

Pre-Training Identification of Graph Winning Tickets in Adaptive Spatial-Temporal Graph Neural Networks

Wenying Duan

wenyingduan@ncu.edu.cn

Jiangxi Provincial Key Laboratory of Intelligent Systems

and Human-Machine Interaction, Nanchang University

Nanchang, China

Hong Rao

raohong@ncu.edu.cn

School of Software

Nanchang University

Nanchang, China

Tianxiang Fang

6109121076@email.ncu.edu.cn

Nanchang University

Nanchang, China

Xiaoxi He*

hexiaoxi@um.edu.mo

Faculty of Science and Technology

University of Macau

Macau, China

ABSTRACT

In this paper, we present a novel method to significantly enhance the computational efficiency of Adaptive Spatial-Temporal Graph Neural Networks (ASTGNNs) by introducing the concept of the Graph Winning Ticket (GWT), derived from the Lottery Ticket Hypothesis (LTH). By adopting a pre-determined star topology as a GWT prior to training, we balance edge reduction with efficient information propagation, reducing computational demands while maintaining high model performance. Both the time and memory computational complexity of generating adaptive spatial-temporal graphs is significantly reduced from $O(N^2)$ to $O(N)$. Our approach streamlines the ASTGNN deployment by eliminating the need for exhaustive training, pruning, and retraining cycles, and demonstrates empirically across various datasets that it is possible to achieve comparable performance to full models with substantially lower computational costs. Specifically, our approach enables training ASTGNNs on the largest scale spatial-temporal dataset using a single A6000 equipped with 48 GB of memory, overcoming the out-of-memory issue encountered during original training and even achieving state-of-the-art performance. Furthermore, we delve into the effectiveness of the GWT from the perspective of spectral graph theory, providing substantial theoretical support. This advancement not only proves the existence of efficient subnetworks within ASTGNNs but also broadens the applicability of the LTH in resource-constrained settings, marking a significant step forward in the field of graph neural networks. Code is available at <https://anonymous.4open.science/r/paper-1430>.

*Corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD '24, August 25–29, 2024, Barcelona, Spain

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0490-1/24/08

<https://doi.org/10.1145/3637528.3671912>

CCS CONCEPTS

- Computing methodologies → Neural networks.

KEYWORDS

spatial-temporal graph neural network, lottery ticket hypothesis, spatial-temporal data mining

ACM Reference Format:

Wenying Duan, Tianxiang Fang, Hong Rao, and Xiaoxi He. 2024. Pre-Training Identification of Graph Winning Tickets in Adaptive Spatial-Temporal Graph Neural Networks. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain*. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3637528.3671912>

1 INTRODUCTION

Spatial-Temporal Graph Neural Networks (STGNNs) have established themselves as a formidable tool for mining the hidden patterns present in spatial-temporal data, displaying remarkable proficiency in modeling spatial dependencies via graph structures [28]. The construction of these spatial graphs is a pivotal aspect of STGNNs, in which the complex and implicit nature of spatial-temporal relationships has paved the way for the recently emerging self-learned methods that dynamically generate graphs to capture these intricate dependencies in a data-driven manner. Adaptive Spatial-Temporal Graph Neural Networks (ASTGNNs), a state-of-the-art approach to spatial-temporal data processing, are particularly adept at creating adaptive graphs through learnable node embeddings, as exemplified by models such as Graph WaveNet [29] and AGCRN [3].

Despite their advanced performance, ASTGNNs are encumbered by substantial computational overheads during both the training and inference phases, primarily due to the exhaustive calculations required for learning the adaptive adjacency matrices of complete graphs, and the computationally intensive nature of the aggregation phase. This presents a significant challenge when dealing with large-scale spatial-temporal data, where computational efficiency is paramount. Pioneering work [10] has explored this aspect, improving the efficiency of ASTGNNs during inference via sparsification of the spatial graph. However, the sparsification of the spatial graph relies heavily on the training framework and can only be conducted

after the training phase, leaving the efficiency of the training phase itself untouched.

In order to improve the efficiency of both the training and inference phases of ASTGNNs, our research introduces and explores the concept of the *Graph Winning Ticket (GWT)* for the learnable spatial graphs in ASTGNNs, an extension of the Lottery Ticket Hypothesis (LTH) in the context of ASTGNN. The original LTH posits the existence of smaller, efficient sub-networks—‘winning tickets’—that can match the performance of the full network with a fraction of the computational cost [12]. This concept has been extended to the realm of ASTGNNs, where the identification of such sub-networks within the learnable spatial graphs, i.e., GWTs, holds the potential to markedly accelerate the training and inference processes. However, a simple adoption of the LTH in the context of ASTGNN is not sufficient for practically improving their efficiency during both training and inference phases, as the traditional method of finding winning tickets involves a compute-intensive cycle of training, pruning, and retraining.

In contrast, our work aims to streamline this process by preemptively identifying a GWT for the spatial graph in ASTGNNs. We posit that a star topology, as a spanning tree of the complete graph, serves as an effective pre-determined GWT, striking a balance between edge reduction and efficient information propagation. We argue that the effectiveness of traditional ASTGNNs is enabled by the adoption of a complete spatial graph, which has a diameter of 1 and thus allows for optimally efficient information propagation. However, by relaxing the diameter of the graph from 1 to 2, our star topology significantly trims the number of edges while still preserving the integrity of spatial-temporal communication. We empirically validate the performance of this star topology across various datasets and benchmarks, solidifying its role as a winning ticket for the spatial graphs in ASTGNNs.

We summarize our main contributions as follows:

- To the best of our knowledge, we are the first to improve the efficiency of ASTGNNs during both training and inference phases, with an emphasis on the training phase. By leveraging the concept of the Lottery Ticket Hypothesis (LTH), we posit that an efficient subgraph of ASTGNN’s spatial graph can achieve comparable performance to the complete graph with significantly reduced computational overhead. We introduce a star topology as this winning ticket, which is not only sparser but also retains the essential connectivity to ensure effective information propagation across the network. This pre-determined topology obviates the need for the traditional, exhaustive search process involving training, pruning, and retraining, thereby streamlining the deployment of ASTGNNs and substantially improving their efficiency during both the training and inference phases.
- Our research also expands the theoretical foundation of the LTH by providing empirical evidence and substantial theoretical support for the existence of winning tickets in the spatial graphs of ASTGNNs. The discovery of a pre-determined winning ticket is a significant stride in the application of the LTH, as it demonstrates that such efficient sub-networks can be identified without resorting to the computationally intensive

methods traditionally employed. This advance not only reaffirms the LTH within the domain of graph neural networks, but also paves the way for its practical implementation in scenarios where computational resources are limited. By circumventing the need for iterative training and pruning, our approach enhances the feasibility of adopting the LTH in real-world settings, where efficiency and scalability are critical.

- We trained two representative ASTGNNs (AGCRN & Graph Wavenet) with our pre-identified GWTs on five of the largest known spatial-temporal datasets. The performance of the ASTGNNs with the GWTs can match or even surpass that of training with the full spatial graph, and its training and inference costs are drastically smaller. This provides empirical evidence for the existence of winning graph tickets in ASTGNNs, demonstrating that the GWTs identified are stable winning tickets of the spatial graphs within ASTGNNs, highlighting their scalability and superiority.

2 RELATED WORK

2.1 Spatial-Temporal Graph Neural Networks

The analysis of spatial-temporal data necessitates an understanding of dynamic interactions within time-varying signals across spatial domains[26, 27]. Spatial-Temporal Graph Neural Networks (STGNNs) are proficient in uncovering latent patterns in these graph-structured data [28]. A key characteristic of STGNNs is their capability to model spatial dependencies among nodes, effectively learning adjacency matrices. Depending on their approach to constructing these matrices, STGNNs can be categorized into pre-defined and self-learned methods.

Pre-defined STGNNs typically employ prior knowledge to construct graphs. For example, ASTGNN [13] and STGCN [33] utilize road network structures for graph creation. However, these pre-defined graphs encounter limitations due to their reliance on extensive domain knowledge and the inherent quality of the graph data. Given the implicit and complex nature of spatial-temporal relationships, self-learned methods for graph generation have gained prominence. These methods introduce innovative techniques to capture complex spatial-temporal dependencies, thereby offering significant advantages over traditional pre-defined models.

Self-learned STGNNs can be further divided into two primary categories: feature-based and randomly initialized methods. Feature-based approaches, such as PDFormer [15] and DG [21], construct dynamic graphs from time-variant inputs, enhancing the accuracy of the model. On the other hand, randomly initialized STGNNs, also known as Adaptive Spatial-Temporal Graph Neural Networks (ASTGNNs), facilitate adaptive graph generation through randomly initialized, learnable node embeddings. Graph WaveNet [29] introduced an Adaptive Graph Convolutional Network (AGCN) layer to learn a normalized adaptive adjacency matrix. AGCRN [3] further developed this concept with a Node Adaptive Parameter Learning enhanced AGCN (NAPL-AGCN) to discern node-specific patterns. Owing to its remarkable performance, the NAPL-AGCN model has been incorporated into various recent models [8, 9, 16].

