A Unified and Biologically Plausible Relational Graph Representation of Vision Transformers

Yuzhong Chen†, Zhenxiang Xiao†, Graduate Student Member, IEEE, Yu Du, Lin Zhao†, Lu Zhang†, Zihao Wu, Dajiang Zhu†, Tuo Zhang†, Dezhong Yao†, Xintao Hu, Tianning Liu*, Senior Member, IEEE, and Xi Jiang†

Abstract—Vision transformer (ViT) and its variants have achieved remarkable success in various tasks. The key characteristic of these ViT models is to adopt different aggregation strategies of spatial patch information within the artificial neural networks (ANNs). However, there is still a key lack of unified representation of different ViT architectures for systematic understanding and assessment of model representation performance. Moreover, how those well-performing ViT ANNs are similar to real biological neural networks (BNNs) is largely unexplored. To answer these fundamental questions, we, for the first time, propose a unified and biologically plausible relational graph representation of ViT models. Specifically, the proposed relational graph representation consists of two key subgraphs: an aggregation graph and an affine graph. The former considers ViT tokens as nodes and describes their spatial interaction, while the latter regards network channels as nodes and reflects the information communication between channels. Using this unified relational graph representation, we found that: 1) model performance was closely related to graph measures; 2) the proposed relational graph representation of ViT has high similarity with real BNNs; and 3) there was a further improvement in model performance when training with a superior model to constrain the aggregation graph.

Index Terms—Artificial neural network (ANN), biological neural network (BNN), relational graph, vision transformer (ViT).

I. INTRODUCTION

The human brain is a hugely complex, highly recurrent, robust, and remarkably efficient nonlinear neural network [1]. Human behavior, cognition, and neural activities critically depend on the graph structure of large-scale brain connectome [2], [3]. Likewise, the representation and predictive capability of an artificial neural network (ANN), such as a convolutional neural network (CNN), are also closely related to the model's graph structure [4], [5]. You et al. [5] hypothesized that biological neural networks (BNNs) (e.g., macaque's brain network graphs) and ANNs (CNN's relational graphs) might share common graph properties. Recently, neural networks of transformers have received extensive attention, and a variety of vision transformers (ViTs) have been developed rapidly in the computer vision field. Tolstikhin et al. [6] and Yu et al. [7] summarized the message passing in transformers as Token Mixer and Channel multilayer perceptron (MLP). The core idea of various ViT models is the well-designed Token Mixer that controls the information communication between spatial tokens, while most ViTs follow a similar design of the Channel MLP (two fully connected linear with GELU function), for example, the self-attention [8] in ViT [9], the shifted window attention in Swin [10], the token-mixing MLP in Mixer [6], [11], the average pooling in PoolFormer [7], and so on. Essentially, exploring ViT's graph structure (in terms of information communication), its relevance to the representation/predictive performance, and its similarity to BNNs is of great importance for a deeper understanding of ViTs and their wider applications.

The motivation is inspired by the relational graph constructed on CNN models [5] which introduced the pioneering concept of relational graph considering channels of CNN features as nodes. They systematically investigated how the relational graph structure of CNN networks affects their predictive performance. They found sweet spots in the relational graphs of CNN networks, which are similar to the real BNNs reconstructed from brain science data. Therefore, a straightforward question is whether there is a general approach to characterize all transformer-based models using the relational graph structure.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
graph. Built upon the inspiring work in [5], in this article, we proposed the following major innovations.

1) The relational graph in [5] was constructed only on the MLP or CNNs. Here, we will focus on defining and constructing relational graphs on ViT models.

2) The relational graph in [5] only considered the communication between the dimensions or channels of CNNs, thus ignoring the spatial information aggregation. Here, we will consider the spatial patch information communication in ViT as a key factor.

3) The relational graph in [5] only described the network’s graph topology, thus ignoring the influence of parameter weights which can reflect the ability to capture embedding features from images.

Here, we will consider the influence of parameter weights in ViT as a key factor. Overall, our work provides a novel and unified relational graph representation of different ViT architectures for systematic understanding and assessment of model representation performance.

Specifically, inspired by PoolFormer and MLP-Mixer [6], [7], we divided the relational graph in ViT models into aggregation graph and affine graph, where the aggregation graph describes the spatial patch information interaction, while the affine graph reflects the information communication between channels. For example, in the ViT transformer, we considered the self-attention as an aggregation network and the feedforward network (FFN) as an affine network. In this work, we explored five types of widely known ViT models, including ViT [9], DeiT [12], Swin [10], Mixer [6], and PoolFormer [7] as examples.

To systematically explore the advantages of ViT models, we investigated the relationships between two sets of key factors.

1) ViT’s relational graph structure vs its predictive performance. To relate to the model parameters rather than only the model’s graph structure, especially when pretraining/fine-tuning has become a popular methodology for various vision tasks, more attention was paid to the feature extraction ability of the ViT model’s backbone. We evaluated the classification performances of these different models with fixed backbones and graph structures.

2) ViT’s relational graph structure vs those of BNNs. The convolutional and pooling layers in CNNs were directly inspired by the classic notions of simple cells and complex cells in visual neuroscience [13], [14].

Here, we explored the similarities of spatial information communication patterns represented by the proposed relational graphs in ViT models and those in real BNNs represented by brain graphs derived from neuroscience data. Finally, based on our constructed aggregation graph, we propose a relational graph loss for further training and improving the model performance. Overall, the main contributions of our work are as follows.

