electronics

Article

A Dynamic Short Cascade Diffusion Prediction Network Based on Meta-Learning-Transformer

Gang Li [†], **Tao Meng** [†], **Min Li**, **Mingle Zhou** * and **Delong Han**

Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan 250014, China
* Correspondence: zhouml@qlu.edu.cn
† These authors contributed equally to this work.

Abstract: The rise of social networks has greatly contributed to creating information cascades. Over time, new nodes are added to the cascade network, which means the cascade network is dynamically variable. At the same time, there are often only a few nodes in the cascade network before new nodes join. Therefore, it becomes a key task to predict the diffusion after the dynamic cascade based on the small number of nodes observed in the previous period. However, existing methods are limited for dynamic short cascades and cannot combine temporal information with structural information well, so a new model, MetaCaFormer, based on meta-learning and the Transformer structure, is proposed in this paper for dynamic short cascade prediction. Considering the limited processing capability of traditional graph neural networks for temporal information, we propose a CaFormer model based on the Transformer structure, which inherits the powerful processing capability of Transformer for temporal information, while considering the neighboring nodes, edges and spatial importance of nodes, effectively combining temporal and structural information. At the same time, to improve the prediction ability for short cascades, we also fuse meta-learning so that it can be quickly adapted to short cascade data. In this paper, MetaCaFormer is applied to two publicly available datasets in different scenarios for experiments to demonstrate its effectiveness and generalization ability. The experimental results show that MetaCaFormer outperforms the currently available baseline methods.

Keywords: dynamic short cascade; social networks; meta-learning; Transformer; prediction

check for updates

Citation: Li, G.; Meng, T.; Li, M.; Zhou, M.; Han, D. A Dynamic Short Cascade Diffusion Prediction Network Based on Meta-Learning-Transformer. *Electronics* **2023**, *12*, 837. https://doi.org/10.3390/electronics12040837

Academic Editor: Mazdak Zamani

Received: 7 January 2023
Revised: 4 February 2023
Accepted: 6 February 2023
Published: 7 February 2023

1. Introduction

Information dissemination in social networks is characterized by fast speed, comprehensive coverage and profound influence. The accurate prediction of information dissemination trends is of great significance to individuals, governments and enterprises. Since the speed of communication on social platforms is directly proportional to the initial retweets, there is an interactive relationship between users [1]. It has become an important topic to understand such an intrinsic law and to predict the subsequent information based on the cascade of information observed in the previous period [2,3]. However, online social media's open and competitive nature makes the information forwarding dynamic, and the length of the initially observed cascade varies. There needs to be more than a simple combination of existing models to handle such dynamic cascade networks.

The connection prediction of a dynamic cascade network can be viewed as predicting the future relationship of a set of nodes where messages are passed. Dynamic cascade networks are not only limited to social networks [4,5], but also have a wide range of applications for scenarios, such as paper citation [6], product recommendation [7], virus propagation [8] and information security [9]. It also greatly facilitates the application of deep learning in blockchain [10–12]. Many methods have been proposed for predicting dynamic cascade networks in recent years. Dynamic graph neural networks [13] are the classical ones that apply nodes' aggregation and classification. However, it cannot give an accurate solution to short cascade prediction because of sharing triangular parameters.

Considering the problem that short cascades are difficult to predict, meta-learning [14] has emerged as a remedy since meta-learning has the property of autonomously learning model parameters, making it shine in the processing of small sample data. Meta-GNN [15] first proposed to combine meta-learning with graph processing models. Many graph processing models incorporating meta-learning were also proposed one after another [16–18]. However, most of these initial models were limited to static cascade networks and did not fully consider the importance of dynamic cascade networks in the importance of temporal preferences. Moreover, models such as TGAT [19], which simply pieced together meta-learning and graph neural networks, did not achieve the expected results and were not very general. Later, Cheng et al. [20] proposed a dynamic graph neural network incorporating meta-learning, which changed the previous problem of simple patchwork, made the model focus on temporal preference information, and enhanced the model's generalization ability. Although existing methods, such as MetaDyGNN [20], have achieved good results for dynamic short cascade prediction, the most critical temporal information in cascade diffusion is limited in graph neural networks, which cannot fully use temporal information. Therefore, better combining temporal and structural information in a meta-learning environment is an urgent challenge to be solved.

To address the above issues, the novel model MetaCaFormer is presented in this paper. In MetaCaFormer, meta-learning is designed to achieve autonomous learning and refine each task to the training of temporal preferences of nodes. Through multiple stages of adaptive learning, fast adaptive short cascades are constructed. Meanwhile, we introduce the Transformer [21] model in the context of meta-learning for processing temporal information. Since the traditional Transformer only works well for serialized information, it has limitations for processing spatially structured information, such as cascade graphs. So, inspired by Graphormer [22], we constructed a cascaded information processing model called CaFormer, which consists of an encoder and a classifier that continues the powerful capability of Transformer for processing temporal information.

The encoder processes the neighboring nodes, edges and spatial importance and feeds them to the classifier for classification. Finally, the potential future link prediction of the nodes is obtained. To verify the effectiveness of MetaCaFormer, experiments are conducted in this paper on two datasets with different scenarios. The results show that MetaCaFormer models can give better predictions.

In summary, the main contributions of this paper are listed as follows.

(1) Considering the way Graphormer processes graph-structured data, this paper creatively proposes a cascaded graph processing model named CaFormer by combining an improved Transformer structure as an encoder and a multilayer perceptron as the classifier. The model's perception of temporal information is enhanced, making it more applicable to the processing of dynamic cascade data.

(2) This paper organically integrates adaptive meta-learning with CaFormer to create a method called MetaCaFormer to solve the problem that short cascades are challenging to predict in dynamic networks, making the model more sensitive to short cascades and more capable of prediction.

(3) In this paper, sufficient comparison experiments are conducted with existing baseline methods. The experimental results show that MetaCaFormer always gives the best prediction, and sufficient ablation experiments are also conducted to prove the effectiveness of each component of MetaCaFormer.

The rest of the paper is as follows: In Section 2, related research on dynamic cascade networks is reviewed. The definition of some concepts and a detailed description of the workflow of the MetaCaFormer model are given in Section 3. Section 4 describes this paper's experiments and the data set and parameter settings. Section 5 further explains and discusses the experimental results and presents the model's advantages in this paper. Section 6 provides a summary and an outlook for the future.

2. Related Work

In this section, current research in graph structure processing and few-shot learning in graph structures are presented separately to help the reader understand each approach's contribution more clearly. A detailed comparison is shown in Table 1

Table 1. Comparison of existing methods with MetaCaFormer.