Despite the enhanced performance of ASTGNNs, they are burdened with considerable computational overhead. This is primarily

due to two factors: *i*) the process of learning an adaptive adjacency matrix necessitates calculating the edge weight between each pair of nodes, and *ii*) the aggregation phase of these networks is inherently computationally intensive. Our research is centered on identifying the *graph winning ticket*—a concept derived from the Lottery Ticket Hypothesis—in order to accelerate training and inference in ASTGNNs. This approach is particularly relevant for handling large-scale spatial-temporal data, where efficiency is crucial.

2.2 Lottery Ticket Hypothesis.

The Lottery Ticket Hypothesis (LTH) suggests that within large neural networks, there exist smaller sub-networks (termed "winning tickets") that, when trained in isolation from the start, can reach a similar performance level as the original network in a comparable number of iterations. [12] This finding has attracted lots of research attention as it implies the potential of training a much smaller network to reach the accuracy of a dense, much larger network without going through the time and cost-consuming pipeline of fully training the dense network, pruning and then retraining it to restore the accuracy. The "Early Bird Lottery Ticket" concept builds on the original LTH. It suggests that winning tickets can be identified very early in the training process, much earlier than what was originally proposed in LTH. This finding could further optimize the training of neural networks by allowing significant pruning and resource reduction very early in the training phase.[7, 31]. Further, [6] generalised LTH to GNNs by iteratively applying UGS to identify graph lottery tickets. GEBT discovers the existence of graph early-bird tickets [32]. DGLT generalizes Dual Lottery Ticket Hypothesis (DLTH) to the graph to address information loss and aggregation failure issues caused by sampling-based GNN pruning algorithms [25]. However, the pruned GNNs are still hard to generalize to unseen graphs [23]. RGLT is proposed to find more robust and generalisable GLT to tackle this issue [24].

For extremely large models and graphs, identifying graph winning tickets typically necessitates a resource-intensive process involving training the network, followed by pruning and retraining. However, our methodology significantly streamlines the deployment of ASTGNNs. It achieves this by obviating the requirement for exhaustive cycles of training, pruning, and retraining.

3 PRELIMINARIES

3.1 Notations and Problem Definition

Frequently used notations are summarized in Table 6. Following the conventions in spatial-temporal graph neural network researches [2, 14, 30, 33], we denote the spatial-temporal data as a sequence of frames: $\{\mathbf{X}^1, \mathbf{X}^2, \dots, \mathbf{X}^t, \dots\}$, where a single frame $\mathbf{X}^t \in \mathbb{R}^{N \times D}$ is the D -dimensional data collated from N different locations at time t . For a chosen task time τ , we aim to learn a function mapping the T_{in} historical observations into the future observations in the next T_{out} timesteps:

$$\mathbf{X}^{(\tau+1):(t+T_{out})} \leftarrow \mathcal{F}(\mathbf{X}^{(\tau-T_{in}+1):\tau}) \quad (1)$$

3.2 GAT vs. AGCN

Graph Attention Network. Given an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, \mathcal{V} is the set of nodes, \mathcal{E} and $\mathbf{X} = \{x_u\}_{u=1}^N \in \mathbb{R}^{N \times D}$ is

Table 1: Summary of Notations

Symbol	Description
\mathcal{G}	Undirected graph
N	Number of nodes in the graph
\mathbf{X}^t	Feature matrix at time step t
E	Learnable node embedding matrix
d	Node embedding dimension
\mathcal{V}	Set of nodes in a graph
\mathcal{E}	Set of edges in a graph
\mathcal{K}_N	Complete graph with N nodes
\mathcal{T}	Spanning tree
\mathcal{T}^\star	Star Topology Spanning Tree
e_c	Node embedding vector of the central node
Θ	Model parameters
\mathbf{A}	Adjacency matrix
GAT	Graph Attention Network function
\mathbf{Z}^t	Output of the model at time step t

the corresponding set of edges and node features, respectively, where $N = |\mathcal{V}|$ is the number of nodes, D is the feature dimension. The adjacent matrix can be denoted as $\mathbf{A} = [\mathbf{A}_{uv}]$, where $\mathbf{A}_{uv} = 1$ if there is an edge $(u, v) \in \mathcal{E}$ and $\mathbf{A}_{uv} = 0$ otherwise. To account for the importance of neighbor nodes in learning graph structure, GAT integrates the attention mechanism into the node aggregation operation as:

$$\begin{aligned} z_u &= \sum_{v \in N_u} \mathbf{A}_{uv} x_v \Theta, \\ \mathbf{A}_{uv} &= \frac{\exp(\text{LeakyReLU}(s_{uv}))}{\sum_{k \in N_i} \exp(\text{LeakyReLU}(s_{uk}))}, s_{uv} = a(x_u, x_v). \end{aligned} \quad (2)$$

Here, $\Theta \in \mathbb{R}^{D \times D'}$ is the weight matrix, $a(\cdot, \cdot)$ is the function of computing attention scores. To simplify, we abbreviate GAT as:

$$\begin{aligned} \mathbf{Z} &= \mathbf{AX}\Theta, \\ \mathbf{A} &= \text{GAT}(\mathcal{G}, \mathbf{X}), \end{aligned} \quad (3)$$

where $\mathbf{Z} \in \mathbb{R}^{N \times D'}$, $\text{GAT}(\cdot)$ is the graph attention function.

Adaptive Graph Convolution Network. Adaptive Graph Convolutional Network (AGCN) facilitates adaptive learning of graph structures through randomly initialized learnable matrices. This approach lays the groundwork for the evolution of Adaptive Spatial-Temporal Graph Neural Networks (ASTGNNs). Among the notable ASTGNN models are Graph WaveNet and AGCRN. Within the Graph WaveNet framework, the AGCN is characterized as follows.

$$\begin{aligned} \mathbf{Z}^t &= \mathbf{AX}^t \Theta, \\ \mathbf{A} &= \text{Softmax}(\text{ReLU}(E_1 E_2^\top)), \end{aligned} \quad (4)$$

where $E_1 \in \mathbb{R}^{N \times d}$ and $E_2 \in \mathbb{R}^{N \times d}$ are the source node embeddings and target node embeddings, respectively. While in AGCRN, AGCN is defined as:

$$\begin{aligned} \mathbf{Z}^t &= \mathbf{AX}^t \Theta, \\ \mathbf{A} &= \text{Softmax}(\text{ReLU}(EE^\top)), \end{aligned} \quad (5)$$

where $E \in \mathbb{R}^{N \times d}$ is the node embeddings. Eq.(4) and Eq.(5) are extremely similar in form, with Eq.(5) being more concise. Therefore,

the general form of AGCN referred to Eq.(5) in this paper. Upon close observation of Eq.(5), it is not difficult to find that AGCN can be reformulated as the following mathematical expression likes GAT:

$$\begin{aligned} z_u^t &= \sum_{v \in \mathcal{V}} \mathbf{A}_{uv} x_v^t \Theta, \\ \mathbf{A}_{uv} &= \frac{\exp(\text{ReLU}(s_{uv}))}{\sum_{k \in \mathcal{V}} \exp(\text{ReLU}(s_{uk}))}, s_{uv} = e_u e_v^\top, \end{aligned} \quad (6)$$

which is similar to Eq.(2). Thus, AGCN can be considered a special kind of graph attention network on a complete graph with self-loops. We further abbreviate AGCN as the following equations:

$$\begin{aligned} Z_t &= \mathbf{A} \mathbf{X}_t \Theta, \\ \mathbf{A} &= \text{GAT}(\tilde{\mathcal{K}}_N, E), \end{aligned} \quad (7)$$

where $\tilde{\mathcal{K}}_N$ is the N -order complete graph \mathcal{K}_N with self-loops. As the diameter of \mathcal{K}_N is 1, AGCN facilitates the aggregation of information from all nodes to each individual node within $\tilde{\mathcal{K}}_N$. This characteristic significantly enhances the network's capability to model global spatial dependencies, culminating in its state-of-the-art performance in relevant tasks, as documented in [3, 8].

