Seminar Report: *Analyzing the Efficacy of Pure Transformers in Graph Learning**

Marawan Emara[1]

[1]RWTH Aachen University

*The following report is based on [Kim et al., 2022].

Contents

0.1 Introduction

Graph learning, a rapidly evolving field within machine learning, has become instrumental in deciphering structured data across various domains. From social network analysis to molecular biology, the ability to process and interpret graph-structured data is critical for unlocking complex relational patterns and insights. This paper delves into the forefront of graph learning, exploring innovative methodologies and their implications in this dynamic field.

The evolution of Transformer models, initially conceptualized for natural language processing (NLP) as introduced by Vaswani et al. [2017], has been remarkable. Initially designed to handle sequential data, these models have transcended their original scope, demonstrating versatility and effectiveness in diverse areas, including graph learning. This expansion aligns with the broader trajectory of machine learning, where methodologies initially developed for specific applications are adapted and refined for broader use, as discussed by Bronstein et al. [2017]. In graph learning, the adaptation of Transformer models represents a significant shift, offering new perspectives and capabilities in analyzing graph-structured data.

Central to our discussion is the groundbreaking work of Jinwoo Kim et al., titled "Pure Transformers are Powerful Graph Learners." This study marks a pivotal advancement in graph learning methodologies, showcasing the adaptability of Transformer models in this domain. Kim et al.'s research brings a novel approach to the table, applying pure Transformer architectures to graph-structured data without necessitating graph-specific modifications. This methodological innovation is explored in depth in sections 0.3 and 0.4, where we examine the TokenGT model's architecture, its training procedures, performance metrics, and empirical results.

The implications of this research are far-reaching, not only enriching our understanding of graph learning but also setting the stage for future innovations in the field. As we navigate through this paper, we will explore the TokenGT model's contributions, limitations, and potential future directions in section 0.5, culminating in a comprehensive conclusion in section 0.6. This paper aims to provide a thorough understanding of the current landscape of graph learning and the transformative role of Transformer models within it, heralding a new era of research and application possibilities.

0.2 Foundations

0.2.1 Transformer Architecture

The inception of the Transformer architecture, as introduced in "Attention is All You Need" by Vaswani et al. [2017], marked a paradigm shift in sequence transduction models. Prior to this, models predominantly relied on recurrent or convolutional neural networks. The Transformer architecture stands out due to its self-attention mechanism, which enables parallel processing of sequences and adeptly captures long-range dependencies. This attribute not only enhances efficiency but also significantly improves effectiveness across various sequence modeling tasks.

Unlike traditional models, the Transformer eschews recurrence entirely and instead employs stacked self-attention and point-wise, fully connected layers for both the encoder and decoder. This design enables the model to weigh the influence of different parts of the input sequence differently, a feature that is particularly beneficial in understanding the context and nuances in language.

The self-attention mechanism computes a weighted sum of all values, where the weight assigned to each value is computed using a function of the query (the current token) and the

corresponding key (all tokens). This allows each token in the sequence to attend to all others, capturing intricate patterns in data.

Following the introduction of the Transformer model, Devlin et al. by Devlin et al. [2018] extended its architecture to develop BERT (Bidirectional Encoder Representations from Transformers), which further revolutionized the field of NLP. BERT's bidirectional training of Transformer encoders enables a deeper understanding of language context and flow compared to single-direction language models. This architecture has set new standards for a variety of NLP tasks, including translation, summarization, and question-answering.

The impact of the Transformer architecture extends beyond its initial domain of natural language processing (NLP). Its versatility and effectiveness have led to adaptations in diverse fields, including those involving graph-structured data. For instance, in computer vision, Transformers have been utilized for image recognition tasks, as demonstrated by Dosovitskiy et al. [2021]. Similarly, in bioinformatics, Transformers have been employed for protein structure prediction, as explored by Jumper et al. [2021]. This broad application spectrum underscores the architecture's foundational role in current neural network research and its potential in facilitating novel approaches to complex data processing challenges. However, adapting vanilla Transformers to these domains is particularly challenging due to their lack of inherent inductive bias for non-sequential data, a feature essential for effectively handling graph-structured information. This adaptation often requires significant architectural modifications or the integration of additional mechanisms to compensate for this limitation, as discussed by Katharopoulos et al. [2020].

0.2.2 Graph Neural Networks

Complementing the advancements in sequence modeling are Graph Neural Networks, which have become a pivotal framework for learning from graph-structured data. This type of data, characterized by intricate relational information and a non-Euclidean framework, is prevalent in various domains, including social networks and molecular biology. The pioneering work of Scarselli et al. by Scarselli et al. [2009] established the foundational principles of GNNs, demonstrating their capacity to extend deep learning techniques to the realm of graphs.

At their core, GNNs operate by learning a representation of each node in a graph, which encapsulates not only the features of the node itself but also the features of its neighboring nodes. This process, often termed as message passing, involves aggregating feature information from a node's local neighborhood and updating the node's representation based on this aggregated information. Such an iterative process allows GNNs to capture the topological structure of the graph effectively.

The architecture of GNNs typically comprises three main components: an aggregation function, an update function, and, often, a readout function. The aggregation function is responsible for collecting and combining feature information from a node's neighbors. The update function then uses this aggregated information to update the node's representation. Finally, the readout function is used to make predictions based on the learned node representations, which can be for tasks like node classification, link prediction, or graph classification.

The uniqueness of GNNs lies in their ability to directly operate on the graph structure, enabling them to capture the relational information inherent in the graph data effectively. This contrasts with traditional neural network architectures, which typically require a fixed-size input and are not inherently suited to handle the variable-sized, unordered collections of nodes and edges found in graphs.