Method	Structure	Time	Dynamic	Few-Shot Learning	Advantage	Limitation
GCN [23]	✓				Perception of global information.	Limited evaluation.
GRN [24]	✓	✓			Convert graph data to sequence form.	Need for more understanding of global information.
GAT [25]	✓				Assigning weights to neighboring nodes.	Ignore time information.
GROVER [26]	✓	✓			Improved attention mechanism.	Limited assessment capacity.
GraphTransformer [27]	✓	✓			Sparsity of the graph is guaranteed.	Limited ability to handle structured data.
Graphormer [22]	✓	✓			Improved Transformer.	Limited ability to handle dynamic data.
DyRep [28]		✓	✓		Introduction of recurrent neural networks for node representation.	Ignore structural information.
GraphSAGE [29]	✓	✓	✓		Convert to snapshot form.	Low evaluation performance.
EvolveGCN [13]	✓	✓	✓		Apply to dynamic network node aggregation classification.	It has limitations in short cascades.
TGAT [19]	✓	✓	✓		Superimposed attention mechanism.	There are restrictions on short cascade predictions.
TGN [30]	✓	✓	✓		Efficient parallelism can be maintained.	Insufficient data processing capability for short cascades.
Meta-GNN [15]	✓	✓		✓	Combined with Meta-learning.	Limited performance.
Meta-Graph [31]	✓	✓		✓	Meta-learning and graph neural network fusion.	Limited by simple overlay.
G-Meta [17]	✓	✓		✓	Submap Modeling.	Limited ability to handle dynamic data.
TAdaNet [32]	✓	✓		✓	Multiple graph structure information combined with meta-learning for adaptive classification.	Insufficient assessment capacity.
MetaTNE [33]	✓	✓		✓	Introducing embedded conversion functions.	Limited by static network.
META-MGNN [34]	✓	✓		✓	Combining meta-learning and GCN.	Limited by static network.
MetaDyGNN [20]	✓	✓	✓	✓	Effective fusion of meta-learning and graph neural networks.	Limited by the graph network's ability to process temporal information.
MetaCaFormer	✓	✓	✓	✓	Adequate combination of structural and temporal information.	Limited ability to forecast macro.

2.1. Diagram Structure Processing

As cascading data become more widespread, the application of deep learning methods [35] to cascading data is gradually being explored. The concept of a graph neural network (GNN) [36] was proposed to enable deep learning to utilize scenarios related to graph data effectively. One of them, the graph convolutional neural network (GCN) [23], proposes to apply the Laplacian matrix to learn the topological structure and node feature information of graphs. The ability to perceive global information is improved. Graph recurrent networks (GRNs) [24] generally use bidirectional recurrent neural networks and long- and short-term memory networks as network architectures. The graph data are also converted into sequences, changing continuously during training. Later, Petar et al. [25] combined the attention mechanism with graph convolution to propose a graph attention network (GAT), which can calculate the importance of its neighboring nodes when each node updates its state and assign different weights to each neighboring node to improve the computational efficiency of the model. However, the graph neural network structure is relatively simple, and although it is more efficient, it has limitations in the prediction accuracy of complex graph networks. Later it was considered that Transformer could be applied to handle graph structure data. In [26], GNN was combined with Transformer to modify the generation of Q, K, and V matrices in the attention layer to produce node and edge representations. Dvivedi et al. [27] proposed to use the adjacency matrix as an attention mask and the Laplace eigenvector as a location encoding in transformer to ensure the sparsity of the graph. Graphormer [22] proposed to transform the graph data into three forms based on Transformer—center encoding, space encoding and edge encoding—which significantly improves the prediction accuracy. These methods improve the processing of graph-structured data to a certain extent, but actual forwarding data, such as social platforms, are often dynamic, and such methods usually do not cope well in the face of dynamic data.

To cope with these problems, methods such as ternary closure law [37], Hawkes process [38], and temporal random wandering [39] have been continuously proposed to be applied to the processing of temporal information of dynamic graph structures. Methods such as DyRep [28] propose using recurrent neural networks to represent node information, but more is needed to facilitate the extraction of structural information. Subsequent work usually transforms the graph structure into a snapshot form in temporal order, after which the temporal information [13,40,41] and the structural information are learned rationally to output node features. GraphSAGE [29] converts the graph into snapshot form and uses the feature information of neighboring nodes to generate node representations of unknown nodes efficiently. EvolveGCN [13] combines recurrent neural networks and GCNs to perform node representation adaptively. TGAT [19] combines GraphSAGE and GAT to incorporate attention mechanisms to facilitate node representation in dynamic networks. TGN [30] proposed to allow models to learn from data sequences, maintaining efficient parallel processing. However, these models usually take the whole graph or snapshots as input, do not consider few-shot learning in a graph structure, and do not perform well in the face of short cascaded data.

2.2. Exploration of Few-Shot Learning in Graph Structure

For the problem of complex short cascade processing, many solutions have emerged, with methods such as transfer learning [42], semi-supervised learning [43], domain adaptive [44,45] and few-shot learning [46–50]. Among them, few-shot learning is the most widely used method, whose main idea is to learn the underlying logic and shared parameters to fit a new task based on previously learned information quickly. It mainly includes two approaches: metric learning-based approaches [51] and meta-learning-based approaches [52,53]. Meta-GNN [15] was the first method to propose the application of MAML [54] to graph structure processing. However, this approach is limited for small samples. Attention has also turned to deep graph learning [55–58]. Meta-Graph [31] combines meta-learning with graph convolutional networks to propose connected predictions across multiple graphs. G-Meta [17] models subgraphs for classification and prediction. TAdaNet [32] proposes to apply multiple graph structure information combined with meta-learning for adaptive classification. MetaTNE [33] introduces an embedding transformation function that compensates for the direct use of meta-learning. META-MGNN [34] proposes to combine meta-learning and GCN for learning graph structure representation. However, the above methods, although considering small samples, are only applicable in static graph networks and do not effectively handle structural and temporal information in dynamic graphs. MetaDyGNN [20] proposes a method combining MAML and TGAT; due to the limitations of graph neural networks, it does not achieve better results in temporal processing information, resulting in a continuous degradation of performance over time.

3. Method

In dynamic cascade networks, predicting the future trends of nodes with only a few links is a more complex process. In this section, the definitions and explanations of some main concepts will be given first, followed by a detailed description of the proposed method in this paper. The meaning and introduction of the relevant symbols are shown in Table 2.

Table 2. Descriptions of key notations.

Notations	Descriptions
G	Static social network
V	Users/Nodes
E	The set of edges
$e = (v_i, v_j, t_e) \in E$	The edge of node v_i and node v_j at time t_e
$\mathcal{N}(v_i)$	The set of neighbors of node v_i
$T_v = (S_v, Q_v)$	The tasks T_v of node V include support set S_v and query set Q_v
E_ψ, C_ψ	Encoder, Classifier
A	The matrix capturing the similarity between Q and K
$Attn(\cdot)$	Self-Attentive computing
SP_{ij}	The shortest path from v_i to v_j
c_{ij}	edge encoding
w	weight
z^-, z^+	The incoming degree and outgoing degree of node v_i
$LN(\cdot)$	Layer Normalization
$MHA(\cdot)$	Multi-head attention.
$MLP(\cdot)$	The multilayer perceptron
$\mathcal{L}$	Loss calculation
p_e	The possibility of connectivity between nodes v_i and v_j
Ψ, ω	Parameter

3.1. Definition

3.1.1. Dynamic Cascade Network

Let $G = (V, E)$ be a dynamic cascade network, where V denotes the set of nodes and E denotes the set of edges. Each edge $e = (v_i, v_j, t_e) \in E$ denotes the edge of node v_i and node v_j at time t_e. And at different times, nodes v_i and v_j can have multiple different connected edges. Define $h_i^{(l)}$ as the representation of the l-th level v_i and $\mathcal{N}(v_i)$ as the set of neighbors of node v_i. A node that joins the dynamic cascade network after a given timestamp t is defined as a new node. The new node has timestamp $t' > t$.