The model utilizing multi-layers of AGCN for modeling spatial dependencies is designated as ASTGNN (Adaptive Spatio-Temporal Graph Neural Network). The spatial-temporal forecasting problem when addressed using ASTGNN is mathematically expressed as:

$$\mathbf{X}^{(\tau+1):(t+T_{out})} \leftarrow \mathcal{F}(\mathbf{X}^{(\tau-T_{in}+1):\tau}; \theta, \tilde{\mathcal{K}}_N), \quad (8)$$

In this formulation, \mathcal{F} represents the forecasting function of ASTGNN parameterised by θ , which predicts future values $\mathbf{X}^{(\tau+1):(t+T_{out})}$ based on the input sequence $\mathbf{X}^{(\tau-T_{in}+1):\tau}$ and the structural information encoded in the graph $\tilde{\mathcal{K}}_N$.

However, a notable limitation arises during the training phase. The computational complexity associated with calculating adjacency matrices and executing graph convolution operations on complete graphs is of $O(N^2)$. This significant computational demand imposes a constraint on the model's scalability, particularly in scenarios involving large spatial-temporal datasets, where reducing computational complexity is crucial for practical applicability.

3.3 Graph Tickets Hypothesis

The Graph Tickets Hypothesis represents an extension of the original Lottery Tickets Hypothesis, initially introduced by UGS [6]. UGS demonstrated that GWT (*i.e.*, compact sub-graphs) are present within randomly initialized GNNs, which can be retrained to achieve performance comparable to, or even surpassing, that of GNNs trained on the original full graph. This finding underscores the potential for efficiency improvements in GNN training methodologies. However, designing a graph pruning method to identify GWTs in ASTGNNs proves to be a nontrivial task. The state-of-the-art, AGS demonstrates that spatial graphs in ASTGNNs can undergo sparsification up to 99.5% with no detrimental impact on test accuracy [10]. Nonetheless, this robustness to sparsification does not hold uniformly; when ASTGNNs, sparsified beyond 99%, are reinitialized and retrained on the same dataset, there is a notable and consistent decline in accuracy. This dichotomy underscores the nuanced complexity inherent in finding winning graph tickets in ASTGNNs and calls for further investigation.

4 METHOD

4.1 Pre-Identifying the Graph Winning Ticket

Our objective is to identify a sparse subgraph of the spatial graph pre-training and to train ASTGNNs efficiently on this subgraph without compromising performance. An ASTGNN with a spatial graph $\hat{\mathcal{G}}$, equipped with K -layer AGCN can be formulated as:

$$\begin{aligned} \mathbf{Z}^t &= \underbrace{\hat{\mathbf{A}} \cdots (\hat{\mathbf{A}}(\hat{\mathbf{A}} \mathbf{X}^t \Theta_1) \Theta_2) \cdots \Theta_K}_{K}, \\ \hat{\mathbf{A}} &= \text{GAT}(\hat{\mathcal{G}}, E), \end{aligned} \quad (9)$$

where $\hat{\mathcal{G}}$ is a sparse subgraph of \mathcal{K}_N . However, employing $\hat{\mathcal{G}}$ alone does not ensure the capability to model global spatial dependencies. To maintain the global spatial modeling ability of AGCN and to train ASTGNNs efficiently, we argue that it is essential to use a spanning tree \mathcal{T} of $\tilde{\mathcal{K}}_N$, instead of $\hat{\mathcal{G}}$, with a sufficient K :

$$\begin{aligned} \mathbf{Z}^t &= \underbrace{\tilde{\mathbf{A}} \cdots (\tilde{\mathbf{A}}(\tilde{\mathbf{A}} \mathbf{X}^t \Theta_1) \Theta_2) \cdots \Theta_K}_{K}, \\ \tilde{\mathbf{A}} &= \text{GAT}(\mathcal{T}, E), \end{aligned} \quad (10)$$

To mitigate the risk of excessive parameters and overfitting due to a high number of network layers, it is crucial to minimize r as much as possible. In light of this, we found that star topology spanning trees (with diameter $r = 2$) can function as GWTs. We make two notes on the star topology spanning tree:

- (1) **Motivation:** GAT is a message-passing network, and AGCN can be viewed as modeling a fully connected GAT, allowing any node to communicate globally. A spanning tree \mathcal{T}_N , as the minimum connected graph of complete graph, can achieve message passing to all other nodes in the graph by stacking k GAT layers, where k is the diameter of \mathcal{T}_N . Our goal is to minimize the computational complexity of ASTGNNs, so it's necessary to minimize k . Clearly, \mathcal{T}_N with a diameter of 1 doesn't exist. So we start with $k = 2$ to examine the existence of spanning trees and we found that there exist \mathcal{T}_N with a diameter of 2, uniquely forming a star topology. We'll detail this motivation in final version.
- (2) **Theoretical Analysis:** Based on spectral graph theory, if one graph is a σ -approximation of another, they have similar eigensystems and properties. We can prove that \mathcal{T}_N is an N -approximation of \mathcal{K}_N . So \mathcal{K}_N and \mathcal{T}_N have similar properties, allowing \mathcal{T}_N with fewer edges to effectively replace \mathcal{K}_N for learning good representations. The complete proof can be found in Appendix A.2.

Hypothesis 1. Given an N -order complete spatial graph \mathcal{K}_N of an ASTGNN, we investigate an associated star spanning tree: $\mathcal{T}^* = \{\mathcal{V}, \mathcal{E}^*\}$, where $\mathcal{E}^* = \{(u_c, v) \mid v \in \mathcal{V} \setminus \{u_c\}\}$, with u_c designated as the central node, v designated as the leaf node. All such \mathcal{T}^* are Graph Winning Ticket (GWT) for the spatial graph of the corresponding ASTGNN.

To ensure the existence of the associated star spanning tree, we have the following proposition:

PROPOSITION 1. *In an N -order complete graph \mathcal{K}_N , there exists a graph \mathcal{T} such that \mathcal{T} is a spanning tree of \mathcal{K}_N and the diameter*

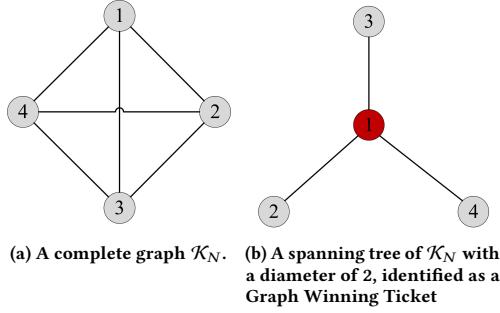


Figure 1: A complete graph and a star spanning tree with a pre-specified node number.

of \mathcal{T} is 2, and the topology of \mathcal{T} unequivocally satisfies definition of star spanning tree in Hypothesis 1.

The proof of Proposition 1 is given in Appendix A.1. To verify Hypothesis 1, we provide empirical evidence demonstrating that such \mathcal{T}^* are GWTs for their corresponding ASTGNNS in Sec. 5.

Sparsity of \mathcal{T}^* . The sparsity of the \mathcal{T}^* is quantified as $1 - \frac{2}{N}$. This represents a significant level of sparsity, particularly as the number of nodes N increases. In such cases, the sparsity becomes increasingly pronounced, highlighting the efficiency of these GWTs in large-scale spatial-temporal datasets.

4.2 Further Enhancements

In this section, we discuss two additional enhancements made to training ASTGNNS within \mathcal{T}^* . In the context of an ASTGNN $\mathcal{F}(\cdot; \theta, \tilde{\mathcal{K}}_N)$, which comprises multiple AGCN layers, a straightforward approach might involve substituting $\tilde{\mathcal{K}}_N$ with \mathcal{T}^* to facilitate rapid training. However, this seemingly intuitive method encounters two primary issues:

- Efficiency: The method lacks optimal efficiency in training.
- Central Node Selection: The random selection of the central node v_c could lead to sub-optimal performance.

Efficiency. The computational complexity of Graph Neural Network (GNN) training and inference encompasses two primary components: Deep Neural Network (DNN) computation and Graph convolution operation. Considering the relaxation of the graph's diameter from 1 to 2, an ASTGNN necessitates a minimum of two layers of AGCN to maintain comprehensive spatial-temporal communication:

$$\begin{aligned} Z^t &= A^*(A^*X^t\Theta_1)\Theta_2, \\ A^* &= \text{GAT}(\mathcal{T}^*, E), \end{aligned} \quad (11)$$

To ameliorate the computational complexity of Equation (11) in terms of DNN computation, we introduce a streamlined formulation by excluding the parameter Θ_2 . This modification facilitates 2-hop message passing within a singular AGCN layer, thereby providing the ability to model the global spatial dependencies:

$$\begin{aligned} Z^t &= A^*(A^*X^t\Theta), \\ A^* &= \text{GAT}(\mathcal{T}^*, E), \end{aligned} \quad (12)$$

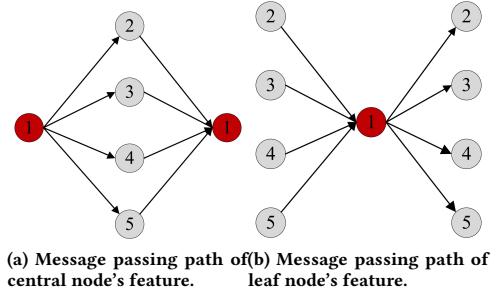


Figure 2: 2-hop message passing path of \mathcal{T}^* with pre-specified node numbers. The red node is the central node u_c and the gray nodes are leaf nodes $v \in \mathcal{V} \setminus \{u_c\}$.

