SATG : Structure Aware Transformers on Graphs for Node Classification

Sumedh B G
Mastercard AI Garage
Sector-26A, Gurugram, India
sumedhbg11@gmail.com

Sanjay Kumar Patnala
Mastercard AI Garage
Sector-26A, Gurugram, India
sanjaykumar.patnala@mastercard.com

Himil Vasava
Mastercard AI Garage
Sector-26A, Gurugram, India
himil.vasava@mastercard.com

Akshay Sethi
Mastercard AI Garage
Sector-26A, Gurugram, India
akshay.sethi@mastercard.com

Sonia Gupta
Mastercard AI Garage
Sector-26A, Gurugram, India
sonia.gupta@mastercard.com

Abstract

Transformers have achieved state-of-the-art performance in the fields of Computer Vision (CV) and Natural Language Processing (NLP). Inspired by this, architectures have come up in recent times that incorporate transformers into the domain of graph neural networks. Most of the existing Graph Transformers either take a set of all the nodes as an input sequence leading to quadratic time complexity or they take only one hop or k-hop neighbours as the input sequence, thereby completely ignoring any long-range interactions. To this end, we propose Structure Aware Transformer on Graphs (SATG), where we capture both short-range and long-range interactions in a computationally efficient manner. When it comes to dealing with non-euclidean spaces like graphs, positional encoding becomes an integral component to provide structural knowledge to the transformer. Upon observing the shortcomings of the existing set of positional encodings, we introduce a new class of positional encodings trained on a Neighbourhood Contrastive Loss that effectively captures the entire topology of the graph. We also introduce a method to effectively capture long-range interactions without having a quadratic time complexity. Extensive experiments done on five benchmark datasets show that SATG consistently outperforms GNNs by a substantial margin and also successfully outperforms other Graph Transformers.

1 Introduction

Graph structured data has found extensive utility across diverse domains, including but not limited to molecular networks, citation networks, and the analysis of relationships within social media. Within a multitude of these domains, Graphs have emerged as the singularly indispensable or even optimal approach for both representing and comprehending complex data patterns. The inherent intricacies of Graphs, stemming from their unique topology and intricate structural nuances, contribute to the challenges associated with deriving meaningful insights from the copious information embedded within their structure and node attributes. Developing efficacious methodologies to extract this wealth

NeurIPS 2023 New Frontiers in Graph Learning Workshop (NeurIPS GLFrontiers 2023).

of knowledge optimally presents a formidable task.

A pre-eminent solution to address the intricacies of graph data structures has been the utilization of Message Passing Graph Neural Networks (MP-GNNs). Among the earliest incarnations of Graph Neural Networks (GNNs) is the work proposed by Gori et al. [2005]. Contemporary advancements have led to the introduction of a plethora of GNN variants, some grounded in convolutional mechanisms for information aggregation, while others leverage attention-based mechanisms. Nonetheless, a common foundational operation across these approaches involves the aggregation of information from immediate neighbors (one-hop) to facilitate the learning of node representations. Despite their pervasive adoption, GNNs are not without limitations. One primary challenge pertains to their capacity to capture long-range dependencies Alon and Yahav [2020]. Additionally, issues such as over-smoothing Yang et al. [2020] and over-squashing Topping et al. [2021] further constrain their effectiveness. However, recent research endeavors Huang et al. [2020], Di Giovanni et al. [2023] have yielded promising techniques that effectively mitigate the challenges of over-smoothing and over-squashing, leading to successfully alleviating them to a substantial extent if not completely.

In the domains of Natural Language Processing (NLP) and Computer Vision, the efficacy of Transformers Vaswani et al. [2017] in achieving cutting-edge performance is well-established. In stark contrast to convolutional methodologies, Transformers operate without a rigid inductive bias dictating the aggregation of information from localized neighborhoods. The distinctive characteristic of Transformers lies in their inherent flexibility in information aggregation. This flexibility emerges from their encoding process, where the boundaries of information aggregation are dynamically determined by the properties of the input. The calculated attention weights govern the weighting assigned to the aggregation of information from various tokens within the input sequence. As a result, the scope of information aggregation remains adaptable, contingent upon the specific characteristics of the input sequence that is fed into the Transformer's encoder. Motivated by these advantages of transformers, there have been several endeavors to transplant this robust architecture into the realm of graph-based representation learning. The fundamental difference is that graph as a data structure has extremely complicated properties regarding their topology and structure. Embedding such intricate properties within the Euclidean space and subsequently channeling them into the transformer framework is not straightforward. The efficacy of the transformer in any downstream task linked to graph data hinges upon its ability to comprehensively capture the underlying structural and topological intricacies intrinsic to the graph. This is inherently accomplished by the attention mechanism, which inherently integrates semantic relationships among nodes.