GNNs have catalyzed significant advancements in processing and learning from graph-structured data. Their ability to extract meaningful patterns and insights from complex, interconnected data structures complements the strengths of Transformer architectures. For example, in the field of social network analysis, GNNs have been instrumental in identifying influential nodes and predicting network dynamics, as explored by Kipf and Welling [2016]. Additionally, in drug discovery, GNNs have shown promise in predicting molecular interactions, a key aspect in the development of new pharmaceuticals, as demonstrated by Gilmer et al. [2017]. Together, these two frameworks underscore the ongoing evolution of neural networks, highlighting their potential in tackling an increasingly diverse array of data-intensive challenges across various domains.

In transitioning to the next section, we build upon the foundational concepts discussed in Transformer architectures and Graph Neural Networks. We will explore how these two seemingly distinct areas converge in the novel realm of employing pure Transformer models for graph learning. This innovative approach signifies a blend of methodologies, leveraging the strengths of Transformers in sequence processing to address the unique challenges posed by graph-structured data.

0.3 Methodology of Pure Transformers in Graph Learning

0.3.1 Tokenization

The critical first step in marrying Transformer models with graph-structured data, as proposed in Kim et al. [2022], is tokenization. This process involves translating the graph's nodes and edges into a sequence of tokens, embedding these discrete elements into a continuous vector space compatible with Transformer models. This essential step paves the way for these models, originally designed for sequential data, to navigate and learn from the graph's topology effectively.

Formally, consider a graph $G = (V, E)$, where V denotes the set of nodes and E represents the set of edges. The tokenization process translates each node $v \in V$ and each edge $e \in E$ into distinct tokens. These tokens are then mapped to embeddings, denoted as $\mathbf{emb}(v)$ for nodes and $\mathbf{emb}(e)$ for edges. The embedding function **emb** effectively projects each token into a high-dimensional vector space. This projection involves the concatenation of linearly independent elements, which ensures that the representation captures the unique characteristics of each node and edge while preserving their distinctiveness. The concatenation process combines various attributes and features associated with the nodes and edges, resulting in comprehensive embeddings that encapsulate the graph's structure and relationships. Importantly, these embeddings are not static; rather, they are trainable parameters that are refined and optimized during the model's learning process. Through this dynamic and integrative approach, the embeddings evolve to more accurately represent the graph's components, enhancing the model's ability to process and interpret graph-structured data effectively.

The key to this tokenization approach lies in how it preserves and represents the graph's structural information. Each node and edge embedding captures not just intrinsic features but also contextual information, reflecting the element's position and role within the overall graph structure. This is achieved by initializing the embeddings with relevant features and subsequently updating them through the Transformer model's self-attention mechanism during training.

Beyond the basic tokenization of nodes and edges, the methodology incorporates the concept

of node and type identifiers. Node identifiers serve to distinguish individual nodes within the graph, while type identifiers categorize the nodes and edges based on their roles or characteristics within the graph. These identifiers are crucial for preserving the graph's structural integrity and ensuring that the Transformer model accurately captures the relationships and interactions between different graph elements.

To illustrate how embeddings identify edges, consider two nodes $v_i, v_j \in V$ with their respective embeddings $\mathbf{emb}(v_i)$ and $\mathbf{emb}(v_j)$. The existence and nature of the edge between these nodes can be inferred by computing the dot product $\mathbf{emb}(v_i) \cdot \mathbf{emb}(v_j)$ as part of the Transformer's attention mechanism. This dot product quantifies the degree of interaction or similarity between the nodes, effectively identifying whether an edge should exist between them and characterizing the nature of this edge. The attention mechanism then uses these calculations to determine how much focus or 'attention' the model should give to each node during processing, based on its relationship with other nodes. This approach allows the Transformer to dynamically capture and represent the intricate web of relationships within the graph, enabling a deeper understanding of the graph's structure and the interactions between its elements.

The use of node and type identifiers is a significant aspect of the model architecture, which is discussed in the following subsection. They play a pivotal role in enhancing the Transformer's ability to recognize and process the complex, hierarchical, and interlinked nature of graph-structured data. This approach represents a sophisticated adaptation of Transformers, traditionally used for linear sequence data like text, to the more intricate domain of graph data.

Tokenization, augmented with node and type identifiers, thus forms the foundational layer for applying Transformer models in graph learning. This innovative approach enables the models to interpret and leverage the graph's structural nuances, setting the stage for advanced graph analysis and representation learning.

0.3.2 Model Architecture

Building on the tokenization framework, the TokenGT model, introduced in Kim et al. [2022], represents a groundbreaking adaptation of Transformer architectures to graph-structured data. This model architecture, emerging from the principles outlined in Subsection 0.3.1, ingeniously integrates node and type identifiers with traditional Transformer elements to effectively manage the intricacies of graph data.

A key feature is the assignment of unique orthonormal identifiers to each node, represented as $\mathbf{id}(v)$ for node v. These identifiers are intricately embedded into token representations, enhancing the Transformer's ability to distinguish individual nodes. The embedding of a node token t_v, inclusive of its node identifier, is formulated as:

$$\mathbf{emb}_{node}(t_v) = \mathbf{emb}(t_v) + \mathbf{id}(v)$$

This integration ensures the model's precise recognition of each node as a distinct entity, crucial in graphs where node identity is significant.

Node identifiers in TokenGT are derived using either Orthogonal Random Features (ORFs) or Laplacian Eigenvectors. ORFs are generated from the rows of a random orthogonal matrix. An orthogonal matrix is one where the rows (and columns) are mutually orthogonal and each has a unit length. In the context of TokenGT, the use of ORFs requires the Transformer to infer the graph structure based solely on these node identifiers. This inference process is analogous

to deciphering patterns without explicit positional or relational cues, as the ORFs provide random, yet structurally significant, representations of nodes.