3.1.2. Short Cascade Prediction

The presence of a new node in the dynamic cascade network $G = (V, E)$ indicates that the message has been further propagated, and the more nodes the message reaches proves that the message is more popular. Let $V_{\text{new}} \subset V$, which denotes the set of nodes appearing after timestamp t. The subsequent links are predicted based on the first K sets of nodes and their links. Since short cascade prediction is to be achieved, K is controlled to be an integer within 10 to achieve the effect of fewer observation cascades.

3.1.3. Parameter Adjustment

This paper also fuses meta-learning so that the model can learn the parameters autonomously. Divide the node set into support set S_v and query set Q_v, and set the initialized global parameters θ to feed into the CaFormer model. In the following, the parameters are updated by feedback after multi-step split training by the meta-learning model until the model achieves the best results.

3.2. Our Proposal

In this section, the MetaCaFormer framework based on meta-learning and Transformer is proposed, and the brief flow is shown in Figure 1. The cascaded subgraph is first divided into a support set and a query set, and then the support set is fed into the CaFormer model for training. Meanwhile, the meta-learner gives CaFormer an initial parameter, and with continuous training, the parameter is continuously iterated until a mature new model is generated after finding the optimal parameter. Finally, the model is tested with a query set to obtain the experimental results. The internal structure of the primary model and the relationship between the components are shown in Figure 2.

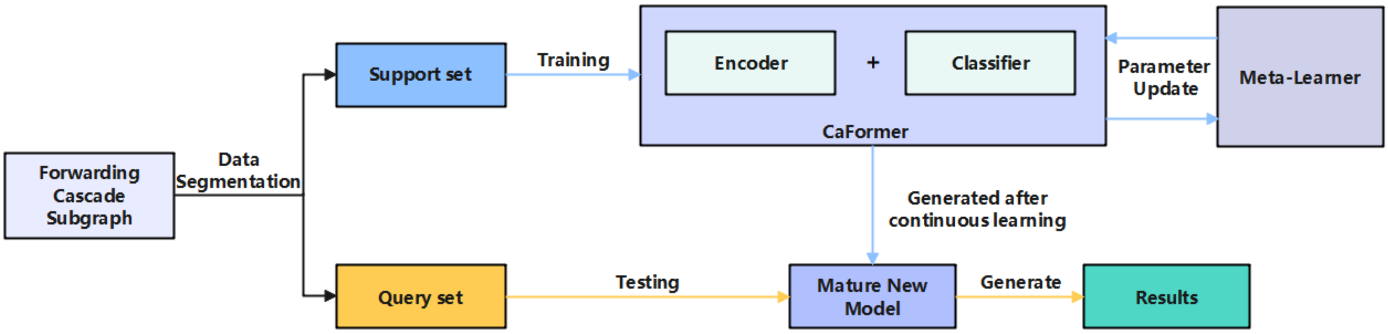

Figure 1. MetaCaFormer Brief Process.

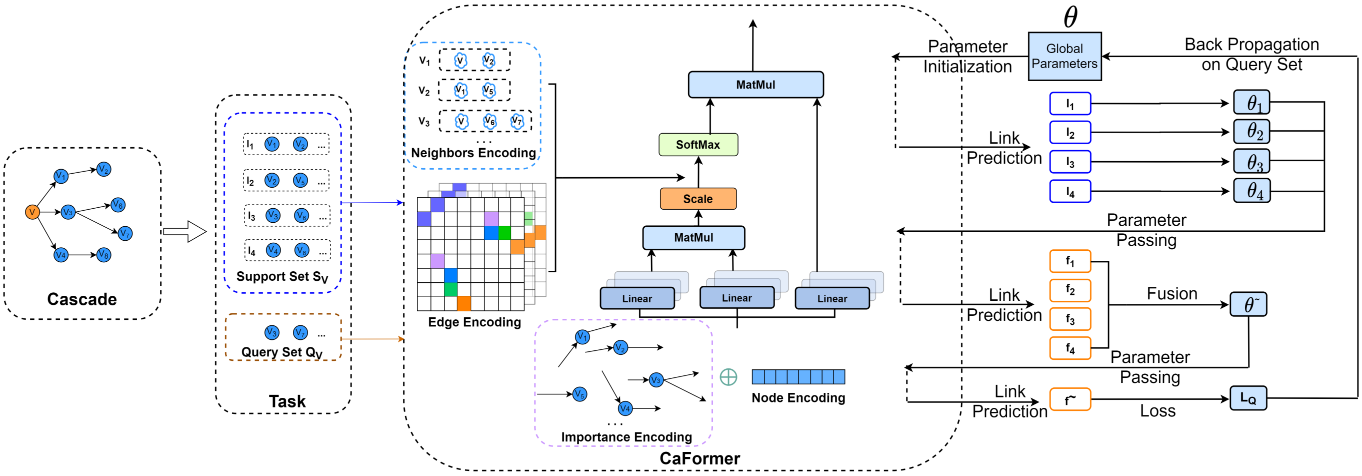

Figure 2. The framework of MetaCaFormer.

3.2.1. Task Formulation

Unlike previous data partitioning, the meta-learning framework divides the training data into tasks. Within each task, the training and test data are named the support and query sets. Meta-learning trains the parameter learning rules in the training task. The parameters can be quickly adapted to the new task in the testing stage. Each task can be viewed as a binary classification with the nodes' positive or negative linkage. As shown in Figure 3, the time span of each node v is divided into n equally spaced time intervals, i.e., $I_v = \{I_v^1, I_v^2, \ldots, I_v^n\}$. The division support set $\mathcal{S}_v = \{\mathcal{S}_v^1, \mathcal{S}_v^2, \ldots, \mathcal{S}_v^{n-1}\}$, where $\mathcal{S}_v^i$ is the set of links in the i-th interval. The query set Q_v is the set of links in the last interval I_v^n. The task of node v is defined as

$$T_v = (S_v, Q_v) \tag{1}$$

In this paper, K links are sampled as the support set of each node. Such a process ensures that the number of links in different time intervals is approximately equal. To achieve short cascade prediction, K will be set to a smaller integer, and the remaining links are used as the query set.

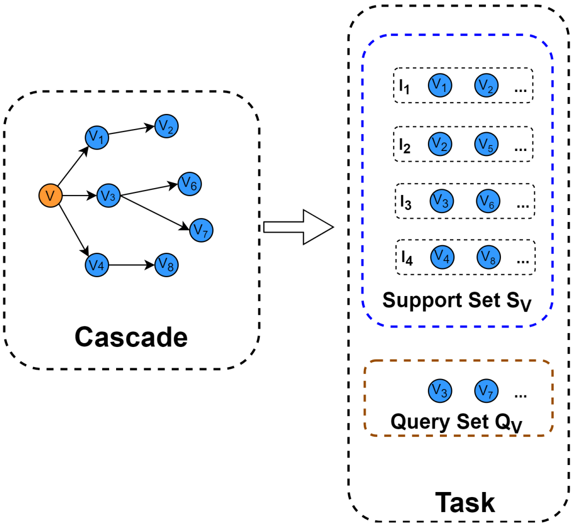

Figure 3. Task formulation.

3.2.2. CaFormer

As shown in Figure 4, the CaFormer model f_θ is composed of two parts, encoder E_ψ and classifier C_ψ. To facilitate the model to process each node information, the neighboring nodes, edges and spatial importance of each node are encoded and fed into the network, thus outputting a representation of each node. In the next, a pair of nodes is fed into the classifier to determine the existence of edges $e = (v_i, v_j, t_e)$.