1) We introduced a novel, unified, and biologically plausible relational graph representation of ViTs for the first time.

2) We discovered the sweet spots of graph measures, including clustering coefficient and average path length, on the relational graphs of various ViT models with significantly improved predictive performances.

3) We proposed an aggregation graph-based graph loss for further improving the performance of the model.

4) We found that the proposed relational graph representation of ViT has high similarity with real BNNs derived from brain science data.

This article is organized as follows. Section II described the most relevant prior work. Section III introduced our framework for constructing relational graphs for various ViTs. Section IV explored the relationship between the graph structure of the model’s relational graph and classification performance. Section V revealed the association of biological and ANNs from the perspective of relational graphs. Section VI introduced a method for optimizing model training based on relational graphs. Sections VII and VIII were the discussion and conclusion.

II. RELATED WORK

A. ViT and Graphs

ViT has developed rapidly in recent years from the ViT [9] to the recent SSA [15], Uniformer [16], ViT-G/14 [17], and so on. With no or little introduction of inductive bias, ViT [17] can achieve satisfactory performance in various computer vision tasks, despite some difficulties in training. Some studies have improved it by effective model training skills or introducing inductive bias. For example, Touvron et al. [12] introduced distillation learning with a distillation token to ViT for data-efficient training. Chen et al. [18] proposed CrossViT for multiscale images. Liu et al. [10] designed a windows attention for greater efficiency by limiting self-attention computation to nonoverlapping local windows. Li et al. [16] combined the advantage of CNNs and ViTs and proposed UniFormer which aggregated local information via convolution at shallow layers and global information via self-attention at deep layers. Tolstikhin et al. [6] introduced a pure MLP structure for computer vision and applied a token mixer MLP for information communication between the tokens. Inspired by Mixer, Yu et al. [7] proposed a PoolFormer that adopted an average pool kernel as the aggregation function instead of self-attention or MLP. One of the core characteristics of these ViTs is to design different attention strategies, which act as different aggregation functions for information communication between tokens. However, there is still a key lack of unified representation of different aggregation strategies in ViT models for systematic understanding and assessment of model representation/predictive performance.

Inspired by the concept of ViT which treats image patches as tokens, some studies have treated image patches as nodes in graphs, thereby using vision graph neural networks (GNNs) which have been widely applied in social networks [19] to handle computer vision tasks [20], [21]. The first and foremost
question is how to build connections between those nodes which we call aggregation graphs. A natural idea is to treat an image as a regular grid graph and each image patch is only connected to its neighborhood image patches, which, however, leads to oversmoothing as the layer goes deep. Consequently, dynamic structures [22], [23], [24], [25] are employed in graph models to mitigate this phenomenon. Motivated by the higher weighting of similar tokens in the transformer, some studies also build the dynamic graph structure through the similarities of feature representations [20], [26]. Similarities between GNNs and ViTs reveal the feasibility of using graphs to characterize ViT models. Inspired by the relational graph [5], we aim to propose a unified relational graph representation of ViTs.

B. Relational Graph

The concept of the relational graph in neural networks was first proposed in [5], where the message exchanges along the graph structure of neural networks. By regarding multiple dimensions or channels of CNN features as nodes in the graph, You et al. [5] converted different random graphs into different structural neural networks in MLP or CNN models. The proposed relational graph has satisfactory generalizability and similarity with real BNNs when compared with the computational graph [4]. More interestingly, You et al. [5] found that the highest-performing model tends to have a similar graph measure which is called the sweet spot. Some studies have employed this approach for the structural design of models. For example, Feng et al. [27] identified that the graphs inside sparse deep neural networks (DNNs) follow the variants of power law distribution, which can be observed in biological or social networks. Therefore, they modeled the sparse DNNs to speed up training, save storage, and learn with fewer samples by truncated power law distribution. Valsesia et al. [28] applied the relational graph (which was called the randomly wired architectures in the article) for effectively scaling up GNNs, mitigating the depth problem, and creating richer representations. However, the proposed relational graph still has certain limitations as discussed in Section I, and we aim to introduce a more general and biologically plausible relational graph representation for ViTs.

C. BNNs and Graphs

In the biological nervous system, the neural circuits [29], which are formed by the interconnection of numerous neurons and carry out various functions, are also known as BNNs. Modeling and interpreting the brain as a complex networked system provides remarkable novel insights into a series of fundamental neuroscience questions, for example, how the brain organizes its structure and function; how spatially distributed brain regions interact during cognitive processes; and how the structural and functional organizations alter in brains with neurological, psychiatric, or psychological diseases [30], [31]. Among these studies, graph theoretical analysis of a neural network describes the topological attributes regarding its adaptability, robustness, general cost and efficiency, and so on. A neural network is typically represented as a graph \( G = (V, E) \), where \( V \) is the set of nodes representing brain regions or neurons, and \( E \) is the set of edges revealing the functional (e.g., synchronization of functional activities), structural (e.g., neural fibers or synapses), or effective (causal relationship) connectivities between node pairs. With such a configuration, graph measures of neural networks can be quantified by borrowing concepts and tools in the field of graph theory. Although the real BNNs derived from brain science data have profoundly inspired the ANNs, for example, the visual circuit inspired the development of convolutional neural networks [32] and a recent study explores the coupling characteristics of the visual representation spaces and semantics between ANNs and BNNs in human brain [33], how these well-performing ANNs such as ViTs are similar to those of BNNs are still largely unexplored and unknown.