On the other hand, Laplacian eigenvectors are derived from the eigen decomposition of the graph's Laplacian matrix. The Laplacian matrix of a graph is a representation that captures the graph's structure by considering the degree of each node and the adjacency relationships between nodes. Specifically, it is calculated as $L = D - A$, where D is the degree matrix (a diagonal matrix holding the degree of each node) and A is the adjacency matrix of the graph. The eigenvectors of this matrix, particularly the lower-order ones, effectively capture the global positioning of nodes within the graph. They act as graph positional embeddings in the TokenGT model, analogous to the sinusoidal positional embeddings used in NLP Transformers. These embeddings provide a sense of 'position' or 'location' within the graph, helping the Transformer model to better understand and utilize the structural context of the graph.

Furthermore, TokenGT employs type identifiers to differentiate between node and edge tokens. Each token's embedding is augmented with a type identifier $\mathbf{type}(t)$, enabling the model to accurately interpret node-edge relationships within the graph. The rationale behind this integration is to provide the Transformer architecture with a clear distinction between different types of elements within the graph. This distinction is crucial as it allows the model to effectively process and respond to the unique characteristics and roles of nodes and edges, leading to more accurate and nuanced understanding of the graph's structure. The enhanced embedding for a token with its type identifier is:

$$\mathbf{emb}_{type}(t) = \mathbf{emb}(t) + \mathbf{type}(t)$$

This approach of integrating type identifiers into the embedding process ensures that the Transformer can distinguish between various elements in the graph, facilitating more precise modeling of the complex interrelationships present in graph-structured data.

At its core, TokenGT adapts the self-attention mechanism, tailored for these augmented embeddings:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where Q, K, V are the query, key, and value matrices, and d_k is the key dimensionality, as introduced in Vaswani et al. [2017]. This mechanism efficiently discerns relational dynamics within the graph, weighting the influence of nodes and edges based on their relevance.

The TokenGT model, by melding these identifiers with the established Transformer framework, showcases the adaptability of Transformers in tackling not just sequential data but also the complexities inherent in graph learning tasks.

0.3.3 Theoretical Analysis

The theoretical basis of the TokenGT model is explored in Kim et al. [2022], linking the principles of Invariant Graph Networks (IGNs) with the capabilities of Transformers in handling graph-structured data. This theoretical grounding ensures that the model is not only practically efficient but also aligns with advanced concepts in graph theory and neural network design.

The analysis commences with a discussion on permutation symmetry in the context of graph representation. Permutation symmetry is pivotal in graph representation, ensuring invariance of graph properties irrespective of node reordering. Mathematically, for a graph $G = (V, E)$

with nodes V and edges E, a permutation π over V should not alter the inherent properties of G. This is crucial in defining isomorphism in graphs.

IGNs are derived from the fundamental concept of permutation symmetry. They utilize tensor representations of graphs to encode relationships and attributes. Formally, an IGN can be represented as:

$$\text{IGN}(G) = f(T(G)), \tag{0.1}$$

where $T(G)$ is the tensor representation of the graph G, and f is an invariant function under node permutation. It has been shown that 2-IGNs surpass the expressiveness of traditional message-passing GNNs by Gilmer et al. [2017].

A central aspect of the analysis examines the ability of the Transformer's self-attention mechanism, which is mentioned in Section 0.3.2, to approximate an equivariant basis, given the inclusion of appropriate auxiliary information. This investigation reveals that, under certain conditions, self-attention can effectively approximate the equivariant basis, a key component in graph processing.

The theoretical results are extended to higher-order self-attention layers, demonstrating that with generalized node and type identifiers, the Transformer can approximate higher-order equivariant linear layers. This advancement is crucial for enhancing the expressiveness of Transformers in the realm of graph analysis. In particular, it aligns their capabilities with the k-Weisfeiler-Lehman (k-WL) graph isomorphism tests, which are fundamental in graph theory for determining the structural similarity of graphs.

The k-WL test is a powerful algorithm used for graph isomorphism testing. Graph isomorphism refers to the problem of determining whether two graphs are structurally the same, meaning there's a one-to-one correspondence between their node sets that preserves edge connectivity. The k-WL test iteratively updates labels assigned to tuples of k nodes in a graph, based on the labels of neighboring node tuples. The process continues until the labels stabilize, resulting in a signature that uniquely represents the graph's structure. The expressiveness of the k-WL test increases with k; the 1-WL test (often referred to simply as the WL test) considers pairs of nodes, while higher k values consider larger tuples, allowing for more complex and subtle graph structure detection (Morris et al. [2019]).

By approximating higher-order equivariant linear layers, Transformers are able to mimic the function of these higher-order WL tests (like 2-WL or higher). This means that Transformers can effectively discern more complex graph structures, akin to the nuanced structural detection achieved by higher-order WL tests. This alignment with the 2-WL graph isomorphism test and established Isomorphic Graph Networks (IGNs), as seen in Kim et al. [2022], signifies a significant step in enhancing the ability of Transformers to process and analyze graph-structured data, capturing intricate and subtle structural details that are crucial for various applications in graph analysis.

The theoretical results are extended to higher-order self-attention layers, demonstrating that with generalized node and type identifiers, the Transformer can approximate higher-order equivariant linear layers. This approximation is pivotal for enhancing the expressiveness of Transformers in graph-related tasks. Specifically, by approximating these higher-order linear layers, Transformers become capable of mimicking the workings of the 2-WL graph isomorphism test. The 2-WL test is a powerful algorithm for assessing graph isomorphism, essentially determining if two graphs are structurally identical. By aligning their capabilities with this test, Transformers gain the ability to more accurately discern subtle structural differences and similarities between graphs, thereby improving their performance in tasks like graph

classification and node identification.