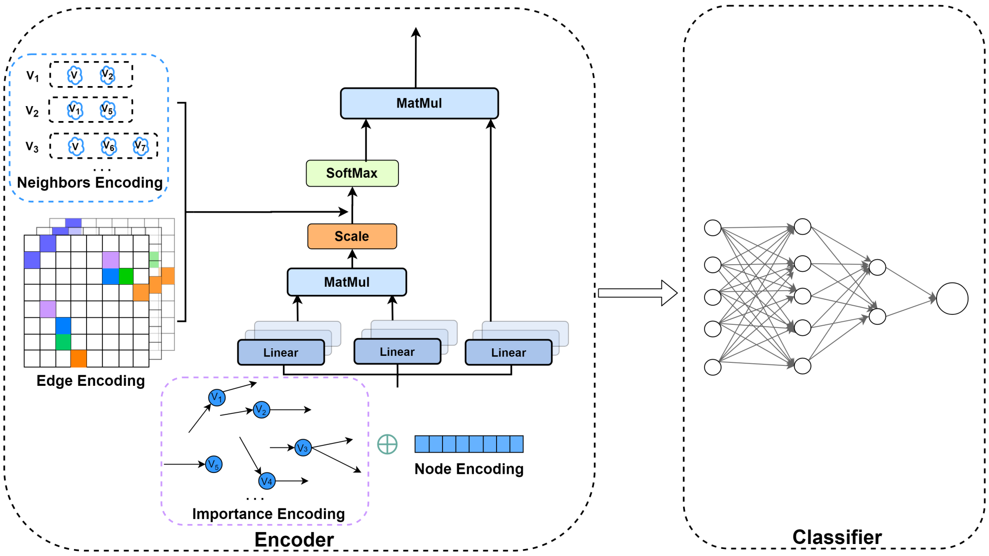

Figure 4. CaFormer model.

Encoder E_ψ: Each encoder E_ψ contains two parts, a self-attentive module and a feedforward neural network. The input of the self-attentive module is $H = \left[h_1^\top, \cdots, h_n^\top\right]^\top \in \mathbb{R}^{n\times d}$, where d is the hidden dimension and $h_i \in \mathbb{R}^{1\times d}$ is the hidden representation at position i. H is transformed into three matrices Q, K and V, and set A is the matrix capturing the similarity between Q and K, i.e.,

$$A = \frac{QK^\top}{\sqrt{d_K}} \tag{2}$$

and

$$\mathrm{Attn}(H) = \mathrm{softmax}(A)V \tag{3}$$

The function $\phi(v_i, v_j) : V \times V \to \mathbb{R}$ denotes the spatial relationship between node v_i and its neighbor node v_j in the cascade graph. If there is a connection between them, let $\phi(v_i, v_j)$ be the shortest path between node v_j and its neighbor node v_j. If there is no connection, the value is assigned to -1. Random Fourier is applied to encode the time, and a learnable scalar is set for each output value as a bias of the self-attention force, i.e.,

$$A_{ij} = \frac{\left(\left(h_i \| \Phi(t_e - t_{i,j})\right) W_Q\right)\left(\left(h_j \| \Phi(t_e - t_{i,j})\right) W_K\right)^T}{\sqrt{d}} + b_{\phi(v_i, v_j)} \tag{4}$$

where $b_{\phi(v_i, v_j)}$ is the learnable scalar, t_e is the timestamp of the interaction between v_i and its neighbor node v_j, $\|$ is the concatenation operator, and W_Q, W_K are the mapping matrices.

As important features in the cascade graph, the edges should also be considered. For each node pair, the shortest path $\mathrm{SP}_{ij} = (e_1, e_2, \ldots, e_N)$ from v_i to v_j is found, and the average of the edge features and the learnable dot product in the path is calculated. The edge encoding c_{ij} is merged into the self-attentive module, i.e.,

$$A_{ij} = \frac{\left(\left(h_i \| \Phi(t_e - t_{i,j})\right) w_Q\right)\left(\left(h_j \| \Phi(t_e - t_{i,j})\right) w_K\right)^T}{\sqrt{d}} + b_{\phi(v_i, v_j)} + c_{ij} \tag{5}$$

and

$$c_{ij} = \frac{1}{N} \sum_{n=1}^{N} x_{e_n} \left(w_n^E\right)^T \tag{6}$$

where x_{e_n} is the feature of the nth edge e_n of the path SP_{ij}, w_n^E is the nth weight embedding.

Cascade graph is a kind of directed graph. In this paper, the degree of entry and exit of each node are extracted as the importance code added to the node features, i.e.,

$$h_i^{(0)} = x_i + z^-_{\deg^-(v_i)} + z^+_{\deg^+(v_i)} \tag{7}$$

where z^- denotes the incoming degree of node v_i, and z^+ denotes the outgoing degree of node v_i.

Based on Transformer, the application of layer normalization (LN) before the feedforward neural network (FFN) is described specifically as

$$h'^{(l)} = \mathrm{MHA}\left(\mathrm{LN}\left(h^{(l-1)}\right)\right) + h^{(l-1)} \tag{8}$$

and

$$h^{(l)} = \mathrm{FFN}\left(\mathrm{LN}\left(h'^{(l)}\right)\right) + h'^{(l)} \tag{9}$$

where $LN(\cdot)$ denotes layer normalization, and $MHA(\cdot)$ denotes multi-head attention.

Classifier C_ψ: After a series of processing by the encoder, the representations of nodes v_i and v_j at timestamp t, $h_i^{(l)}$ and $h_j^{(l)}$ can be obtained, and the probability of connection of nodes v_i and v_j is predicted as

$$\begin{aligned} p_e &= c_\omega\left(h_i^{(l)}, h_j^{(l)}\right) \\ &= \sigma_2\left(\mathrm{MLP}\left(h_i^{(l)} \| h_j^{(l)}\right)\right) \end{aligned} \tag{10}$$

where σ_2 is the sigmoid function, $\|$ is the concatenation operator, and $MLP(\cdot)$ is the multilayer perceptron. Minimize the loss of each edge e to train the model, i.e.,

$$\mathcal{L}_e = -y_e \log p_e - (1 - y_e) \log(1 - p_e) \tag{11}$$

If e belongs to the set of edges E, $y_e = 1$; otherwise $y_e = 0$.

3.2.3. Meta-Learner

In real life, the forwarding cascade of social platforms is often short. This paper combines meta-learning to learn parameters autonomously to meet the demand for accurate prediction of short cascades. The meta-learner consists of three parts, encoding adaptation, classification adaptive and optimization.

The encoding adaptation step is to fine-tune the global parameters Ψ of the encoder for each time interval. The model is trained with the support set $\mathcal{S}_v$. The global parameters Ψ are updated adaptively according to the loss of the set $\mathcal{S}_v^k$ in each support set $\mathcal{S}_v$, and the node preferences for the kth time interval are obtained. For each edge $e = \left(v_i, v_j, t_e\right) \in \mathcal{S}_v^k$, the node v_i is represented as

$$\mathbf{h}_i^L\left(t^k, t_e\right) = E_\psi\left(v_i, \mathcal{N}_{v_i}\left(t^k\right), t_e\right) \tag{12}$$

where t^k denotes the timestamp at the beginning of kth time interval. The loss is calculated as

$$\mathcal{L}\left(\psi, \omega, S_v^k\right) = -\sum_{e \in \left\{S_v^k \cap E\right\}} \log(p_e) - \sum_{e \in \left\{S_v^k \setminus E\right\}} \log(1 - p_e) \tag{13}$$

$$p_e = h_\omega\left(\mathbf{h}_i^L\left(t^k, t_e\right), \mathbf{h}_j^L\left(t^k, t_e\right)\right) \tag{14}$$

where p_e denotes the possibility of connectivity between nodes v_i and v_j.