III. RELATIONAL GRAPH IN ViTs

A. Structure of ViT Models

As shown in Fig. 1(a), there are three core parts in ViTs: Patch Embedding 1(b), Token Mixer 1(c), and Channel MLP 1(d). The Patch Embedding layer projects the input images into nonoverlapping image tokens as word tokens in NLP. The Token Mixer allows information communication between different spatial locations (image tokens) while the Channel MLP allows information communication between different channels. Therefore, the overall structure of ViTs is represented as

\[
X = \text{PatchEmbed}(\text{Image}) \quad (1)
\]

\[
Y = \text{TokenMixer}(\text{Norm}(X)) + X \quad (2)
\]

\[
Z = \text{ChannelMLP}(\text{Norm}(Y)) + Y \quad (3)
\]

where Norm is the normalization function. Note that there are also some other model designs such as the learnable absolute position embedding and class token in ViTs, the relative position bias in Swin, the distillation token in DeiT, and so on. Here, we only discuss the backbone of the model to simplify the issue.

B. Aggregation and Affine Graphs

Inspired by the PoolFormer model [7], the two core message exchange functions of ViTs are the Token Mixer and Channel MLP. The Token Mixer aggregates information from tokens, while the Channel MLP allows information communication between different channels [6]. Therefore, we defined the relational graph of ViTs as two key subgraphs: Token Mixer Graph and Channel Mixer Graph, and formally annotated them as Aggregation Graph and Affine Graph according to their major roles.

1) Aggregation Graph: By considering each image token as a node in the graph as shown in Fig. 1(c), the token mixer (e.g., self-attention) plays the same role as the aggregation function in GNNs via updating their embedding features by aggregation of the information from its neighborhoods. With this message-passing paradigm, the ViT model can globally integrate information from different regions (tokens) in each layer, whereas for CNN models, message passing is restricted.
to the kernel size. For example, the propagation function between tokens of Self-Attention is written as

\[ x = x + \text{Softmax} \left( \frac{\hat{x} W_q W_k^T \hat{x}^T}{\sqrt{\text{dim}}} \right) \hat{x} W_v \]

where \( \hat{x} = \text{LN}(x) \) and LN is the LayerNorm function and \( W_q, W_k, W_v \) is the set of learning parameters. If we regard each token as a node in the graph, this function is written as \( \text{TokenMixer}(x) = W_T x I \). Therefore, the normalized adjacent matrix of the aggregation graph of Self-Attention is \( \text{Softmax}(\hat{x} W_q W_k^T \hat{x}^T) \). Note that we used the same normalization function as [8] and [9], which can also be written as \( \hat{A} = AD^{-1} \), where \( A = \exp((\hat{x} W_q W_k^T \hat{x}^T)/\sqrt{\text{dim}}) \) and \( D_{ij} = \sum_j A_{ij} \).

2) Affine Graph: By considering each channel as a node in the graph as shown in Fig. 1(c), the channel mixer allows the message passing along channels in the affine graph. Different from the aggregation graph which conveys spatial information about tokens, the affine graph is designed for the information communication between different channels. The channel mixer always contains two full-connection layers and a nonlinear activation function as

\[ x = x + \delta(\text{LN}(x) W_j W_2) \]

where \( \delta \) is the nonlinear activation function such as GELU and ReLU. Therefore, the adjacent matrix of the affine graph is written as \( \text{Softmax}(W_j W_2^T)/\sqrt{\text{dim}}) \). We add the Softmax function to norm the matrix and make the connection matrix nonnegative, in alignment with the aggregation graph.

C. Construction of the Aggregation Relational Graph in ViTs

Since the key difference among different ViT models is the different design of Token Mixer, that is, aggregation graph, we, therefore, mainly focused on a unified aggregation relational graph representation of different ViT models, instead of an affine one in this study. On the other hand, recent parameter-efficient transfer learning methods [34], [35] that only update the projection matrices in self-attention (i.e., Aggregation Graph) also illustrate the importance of the aggregation graph. Specifically, we selected ViT [9], DeiT [12], Swin [10], Mixer [6], and PoolFormer [7] as five representative examples among all ViT models as introduced in Section II-A due to the space limit.

1) ViT/DeiT: The ViT [9] and DeiT [12] models both follow a pure transformer [8] design while more training schemes are adopted in DeiT. Both ViT and DeiT adopt the multihead self-attention (MHSA) as the Token Mixer, which can be described as

\[ \text{Softmax}(x W_q W_k^T x^T) x W_v. \]

In addition, they keep the position information with learnable absolute position embedding \( x = x + P \), where \( P \) is the position embedding. Positional embedding is widely used to represent the positional relations between different tokens in visual, language, as well as graph models. Similar to [36], we also mainly use positional embedding to construct the relational graph between different image patches. For the sake of analysis, we assumed that different tokens are independent of each other, and the information communication between tokens is carried out through position embedding. Therefore, the adjacent matrix of the aggregation graph is written as

\[ \hat{A} = \text{Softmax}(P W_q W_k^T P^T)/\sqrt{\text{dim}}. \]

It is worth noting that we did not use the information of the \( W_v \) matrix due to its role as a feature projection rather than an exchange of spatial information between tokens.