Furthermore, approximating an equivariant basis is beneficial for practical graph learning as it enables Transformers to maintain consistency in their treatment of graph structures. Equivariance, in this context, refers to the property of producing consistent outputs when inputs are transformed in a certain way (e.g., relabeling nodes). This consistency ensures that the Transformer's interpretation of the graph is not arbitrarily affected by how the graph is presented or labeled, but rather is based on the intrinsic properties of the graph itself. As a result, the model's learning and predictions become more reliable and generalizable across different graph structures.

By approximating higher-order equivariant linear layers, Transformers align their expressiveness with the 2-WL graph isomorphism test and established Isomorphic Graph Networks (IGNs), as seen in the study by Kim et al. [2022]. This alignment significantly enhances their practical utility in graph learning, enabling them to effectively capture and interpret complex graph structures and relationships.

Empirically, the TokenGT model has showcased its prowess in graph learning, outperforming traditional GNNs and rivalling advanced Transformer variants. Its theoretical and practical efficacy underscores its potential as a transformative tool in graph-structured data analysis.

0.3.4 Training TokenGT

Training the TokenGT model, as described in Kim et al. [2022], involves unique considerations due to the complex nature of graph-structured data. This process deviates from standard neural network training, accommodating the irregularities and variability inherent in graphs. Such considerations are crucial for the model to accurately learn and represent graph patterns and relationships.

In the TokenGT model, node and type identifiers play a crucial role. The model utilizes ORFs and Laplacian eigenvectors as node identifiers. ORFs are generated using rows of a random orthogonal matrix obtained through the QR decomposition of a random Gaussian matrix. This approach does not inherently encode the graph structure but requires the model to infer it from the provided identifiers.

Laplacian eigenvectors, on the other hand, are obtained from the eigendecomposition of the graph's Laplacian matrix. These eigenvectors provide a form of positional embedding, reflecting the distances between nodes in the graph. This method is more informative than ORFs as it incorporates aspects of the graph's structure directly into the embeddings.

For type identifiers, the model differentiates between node and edge tokens. These identifiers are crucial for the Transformer's attention mechanisms, enabling the model to focus on relevant token types during processing.

The TokenGT model employs the AdamW optimizer, a variant of the Adam optimizer with improved weight decay regularization. The model's training involves different batch sizes based on the complexity and size of the input data. For instance, sparse inputs might use larger batch sizes compared to dense inputs due to memory constraints. The learning rate is typically set with a warm-up phase followed by a decay phase, aligning with common practices in neural network training.

Furthermore, the TokenGT model accommodates both sparse and dense input representations, a consideration crucial for graph data with varying degrees of connectivity. Sparse representations might involve direct embeddings of the graph's nodes and edges, while dense representations could include all possible pairwise edges.

Regularization is also key in preventing overfitting and enhancing the model's generalization capabilities, especially in large-scale graph learning. Similar to its use in other neural network architectures, dropout in the TokenGT model involves randomly omitting a subset of features or nodes during training. This technique is essential for improving the robustness of the model.

Stochastic depth involves randomly dropping layers during training, which can be particularly effective in deep architectures. Eigenvector dropout, specific to the TokenGT model, perturbs the Laplacian eigenvectors, adding an additional layer of randomness and robustness to the training process. As part of the AdamW optimizer, weight decay helps in regularizing the model, especially in the context of the complex and high-dimensional data involved in graph learning.

In graph learning, particularly for models like TokenGT, the choice of loss functions and optimization algorithms is critical. The choice of the loss function and the gradient descent algorithm plays a vital role here. The TokenGT model often employs L1 (Mean Absolute Error) or L2 (Mean Squared Error) loss functions. These functions are particularly effective in regression tasks common in graph learning, such as predicting properties of nodes or entire graphs. Furthermore, standard gradient descent algorithms, particularly their adaptive variants like Adam or AdamW, are used for training. These algorithms are favored due to their efficiency in handling large datasets and high-dimensional parameter spaces typical in graph neural networks.

The training of the TokenGT model in graph learning encompasses a range of strategies, from specific implementations of node and type identifiers to adaptations in loss functions and optimization techniques, all tailored to address the unique challenges posed by graph-structured data.

0.3.5 Assessing TokenGT with the Right Metrics

In evaluating the performance of graph learning models like TokenGT, selecting appropriate evaluation metrics is crucial. These metrics, as detailed in Kim et al. [2022], are designed to measure the model's effectiveness in capturing the intricate relationships and properties unique to graph-structured data.

In the realm of graph learning, common metrics include accuracy, precision, recall, and F1-score for classification tasks. For regression tasks, mean absolute error (MAE) and mean squared error (MSE) are frequently used. The choice of metric often depends on the specific nature of the task – for instance, node classification, graph classification, or link prediction.

In the context of the TokenGT model, particularly in tasks like quantum chemistry regression (as in the PCQM4Mv2 dataset), MAE is a primary metric due to its focus on prediction accuracy in continuous output spaces. This metric provides a direct measure of the average magnitude of errors between the predicted values and the actual values, making it suitable for evaluating the model's performance in predicting quantitative properties.

Additionally, in graph learning tasks where the structure and connectivity of the graph are of interest, metrics such as ROC-AUC (Receiver Operating Characteristic - Area Under Curve) and Precision-Recall AUC can be employed, especially in scenarios involving imbalanced datasets or when the prediction of relationships (edges) between nodes is critical.

