Harnessing Topological Semantics for Network Topology Diagram Retrieval

Liangyun Sun, Yanfang Zhang, Yang Wei, Zhipeng Zou, Shengwei Zhong, and Chen Gong†

***Abstract*—Network topology diagram retrieval aims to retrieve the topology diagrams with both visual and topological similarities to a query diagram within the domain of network science and engineering, which can be viewed as a specific image retrieval task. However, current image retrieval approaches are mostly designed for natural images which typically identify the similar candidates by measuring the similarity of visual features. Therefore, these approaches are inherently limited in retrieving network topology diagrams, as they struggle to effectively capture the crucial topological semantics (*i.e.*, the semantics characterized by device roles, connection types, and network structures) embedded in network topology diagrams. To address this issue, we propose the Network Topology-aware Retrieval Framework (NTRF) by incorporating network domain knowledge, which emphasizes the functionality conveyed by the topological semantics to enhance retrieval performance. To be specific, beyond the traditional visual retrieval module, we design a Network Topology Encoder (NTE) to capture the topological semantics by simultaneously representing the device roles, connection types, and network structures. Furthermore, we introduce a SubGraph-aware Re-ranking Module (SGRM) to refine the ranking of candidates, which pays attention to the core diagram regions with significant topological semantics. Intensive experimental results on the collected dataset based on Huawei's Product Documentation demonstrate that our NTRF outperforms state-of-the-art image retrieval methods by 2.75%, 6.19%, and 9.97% in terms of Recall@1, Recall@5, and Recall@10, respectively.**

***Index Terms*—Network topology diagram, image retrieval, topological semantics.**

I. Introduction

COMMUNICATION networks have become greatly expanded and complex with the rapid development of wired and wireless technologies, which makes the setup, management, and analysis of communication networks more and more difficult. The communication networks are usually accompanied by a large number of network topology diagrams, which visually represent the structure and configuration of network environments, and are often utilized for network designing, monitoring, and troubleshooting. Practically, network engineers will spend substantial effort in analyzing network topology diagrams and identify similar historical cases for network designing and troubleshooting. Therefore, developing an automatic network topology diagram retrieval system is critical to enhancing the efficiency of network management.

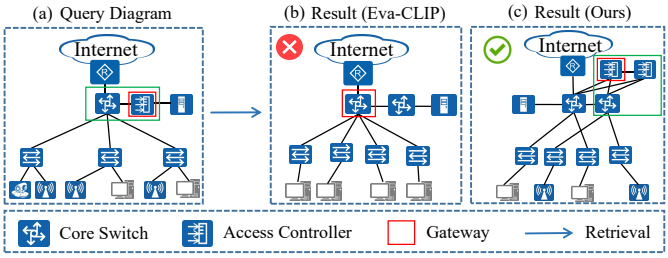

Fig. 1. An example of retrieval results with and without the incorporation of topological semantics. (a) denotes the given query diagram; (b) represents the incorrect retrieval result of Eva-CLIP [1], which considers only visual patterns during retrieval; in contrast, (c) is the correct retrieval result of our method, which harnesses topological semantics for network topology diagram retrieval. The gateways for wireless users are highlighted in the red boxes, and the same gateway configurations are highlighted in the green boxes. The gateways for wireless users in (a) and (c) are access controllers, while the gateway in (b) is consisted of a core switch.

Actually, network topology diagram retrieval can be regarded as an image retrieval problem, in which the target is to find images within a large-scale database that are similar to a given query image. With the advancement of deep learning technologies, numerous deep image retrieval methods have been proposed [2]–[4]. Despite their strong performance in natural images, they can hardly be applied to retrieve network topology diagrams due to their limited ability to capture topological semantics. Different from natural images, network topology diagrams contain specific network topological semantics, which are characterized by device roles, connection types, and network structures. These topological semantics reflect the functionality of communication networks. However, existing image retrieval methods only interpret network topology diagrams as simple visual patterns, which leads to weak topological understanding and reduced retrieval accuracy. For example, although Fig. 1(a) and Fig. 1(b) appear visually similar, they differ significantly in functionality due to their distinct gateway configurations. Specifically, in Fig. 1(a), the access controller serves as the gateway for wireless users, while in Fig. 1(b), the core switch serves as the gateway directly. Obviously, Eva-CLIP [1] overlooks such differences in topological semantics and produces an incorrect result. Meanwhile, Fig. 1(c) shows the correct result retrieved by our method, which explicitly incorporates topological semantics. Specifically, as in Fig. 1(a), the access controllers in Fig. 1(c)

This work was supported in part by the National Natural Science Foundation (NSF) of China under Grant 62336003, Grant 12371510, Grant 62571246, and Grant 62101261.

L. Sun, Y. Zhang, Z. Zou, and S. Zhong are with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: {sly4005630, yanfangzhang, zzp2019}@njust.edu.cn; zhong_sw91@foxmail.com).

Y. Wei is with the School of Software, North University of China, Taiyuan 030051, China (e-mail: csywei@nuc.edu.cn).

C. Gong is with the School of Automation and Intelligent Sensing, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: chen.gong@sjtu.edu.cn).

†Corresponding author: C. Gong (e-mail: chen.gong@sjtu.edu.cn).

also serve as the gateways for wireless users. Therefore, effectively representing topological semantics can facilitate accurate similarity measurement between network topology diagrams, which further improves retrieval performance.

In recent years, communication networks have been naturally modeled as graph data to describe their topological structures. Accordingly, due to the strong capability of representing graph data, Graph Neural Networks (GNNs) [5] have been employed by previous works [6], [7] to model the topology of communication networks. However, these methods still face some challenges in representing and utilizing the topological semantics of communication networks: **(1) Difficulty in representing device roles.** The role of a device is determined by both its type and its position within the network hierarchy. This critical semantic attribute defines the primary functionality of a device (*e.g.*, firewalls in the network egress layer function as gateways and regulate inbound and outbound traffic). However, previous works solely focus on the structural attributes of devices in communication networks and face challenges in capturing the semantic attributes of devices, *i.e.*, their roles. **(2) Absence of connection type representation.** Connections between devices are typically associated with specific types, which reflect the underlying characteristics of data transmission. For example, an Ethernet trunking connection between two devices typically indicates an aggregated data path, while a basic connection suggests a simpler point-to-point transmission. However, existing methods mainly focus on whether a connection exists and ignore the semantic information conveyed by connection types. **(3) Ignorance of core-region importance.** Core regions are the substructures within a communication network, which are essential to deciding the functionality of the overall network [8]. Therefore, core regions contain the essential components of topological semantics in communication networks. Nevertheless, existing methods overlook the importance of core regions as they treat all regions in communication networks equally, which results in suboptimal matching performance.

To address the aforementioned challenges, we propose the **N**etwork **T**opology-aware **R**etrieval **F**ramework (NTRF), which is specially designed for the network topology diagram retrieval. NTRF initially recalls candidates based on their visual patterns and topological semantics, and further incorporates core-region-level similarity to re-rank the candidates. In the initial retrieval stage, we introduce a **N**etwork **T**opology **E**ncoder (NTE) to precisely represent topological semantics. Specifically, NTE simultaneously encodes the role of each device and the connection types as inputs to GNNs. To encode device roles effectively, NTE employs a feature aggregation mechanism to capture device types and their positions within the network hierarchy. In the re-ranking stage, we design a subgraph-aware re-ranking module to pay more attention to the core diagram regions. The subgraph-aware re-ranking module identifies core diagram regions based on device roles and then incorporates global (visual and topological) similarity with core-region-level similarity to refine the ranking of candidates. Benefiting from the effective representation and utilization of network topological semantics, our NTRF can successfully retrieve network topology diagrams related to the query from the database. The contributions of our NTRF are summarized as follows:

- We propose a novel framework for network topology diagram retrieval, termed NTRF, which explicitly incorporates topological semantics to deliver competitive results. To the best of our knowledge, this is the first method designed for network topology diagram retrieval.
- We develop a network topology encoder and a subgraph-aware re-ranking module to jointly capture topological semantics and prioritize core diagram regions, which substantially enhance the effectiveness of retrieval.
- Intensive experimental results demonstrate the superiority of our proposed NTRF. Specifically, it outperforms state-of-the-art image retrieval methods by 2.75%, 6.19%, and 9.97% in terms of Recall@1, Recall@5, and Recall@10, respectively.

II. Related Work

In this section, we review the relevant work, including image retrieval and graph representation learning.

