARI. This suite of metrics allows for a comprehensive view of model performance across both regression and classification paradigms.

III. Experiments and Results

A. Regression on Cultivation Media Temperature

1) Dataset Description: The DSMZ 16S rRNA dataset comprises sequences ranging from approximately 500 to 1500 nucleotides in length. After applying k-mer transformation with k=4 to the 16S rRNA sequences, the dataset was reduced to dimensions of (65,023, 29,071). We performed a train-test split with 20% allocated for testing, resulting in 52,018 training samples and 13,005 test samples. The data was normalized using a robust scaler, followed by variance filtering and feature selection to retain the 1,000 most informative features.

Our experimental evaluation focused on predicting optimal temperature values for cultivation media. Table I presents a comprehensive performance comparison across different

TABLE I
Performance Comparison of Models for Temperature Prediction

Model	R^2 Score	Mean Squared Error
Random Forest	0.39	17.75
Multi-Layer Perceptron (3LP)	0.41	17.73
GASNN-RMS	0.76	7.13
GASNN-combined	**0.90**	**3.03**

models. The baseline models, Random Forest and Multi-Layer Perceptron (3LP), demonstrated moderate performance with R^2 scores of 0.39 and 0.41, respectively. Our proposed GASNN variants exhibited substantial improvements, with GASNN-RMS achieving an R^2 score of 0.76 and GASNN-combined further excelling with a score of 0.90. The mean squared error showed marked improvement, decreasing from approximately 17.7 for baseline models to 3.03 for GASNN-combined.

Figure 2 illustrates the comparative prediction performance across models through scatter plots of predicted versus true temperature values on the test set. Points aligned closer to the diagonal line indicate higher prediction accuracy. The visualization demonstrates a clear progression in prediction quality from left to right, with GASNN-combined exhibiting the most concentrated clustering around the ideal prediction line, substantiating its superior performance.

2) Progressive Feature Integration Analysis: To understand the contribution of multiple clustering configurations in GASNN-combined, we conducted an ablation study by progressively incorporating clustering configurations into the feature set. Figure 3 illustrates how the prediction error (MSE) changes as additional configurations are integrated. The introduction of the first configuration results in a dramatic reduction in MSE, demonstrating the significant impact of initial graph-based features. Subsequent configurations show a generally decreasing trend in prediction error, albeit with some fluctuations, suggesting complex interactions between different clustering configurations. While the overall trend indicates

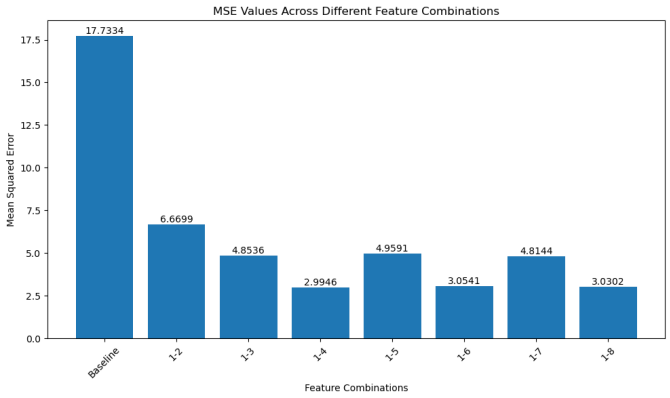

Fig. 3. Progressive change in prediction performance (measured by Mean Squared Error) as additional clustering configurations are integrated into GASNN-combined. The overall decreasing trend with fluctuations demonstrates the complex interaction between multiple graph-based feature representations.

TABLE II
PERFORMANCE COMPARISON OF MODELS FOR MNIST CLASSIFICATION

Model	Accuracy
Random Forest	0.9692
Multi-Layer Perceptron (3LP)	0.9690
GASNN-RMS	**0.9890**

improved performance with more configurations, the non-monotonic nature of the improvements highlights the intricate relationship between different graph-based representations and their collective impact on prediction accuracy.