Fig. 1. Aggregation and affine graph in different ViT models. (a) Structure of ViT models. (b) Patch embedding in ViT models. (c) Construction of aggregation graph. (d) Affine graph in ViT models. (All Norm functions are omitted for clarity of expressions.)
2) Swin: Compared with ViTs, Swin [10] proposes the window’s attention and shifted windows’ attention to introduce inductive bias into ViTs. Additionally, relative position bias rather than absolute position embedding is applied in Swin. The Token Mixer in the windows of Swin is

\[
\text{Softmax} \left( x W_o W_i^T x^T / \sqrt{\dim} + B + \text{Mask} \right) x W_v
\]

where \(B\) is the relative position bias and Mask is designed for shifted attention operation, which is window-specific. Therefore, the aggregation graph in each window is

\[
\hat{A} = \text{Softmax} \left( I / \sqrt{\dim} + B + \text{Mask} \right)
\]

where \(I\) is the unit matrix.

3) Mixer: The MLP-Mixer is an MLP-only vision model [6], [11], primarily composed of token-mixing MLP and channel-mixing MLP. The token-mixing MLP acts on the transpose matrix of the image token, thus aggregating spatial information between different tokens while the channel-mixing MLP is the same as the FFN in the transformer. Note that there is no position embedding in the Mixer as the token-mixing MLPs are sensitive to the order of the input tokens. Based on the token mixer defined in (4), the aggregation graph is written as

\[
\hat{A} = \text{Softmax} \left( W / \sqrt{\dim} \right)
\]

where \(W\) is the set of learning parameters of MixingToken in (4).

4) PoolFormer: Based on the MLP-Mixer, Yu et al. [7] proposed a PoolFormer that uses pooling functions to perform information communication between tokens and greatly reduces the computational effort and parameters of the model. Therefore, the aggregation graph of PoolFormer is

\[
a_{i,j} = 1 / K^2, \quad j \in \text{Ner}(i)
\]

where \(K\) is the kernel size and Ner(*) is the neighborhood set.

D. Aggregation Graph Between Layers

The considerable difference in depth and width of different ViT models makes it challenging to adopt a unified framework for model assessment. You [5] assumed multiple features as one node in order and layers as rounds, thus making different models comparable. Inspired by this study, we fixed the aggregation graph size as 14 \times 14+1 nodes and also regarded the layers as rounds. Therefore, for the high-resolution aggregation graph, we downsampled it by

\[
\text{Downsample}(\hat{A}_{xy}) = 1 / K \times \sum_{x} \sum_{y} \hat{A}_{ij}
\]

where \(K\) is the downsampling rate and Ner(*) is the neighborhood set in the high-resolution image. For the low-resolution aggregation graph, we upsampled it by

\[
\text{Upsample}(\hat{A}_{xy}) = 1 / K \hat{A}_{ij}, \quad i = x / K, \quad j = y / K
\]

where \(i, j\) is the index of the raw resolution image and \(x, y\) is the index of the downsampled or upsampled image. The final aggregation graph of a model is

\[
\text{Final} \hat{A} = \prod_{\text{Layers}} \text{Sampled}(\hat{A}).
\]

Note that we normalized (Softmax(Final\(\hat{A}\))) matrix for ease of analysis.

IV. RELATIONAL GRAPH AND MODEL PERFORMANCE

Following the suggestion in [5], we adopted two graph measures, the average path length, and clustering coefficient that characterize the integration and segregation of a network, respectively, to systematically explore: 1) the relationship between these graph measures and downstream task performances of ViT models and 2) the effectiveness of these graph measures as ViT model training indicators from scratch.

A. Graphs Measures

Two representative graph measures including clustering coefficient and average path length are applied to characterize the graph structure of relational graph in transformers consistent with previous work [5]. Notably, these two graph measures have been demonstrated to effectively measure the global organization of large-scale networks [37] and are also widely used in neuroscience [38], [39]. More other graph measures are provided in Appendix A.

1) Clustering Coefficient: The clustering coefficient is a measure of the degree to which nodes in a graph tend to cluster together, which is calculated by

\[
l = \frac{1}{n(n+1)} \sum_{i \neq j} d(v_i, v_j)
\]

where \(d(v_i, v_j)\) denotes the shortest distance between nodes \(v_i\) and \(v_j\).

2) Average Path Length: Average path length, or average shortest path length, is a concept in network topology that is defined as the average number of steps along the shortest paths for all possible pairs of network nodes. It is a measure of the efficiency of information or mass transport on a network. The graph measure of average path length (\(l\)) and clustering coefficient (\(c\)) are calculated by the following equations:

\[
c = \frac{1}{n} \sum_i c_i
\]

\[
c_i = \frac{1}{k_i(k_i - 1)} \sum_{i,k} A_{ij} A_{jk} A_{ki}
\]

\[
k_i = \sum_j A_{ij}
\]

where \(A\) is the adjacent matrix of the graph.