After that, the global parameter Ψ is adjusted according to the gradient descent to obtain the dedicated parameters for the kth time interval node v_i, i.e.,

$$\psi_v^k = \psi - \beta \frac{\partial \mathcal{L}\left(\psi, \omega, S_v^k\right)}{\partial \psi} \tag{15}$$

where β is the learning rate.

In the classification adaptive step, the global parameter ω of the classifier is fine-tuned according to the adaptive parameter ψ_v^k, which is updated by gradient descent to fit each set S_v^k, i.e.,

$$\omega_v = \omega + \mathbf{h}_v^0 \cdot W_\omega \tag{16}$$

$$\omega_v^k = \omega_v - \eta \frac{\partial \mathcal{L}\left(\psi_v^k, \omega_v, S_v^k\right)}{\partial \omega_v} \tag{17}$$

where W_ω is the projection matrix and η is the learning rate.

In the optimization step, the adaptive parameters of each node are first fused such that

$$\psi_v^* = \sum_{k=1}^{n-1} a_v^k \psi_v^k, \omega_v^* = \sum_{k=1}^{n-1} a_v^k \omega_v^k \tag{18}$$

and

$$a_v^k = \text{softmax}\left(-\mathcal{L}\left(\psi_v^k, \omega_v^k, S_v\right)\right) \tag{19}$$

where a_v^k is the weight of θ_v^k derived over the entire support set, which accelerates the convergence of the meta-learning training phase. θ_v^k is the mapping of the global parameter θ for the node over the kth time interval, i.e.,

$$\theta_v^k = \left\{\psi_v^k, \omega_v^k\right\} \tag{20}$$

Finally, the loss in the query set Q_v is minimized back propagation to update the global parameters θ and the mapping matrix W_ω:

$$\theta \leftarrow \theta - \gamma \nabla_\theta \sum_{v \in V} \mathcal{L}(\psi_v^*, \omega_v^*, Q_v) \tag{21}$$

$$W_\omega \leftarrow W_\omega - \gamma \nabla_{W_\omega} \sum_{v \in V} \mathcal{L}(\psi_v^*, \omega_v^*, Q_v) \tag{22}$$

where γ is the learning rate.

4. Experiments

4.1. Datasets

In order to evaluate the MetaCaFormer, this paper conducted experiments on two datasets, the Reddit post dataset and the DBLP paper citation dataset. The data statistics are shown in Table 3.

In the social platform reposting scenario, this paper uses the Reddit dataset from JODIE [2]. The Reddit dataset captures one month of user posts in subreddits, retaining the 10,000 most active users and the 1000 most active subreddits. Meanwhile, 672,448 interactions were generated, where nodes denote users and posts, and connections denote timestamps.

In the paper citation scenario, this paper uses the DBLP dataset [59]. DBLP is a co-author network that is considered to generate a new link whenever a citation occurs. In this dataset, each node represents the author and its article, and the link represents the newly generated cited article in the corresponding year.

To facilitate the experiments, this paper uses a data treatment similar to that of MetaDyGNN [20], i.e., dividing 60%, 20%, and 20% of each cascade into a training set, a validation set and a test set, respectively, in chronological order. The training set includes existing nodes and associated connected edges. The validation set includes the nodes and their associated edges that appear in the validation interval for the first time. The test set contains nodes and associated edges that appear only after the validation interval.

Table 3. Datasets information statistics.

Dataset	Reddit	DBLP
Node	10,984	28,085
Dynamic edge	672,448	286,894
Timestamp	continuous	27 snapshots

4.2. Baseline Methods

Three methods are introduced in Section 2, static graph networks, dynamic graph networks, and graph networks with small sample learning. The authors selected a few representative methods in each category as the baseline. MetaCaFormer is compared with the following baselines. GraphSAGE [29] proposes to convert the graph into a snapshot and use the feature information of neighborhood nodes to generate representations for unknown nodes efficiently. GAT [25] combines an attention mechanism with a graph convolutional network to compute the importance of its neighboring nodes when each node updates its state using the attention mechanism, assigning different weights to each neighboring node and later aggregating the neighboring nodes to obtain the node representation. EvolveGCN [13] introduces recurrent neural networks (RNNs) to update the node representation of GCNs by operating RNNs in discrete snapshots, sharing trainable parameters among all nodes. TGAT [19] combines GraphSAGE and GAT to encode continuous time and incorporates attention to combine the neighborhood node information with temporal information. The final node representation is obtained. Meta-GNN [15] was the first approach to propose the application of MAML [54] to graph structure processing. TGAT+MAML [19,54] combines TGAT with meta-learning in a simple way and divides the task according to meta-learning, using meta-learning to learn parameters autonomously while learning node properties. MetaDyGNN [20] proposed a dynamic graph network framework in a meta-learning environment to model the prediction of links to new nodes on the graph. Changing the previous problem of the simple patchwork using a hierarchical meta-learner also gives the model a more comprehensive grasp of global information.

4.3. Experimental Settings

Assuming that the first $K = 2, 4, 6, 8$ links in the validation and test sets are known, the predictions are converted into a classification issue to predict the future connectivity of nodes by outputting a positive or negative weight for each node. In order to fully reflect the effectiveness of MetaCaFormer, a detailed comparison is made of three different evaluation metrics. ACC, as the most common evaluation metric in dichotomous classification problems, can reflect the percentage of samples correctly predicted by the classifier to the total number of samples and can be used as an indicator to evaluate the good or bad performance of the classifier. However, when faced with unbalanced data, one must focus on more than just the ACC score. AUC is calculated by ranking the relative prediction probabilities of all instances, so the influence of unbalanced classes can be eliminated, and the classifier can be better evaluated. Macro-F1 is a macro-level evaluation metric, and the F1 of each class is calculated separately and averaged as macro-F1, which is more suitable for the cascade prediction scenario in this paper. Therefore, under careful consideration, we select ACC, AUC and Macro-F1 as the evaluation metrics in this paper to evaluate the MetaCaFormer model and compare the baseline methods.

For the baseline methods, all the baseline methods compared in this paper will output a positive or negative weight, and the prediction is treated as a kind of binary classification problem or ranking problem. Other parameters are consistent with those used in related papers. We use the same evaluation procedure for all baseline models and record the best results for comparison.

In this paper, the Adam optimizer is used for training, epochs are set to a maximum of 30, embedding dimension is 64, the batch size is 64, and encoder is two layers in CaFormer. In meta-learning, the time interval is 3, the dropout is 0.5, the meta-learning rate is 0.001, the coding adaptive learning rate is 0.0002, and the classification adaptive learning rate is 0.025.

Our experiments were run on a Windows Server 2019 server with 60G RAM containing a single GPU (NVIDIA A100-SXM4) and CPU (Intel Xeon Processor (Icelake) 2.59 GHz).

