



OPEN Semantic ECG hash similarity graph

Yixian Fang^{1,3}✉, Shilin Zhang² & Yuwei Ren^{2,3}

Graph-based methods have made significant progress in addressing the dependent correlations among ECG time series variables. However, most existing graph structures primarily focus on local similarity while overlooking global semantic correlation. Additionally, the adjacency matrix is highly susceptible to noise interference, leading to unreliable node connections. In this paper, we present a novel graph generation learning framework that incorporates semantic hash coding to capture the intricate associations both within and between ECG signals, thereby significantly enhancing the retrieval efficiency of subsequent graph-based deep learning models. Specifically, the Semantic Hash Similarity Graph (SHSG) initially leverages the similarity within the label space to generate a hash representation for the supervised signal. Subsequently, a lightweight linear hash function is utilized to produce a hash representation for the unseen signal. Thereafter, a comprehensive global hash dictionary is systematically constructed. Finally, the graph topology is meticulously assembled by leveraging Hamming similarity. Additionally, to ensure the maintenance of semantic similarity, we propose an iterative optimization approach in the orthogonal domain for generating hash representations. To validate the efficacy of the generated graph, we utilized a fast Graph Convolutional Network (GCN) for ECG recognition. The experimental outcomes on multiple publicly available ECG datasets corroborate the robustness and effectiveness of our proposed method.

Keywords ECG, Semantic hash similarity, Graph, Fast GCN

Electrocardiogram (ECG) is a widely employed tool for monitoring cardiovascular diseases in clinical settings. However, the interpretation of ECGs can be subjective and may vary among different observers. Consequently, the integration of computer-assisted diagnostic support can significantly enhance the accuracy and consistency of diagnoses, thereby improving the overall quality of the healthcare system, particularly when combined with the expertise of clinical professionals. Consequently, research on electrocardiogram detection is of paramount importance for the advancement of intelligent medical treatment and has attracted growing attention in the field of machine learning^{1–6}.

ECG records cardiac electrical activity through electrodes on the body surface. Each wave's characteristics provide critical clinical information. Standard ECG beats include the P wave, QRS complex, and T wave, which have temporal and spatial dependencies. Owing to the rapid advancement in computing resources and the availability of extensive experimental datasets, a variety of deep learning models, such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory networks (LSTMs), have been successfully utilized for ECG detection and recognition. However, they struggle to effectively capture complex neighborhood information, which imposes limitations on their ability to parse the structure of the signal. Graph learning, leveraging its capacity to represent complex physical processes and irregular relationships, can enhance the accuracy of ECG signal detection and classification by treating segmentation as nodes.

In recent years, there has been a notable emergence of graph-based methods^{7–10} within the field of ECG signal detection. These approaches have systematically validated the dependencies among signals by modeling variables as nodes in a graph structure and employing graph edges to characterize interactions between variables. This has significantly propelled the development of graph-based ECG anomaly detection technology. Such methods are fundamentally rooted in the framework of graph neural networks (GNNs)¹¹, which have demonstrated remarkable achievements in graph representation learning. However, experimental observations highlight significant limitations in the current graph construction strategies. For example, in our preliminary studies on the AF dataset¹², it was observed that KNN-based adjacency matrices (a) fail to effectively capture long-range semantic correlations, such as misclassifying arrhythmia patterns with similar local features but distinct global morphologies, and (b) demonstrate performance degradation under noisy conditions due to erroneous connections between dissimilar nodes. These issues arise because KNN-based methods:

- Reliance on local proximity leads to the neglect of global label semantics, such as conflating classes that share overlapping local features yet possess distinct clinical meanings.

¹School of Information Engineering, Shandong Management University, Jinan 250357, China. ²School of Information Science and Engineering (Institute of Data Science and Technology), Shandong Normal University, Jinan 250014, China. ³Y. Fang and Y. Ren have contributed equally to this work. ✉email: jiahao218@126.com

- Generate density-biased edges, which may result in redundant connections within high-density regions and insufficient coverage in sparsely represented areas.
- Lack robustness to noise, as the similarity based on Euclidean distance is highly susceptible to disruption by artifacts.

The quality of an initial graph structure is significantly influenced by the connections among its nearest neighboring nodes. However, performing nearest neighbor search on large-scale datasets incurs prohibitively high computational complexity. To address this challenge, hash technology has been recognized as an efficient solution^{13–15}. It employs compact binary codes to represent high-dimensional data and enables low-complexity nearest neighbor search processes via Hamming distance calculations^{16,17}. Furthermore, the binary encoding strategy in hashing inherently possesses the capability to suppress feature noise¹⁸. Drawing inspiration from this, we leverage hashing for semantic reconstruction and introduce a novel graph structure learning framework. This framework defines graph nodes based on semantic hash similarity, rather than exclusively relying on nearest neighbor relationships. Employing semantic hash similarity for constructing graph structures provides several advantages: (1) Compared to the nearest neighbor approach, semantic hashing is more effective in capturing global semantic similarity between nodes, rather than focusing solely on local structural relationships. (2) Hash codes assess similarity through efficient bit operations, such as calculating the Hamming distance, which offers superior computational efficiency compared to traditional methods like Euclidean distance or cosine similarity. (3) Semantic hashing demonstrates robustness against noise and redundant information by effectively filtering out insignificant details when mapping data into a low-dimensional space via a hash function. The primary contributions can be succinctly summarized as follows:

- We propose a concise semantic hash graph learning framework, which initially acquires the hash representation of signals and employs Hamming similarity to alleviate the rigid similarity in the label space, and simultaneously addresses the issue of incomplete or redundant adjacency in similar graphs.
- We propose a hash learning optimization algorithm in richness of supervised learned hash representations. Additionally, it generates hash representations for unseen signals using simple hash functions, constructs a global hash dictionary by combining trained and unseen signal hashes, and establishes a global graph structure based on Hamming similarity.
- By evaluating the generated graph structures using a classic fast GCN, the results demonstrate that the obtained graphs outperform current popular baseline methods, particularly in scenarios involving large intervals.

Related work

Graph-based representation and recognition of ECG signals

Owing to its powerful capability to represent complex data structures, the graph-based approach (GA) has gained significant popularity in the realm of ECG representation and detection. By modeling the interdependencies among signals, this method can uncover the underlying relationship patterns that exist between different signals. This attribute proves to be highly valuable for anomaly detection.

For example, Arora *et al.*⁷ proposed a multimodal graph neural network (GNNs) to capture the intra and inter-ECG dependencies. Yang *et al.*⁸ transformed the ECG from the time domain to the graph domain through Graph Fourier Transform (GFT); then they converted the GFT results of ECG into graph bispectra to extract graph integral bispectra for recognition of arrhythmia. Chen *et al.*⁹ proposed local subgraph learning and signal-enhanced graph learning, which are respectively applied to the learning of electrocardiogram representation and signal-enhanced graph learning. For the multiperceptive region, Chen *et al.*¹⁰ proposed a spatio-temporal graph convolutional contraction network for arrhythmia recognition. Zhang *et al.*¹⁹ introduced a novel spatio-temporal residual GCN for the diagnosis of coronary heart disease by segmenting the ECG signals into small, single-channel blocks and converting these blocks into graph nodes. Iqbal *et al.*²⁰ integrated a double-layer graph convolutional layer with a spatial attention mechanism capture the local and global dependencies, thereby achieving more effective classification of arrhythmias.

The aforementioned methods have improved the detection efficiency of ECG through the use of graph-based approaches. However, the predefined graphs utilized in these methods focus solely on the similarity between neighboring nodes, thereby neglecting global semantic correlations. Additionally, the construction of such graphs involves quadratic complexity, rendering them inefficient for large-scale datasets.

Graph structure learning

The quality of a graph-based model is directly influenced by the quality of the graph. Rapidly constructing a high-quality graph is crucial for advancing graph-based learning. Graph structure learning (GSL)^{21–23} can significantly enhance the quality of graphs by reconstructing or refining the original graph topology, which has recently garnered significant research attention^{24–26}.

The two most popular methods for constructing graphs are as follows: k Nearest Neighbors (kNN graphs)²⁷ and ϵ proximity thresholding (ϵ -graphs)²⁸. A kNN graph is one that connects nodes through K -nearest neighbors, and the later establishes the link by measuring that the similarity of two nodes is less than the preset threshold ϵ . Nevertheless, these two approaches not only entail significant computational complexity but also fail to address the challenge of sparse data effectively. Furthermore, it is possible for similar nodes to possess distinct labels, indicating that the initial graph constructed based on neighboring nodes may also incorporate a significant amount of noise²⁹. Consequently, scholars have proposed numerous graph regularization methodologies to address the aforementioned challenges. They includes sparse constraint³⁰, smooth constraint³¹, low rank constraint^{32,33}, matrix factorization^{34,35} and so on. However, the constraints employed in these methods are

imposed solely on the learner, while the graph structure is treated as a predefined condition. Fundamentally, these approaches still align with the categories of KNN graph or ϵ -graphs.

Unlike models that solely focus on local neighborhoods, the Transformer architecture³⁶ facilitates message propagation between all nodes, thereby uncovering novel relationships. Kreuzer *et al.*³⁷ propose a learnable position encoding mechanism that employs an additional Transformer encoder to transform the Laplace spectrum of the graph. Dwivedi *et al.*³⁸ proposed the decoupling of structural embeddedness from positional embeddedness, allowing both to be updated in tandem with other parameters. These methods, which leverage kernel functions for node representation or encode both local and global characteristics of the graph structure through additional diffusion kernel, do enhance the structural optimization of the graph. However, they all rely on deep learning techniques and entail considerable training time and computational expenses.

Graph learning with hash-based approaches

Owing to its low retrieval time complexity ($\mathcal{O}(1)$) and minimal storage requirements, hash coding has garnered significant attention from researchers in recent years^{39,40}. There exist several methodologies that integrate hash-based techniques with graph learning approaches. For instance, Lu *et al.*³⁹ introduced the Asymmetric Transfer Hashing (ATH) framework to address the cross-domain heterogeneous search challenge. They optimized domain distribution and feature discrepancies by employing a dual-domain hash function and an adaptive bipartite graph. Wang *et al.* employ a graph embedding learning approach to generate hash codes suitable for model replication detection through the design of a fusion quantization and triplet loss hash network. Huang *et al.*⁴¹ employ a graph embedding learning approach to generate hash codes suitable for model replication detection through the design of a fusion quantization and triplet loss hash network. Hamming Spatial Graph Convolutional Networks (HS-GCN)⁴² model incorporated the concept of Hamming similarity into its framework, thereby enhancing the representation of both users and projects. Wang *et al.*⁴³ employed anchor graphs to effectively capture the intrinsic neighborhood structure and subsequently developed a graph-based hashing model.

In contrast to existing methods that primarily focus on learning hash codes utilizing graph structures, this paper proposes a novel approach where hash coding is employed to infer graph structures. This method not only reduces the computational complexity associated with the initial adjacency matrix but also mitigates the impact of graph noise, thereby providing a semantically enriched initial graph for graph-based deep learning models.

Materials and methods

The focus of this section lies on the proposed architecture for similarity reconstruction based on semantic hash. Once the global graph is generated, it is forwarded to fast GCN for efficient ECG detection. The entire process is depicted in Fig. 1. The section provides a comprehensive exposition of each component in the proposed approach, encompassing its formulation and solution process. Additionally, we elucidate the algorithm's complexity and convergence properties.