From the perspective of graph convolution operations, (11) exhibits informational redundancy in its message-passing process. The message-passing trajectory delineated in 2 reveals that the paths $u_c \rightarrow v$ and $v \rightarrow u_c$ are executed twice, engendering superfluous aggregation. Such redundancy could potentially impede the model's efficiency. We therefore perform message passing as illustrated in Figure 2b using two directed graphs, denoted as $\overleftarrow{\mathcal{T}^*}$ and $\overrightarrow{\mathcal{T}^*}$. This process can be expressed by the following equations:

$$\begin{aligned} Z^t &= U^*(L^*X^t\Theta), \\ L^* &= \text{GAT}(\overleftarrow{\mathcal{T}^*}, E), \\ U^* &= \text{GAT}(\overrightarrow{\mathcal{T}^*}, E), \end{aligned} \quad (13)$$

here, $\overleftarrow{\mathcal{T}^*} = \{\mathcal{V}, \overleftarrow{\mathcal{E}}\}$, where $\overleftarrow{\mathcal{E}} = \langle v, u_c \rangle | v \in \mathcal{V} \setminus \{u_c\}$. $\overrightarrow{\mathcal{T}^*} = \{\mathcal{V}, \overrightarrow{\mathcal{E}}\}$, where $\overrightarrow{\mathcal{E}} = \langle u_c, v \rangle | v \in \mathcal{V} \setminus \{u_c\}$. The computational complexity of graph convolution operations experiences a notable reduction in Eq.(13). To elaborate, the complexity in Eq.(11) is $O(2N)$, whereas it is diminished to $O(N)$ in Eq.(13).

Despite this enhancement, Eq.(11) still faces limitations in terms of hardware compatibility. At the hardware level, graph convolution operations are intrinsically linked to the sparse and irregular nature of graph structures. This characteristic might not be compatible with certain hardware architectures, leading to an increased frequency of random memory accesses and limited opportunities for data reuse. Consequently, this can result in significantly higher inference latency for graph convolutions when compared to other neural network architectures. Then, we introduce a self-loop to the central node u_c in both $\overleftarrow{\mathcal{T}^*}$ and $\overrightarrow{\mathcal{T}^*}$. Consequently, we reformulate Eq.(13) to a network namely GWT-AGCN as follows:

$$Z^t = \text{Softmax}(\text{ReLU}(Ee_c^\top)) \text{Softmax}(\text{ReLU}(e_cE^\top)) X^t\Theta \quad (14)$$

Here, $e_c \in \mathbb{R}^{1 \times d}$ represents the node embedding vector of node u_c . This GWT-AGCN layer can serve as an alternative to the AGCN layer in constructing ASTGNNS.

The advantages of Eq.(14) are manifold: The equation solely comprises matrix multiplication and standard activation functions, thereby enhancing its compatibility with hardware. In contrast to

Table 2: Spatial-temporal datasets statistics.

Dataset	#Nodes	Time Range
PEMS07	883	05/01/2017-08/06/2017
CA	8,600	01/01/2017-12/31/2021
GLA	3,834	01/01/2019-12/31/2019
GBA	2,352	01/01/2019-12/31/2019
SD	716	01/01/2019-12/31/2019

Eq.(13), the complexity increased by only $O(2)$, a change that can be considered inconsequential.

Central Node Selection. Owing to the non-uniqueness of \mathcal{T}^* in the complete graph \mathcal{K}_N , directly employing \mathcal{T}^* for training ASTGNNs presents the challenge of central node selection. Viewed through the lens of AGCN, the random selection of a node u_c from the vertex set \mathcal{V} is analogous to initializing the node embedding e_c randomly. This approach, however, might introduce bias in the construction of the adaptive graph. To ensure that the selected central node embedding vector e_c is positioned at the physical center of the node embedding space E , we opt for a setting where $e_c = \text{Mean}(E)$, a technique we refer to as averaged initialization. We empirically show that such operation provides better on the prediction accuracy (see Sec. 5.3).

5 EVALUATION

5.1 Experimental Settings

In this section, we conduct extensive experiments to validate our Hypothesis 1.

Neural Network Architecture. We evaluate the existence of GWT on two quintessential ASTGNN architectures: **AGCRN** and **Graph WaveNet (GWNET)**. AGCRN integrates an RNN framework, specifically combining AGCN layers with Gated Recurrent Unit (GRU) layers. The AGCN layers are adept at capturing spatial dependencies, whereas the GRU layers are employed to model the temporal dependencies effectively. Conversely, GWNET represents a CNN-based ASTGNN architecture. It amalgamates AGCN, GCN layers, and dilated 1D convolution networks. Here, both GCN and AGCN layers are instrumental in capturing spatial dependencies, whilst the dilated 1D convolution networks are utilized to model the temporal dependencies. AGCRN* and GWENT* respectively represent AGCRN and GWNET trained within \mathcal{T}^* , while AGCRN* and GWNET* represent AGCRN and GWNET with in GWT-AGCN described in Sec. 4.2, respectively.

Datasets. We conduct experiments on five of the largest known spatial-temporal datasets. These include PEMS07, a dataset extensively studied [5], along with SD, GBA, GLA, and CA, which were recently introduced in the LargeST dataset [20]. Table 2 summarizes the specifications of the datasets used in our experiments. These datasets were partitioned in a 6:2:2 ratio for training, validation, and testing, respectively. The traffic flow data in PEMS07 is aggregated into 5-minute intervals, whereas for SD, GBA, GLA, and CA, the aggregation occurs in 15-minute intervals. We implemented a 12-sequence-to-12-sequence forecast, adhering to the standard protocol in this research domain.

Implementation Details. For all evaluated models, we set the number of training iterations to 100. Other training-related configurations adhere to the recommended settings provided in the respective code repositories. To ensure reproducibility and reliability, experiments were conducted ten times on all datasets, except for CA and GLA. Due to their substantially larger data scales, experiments on CA and GLA were limited to three repetitions. These experiments were performed on an NVIDIA RTX A6000 GPU, equipped with 48 GB of memory.

Metrics. Our comprehensive evaluation encompasses the following dimensions: *i) Performance:* We assess the forecasting accuracy using three established metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE), and *ii) Efficiency:* Model efficiency is evaluated in terms of both training and inference wall-clock time. Additionally, the batch size during training is reported, reflecting the models' capability to manage large-scale datasets. We set a maximum batch size limit of 64. If a model is unable to operate with this configuration, we progressively reduce the batch size to the highest possible value that fully utilizes the memory capacity of the A6000 GPU.

5.2 Main Results

The experimental results are organised as follows: Test accuracies and efficiency comparisons are reported in Table 3 and Table 4, respectively. We also compare the

We make following observations from Table 3 and Table 4:

- Graph lottery tickets are existent in ASTGNNs. Specifically, AGCRN* and GWENT* demonstrate performance that is comparable or even superior across all datasets. These findings indicate that \mathcal{T}^* is a stable 'graph winning lottery ticket' within ASTGNNs when evaluated on datasets such as PEMS07, SD, GBA, GLA, and CA.
- Our proposed approach is demonstrably scalable. The CA dataset presents substantial challenges to existing ASTGNNs, evidenced by AGCRN's inability to operate on it. However, the proposed approach facilitates the training of AGCRN on the CA dataset. This capability not only underscores the scalability of the proposed approach but also its superiority. Conventional pruning-based methods necessitate starting the training process with a complete graph. This approach often leads to their inadequacy in identifying graph lottery tickets in large-scale datasets like CA, a limitation that the proposed approach effectively overcomes.
- GWT-AGCN has the potential to be an ideal substitute for AGCN. In comparison, ASTGNN within GWT-AGCN demonstrates enhanced overall performance, particularly in terms of speed, surpassing its predecessor.
- GWT-AGCN significantly accelerates the training and inference of the ASTGNNs. The acceleration is more prominent on AGCRNs against GWNETs because a larger portion of the total computation required by GWNETs is used on their GCN layers.

Table 3: Performance comparisons. We bold the best results. Standard deviations are suppressed for the sake of room.