A. Image Retrieval Methods

Image retrieval systems [9]–[11] are tasked with searching large databases [12] to find visual content similar to a query image. Early image retrieval approaches were built upon handcrafted local features [13]. Some studies proposed image retrieval methods that directly relied on local features [13], while others leveraged these features to construct global representations based on the Bag-of-Words model and similar techniques [14]–[17]. Driven by the advances in deep learning, recent methods have enhanced these traditional techniques with deep learning components, including deep local feature-based retrieval [18]–[20], deep local feature aggregation [21]–[23], and deep global feature modeling [24]–[27]. Among them, DELF [18] proposed an attentive local feature for large-scale image retrieval. However, it relied on a single-scale feature map and thus neglected multi-scale objects within images. To address this limitation, DOLG [26] designed a local branch with multi-atrous convolutions to simulate the image-pyramid trick utilized in SIFT [13]. To capture global visual content, Babenko et al. [24] used top-layer CNN activations as holistic image descriptors for retrieval, which were compact and robust to compression. Gordo et al. [25] improved deep global descriptors for instance-level retrieval by cleaning noisy data and using triplet-based training. Jang et al. [27] learned global features through a self-supervised product quantization framework, which enables representation learning even without labels. However, compared with large-scale vision pre-trained models, traditional deep learning-based image retrieval methods are limited in visual representation due to their small model capacity and task-specific training.

More recently, large-scale vision pre-training models [28]–[30] have exhibited strong visual understanding capabilities and shown great potential in image retrieval [31], [32]. Despite their strong performance on natural images, these methods face the dilemma of performance degradation in network topology diagram retrieval. Specifically, these methods interpreted network topology diagrams as simple visual patterns, and thus,

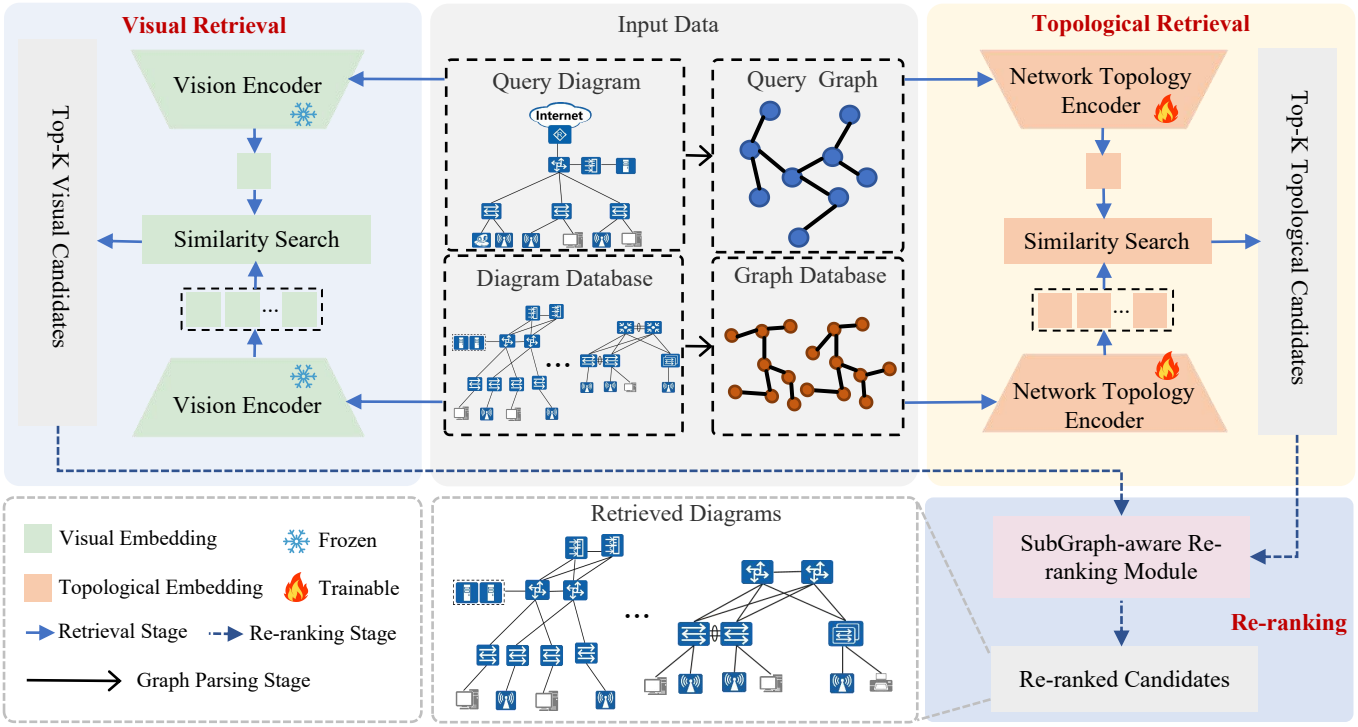

Fig. 2. The overview of our proposed NTRF. Given a query diagram, we initially perform visual retrieval and topological retrieval to identify the candidate diagrams from the diagram database. To incorporate topological semantics into the topological retrieval stage, we first parse the network topology diagram as a graph and then design a network topology encoder to capture its topological semantics. We introduce a subgraph-aware re-ranking module to pay more attention to the critical topological semantics in the core diagram regions, which refines the ranking of candidates and improves retrieval performance.

the inherent topological semantics of the diagrams are neglected. In contrast, we develop a Network Topology Encoder (NTE) to capture the topological semantics by simultaneously representing the device roles, connection types, and network structures. Furthermore, we introduce a SubGraph-aware Re-ranking Module (SGRM) to prioritize critical topological semantics, thereby enhancing the performance of network topology diagram retrieval. As a result, it is critical to capturing topological semantics for network topology diagram retrieval.

B. Graph Representation Learning

Graph representation learning has witnessed substantial advancements [33], with GNNs [34]–[36] emerging as the most prominent architecture in this domain. GNNs are commonly categorized into spectral [37], [38] and non-spectral methods [39], [40]. Bruna et al. [37] introduced a method for performing graph convolution in the spectral domain. In contrast, non-spectral methods perform convolution by aggregating information from nodes located near each other in the spatial structure of the graph. Veličković et al. [5] assigned unique weight matrices to nodes according to their degree. Meanwhile, Hamilton et al. [41] introduced trainable aggregator functions designed to capture and integrate information from neighboring nodes for learning graph representations. Shi et al. [42] applied multi-head self-attention to graph-structured data, replacing traditional aggregation with learnable attention weights. They also incorporated edge features and gated residual connections to further enhance representation learning. Brody et al. [43] enhanced the original Graph Attention Network. They made the attention mechanism adjust dynamically according to the features of both the source and target nodes.

Recently, Graph Contrastive Learning (GCL) [44], [45] has emerged as a prevailing paradigm for self-supervised graph representation learning. Early studies primarily design the contrastive objective around simple augmentations like random edge dropping and feature masking. Furthermore, GraphCL [46] and InfoGCL [47] introduce node dropping and subgraph sampling to expand the augmentation strategies, which simultaneously perturb both graph structure and node attributes. There are also some methods that use unique augmentation strategies. For instance, MVGRL [48] utilizes graph diffusion to generate contrastive views, while SimGCL [49] injects uniform noise directly into the embedding space. Subsequently, to overcome the limitations of uniform random perturbations, recent research has increasingly shifted towards adaptive data augmentation strategies [50], [51]. Specifically, GCS [52] screens graph substructures using gradient-based contrastive saliency, while NCLA [53] learns graph augmentations through a multi-head attention mechanism.

Developments in graph representation learning have made graph retrieval much more effective [54]–[56]. In general, these methods matched substructures of one object with those of another to maximize the overall similarity between the two objects [54]. However, these methods cannot handle the task of topology diagram retrieval accurately, due to their limited capability in capturing topological semantics. To be specific, different from conventional graph data, network

topology diagrams contain topological semantics, which are characterized by device roles, connection types, and network structures. Hence, designing a topology-specific model is of great significance for capturing network topological semantics.

III. METHOD

This section briefly introduces the proposed network topology diagram retrieval framework (*i.e.*, NTRF). As shown in Fig. 2, our NTRF works in a two-stage paradigm, which consists of an initial retrieval stage and a subsequent re-ranking process. During the initial retrieval stage, we conduct a dual-path retrieval process to identify the candidates from the diagram database based on both visual and topological information. Notably, we design the **N**etwork **T**opology **E**ncoder (NTE) to capture the network topological semantics. Furthermore, we design the **S**ub**G**raph-aware **R**e-ranking **M**odule (SGRM) to re-rank the initially recalled candidates.