B. Relationship Between Aggregation Graph Measures and Classification Performances

We used image classification as the downstream task and evaluated the performance of the model’s backbone on four different and widely known datasets: ImageNet-1k [40] including 1000 categories of images with more than 1.28 million images, CIFAR10 [41] including 50 K training images and 10 K validation images, Animal10 [42] including 50k training images and 5k validation images, and Flower17 [43] including 80 images (60 for training, 10 for testing, and 10 for validation) for each category.
TABLE I
DETAILED INFORMATION OF ALL PRETRAINED MODELS

| Model                  | Params (M) | Layers | Patch Resolution | Dim   |
|------------------------|------------|--------|------------------|-------|
| vit-tiny-patch16-224   | 5.7        | [12]   | 224x[1/16]       | 192   |
| vit-tiny-patch16-224-in21k | 5.7     | [12]   | 224x[1/16]       | 192   |
| vit-small-patch16-224  | 22.1       | [12]   | 224x[1/16]       | 384   |
| vit-small-patch16-224-in21k | 22.1    | [12]   | 224x[1/16]       | 384   |
| vit-base-patch16-224   | 86.7       | [12]   | 224x[1/16]       | 768   |
| vit-base-patch16-224-in21k | 86.7     | [12]   | 224x[1/16]       | 768   |
| swin-tiny-patch4-window7-224 [10] | 28.3   | [2,2,6,2] | 224x[1/4,1/8,1/16,1/32] | [9,9,9,9] |
| swin-tiny-patch4-window7-224 [10] | 49.6   | [2,2,18,2] | 224x[1/4,1/8,1/16,1/32] | [9,9,9,9] |
| swin-base-patch4-window7-224 [10] | 87.8   | [2,2,18,2] | 224x[1/4,1/8,1/16,1/32] | [9,9,9,9] |
| dep-tiny-patch16-224 [12] | 5.7      | [12]   | 224x[1/16]       | 192   |
| dest-small-patch16-224 [12] | 22.1     | [12]   | 224x[1/16]       | 384   |
| resmlp-12-224 [11]     | 15.4       | [12]   | 224x[1/16]       | 384   |
| resmlp-12-distilled-224 [11] | 15.4    | [12]   | 224x[1/16]       | 384   |
| resmlp-24-224 [11]     | 30         | [24]   | 224x[1/16]       | 384   |
| resmlp-24-distilled-224 [11] | 30      | [24]   | 224x[1/16]       | 384   |
| resmlp-36-224 [11]     | 44.7       | [36]   | 224x[1/16]       | 384   |
| poolformer-s12 [7]     | 11.9       | [2,2,6,2] | 224x[1/4,1/8,1/16,1/32] | [64,128,320,512] |
| poolformer-s24 [7]     | 21.4       | [4,4,12,4] | 224x[1/4,1/8,1/16,1/32] | [64,128,320,512] |
| poolformer-p36 [7]     | 30.9       | [6,6,18,6] | 224x[1/4,1/8,1/16,1/32] | [64,128,320,512] |
| poolformer-m36 [7]     | 56.2       | [6,6,18,6] | 224x[1/4,1/8,1/16,1/32] | [96,192,384,768] |
| poolformer-m48 [7]     | 73.5       | [8,8,24,8] | 224x[1/4,1/8,1/16,1/32] | [96,192,384,768] |

Fig. 2. Relationship between two aggregation graph measures and classification performance of 21 ViT models on four datasets. A larger spot size represents larger model parameters. (a)–(d) Average path length. (e)–(h) Clustering coefficient.

1) Models: We adopted 21 publicly available pretrained models of the five ViTs (ViT, DeiT, Swin, Mixer, and PoolFormer) provided in [44]. The detailed information of all 21 models is listed in Table I. The pretrained vision models were all downloaded from timm.1

2) Settings: We followed the same setting of [45] for better model training.2 An SGD optimizer with a weight decay of 0.0001 was applied. We employed an OneCycleLR [46] learning rate scheduler and the initial (max) learning rate was 0.0005 (0.001) with cosine annealing. The percentage of the cycle spent increasing the learning rate was set to 0.1. The label smoothing of CrossEntropy Loss was set to 0.2 and the batch size was 128/256. The data augmentations included Cutout [47] and RandAugment [48]. All images were resized to 224 × 224 consistent with the pretrained model. All models were with pretrained parameter weights and only the parameters of the classification header would change while the backbone was fixed. For the ImageNet-1k dataset, no parameter was learned and we just tested on the validation dataset as they have been pretrained on the ImageNet-1k. Three ViTs that are pretrained on the different ImageNet-21k were discarded to keep the consistency of the pretrained model.

3) Result: There was a smooth U-shape correlation between the model classification performance and each of the two graph measures in the first three datasets using a second-degree polynomial regression as illustrated in Fig. 2(a)–(c) and (e)–(g). Similarly, in [5], we successfully identified stable sweet spots $C \in [0.809, 0.842]$ and $L \in [1.256, 1.439]$ across the three different datasets as highlighted as red dashed

---

1https://github.com/rwightman/pytorch-image-models
2https://github.com/Alibaba-MIIL/ImageNet21K
lines in Fig. 2(a)–(c) and (e)–(g) corresponding to the best classification performance. As illustrated in Fig. 2(d) and (h), the relationship between the graph measure and classification performance in the fourth dataset (Flower17) exhibited a significant linear correlation instead of a U-shape one. The interpretation was that the Flower17 dataset has only 60 images per category similar to the few-shot learning task. Therefore, these two graph measures of the proposed relational graph not only helped us identify the sweet spot with large sample data, but also reflected the adaptive learning ability of the model with few sample data. The result of another graph measure is in Appendix A. In summary, our results illustrated that the two graph measures can be used as effective indicators of model classification performance.

C. Dynamic Changing Characteristics of Graph Measures During Model Training

1) Setting: To explore the dynamic changing characteristics of the two graph measures during model training and their effectiveness as ViT model training indicators, we trained the ViT-Ti [9] from scratch on ImageNet-1k [49] using the same setting as in Section IV-B2 and recorded the accuracy and graph measures for each epoch. We used Adam Optimizer with a batch size of 800 for better performance.