Within the ambit of introducing structural insight to transformers, three primary methodologies have emerged: positional encodings, node sampling strategies, and explicit biases. While the vanilla transformer Vaswani et al. [2017] employs sinusoidal functions for positional encodings, the graph domain lacks a canonical grid, impeding the derivation of mathematical functions to compute positional encodings. Consequently, a multitude of alternatives have been explored, such as Laplacian eigenvectors Kreuzer et al. [2021], Nguyen et al. [2021], Chen et al. [2022] and random walk probabilities Dwivedi et al. [2021], Ma et al. [2023]. Despite their prevalence, Laplacian eigenvectors face notable limitations due to challenges in eigenvalue multiplicity, eigenvector sign ambiguity, and normalization issues.

However, a crucial observation often overlooked in extant research is that the dot product of positional encodings between two nodes should inherently quantify their structural relation. In other words, nodes with strong structural ties should yield high dot product values for their positional encodings. To transcend the constraints of current positional encoding approaches and to embrace this pivotal observation, we introduce a novel class of positional encodings trained via the Neighborhood Contrastive Loss (NCE loss). This incorporation of loss-driven positional embeddings aims to effectively encapsulate the intricate topology of the entire graph within the positional embedding space.

In addition to positional encodings, an avenue for integrating structural biases into the transformer architecture lies in node sampling. The reason behind node sampling is pivotal in determining the input sequence for the transformer. This critical step dictates the ensemble of nodes from which a given reference node will accumulate information. In the context of Message Passing Neural Networks (MPNNs), this node ensemble is often constrained to the immediate or one-hop neighbors. Early endeavors in graph transformers Ying et al. [2021], Dwivedi and Bresson [2020], Dwivedi et al. [2021], Kreuzer et al. [2021] initially adopted a simplistic approach, embedding all N graph nodes into a single extend input sequence. This methodology, however, manifests a series of drawbacks, most notably its quadratic time complexity in relation to the total node count N. Given these limitations, contemporary research endeavors have culminated in more refined sampling methodologies, such as Random Walks, Ego-Graphs, and substructure sampling. But, even these methods have some drawbacks especially linked to the efficiency, thoroughness, and uniformity of sampling, because non-uniform and non-superficial sampling of random walks or ego-graphs leads to a substantial loss of information.

A recent and notably efficient sampling technique is exemplified in Chen et al. [2022], wherein a distinct input sequence is crafted for each node. In this schema, the input token i aggregates node features from ith hop neighbors. This strategy, serving as a sampling mechanism, exhibits remarkable efficiency while incurring minimal information loss - a notable departure from preceding techniques. Nonetheless, a drawback surfaces in the form of disregarding long-range interactions beyond K hops, where $K + 1$ represents the input sequence's length - an aspect determined by a hyper-parameter. We propose an innovative sampling approach that extends and improves upon the framework delineated by Chen et al. [2022]. Our sampling strategy further endeavors to encapsulate long-range interactions surpassing the Kth hop, augmenting the efficient capture of short-range interactions, all without having a negative impact on time complexity.

The primary contributions of our research encompasses:

- Introduction of an innovative category of positional encodings, designed and trained via a neighborhood contrastive loss mechanism. This strategic development is meticulously crafted to facilitate the embedding space's comprehensive representation of graph topology.
- A sampling strategy, influenced by the pioneering work of Chen et al. [2022], enhanced by the incorporation of an additional token. This augmentation significantly bolsters the transformer's capacity to capture long-range interactions, all the while without taking a toll on the computational complexity.
- A series of extensive experiments conducted across five distinct datasets, which empirically validate the superiority of our Structure Aware Transformer on Graphs (SATG) when contrasted with both Message Passing Graph Neural Networks (MP-GNNs) and prevailing graph transformer models.