B. Benchmarking Tasks

1) MNIST Classification: To validate our approach on a standard benchmark dataset, we evaluated GASNN on the MNIST handwritten digit classification task [12]. The MNIST dataset consists of 70,000 28x28 grayscale images of handwritten digits (0-9), with 60,000 training images and 10,000 test images. We flattened each image into a 784-dimensional vector and normalized the pixel values to [0,1].

Table II shows the classification accuracy of different models on the MNIST test set. Both baseline models achieved similar performance, with Random Forest and 3LP reaching approximately 96.9% accuracy. GASNN-RMS demonstrated superior performance with 98.9% accuracy, representing a significant improvement over the baseline methods.

Figure 4 presents a comparison of confusion matrices across different approaches. The leftmost matrix shows the clustering structure discovered through unsupervised segmentation, demonstrating the layer's ability to identify natural groupings in the data without any prior knowledge of the digit classes. While the segmentation effectively captures most digit classes, we observe oversplitting in the cases of digits 1 and 9, with the former case being particularly non-trivial and warranting further discussion in the next section. The middle matrix displays the baseline 3LP performance, which achieves respectable accuracy. The rightmost matrix shows GASNN-RMS predictions, characterized by strong diagonal elements that indicate excellent classification accuracy across all digit classes. The scarcity of off-diagonal elements, even for traditionally challenging digit pairs such as 4-9, demonstrates that GASNN-RMS effectively utilizes the learned data structure to disambiguate similar cases. These visualizations confirm that our approach not only enhances overall accuracy but also maintains consistent performance across the full range of digit classes.

IV. DISCUSSION

A. Insights from the Results

Our experimental results demonstrate several key insights about GASNN's capabilities and advantages. First, GASNN significantly improves the performance of simple neural networks like 3LP, as evidenced by the substantial increase in R^2 scores from 0.41 to 0.90 for temperature prediction and classification accuracy from 96.9% to 98.9% on MNIST. This improvement is particularly notable in scenarios with limited training data or computational resources.

The success of GASNN can be attributed to two main factors. First, as visualized in Figure 5, the unsupervised segmentation layer effectively exposes the underlying manifold structure of the data to the predictor at an early stage. This is demonstrated by the clear clustering patterns in the t-SNE visualization, where digits naturally separate into distinct regions despite no class labels being used in the segmentation process.

Second, while the unsupervised segmentation may occasionally produce imperfect clusterings (such as the oversegmentation of digit '1' seen in Figure 5), the subsequent neural network effectively learns to handle these cases. This is evidenced by the high classification accuracy achieved despite such segmentation artifacts. The neural network appears to leverage the multiple configuration views provided by GASNN to resolve ambiguities and merge oversegmented clusters when appropriate.

An interesting observation is that GASNN-combined, which involves test data during the segmentation process, achieves notably higher performance (R^2 of 0.90) compared to GASNN-RMS (R^2 of 0.76) which segments training and test data separately. While this may seem to challenge conventional wisdom about test data isolation, it's important to note that GASNN-RMS still provides significant improvements over baseline models without requiring test data access. The performance gap between the two variants suggests that joint segmentation in GASNN-combined allows better capture of the full data manifold structure, though at the cost of requiring test data during the RMS process. GASNN-RMS represents a more practical deployment scenario, where new samples must be processed independently, while still leveraging the benefits of manifold-aware feature augmentation.

The temperature prediction results further validate these insights in a regression context. The dramatic improvement in R^2 score and reduction in mean squared error suggest that GASNN's ability to capture data structure generalizes well beyond classification tasks. This is particularly impressive given the complexity of the relationship between 16S rRNA sequences and optimal cultivation temperature values.