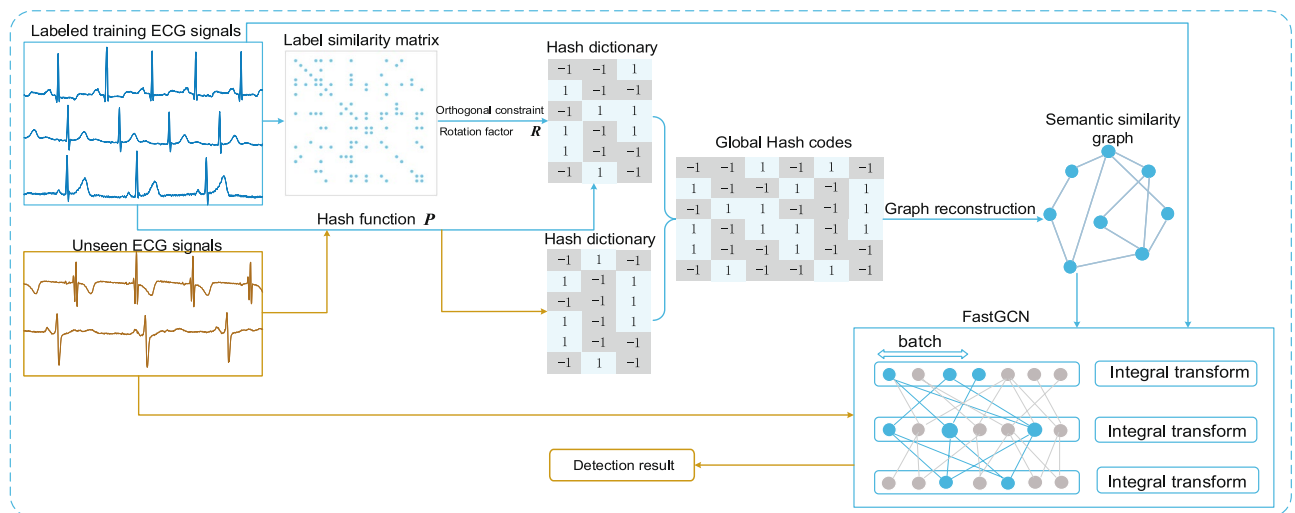


Fig. 1. Basic framework proposed ECG recognition based on semantic hash similarity graph learning. The entire framework comprises three modules: the hash encoding module, the graph reconstruction module, and the fast GCN module. The hash encoding module is primarily responsible for performing hash encoding on ECG signals to construct a global hash dictionary. The graph reconstruction module involves weighted reconstruction of the semantic similarity graph by integrating the dynamic similarity graph with the hash similarity graph. Lastly, the reconstructed semantic similarity graph is input into the FastCNG network to facilitate ECG retrieval.

Preliminary and notations

Throughout the paper, vectors (matrices) are denoted by lowercase (uppercase) bold letters, e.g., \mathbf{b} and \mathbf{X} . $\|\mathbf{X}\|_F = \sqrt{\sum_{i=1}^p \sum_{j=1}^q x_{ij}^2}$ is the Frobenius norm of a matrix $\mathbf{X} = (x_{ij})_{p \times q}$ and $\|\mathbf{b}\|_2 = \sqrt{\sum_{i=1}^p b_i^2}$ is used to denote the ℓ_2 -norm of a vector $\mathbf{b} \in \mathcal{R}^p$. Notation \mathcal{S}_n^p denotes an orthogonal feasible region and $\text{tr}(\mathbf{X})$ denotes the trace of matrix \mathbf{X} . We use $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_n]^T \in \{-1, 1\}^{n \times k}$ to denote the hash dictionary. The ECG training dataset is denoted as $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^n$ with n samples, where $\mathbf{x}_i \in \mathcal{R}^d$ and $y_i \in \{1, 2, \dots, c\} \subset \mathcal{N}$ are the i th sample and its corresponding label, respectively.

Let $G = \{\mathbf{V}, \mathbf{E}, \mathbf{X}\}$ denote a graph, where $\mathbf{V} = \{v_1, \dots, v_n\}$ represents a sequence of nodes with $|\mathbf{V}| = n$ and \mathbf{E} represents the set of edges connecting them. Edges depict the relationships between nodes and can also be represented as an adjacency matrix $\mathbf{A} = [a_{ij}] \in \mathcal{R}^{n \times n}$, with a_{ij} denoting the relationship between nodes v_i and v_j .

Additionally, the optimization process employs the following notations: the i -th column of matrix $\mathbf{X} \in \mathcal{R}^{p \times q}$ is denoted by \mathbf{x}_i . $\mathbf{X}_{\bar{i}} = [\mathbf{x}_1, \dots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \dots, \mathbf{x}_q] \in \mathcal{R}^{p \times (q-1)}$ means that the i -th column of \mathbf{X} is removed. $\mathbf{X}_{i,v} = [\mathbf{x}_1, \dots, \mathbf{x}_{i-1}, \mathbf{v}, \mathbf{x}_{i+1}, \dots, \mathbf{x}_q] \in \mathcal{R}^{p \times q}$ denotes \mathbf{X} with its i -th column replaced by a given vector \mathbf{v} . And \otimes denotes the element-wise product. Finally, $\mathcal{B}(c, r) = \{\mathbf{x} \in \mathcal{R}^p \mid \|\mathbf{x} - c\|_2 \leq r\}$ is a p -dimensional hypersphere, where $c \in \mathcal{R}^p$ is the center and r is the radius.

Semantic hash similarity graph learning

Problem formulation

The binary-coded hash representation of data ensures consistent similarity between data and semantic affinity. Building upon this foundation, a novel framework has recently emerged in the field of hash learning, aiming to bridge the spatial gap between semantic space and Hamming space by distilling discrete hash codes from semantic affinity. The next step will involve the creation of a hash dictionary.

The approach specifically utilizes one-hot tag vectors for constructing semantic affinity and subsequently employs discrete symmetric matrix factorization to generate discriminant compact hash codes. Given any two label vectors $\mathbf{y}_i, \mathbf{y}_j \in \{0, 1\}^c$, their similarity can be represented by $S_{ij} = \langle \mathbf{y}_i, \mathbf{y}_j \rangle$ or $S_{ij} = \frac{\langle \mathbf{y}_i, \mathbf{y}_j \rangle}{\|\mathbf{y}_i\| \|\mathbf{y}_j\|}$ (multi-label). So the affinity matrix can be expressed as $\mathbf{S} = \mathbf{Y}^T \mathbf{Y} \in \mathcal{R}^{n \times n}$. Symmetric decomposition model is to make the binary hash row vector $\mathbf{b}_i \in \{-1, 1\}^k$ of code length k satisfy $\mathbf{b}_i \mathbf{b}_j^T \approx k S_{ij}$ as far as possible, and its matrix form can be expressed as $\mathbf{B} \mathbf{B}^T \approx k \mathbf{S}$. A universal objective function can be unified as follows:

$$\min_{\mathbf{B} \in \{-1, 1\}^{n \times k}} \mathcal{F} = \|\mathbf{kS} - \mathbf{B} \mathbf{B}^T\|_F^2, \text{ s.t. } \mathbf{B}^T \mathbf{B} = n \mathbf{I}_k. \quad (1)$$

Equation 1 represents the classical objective function for learning hash codes based on semantics^{44–47}. However, due to the impact of the binary discrete constraint inherent in hashing, optimizing this function has consistently been a challenging problem in the field. Currently, there are primarily two strategies for optimizing Eq. 1: introducing intermediate variables to preserve the discreteness of hash coding^{44,45}, or employing a semi-relaxation strategy for optimization^{46,47}.

Unlike the aforementioned methods, we propose an equivalent objective function to overcome the optimization difficulty of Eq. 1. By re-evaluating Eq. 1, we find that its aim is $k \mathbf{S} \approx \mathbf{B} \mathbf{B}^T$, i.e., $k \mathbf{Y}^T \mathbf{Y} \approx \mathbf{B} \mathbf{B}^T$. The accomplishment of $\sqrt{k} \mathbf{Y}^T \approx \mathbf{B}$ can serve as a viable alternative to the aforementioned objective. However, there exists a dimensional discrepancy between the label space and the Hamming space, which renders the achievement of this objective infeasible. To do this, we introduce an orthogonal rotation factor $\mathbf{R} \in \mathcal{R}^{c \times k}$ to effectively mitigate the dimensional difference, i.e., $\sqrt{k} \mathbf{Y}^T \approx \mathbf{B} \mathbf{R}^T$. As long as $\mathbf{R}^T \mathbf{R} \equiv \mathbf{I}_k$, we have $(\mathbf{B} \mathbf{R}^T)(\mathbf{B} \mathbf{R}^T)^T = \mathbf{B} \mathbf{R}^T \mathbf{R} \mathbf{B}^T = \mathbf{B} \mathbf{B}^T = \sqrt{k} \mathbf{Y}^T (\sqrt{k} \mathbf{Y}^T)^T = k \mathbf{Y}^T \mathbf{Y}$. It is worth noting that the condition for the aforementioned equivalence is the constant orthogonality of the rotation factor. In other words, the subsequent optimization process must ensure that the rotation factor consistently resides on the Stiefel manifold. To sum up, Eq. 1 can be equivalently expressed as:

$$\min_{\mathbf{B} \in \{-1, 1\}^{n \times k}, \mathbf{R} \in \mathcal{R}^{c \times k}} \mathcal{F} = \|\tilde{\mathbf{Y}}^T - \mathbf{B} \mathbf{R}^T\|_F^2, \text{ s.t. } \mathbf{B}^T \mathbf{B} = n \mathbf{I}_k, \mathbf{R}^T \mathbf{R} \equiv \mathbf{I}_k. \quad (2)$$

where $\tilde{\mathbf{Y}} = \sqrt{k} \mathbf{Y}$. The subsequent optimization then shifted its focus to \mathbf{R} . In order to effectively approach Eq. 1, it is imperative to ensure that in the subsequent iterations, \mathbf{R} progresses orthogonally step by step, thereby being constrained within the orthogonal feasible region.

Optimization

Due to the indefinite magnitudes of c and k , the strategy based on Stiefel manifold is not applicable to Eq. 2. Therefore, we adopt a two-step optimization approach. Firstly, we use the proximal linearized approximation of the augmented Lagrangian algorithm^{48,49} to minimize the function value. Then, we utilize column-wise block coordinate descent method to handle orthogonal constraints. Eq. 2 w.r.t. \mathbf{R} can be formulated as:

$$\min_{\mathbf{R} \in \mathcal{R}^{c \times k}} \mathcal{F}(\mathbf{R}) = \text{tr}(\mathbf{R} \mathbf{B}^T \mathbf{B} \mathbf{R}^T) - 2 \text{tr}(\tilde{\mathbf{Y}} \mathbf{B} \mathbf{R}^T) \text{ s.t. } \mathbf{R}^T \mathbf{R} \equiv \mathbf{I}_k. \quad (3)$$

The Lagrangian function of Eq. 3 can be expressed as follows, where Λ represents the Lagrangian multiplier matrix for the orthogonal constraint:

$$\mathcal{L} = \mathcal{F}(\mathbf{R}) - \frac{1}{2} \text{tr}(\Lambda(\mathbf{R}^T \mathbf{R} - \mathbf{I}_k)). \tag{4}$$

Then the gradient of Eq. 4 for \mathbf{R} is $\nabla_{\mathbf{R}} \mathcal{L} = \nabla \mathcal{F}(\mathbf{R}) - \mathbf{R}\Lambda$. Let $\nabla_{\mathbf{R}} \mathcal{L} = 0$, combined with the first-order optimization conditions (Substationarity: $(\mathbf{I}_c - \mathbf{R}\mathbf{R}^T)\nabla \mathcal{F}(\mathbf{R}) = 0$, Symmetry: $\mathbf{R}^T \nabla \mathcal{F}(\mathbf{R}) = \mathbf{R} \nabla \mathcal{F}(\mathbf{R})^T$, Feasibility: $\mathbf{R}^T \mathbf{R} = \mathbf{I}_k$), we have $\Lambda = \mathbf{R}^T \nabla \mathcal{F}(\mathbf{R}) = \nabla \mathcal{F}(\mathbf{R})^T \mathbf{R}$, $\nabla \mathcal{F}(\mathbf{R}) = \mathbf{R}\Lambda$ and $\nabla \mathcal{F}(\mathbf{R}) - \mathbf{R} \nabla \mathcal{F}(\mathbf{R})^T \mathbf{R} = 0$.