The outcomes of our research underscore the compelling advancements achieved through our devised positional encodings, sampling strategy refinement, and the resulting SATG model. These contributions collectively substantiate the efficacy and applicability of our approach, showcasing notable performance enhancements over established MP-GNNs and existing graph transformer architectures.

2 Background

2.1 Problem Formulation

Given an unweighted and undirected attributed graph $G = (V, E)$, where $V = \{v_1, v_2, , v_n\}$, and $n = |V|$, its graph structure information can be represented as an adjacency matrix $A \in \mathbb{R}^{n \times n}$ and D represents the diagonal degree matrix. The normalized adjacenecy matrix is expressed as $\hat{A} = \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}$, here $\tilde{A}$ denotes adjacency matrix with self-loops and $\tilde{D}$ denotes the corresponding degree matrix. Every node $v \in V$ is associated with a feature vector $x_v \in X$, here $X \in \mathbb{R}^{n \times d}$ is the feature matrix that describes the node attributes and d is the dimension of the node feature vector. Given we are performing a node classification task, we have a labeled node set denoted by V_l and an unlabelled node set denoted by V_u. In this case, let $Y \in \mathbb{R}^{n \times c}$ be the label matrix, where c is the number of classes. Our task is that given labels Y_{V_l}, we must predict the labels Y_{V_u} for the unlabelled nodes.

2.2 Graph Neural Network

Graph Neural Networks (GNNs) have firmly established their role as the preferred architectural framework for the manipulation of data structured in graph form. The genesis of GNNs can be traced back the pioneering work of Scarselli et al. [2008], heralding their inception. Drawing inspiration from the achievements of Convolutional Neural Networks (CNNs), the Graph Convoltional Network (GCN) was introduced by Kipf and Welling [2016] in 2017. GCN's operational paradigm involves aggregating information from immediate one-hop neighbors, employing a simplified first-order approximation of spectral convolution as elucidated by Defferrard et al. [2016]. Subsequent to this landmark development, a diverse array of GNN iterations emerged within the research landscape, as evidenced by contributions such as those by Hamilton et al. [2017] and Casanova et al. [2018].

In spite of the extensive integration of GNNs for processing graph-structured data, the fundamental message passing schema itself is not devoid of limitations. In the context of GCNs, where neighborhood aggregation and feature transformation are conjoined in a single step, the potential for encountering over-smoothing is inherent. This challenge has prompted the exploration of novel methodologies to address the concern of over-smoothing, giving rise to innovative approaches, as showcased by Gasteiger et al. [2018] and Wu et al. [2019].

2.3 Transformer

The transformer was initially widely used in the domain of Natural Language Processing (NLP). The Encoder of the Transformer contains a sequence of layers where each layer consists of two components, multi-head self-attention (MSA) and position-wise feed-forward network (FFN). Let $H \in \mathbb{R}^{n \times d}$ be the input to the self-attention module where n is the number of tokens, and d is the hidden dimension. This input H is projected into three matrices Q, K, V, using $W_Q \in \mathbb{R}^{d \times d_K}$, $W_K \in \mathbb{R}^{d \times d_K}$, $W_V \in \mathbb{R}^{d \times d_V}$, the self attention is then calculated as:

$$Q = HW_Q, K = HW_K, V = HW_V \tag{1}$$

$$Attn(H) = softmax(\frac{QK^T}{\sqrt{d_K}}) \tag{2}$$

The role of this attention matrix is to capture the pair-wise semantic similarity between the tokens in the input sequence.

2.4 Graph Transformer

Recent times have witnessed a surge in endeavors to extend the application of Transformers to encompass non-Euclidean structures as Graphs. Nonetheless, adopting a simplistic approach of conveying node features as input tokens to the transformer proves inadequate, engendering a considerable loss of structural insight by treating nodes as an unordered set and disregarding their interconnecting edges. Early endeavors within this domain, as exemplified by Ying et al. [2021], Dwivedi and Bresson [2020], Dwivedi et al. [2021], center around bundling all nodes within a single input sequence. However, this approach carries its own array of limitations, notably yielding a quadratic time complexity.