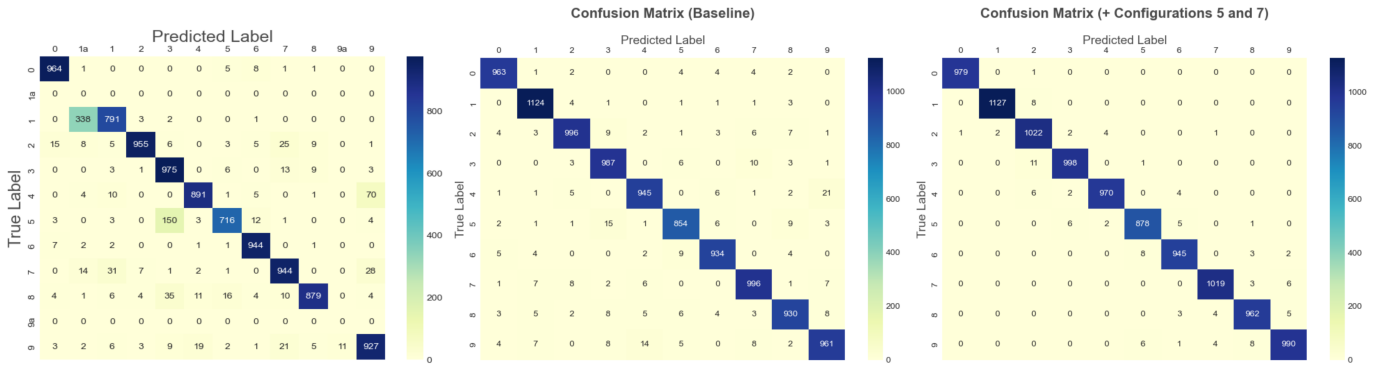

Fig. 4. Comparison of confusion matrices for MNIST digit classification. The leftmost matrix shows the unsupervised segmentation results, the middle shows the baseline 3LP model performance, and the rightmost shows GASNN-RMS predictions. Strong diagonal patterns indicate accurate classifications, while off-diagonal elements reveal misclassification patterns between digit pairs.

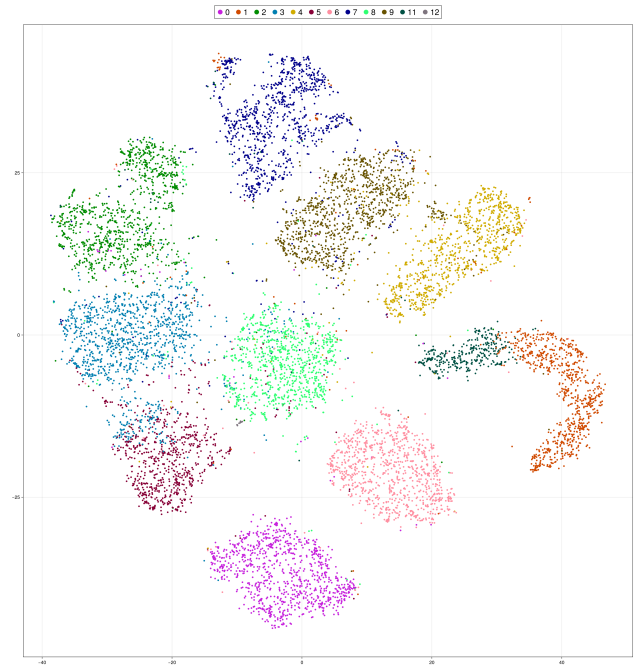

Fig. 5. t-SNE visualization of a selected configuration from the unsupervised segmentation layer for MNIST classification. The high-dimensional data is projected into 2D space, with each point representing a digit and colors indicating cluster assignments. Note the oversegmentation of digit '1', where a single digit class is split into multiple clusters - a phenomenon discussed in Section II-D.

B. Potential Applications

The demonstrated success of GASNN suggests several promising applications, particularly in domains where data exhibits strong manifold structure or where labeled data is expensive to obtain. In the context of 16S rRNA analysis, GASNN could significantly advance selective cultivation of beneficial microorganisms. For example, in fecal microbiota transplantation (FMT) therapy, accurately predicting optimal growth conditions could improve the isolation and preservation of therapeutic bacterial strains. Similarly, in industrial fermentation, GASNN could help optimize cultivation conditions for producing specific metabolites or enzymes, reducing the extensive trial-and-error typically required.

Beyond microbiology, the method is especially valuable in scenarios where obtaining ground truth labels requires significant expertise or resources. For instance, in materials science, determining the properties of new compounds often requires extensive laboratory testing. GASNN could leverage the natural clustering of material structures to improve predictions while requiring fewer labeled examples. Similarly, in pharmaceutical development, where testing drug efficacy requires expensive clinical trials, the model could utilize the inherent patterns in molecular structures to enhance prediction accuracy with limited training data.