In order to uphold the orthogonality constraint, we incorporate an additional penalty term for orthogonality. Thus, Eq. 4 be reformulated as follows:

$$\mathcal{L}_\beta = \mathcal{F}(\mathbf{R}) - \frac{1}{2} \text{tr}(\Lambda(\mathbf{R}^T \mathbf{R} - \mathbf{I}_k)) + \frac{\beta}{4} \|\mathbf{R}^T \mathbf{R} - \mathbf{I}_k\|_F^2. \tag{5}$$

where β is the penalty regulation coefficient. Then $\nabla_{\mathbf{R}} \mathcal{L}$ can be rewritten as:

$$\nabla_{\mathbf{R}} \mathcal{L}_\beta(\mathbf{R}, \Lambda) = \nabla \mathcal{F}(\mathbf{R}) - \mathbf{R}\Lambda - \beta \mathbf{R}(\mathbf{I}_k - \mathbf{R}^T \mathbf{R}). \tag{6}$$

For the current iteration $\mathbf{R}^{(v)}$, the proximal linearized augmented Lagrangian function is defined as:

$$\begin{cases} \Lambda^{(v)} = \Psi(\nabla \mathcal{F}(\mathbf{R}^{(v)})^T \mathbf{R}^{(v)}), \\ \tilde{\mathcal{L}}_\beta = \text{tr}((\mathbf{G}^T (\mathbf{R} - \mathbf{R}^{(v)})) + \frac{\eta}{2} \|\mathbf{R} - \mathbf{R}^{(v)}\|_F^2), \\ \mathbf{R}^{(v+1)} = \arg \min_{\mathbf{R}} \tilde{\mathcal{L}}_\beta(\mathbf{R}). \end{cases} \tag{7}$$

where $\mathbf{G} = \nabla_{\mathbf{R}} \mathcal{L}_\beta(\mathbf{R}^{(v)}, \Lambda^{(v)})$, $\frac{1}{\eta}$ is the step size of gradient, $\Psi(\mathbf{A}) = \frac{\mathbf{A} + \mathbf{A}^T}{2}$ is used to ensure the symmetry in the first-order optimization conditions.

For redundant spherical constraints in the orthogonal feasible region, we add an additional term for $\nabla \mathcal{F}(\mathbf{R})$, i.e., $\nabla \mathcal{F}(\mathbf{R}) = \mathbf{R}\Lambda + \mathbf{R}\mathbf{D}$, where \mathbf{D} is a diagonal matrix determined by the Lagrangian multiplier. Therefore, the iterative formula of $\Lambda^{(v)}$ in Eq. 7 can be expressed as:

$$\Lambda^{(v)} = \Psi(\nabla \mathcal{F}(\mathbf{R}^{(v)})^T \mathbf{R}^{(v)}) + \Xi^{(v)}. \tag{8}$$

where $\Xi = \text{Diag}(\mathbf{R}^T \nabla_{\mathbf{R}} \mathcal{L}_\beta(\mathbf{R}, \Psi(\nabla \mathcal{F}(\mathbf{R})^T \mathbf{R}))$.

The updating rule of variable \mathbf{R} provided by Eq. 7 does not guarantee that each iteration optimization is within the orthogonal feasible region. In order to address this issue, we adopt column-wise block coordinate descent method to achieve the second objective optimization of Eq. 7. Concretely, we fix the $k - 1$ columns of \mathbf{R} and use the i -th column as a variable. The sub-target of Eq. 7 can be expressed as:

$$\min_{\mathbf{r} \in \mathcal{R}^c} \tilde{\mathcal{L}}_{\beta, \mathbf{R}}^i(\mathbf{r}), \text{ s.t. } \|\mathbf{r}\|_2 = 1, \mathbf{R}_{\bar{i}}^T \mathbf{r} = 0. \tag{9}$$

where $\tilde{\mathcal{L}}_{\beta, \mathbf{R}}^i(\mathbf{r}) = \tilde{\mathcal{L}}_\beta(\mathbf{R}_{i, \mathbf{r}})$. The constraint $\mathbf{R}_{\bar{i}}^T \mathbf{r} = 0$ indicates the solution $\mathbf{r}^* \in \text{Null}(\mathbf{R}_{\bar{i}}) = \{\mathbf{r} \in \mathcal{R}^c | \mathbf{R}_{\bar{i}}^T \mathbf{r} = 0\}$, which is equivalent to $\mathbf{r} = (\mathbf{I}_c - \mathbf{R}_{\bar{i}} \mathbf{R}_{\bar{i}}^T) \mathbf{r}$.

Therefore, Eq. 5 can be simplified as:

$$\min_{\mathbf{r} \in \mathcal{R}^c} f_i(\mathbf{r}) = \tilde{\mathcal{L}}_{\beta, \mathbf{R}}^i((\mathbf{I}_c - \mathbf{R}_{\bar{i}} \mathbf{R}_{\bar{i}}^T) \mathbf{r}), \text{ s.t. } \|\mathbf{r}\|_2 = 1, \mathbf{r} \in \text{Null}(\mathbf{R}_{\bar{i}}). \tag{10}$$

For the i -th column \mathbf{r}_i , the gradient of Eq. 10 can be expressed as:

$$\nabla f_i(\mathbf{r}_i) = (\mathbf{I}_c - \mathbf{R}_{\bar{i}} \mathbf{R}_{\bar{i}}^T) \nabla \tilde{\mathcal{L}}_{\beta, \mathbf{R}}^i((\mathbf{I}_c - \mathbf{R}_{\bar{i}} \mathbf{R}_{\bar{i}}^T) \mathbf{r}_i) = (\mathbf{I}_c - \mathbf{R}_{\bar{i}} \mathbf{R}_{\bar{i}}^T) \nabla f_i(\mathbf{r}_i). \tag{11}$$

From Eq. 11, it is not difficult to find: $\nabla f_i(\mathbf{r}_i) \in \text{Null}(\mathbf{R}_{\bar{i}})$. Therefore, both \mathbf{r}_i and $\nabla f_i(\mathbf{r}_i)$ all lie in $\text{Null}(\mathbf{R}_{\bar{i}})$. And we know that the null space of $\mathbf{R}_{\bar{i}}$ is the orthocomplement of the row space of $\mathbf{R}_{\bar{i}}$, i.e., $(\text{Row}(\mathbf{R}_{\bar{i}}))^\perp = \text{Null}(\mathbf{R}_{\bar{i}})$, so any point in the span of \mathbf{r}_i and $\nabla f_i(\mathbf{r}_i)$, i.e., $\text{Span}\{\mathbf{r}_i, \nabla f_i(\mathbf{r}_i)\}$, satisfies the orthogonal constraint. Let $\tilde{\mathbf{r}}_i = \mathbf{r}_i - \iota \nabla f_i(\mathbf{r}_i) \in \text{Span}\{\mathbf{r}_i, \nabla f_i(\mathbf{r}_i)\}$, and then we can get the orthogonal feasible solution:

$$\mathbf{r}_i^{(v+1)} = \frac{\mathbf{r}_i^v - \iota \nabla f_i(\mathbf{r}_i^v)}{\|\mathbf{r}_i^v - \iota \nabla f_i(\mathbf{r}_i^v)\|_2}. \tag{12}$$

where $\iota = \frac{1}{\eta}$ is the step size.

The optimization of the hash dictionary can be resolved immediately once the rotation factor \mathbf{R} is optimized. To reiterate Eq. 3, minimizing $\mathcal{F}(\mathbf{B})$ is equivalent to maximizing:

$$\max \mathcal{F}(\mathbf{B}) = \text{tr}(\tilde{\mathbf{Y}} \mathbf{B} \mathbf{R}^T) = \text{tr}(\mathbf{B}^T \tilde{\mathbf{Y}}^T \mathbf{R}). \tag{13}$$

By utilizing the property of inequality, the concise and closed form of the hash dictionary can be derived:

$$\mathbf{B} = \text{sgn}(\tilde{\mathbf{Y}}^T \mathbf{R}). \quad (14)$$

Unseen signal hash representation

The establishment of communication between the primitive space and Hamming space necessitates the utilization of an explicit hash function, which serves as a practical tool for generating hash codes for previously unseen signal data. For simplicity, the most straightforward linear hash function is used to establish links.

We minimize the following square loss:

$$\mathcal{H}(\mathbf{P}) = \|\mathbf{B}^T - \mathbf{P}\mathbf{X}\|_F^2 + \lambda\|\mathbf{P}\|_F^2. \quad (15)$$

where λ is a regularization parameter, which is used to prevent over learning. To minimize Eq. 15, we can get $\mathbf{P} = \mathbf{B}^T \mathbf{X}^T (\mathbf{X}\mathbf{X}^T + \lambda \mathbf{I}_d)^{-1}$. Confronted with an unobserved signal dataset $\mathbf{X}_{te} \in \mathcal{R}^{d \times m}$, we can efficiently produce its hash representation $\mathbf{B}_{te}^T = \text{sgn}(\mathbf{P}\mathbf{X}_{te})$ using this linear hashing function.

The integration of the training hash dictionary and the unseen hash representation enables the acquisition of the global hash representation $\mathbf{B}_g^T = [\mathbf{B}^T, \mathbf{B}_{te}^T]$. The global graph similarity can be computed by evaluating the Hamming similarity of each hash code. The more intuitive procedure is summarized in Algorithm 1.

Input:

Labels $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_n]$;
 Training sample set \mathbf{X} and test set \mathbf{X}_{te} ;
 Hash length k and parameters β, ι, λ ;

Output:

Semantic hash similarity graph \mathbf{A}_H ;
 1: Orthogonally initialize \mathbf{R} and binary initialize \mathbf{B} ;
 2: **repeat**
 3: **for** $i = 1, 2, \dots, k$ **do**
 4: Update \mathbf{r}_i by Eq. 5, Eq. 7, Eq. 9 and Eq. 12;
 5: $i := i + 1$;
 6: **end for**
 7: $\mathbf{R} = [\mathbf{r}_1, \dots, \mathbf{r}_k]$;
 8: Binary initialize \mathbf{B} ;
 9: Update \mathbf{B} by Eq. 14;
 10: **until** the iteration stop condition;
 11: Calculate the hash function \mathbf{P} by Eq. 15 ;
 12: Generate the unseen signal hash representation by $\mathbf{B}_{te}^T = \text{sgn}(\mathbf{P}\mathbf{X}_{te})$;
 13: Global hash representation $\mathbf{B}_g^T = [\mathbf{B}^T, \mathbf{B}_{te}^T]$;
 14: Measure the Hamming similarity of $\mathbf{b}_i \in \mathbf{B}_g$ and $\mathbf{b}_j \in \mathbf{B}_g$ to generate the semantic similarity graph \mathbf{A}_H .