A. Dual-path Retrieval

In this stage, we perform both visual retrieval and topological retrieval to identify the candidates from the network topology diagram database.

Visual Retrieval. To measure the visual similarity between network topology diagrams, we project both the query diagram and the diagrams in the database into a shared vector space for comparison. Specifically, we leverage a pre-trained visual encoder to transform network topology diagrams into visual feature vectors, and then utilize the cosine similarity metric to assess their closeness. The visual similarity score S_v of each diagram in the database is calculated as

$$S_v^m = \frac{\mathbf{v}_q}{\|\mathbf{v}_q\|} \cdot \frac{\mathbf{v}_m}{\|\mathbf{v}_m\|}, \quad m = 1, \ldots, M, \tag{1}$$

where $\mathbf{v}_q = \Phi(d_q)$ and $\mathbf{v}_m = \Phi(d_m)$ denote the visual embeddings of the query diagram d_q and the m-th diagram d_m in the database, respectively. Besides, M denotes the total number of diagrams in the database, and $\Phi(\cdot)$ is the pre-trained visual encoder (*e.g.*, Eva-CLIP [1]). Based on the visual similarity scores S_v^m, we select the top-K best-matched diagrams from the diagram database to form the candidate set $\mathcal{D}_v = \{d_v^1, \ldots, d_v^K\}$.

Although the visual retrieval process can identify candidates that are visually similar, it may still miss candidates with similar network functionality to the query diagram. This limitation arises because the network functionality is overlooked in the visual retrieval. To face this challenge, we propose incorporating network domain knowledge and topological semantics to convey the network functionality during the subsequent topological retrieval.

Topological Retrieval. The topological semantics of a network topology diagram reflects the functionality of the communication network, which is characterized by the device roles, connection types, and network structures. The functionality of communication networks includes data transmission, device interconnection, and security control. Networks with different topological semantics typically exhibit different functionalities. Therefore, evaluating the topological semantic similarity of network topology diagrams is crucial for improving retrieval accuracy. To this end, we first design a network topology encoder to effectively represent the network topological semantics. Subsequently, we identify the candidates similar to the query diagram based on the topological semantics.

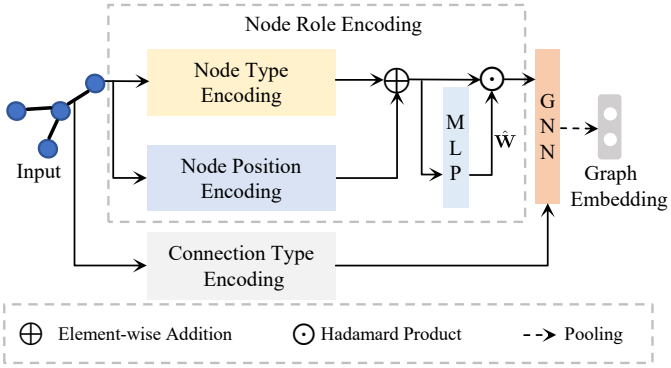

Fig. 3. Illustration of the network topology encoder. The input is the graph that is parsed from a network topology diagram. Each node in the input has a textual attribute that indicates its type. The matrix $\hat{\mathbf{W}}$ consists of $\hat{N}$ modulation vectors, each of which lies in $\mathbb{R}^F$.

1) Network Topology Encoder. Firstly, we parse the network topology diagram as a graph, in which the nodes are devices and the edges are connective relationships among them (See Appendix A). We use $\mathcal{G} = (\mathcal{I}, \mathcal{E})$ to denote a graph, where $\mathcal{I}$ is the set of nodes, $\mathcal{E}$ is the set of edges. Furthermore, as shown in Fig. 3, the NTE is designed to simultaneously represent the device roles, connection types, and network structures.

In communication networks, the role of each device is jointly determined by its type and its position within the network hierarchy. Accordingly, NTE encodes both the type of each node and its position within the network hierarchy, and then fuses them to represent the role of the node. To encode node types, we first apply the template matching algorithm [57] to identify device types in network topology diagrams. Each node is then assigned a textual attribute that indicates its type. Subsequently, NTE incorporates a pre-trained language model (*e.g.*, BERT [58]) to obtain the encoding of the type of each node, which is formulated as

$$\mathbf{t}_i = \mathrm{PLM}(T_i) \in \mathbb{R}^F, \quad i \in \mathcal{I}, \tag{2}$$

where T_i is the text attribute of node i. The function $\mathrm{PLM}(\cdot)$ is the output of the pre-trained language model. We denote the dimension of $\mathbf{t}_i$ by F.

Moreover, the devices of the same type may assume different roles at different positions in the network hierarchy. For example, unlike firewalls in the core layer, firewalls in the network egress layer also function as gateways and regulate inbound and outbound traffic. Therefore, NTE further encodes the position in the network hierarchy of each node. Specifically, we first identify the network layer to which node i belongs based on its type and the types of its connected nodes. Then, the position of node i is set to the layer ID of its corresponding network layer. Subsequently, the positional encoding $\mathbf{p}_i \in \mathbb{R}^F$ can be expressed as

$$\mathbf{p}_i = [p_i^0, \ldots, p_i^c, \ldots, p_i^{F-1}] \tag{3}$$

and

$$p_i^c = \mathrm{PE}(pos_i, c), \quad i \in \mathcal{I}, c = 0, \ldots, F-1, \tag{4}$$

where pos_i denotes the position in the network hierarchy of node i, and c denotes the element index of $\mathbf{p}_i$. The sinusoidal encoding function $\mathrm{PE}(\cdot,\cdot)$ [59] is defined as

$$\mathrm{PE}(pos, c) = \begin{cases} \sin\left(\frac{pos}{10000^{\frac{c}{F}}}\right), & \text{if } c \bmod 2 = 0, \\ \cos\left(\frac{pos}{10000^{\frac{c-1}{F}}}\right), & \text{if } c \bmod 2 = 1. \end{cases} \tag{5}$$

Afterwards, we design a feature aggregation mechanism to represent the role of each node, *i.e.*, the device role. Specifically, we first fuse the type encoding and positional encoding via element-wise addition to obtain an initial representation $\mathbf{r}_i$, which is defined as

$$\mathbf{r}_i = \mathbf{t}_i + \mathbf{p}_i, \quad i \in \mathcal{I}. \tag{6}$$

Subsequently, inspired by [60], a modulation vector is learned based on the initial representation of the role of each node, which allows the model to focus on task-relevant features and suppress irrelevant ones. The process of learning this modulation vector $\mathbf{w}_i$ can be described as

$$\mathbf{w}_i = f(\mathbf{r}_i) \in \mathbb{R}^F, \quad i \in \mathcal{I}, \tag{7}$$

where $f(\cdot)$ is a 2-layer MLP. The process of representing the role of node i is then formulated as

$$\tilde{\mathbf{r}}_i = \mathbf{w}_i \odot \mathbf{r}_i \in \mathbb{R}^F, \quad i \in \mathcal{I}, \tag{8}$$

where $\odot$ is the Hadamard product.