The multi-configuration approach could also benefit domains with inherent hierarchical structures. In medical imaging, tissue samples often have natural organizational levels (cells, tissues, organs) that could benefit from multiple clustering views. Similarly, in remote sensing, land cover classification involves natural hierarchies of terrain features that could be better captured through GASNN's architecture. These applications particularly benefit from GASNN's ability to capture multiple levels of data organization while requiring minimal labeled examples, making it especially valuable when comprehensive labeling is prohibitively expensive or time-consuming.

C. Limitations and Future Work

Despite its promising results, GASNN faces several key limitations. A primary challenge lies in constructing meaningful manifold structures from feature data to graph data, which requires sufficient sample variance to establish connections between data points. While this can be partially addressed by increasing the k parameter in k-nearest neighbors, an excessively high k value may create artificial or meaningless connections that do not reflect true data relationships. To mitigate this limitation, feature augmentation techniques could be explored, particularly in computer vision and imaging applications where additional contextual or transformed features might help establish more robust connections.

Another significant limitation emerges when considering deeper neural networks or architectures with more parameters. While GASNN demonstrably improves the performance of simple neural networks, this benefit may not extend to more

complex models. In fact, deeper networks may already be capable of learning the manifold structures that GASNN's unsupervised segmentation layer aims to capture. In some cases, the additional layer of graph-based feature extraction might even interfere with the network's natural ability to discover hierarchical representations, potentially degrading performance. This limitation requires careful consideration when applying GASNN to problems where deep learning architectures are standard, and we acknowledge that comprehensive testing with deeper networks remains an important area for future investigation.

These limitations suggest that GASNN's effectiveness may be most pronounced in specific scenarios: when working with simpler neural architectures, when the underlying data has clear but complex manifold structure, and when sufficient samples are available to construct meaningful graph representations. Understanding these boundaries and constraints is crucial for appropriately applying GASNN in practical settings.

V. Conclusion

A. Summary of Contributions

In this paper, we introduced GASNN, a novel approach that combines graph-affiliated unsupervised segmentation with simple neural networks to improve the analysis of 16S rRNA-seq data. Our key contributions include:

- Development of a graph-based feature augmentation technique that captures the intrinsic manifold structure of high-dimensional biological data through multiple clustering configurations
- Demonstration of significant performance improvements in predicting cultivation media temperature from 16S rRNA sequences, achieving an R^2 of 0.90 with GASNN-combined and 0.76 with GASNN-RMS, compared to baseline models
- Validation of the method's generalizability through successful application to standard machine learning benchmarks, including MNIST classification
- Introduction of two GASNN variants (GASNN-combined and GASNN-RMS) that offer different trade-offs between performance and practical deployment considerations

These contributions advance both the theoretical understanding of manifold-aware feature augmentation and its practical application in microbial cultivation optimization.

B. Future Directions

Several promising avenues for future research emerge from this work:

- **Enhanced Graph Construction:** Investigation of alternative methods for constructing meaningful graph representations from feature data, particularly in cases with limited sample variance or sparse connectivity
- **Integration with Deep Architectures:** Exploration of how GASNN's principles can be adapted to complement deeper neural networks without interfering with their inherent representation learning capabilities
- **Extended Bio Applications:** Application of GASNN to other biological prediction tasks, such as optimal pH levels, nutrient requirements, or growth rates for bacterial cultivation
- **Theoretical Framework:** Development of a formal theoretical framework to better understand the relationship between manifold structure, multiple clustering configurations, and prediction performance
- **Real-time Adaptation:** Investigation of methods to dynamically update the graph structure and clustering configurations as new data becomes available, enabling continuous learning in practical applications

These future directions aim to address current limitations while expanding the utility of GASNN across broader application domains.

Appendix A
Proof of the First Zonklar Equation

Appendix one text goes here.

Appendix B

Appendix two text goes here.

Acknowledgment

The authors would like to thank...

References

[1] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. "Fast Unfolding of Communities in Large Networks". In: *Journal of Statistical Mechanics: Theory and Experiment* 2008.10 (Oct. 2008), P10008.