Algorithm 1. Semantic Hash Similarity Graph

Dynamic similarity graph

The aforementioned semantic hash similarity graph can be derived expeditiously from the label space. However, it fails to consider the structural interconnections among the original signals. Hence, it becomes imperative to amalgamate the structural similarity in order to construct the ultimate analogous structure. The previous approaches for constructing similar matrices typically rely on inner product or Euclidean distance, which are not suitable for paired time series due to the potential cascade effect between the series leading to temporal lag. The Dynamic Time Warping distance (DTW)⁵⁰ is a robust measure of signal similarity that accounts for temporal variations. Therefore, we utilize the DTW distance to construct a structure analogous to the original signal.

Formally, given any two signal sequences $\mathbf{x}_i, \mathbf{x}_j$, their similarity A_D^{ij} is computed utilizing the unbounded distance of DTW.

$$A_D^{ij} = \exp\left(-\frac{\text{DTW}(\mathbf{x}_i, \mathbf{x}_j, \omega)}{\epsilon}\right). \quad (16)$$

where ϵ is a hyper-parameter, ω is local constraint window size, and A_D is referred to as dynamic similarity graph. The ultimate global graph structure is formed by the weighted summation of the semantic similarity graph and the dynamic structural similarity graph, i.e., $\mathbf{A} = \kappa \mathbf{A}_D + (1 - \kappa) \mathbf{A}_H$. The above hyper-parameters are set to $\epsilon = 0.5$, $\omega = 3$ and $\kappa = 0.3$ in order to minimize parameter interdependence and reduce algorithm debugging time.

Convergence analysis and complexity discussion

Next we discuss the convergence of Algorithms 1. The iteration of two variables is involved in Algorithm 1, where the iteration of the hash dictionary does not play a crucial role in minimizing the objective. This is because learning the hash dictionary essentially involves selecting the best vertex from a k -dimensional hypercube vertex. Therefore, we can only focus on discussing the iterative convergence of the rotation factor R . The experimental section provides empirical validation while the appendix presents theoretical analysis.

The complexity analysis of Algorithm 1 is outlined below. Initializing R and B takes $\mathcal{O}(nk + ck)$. Each iteration to update R and B requires calculation costs of $\mathcal{O}(ck + 4ck^2)$ and $\mathcal{O}(cnk)$, respectively. Calculating P needs $\mathcal{O}(nd^2 + d^3 + knd + kd^2)$. The generation of B_{te} requires $\mathcal{O}(kmd)$. The computation cost for the ultimate generation of the global semantic graph structure A amounts to $\mathcal{O}((m + n)^2)$.

Experiments and discussions

To validate the effectiveness of the constructed hash similarity graph, a deep graph model framework is required. In this study, we employ the fast CNG model proposed by Yang *et al.*⁵¹. The FastGCN converts the set of graph vertices into independent and identically distributed (i.i.d.) samples drawn from a probability distribution, thereby facilitating the uniform estimation of the loss gradient for parameter updates. Following this, the full GCN architecture is utilized to compute embeddings for the newly introduced vertices.

Experiment setting

(1) Datasets:

- MIT-BIH arrhythmia database⁵²: The dataset was continuously recorded for a duration of 30 minutes, with a sampling rate of 360 samples per second. Each cycle within the dataset was annotated with a reference value based on the R-peak, serving as an analytical ground truth. For this study, a subset of the dataset consisting of 10,000 ECG fragments from II leads was selected and categorized into four different classes: normal heartbeat (N), left bundle branch block heartbeat (L), right bundle branch block heartbeat (R), and ventricular premature beat (V). A training set comprising 8,000 randomly chosen samples was created, including 7,000 training instances and 1,000 validation instances. The remaining samples were allocated to the test set.
- The AF ECG dataset¹²: It was obtained from the 2017 PhysioNet/CinC Challenge and comprised of 8528 ECG segments ranging in duration from 30 to 60 seconds. This dataset was categorized into four groups: normal sinus rhythm (N), atrial fibrillation (A), other cardiac rhythm (O), and noise segment (\sim). For training purposes, we randomly selected 7000 sample signals while the remaining were used as a test set. Additionally, we extracted 500 sample signals from the training set for validation.

(2) Compared methods: The detection performance of the constructed semantic hash similarity graph in the FastGCN network is verified by conducting experimental comparisons with several commonly used intelligent algorithms. These algorithms include machine learning methods such as long short-term memory based on CNN (CNN-LSTM)⁵³, bidirectional LSTM (BiLSTM)⁵⁴, one-dimensional convolutional neural networks (1DCNN)³, attentive recurrent neural networks (ARNN)⁴, GCN⁵⁵, CNN-BiLSTM-ATT², ST-Re-GE¹⁹ and MPR-STSGCN¹⁰.

(3) Metrics: The metrics employed in our study include accuracy for each category, average accuracy (Acc), Precision, Recall (sensitivity), and Macro-F1 score based on the AAMI guidelines. The following metrics are presented:

$$\begin{cases} Acc = \frac{TP+TN}{TP+FP+FN+TN}, \\ Precision = \frac{TP}{TP+FP}, \\ Recall = \frac{TP}{TP+FN}, \\ F_{1,ma} = \frac{2Precision*Recall}{Precision+Recall} = \frac{2TP}{2TP+FP+FN}. \end{cases} \quad (17)$$

where the abbreviations TP, FP, TN, and FN respectively stand for true positive, false positive, true negative, and false negative.

(4) Experimental protocol: For Algorithm 1, it involves three parameters β , ι and λ , which are set to $\min(\max(10^{-2}, 10^{-3}\|B^T B\|_F^2), 10^5)$, $1e - 4$ and $1e - 1$ respectively. The hash code length k is set to 32. The number of epochs for FastGCN is set to 30, the number of hidden units is set to 256, the ADAM learning rate is set to 10^{-2} , the batch size is set to 200, and the resample size is set to 400. Since FastGCN has only two layers, we kernelize the samples in advance to improve its nonlinearity. In addition, to facilitate the reproducibility of the experiment in FastGCN, we provide two datasets along with their corresponding semantic hash graphs (<https://pan.baidu.com/s/19-APd8zAYim2hBr6YvCutg?pwd=7neo>). All experiments are carefully performed on a workstation with Intel(R) Core(TM) i7-9700 CPU@3.00GHz, NVIDIA GeForce RTX 3070, 64 GB RAM.

Method	Dataset MIT-BIH				Dataset AF			
	Acc	Precision	Recall	$F_{1,ma}$	Acc	Precision	Recall	$F_{1,ma}$
CNN-LSTM ⁵³	0.9728	0.9773	0.9728	0.9727	0.6649	0.7086	0.5204	0.4857
BiLSTM ⁵⁴	0.9835	0.9836	0.9842	0.9837	0.4980	0.4983	0.4223	0.3770
1DCNN ³	0.9870	0.9875	0.9872	0.9873	<i>0.7101</i>	0.5793	0.5332	<i>0.5517</i>
ARNN ⁴	0.9870	<i>0.9928</i>	0.9932	<i>0.9929</i>	0.6852	0.5370	0.4719	0.4229
GCN ⁵⁵	0.9810	0.9814	0.9815	0.9813	0.5216	0.5263	0.4516	0.3885
CNN-BiLSTM-ATT ²	0.9845	0.9847	0.9850	0.9848	0.6852	0.7296	0.5314	0.4494
ST-ReGE ¹⁹	0.9897	0.9827	0.9894	0.9888	0.6932	0.6996	0.5431	0.5163
MPR-STSGCN ¹⁰	<i>0.9915</i>	0.9895	0.9916	0.9903	0.6977	<i>0.7136</i>	<i>0.5577</i>	0.5497
Ours	0.9930	0.9931	<i>0.9930</i>	0.9930	0.7277	0.6259	0.5781	0.5911

Table 1. Comparison results of various methods on two data sets. bold text indicates the best and italic text indicates the second best result, respectively.

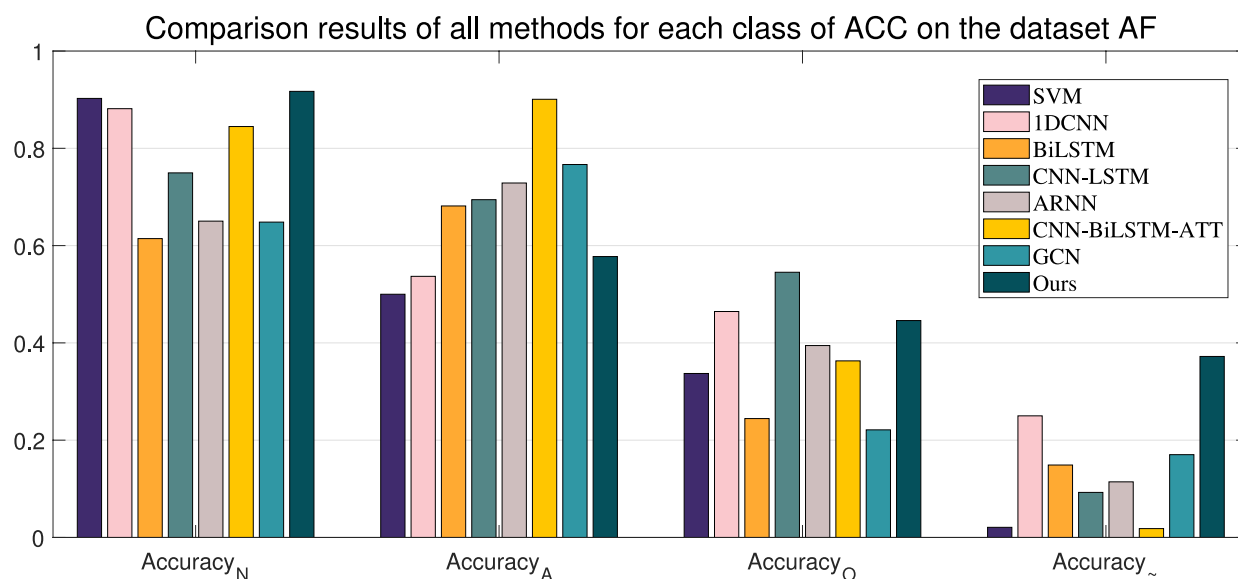


Fig. 2. Comparison results of all methods for each class of ACC on the dataset MIT.