Data	Method	Horizon3			Horizon6			Horizon12			Average		
		MAE	RMSE	MAPE									
PEMS07	AGCRN	19.31	31.68	8.18%	20.70	34.52	8.66%	22.74	37.94	9.71%	20.64	34.39	8.74%
	AGCRN*	19.36	31.56	8.11%	20.67	33.95	8.63%	22.76	37.32	9.59%	20.67	33.95	8.67%
	AGCRN*	19.27	31.83	8.09%	20.57	34.41	8.57%	22.63	37.97	9.39%	20.57	34.42	8.59%
	GWNET	18.69	30.69	8.02%	20.26	33.37	8.56%	22.79	37.11	9.73%	20.25	33.32	8.63%
	GWNET*	18.64	30.61	8.01%	20.36	33.57	8.68%	22.39	36.61	9.51%	20.09	33.13	8.59%
	GWNET*	18.48	30.42	8.20%	19.91	32.98	8.71%	21.81	36.21	9.72%	19.80	32.84	8.62%
SD	AGCRN	15.71	27.85	11.48%	18.06	31.51	13.06%	21.86	39.44	16.52%	18.09	32.01	13.28%
	AGCRN*	16.06	28.56	11.59%	18.32	31.65	12.41%	22.67	39.06	16.13%	18.56	31.89	13.19%
	AGCRN*	15.49	25.89	10.55%	18.12	30.96	12.13%	22.24	38.79	15.36%	18.13	30.92	12.41%
	GWNET	15.24	25.13	9.86%	17.74	29.51	11.70%	21.56	36.82	15.13%	17.74	29.62	11.88%
	GWNET*	15.19	24.97	10.09%	17.39	29.01	11.82%	21.61	36.55	15.35%	17.90	29.56	13.01%
	GWNET*	15.24	25.07	10.71%	17.87	29.18	12.21%	21.94	36.69	15.17%	17.89	30.06	12.57%
GBA	AGCRN	18.31	30.24	14.27%	21.27	34.72	16.89%	24.85	40.18	20.80%	21.01	34.25	16.90%
	AGCRN*	18.24	30.18	14.09%	21.27	34.37	16.77%	24.82	39.68	20.80%	20.71	33.75	15.93%
	AGCRN*	17.62	29.49	12.99%	20.73	34.25	15.65%	24.72	40.37	18.72%	20.51	33.57	15.33%
	GWNET	17.85	29.12	13.92%	21.11	33.69	17.79%	25.58	40.19	23.48%	20.91	33.41	17.66%
	GWNET*	17.72	28.98	13.86%	20.95	33.47	17.81%	25.55	41.02	23.68%	20.96	34.16	17.79%
	GWNET*	17.76	28.97	13.53%	21.09	33.46	17.58%	26.01	40.63	24.11%	21.01	33.38	17.66%
GLA	AGCRN	17.27	29.70	10.78%	20.38	34.82	12.70%	24.59	42.59	16.03%	20.25	34.84	12.87%
	AGCRN*	17.12	28.10	10.81%	20.42	33.02	11.77%	24.48	42.33	15.83%	20.15	32.57	12.17%
	AGCRN*	16.85	27.38	10.83%	20.13	32.90	11.84%	24.71	40.36	14.85%	20.04	32.72	12.03%
	GWNET	17.28	27.68	10.18%	21.31	33.70	13.02%	26.99	42.51	17.64%	21.20	33.58	13.18%
	GWNET*	17.18	27.32	10.65%	21.0	33.00	13.29%	26.39	42.01	17.05%	21.10	32.97	13.34%
	GWNET*	17.16	27.15	10.87%	21.23	33.17	13.12%	26.41	41.95	17.10%	21.08	33.03	13.13%
CA	AGCRN	—	—	—	—	—	—	—	—	—	—	—	—
	AGCRN*	16.56	26.88	11.93%	20.13	31.87	15.25%	24.59	39.65	19.86%	19.77	31.79	15.21%
	AGCRN*	16.44	26.97	12.05%	19.90	32.20	15.15%	24.78	39.55	19.90%	19.78	31.98	15.19%
	GWNET	17.14	27.81	12.62%	21.68	34.16	17.14%	28.58	44.13	24.24%	21.72	34.20	17.40%
	GWNET*	17.19	28.16	10.09%	22.03	24.66	13.62%	27.05	40.83	21.93%	21.38	33.16	16.58%
	GWNET*	17.44	28.08	10.49%	21.23	24.06	13.12%	27.08	41.39	22.33%	21.08	32.90	16.12%

Table 4: Efficiency comparisons. BS: batch size set during training. Train: training time (in seconds) per epoch. Infer: inference time (in seconds) on the validation set. Total: total training time (in hours).

Method	PEMS07				SD				GBA				GLA				CA			
	BS	Train	Infer	Total	BS	Train	Infer	Total	BS	Train	Infer	Total	BS	Train	Infer	Total	BS	Train	Infer	Total
AGCRN	64	131	22	4	64	92	15	3	64	536	83	17	45	1413	245	46	—	—	—	—
AGCRN*	64	86	14	3	64	78	13	3	64	222	33	7	45	430	69	14	11	1513	196	47
AGCRN*	64	56	10	2	64	55	10	2	64	153	24	4	64	286	46	9	16	1021	132	38
GWNet	64	161	21	5	64	82	12	3	64	483	66	15	64	1028	139	32	44	4105	548	113
GWNet*	64	138	15	4	64	76	12	3	64	380	55	12	64	818	112	26	50	2583	319	81
GWNet*	64	116	13	4	64	71	11	2	64	327	47	1	64	700	96	19	50	2382	295	74

5.3 Analysis

Convergence. Figure 3 illustrates the training loss and test Mean Absolute Error (MAE) curves of the original AGCRN and AGCRN* under identical hyper-parameter settings on the PEMS07 dataset. Similarly, Figure 4 presents these curves for the same models on the SD dataset. Pre-determined GWT ensures convergence that is as consistent, rapid, and stable as a complete graph model. This feature is particularly advantageous for training on large-scale spatial-temporal data, as it significantly reduces computational overhead without compromising the quality of convergence.

Additionally, the convergence behavior of AGCRN* demonstrates its robustness in capturing complex spatial-temporal dependencies. This attribute is crucial for reliable forecasting in dynamic systems, such as traffic networks, where understanding intricate patterns is key to accuracy.

AGCRN* & GWNET* vs. SOTAs. AGCRN and GWNET, as representative ASTGNNs introduced between 2019 and 2020, are of significant interest in our study. Our objective is to evaluate the performance of AGCRN and GWNET, particularly when trained using GWT, in comparison with the current state-of-the-art STGNNs. To

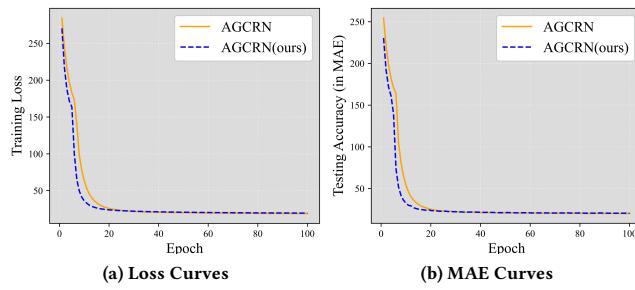


Figure 3: Training loss (a) and testing MAE (b) curve of original AGCRN and AGCRN* trained on PEMS07, respectively.

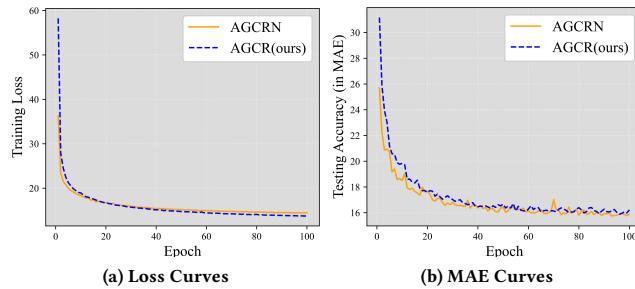


Figure 4: Training loss (a) and testing MAE (b) curve of original AGCRN and AGCRN* trained on SD, respectively.

this end, we selected five advanced STGNNs as baselines: DGCRN [19], MegaCRN [17], STGODE [11], D²STGNN [22], and DSTAGNN [18]. These models reflect the most recent trends in the field. DGCRN and MegaCRN, seen as variations of AGCRN, epitomize the latest developments in ASTGNN. STGODE employs neural ordinary differential equations innovatively to effectively model the continuous dynamics of traffic signals. In contrast, DSTAGNN and D²STGNN focus on capturing the dynamic correlations among sensors in traffic networks. From the results presented in Table 6, we make the following observations: *i*) ASTGNNs such as DGCRN and MegaCRN consistently exhibit strong performance across most benchmarks. However, their intricate model designs limit scalability, particularly in larger datasets like GLA and CA. *ii*) Methods introduced four years ago, such as AGCRN when trained within GWT-AGCN (*i.e.*, AGCRN*), continue to demonstrate robust performance across various evaluated datasets. Remarkably, they achieve state-of-the-art performance on specific datasets including GBA, GLA, and CA. These findings suggest that GWT-AGCN could play a crucial role in the development of scalable ASTGNNs for future research.