[2] G. Cammarota, G. Ianiro, and A. Gasbarrini. "Fecal Microbiota Transplantation for the Treatment of Clostridium Difficile Infection: A Systematic Review". In: *Journal of Clinical Gastroenterology* 48.8 (Sept. 2014), pp. 693–702. pmid: 24440934.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "ImageNet: A Large-Scale Hierarchical Image Database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009 IEEE Conference on Computer Vision and Pattern Recognition. June 2009, pp. 248–255.

[4] L. Deng. "The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]". In: *IEEE Signal Processing Magazine* 29.6 (Nov. 2012), pp. 141–142.

[5] German Collection of Microorganisms and Cell Cultures GmbH. *DSMZ – German Collection of Microorganisms and Cell Cultures*. 2025.

[6] *How to Apply de Bruijn Graphs to Genome Assembly — Nature Biotechnology*. URL: https://www.nature.com/articles/nbt.2023 (visited on 02/04/2025).

[7] L. J. Hubert and P. Arabie. "Comparing Partitions". In: *Journal of Classification* 2.2–3 (1985), pp. 193–218.

[8] J. M. Janda and S. L. Abbott. “16S rRNA Gene Sequencing for Bacterial Identification in the Diagnostic Laboratory: Pluses, Perils, and Pitfalls”. In: *Journal of Clinical Microbiology* 45.9 (Sept. 2007), pp. 2761–2764. pmid: 17626177.

[9] J. S. Johnson, D. J. Spakowicz, B.-Y. Hong, L. M. Petersen, P. Demkowicz, L. Chen, S. R. Leopold, B. M. Hanson, H. O. Agresta, M. Gerstein, E. Sodergren, and G. M. Weinstock. “Evaluation of 16S rRNA Gene Sequencing for Species and Strain-Level Microbiome Analysis”. In: *Nature Communications* 10.1 (Nov. 6, 2019), p. 5029.

[10] D. P. Kingma and J. Ba. “Adam: A Method for Stochastic Optimization”. In: *arXiv:1412.6980 [cs.LG]* (Jan. 30, 2017). arXiv: 1412.6980 [cs.LG].

[11] *Kraken: Ultrafast Metagenomic Sequence Classification Using Exact Alignments — Genome Biology — Full Text*. URL: https://genomebiology.biomedcentral.com/articles/10.1186/gb-2014-15-3-r46 (visited on 02/04/2025).

[12] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. “Gradient-Based Learning Applied to Document Recognition”. In: *Proceedings of the IEEE* 86.11 (Nov. 1998), pp. 2278–2324.

[13] V. Nair and G. E. Hinton. “Rectified Linear Units Improve Restricted Boltzmann Machines”. In: 2010.

[14] *National Institutes of Health (NIH)*. National Institutes of Health (NIH). URL: https://www.nih.gov/ (visited on 02/04/2025).

[15] *Pattern Recognition and Machine Learning — SpringerLink*. URL: https://link.springer.com/book/9780387310732 (visited on 02/04/2025).

[16] N. Pitsianis, A.-S. Iliopoulos, D. Floros, and X. Sun. “Spaceland Embedding of Sparse Stochastic Graphs”. In: *2019 IEEE High Performance Extreme Computing Conference (HPEC)*. 2019 IEEE High Performance Extreme Computing Conference (HPEC). Sept. 2019, pp. 1–8.

[17] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.

[18] V. A. Traag, L. Waltman, and N. J. Van Eck. “From Louvain to Leiden: Guaranteeing Well-Connected Communities”. In: *Scientific Reports* 9.1 (Mar. 26, 2019), p. 5233.

[19] Q. Wang, G. M. Garrity, J. M. Tiedje, and J. R. Cole. “Naive Bayesian Classifier for Rapid Assignment of rRNA Sequences into the New Bacterial Taxonomy”. In: *Applied and Environmental Microbiology* 73.16 (Aug. 2007), pp. 5261–5267. pmid: 17586664.

Michael Shell Biography text here.

John Doe Biography text here.

Jane Doe Biography text here.