Analysis of experimental results

- (1) MIT-BIH: The Acc, Precision, Recall and F1 score results of all baselines on MIT-BIH are listed in Table 1. The comparison of Acc results for all baselines in each category is shown in Fig. 2. The feedback from the results indicates that the category features in this dataset are notably discernible, and all methods exhibit relatively high detection performance. The superiority of our method is evident in terms of Acc, precision, and F1 score compared to other methods. The value of recall is slightly weaker compared to ARNN. All the benchmarks are based on deep learning methodologies. The results indicate that all approaches have achieved satisfactory outcomes, suggesting that the deep learning architecture exhibits superior performance in extracting critical signal features. However, it should be noted that deep networks require longer training time. The ARNN model, in addition to achieving the second highest performance among all methods, suggests that employing an attention mechanism in time-series networks is more appropriate for analyzing ECG signals. The FastGCN structure, despite its mere two layers, outperforms other baselines in terms of recognition results. This indirectly suggests that the semantic hash similarity graph we constructed is abundant in semantic information, which greatly facilitates the efficient detection of the FastGCN model.
- (2) AF dataset: The baseline comparison results have been summarized in Table 1, while Fig. 3 displays the accuracy (ACC) for each type of detection. The results presented in Table 1 demonstrate that our method outperforms all baselines in terms of ACC, recall, and F1 scores over a wide range of intervals. This also demonstrates the effectiveness of the proposed semantic hash graph learning framework in mitigating the inherent rigidity within the label space, as well as addressing issues pertaining to incomplete or redundant adjacency within the similarity graph. However, it should be noted that the precision results are compara-

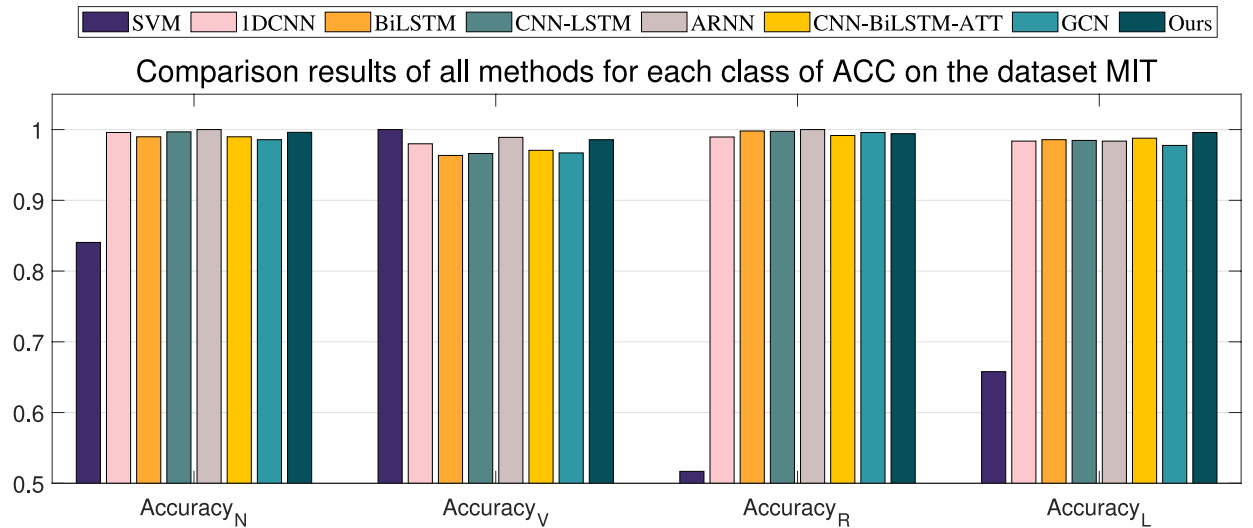


Fig. 3. Comparison results of all methods for each class of ACC on the dataset AF.

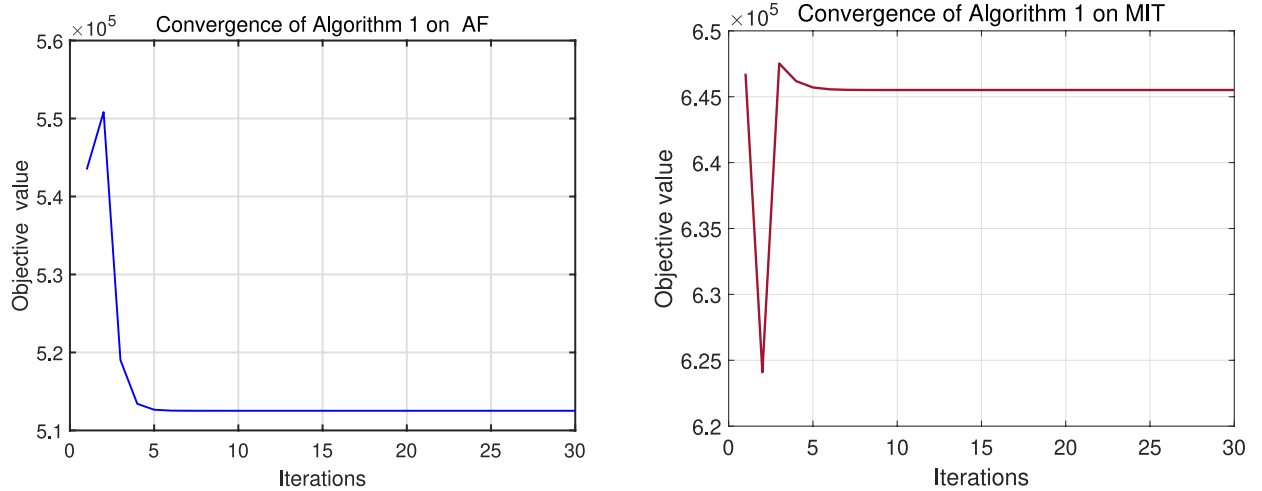


Fig. 4. Convergence of algorithm 1 on two datasets.

tively weaker. The absence of an attention module in FastGCN model may contribute to this phenomenon, potentially indicating a class imbalance.

The results of each baseline method in each type of detection vary significantly due to the imbalance in the number of different samples in this dataset, as illustrated by Fig. 2. Among these variations, the recognition rate for normal rhythm (N) and atrial fibrillation (A) is the highest, while the recognition rate for the other two types is unsatisfactory. Our method performs better in the first and fourth categories of the recognition results, while CNN-BiLSTM-ATT significantly leads in the second category of the detection results; however, its performance in the fourth category of the detection results is not satisfactory. The results from the graph show that our method, as well as ST-ReGE and MPR-STSGCN, have relatively high recognition accuracy in each category, indicating that these three methods are not sensitive to class imbalance, as the dataset is highly imbalanced in terms of categories. The potential reason is that the parameters of the attention module are updated solely through loss backpropagation from labels and predicted values, without any additional supervision information being introduced. Consequently, its supervision is constrained, rendering it susceptible to label overfitting.

Convergency analysis and training time

The convergence of algorithm 1 on two datasets is investigated, and the results are depicted in Fig. 4. A remarkably rapid convergence rate (less than 5 iterations) is observed, primarily attributed to the extraction of hash books from tag space rather than semantic associations. Despite the increased complexity of orthogonal rotation factor optimization, its volume remains minimal, resulting in a highly expedited optimization speed. The convergence diagram on MIT reveals that, in addition, the objective function does not exhibit a genuine

Method	Dataset MIT-BIH		Dataset AF	
	training time (s)	retrieval time (s)	training time (s)	retrieval time (s)
CNN-LSTM ⁵³	63.2452	0.6778	55.3131	0.3759
BiLSTM ⁵⁴	42.3434	0.5645	36.7406	0.5474
IDCNN ³	55.502	0.4274	40.4976	0.3517
ARNN ⁴	29.9912	0.144	27.5511	0.1497
GCN ⁵⁵	39.8854	0.3626	34.1061	0.3771
CNN-BiLSTM-ATT ²	241.8406	0.8446	350.7574	0.7907
ST-ReGE ¹⁹	461.2736	0.9637	688.7921	0.8963
MPR-STSGCN ¹⁰	735.3691	1.2438	966.5514	1.9886
Generate graph+FastGCN	6.0311+23.4313	0.4851	2.6438+8.2063	0.5376

Table 2. Comparison of training time and retrieval time on two data sets. bold text indicates the best and italic text indicates the second best result, respectively.

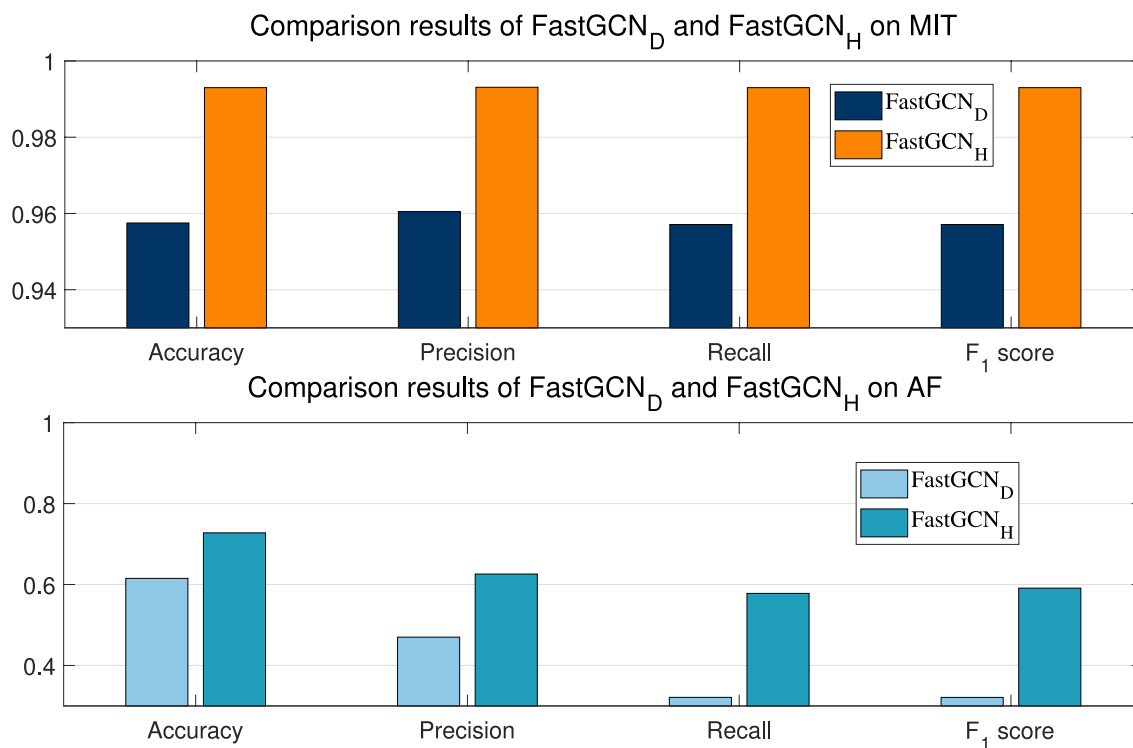


Fig. 5. Comparison results between FastGCN_H and FastGCN_D on two datasets.

decrease in value. The primary reason is that updating the hash code is crucial for selecting the optimal vertex on the K-dimensional hypercube network, and it does not positively impact the minimization of the objective function value; in fact, it may even have a detrimental effect. Furthermore, rotation factor R has a significantly smaller volume compared to B, resulting in its contribution to the target value being much less significant than that of B. Therefore, we only need to ensure that the algorithm can converge normally. In the appendix, we give a strict proof that updating the rotation factor R can guarantee the reduction of the target value.

The training time and retrieval time at all baselines are also examined in our investigation. To be fair, the epochs number for each network is set to 30. The training time and retrieval time for all baselines are listed in Table 2, revealing that our method outperforms other deep learning approaches. It is noteworthy that the construction of a global similarity graph by combining semantic hash similarity only requires a few seconds, which falls within an acceptable timeframe.