2) Result: We averaged the graph measures of each layer as the representation of the final graph. A threshold of 1/192 (embedding dim/node numbers) for the affine graph and 1/197 (token numbers) for the aggregation graph was selected to illustrate the dynamic change of graph structure during model training. Note that we concatenated all aggregation graphs along the diagonal rather than through rounds to better highlight the changing trend of the model. We reported the top-1 accuracy on the validation dataset and two graph measures in Fig. 3. Interestingly, except for the initial 25 epochs, there was considerable consistency between the graph measure and the accuracy of the model, indicating that the two graph measures may also serve as a more general and effective indicator of model training.

V. RELATIONAL GRAPH BETWEEN ANNs AND BNNs

In this section, based on the proposed graph characterization method of the relational graph, we explored the similarity of ANNs and BNNs according to the global graph measures.

A. Graph of BNNs

A variety of real BNNs were compared to the ViT relational graphs, including the synaptic connectomes of the anterior (Wrom_279) and posterior (Worm_269) nervous systems of the C. elegans; the synaptic connectomes for the pharyngeal nervous systems of two nematodes with divergent feeding behavior (Worm_54 and Worm_50); the neuronal connectomes of the rat brain revealed by neuronal pathway tracers (Rat_493, Rat_503a and Rat_503b); the macroscopic brain network of cat as reconstructed from tract tracing data (Cat_65); the structural connectome of a macaque monkey derived from a collation of tract tracing studies (Macaque_242); the interareal connectivities of macaque monkeys (Macaque_V91, Macaque_CC91 and Macaque_93) revealed by retrograde tracers. We adopted the publicly available BNN data with approval, and a detailed description of those BNNs is referred to. The Euclidean distance between the graph measures of BNNs and ViTs was calculated as the topological similarity between BNNs and ViTs.

B. Graph Measure Similarity Between ViTs and BNNs

The graph measure similarity between ViT models and BNNs is shown in Fig. 4. Note that the relational graphs of MLP- and CNN-based models are from [5]. We see that the graph measures of ViTs were close to the rat, cat, and macaque's brain neural networks, while those of the MLP- and CNN-based models are similar to worms. The result of another graph measure is in Appendix B. This inspiring result might suggest the superiority of ViTs over MLP/CNN in terms of information communication and exchange efficiency, given that mammalian brains such as rats, cats, and macaques are considered to be much more advanced and optimized than the worm's neural networks.

VI. AGGREGATION GRAPH ON MODEL DESIGN

A. Relational Graph Loss

The result of different models with different aggregation graphs on image classification demonstrates that the closer the model is to the sweet spot, the higher the performance of the model. Therefore, it inspires us to optimize the model design for better performance by guiding the target model and the reference model to have a similar aggregation relational graph. To achieve a closeness between the relational graphs of the target model and the reference model, we define the similarity loss of the relational graph as

$$\mathcal{L}(\theta_t, \theta_r) = \sum_{\text{Layer}} |A_t - A_r|$$

where $\theta_t, \theta_r$ are the parameters of the target model and reference model, and $A_t, A_r$ are the aggregation graph defined in Section III-C. Therefore, the training of the target model could be written as Algorithm 1. Unlike traditional knowledge distillation learning that requires data for in-model information interaction [50], we do not use any additional image data for training, which results in a short training time (several minutes in this study).

3https://neurodata.io/project/connectomes/
Algorithm 1 Algorithm of Training With Relational Graph Loss

\textbf{Require:} Reference model with parameters $\theta_r$ and Target model with parameters $\theta_t$.
1: Initialize learning rate $\eta$;
2: \textbf{while} Training \textbf{do}
3: \hspace{1em} \textbf{Loss} $\leftarrow L(\theta_t, \theta_r)$ in Eq. 19
4: \hspace{1em} $\theta_t \leftarrow \theta_t - \eta \nabla L(\theta_t)$
5: \textbf{end while}

the accuracy of the model improved from an initial 66.43\% to a maximum of 66.51\% and finally decreased to 66.37\%. The results showed that the aggregation graph and the model accuracy were closely related, and higher similarity with a better model would improve the accuracy of the model. However, as the model was further trained, the model accuracy began to decrease, which might be attributed to the lack of classification. An effective suggestion is to stop training when the graph loss is partially reduced, for example, 5\% which was shown in the results to be appropriate in almost all experiments. More results of the training process are shown in Appendix C.

C. Result
To verify the applicability of graph loss, we used the ImageNet-1k pretrained models as the target models and the ImageNet-21k and ImageNet-1k pretrained models as the reference models, respectively. The results in Fig. 6 illustrate that when using a superior model as the reference model, the accuracy of the model will be improved after a short period of training. Conversely, when using a less superior model as the reference model, the accuracy of the model will be reduced, which proves the effectiveness of our method, especially since the model does not use any image data.

VII. DISCUSSION

A. Fine-Tune Partial Versus Full Model
The construction of the proposed relational graph was affected by ViT model parameters. To keep consistent with the graph structure, we froze the parameters of the ViT model’s
backbone and only updated the classification header. To justify whether a fixed backbone would affect the assessment of model performance, we provided the predictive performance on the Flower17 dataset with/without freezing the backbone parameters in Fig. 7. We observed that the performance with/without freezing has a high linear correlation, indicating that it is reasonable to assess the model performance by graph measures with a fixed backbone.