Communication networks exhibit diverse connection types, including basic connections, stacking connections, and Ethernet trunking connections. These connection types reflect how data is transmitted between devices and further influence the topological semantics of the network. Therefore, connection types are further encoded. Specifically, we first identify the connection type between two nodes based on their types and the number of connections between them (see Appendix A). Subsequently, a representation vector of the connection type between two nodes is obtained by using one-hot encoding. The representation vector of the connection type between node i and node j is defined as $\mathbf{e}_{ij} \in \mathbb{R}^{F_e}$, where F_e is the dimension of $\mathbf{e}_{ij}$. Finally, we employ the GNN [43] to capture the network structure. Meanwhile, we use $\tilde{\mathbf{r}}_i$ and $\mathbf{e}_{ij}$ as the node and edge inputs of the GNN, respectively, so that the network topological semantics is effectively represented. Specifically, in a L-layer GNN, we update the representation $\mathbf{h}_i^{(l)} \in \mathbb{R}^F$ of node i by

$$\mathbf{h}_i^{(l)} = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(l)} \mathbf{W}^{(l)} \mathbf{h}_j^{(l-1)}\right), \quad l = 1, \ldots, L, \tag{9}$$

where $\sigma(\cdot)$ is the activation function, and $\mathbf{h}_i^{(0)} = \tilde{\mathbf{r}}_i$. We denote the set of the neighbors of node i by $\mathcal{N}(i)$, and denote the attention coefficients by $\alpha_{ij}^{(l)}$, which can be computed as

$$\alpha_{ij}^{(l)} = \frac{\exp\left(E(\mathbf{h}_i^{(l-1)}, \mathbf{h}_j^{(l-1)})\right)}{\sum_{j' \in \mathcal{N}(i)} \exp\left(E(\mathbf{h}_i^{(l-1)}, \mathbf{h}_{j'}^{(l-1)})\right)}, \tag{10}$$

where

$$E(\mathbf{h}_i^{(l-1)}, \mathbf{h}_j^{(l-1)}) = \mathbf{a}^\top \mathrm{g}\Big(\mathbf{W}^{(l)}\big(\mathbf{h}_i^{(l-1)} + \mathbf{h}_j^{(l-1)}\big) + \mathbf{W}_e^{(l)} \mathbf{e}_{ij}\Big). \tag{11}$$

In the above equation, $\mathrm{g}(\cdot)$ is a LeakyReLU activation function and $\mathbf{a} \in \mathbb{R}^{F'}$ is a learnable weight vector. The dimension of $\mathbf{a}$ is denoted by F'. The learnable linear transformation matrices for node and edge features are denoted by $\mathbf{W} \in \mathbb{R}^{F' \times F}$ and $\mathbf{W}_e \in \mathbb{R}^{F' \times F_e}$, respectively. Furthermore, NTE represents the network topological semantics by aggregating information from all nodes in the graph through a pooling operation. The global embedding $\mathbf{z} \in \mathbb{R}^F$ of the graph can be defined as

$$\mathbf{z} = \mathrm{POOL}\left(\{\mathbf{h}_i^{(L)} \mid i \in \mathcal{I}\}\right), \tag{12}$$

where $\mathrm{POOL}(\cdot)$ denotes the global average pooling operation, and L is the index of the last GNN layer.

Consequently, the structure of NTE has been established, which captures the network topological semantics by simultaneously representing device roles, connection types, and network structures.

2) Topological Similarity Search. In this step, we employ NTE to capture the network topological semantics and then recall candidates that are topologically similar to the query. We first employ a graph contrastive learning framework to train NTE. For a given graph $\mathcal{G}$, two correlated views are generated as a positive pair by randomly masking or replacing the nodes that correspond to terminal devices (detailed in Appendix E). This operation stems from the observation that terminal devices have minimal influence on the network topological semantics. Meanwhile, negative pairs are generated from the other $N-1$ augmented graphs within the same mini-batch as in [60]. To be specific, for an anchor view of the n-th graph, the augmented views of the remaining $N-1$ graphs are regarded as negatives. The contrastive objective is defined as

$$\ell_n = -\log \frac{\exp\left(\mathrm{sim}(\mathbf{z}_{n,i^*}, \mathbf{z}_{n,j^*})/\tau\right)}{\sum_{n'=1,\, n' \neq n}^{N} \exp\left(\mathrm{sim}(\mathbf{z}_{n,i^*}, \mathbf{z}_{n',j^*})/\tau\right)}, \tag{13}$$

where N denotes the mini-batch size, and τ is the temperature parameter. The indices of two views augmented from the original graph are denoted by i^* and j^*. The function $\mathrm{sim}(\cdot,\cdot)$ denotes cosine similarity, which is defined as $\mathrm{sim}(\mathbf{z}_{n,i^*}, \mathbf{z}_{n,j^*}) = \mathbf{z}_{n,i^*}^\top \mathbf{z}_{n,j^*} / \|\mathbf{z}_{n,i^*}\| \|\mathbf{z}_{n,j^*}\|$. Subsequently, the NTE serves to capture the network topological semantics, which is then utilized for topological similarity search. The topological similarity score S_g of each diagram in the database is calculated as

$$S_g^m = \frac{\mathbf{z}_q}{\|\mathbf{z}_q\|} \cdot \frac{\mathbf{z}_m}{\|\mathbf{z}_m\|}, \quad m = 1, \ldots, M, \tag{14}$$

where $\mathbf{z}_q$ and $\mathbf{z}_m$ denote the topological embeddings of the query diagram and the diagram in the database, respectively. Furthermore, based on S_g^m, we perform topological similarity search and select the top-K most topological similar diagrams from the diagram database, *i.e.*, $\mathcal{D}_g = \{d_g^1, \ldots, d_g^K\}$.

In the above retrieval stage, we first capture both visual and topological information to characterize the visual similarity and network functionality of diagrams, respectively. This

strategy ensures that the candidate diagrams and the query diagram are similar in terms of visual or topological semantics, which enhances the thoroughness of the retrieval process. To rank those candidates more precisely, we further introduce a subgraph-aware re-ranking module to enhance the attention to the similarity of core diagram regions.

B. SubGraph-aware Re-ranking Module

In this stage, we further incorporate the core-region level similarity to refine the ranking of candidates obtained by our dual-path retrieval mechanism. Specifically, the primary functionality of a communication network is determined by the core device and its adjacent devices, as the core device assumes the most essential role in the network (see Appendix E for core device identification). For example, the core switch manages high-speed data switching to maintain stable network performance. Therefore, we use the $\hat{k}$-order ego graph [61] of the node corresponding to the core device to characterize the core region of the network. Subsequently, we compute the core-region-level similarity score for each candidate as

$$S^k_{core} = \frac{\mathbf{z}^q_{core}}{\|\mathbf{z}^q_{core}\|} \cdot \frac{\mathbf{z}^k_{core}}{\|\mathbf{z}^k_{core}\|}, \quad k = 1, 2, 3, \ldots, 2K, \tag{15}$$

where $\mathbf{z}^q_{core}$ and $\mathbf{z}^k_{core}$ denote the topological embeddings of the core regions of the query diagram and the k-th candidate, respectively. The final similarity score S_f of each candidate is then calculated as

$$\begin{aligned} S_f(d) = \omega_1 \frac{\mathbb{I}[d \in \mathcal{D}_v]}{\mu + R_v(d)} + \omega_2 \frac{\mathbb{I}[d \in \mathcal{D}_g]}{\mu + R_g(d)} \\ + \omega_3 \frac{\mathbb{I}[d \in \mathcal{D}_v \cup \mathcal{D}_g]}{\mu + R_{\text{core}}(d)}, \end{aligned} \tag{16}$$

where $R_v(d)$, $R_g(d)$, and $R_{core}(d)$ denote the rank of candidate d when sorted by the visual similarity score, the topological similarity score, and the core-region-level similarity score, respectively. The parameters ω_1, ω_2, and ω_3 are non-negative weighting factors. The indicator function is denoted by $\mathbb{I}[\cdot]$, and μ is a smoothing parameter used to reduce the dominance of highly ranked candidates [62]. The rationale behind adopting this Reciprocal Rank Fusion (RRF) strategy is to address the inherent discrepancy in score distributions between the topological retrieval branch and the visual retrieval branch. By relying on relative rankings rather than absolute scores, the reciprocal weighting mechanism naturally aligns these inconsistent distributions, ensuring robust fusion and the improved retrieval accuracy. Finally, we sort the candidates based on the final similarity score and select top-$\hat{K}$ candidates as the final retrieval results.

In summary, the proposed NTRF suggests that exploring topological semantics can improve the accuracy of network diagram retrieval in addition to the commonly used visual retrieval. Therefore, the candidates obtained by the initial dual-path retrieval are similar to the query diagram either visually or topologically. Furthermore, by considering the core diagram regions with crucial topological semantics, NTRF emphasizes the network functionality similarity of core diagram regions to refine the ranking of candidates and further improve the final retrieval accuracy.

IV. Experiment

In this section, we conduct intensive experiments on **N**etwork **T**opology **D**iagram **R**etrieval (NTDR) dataset to evaluate the effectiveness of our proposed NTRF.

A. Dataset Preparation

The NTDR dataset comprises 1.5k network topology diagrams, which are collected from Huawei's Product Documentation[1]. We convert all of these network topology diagrams into graphs for parsing their topological semantics. More details about converting network topology diagrams into graphs are provided in Appendix A.