Ablation analysis

The contribution of the semantic hash similarity graph to the experimental performance is further analyzed through ablation experiments conducted on two datasets, providing deeper insights into the module. The abbreviated symbols FastGCN_H and FastGCN_D are utilized for the purpose of expressing the combined

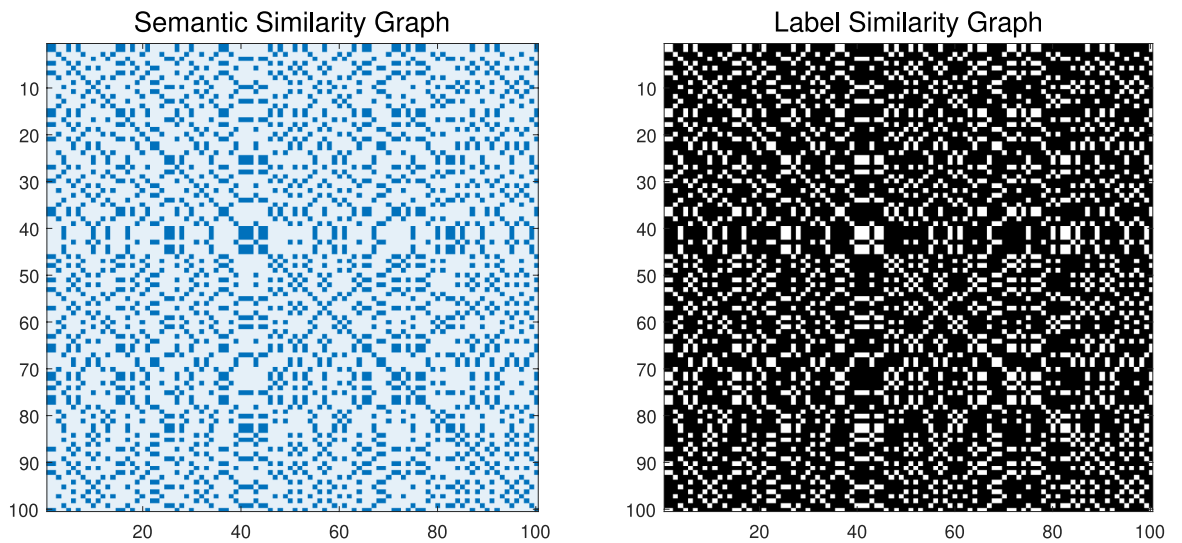


Fig. 6. The visualization comparison of semantic similarity graphs and label similarity graphs on MIT.

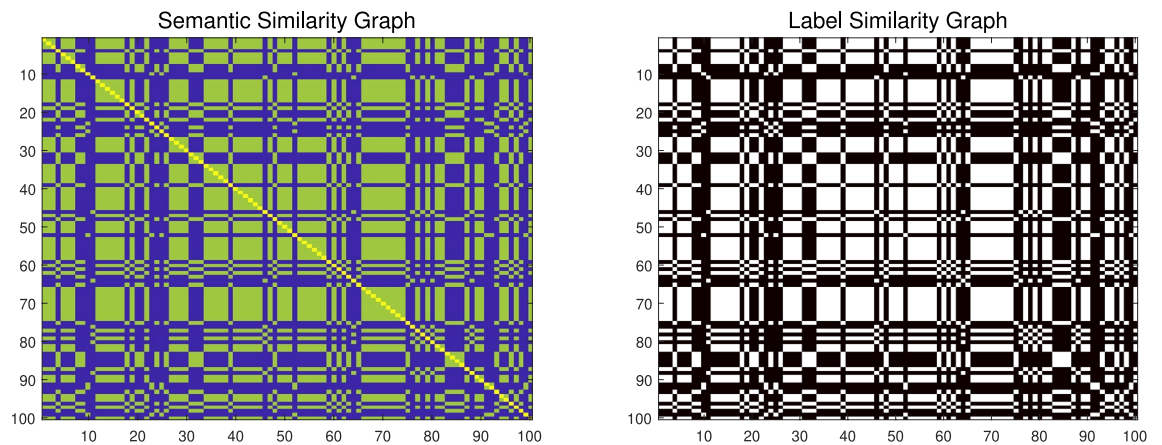


Fig. 7. The visualization comparison of semantic similarity graphs and label similarity graphs on AF.

semantic hash graph and its excised version more conveniently. The FastGCN_D version is obtained when $\kappa = 1$ in our method.

The experimental comparison results of the two methods on the two datasets are presented in Fig. 5. The findings clearly demonstrate that the incorporation of semantic hash graph significantly enhances the performance of FastGCN, thereby confirming the presence of abundant semantic information in the learned graph for effective class separation.

To make it more intuitive, we selected the first 100 signals from each of the two datasets and compared their semantic similarity graphs and label similarity graphs, as shown in Figs. 6 and 7. The extracted semantic similarity graphs fit the label similarities very well.

Parameter sensitivity

The above experiments are conducted after preconfiguring all parameters. Subsequently, we examine the impact of parameter variations on the outcomes using two original datasets. The focus of our research lies not in extensive exploration of network parameters, but rather in examining the impact of these parameters within the generated semantic hash graph model on the outcomes.

The weight parameter κ between semantic similarity graph and dynamic structural similarity graph exerts the most significant impact on the outcome. The impact of varying this parameter on the outcome was investigated in our experiment, and the results are illustrated in Fig. 8. The results depicted in the figure indicate that it is more suitable to choose the interval [0.1, 0.5] for parameter κ on data set MIT-BIH, while selecting the interval [0.1, 0.6] on data set AF would be more appropriate.

For penalty parameter β , we have set a general range of adjustments, which we will not investigate further here. Based on the experimental findings, it was conclusively determined that the variation of β has no impact on the experimental results. Instead, the results exhibited remarkable stability throughout the experiment. However,

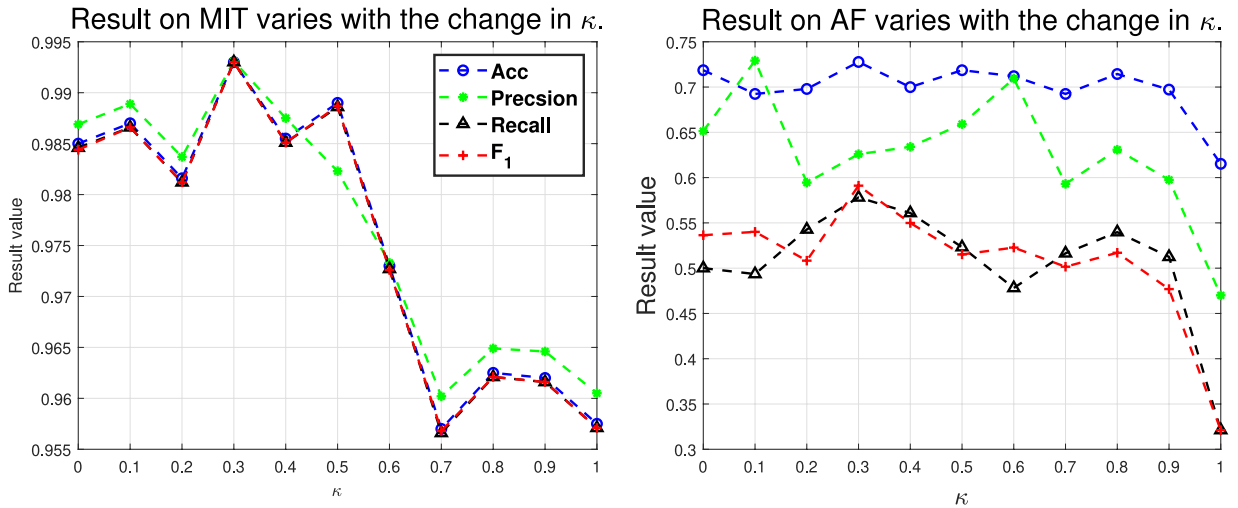


Fig. 8. Sensitivity of parameter κ on two datasets.

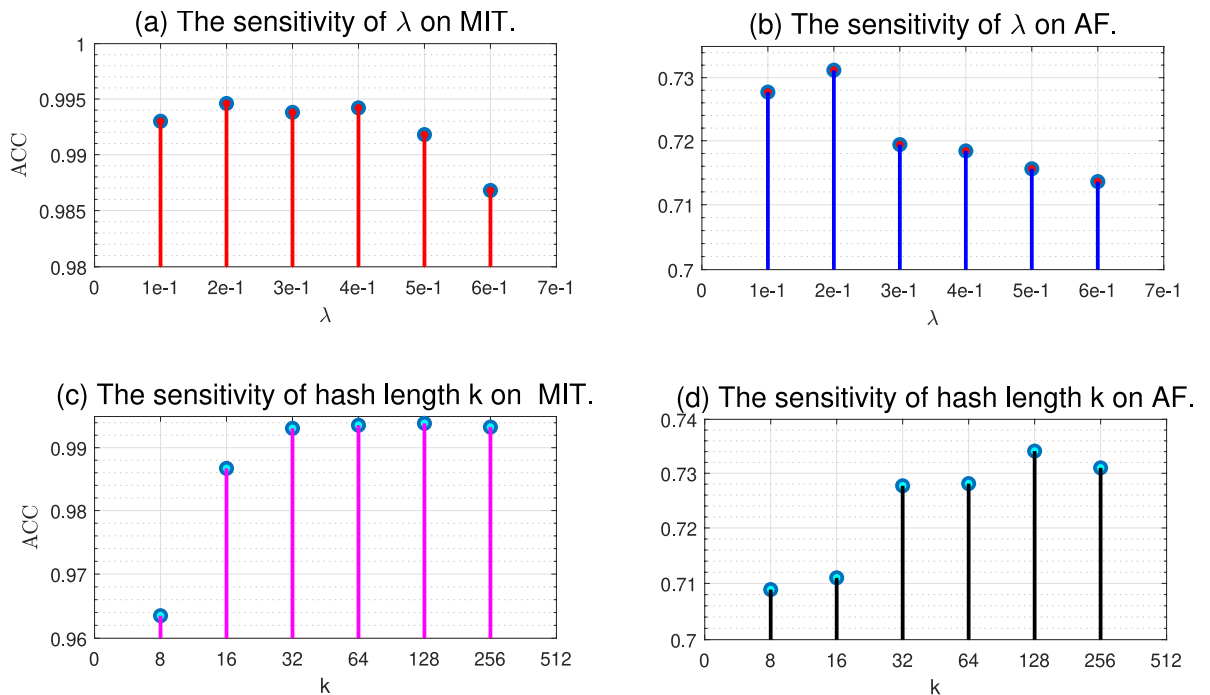


Fig. 9. Sensitivity of λ and k on two datasets.

considering the convergence of the algorithm, it is better to choose a smaller step size as far as possible, such as $\iota \leq 1e - 3$. The aforementioned two parameters solely impact the algorithm’s convergence speed, without significantly affecting the outcome.

The regularization parameter λ is solely utilized to prevent the linear projection function from overfitting. All experiments in this paper are based on $\lambda = 1e - 1$, but from the sensitivity experiment of λ in Fig. 9, it can be seen that taking $\lambda = 2e - 1$ is the best choice on the two datasets.

The length of the hash code plays a crucial role in capturing signal semantics. A longer hash code length typically encompasses richer semantic information. As illustrated in Fig. 9, increasing the hash code length k enhances the semantic similarity graph, which in turn significantly boosts the retrieval performance of the FastGCN network. However, longer hash codes come at the cost of increased computational complexity. Despite Fig. 9 indicating that $k = 128$ yields the optimal results, this study opts for $k = 32$ to strike a balance between retrieval effectiveness and computational efficiency.