The careful selection and application of these evaluation metrics are vital for a thorough assessment of a graph learning model's performance, ensuring it not only predicts accurately but also encapsulates the depth and complexity of graph-structured data.

Building upon the detailed methodology of applying Transformer models to graph-structured

data, the next section shifts focus to the empirical domain. We will explore the experimental results obtained from implementing the TokenGT model, showcasing how this theoretical framework translates into tangible outcomes in various graph learning scenarios

0.4 Experimental Results

0.4.1 Datasets

To thoroughly evaluate the TokenGT model, a wide array of datasets was employed, ranging from synthetic ones like Barabási-Albert (BA) random graphs to the PCQM4Mv2 large-scale quantum chemistry regression dataset. Additionally, various transductive node classification datasets were included, encompassing co-authorship, co-purchase, and Wikipedia page networks.

Barabási-Albert random graphs are renowned for their scale-free properties, characterized by a power-law distribution in node connectivity. This scale-free nature is achieved through a preferential attachment process, making BA graphs a pertinent choice for assessing the model's ability to handle real-world-like network structures.

The PCQM4Mv2 dataset, from the Open Graph Benchmark Large-Scale Challenge by Hu et al. [2020], is pivotal for quantum chemistry regression tasks. Comprising over 3.7 million molecular graphs, each graph in this dataset represents a unique molecule, annotated with its quantum mechanical properties. This dataset is instrumental in evaluating the model's accuracy in predicting molecular characteristics based on graph structure and node features.

Table 1: Statistics of the transductive node classification datasets by Kim et al. [2022].

Dataset	CS	Physics	Photo	Computers	Chameleon	Crocodile
# nodes	18,333	34,493	7,650	13,752	2,277	11,631
# edges	81,894	247,962	119,081	245,861	36,101	180,020
# classes	15	5	8	10	6	6

Additionally, the TokenGT model is assessed using transductive node classification datasets, as seen in Table 1, including co-authorship (CS, Physics), co-purchase (Photo, Computers), and Wikipedia page networks (Chameleon, Crocodile). These datasets encompass large graphs, some with tens of thousands of nodes, providing a comprehensive evaluation platform for the model's performance in various real-world scenarios.

This blend of synthetic and real-world datasets provides a multi-faceted view of the TokenGT model's capabilities, ensuring a nuanced and thorough evaluation across different graph learning scenarios and challenges.

0.4.2 Performance Comparison

The TokenGT model's performance is benchmarked against a range of state-of-the-art models, offering a comprehensive comparative analysis as detailed in Kim et al. [2022]. Particularly in transductive node classification tasks, the model was tested on diverse datasets, from social networks to informational networks, as seen in Table 2.

The TokenGT model and its variants, such as TokenGT (Lap) + Performer and TokenGT (Lap) + Performer + SEB, demonstrate robust performance across these varied datasets. Notably, in datasets related to co-authorship networks like CS and Physics, TokenGT exhibits

performance that is either on par with or surpasses that of well-established graph neural networks like GCN, GAT, and GIN. This is evident in the Physics dataset, where TokenGT shows a high level of accuracy, reflecting its capability in handling intricate graph structures.

Table 2: Transductive node classification. OOM denotes out-of-memory error by Kim et al. [2022].

	CS	Physics	Photo	Computers	Chameleon	Crocodile
GCN	0.895 ± 0.004	0.932 ± 0.004	0.926 ± 0.008	0.873 ± 0.004	0.593 ± 0.01	0.660 ± 0.01
GAT	0.893 ± 0.005	0.937 ± 0.01	0.947 ± 0.006	0.914 ± 0.002	0.632 ± 0.011	0.692 ± 0.017
GIN	0.895 ± 0.005	0.886 ± 0.046	0.886 ± 0.017	0.362 ± 0.051	0.479 ± 0.027	0.515 ± 0.041
Graphormer	0.791 ± 0.015	OOM	0.894 ± 0.004	0.814 ± 0.013	0.457 ± 0.011	0.489 ± 0.014
TokenGT (Near-ORF) + Performer	0.882 ± 0.007	0.931 ± 0.009	0.872 ± 0.011	0.82 ± 0.019	0.568 ± 0.019	0.583 ± 0.024
TokenGT (Lap) + Performer	0.902 ± 0.004	0.941 ± 0.007	0.919 ± 0.009	0.86 ± 0.012	0.637 ± 0.032	0.638 ± 0.025
TokenGT (Lap) + Performer + SEB	0.903 ± 0.004	0.950 ± 0.003	0.949 ± 0.007	0.912 ± 0.006	0.653 ± 0.029	0.718 ± 0.012

A key aspect of the TokenGT model is its efficiency in managing large graphs. Unlike some baseline models such as Graphormer, which face memory limitations in large-graph scenarios due to $O(n^2)$ memory requirements, TokenGT efficiently handles these challenges. It employs orthonormal vectors for node identification and Performer kernel attention for computational efficiency, enhancing its scalability and practicality in large-scale graph applications.

The comparative analysis also highlights limitations in competing models, particularly Graphormer, which encounters out-of-memory issues on datasets with extensive graphs. This issue primarily arises due to Graphormer's design, which has a quadratic complexity in capturing the relationships between edges in large graphs. This complexity leads to a significant increase in memory requirements as the size of the graph grows, making it challenging to process extensive graphs without encountering memory limitations.

Additionally, the study includes a comparison with the Graph Attention Network (GAT) model introduced by Velickovic et al. [2017]. GAT represents a significant advancement in the field, introducing an attention-based mechanism that allows for more nuanced weighting of node features. TokenGT's performance, when juxtaposed with GAT, further illustrates its capability to effectively capture and process complex relational patterns within graphs, a testament to its innovative architecture and the effectiveness of its attention mechanisms.