5. Results and Discussion

5.1. Comparison Results

This section evaluates the MetaCaFormer in the Reddit post dataset and the DBLP paper citation dataset and compares it to the baseline model. When $K = 8$, the results are shown in Table 4. In terms of AUC, MetaCaFormer improves about 1% over the best baseline approach in the Reddit post dataset and about 2% in the DBLP paper citation dataset. It can be seen that MetaCaFormer can still give stable and accurate prediction results in the observation cascades.

The experimental results show that when meta-learning is simply superimposed with graph neural networks, such as Meta-GNN and TGAT+MAML, the prediction effect for short cascades is not obvious. Even in the Reddit dataset, the superimposed meta-learning method is not as good as the method with only graph networks. Meanwhile, the comparison with MetaDyGNN shows the performance of our proposed CaFormer for processing temporal information and demonstrates the key role of temporal information in cascade processing.

To further demonstrate the effectiveness of MetaCaFormer in short cascade prediction, a complete comparison was conducted in this paper at three states of $K = 2, 4, 6$. For more obvious observations, only the MetaDyGNN with the best results was retained as our baseline method. The results are shown in Figure 5, and the experiments show that MetaCaFormer gives optimal results in all states.

Table 4. Performance comparison results.

Dataset	**Reddit**			**DBLP**		
Model/Result	ACC	AUC	Maco-F1	ACC	AUC	Maco-F1
GraphSAGE [29]	88.92%	93.12%	87.98%	72.15%	76.65%	71.32%
GAT [25]	88.76%	92.96%	88.34%	73.38%	76.94%	72.31%
EvolveGCN [13]	59.21%	61.64%	57.02%	57.88%	63.15%	56.53%
TGAT [19]	93.15%	94.43%	92.96%	77.21%	81.02%	76.45%
Meta-GNN [15]	85.97%	91.06%	85.21%	74.98%	79.85%	74.52%
TGAT+MAML [19,54]	87.85%	91.56%	87.42%	73.53%	77.46%	72.62%
MetaDyGNN [20]	95.97%	97.46%	95.68%	83.02%	87.57%	82.04%
MetaCaFormer	**97.95%**	**98.21%**	**96.88%**	**85.26%**	**89.92%**	**84.09%**

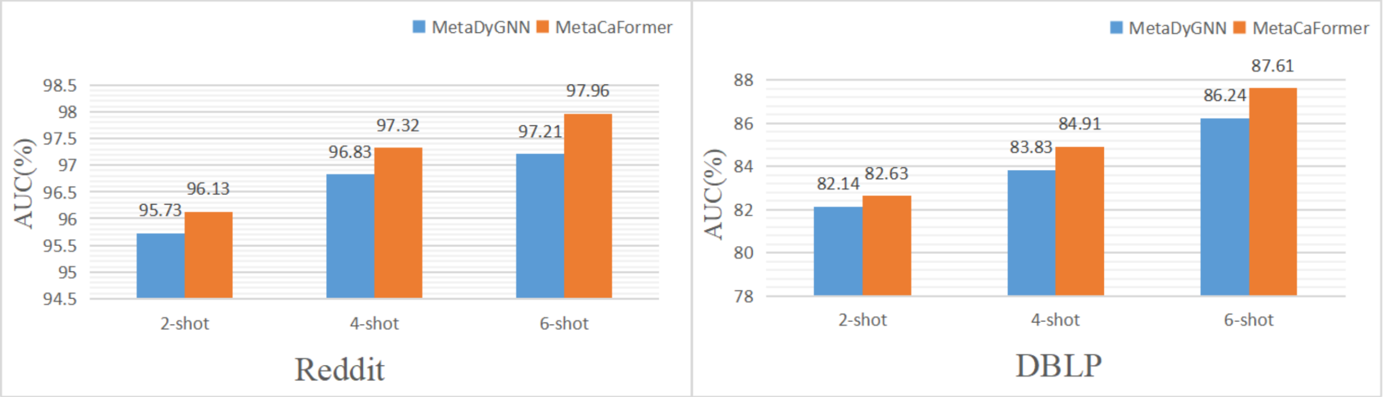

Figure 5. Performance comparison results.

5.2. Ablation

For dynamic cascade networks, temporal information plays a decisive role in the prediction effect. However, in the past baseline methods, only graph neural networks are often applied to process the cascade graph structure. Although graph neural networks have apparent effects on the processing of graph structures, they do not capture the temporal information well, which is a drawback for dynamic cascade networks. Therefore, we propose to apply the CaFormer model based on the Transformer structure in dynamic cascade networks. It can effectively avoid the problem of time information loss and make the structural information of the cascade graph better combined with time information.To verify the validity of the CaFormer, the meta-learning part of MetaCaFormer was removed, and only the simple CaFormer was kept to create the variant experiment MetaCaFormer-noMAML. As shown in Figure 6, MetaCaFormer-noMAML is better than TGAT, proving that CaFormer has a powerful processing capability for dynamic cascade networks.

We construct a variant of CaFormer+ MAML by piecing together CaFormer and meta-learning, as shown in Figure 7. The experiment shows that CaFormer+ MAML is better than the simple CaFormer for short cascade processing but not as good as MetaCaFormer. It demonstrates the effectiveness of an adaptive framework that incorporates meta-learning.

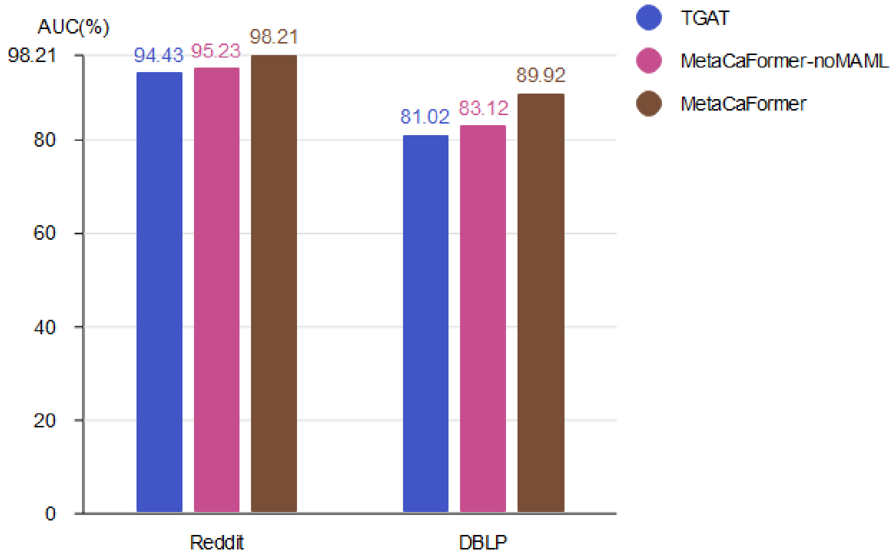

Figure 6. performance of variants.