More recent advancements, as seen in Chen et al. [2022], Zhao et al. [2021], Zhang et al. [2020], tackle this quandary by embracing efficient sampling strategies. Prominent methodologies encompass the utilization of Random Walks Zhang et al. [2020], Ego-Graphs Zhao et al. [2021], and Sampled Sub-Structures Zhao et al. [2023]. A distinctive sampling approach, introduced by Chen et al. [2022], capitalizes on Neighbourhood Features Aggregation. In this schema, each node is treated as a distinct sequence, wherein sequence tokens encapsulate the aggregated features of neighboring nodes. This innovative perspective not only facilitates the exploitation of mini-batch training, a boon when faced with resource limitations, but also caters to the overarching necessity of preserving structural context.

Positional Encoding represents another pivotal facet of Transformers. While sinusoidal functions prove effective for Positional Encodings in Euclidean contexts, the Graph scenario necessitates Positional Encodings capable of encapsulating structural intricacies. This demand has yield a repertoire of choices for Positional Encodings encompassing Laplacian Eigen Vectors, Random Walk Probabilities, Node Degree, and Learnable Positional Encodings, among others. Each class of Positional Encodings harbors distinct merits and limitations, paving the way for a nuanced selection based on the specific application context.

2.5 Contrastive Learning

Contrastive learning serves as a ubiquitous technique within the realm of self-supervised learning, as evident in prior works such as You et al. [2020, 2021]. Moreover, the versatility of contrastive learning extends to supervised scenarios, exemplified by the research of Khosla et al. [2020].

In the domain of natural language processing, Gunel et al. [2020] harness supervised contrastive learning to enhance the pre-training of expansive language models through auxiliary tasks. The application of self-supervised learning is not restricted to language processing alone, but also extends to graph learning, as detailed in Liu et al. [2022], where a comprehensive overview of the subject can be gleaned.

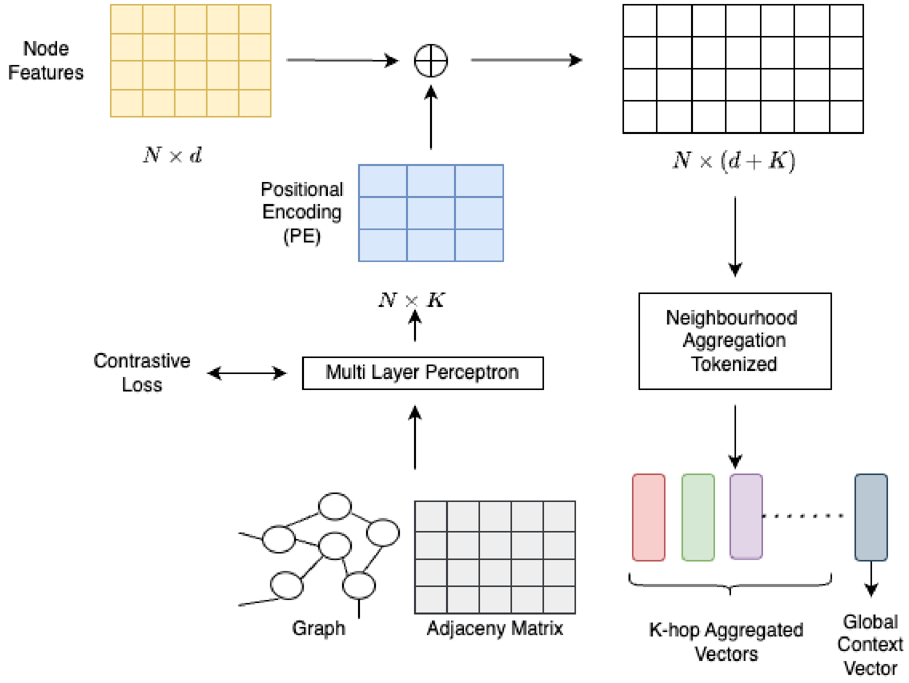

Figure 1: Proposed Architecture of SATG

3 THE PROPOSED METHOD

This section contains the details of the proposed approach and has three sub-sections, the first one explains the details about the proposed Positional Encodings, the second one deals with Node Sampling and the third one deals with capturing Long Range Interactions.

3.1 Positional Encodings

When applying the Transformer architecture to graph data structures, the inclusion of positional encoding assumes a pivotal role in furnishing structural insights. The absence of positional encoding renders attention scores solely reliant on the semantic similarity of node features - a suboptimal approach for weighting information aggregation. Thus, the core impetus behind introducing Positional Encoding lies in capturing intricate structural nuances and explicitly integrating this information within the Transformer architecture.