Among the 1.5k typical network topology diagrams, 0.5k diagrams are used to train NTE. The remaining 1k diagrams are used as test examples. Among these test examples, 40 diagrams serve as query diagrams, and the remaining diagrams form the network topology diagram database. During the data annotation stage, we manually annotated the dataset by considering both the visual patterns and topological semantics of network topology diagrams. To be specific, we annotated the ten most similar network topology diagrams in the database for each query diagram.

B. Experimental Setup

Evaluation Metrics. We utilize the standard metric Recall@K to evaluate retrieval performance. The Recall@K is defined as the proportion of the relevant items retrieved within the top-K results relative to the total number of relevant items for a given query. Mathematically, it can be defined as

$$\text{Recall@}K = \frac{1}{U} \sum_{u=1}^{U} \frac{|R_u \cap P_u(K)|}{|R_u|}, \tag{17}$$

where U is the total number of queries, R_u denotes the set of relevant items in the database for the u-th query, and $P_u(K)$ represents the top-K items returned by the retrieval system for the u-th query.

Implementation Details. We compute the visual embedding for the network topology diagrams with a frozen Eva-CLIP vision encoder (Eva-CLIP-8B) [1]. We use the pooled embeddings from the final layer as visual features to compute the cosine similarity between diagrams by using the FAISS library [15]. Our NTE adopts a two-layer GATv2 [43] as the backbone network and employs Adam [63] as the optimizer with a weight decay of 1×10^{-5}. The NTE is trained for 1000 epochs with a learning rate of 1×10^{-4} and a batch size of 32. The learning rate is reduced by a factor of 0.5 using a ReduceLROnPlateau[2] scheduler if the validation loss stays stagnant for 5 consecutive epochs. The value of $\hat{k}$ is set to 2, which reflects the size of the core region. We set the temperature parameter τ in the contrastive learning framework to 0.4. Additionally, the weighting factors in Eq. (16) are configured as $\omega_1 = 5/11$, $\omega_2 = 5/11$, and $\omega_3 = 1/11$. The

[1]https://support.huawei.com/hedex/hdx.do?docid=EDOC1100407960&id=index

[2]https://docs.pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html

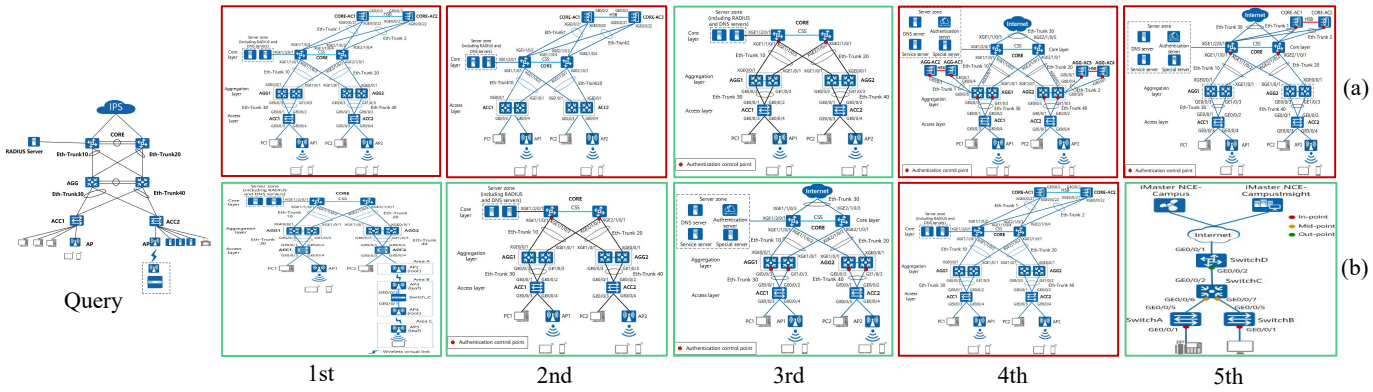

Fig. 4. Examples of retrieval results from different methods: (a) Eva-CLIP and (b) our proposed NTRF. The diagrams outlined in green are relevant to the query, whereas those outlined in red are not. Among the top-5 candidates retrieved by our method, four of them are relevant to the query, whereas Eva-CLIP retrieves only one relevant candidate.

smoothing parameter μ is set to 60, as suggested in [62]. In the initial retrieval process, each retrieval path yields a total of 20 candidates, *i.e.*, K is set to 20. After re-ranking, the top 10 candidates are selected as the final retrieval results, *i.e.*, $\hat{K}$ is set to 10. A pre-trained BERT [58] model is adopted in NTE to encode node types. The NTE is trained on a single NVIDIA GeForce RTX 4090 GPU with 24 GB of memory.

C. Main Result

In this section, we conduct intensive experiments on NTDR dataset to evaluate the performance of our proposed NTRF against vision-based baseline methods and graph-based baseline methods. Vision-based baseline methods retrieve network topology diagrams by using only visual features, whereas graph-based baseline methods rely only on graph representations of these diagrams.

The vision-based baseline methods include SPQ [27], CLIP (ViT-H/14) [64], Eva-CLIP (Eva-CLIP-8B) [1], and SigLIP 2 (ViT-So/14) [32]. As shown in Table I, compared with the state-of-the-art vision-based baseline methods, our NTRF achieves the improvements of 2.75%, 6.19%, and 9.97% in terms of Recall@1, Recall@5, and Recall@10, respectively. This strong performance benefits from the effective integration of the topological semantics during the retrieval process.

Furthermore, our NTRF demonstrates substantial improvements across all evaluation metrics when compared with graph-based baseline methods, including GCN [39], GTN [42], GATv2 [43], WWL [54], and GMN [55]. Among these methods, GCN [39], GTN [42], and GATv2 [43] are classical methods for graph representation, whereas WWL [54] and GMN [55] are representative methods for graph retrieval.

Moreover, we conduct a comparison with the joint retrieval framework that integrates graph-based baseline methods and Eva-CLIP. As shown in Table I, our proposed NTRF outperforms the Eva-CLIP+GCN by 2.55%, 4.78%, and 9.89% in terms of Recall@1, Recall@5, and Recall@10, respectively. These results support that, compared with graph-based baseline methods, our proposed NTE achieves effective representation of device roles and connection types, which leads to the improved performance in network topology diagram retrieval.

TABLE I
RESULTS (% RECALL) OF DIFFERENT BASELINE METHODS. THE SYMBOL "↑" INDICATES THAT LARGER VALUES ARE BETTER. THE BEST RESULTS ARE INDICATED IN **BOLD**.

Type	Method	Recall@K (↑)		
		K=1	K=5	K=10
Graph-based	WWL [54]	1.49	12.50	29.76
	GMN [55]	2.20	15.19	27.43
	GCN [39]	3.16	15.05	30.13
	GTN [42]	2.53	11.28	22.11
	GATv2 [43]	3.40	13.94	26.46
Vision-based	SPQ [27]	0.25	5.47	8.65
	CLIP [64]	4.59	19.53	36.76
	Eva-CLIP [1]	4.06	23.32	41.38
	SigLIP 2 [32]	3.18	14.81	32.71
Joint	Eva-CLIP+WWL	4.75	22.80	43.44
	Eva-CLIP+GMN	5.91	25.91	39.79
	Eva-CLIP+GCN	4.26	24.73	41.46
	Eva-CLIP+GATv2	5.08	22.46	41.17
	Eva-CLIP+GTN	5.29	20.12	33.88
	NTRF (ours)	**6.81**	**29.51**	**51.35**

TABLE II
ABLATION STUDY OF NTE AND SGRM IN TERMS OF RECALL (%). THE SYMBOL "✓" INDICATES THE MODULE IS USED. THE SYMBOL "↑" INDICATES THAT LARGER VALUES ARE BETTER. THE BEST RESULTS ARE INDICATED IN **BOLD**.