**B. Graph Structures With Different Sampling Efficiencies**

Section IV-C demonstrated that the graph measures changed during model training with the same training setting. We further reported the graph measure changes of ViTs with different training strategies such as training with more datasets or with distillation learning in Fig. 8. Compared with DeiT with more training schemes [12], the graph measures of models pretrained with more data (ImageNet-21k pretrained) had higher similarity with the ImageNet-1k pretrained models. More interestingly, these graph measure changes also corresponded to the model generalizability, which decreased as the average path length increased and the clustering coefficient decreased. These findings were consistent with those in Fig. 2 and further indicated that the graph measures could reflect not only the model classification performance, but also the model sampling efficiency.

**C. Aggregation Graphs Between Layers**

In addition to the overall aggregation graph of different ViT models, we also investigated the aggregation graphs of different layers of the model. As shown in Fig. 9, the clustering coefficient tends to increase with the number of layers [Fig. 9(a)], which was further demonstrated by a visible example (ViT-B) in Fig. 9(b). The clustering coefficients reflected the degree of aggregation of the network. According to previous studies [51], [52], the representation in ViTs became more similar as the layer was deeper which also led to an increase in the clustering coefficient. The clustering coefficients reflected the degree of aggregation of the network that low-level features continuously aggregated into high-level features as the model layers became deeper. Our findings are consistent with previous studies [51] from a graph measure perspective. Therefore, some models use a pyramid architecture, such as Swin and PoolFormer, to improve the performance of the model. Our study demonstrates the plausibility of this phenomenon from the perspective of a relational graph.

**D. Impact on Model Design and Optimization of ANNs**

The proposed relational graph provides a unified paradigm to represent various ViT models, thus enabling effective comparisons among different ANNs and their associations with BNNs. Given that the brains are already highly optimized BNNs and these structural and functional BNNs possess a variety of meaningful graph properties such as the aforementioned sweet spots, we premise that BNNs’ graph characteristics could provide potential guidance for the design and optimization of ANNs, for example, through the structural characteristics of the brain [53] and neural architecture search (NAS) [5], and could offer benchmarks for evaluating those optimized ANNs [54]. Also, additional graph models, abstractions, and common frameworks could be explored and summarized from BNNs and then used to inform and guide the design and optimization of next-generation ANNs [54] and their wide applications in the future.

**E. Limitation**

1) **Limitation to Other Neural Networks:** Currently, our proposed graph representation methodology framework can be effectively applied to transformer-like models for a wide range of tasks such as visual and language models. However, it is still not applicable to other models. The transformer-like models are closely related to GNNs and have a more universally applicable framework, which makes it easier to adopt generalized graph representation methods to characterize the model structure. A more generalized graph representation...
model for all neural networks remains to be explored in future work. One possible approach is to convert the model into a transformer-like model by structural reparameterization. For example, some studies construct the Conv and BN layers parallel to an FC for training and then merge the parameters into the FC for inference [55] which makes it possible to convert a complex network into a simple network with only FC layers, and thus the relational graph could be constructed as Mixer which is an MLP (with two FC layers)-only vision model.

2) Limitation on Image Content: The spatial information exchange in ViT is often associated with the content of images. However, it is difficult to integrate specific data information into the graph construction of networks. Here, we assumed that the patches of images are independent of each other and the spatial relationship is retained only through positional information. However, some models such as Mixer [6] and ResMLP [11] did not use position embedding and might be sensitive to the order of the input tokens [6]. Therefore, integrating image content into the model structure deserves future exploration.

3) Affine Graph in Transformers: In this article, we mainly consider the differences between different models of the Aggregation Graph without focusing on the Affine Graph. One of the main reasons is that almost all ViTs use the same Channels MLP (e.g., two fully connected linear with GeLU) for the communication between channels. Another reason is that there is no suitable solution to the problem of different width dimensions. Unlike visual images which are strongly local, thus allowing us to do sampling operations on images, the interaction between features is often global and it is difficult to unify the size. In the future, there is still a requirement to explore more effective ways to combine these two kinds of graphs.

VIII. Conclusion

In this article, we proposed a novel unified and biologically plausible relational graph representation of ViTs. By decomposing the information transfer graph of the network into the aggregation and affine graphs, our method can be applied to almost all representative visual models. By exploring the differences in graph measures of aggregation graphs in different ViTs, we found that the model performance is closely related to the graph measures, especially when the sample size is small. We also found that the proposed relational graph representation of ViTs has high similarity with real BNNs derived from brain science data. Finally, we also proposed a graph loss to further improve the model performance. Overall, we provided an interpretable and effective way for analyzing ViT models and linking models’ relational graphs with BNNs and offered novel insights on the design of ANNs.

ACKNOWLEDGMENT

The authors would like to thank David Weizhong Liu from the University of Georgia who helped provide constructive comments and revisions to improve the article.
[53] X. Jiang, T. Zhang, S. Zhang, K. M. Kendrick, and T. Liu, “Fundamental functional differences between gyri and sulci: Implications for brain function, cognition, and behavior,” *Psychoradiology*, vol. 1, no. 1, pp. 23–41, Mar. 2021.

[54] X. Yu et al., “Core-periphery principle guided redesign of self-attention in transformers,” 2023, arXiv:2303.15569.