A prevalent choice for Positional Encoding in numerous studies involvles Laplacian Eigen Vectors. Despite their widespread popularity, these encodings are accompanied by inherent drawbacks. Kreuzer et al. [2021] expounds upon the limitations and uncertainties associated with utilizing Laplacian Eigen Vectors as Positional Encodings, alongside suggesting potential remedies for addressing these complexities. The drawbacks of Laplacian Eigen Vectors primarily encompass Eigen Value Multiplicities, Sign Invariance, and Normalization.

Certain works have adopted the node degree as a determinant for positional encoding. In comprehending the foundational mechanics of the transformer, one must gain a detailed understanding of the information aggregation process. The computation of aggregation weights hinges upon the dot product of linear transformations involving final node features - comprising raw node attributes in conjunction with positional encodings. In light of this foundational operation, the imperative arises to select positional encodings in a manner that yields a dot product between the encodings of two nodes capable of encapsulating their relative structural association. By extension, nodes exerting strong structural influence should yield a substantial dot product between their positional encodings, and conversely. This criterion serves as a fundamental guideline governing positional encoding selection.

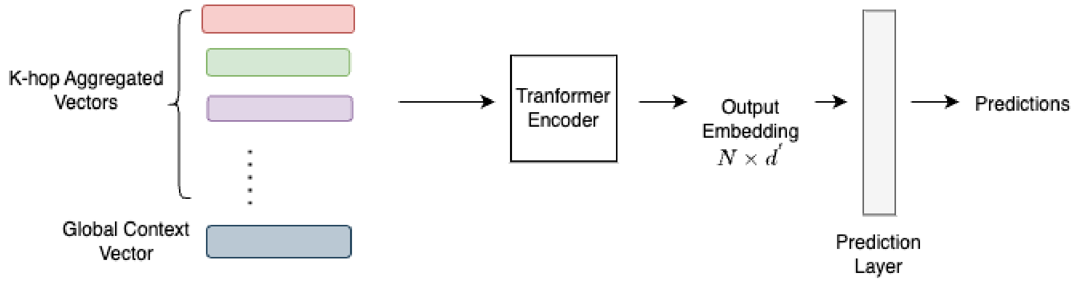

Figure 2: Structure and Global Context Capture in SATG

With this selection criterion delineated, a conspicuous solution emerges - one that aligns with the specified requisites. This solution entails training positional encodings using a contrastive loss framework, where positive samples represent proximate neighbors and negative samples denote distant non-neighbors. From the spectrum of Contrastive losses, the Neighbourhood Contrastive Loss (NCE) emerges as our chosen paradigm.

$$l_i = -log\left(\frac{\sum_{j=1}^{B} \gamma_{ij}\, exp(sim(z_i, z_j)/\tau)}{\sum_{k=1}^{B} exp(sim(z_i, z_k)/\tau)}\right) \tag{3}$$

here sim indicates the cosine similarity and τ denotes the temperature parameter. Here we have the option of choosing the hyper-parameter r as we chose the r-hop neighbours as the positive sample in the loss.

The idea behind choosing contrastive loss as the backbone for generating positional embedding was to craft a method such that we can capture the entire topology of the graph in a Euclidean embedding space. By using this contrastive loss we are able to ensure that the dot product between the positional encodings quantifies the relative structural relation between the nodes.

3.2 Input Sequence Sampling

Determining the optimal input sequence for the transformer constitutes a pivotal juncture, as this step dictates the nodes from which a specific node draws its information aggregation. Early works, as evidenced by Ying et al. [2021], Dwivedi and Bresson [2020], Dwivedi et al. [2021], Kreuzer et al. [2021], initially adopted the straightforward approach of coalescing all N nodes into a singular input sequence. However, this methodology clearly falls short of being the most effective means of handling input sequences, given its imposition of quadratic time complexity in relation to the node count N.

The realm of Message Passing Graph Neural Networks (MP-GNNs) thrives on aggregating messages solely from immediate one-hop neighbors. Thus, merely incorporating one-hop neighbors as input tokens would be incongruous, effectively equating the scenario to that of MP-GNNs. A prevalent sampling strategy involves employing Random Walks and utilizing these sequences of random walks as input data, as observed in the work of Zhang et al. [2020] and Nguyen et al. [2021]. Nonetheless, this approach carries its own caveats, particularly concerning the efficacy of random walk sampling. Sub-optimal sampling, devoid of uniformity and comprehensiveness, can lead to substantial information loss, thus impeding thorough information aggregation with potential repercussions on downstream task performance.