NTE	SGRM	Recall@K (↑)		
		K=1	K=5	K=10
		4.06	23.32	41.38
✓		6.06	28.21	48.95
✓	✓	**6.81**	**29.51**	**51.35**

D. Ablation Study

In this section, we first validate the effectiveness of the NTE and SGRM. The results are presented in Table II. Initially, based on the visual retrieval framework, we incorporate NTE to further introduce topological semantics into the retrieval process, which leads to performance improvements of 2%, 4.89%, and 7.57% in terms of Recall@1, Recall@5, and

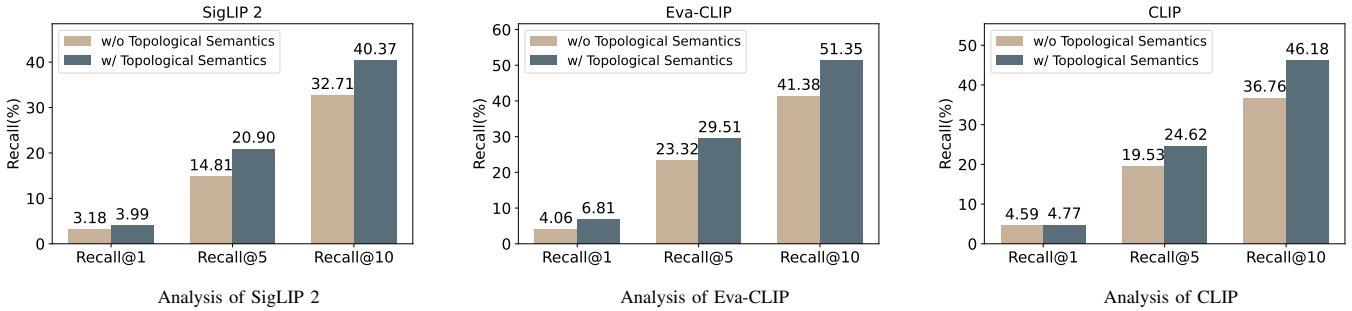

Fig. 5. Analysis of retrieval performance across different visual encoders. The abbreviation "w" denotes the retrieval process that incorporates topological semantics, whereas "w/o" denotes the process with the setting where topological semantics are absent.

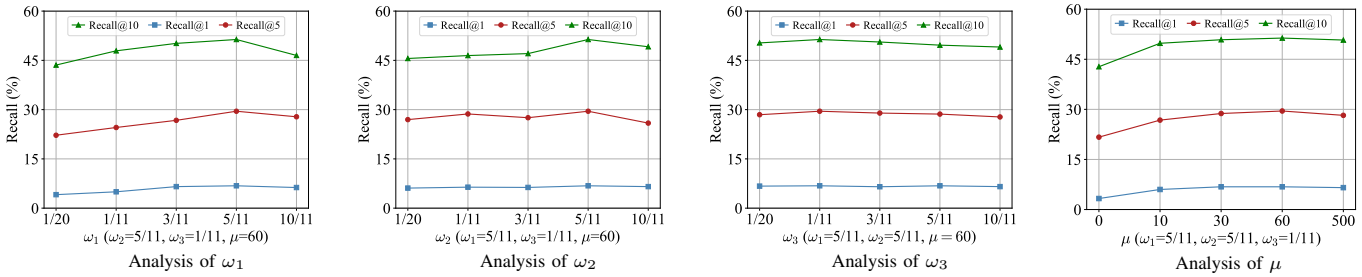

Fig. 6. Parametric sensitivities of ω_1, ω_2, ω_3, and μ in Eq. (16). During the variation of one weight parameter, the remaining three are kept constant.

TABLE III
ABLATION STUDY OF THE KEY COMPONENTS IN NTE IN TERMS OF RECALL (%). THE SYMBOL "↑" INDICATES THAT LARGER VALUES ARE BETTER. THE BEST RESULTS ARE INDICATED IN **BOLD**.

Method	**Recall@K (↑)**		
	K=1	K=5	K=10
w/o NRE	4.91	23.01	45.56
w/o CTE	4.66	23.13	44.69
w/o MVL	4.38	23.40	41.40
NTRF (ours)	**6.81**	**29.51**	**51.35**

Recall@10, respectively. Next, we further incorporate SGRM to pay attention to the core diagram regions, which leads to notable enhancements in retrieval performance across all evaluation metrics. Specifically, the introduction of SGRM improves the performance by 0.75%, 1.3%, and 2.4% on recall@1, recall@5, and recall@10, respectively.

We also investigate the effectiveness of different key components of our NTE, including node role encoding (*i.e.,* NRE), connection type encoding (*i.e.,* CTE), and modulation vector learning (*i.e.,* MVL). From Table III, we can observe that the absence of any component leads to a significant decrease in retrieval accuracy. The underlying rationale is that NRE and CTE are essential for capturing device roles and connection types, which determine the primary functionality of a device and the underlying characteristics of data transmission, respectively. Furthermore, MVL is critical for adaptively emphasizing task-relevant features. Consequently, the absence of these modules hinders the accurate capture of network topological semantics, resulting in the degradation of retrieval performance.

E. Visualization of Retrieval Results

In this section, we adopt Eva-CLIP as the comparative model in our visualizations due to its superior performance when compared with other vision-based baseline methods. As shown in Fig. 4, our method demonstrates superior retrieval performance to Eva-CLIP. In particular, among the top-5 candidates retrieved by our method, four of them are relevant to the query, whereas Eva-CLIP retrieves only one relevant candidate. Although the second candidate in Fig. 4(a) is structurally similar to the query from the perspective of visuality, it differs significantly in topological semantics and functionality. Specifically, the second candidate in Fig. 4(a) represents a two-level network that lacks an aggregation layer, whereas the query corresponds to a three-level network that includes an aggregation layer. In practice, the functionalities of two-level network architectures significantly differ from those of three-level architectures. Additionally, the first, third, and fifth candidates in Fig. 4(a) differ in functionality from the query due to differences in the deployment patterns of access controllers. To be specific, the query does not include any access controller, whereas access controllers are deployed in the first, third, and fifth candidates in Fig. 4(a). Generally, differences in the deployment pattern of an access controller can result in variations in gateway configurations, which in turn lead to differences in network functionality. Therefore, this visualization result further validates the effectiveness of our proposed NTRF in harnessing topological semantics.

F. Effectiveness of Topological Semantics

In this section, we provide additional experimental results to further validate the effectiveness of the topological semantics. To be specific, we analyze the retrieval performance of the

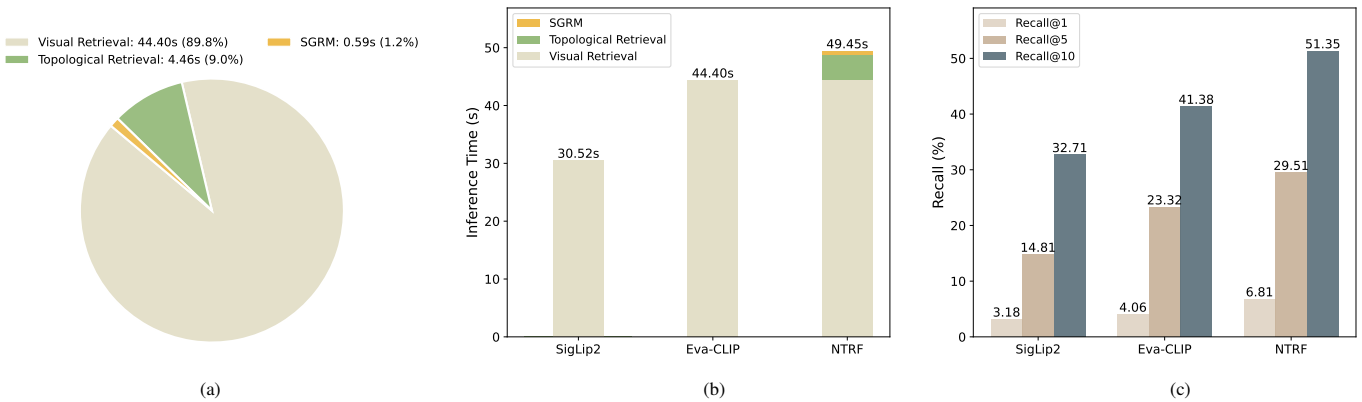

Fig. 7. Analysis of the inference time of NTRF. (a) Proportion of inference time for each stage in NTRF; (b) Inference time comparison of NTRF and baseline methods; (c) Performance comparison of NTRF and baseline methods.

proposed NTRF across multiple representative pre-trained visual encoders. These visual encoders include CLIP (ViT-H/14) [64], Eva-CLIP (Eva-CLIP-8B) [1], and SigLIP 2 (ViT-So/14) [32], all of which have demonstrated strong performance on natural image retrieval tasks. As illustrated in Fig. 5, the incorporation of topological semantics leads to significant improvements in retrieval performance across multiple visual encoders, when compared with the setting where topological semantics are absent. This result further confirms the effectiveness of incorporating topological semantics into the retrieval process. Moreover, it demonstrates that our method remains effective across various visual encoders, which reflects the plug-and-play capability of our proposed NTRF.