Through this extensive performance comparison, the TokenGT model demonstrates its robust capabilities in various graph learning scenarios, affirming its place as a competitive and scalable option within the graph neural network landscape.

0.4.3 Ablation Studies

Ablation studies on the TokenGT model provide critical insights into the impact of different model components, especially the use of various node identifiers, on its overall graph learning performance.

The study revealed that the choice of node identifiers significantly influences the model's ability to understand and process graph-structured data. Using Orthogonal Random Features (ORFs) as node identifiers in TokenGT (ORF) led to a notable improvement in performance, achieving a mean absolute error (MAE) of 0.0962 on the PCQM4Mv2 dataset. This was an advancement over all Graph Neural Network (GNN) baselines, indicating the model's capability of implicitly learning graph structures, despite the absence of explicit graph encoding (or inductive bias) in ORFs and the standard Transformer architecture.

Further enhancement was observed with the introduction of Laplacian eigenvectors in TokenGT (Lap). These eigenvectors provide positional information on graphs, which, when used

Table 3: Results on PCQM4Mv2 large-scale graph regression benchmark by Kim et al. [2022].

Method	# parameters	valid MAE ↓	test-dev MAE ↓
Message-passing GNNs			
GCN	2.0M	0.1379	0.1398
GIN	3.8M	0.1195	0.1218
GAT	6.7M	0.1302	N/A
GCN-VN	4.9M	0.1153	0.1152
GIN-VN	6.7M	0.1083	0.1084
GAT-VN	6.7M	0.1192	N/A
GAT-VN (large)	55.2M	0.1361	N/A
Transformers with strong graph-specific modifications			
Graphormer	48.3M	0.0864	N/A
EGT	89.3M	0.0869	0.0872
GRPE	46.2M	0.0890	0.0898
Pure Transformers			
Transformer	48.5M	0.2340	N/A
TokenGT (ORF)	48.6M	0.0962	N/A
TokenGT (Lap)	48.5M	0.0910	0.0919
TokenGT (Lap) + Performer	48.5M	0.0935	N/A

as node identifiers, improved the model's performance, achieving an MAE of 0.0910, as seen in Table 3. This performance was comparable to Transformers with complex graph-specific modifications, highlighting the effectiveness of Laplacian eigenvectors in learning diverse and useful attention patterns.

Additionally, integrating the Performer mechanism in TokenGT (Lap) + Performer and TokenGT (Near-ORF) + Performer variants brought computational efficiency to the model, particularly beneficial for handling large-scale graph data. The Performer, an efficient approximation of self-attention, complements the structural information provided by Laplacian eigenvectors and near-orthonormal vectors, enhancing the model's scalability and practicality.

Moreover, the inclusion of a Sparse Equivariant Basis (SEB) in TokenGT (Lap) + Performer + SEB further refined the model's ability to represent complex graph structures, indicating a promising direction for future enhancements in graph positional encoding techniques.

These ablation studies, as reported in Kim et al. [2022], highlight the critical role of node identifiers in enhancing the TokenGT model's efficacy in processing and interpreting graph-structured data. The findings from these studies not only shed light on the importance of each component within the TokenGT framework but also pave the way for future research focused on optimizing node identifier strategies and improving Transformer-based models for graph learning applications.

Following the in-depth exploration of the TokenGT model's efficacy through ablation studies, we now transition to a broader discussion. This next section will reflect on the implications and broader significance of our findings, considering the potential impact and future directions in the field of graph learning and Transformer model applications.

0.5 Discussion

0.5.1 Role of Laplacian Eigenvectors in Enhancing TokenGT Performance

The introduction of Laplacian eigenvectors as node identifiers in TokenGT (TokenGT (Lap)) and the consequent improvement in performance highlight an interesting aspect of the model's functioning. While the use of Orthogonal Random Features (ORFs) allows TokenGT to implicitly learn graph structures, the explicit encoding of the graph's structural information

through Laplacian eigenvectors offers additional advantages.

Laplacian eigenvectors provide a detailed representation of a graph's topology, encapsulating crucial information about node connectivity and the graph's layout. This explicit encoding of graph structure augments the Transformer's inherent capacity to discern and learn relational patterns within the graph data. The Laplacian eigenvectors do more than provide positional information; they offer a direct, mathematical representation of the graph's structure, thereby enhancing the contextual understanding of the model. This helps the model comprehend how nodes are positioned relative to each other, leading to a more nuanced interpretation of node relationships and an improved ability to recognize and learn structural patterns within graphs (Kim et al. [2022]).

Moreover, this explicit structural encoding can potentially reduce the learning complexity for the model. By providing a clear structural framework, Laplacian eigenvectors enable the model to more efficiently learn and process graph-structured data, as evidenced by the improved performance metrics. This efficiency is especially beneficial in dealing with complex graphs where relational patterns might not be immediately apparent through inference alone.

In addition, the integration of explicit structural information through Laplacian eigenvectors might contribute to the model's robustness when handling a diverse range of graph sizes and types. It provides a consistent and reliable foundation for the model to understand and interpret different graphs, potentially enhancing its generalizability across various graph learning tasks.

The use of Laplacian eigenvectors in TokenGT underscores the importance of combining implicit learning capabilities with explicit structural encodings in Transformer-based models. This approach not only facilitates a deeper understanding of graph structures but also points to potential benefits in enhancing model performance. It offers valuable insights into the dynamics of graph learning and highlights the synergy between implicit pattern learning and explicit structural information in improving the efficacy of models dealing with graph-structured data.