In a similar vein, Zhao et al. [2021] adopts ego-graphs as input sequences, effectively encountering the same issues that afflict random walk-based approaches. Here again, the uniformity and comprehensiveness of ego-graph sampling prove pivotal. Our approach aligns closely with the strategy elucidated by Chen et al. [2022]. Rather than treating each node as token within an expansive input sequence, we craft a unique input sequence for every individual node. This strategic shift introduces a distinctive facet - each node is associated with an individual input sequence. This novel perspective extends the capability of training the transformer model on extensive datasets in a mini-batch fashion, a feature not possible when nodes are treated as tokens within a monolithic input sequence.

To understand the formation of an input sequence for a given node v , let us define its k-hop neighbour as any node u satisfying the condition

$$d(u, v) = k \tag{4}$$

here d is the shortest-path distance between node u and node v. Now before moving to aggregating features we concatenate the raw node attributes with the positional encodings trained on the Neighbourhood Contrastive Loss. The modified feature vector for a node v is represented by $H(v)$, and defined as :

$$H(v) = F(v) \parallel P(v) \tag{5}$$

here $F(v)$ is the raw node attribute of node v and $P(v)$ is the positional encoding vector associated with node v and $\parallel$ represents the concatenation operator.

$$x_v^k = \sum H(u) \quad where\ d(u, v) = k. \tag{6}$$

The input sequence for node v would look like $S_v = (x_v^0, x_v^1, ..., x_v^K)$. Here K is a hyper-parameter.

3.3 Capturing Long-Range Interactions

By embracing the aforementioned sampling strategy, we stand poised to proficiently capture information encompassing the K-hop neighborhood. However, interactions involving nodes situated beyond the K-hop realm invariably elude this approach. While a seemingly straightforward solution involves setting a substantially large value for K, this recourse proves suboptimal - both from computational efficiency and performance perspectives. We present an innovative remedy to this quandary, devised to circumvent any escalation in the sampling's time complexity.

We propose a solution to this issue in such a way that there is no increase in the time complexity of sampling. Within the input sequence tailored for each node, we introduce an additional token appended at the sequence's terminus. This supplemental token embodies the aggregated features of nodes positioned beyond the K-hop vicinity relative to the reference node. Calculating this token's value can be executed with remarkable efficiency. By precomputing and caching the summation of modified features for all N nodes, this summation remains accessible for all nodes. Consequently, for each node the summation of its tokens (within its sequence) can be subtracted from the precomputed total sum, yielding the summation of modified features associated with nodes surpassing the K-hop boundary. The introduction of this supplementary token within the input sequence extends the model's

Table 1: Evaluation Results

Approach	Cora	citetseer	Pubmed	Photo	Computer
GCN	80.1 ± 0.5	67.9 ± 0.5	78.9 ± 0.7	92.70 ± 0.20	89.65 ± 0.52
GAT	83.0 ± 0.7	72.5 ± 0.7	79.0 ± 0.3	93.87 ± 0.11	90.78 ± 0.13
SAN	74.02 ± 1.01	70.64 ± 0.94	86.22 ± 0.43	94.86 ± 0.1	89.83 ± 0.16
Graphormer	72.85 ± 0.76	66.21 ± 0.83	OOM	92.74 ± 0.14	OOM
NAGphormer	91.11 ± 0.32	79.55 ± 0.73	89.70 ± 0.19	95.49 ± 0.11	91.22 ± 0.14
SATG(Ours)	91.90 ± 0.27	80.80 ± 0.41	89.85 ± 0.14	96.41 ± 0.11	92.36 ± 0.17

capacity to encapsulate prospective long-range global interactions. Remarkably, this augmentation is achieved without any compromise on the front of time complexity. In scenarios where the appended token fails to provide salient information, its associated attention weight dwindles to zero. However, when this token conveys meaningful insights, a marked enhancement in performance is discernible - a testament to the value it introduces.