Impact of averaged initialization of node embedding e_c . In this study, we employ AGCRN* as a benchmark to evaluate the impact of averaged initialization of node embedding e_c . Table 7 presents a comparative analysis between AGCRN* and AGCRN*, *i.e.*, random initialization of e_c . The results indicate that AGCRN* consistently outperforms AGCRN* in terms of forecasting accuracy. This finding underscores the significance of deliberate initialization

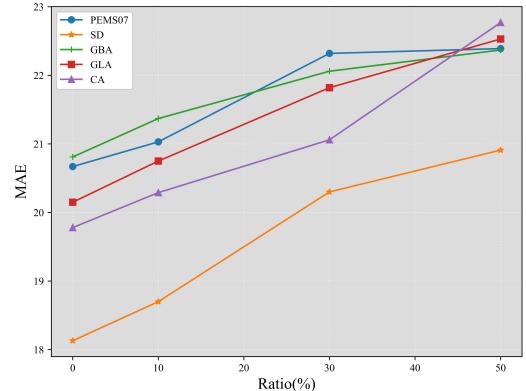


Figure 5: Testing accuracies, measured in MAE, for AGCRN on the PEMS07, SD, GBA, GLA, and CA datasets, with a perturbation ratio of p ranging from 0% to 50%.

strategies for e_c in enhancing the predictive performance of the model.

Comparison with AGS. We compared our method with AGS, the state-of-the-art approach, to validate its superiority. The performance of AGS with a sparsity of 99.7% is reported on PEMS07, SD, GBA and GLA, while the sparsity of our method is 99.8%/99.7%/99.99%/99.99% for PEMS07/SD/GBA/GLA. Since AGS does not provide an implementation on GWNet, we only report the results for AGCRN. The lack of CA results is due to AGS encountering out-of-memory (OOM) issues. From Table 5, we can see that our method significantly outperforms AGS.

Perturbed \mathcal{T}^* . We attribute the effectiveness of \mathcal{T}^* to its robust connectivity, which is crucial for ASTGNN's ability to model global spatial dependencies. To further validate this perspective, we introduce a perturbation process illustrated in Figure 6 to \mathcal{T}^* , resulting in $\tilde{\mathcal{T}}^*$, according to the following steps:

- (1) For a given \mathcal{T}^* with N nodes, we randomly remove M edges between the center node and the leaf node. The perturbation ratio p of the removed edges is defined as $\frac{M}{N-1}$.
- (2) Subsequently, we randomly add M new edges, connecting previously isolated nodes.

These steps intentionally disrupt the original connectivity in \mathcal{T}^* , while ensuring that the overall sparsity of the network remains constant. Figure 5 show the MAE curves of AGCRN trained via $\tilde{\mathcal{T}}^*$ of a ratio p from 0 to 50%. We can see that as p increases, the accuracy of the model decreases. This indicates the importance of preserving the graph's connectivity to model global spatial dependencies.

6 CONCLUSION

This paper introduces a novel approach in the realm of ASTGNNs by leveraging the GWT concept, inspired by the Lottery Ticket Hypothesis. This method markedly reduces the computational complexity of ASTGNNs, transitioning from a quadratic to a linear scale, thereby streamlining their deployment. Our innovative strategy of adopting a star topology for GWT, without necessitating exhaustive

Table 5: Comparative experimental results between AGS and our method within AGCRN.

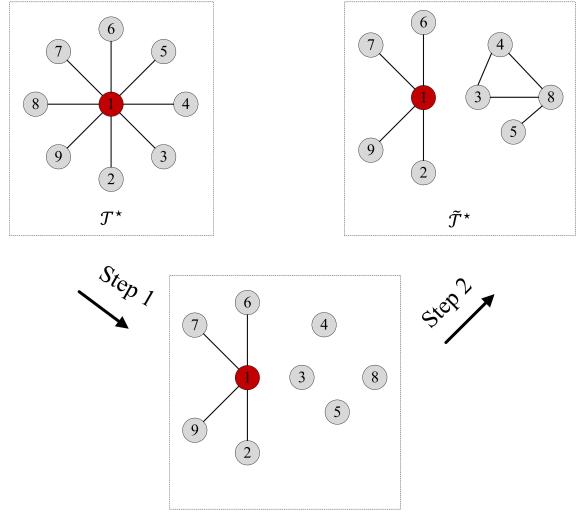
Methods	PEMS07			SD			GBA			GLA		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
AGCRN(AGS)@99.7%	24.37	39.33	9.57%	21.83	37.10	15.03%	23.61	38.32	17.19%	22.23	35.55	13.39%
AGCRN*	20.67	33.95	8.67%	18.56	31.89	13.19%	20.71	33.75	15.95%	20.15	32.57	12.17%
AGCRN*	20.57	34.42	8.59%	18.13	30.92	12.41%	20.51	33.57	15.33%	20.04	32.72	12.03%

Table 6: AGCRN* and GWNET* compared with current state-of-the-art STGNNs. We report the results averaged in 12 horizons. Results without standard deviations are sourced from published papers.

Dataset	Method	MAE	RMSE	MAPE(%)
PEMS07	AGCRN*	20.57±0.11	34.42±0.05	8.59±0.13
	GWNET*	19.80±0.19	32.84±0.25	8.62±0.11
	DGCRN	20.5±0.23	33.32±0.23	8.45±0.20
	MegaCRN	19.86±0.36	32.69±0.36	8.62±0.21
	STGODE	22.99	37.54	10.14
	D2STGNN	20.50	33.08	8.42
SD	DSTAGNN	20.50	34.51	9.01
	AGCRN*	18.13±0.30	30.92±0.32	12.41±0.10
	GWNET*	17.89±0.18	30.06±0.19	12.57±0.21
	DGCRN	18.02	30.09	12.07
	MegaCRN	17.76±0.21	29.62±0.17	12.69±0.13
	STGODE	21.79	35.37	13.22
GBA	D2STGNN	17.85	29.51	11.54
	DSTAGNN	21.82	34.68	14.40
	AGCRN*	20.51±0.13	33.57±0.27	15.3±0.17
	GWNET*	21.01±0.07	33.38±0.30	17.66±0.15
	DGCRN	20.91	33.83	16.88
	MegaCRN	20.69±0.17	33.61±0.14	15.52±0.09
GLA	STGODE	21.79	35.37	18.26
	D2STGNN	20.71	33.65	15.04
	DSTAGNN	23.82	37.29	20.16
	AGCRN*	20.04±0.07	32.72±0.19	12.03±0.18
CA	GWNET*	21.08±0.05	33.03±0.11	13.13±0.20
	STGODE	21.49	36.14	13.72%
	DSTAGNN	24.13	38.15	15.07%

Table 7: Ablation study of averaged initialization of e_c . We report the results averaged in 12 horizons.

Dataset	Method	MAE	RMSE	MAPE(%)
PEMS07	AGCRN*	20.57±0.11	34.42±0.05	8.59±0.13
	AGCRN*	20.93±0.17	35.17±0.12	8.87±0.26
SD	AGCRN*	18.13±0.30	30.92±0.32	12.41±0.10
	AGCRN*	18.63±0.28	32.19±0.29	12.67±0.24
GBA	AGCRN*	20.51±0.13	33.57±0.27	15.3±0.17
	AGCRN*	21.16±0.37	34.28±0.30	16.12±0.35
GLA	AGCRN*	20.04±0.07	32.72±0.19	12.03±0.18
	AGCRN*	20.23±0.09	33.16±0.20	12.32±0.17
CA	AGCRN*	19.78±0.08	31.98±0.29	15.59±0.43
	AGCRN*	20.01±0.06	32.64±0.19	16.73±0.56

**Figure 6: Perturbation process with a perturbation ratio of p , with a pre-specified node number.**

training cycles, maintains high model performance with significantly lower computational demands. Empirical validations across various datasets underscore our method's capability to achieve performance on par with full models, but at a fraction of the computational cost. This breakthrough not only underscores the existence of efficient sub-networks of the spatial graphs within ASTGNNs, but also extends the applicability of the Lottery Ticket Hypothesis to scenarios where resources are limited. Consequently, this work represents a significant leap forward in the optimization and practical application of graph neural networks, particularly in environments where computational resources are constrained. In the future, we will develop new STGNNs based on pre-determined GWT, aimed at long-term spatial-temporal forecasting.