0.5.2 Implications

The Tokenized Graph Transformer (TokenGT) model, as outlined in Kim et al. [2022], signifies a transformative approach in graph learning, focusing on a minimal graph-specific inductive bias to learn directly from data. This aligns with evolving paradigms in large-scale data learning as noted in Lee et al. (2019) by Lee et al. [2019]. TokenGT's architecture, integrating node and type identifiers into a Transformer encoder, mirrors the Graphormer but introduces critical differences. The use of orthogonal random features and Laplacian eigenvectors as node identifiers highlights TokenGT's proficiency in capturing the graph's structural intricacies, which are implicitly embedded in these identifiers.

Unlike standard Transformers, which do not intrinsically discern graph structure, TokenGT demonstrates a pronounced ability to attain lower mean absolute errors (MAE) compared to traditional Graph Neural Network (GNN) baselines. This is particularly evident when employing Laplacian eigenvectors, underscoring the model's proficiency in interpreting graph structural information. This minimal dependence on graph-specific inductive bias confers upon TokenGT a remarkable adaptability and robustness in learning from data. The model exhibits versatility, akin to that observed in Vision Transformers, in attending to both global and local graph structures. This feature paves the way for potential developments in hybrid architectures that may integrate convolutional techniques into graph learning.

The implications of TokenGT extend beyond its immediate applications. It highlights the broader potential of Transformer-based architectures in graph learning, especially in handling

increasingly complex datasets. Furthermore, TokenGT's approach to treating input nodes and edges as independent tokens and applying self-attention mirrors similar methodologies in language and vision Transformers. This less biased approach compared to traditional GNNs, which incorporate the sparse graph structure or permutation symmetry of graphs into each layer, marks a significant shift in graph learning paradigms.

TokenGT's minimal graph-specific inductive bias necessitates that it learn the internal computation structure largely from data. Such a characteristic is known to be effective with large-scale data, a premise explored with the PCQM4Mv2 quantum chemistry regression dataset. The model's performance, particularly when enhanced with Laplacian eigenvectors, indicates that it can learn diverse and useful computation structures from data. This learning is facilitated by the graph structure information inherent in these eigenvectors, suggesting that the choice of node identifiers is crucial in defining the model's efficiency and versatility.

The empirical performance of TokenGT, as detailed in the research, also sheds light on its ability to adapt to equivariant layers by learning fixed equivariant bases at each attention head. In practice, however, the model can utilize multihead self-attention to develop less restricted and potentially more effective computation structures. This adaptability is reflected in the varying attention patterns observed in different layers of the model, resonating with behaviors seen in Vision Transformers. The diverse attention structures learned by TokenGT, especially when using Laplacian eigenvectors, indicate a higher efficacy and call for further exploration into improved node identifiers based on graph positional encodings.

TokenGT stands as a notable advancement in graph learning, showcasing the potency of Transformer-based architectures in this domain. Its capability to adapt to various graph structures and compete with traditional GNNs, while maintaining minimal graph-specific inductive bias, marks it as a precursor to emerging trends in graph learning. The model's flexibility and potential for integration with other architectural paradigms underscore its importance and the wide-ranging implications for future research and applications in graph learning.

0.5.3 Limitations

While the TokenGT model marks a breakthrough in graph learning, it is not without limitations, as highlighted in Kim et al. [2022]. These limitations, including reliance on large-scale data and comparative performance challenges, are critical for a holistic understanding and for guiding future enhancements in the field.

A primary limitation of TokenGT is its reliance on extensive datasets for effective learning. This model, with its minimal graph-specific inductive bias, requires a substantial amount of data to learn and interpret the intricate structures within graphs. This characteristic is particularly evident in its application to datasets like PCQM4Mv2, one of the largest available, containing 3.7 million molecular graphs. The dependency on large-scale data, while enabling comprehensive learning, poses challenges in scenarios with limited data availability, questioning the model's efficiency and broader applicability.

Another critical aspect is TokenGT's performance in comparison to advanced models such as Graphormer. Despite its innovative use of node identifiers like ORFs and Laplacian eigenvectors, TokenGT exhibits slightly lower performance compared to these state-of-the-art models. The ORFs, while enhancing the model's capacity to learn structural representations, do not completely address its limitations in handling complex graph structures. Similarly, the use of Laplacian eigenvectors, which provide positional information, results in a performance boost

but does not elevate the model to the peak performance levels achieved by more specialized graph Transformer models.

The issues with ORFs and Laplacian eigenvectors indicate that there is room for improvement in the model's ability to process and interpret graph data. The performance gap observed with more specialized models underscores the necessity for further research and development in this area. Enhancing the model's efficiency with varying scales of data, improving the node identifier mechanisms, and exploring new architectural modifications are potential avenues to address these limitations. These enhancements could help in aligning TokenGT's performance with state-of-the-art models, thereby expanding its utility and contributing to the advancement of graph learning technologies.

TokenGT's current limitations illuminate essential aspects for future development in graph learning. Addressing these challenges is crucial for the evolution of the model and for the field's progression. This exploration of limitations not only sheds light on the model's present state but also opens avenues for innovative solutions and further advancements.

0.5.4 Future Work

The research on TokenGT, as detailed in Kim et al. [2022], lays the groundwork for several promising directions in graph learning. Future research aims to not only refine TokenGT's capabilities but also to broaden the scope and efficiency of graph learning models more generally.

One of the most critical areas for future research is the development of more effective node identifiers. The current use of ORFs and Laplacian eigenvectors, while beneficial, suggests that there is still significant room for improvement. Future work could explore new types of graph positional encodings or even consider hybrid architectures that blend various graph learning approaches. Such advancements could lead to more nuanced and effective ways of encoding and interpreting the complex structures within graphs.