G. Parametric Sensitivity

There are four trade-off parameters in our NTRF, including ω_1, ω_2, ω_3, and μ in Eq. (16). To analyze parameter sensitivities, we vary one parameter while keeping the remaining parameters fixed during retrieval. From Fig. 6, we can observe that the performance of our method remains stable across different parameter settings, so these parameters are easy to tune in practical use.

H. Inference Time

In this section, we analyze the inference time of NTRF. Specifically, we compare the inference time of our proposed NTRF with various baseline methods and analyze how each stage contributes to the total inference time. As shown in Fig. 7(a), the inference time of our proposed topological retrieval module and SGRM accounts for only 9.0% and 1.2% of the total inference time, respectively. Moreover, as shown in Fig. 7(b) and Fig. 7(c), our method achieves a substantial performance improvement with only a small additional inference cost. To be specific, our method achieves an improvement in terms of recall@10 from 41.38% to 51.35% compared with Eva-CLIP, with only about a 10% increase in inference time. All these experimental results further validate that our method achieves a favorable trade-off between inference time and retrieval performance.

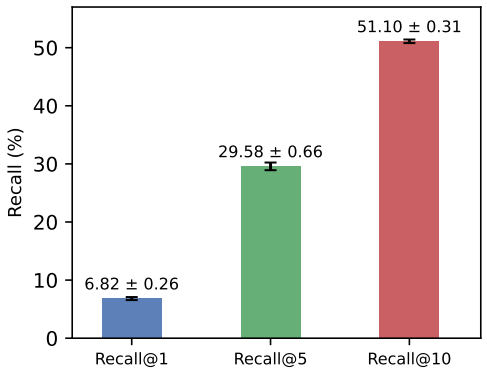

Fig. 8. Analysis of the stability of NTRF. Results are averaged over three independent runs. For each run, the NTRF is trained with a different seed.

I. Stability of NTRF

To validate the stability of NTRF in achieving promising performance, NTRF is trained by using three different random seeds. Fig. 8 presents the results, which include the average and standard deviation of Recall@1, Recall@5, and Recall@10 of our method. We can observe that NTRF consistently exhibits small standard deviations across different training conditions, indicating that its performance is stable.

V. Conclusion

In this paper, we propose a novel method termed "**N**etwork **T**opology-aware **R**etrieval **F**ramework" (NTRF), which incorporates network domain knowledge to harness topological semantics and enhance the performance of network topology diagram retrieval. Specifically, we design a **N**etwork **T**opology **E**ncoder (NTE) to capture topological semantics, which is introduced to measure the similarity between network topology diagrams along with visual features. Furthermore, we develop a **S**ub**G**raph-aware **R**eranking **M**odule (SGRM) to refine the ranking of candidates by emphasizing the critical topological semantics embedded in core diagram regions. In a nutshell, our method effectively captures and utilizes network topological semantics. As a result, NTRF achieves state-of-the-art performance compared with representative baseline

methods. To the best of our knowledge, this is the first method specifically designed for network topology diagram retrieval. In future work, we aim to extend our approach to other graph-centric tasks, such as flowchart retrieval. Furthermore, we intend to improve our diagram parsing pipeline to accurately process unseen icon styles.

REFERENCES

[1] Q. Sun, J. Wang, Q. Yu, Y. Cui, F. Zhang, X. Zhang, and X. Wang, "Eva-clip-18b: Scaling clip to 18 billion parameters," *arXiv preprint arXiv:2402.04252*, 2024.

[2] C.-X. Li, D. Zhang, Z. Hu, and X.-J. Wu, "Modality fused class-proxy with knowledge distillation for zero-shot sketch-based image retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, 2025.

[3] L. Tang, Y. Lv, D. Ye, Y. He, Z. Liu, and C. Xie, "Towards a universal, transferable and robust adversarial perturbation framework against deep hashing-based facial image retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, 2025.

[4] Y. Xu, J. Wei, Y. Bin, Y. Yang, Z. Ma, and H. T. Shen, "Set of diverse queries with uncertainty regularization for composed image retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 10, pp. 10494–10506, 2024.

[5] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proceedings of the International Conference on Learning Representations*, pp. 1–12, 2018.

[6] W. Jiang, "Graph-based deep learning for communication networks: A survey," *Computer Communications*, vol. 185, pp. 40–54, 2022.

[7] Z. Li, X. Wang, L. Pan, L. Zhu, Z. Wang, J. Feng, C. Deng, and L. Huang, "Network topology optimization via deep reinforcement learning," *IEEE Transactions on Communications*, vol. 71, no. 5, pp. 2847–2859, 2023.

[8] X. Han, Q. Huangpeng, Q. Gao, Y. Fu, and X. Duan, "Study of data center communication network topologies using complex network propagation model," *Frontiers in Physics*, vol. 11, p. 1174099, 2023.

[9] D. An, X. Zhang, D. Hao, R. Zhao, and Y. Zhang, "Privacy-preserving image retrieval based on thumbnail-preserving visual features," *IEEE Transactions on Circuits and Systems for Video Technology*, 2025.

[10] L. Ma, X. Luo, Y. Shi, F. Meng, Q. Wu, and H. Hong, "Optimal transport quantization based on cross-x semantic hypergraph learning for fine-grained image retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, 2025.

[11] J. Li, W. K. Wong, L. Jiang, X. Fang, S. Xie, and Y. Xu, "Ckdh: Clip-based knowledge distillation hashing for cross-modal retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 7, pp. 6530–6541, 2024.

[12] C. Zheng, Z. Shi, R. Miao, W. Liu, T. Yang, B. Cui, and S. Uhlig, "Answering subset query over multi-attribute data streams using hyper-uss," *IEEE Transactions on Knowledge and Data Engineering*, 2025.

[13] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," in *Proceedings of the International Conference on Machine Learning*, pp. 91–110, 2004.

[14] Sivic and Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1470–1477, 2003.

[15] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3304–3311, 2010.

[16] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Proceedings of the European Conference on Computer Vision*, pp. 304–317, 2008.

[17] H. Jgou, F. Perronnin, M. Douze, J. Snchez, P. Prez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1704–1716, 2012.

[18] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, "Large-scale image retrieval with attentive deep local features," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3456–3465, 2017.

[19] M. Hosseinzadeh and Y. Wang, "Composed query image retrieval using locally bounded features," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3596–3605, 2020.

[20] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[21] M. Teichmann, A. Araujo, M. Zhu, and J. Sim, "Detect-to-retrieve: Efficient regional aggregation for image search," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5109–5118, 2019.

[22] G. Tolias, T. Jenicek, and O. Chum, "Learning and aggregating deep local descriptors for instance-level recognition," in *Proceedings of the European Conference on Computer Vision*, pp. 460–477, 2020.

[23] P. Weinzaepfel, T. Lucas, D. Larlus, and Y. Kalantidis, "Learning super-features for image retrieval," *arXiv preprint arXiv:2201.13182*, 2022.

[24] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," in *Proceedings of the European Conference on Computer Vision*, pp. 584–599, 2014.

[25] A. Gordo, J. Almazan, J. Revaud, and D. Larlus, "End-to-end learning of deep visual representations for image retrieval," *International Journal of Computer Vision*, vol. 124, no. 2, pp. 237–254, 2017.

[26] M. Yang, D. He, M. Fan, B. Shi, X. Xue, F. Li, E. Ding, and J. Huang, "Dolg: Single-stage image retrieval with deep orthogonal fusion of local and global features," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11772–11781, 2021.

[27] Y. K. Jang and N. I. Cho, "Self-supervised product quantization for deep unsupervised image retrieval," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12085–12094, 2021.

[28] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

[29] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11975–11986, 2023.

[30] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *Proceedings of the International Conference on Machine Learning*, pp. 19730–19742, 2023.

[31] Y. Chen, L. Meng, W. Peng, Z. Wu, and Y.-G. Jiang, "Comp: Continual multimodal pre-training for vision foundation models," *arXiv preprint arXiv:2503.18931*, 2025.

[32] M. Tschannen, A. Gritsenko, X. Wang, M. F. Naeem, I. Alabdulmohsin, N. Parthasarathy, T. Evans, L. Beyer, Y. Xia, B. Mustafa, *et al.*, "Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features," *arXiv preprint arXiv:2502.14786*, 2025.

[33] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proceedings of the International Conference on Machine Learning*, pp. 1263–1272, 2017.