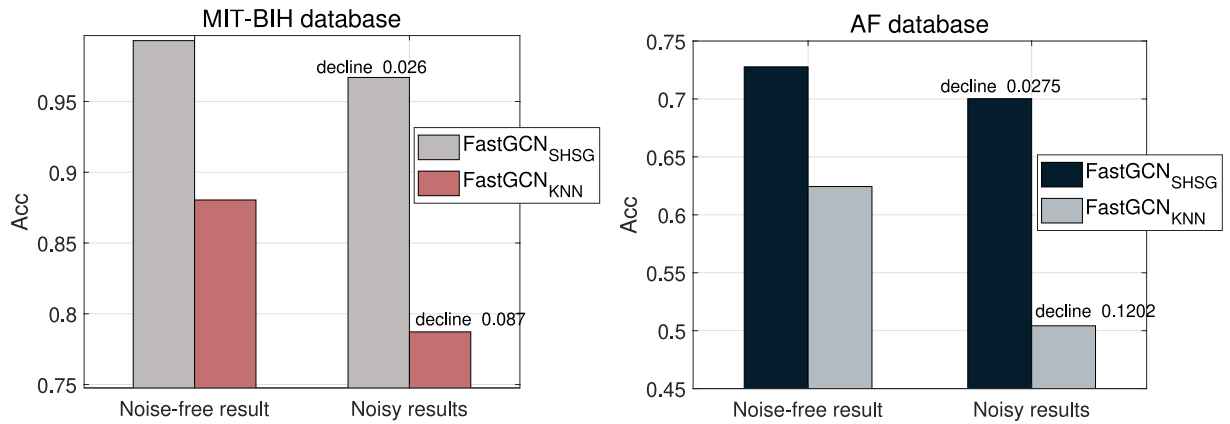


Fig. 10. The comparison of the noise resistance capabilities of SHSG and KNN on two datasets.

Noise sensitivity

Furthermore, to verify the noise resistance of our graph construction method, we introduce Gaussian white noise with a signal-to-noise ratio (SNR) of 10 into each of the two datasets. Subsequently, we conduct a comparative analysis of the original graphs constructed using the KNN method and our proposed method within the FastGCN framework. Additionally, experimental comparisons are performed on FastGCN for the two graph structures built based on Gaussian white noise. The experimental results are shown in Fig. 10. Our results dropped by 3% under noise, while KNN-based results decreased by 11% on MIT-BIH. And Our results experienced a 3-percentage-point drop under noisy conditions, whereas the KNN-based results exhibited a 12-percentage-point decrease on AF. The results demonstrate that our graph learning method possesses the capability of noise mitigation. This is primarily attributed to the fact that hash coding can effectively diminish the impact of noise.

Conclusion

Aiming at the dual dilemma that the learning of existing graph structures over-relies on local topological relations while ignoring global semantic similarity, and that it is difficult to effectively capture node association features in sparse data scenarios, This paper proposes a Semantic Hashing Similarity Graph (SHSG) learning framework based on supervisory mechanism. The framework systematically models the internal feature association and cross-sample semantic similarity of ECG signals through multi-level semantic fusion strategy, and provides a strong discriminant initial graph structure for depth map neural networks. SHSG first constructs a compact hash representation of supervised signals based on semantic consistency constraints of tag supervision. Secondly, a lightweight linear hash function is designed to generate the generalization representation of the unobserved signal. Then, the hash space embedding of the training set and test set samples is integrated to construct a globally traceable hash dictionary. Finally, based on Hamming similarity, the graph topology is generated. In order to verify the effectiveness of the proposed method, experiments are carried out on the double-layer FastGCN architecture. The experimental results verify the dual advantages of the proposed method in terms of feature characterization ability and computational efficiency.

Next, we will focus on the construction of unsupervised semantic similarity graphs and combine them with more advanced graph neural network (GNN) architectures, such as graph attention networks and graph isomorphism networks, to further enhance feature extraction and learning capabilities.

Data Availability

Data is provided within the manuscript or supplementary information files

Analysis of algorithm 1 convergence

The purpose of updating hash codes is to select the optimal vertex on a k -dimensional hypercube network, which does not significantly impact target descent. Therefore, our main focus is on discussing the convergence of rotation factor \mathbb{R} .

Considering that the orthogonal feasible solution $\tilde{r}_i = r_i - \iota \nabla f_i(r_i) \in \text{Span}\{r_i, \nabla f_i(r_i)\}$, which is equivalent to $\tilde{r}_i \in \mathcal{B}(r_i - \iota \nabla f_i(r_i), \iota \|\nabla f_i(r_i)\|_2)$. On this basis, we can derive the following theorem.

Theorem 1 For any $\tilde{r}_i \in \mathcal{B}(r_i - \iota \nabla f_i(r_i), \iota \|\nabla f_i(r_i)\|_2)$, there holds that

$$f_i(\tilde{r}_i) \leq f_i(r_i). \quad (18)$$

Proof For any $\tilde{r}_i \in \mathcal{B}(r_i - \iota \nabla f_i(r_i), \iota \|\nabla f_i(r_i)\|_2)$, we have

$$\langle \tilde{r}_i - r_i, \tilde{r}_i - r_i + 2\iota \nabla f_i(r_i) \rangle \leq 0. \quad (19)$$

That is because

$$\begin{aligned}
 & (\tilde{\mathbf{r}}_i - \mathbf{r}_i)^T (\tilde{\mathbf{r}}_i - \mathbf{r}_i + 2\iota \nabla f_i(\mathbf{r}_i)) \\
 &= \|\tilde{\mathbf{r}}_i - \mathbf{r}_i\|_2^2 + 2\iota (\tilde{\mathbf{r}}_i - \mathbf{r}_i)^T \nabla f_i(\mathbf{r}_i) \\
 &\leq \|\tilde{\mathbf{r}}_i - \mathbf{r}_i\|_2^2 + 2\iota (-\iota \nabla f_i(\mathbf{r}_i))^T \nabla f_i(\mathbf{r}_i) \\
 &= \|\tilde{\mathbf{r}}_i - \mathbf{r}_i\|_2^2 - 2\iota^2 \|\nabla f_i(\mathbf{r}_i)\|_2^2 \\
 &\leq \iota^2 \|\nabla f_i(\mathbf{r}_i)\|_2^2 - 2\iota^2 \|\nabla f_i(\mathbf{r}_i)\|_2^2 \leq 0.
 \end{aligned} \tag{20}$$

Using the second order Taylor expansion of f_i , we have

$$f_i(\tilde{\mathbf{r}}_i) = f_i(\mathbf{r}_i) + \langle \tilde{\mathbf{r}}_i - \mathbf{r}_i, \nabla f_i(\mathbf{r}_i) \rangle + \frac{1}{2} (\tilde{\mathbf{r}}_i - \mathbf{r}_i)^T \nabla^2 f_i(\mathbf{r}_i) (\tilde{\mathbf{r}}_i - \mathbf{r}_i) \tag{21}$$

Since f_i is a convex function of \mathbf{r}_i , the Hessian matrix $\nabla^2 f_i(\mathbf{r}_i)$ is symmetrically positive definite, and then there exists an invertible matrix \mathbf{Q} that satisfies $\nabla^2 f_i(\mathbf{r}_i) = \mathbf{Q}^T \mathbf{Q}$. So $(\tilde{\mathbf{r}}_i - \mathbf{r}_i)^T \nabla^2 f_i(\mathbf{r}_i) (\tilde{\mathbf{r}}_i - \mathbf{r}_i) = \|\mathbf{Q}(\tilde{\mathbf{r}}_i - \mathbf{r}_i)\|_2^2 \leq \rho \|\tilde{\mathbf{r}}_i - \mathbf{r}_i\|_2^2$, where $\rho = \|\mathbf{Q}\|_F^2$. Therefore, in combination with Eqs.(23) (25), we have

$$\begin{aligned}
 f_i(\tilde{\mathbf{r}}_i) &\leq f_i(\mathbf{r}_i) + \langle \tilde{\mathbf{r}}_i - \mathbf{r}_i, \nabla f_i(\mathbf{r}_i) \rangle + \frac{\rho}{2} \|\tilde{\mathbf{r}}_i - \mathbf{r}_i\|_2^2 \\
 &= f_i(\mathbf{r}_i) + \frac{1}{2\iota} \langle \tilde{\mathbf{r}}_i - \mathbf{r}_i, \tilde{\mathbf{r}}_i - \mathbf{r}_i + 2\iota \nabla f_i(\mathbf{r}_i) \rangle - \frac{\iota^{-1} - \rho}{2} \|\tilde{\mathbf{r}}_i - \mathbf{r}_i\|_2^2 \\
 &\leq f_i(\mathbf{r}_i) - \frac{\iota^{-1} - \rho}{2} \|\tilde{\mathbf{r}}_i - \mathbf{r}_i\|_2^2.
 \end{aligned} \tag{22}$$

So as long as the step size $\iota \in (0, \rho^{-1})$, we can derive $f_i(\tilde{\mathbf{r}}_i) \leq f_i(\mathbf{r}_i)$.

Received: 17 February 2025; Accepted: 17 June 2025

Published online: 03 July 2025

References

- Morokuma, S. et al. Prediction of ecg signals from ballistocardiography using deep learning for the unconstrained measurement of heartbeat intervals. *Sci. Rep.* **15**, 999 (2025).
- Lin, H. et al. A new method for heart rate prediction based on lstm-bilstm-att. *Measurement* **207**, 112384. <https://doi.org/10.1016/j.measurement.2022.112384> (2023).
- Sabor, N., Gendy, G., Mohammed, H., Wang, G. & Lian, Y. Robust arrhythmia classification based on qrs detection and a compact 1d-cnn for wearable ecg devices. *IEEE J. Biomed. Health Inform.* **26**, 5918–5929. <https://doi.org/10.1109/JBHI.2022.3207456> (2022).
- Prabhakararao, E. & Dandapat, S. Attentive rnn-based network to fuse 12-lead ecg and clinical features for improved myocardial infarction diagnosis. *IEEE Signal Process. Lett.* **27**, 2029–2033. <https://doi.org/10.1109/LSP.2020.3036314> (2020).
- Li, W., Tang, Y. M., Yu, K. M. & To, S. Slc-gan: An automated myocardial infarction detection model based on generative adversarial networks and convolutional neural networks with single-lead electrocardiogram synthesis. *Inf. Sci.* **589**, 738–750. <https://doi.org/10.1016/j.ins.2021.12.083> (2022).
- Dohare, A. K., Kumar, V. & Kumar, R. Detection of myocardial infarction in 12 lead ecg using support vector machine. *Appl. Soft Comput.* **64**, 138–147. <https://doi.org/10.1016/j.asoc.2017.12.001> (2018).
- Arora, H. & Sinha, R. Multi-modal graph neural networks for physiological signal analysis. In *2025 4th International Conference on Sentiment Analysis and Deep Learning (ICSADL)*, 1202–1208. <https://doi.org/10.1109/ICSADL65848.2025.10933022> (2025).
- Shiyilin, Y., Jie, S., Xin, Y., Xin, C. & Xingxing, W. Ecg arrhythmias classification with a graph bispectrum method. In *2023 International Conference on Intelligent Supercomputing and BioPharma (ISBP)*, 104–108. <https://doi.org/10.1109/ISBP57705.2023.10061314> (2023).
- Chen, J. et al. Graph-enhanced low-resource ecg representation learning for emotion recognition based on wearable internet of things. *IEEE Internet Things J.* **11**, 39056–39068. <https://doi.org/10.1109/IOT.2024.3430297> (2024).
- Chen, Y., Qiu, S., Wang, Z., Zhao, H. & Cao, X. Multiperspective region of spatial temporal graph convolutional shrinkage network for arrhythmia recognition. *IEEE Trans. Instrum. Meas.* **73**, 1–11. <https://doi.org/10.1109/TIM.2024.3376017> (2024).
- Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M. & Monfardini, G. The graph neural network model. *IEEE Trans. Neural Netw.* **20**, 61–80. <https://doi.org/10.1109/TNN.2008.2005605> (2009).
- Clifford, G. D. et al. Af classification from a short single lead ecg recording: The physionet/computing in cardiology challenge 2017. In *2017 Computing in Cardiology (CinC)*, 1–4. <https://doi.org/10.22489/CinC.2017.065-469> (2017).
- Zhang, D., Wu, X.-J. & Yu, J. Discrete bidirectional matrix factorization hashing for zero-shot cross-media retrieval. In *Pattern Recognition and Computer Vision* (eds Ma, H. et al.) 524–536 (Springer, Cham, 2021).
- Gao, X., Chen, Z., Zhang, B. & Wei, J. Deep learning to hash with application to cross-view nearest neighbor search. *IEEE Trans. Circuits Syst. Video Technol.* **35**, 3882–3892. <https://doi.org/10.1109/TCSVT.2023.3273400> (2025).
- Zhang, D. & Wu, X.-J. Robust and discrete matrix factorization hashing for cross-modal retrieval. *Pattern Recognit.* **122**, 108343. <https://doi.org/10.1016/j.patcog.2021.108343> (2022).
- Zhang, D., Wu, X.-J. & Yu, J. Label consistent flexible matrix factorization hashing for efficient cross-modal retrieval. *ACM Trans. Multimed. Comput. Commun. Appl.* <https://doi.org/10.1145/3446774> (2021).
- Zhang, B. et al. Unsupervised dual deep hashing with semantic-index and content-code for cross-modal retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.* **47**, 387–399. <https://doi.org/10.1109/TPAMI.2024.3467130> (2025).
- Zhang, D. & Wu, X.-J. Scalable discrete matrix factorization and semantic autoencoder for cross-media retrieval. *IEEE Trans. Cybern.* **52**, 5947–5960. <https://doi.org/10.1109/TCYB.2020.3032017> (2022).
- Zhang, H. et al. St-rege: A novel spatial-temporal residual graph convolutional network for cvd. *IEEE J. Biomed. Health Inform.* **28**, 216–227. <https://doi.org/10.1109/JBHI.2023.3327025> (2024).