Another promising direction is the adoption of Transformer engineering techniques from other domains like vision and language processing. This could include strategies such as data scaling, deepening the network architecture, and incorporating elements from different neural network architectures. These techniques have shown success in other domains and adapting them for graph learning could enhance the performance of graph Transformers. In addition, self-supervision techniques, which have been successful in domains like natural language processing, could be adapted for graph learning. This would potentially improve the model's ability to learn from unlabeled data, a common scenario in graph datasets.

Extending the theoretical framework of TokenGT to accommodate general discrete group actions presents another intriguing research opportunity. Such an extension would broaden the model's applicability to a wider range of graph-structured data and tasks, potentially opening new horizons in graph learning.

Furthermore, applying these advanced models to various real-world scenarios is crucial. In line with the work of Kipf and Welling by Kipf and Welling [2016], exploring applications in social network analysis, biological network modeling, and recommendation systems could be highly beneficial. These areas could significantly benefit from the enhanced modeling capabilities of advanced graph Transformers.

Lastly, the issue of interpretability, especially in decision-making contexts involving graph-structured inputs, is an area that demands attention. Developing methods to interpret the self-attention mechanisms in graph neural networks could lead to more transparent and trustworthy models. This is particularly vital in sensitive applications where understanding the model's

decision-making process is as important as the decision itself.

The insights gained from the TokenGT study set a solid foundation for future research in graph learning. These future directions are geared towards overcoming current limitations and unlocking new possibilities in handling complex graph-structured data, promising to propel the field forward in understanding and capabilities.

0.6 Conclusion

The exploration and evaluation of the Tokenized Graph Transformer (TokenGT), as detailed in Kim et al. [2022], signify a major stride in the realm of graph learning. This study has successfully demonstrated how Transformer models, traditionally applied to sequential data, can be innovatively adapted for graph-structured data. The TokenGT model stands out for its ability to intuitively grasp graph structures with minimal inductive bias, showcasing a competitive edge over leading graph learning models.

Key achievements of TokenGT include its nuanced approach to processing graph data, leveraging Orthogonal Random Features (ORFs) and Laplacian eigenvectors. This method enhances the model's capacity to discern and utilize the intricate structural aspects of graphs. Additionally, its performance across diverse datasets — from synthetic graphs to complex real-world networks — underlines its versatility and robustness.

Looking ahead, this research paves the way for numerous future explorations. There is potential for developing more advanced node identifiers, which could further refine the model's ability to interpret graph data intricately. Additionally, adapting Transformer engineering techniques from other fields could offer novel ways to enhance graph learning capabilities. Exploring self-supervision techniques and expanding the theoretical framework around TokenGT could also yield more efficient and broadly applicable models.

Moreover, the study emphasizes the importance of practical applications, particularly in real-world scenarios where interpretability is crucial. Enhancing the model's interpretability, especially in decision-making processes involving complex graph-structured data, remains a vital area of focus.

In conclusion, the TokenGT study not only deepens our understanding of graph learning mechanisms but also lays a foundation for future innovations in the field. Its implications extend beyond mere technical advancement, offering a glimpse into the potential for more sophisticated, efficient, and interpretable models that can adeptly navigate the complexities of graph-structured data. The TokenGT model, thus, represents a significant milestone in graph learning, heralding a new era of research and application in this dynamic and evolving domain.

Bibliography

[1] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric Deep Learning: Going Beyond Euclidean Data. *IEEE Signal Processing Magazine*, pages 18–42, 2017. 3

[2] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint*, 2018. URL `https://arxiv.org/abs/1810.04805`. 4

[3] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv preprint*, 2021. URL `https://arxiv.org/abs/2010.11929`. 4

[4] J. Gilmer, S. Schoenholz, P. Riley, O. Vinyals, and G. Dahl. Neural Message Passing for Quantum Chemistry. *arXiv preprint*, 2017. URL `https://arxiv.org/abs/1704.01212`. 5, 8

[5] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec. Open Graph Benchmark: Datasets for Machine Learning on Graphs. *arXiv preprint*, 2020. URL `https://arxiv.org/abs/2005.00687`. 11

[6] J. Jumper, R. Evans, A. Pritzel, and et al. Highly Accurate Protein Structure Prediction with AlphaFold. *Nature*, 2021. URL `https://doi.org/10.1038/s41586-021-03819-2`. 4

[7] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret. Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention. *arXiv preprint*, 2020. URL `https://arxiv.org/abs/2006.16236`. 4

[8] J. Kim, D. T. Nguyen, S. Min, S. Cho, M. Lee, H. Lee, and S. Hong. Pure Transformers are Powerful Graph Learners. In *Advances in Neural Information Processing Systems*, 2022. URL `https://arxiv.org/abs/2207.02505`. 1, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17

[9] T. N. Kipf and M. Welling. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv preprint*, 2016. URL `https://arxiv.org/abs/1609.02907`. 5, 16

[10] J. Lee, I. Lee, and J. Kang. Self-Attention Graph Pooling. *arXiv preprint*, 2019. URL `https://arxiv.org/abs/1904.08082`. 14

[11] C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. E. Lenssen, G. Rattan, and M. Grohe. Weisfeiler and Leman Go Neural: Higher-Order Graph Neural Networks. *AAAI*, 2019. URL `https://arxiv.org/abs/1810.02244`. 8

[12] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, pages 61–80, 2009. 4

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Kaiser, and I. Polosukhin. Attention is All You Need. In *Advances in Neural Information Processing Systems*, 2017. 3, 7

[14] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph Attention Networks. *arXiv preprint*, 2017. URL `https://arxiv.org/abs/1710.10903`. 12