As previously highlighted, our approach not only emphasizes the performance of the algorithm but also its scalability. By leveraging a neighborhood aggregation sampling approach, we've significantly narrowed the attention scope from N^2 nodes to a mere K nodes. This results in the computational complexity of our attention mechanism being reduced to $O(K^2 * N * d)$ from the original $O(N^2)$, where d denotes the feature dimension. Moreover, the aggregation operations involved in generating input tokens can be performed offline, enhancing the efficiency. Hence, the overall complexity of our technique is $O(K^2 * N * d)$. This linear time complexity ensures that our method, SATG, is well-equipped to manage larger graphs efficiently leading to a scalable graph learning architecture.

4 EXPERIMENTS

4.1 Experimental Settings

4.1.1 Datasets

: We performed extensive experiments on 5 public benchmark datasets including 3 citation networks, A citation network where each node represents a paper and an edge represents a citation between those papers. The citation networks used are Cora(consists of 2708 nodes, and each node belongs to one of 7 classes, 5429 edges, and the node feature size is 1433), citetseer(consists of 3312 nodes, and each node belongs to one of 6 classes, 4732 edges, and the node feature size is 3703) , and the other two being A-Photo and A-Computer. On all the mentioned datasets, we have used $60\%/20\%/20\%$ train/val/test random splits.

4.1.2 Baselines

: We conduct a comprehensive comparison of our proposed Structure Aware Transformer on Graph (SATG) against a set of five sophisticated baseline models. This lineup encompasses:

- Two full batch Graph Neural Networks (GNNs), namely Graph Convolutional Network (GCN) Kipf and Welling [2016] and Graph Attention Network (GAT) Veličković et al. [2017].
- Three distinct graph transformers, namely Self-Attention Network (SAN) Kreuzer et al. [2021], Graphormer Ying et al. [2021] and Nagphormer Chen et al. [2022].

4.2 Evaluation

For experiments, we conduct 10 runs and report the average results of these 10 runs with the corresponding standard deviation. The results for all the 5 datasets have been reported in Table 1. Upon observing the results it is clear that SATG consistently outperforms MP-GNNs by a substantial margin thereby proving the fact that fundamental message-passing GNNs are not the most optimal way of dealing with graph data-structures. SATG also outperforms all the graph-transformer baselines as well. Looking at the result table we also understand that there are some graph transformers that run

Table 2: Ablations

Component					
	Cora	citetseer	Pubmed	Photo	Computer
No PEs and No Global Nodes	90.37	80.12	89.39	95.36	90.87
Global Nodes and no PEs	90.93	80.71	89.7	95.94	92.25
PEs and no Global Nodes	91.11	80.72	89.55	96.27	91.71
PEs and Global Nodes	91.90	80.80	89.85	96.41	92.36

out-of-memory(OOM) on medium size datasets, as we can see that Graphormer, SAN and Graph-GPS have run out-of-memory on various datasets.

4.3 Ablation Study

Positional Encoding: To understand if positional encodings are actually useful we perform a series of experiments on all 5 datasets without concatenating the node features with positional embeddings, and see the improvements caused solely by the inclusion of positional vectors trained on the neighbourhood contrastive loss. The results are summarized in Table 2. Since it is clear that positional encodings were fundamentally added to provide topological or structural knowledge of the graph, we will see that the percentage improvement for each dataset is different. We observe that positional encodings are of greater help to datasets with high dependence on structural knowledge and inductive biases, when compared to datasets where the semantic knowledge of node features is overpowering the topology.

Global Contextual Information: The idea of sum pooling all the modified node features(raw node attributes + positional encodings) beyond K hops and introducing this as the last token in the input sequence is being referred to as giving **global contextual information** here. We performed experiments on all 5 datasets by removing this extra token that is the sum pool of all the modified node features(raw node attributes + positional encodings) beyond K hops and have reported the ablation studies in Table 3. It is evident that this extra token of global contextual information gains substantial importance if we are dealing with a dataset that has extremely low inductive biases and a good number of long-range interactions. It is also very clear that the percentage gain solely due to the provision of this extra token varies over datasets.