[34] B. Samanta, A. De, G. Jana, V. Gómez, P. Chattaraj, N. Ganguly, and M. Gomez-Rodriguez, "Nevae: A deep generative model for molecular graphs," *Journal of Machine Learning Research*, vol. 21, no. 114, pp. 1–33, 2020.

[35] J. Zhao, X. Wang, C. Shi, B. Hu, G. Song, and Y. Ye, "Heterogeneous graph structure learning for graph neural networks," in *Proceedings of the American Association for Artificial Intelligence*, pp. 4697–4705, 2021.

[36] P. Han, P. Zhao, C. Lu, J. Huang, J. Wu, S. Shang, B. Yao, and X. Zhang, "Gnn-retro: Retrosynthetic planning with graph neural networks," in *Proceedings of the American Association for Artificial Intelligence*, pp. 4014–4021, 2022.

[37] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013.

[38] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 3844–3852, 2016.

[39] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proceedings of the International Conference on Learning Representations*, pp. 1–14, 2017.

[40] T. Lei, W. Jin, R. Barzilay, and T. Jaakkola, "Deriving neural architectures from sequence and graph kernels," in *Proceedings of the International Conference on Machine Learning*, pp. 2024–2033, 2017.

[41] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 1024–1034, 2017.

[42] Y. Shi, Z. Huang, S. Feng, H. Zhong, W. Wang, and Y. Sun, "Masked label prediction: Unified message passing model for semi-supervised classification," in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 1548–1554, 2021.
[43] S. Brody, U. Alon, and E. Yahav, "How attentive are graph attention networks?," in *Proceedings of the International Conference on Learning Representations*, pp. 1–26, 2022.
[44] J. Zhuo, F. Qin, C. Cui, K. Fu, B. Niu, M. Wang, Y. Guo, C. Wang, Z. Wang, X. Cao, *et al.*, "Improving graph contrastive learning via adaptive positive sampling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23179–23187, 2024.
[45] H. Yang, Y. Wang, X. Zhao, H. Chen, H. Yin, Q. Li, and G. Xu, "Multi-level graph knowledge contrastive learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 12, pp. 8829–8841, 2024.
[46] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 5812–5823, 2020.
[47] D. Xu, W. Cheng, D. Luo, H. Chen, and X. Zhang, "Infogcl: Information-aware graph contrastive learning," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 30414–30425, 2021.
[48] K. Hassani and A. H. Khasahmadi, "Contrastive multi-view representation learning on graphs," in *Proceedings of the International Conference on Machine Learning*, pp. 4116–4126, 2020.
[49] J. Yu, H. Yin, X. Xia, T. Chen, L. Cui, and Q. V. H. Nguyen, "Are graph augmentations necessary? simple graph contrastive learning for recommendation," in *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1294–1303, 2022.
[50] S. Li, X. Wang, A. Zhang, Y. Wu, X. He, and T.-S. Chua, "Let invariant rationale discovery inspire graph contrastive learning," in *Proceedings of the International Conference on Machine Learning*, pp. 13052–13065, 2022.
[51] X. Zhang, Q. Tan, X. Huang, and B. Li, "Graph contrastive learning with personalized augmentation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 11, pp. 6305–6316, 2024.
[52] C. Wei, Y. Wang, B. Bai, K. Ni, D. Brady, and L. Fang, "Boosting graph contrastive learning via graph contrastive saliency," in *Proceedings of the International Conference on Machine Learning*, pp. 36839–36855, 2023.
[53] X. Shen, D. Sun, S. Pan, X. Zhou, and L. T. Yang, "Neighbor contrastive learning on learnable graph augmentation," in *Proceedings of the American Association for Artificial Intelligence*, pp. 9782–9791, 2023.
[54] M. Togninalli, E. Ghisu, F. Llinares-López, B. Rieck, and K. Borgwardt, "Wasserstein weisfeiler-lehman graph kernels," in *Proceedings of the Advances in Neural Information Processing Systems*, 2019.
[55] Y. Li, C. Gu, T. Dullien, O. Vinyals, and P. Kohli, "Graph matching networks for learning the similarity of graph structured objects," in *Proceedings of the International Conference on Machine Learning*, pp. 3835–3845, 2019.
[56] Y. Bai, H. Ding, K. Gu, Y. Sun, and W. Wang, "Learning-based efficient graph similarity computation via multi-scale convolutional set matching," in *Proceedings of the American Association for Artificial Intelligence*, pp. 3219–3226, 2020.
[57] M. Hisham, S. N. Yaakob, R. Raof, A. A. Nazren, and N. Wafi, "Template matching using sum of squared difference and normalized cross correlation," in *Proceedings of the Student Conference on Research and Development*, pp. 100–104, 2015.
[58] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the North American Chapter of the Association for Computational Linguistics*, pp. 4171–4186, 2019.
[59] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 6000–6010, 2017.
[60] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, 2018.
[61] Y. Hu, Z. Lei, Z. Zhang, B. Pan, C. Ling, and L. Zhao, "Grag: Graph retrieval-augmented generation," in *Proceedings of the North American Chapter of the Association for Computational Linguistics*, pp. 4145–4157, 2025.
[62] G. V. Cormack, C. L. Clarke, and S. Buettcher, "Reciprocal rank fusion outperforms condorcet and individual rank learning methods," in *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 758–789, 2009.
[63] D. P. Kingma, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations*, pp. 1–15, 2015.
[64] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *Proceedings of the International Conference on Machine Learning*, pp. 8748–8763, 2021.

Liangyun Sun is currently pursuing the Ph.D. degree in the School of Computer Science and Engineering, Nanjing University of Science and Technology, China. He is affiliated with the PCA Lab, the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of the Ministry of Education, and the Jiangsu Key Laboratory of Image and Video Understanding for Social Security. His research interests include computer vision and multimodal large language models.

Yanfang Zhang is currently pursuing the Ph.D. degree in the School of Computer Science and Engineering, Nanjing University of Science and Technology, China. She is affiliated with the PCA Lab, the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of the Ministry of Education, and the Jiangsu Key Laboratory of Image and Video Understanding for Social Security. Her research interests include multimodal large language models and reasoning enhancement.

Yang Wei received her Ph.D. degree from the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China, in 2025. Her current research interests include pattern recognition, incomplete data-based learning, and deep learning.

Zhipeng Zou is currently pursuing the M.S. degree in the School of Computer Science and Engineering, Nanjing University of Science and Technology, China. He is affiliated with the PCA Lab, the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of the Ministry of Education, and the Jiangsu Key Laboratory of Image and Video Understanding for Social Security. His research interests include out-of-distribution detection and trusted machine learning.

Shengwei Zhong (Member, IEEE) received the B.E. degree in information countermeasure technology and the M.S. and Ph.D. degrees in electronics and communication engineering from Harbin Institute of Technology, Harbin, China, in 2013, 2015, and 2020, respectively. She was an Exchange Ph.D. Student visiting the Remote Sensing Signal and Image Processing Laboratory (RSSIPL), University of Maryland, Baltimore County (UMBC), Baltimore, MD, USA, as a Faculty Research Assistant. She is currently an Associate Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. Her research interests include hyperspectral image processing, remote sensing image fusion, and applications.

Chen Gong received his dual doctoral degree from Shanghai Jiao Tong University (SJTU) and University of Technology Sydney (UTS), respectively. Currently, he is a full professor of Shanghai Jiao Tong University. His research interests mainly include machine learning, data mining, and learning-based vision problems. He has published more than 130 technical papers at prominent journals and conferences such as IEEE T-PAMI, JMLR, IJCV, IEEE T-NNLS, IEEE T-IP, ICML, NeurIPS, ICLR, CVPR, ICCV, ECCV, AAAI, IJCAI, ICDM, etc. He serves as the associate editor for IEEE T-CSVT, Neural Networks, and NePL, and also the Area Chair or Senior PC member of several top-tier conferences such as AAAI, IJCAI, ICML, ICLR, ECML-PKDD, AISTATS, ICDM, ACM MM, etc. He won the ICDM Best Student Paper Runner-Up Award, the second prize of Natural Science Award of the Chinese Institute of Electronics, "Excellent Doctorial Dissertation Award" of Chinese Association for Artificial Intelligence, "Wu Wen-Jun AI Excellent Youth Scholar Award", and the Scientific Fund for Distinguished Young Scholars of Jiangsu Province. He was also selected as the "Global Top Chinese Young Scholars in AI" released by Baidu, and "World's Top 2% Scientists" released by Stanford University.