20. Iqbal, S. et al. FusionGcn: An IoT-based novel spatiotemporal graph convolutional network for ECG arrhythmia detection. *IEEE Internet Things J.* <https://doi.org/10.1109/JIOT.2025.3560344> (2025).
21. Zhu, Y. et al. A survey on graph structure learning: Progress and opportunities (2021). [arXiv:2103.03036](https://arxiv.org/abs/2103.03036).
22. Shen, Z., Wang, S. & Kang, Z. Beyond redundancy: Information-aware unsupervised multiplex graph structure learning (2024). [arXiv:2409.17386](https://arxiv.org/abs/2409.17386).
23. Shen, Z. & Kang, Z. When heterophily meets heterogeneous graphs: Latent graphs guided unsupervised representation learning (2024). [arXiv:2409.00687](https://arxiv.org/abs/2409.00687).
24. Chen, Y., Wu, L. & Zaki, M. Iterative deep graph learning for graph neural networks: Better and robust node embeddings. In *Advances in Neural Information Processing Systems* Vol. 33 (eds Larochelle, H. et al.) 19314–19326 (Curran Associates Inc, 2020).
25. Li, M. et al. Gsgsl: Gravity-driven self-supervised graph structure learning. *Inf. Process. Manag.* **61**, 103744. <https://doi.org/10.1016/j.ipm.2024.103744> (2024).
26. Liu, F. & Liu, W. Graph neural network based multi-instance learning with graph structure learning. In *2024 7th International Conference on Artificial Intelligence and Big Data (ICAIBD)*, 505–510. <https://doi.org/10.1109/ICAIBD62003.2024.10604509> (2024).
27. Preparata, F. P. & Shamos, M. I. *Computational Geometry: An Introduction* (Springer, 1985).
28. Bentley, J. L., Stanat, D. F. & Williams, E. The complexity of finding fixed-radius near neighbors. *Inf. Process. Lett.* **6**, 209–212. [https://doi.org/10.1016/0020-0190\(77\)90070-9](https://doi.org/10.1016/0020-0190(77)90070-9) (1977).
29. Zhu, Y., Xu, Y., Yu, F., Wu, S. & Wang, L. Caggn: Cluster-aware graph neural networks for unsupervised graph representation learning (2020). [arXiv:2009.01674](https://arxiv.org/abs/2009.01674).
30. Louizos, C., Welling, M. & Kingma, D. P. Learning sparse neural networks through l_0 regularization (2018). [arXiv:1712.01312](https://arxiv.org/abs/1712.01312).
31. Ortega, A., Frossard, P., Kovacevic, J., Moura, J. M. F. & Vandergheynst, P. Graph signal processing: Overview, challenges, and applications. *Proc. IEEE* **106**, 808–828. <https://doi.org/10.1109/JPROC.2018.2820126> (2018).
32. Cai, J.-F., Candès, E. J. & Shen, Z. A singular value thresholding algorithm for matrix completion. *SIAM J. Optim.* **20**, 1956–1982. <https://doi.org/10.1137/080738970> (2010).
33. Cai, D., He, X., Han, J. & Huang, T. S. Graph regularized nonnegative matrix factorization for data representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**, 1548–1560. <https://doi.org/10.1109/TPAMI.2010.231> (2011).
34. Fan, F., Jing, P., Nie, L., Gu, H. & Su, Y. Sadcmf: Self-attentive deep consistent matrix factorization for micro-video multi-label classification. *IEEE Trans. Multimed.* **26**, 10331–10341. <https://doi.org/10.1109/TMM.2024.3406196> (2024).
35. Fan, F. et al. Dual-domain aligned deep hierarchical matrix factorization method for micro-video multi-label classification. *IEEE Trans. Multimed.* **26**, 2598–2607. <https://doi.org/10.1109/TMM.2023.3301224> (2024).
36. Vaswani, A. et al. Attention is all you need. In *Advances in Neural Information Processing Systems* Vol. 30 (eds Guyon, I. et al.) (Curran Associates Inc, 2017).
37. Kreuzer, D., Beaini, D., Hamilton, W. L., Letourneau, V. & Tossou, P. Rethinking graph transformers with spectral attention. [arXiv:abs/2106.03893](https://arxiv.org/abs/2106.03893) (2021).
38. Dwivedi, V. P., Luu, A. T., Laurent, T., Bengio, Y. & Bresson, X. Graph neural networks with learnable structural and positional representations. [arXiv:abs/2110.07875](https://arxiv.org/abs/2110.07875) (2021).
39. Lu, J., Zhou, J., Chen, Y., Pedrycz, W. & Hung, K.-W. Asymmetric transfer hashing with adaptive bipartite graph learning. *IEEE Trans. Cybern.* **54**, 533–545. <https://doi.org/10.1109/TCYB.2022.3232787> (2024).
40. Jing, P., Sun, H., Nie, L., Li, Y. & Su, Y. Deep multi-modal hashing with semantic enhancement for multi-label micro-video retrieval. *IEEE Trans. Knowl. Data Eng.* **36**, 5080–5091. <https://doi.org/10.1109/TKDE.2023.3337077> (2024).
41. Huang, L., Tao, Y., Qin, C. & Zhang, X. Robust hashing for neural network models via heterogeneous graph representation. *IEEE Signal Process. Lett.* **31**, 2640–2644. <https://doi.org/10.1109/LSP.2024.3465898> (2024).
42. Liu, H., Wei, Y., Yin, J. & Nie, L. Hs-gcn: Hamming spatial graph convolutional networks for recommendation. *IEEE Trans. Knowl. Data Eng.* **35**, 5977–5990. <https://doi.org/10.1109/TKDE.2022.3158317> (2023).
43. Wang, S., Li, C. & Shen, H.-L. Distributed graph hashing. *IEEE Trans. Cybern.* **49**, 1896–1908. <https://doi.org/10.1109/TCYB.2018.2816791> (2019).
44. Wang, Y. et al. Batch: A scalable asymmetric discrete cross-modal hashing. *IEEE Transactions on Knowledge and Data Engineering* **33**, 3507–3519. <https://doi.org/10.1109/TKDE.2020.2974825> (2021).
45. Wang, Y., Chen, Z.-D., Luo, X. & Xu, X.-S. A high-dimensional sparse hashing framework for cross-modal retrieval. *IEEE Trans. Circuits Syst. Video Technol.* **32**, 8822–8836. <https://doi.org/10.1109/TCSVT.2022.3195874> (2022).
46. Wang, L., Zareapoor, M., Yang, J. & Zheng, Z. Asymmetric correlation quantization hashing for cross-modal retrieval. *IEEE Trans. Multimed.* **24**, 3665–3678. <https://doi.org/10.1109/TMM.2021.3105824> (2022).
47. Yang, F. et al. Semantic preserving asymmetric discrete hashing for cross-modal retrieval. *Appl. Intell.* **53**, 15352–15371. <https://doi.org/10.1007/s10489-022-04282-w> (2023).
48. Gao, B., Liu, X. & Xiang Yuan, Y. Parallelizable algorithms for optimization problems with orthogonality constraints. *SIAM J. Sci. Comput.* **41**, 1949–1983 (2018).
49. Gao, B., Liu, X., Chen, X. & Yuan, Y.-X. A new first-order algorithmic framework for optimization problems with orthogonality constraints. *SIAM J. Optim.* **28**, 302–332. <https://doi.org/10.1137/16M1098759> (2018).
50. Sakoe, H. & Chiba, S. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoust. Speech Signal Process.* **26**, 43–49. <https://doi.org/10.1109/TASSP.1978.1163055> (1978).
51. Chen, J., Ma, T. & Xiao, C. Fastgcn: Fast learning with graph convolutional networks via importance sampling. *CoRR* [abs/1801.10247](https://arxiv.org/abs/1801.10247) (2018). [arXiv:1801.10247](https://arxiv.org/abs/1801.10247).
52. Yang, W., Si, Y., Wang, D. & Gong, Z. A novel method for identifying electrocardiograms using an independent component analysis and principal component analysis network. *Measurement* **152**, 107363. <https://doi.org/10.1016/j.measurement.2019.107363> (2019).
53. Chen, C., Hua, Z., Zhang, R., Liu, G. & Wen, W. Automated arrhythmia classification based on a combination network of CNN and LSTM. *Biomed. Signal Process. Control* **57**, 101819. <https://doi.org/10.1016/j.bspc.2019.101819> (2020).
54. Yildirim, Ö. A novel wavelet sequence based on deep bidirectional LSTM network model for ECG signal classification. *Comput. Biol. Med.* **96**, 189–202. <https://doi.org/10.1016/j.compbiomed.2018.03.016> (2018).
55. Jiang, Z. et al. Diagnostic of multiple cardiac disorders from 12-lead ECGs using graph convolutional network based multi-label classification. In *2020 Computing in Cardiology*, 1–4. <https://doi.org/10.22489/CinC.2020.135> (2020).

Acknowledgements

This work was partially supported by the National Science Foundation of China (62271293), the Natural Science Foundation of Shandong Province, PR China (ZR2021MF035), the Social Science Planning Project of Shandong Province, PR China (22CYYJ13).

Author contributions

Yixian Fang and Shilin Zhang: concept and design of the study. Yixian Fang wrote the manuscript. Yuwei Ren: data collection and pre-analysis. Yixian Fang and Yuwei Ren: funding acquisition, supervision, Writing - review

& editing. All authors reviewed the manuscript.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1038/s41598-025-07838-1>.

Correspondence and requests for materials should be addressed to Y.F.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025