REFERENCES

- [1] Daniel A.Spielman. [n. d.]. Spectral and Algebraic Graph Theory. Online. <http://cs-www.cs.yale.edu/homes/spielman/sagt/sagt.pdf>
- [2] Lei Bai, Lina Yao, Salil S. Kanhere, Xianzhi Wang, and Quan Z. Sheng. 2019. STG2Seq: Spatial-Temporal Graph to Sequence Model for Multi-step Passenger Demand Forecasting. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, Sarit Kraus (Ed.). ijcai.org, 1981–1987.
- [3] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. 2020. Adaptive graph convolutional recurrent network for traffic forecasting. In *Advances in Neural Information Processing Systems*. 17804–17815.
- [4] Joshua Batson et al. 2013. Spectral sparsification of graphs: theory and algorithms. *Commun. ACM* 56, 8 (2013), 87–94.
- [5] Chao Chen, Karl Petty, Alexander Skabardonis, Pravin Varaiya, and Zhanfeng Jia. 2001. Freeway Performance Measurement System: Mining Loop Detector Data. *Transportation Research Record* 1748, 1 (2001), 96–102. <https://doi.org/10.3141/1748-12>
- [6] Tianlong Chen, Yongduo Sui, Xuxi Chen, Aston Zhang, and Zhangyang Wang. 2021. A Unified Lottery Ticket Hypothesis for Graph Neural Networks. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 1695–1706.
- [7] Xiaohan Chen, Yu Cheng, Shuohang Wang, Zhe Gan, Zhangyang Wang, and Jingjing Liu. 2021. EarlyBERT: Efficient BERT Training via Early-bird Lottery Tickets. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2195–2207.
- [8] Yuzhou Chen, Ignacio Segovia-Dominguez, Baris Coskunuzer, and Yulia R. Gel. 2022. TAMP-S2GCNets: Coupling Time-Aware Multipersistence Knowledge Representation with Spatio-Supra Graph Convolutional Networks for Time-Series Forecasting. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.
- [9] Jeongwhan Choi, Hwangyong Choi, Jeehyun Hwang, and Noseong Park. 2022. Graph Neural Controlled Differential Equations for Traffic Forecasting. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelfth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*. AAAI Press, 6367–6374.
- [10] Wenying Duan, Xiaoxi He, Zima Zhou, Lothar Thiele, and Hong Rao. 2023. Localised Adaptive Spatial-Temporal Graph Neural Network. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*. ACM, 448–458. <https://doi.org/10.1145/3580305.3599418>
- [11] Zheng Fang, Qingqing Long, Guojie Song, and Kunqing Xie. 2021. Spatial-temporal graph node networks for traffic flow forecasting. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*. 364–373.
- [12] Jonathan Frankle and Michael Carbin. 2019. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In *International Conference on Learning Representations*.
- [13] Shengnan Guo, Youfang Lin, Huaiyu Wan, Xiucheng Li, and Gao Cong. 2021. Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. *IEEE Transactions on Knowledge and Data Engineering* 34, 11 (2021), 5415–5428.
- [14] Rongzhou Huang, Chuyin Huang, Yubao Liu, Genan Dai, and Weiyang Kong. 2020. LSGCN: Long Short-Term Traffic Prediction with Graph Convolutional Networks. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, Christian Bessiere (Ed.). ijcai.org, 2355–2361.
- [15] Jiawei Jiang, Chengkai Han, Wayne Xin Zhao, and Jingyuan Wang. 2023. PDFormer: Propagation Delay-Aware Dynamic Long-Range Transformer for Traffic Flow Prediction. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, Brian Williams, Yiling Chen, and Jennifer Neville (Eds.). AAAI Press, 4365–4373. <https://ojs.aaai.org/index.php/AAAI/article/view/25556>
- [16] Renhe Jiang, Zhaonan Wang, Jiawei Yong, Puneet Jeph, Quanjun Chen, Yasumasa Kobayashi, Xuan Song, Shintaro Fukushima, and Toyotaro Suzumura. 2023. Spatio-Temporal Meta-Graph Learning for Traffic Forecasting. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, Brian Williams, Yiling Chen, and Jennifer Neville (Eds.). AAAI Press, 8078–8086. <https://doi.org/10.1609/aaai.v37i7.25976>
- [17] Renhe Jiang, Zhaonan Wang, Jiawei Yong, Puneet Jeph, Quanjun Chen, Yasumasa Kobayashi, Xuan Song, Shintaro Fukushima, and Toyotaro Suzumura. 2023. Spatio-temporal meta-graph learning for traffic forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 37. 8078–8086.
- [18] Shiyong Lan, Yitong Ma, Weikang Huang, Wenwu Wang, Hongyu Yang, and Pyang Li. 2022. Dstagnn: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting. In *International conference on machine learning*. PMLR, 11906–11917.
- [19] Fuxian Li, Jie Feng, Huan Yan, Guangyin Jin, Fan Yang, Funing Sun, Depeng Jin, and Yong Li. 2023. Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution. *ACM Transactions on Knowledge Discovery from Data* 17, 1 (2023), 1–21.
- [20] Xu Liu, Yutong Xia, Yuxuan Liang, Junfeng Hu, Yiwei Wang, Lei Bai, Chao Huang, Zhenguang Liu, Bryan Hooi, and Roger Zimmermann. 2023. LargeST: A Benchmark Dataset for Large-Scale Traffic Forecasting. In *Advances in Neural Information Processing Systems*.
- [21] Hao Peng, Bowen Du, Mingsheng Liu, Mingzhe Liu, Shumei Ji, Senzhang Wang, Xu Zhang, and Lifang He. 2021. Dynamic graph convolutional network for long-term traffic flow prediction with reinforcement learning. *Inf. Sci.* 578 (2021), 401–416. <https://doi.org/10.1016/J.IJINS.2021.07.007>
- [22] Zezhi Shao, Zhao Zhang, Wei Wei, Fei Wang, Yongjun Xu, Xin Cao, and Christian S Jensen. 2022. Decoupled dynamic spatial-temporal graph neural network for traffic forecasting. *Proceedings of the VLDB Endowment* 15, 11 (2022), 2733–2746.
- [23] Kun Wang, Guohao Li, Shilong Wang, Guibin Zhang, Kai Wang, Yang You, Xiaojiang Peng, Yuxuan Liang, and Yang Wang. 2023. The snowflake hypothesis: Training deep GNN with one node one receptive field. *arXiv preprint arXiv:2308.10051* (2023).
- [24] Kun Wang, Yuxuan Liang, Xinglin Li, Guohao Li, Bernard Ghanem, Roger Zimmermann, Huahui Yi, Yudong Zhang, Yang Wang, et al. 2023. Brave the wind and the waves: Discovering robust and generalizable graph lottery tickets. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).
- [25] Kun Wang, Yuxuan Liang, Pengkun Wang, Xu Wang, Pengfei Gu, Junfeng Fang, and Yang Wang. 2023. Searching Lottery Tickets in Graph Neural Networks: A Dual Perspective. In *The Eleventh International Conference on Learning Representations*. <https://openreview.net/forum?id=Dvs-a3aymPe>
- [26] Kun Wang, Hao Wu, Guibin Zhang, Junfeng Fang, Yuxuan Liang, Yuankai Wu, Roger Zimmermann, and Yang Wang. 2024. Modeling spatio-temporal dynamical systems with neural discrete learning and levels-of-experts. *IEEE Transactions on Knowledge and Data Engineering* (2024).
- [27] Hao Wu, Yuxuan Liang, Wei Xiong, Zhengyang Zhou, Wei Huang, Shilong Wang, and Kun Wang. 2024. Earthfarser: Versatile Spatio-Temporal Dynamical Systems Modeling in One Model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 15906–15914.
- [28] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems* 32, 1 (2020), 4–24.
- [29] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Graph Wavenet for Deep Spatial-Temporal Graph Modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (Macao, China) (IJCAI'19)*. AAAI Press, 1907–1913.
- [30] Sijie Yan, Yuanjiong Xiong, and Dahua Lin. 2018. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, Sheila A. McIlraith and Kilian Q. Weinberger (Eds.). AAAI Press, 7444–7452.
- [31] Haoran You, Chaojian Li, Pengfei Xu, Yonggan Fu, Yue Wang, Xiaohan Chen, Yingyan Lin, Zhangyang Wang, and Richard G. Baraniuk. 2020. Drawing Early-Bird Tickets: Toward More Efficient Training of Deep Networks. In *International Conference on Learning Representations*.
- [32] Haoran You, Zhihan Lu, Zijian Zhou, Yonggan Fu, and Yingyan Lin. 2022. Early-bird gcn: Graph-network co-optimization towards more efficient gcn training and inference via drawing early-bird lottery tickets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 8910–8918.
- [33] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2018. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, Jérôme Lang (Ed.). ijcai.org, 3634–3640.