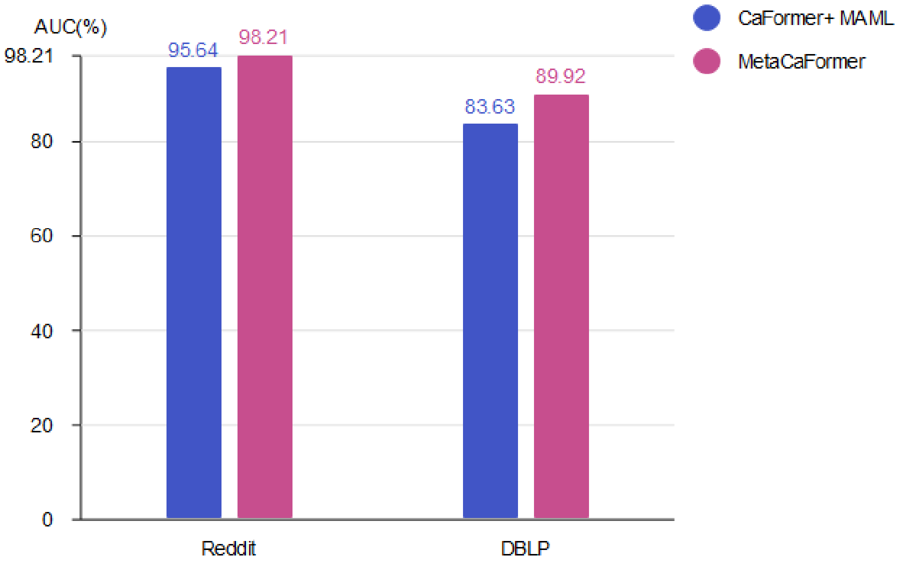

Figure 7. performance of variants.

6. Conclusions

This paper proposes a novel MetaCaFormer based on meta-learning and Transformer for dynamic short cascade prediction. Our proposed CaFormer model inherits the powerful processing capability of Transformer for temporal information while considering the neighboring nodes, edges and spatial importance of nodes. It solves the problem of the limited processing capability of traditional graph neural networks for temporal information and effectively integrates temporal and structural information. At the same time, it also incorporates meta-learning, which enables it to adapt to short cascade data quickly and improves the prediction ability for short cascades. MetaCaFormer gives better prediction results than other baseline methods in both Reddit and DBLP datasets. Therefore, this study can be a heuristic method for researchers to design short cascade prediction models and provide a new solution to the short cascade prediction problem.

MetaCaFormer is proposed to satisfy the need to accurately capture and predict only a few forwarding behaviors in dynamic cascade networks. In future work, dynamic short cascade prediction can be extended to epidemic transmission and recommender systems, where accurate prediction is meaningful for individuals and society to study.

However, MetaCaFormer also has shortcomings, and its focus on each node will also limit its prediction ability for macroscopic scenarios. Future research will also focus on how to predict macroscopic scenarios better.

Author Contributions: Conceptualization, T.M.; Methodology, T.M. and G.L.; Software, G.L.; Validation, T.M. and M.Z.; Formal analysis, M.L.; Investigation, G.L.; Resources, G.L.; Data curation, D.H; Writing—original draft preparation, T.M.; Writing—review and editing, M.Z.; Visualization, G.L.; Supervision, M.L. and D.H.; Project administration, G.L.and T.M.; Funding acquisition, G.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Taishan Scholars Program (NO.tsqn202103097), and the Key R & D Program of Shandong Province (Soft Science Project) (2022RZB02012).

Data Availability Statement: Datasets can be accessed upon request to the corresponding author.

Acknowledgments: The authors would like to thank all the anonymous reviewers for their insightful comments and constructive suggestions.

Conflicts of Interest: The authors declare that there are no conflict of interest regarding the publication of this paper.

References

1. Liu, B.; Yang, D.; Shi, Y.; Wang, Y. Improving Information Cascade Modeling by Social Topology and Dual Role User Dependency. In Proceedings of the Database Systems for Advanced Applications: 27th International Conference, DASFAA 2022, Virtual Event, 11–14 April 2022; Proceedings, Part I; Springer: Berlin/Heidelberg, Germany, 2022; pp. 425–440.
2. Kumar, S.; Zhang, X.; Leskovec, J. Predicting dynamic embedding trajectory in temporal interaction networks. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 1269–1278.
3. Zhou, F.; Xu, X.; Trajcevski, G.; Zhang, K. A survey of information cascade analysis: Models, predictions, and recent advances. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–36. [CrossRef]
4. Shang, Y.; Zhou, B.; Wang, Y.; Li, A.; Chen, K.; Song, Y.; Lin, C. Popularity prediction of online contents via cascade graph and temporal information. *Axioms* **2021**, *10*, 159. [CrossRef]
5. Chen, L.; Wang, L.; Zeng, C.; Liu, H.; Chen, J. DHGEEP: A Dynamic Heterogeneous Graph-Embedding Method for Evolutionary Prediction. *Mathematics* **2022**, *10*, 4193. [CrossRef]
6. Wang, Y.; Wang, X.; Ran, Y.; Michalski, R.; Jia, T. CasSeqGCN: Combining network structure and temporal sequence to predict information cascades. *Expert Syst. Appl.* **2022**, *206*, 117693. [CrossRef]
7. Wu, Q.; Gao, Y.; Gao, X.; Weng, P.; Chen, G. Dual sequential prediction models linking sequential recommendation and information dissemination. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 447–457.
8. Robles, J.F.; Chica, M.; Cordon, O. Evolutionary multiobjective optimization to target social network influentials in viral marketing. *Expert Syst. Appl.* **2020**, *147*, 113183. [CrossRef]
9. Zhao, L.; Chen, J.; Chen, F.; Jin, F.; Wang, W.; Lu, C.T.; Ramakrishnan, N. Online flu epidemiological deep modeling on disease contact network. *GeoInformatica* **2020**, *24*, 443–475. [CrossRef]
10. Kumar, R.; Kumar, P.; Tripathi, R.; Gupta, G.P.; Kumar, N.; Hassan, M.M. A privacy-preserving-based secure framework using blockchain-enabled deep-learning in cooperative intelligent transport system. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 16492–16503. [CrossRef]
11. Kumar, R.; Kumar, P.; Aljuhani, A.; Islam, A.; Jolfaei, A.; Garg, S. Deep Learning and Smart Contract-Assisted Secure Data Sharing for IoT-Based Intelligent Agriculture. *IEEE Intell. Syst.* **2022**, 1–8. [CrossRef]
12. Kumar, R.; Kumar, P.; Aloqaily, M.; Aljuhani, A. Deep Learning-based Blockchain for Secure Zero Touch Networks. *IEEE Commun. Mag.* **2022**, 1–7. [CrossRef]
13. Pareja, A.; Domeniconi, G.; Chen, J.; Ma, T.; Suzumura, T.; Kanezashi, H.; Kaler, T.; Schardl, T.; Leiserson, C. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 5363–5370.
14. Vilalta, R.; Drissi, Y. A perspective view and survey of meta-learning. *Artif. Intell. Rev.* **2002**, *18*, 77–95. [CrossRef]
15. Zhou, F.; Cao, C.; Zhang, K.; Trajcevski, G.; Zhong, T.; Geng, J. Meta-gnn: On few-shot node classification in graph meta-learning. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 2357–2360.
16. Chauhan, J.; Nathani, D.; Kaul, M. Few-shot learning on graphs via super-classes based on graph spectral measures. *arXiv* **2020**, arXiv:2002.12815.
17. Huang, K.; Zitnik, M. Graph meta learning via local subgraphs. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 5862–5874.