5 Conclusion

In this work, we have proposed SATG, a novel graph transformer model that effectively captures both short-range and long-range dependencies in graphs. Our main contributions are three-fold:

- We introduce a new class of positional encodings that are trained using a neighborhood contrastive loss to embed the graph topology into the encoding space. Experiments show these learned encodings are more effective than prior positional encoding schemes like Laplacian eigenvectors.
- We present an input sampling strategy that constructs an efficient input sequence per node consisting of its k-hop neighbor features. We further append a global node token to capture long-range interactions beyond k hops without increasing time complexity.
- We conduct extensive experiments on 5 benchmark datasets. Results demonstrate that SATG consistently and significantly outperforms state-of-the-art MP-GNNs like GCN, GAT, and graph transformers like Graphormer and SAN. This highlights the benefits of our proposed components.

In summary, SATG provides an effective way to equip graph transformers with comprehensive global and local structural knowledge. The proposed techniques for learning topological encodings and capturing multi-scale interactions enable SATG to achieve new state-of-the-art performance on node classification. Directions for future work include extending SATG to additional graph-based tasks such as link prediction and graph clustering.

References

Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. *arXiv preprint arXiv:2006.05205*, 2020.

Petar Veličković Guillem Cucurull Arantxa Casanova, Adriana Romero Pietro Lio, and Yoshua Bengio. Graph attention networks. *ICLR. Petar Velickovic Guillem Cucurull Arantxa Casanova Adriana Romero Pietro Liò and Yoshua Bengio*, 2018.

Jinsong Chen, Kaiyuan Gao, Gaichao Li, and Kun He. Nagphormer: A tokenized graph transformer for node classification in large graphs. In *The Eleventh International Conference on Learning Representations*, 2022.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

Francesco Di Giovanni, Lorenzo Giusti, Federico Barbero, Giulia Luise, Pietro Lio, and Michael M Bronstein. On over-squashing in message passing neural networks: The impact of width, depth, and topology. In *International Conference on Machine Learning*, pages 7865–7885. PMLR, 2023.

Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699*, 2020.

Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph neural networks with learnable structural and positional representations. *arXiv preprint arXiv:2110.07875*, 2021.

Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997*, 2018.

Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734. IEEE, 2005.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Ves Stoyanov. Supervised contrastive learning for pre-trained language model fine-tuning. *arXiv preprint arXiv:2011.01403*, 2020.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Wenbing Huang, Yu Rong, Tingyang Xu, Fuchun Sun, and Junzhou Huang. Tackling over-smoothing for general graph convolutional networks. *arXiv preprint arXiv:2008.09864*, 2020.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. *Advances in Neural Information Processing Systems*, 34:21618–21629, 2021.

Yixin Liu, Ming Jin, Shirui Pan, Chuan Zhou, Yu Zheng, Feng Xia, and S Yu Philip. Graph self-supervised learning: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(6): 5879–5900, 2022.

Liheng Ma, Chen Lin, Derek Lim, Adriana Romero-Soriano, Puneet K Dokania, Mark Coates, Philip Torr, and Ser-Nam Lim. Graph inductive biases in transformers without message passing. *arXiv preprint arXiv:2305.17589*, 2023.

Dai Quoc Nguyen, Tu Dinh Nguyen, and Dinh Phung. A self-attention network based node embedding model. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part III*, pages 364–377. Springer, 2021.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M Bronstein. Understanding over-squashing and bottlenecks on graphs via curvature. *arXiv preprint arXiv:2111.14522*, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *International conference on machine learning*, pages 6861–6871. PMLR, 2019.

Chaoqi Yang, Ruijie Wang, Shuochao Yao, Shengzhong Liu, and Tarek Abdelzaher. Revisiting over-smoothing in deep gcns. *arXiv preprint arXiv:2003.13663*, 2020.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34:28877–28888, 2021.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 33: 5812–5823, 2020.

Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. Graph contrastive learning automated. In *International Conference on Machine Learning*, pages 12121–12132. PMLR, 2021.

Jiawei Zhang, Haopeng Zhang, Congying Xia, and Li Sun. Graph-bert: Only attention is needed for learning graph representations. *arXiv preprint arXiv:2001.05140*, 2020.

Haiteng Zhao, Shuming Ma, Dongdong Zhang, Zhi-Hong Deng, and Furu Wei. Are more layers beneficial to graph transformers? *arXiv preprint arXiv:2303.00579*, 2023.

Jianan Zhao, Chaozhuo Li, Qianlong Wen, Yiqi Wang, Yuming Liu, Hao Sun, Xing Xie, and Yanfang Ye. Gophormer: Ego-graph transformer for node classification. *arXiv preprint arXiv:2110.13094*, 2021.