A APPENDIX

A.1 Proof of Proposition 1

Initially, in the case of $N = 3$ as illustrated in Figure 7, all spanning trees of this complete graph meet the diameter $r = 2$, and satisfy definition of star spanning tree in Hypothesis 1. Their count corresponds to the number of nodes N in the complete graph.

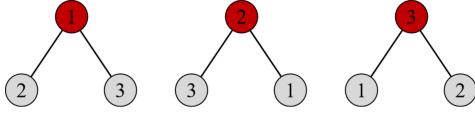


Figure 7: The 3-order spanning trees of the complete graph are all star spanning trees.

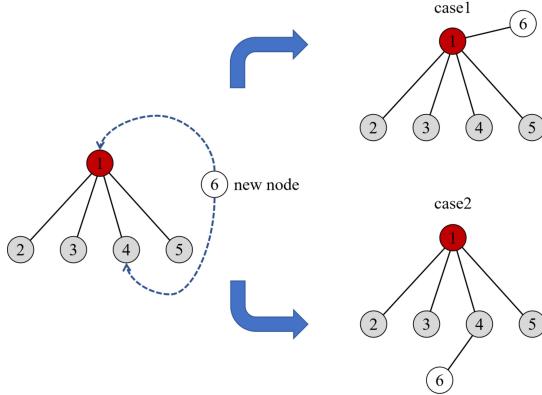


Figure 8: Adds the new node to the native tree.

Subsequently, assuming $N = k - 1$, the complete graph \mathcal{K}_{k-1} aligns with this conclusion, and the star spanning tree is \mathcal{T}_{k-1}^* .

In the scenario where $N = k$, the original graph is equivalent to inserting a new node into \mathcal{T}_{k-1}^* . Figure 8 shows two possible scenarios.

Only in the first scenario, does the spanning tree \mathcal{T}_k^* meet the diameter $r = 2$. The second scenario will increase some paths that are longer than 2. For the spanning tree \mathcal{T}_k^* formed in the first scenario, it still conforms to the definition of star spanning tree in Hypothesis 1.

A.2 Justify the effectiveness of star topology theoretically

Given two graphs \mathcal{G} and $\hat{\mathcal{G}}$, if $L_{\mathcal{G}} \leq L_{\hat{\mathcal{G}}}$, we denote this as: $\mathcal{G} \leq \hat{\mathcal{G}}$.

Here, $L_{\mathcal{G}}$ and $L_{\hat{\mathcal{G}}}$ represent the Laplacians of \mathcal{G} and $\hat{\mathcal{G}}$, respectively. The symbol \leq denotes the Loewner partial order, applicable to certain pairs of symmetric matrices.

The Courant-Fisher Theorem provides that:

$$\lambda_i(A) = \max_{S: \dim(S)=i} \min_{x \in S} \frac{x^T Ax}{x^T x}. \quad (15)$$

Thus, assuming $\lambda_1, \dots, \lambda_N$ are the eigenvalues of $L_{\mathcal{G}}$ and $\hat{\lambda}_1, \dots, \hat{\lambda}_N$ are the eigenvalues of $L_{\hat{\mathcal{G}}}$. The relation $L_{\mathcal{G}} \leq L_{\hat{\mathcal{G}}}$ means for all i , $\lambda_i \leq \hat{\lambda}_i$.

Graph Spectral Similarity. [1][4] If $L_{\hat{\mathcal{G}}} / \sigma \leq L_{\mathcal{G}} \leq \sigma L_{\hat{\mathcal{G}}}$, we say graphs \mathcal{G} and $\hat{\mathcal{G}}$ are σ -spectral similar. Thus, $\hat{\mathcal{G}}$ is a σ -approximation of \mathcal{G} .

Based on spectral graph theory [1][4], if a graph is a σ -approximation of another one. We mean they have similar eigensystems, therefore with similar properties. Thus, if $L_{\mathcal{T}_N} / \sigma \leq L_{\mathcal{K}_N} \leq \sigma L_{\mathcal{T}_N}$, \mathcal{K}_N

and \mathcal{T}_N have similar properties. Such a \mathcal{T}_N can effectively replace \mathcal{K}_N to learn a good representation, where the edges of \mathcal{T}_N are much fewer than those of \mathcal{K}_N . Below, we will prove that \mathcal{T}_N is a σ -approximation of \mathcal{K}_N .

LEMMA 1. *The laplacian of \mathcal{K}_N has eigenvalue 0 with multiplicity 1 and N with multiplicity $N - 1$.*

PROOF OF LEMMA 1. To compute the non-zero eigenvalues, let ψ be any non-zero vector orthogonal to the all-1s vector, so

$$\sum_a \psi(a) = 0. \quad (16)$$

The Laplacian Matrix of a weighted graph $\mathcal{G} = (V, E, w)$, $w : E \rightarrow \mathbb{R}^+$, is designed to capture the Laplacian quadratic form:

$$\begin{aligned} (L_{\mathcal{G}} \mathbf{x})(a) &= \sum_{(a,b) \in E} w_{a,b} (\mathbf{x}(a) - \mathbf{x}(b)) \\ &= d(a)\mathbf{x}(a) - \sum_{(a,b) \in E} w_{a,b}\mathbf{x}(b). \end{aligned} \quad (17)$$

We now compute the first coordinate of $L_{\mathcal{K}_N} \psi$. Using the expression for the action of the Laplacian as an operator, we find

$$\begin{aligned} (L_{\mathcal{K}_N} \psi)(1) &= \sum_{v \geq 2} (\psi(1) - \psi(v)) \\ &= (n-1)\psi(1) - \sum_{v=2}^n \psi(v) = n\psi(1). \end{aligned} \quad (18)$$

As the choice of coordinate was arbitrary, we have $L\psi = N\psi$. So, every vector orthogonal to the all-1s vector is an eigenvector of eigenvalue N . \square

LEMMA 2. *Let $\mathcal{G} = (V, E)$ be a graph, and let a and b be vertices of degree one that are both connected to another vertex c . Then, the vector $\psi = \delta_a - \delta_b$ is an eigenvector of $L_{\mathcal{G}}$ of eigenvalue 1*

PROOF OF LEMMA 2. Just multiply $L_{\mathcal{G}}$ by ψ , and check (using (17)) vertex-by-vertex that it equals ψ .

As eigenvectors of different eigenvalues are orthogonal, this implies that $\psi_a = \psi_b$ for every eigenvector with eigenvalue different from 1. \square

LEMMA 3. *The laplacian of \mathcal{T}_N has eigenvalue 0 with multiplicity 1, eigenvalue 1 with multiplicity $N - 2$, and eigenvalue N with multiplicity 1.*

PROOF OF LEMMA 3. Applying Lemma 2.1 to vertices i and $i + 1$ for $2 \leq i < N$, we find $N - 2$ linearly independent eigenvectors of the form $\delta_i - \delta_{i+1}$, all with eigenvalue 1. As 0 is also an eigenvalue, only one eigenvalue remains to be determined. Recall that the trace of a matrix equals both the sum of its diagonal entries and the sum of its eigenvalues. We know that the trace of $L_{\mathcal{T}_N}$ is $2N - 2$, and we have identified $N - 1$ eigenvalues that sum to $N - 2$. So, the remaining eigenvalue must be N . \square

From Lemma 1-3, we deduce:

LEMMA 4. *\mathcal{T}_N is an N -approximation of \mathcal{K}_N .*

PROOF OF LEMMA 5. Assume $\lambda_1, \dots, \lambda_N$ are the eigenvalues of The Laplacian of \mathcal{K}_N , and u_1, \dots, u_N are the eigenvalues of the Laplacian of \mathcal{T}_N . For $i = 1$, $\lambda_i = N$, $u_i = N$, satisfying $u_i/N \leq \lambda_i \leq Nu_i$. For $2 \leq i \leq N - 1$, $\lambda_i = N$, $u_i = 1$, satisfying $u_i/N \leq \lambda_i \leq Nu_i$.

For $i = N$, $\lambda_i = 0$, $u_i = 0$, satisfying $u_i/N \leq \lambda_i \leq Nu_i$. Thus, for all i , $u_i/N \leq \lambda_i \leq Nu_i$, i.e., $L_{\mathcal{T}_N}/N \leq L_{\mathcal{K}_N} \leq NL_{\mathcal{T}_N}$. \square

In conclusion, we have theoretically proven that star topology \mathcal{T}_N is a good approximation of \mathcal{K}_N , and therefore, can learn spatiotemporal dependencies effectively.