[55] X. Ding, C. Xia, X. Zhang, X. Chu, J. Han, and G. Ding, “RepMLP: Re-parameterizing convolutions into fully-connected layers for image recognition,” 2021, arXiv:2105.01883.

Yuzhong Chen received the B.E. degree from the School of Life Science and Technology, University of Electronic Science and Technology of China, Chengdu, China, in 2020, where he is currently pursuing the master’s degree, under the supervision of Dr. Xi Jiang.

His current research interests include graph convolutional neural networks, vision transformers, and deep-learning-based medical image analysis.

Zhenxiao Xiao (Graduate Student Member, IEEE) received the Biomedical Engineering degree from the School of Life Science and Technology, University of Electronic Science and Technology of China, Chengdu, China, in 2021, where he is currently pursuing the master’s degree, under the supervision of Dr. Xi Jiang.

His research interests include brain functional network analysis and graph convolutional neural networks.

Yu Du received the bachelor’s degree in engineering from the School of Computer Science and Technology, Henan University of China, Kaifeng, China, in 2021, where she is currently pursuing the master’s degree, under the supervision of Dr. Xintao Hu.

Her current main research direction is medical image analysis.

Lin Zhao received the B.E. degree from Northwestern Polytechnical University, Xi’an, China, in 2017. He is currently pursuing the Ph.D. degree with the School of Computing, University of Georgia, Athens, GA, USA, under the supervision of Prof. Tianming Liu.

His current research interests include deep learning and medical image analysis.

Lu Zhang received the B.S. and M.S. degrees in computer science from Northwestern Polytechnical University, Xi’an, China, in 2015 and 2018, respectively. She is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX, USA, under the supervision of Dr. Dajiang Zhu.

Her current research interests include computational neuroscience, brain imaging, and deep learning.

Zihao Wu received the B.E. degree from the School of Microelectronics, Tianjin University, Tianjin, China, in 2017, and the master’s degree from the Electrical Engineering and Computer Science Department, Vanderbilt University, Nashville, TN, USA, in 2020. He is currently pursuing the Ph.D. degree in computer science with the University of Georgia, Athens, GA, USA, under the supervision of Dr. Tianming Liu.

His current research interests include brain-inspired AI and deep-learning-based medical image analysis.

Dajiang Zhu received the Ph.D. degree from the Department of Computer Science, University of Georgia, Athens, GA, USA, in 2014. His current research interests include machine learning, neuroimaging, and computational neuroscience.

Tuo Zhang received the B.E. and Ph.D. degrees from Northwestern Polytechnical University, Xi’an, China, in 2007 and 2015, respectively.

He is currently a Research Associate with Northwestern Polytechnical University. His research interests include machine learning and medical image analysis.

Dezhong Yao received the Ph.D. degree in applied geophysics from the Chengdu University of Technology, Chengdu, China, in 1991, and the Ph.D. degree in biomedical science from Aalborg University, Aalborg, Denmark, in 2005.

He is currently a Professor of neuroengineering and neurodata with the University of Electronic Science and Technology of China (UESTC), Chengdu. He is also the Dean of the Sichuan Institute for Brain Science and Brain-Inspired Intelligence, Chengdu. Since 1991, he has been working at UESTC as a post-doctoral, an Associate Professor in 1992, a Full Professor in 1995, and a Changjiang Scholar Professor in 2006. He was the Founder of the School of Life Science and Technology in 2001 and the Key Laboratory for Neuroinformation of the Ministry of Education, UESTC, in 2009. His research interests include EEG, simultaneous EEG and fMRI, and brain–apparatus communication.

Dr. Yao has been a fellow of AIMBE since 2017. He received the Roy John Award in 2018, the Vice-President of the Chinese Society of Biomedical Engineering in 2014, and the Chairperson of the Chinese EEG Consortium in 2019. He is currently the Chief Editor of Brain-Apparatus Communication and an Associate Editor of IEEE Transactions on Neural Systems and Rehabilitation Engineering.
Xintao Hu received the Ph.D. degree from Northwestern Polytechnical University (NWPU), Xi’an, China, in 2011. He is currently a Professor with the School of Automation, NWPU. His research interests include brain image computing and brain-inspired artificial intelligence.

Tianming Liu (Senior Member, IEEE) is a Distinguished Research Professor of computer science at UGA, Athens, GA, USA. He has published more than 400 papers in these areas, his Google Scholar citations are over 10,000, and his H-index is 54. His research interests are brain imaging, computational neuroscience, and brain-inspired artificial intelligence.

Dr. Liu is an Elected Fellow of the American Institute for Medical and Biological Engineering. He was a recipient of the NIH Career Award and the NSF CAREER Award. He serves on the Editorial Board of multiple journals, including Medical Image Analysis, IEEE Transactions on Medical Imaging, IEEE Reviews in Biomedical Engineering, IEEE/ACM Transactions on Computational Biology and Bioinformatics, and IEEE Journal of Biomedical and Health Informatics.

Xi Jiang received the B.E. degree in automation from Northwestern Polytechnical University, Xi’an, China, in 2009, and the Ph.D. degree in computer science from the University of Georgia, Athens, GA, USA, in 2016. Since 2017, he has been an Associate Professor with the School of Life Science and Technology, University of Electronic Science and Technology of China, Chengdu, China. His research interests include machine-learning/deep-learning-based medical image analysis.

Dr. Jiang was a recipient of the Li Foundation Heritage Prize, USA, for his outstanding research and contributions to the interdisciplinary field of brain science, in 2019.