18. Yao, H.; Zhang, C.; Wei, Y.; Jiang, M.; Wang, S.; Huang, J.; Chawla, N.; Li, Z. Graph few-shot learning via knowledge transfer. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 6656–6663.
19. Xu, D.; Ruan, C.; Korpeoglu, E.; Kumar, S.; Achan, K. Inductive representation learning on temporal graphs. *arXiv* **2020**, arXiv:2002.07962.
20. Yang, C.; Wang, C.; Lu, Y.; Gong, X.; Shi, C.; Wang, W.; Zhang, X. Few-shot Link Prediction in Dynamic Networks. In Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, Virtual, 21–25 February 2022; pp. 1245–1255.
21. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. In Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 4–9 December 2017.
22. Ying, C.; Cai, T.; Luo, S.; Zheng, S.; Ke, G.; He, D.; Shen, Y.; Liu, T.Y. Do transformers really perform bad for graph representation? *arXiv* **2021**, arXiv:2106.05234.
23. Bruna, J.; Zaremba, W.; Szlam, A.; LeCun, Y. Spectral networks and locally connected networks on graphs. *arXiv* **2013**, arXiv:1312.6203.
24. Huang, X.; Song, Q.; Li, Y.; Hu, X. Graph recurrent networks with attributed random walks. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 732–740.
25. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.
26. Rong, Y.; Bian, Y.; Xu, T.; Xie, W.; Wei, Y.; Huang, W.; Huang, J. Self-supervised graph transformer on large-scale molecular data. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 12559–12571.
27. Dwivedi, V.P.; Bresson, X. A generalization of transformer networks to graphs. *arXiv* **2020**, arXiv:2012.09699.
28. Trivedi, R.; Farajtabar, M.; Biswal, P.; Zha, H. Dyrep: Learning representations over dynamic graphs. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
29. Hamilton, W.; Ying, Z.; Leskovec, J. Inductive representation learning on large graphs. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1024–1034.
30. Rossi, E.; Chamberlain, B.; Frasca, F.; Eynard, D.; Monti, F.; Bronstein, M. Temporal graph networks for deep learning on dynamic graphs. *arXiv* **2020**, arXiv:2006.10637.
31. Bose, A.J.; Jain, A.; Molino, P.; Hamilton, W.L. Meta-graph: Few shot link prediction via meta learning. *arXiv* **2019**, arXiv:1912.09867.
32. Suo, Q.; Chou, J.; Zhong, W.; Zhang, A. Tadanet: Task-adaptive network for graph-enriched meta-learning. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual, 6–10 July 2020; pp. 1789–1799.
33. Lan, L.; Wang, P.; Du, X.; Song, K.; Tao, J.; Guan, X. Node classification on graphs with few-shot novel labels via meta transformed network embedding. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 16520–16531.
34. Guo, Z.; Zhang, C.; Yu, W.; Herr, J.; Wiest, O.; Jiang, M.; Chawla, N.V. Few-shot graph learning for molecular property prediction. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; pp. 2559–2567.
35. Kumar, P.; Kumar, R.; Gupta, G.P.; Tripathi, R.; Jolfaei, A.; Islam, A.N. A blockchain-orchestrated deep learning approach for secure data transmission in IoT-enabled healthcare system. *J. Parallel Distrib. Comput.* **2023**, *172*, 69–83. [CrossRef]
36. Gori, M.; Monfardini, G.; Scarselli, F. A new model for learning in graph domains. In Proceedings of the 2005 IEEE International Joint Conference on Neural Networks, Montreal, QC, Canada, 31 July–4 August 2005; Volume 2, pp. 729–734.
37. Zhou, L.; Yang, Y.; Ren, X.; Wu, F.; Zhuang, Y. Dynamic network embedding by modeling triadic closure process. In Proceedings of the AAAI conference on artificial intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
38. Zuo, Y.; Liu, G.; Lin, H.; Guo, J.; Hu, X.; Wu, J. Embedding temporal network via neighborhood formation. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 2857–2866.
39. Wang, Y.; Chang, Y.Y.; Liu, Y.; Leskovec, J.; Li, P. Inductive representation learning in temporal networks via causal anonymous walks. *arXiv* **2021**, arXiv:2101.05974.
40. Goyal, P.; Chhetri, S.R.; Canedo, A. dyngraph2vec: Capturing network dynamics using dynamic graph representation learning. *Knowl. Based Syst.* **2020**, *187*, 104816. [CrossRef]
41. Manessi, F.; Rozza, A.; Manzo, M. Dynamic graph convolutional networks. *Pattern Recognit.* **2020**, *97*, 107000. [CrossRef]
42. Li, Z.; Kumar, M.; Headden, W.; Yin, B.; Wei, Y.; Zhang, Y.; Yang, Q. Learn to cross-lingual transfer with meta graph learning across heterogeneous languages. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 2290–2301.
43. Chapelle, O.; Scholkopf, B.; Zien, A. Semi-supervised learning (chapelle, o. et al., eds.; 2006) [book reviews]. *IEEE Trans. Neural Netw.* **2009**, *20*, 542. [CrossRef]
44. Li, Z.; Li, X.; Wei, Y.; Bing, L.; Zhang, Y.; Yang, Q. Transferable end-to-end aspect-based sentiment analysis with selective adversarial learning. *arXiv* **2019**, arXiv:1910.14192.
45. Li, Z.; Wei, Y.; Zhang, Y.; Zhang, X.; Li, X. Exploiting coarse-to-fine task transfer for aspect-level sentiment classification. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 4253–4260.

46. Wang, Y.; Yao, Q.; Kwok, J.T.; Ni, L.M. Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv. (CSUR)* **2020**, *53*, 1–34. [CrossRef]
47. Zha, J.; Li, Z.; Wei, Y.; Zhang, Y. Disentangling task relations for few-shot text classification via self-supervised hierarchical task clustering. *arXiv* **2022**, arXiv:2211.08588.
48. Liu, Y.; Lee, J.; Park, M.; Kim, S.; Yang, E.; Hwang, S.J.; Yang, Y. Learning to propagate labels: Transductive propagation network for few-shot learning. *arXiv* **2018**, arXiv:1805.10002.
49. Yao, H.; Wu, X.; Tao, Z.; Li, Y.; Ding, B.; Li, R.; Li, Z. Automated relational meta-learning. *arXiv* **2020**, arXiv:2001.00745.
50. Ding, K.; Wang, J.; Li, J.; Shu, K.; Liu, C.; Liu, H. Graph prototypical networks for few-shot learning on attributed networks. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Online, 19–23 October 2020; pp. 295–304.
51. Snell, J.; Swersky, K.; Zemel, R. Prototypical Networks for Few-shot Learning. In *Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
52. Li, Z.; Zhang, D.; Cao, T.; Wei, Y.; Song, Y.; Yin, B. Metats: Meta teacher-student network for multilingual sequence labeling with minimal supervision. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online, 7–11 November 2021; pp. 3183–3196.
53. Finn, C.; Xu, K.; Levine, S. Probabilistic Model-Agnostic Meta-Learning. In Proceedings of the 32nd Conference on Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 3–8 December 2018.
54. Finn, C.; Abbeel, P.; Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the International conference on machine learning. PMLR, Sydney, Australia, 6–11 August 2017; pp. 1126–1135.
55. Li, J.; Shao, H.; Sun, D.; Wang, R.; Yan, Y.; Li, J.; Liu, S.; Tong, H.; Abdelzaher, T. Unsupervised Belief Representation Learning in Polarized Networks with Information-Theoretic Variational Graph Auto-Encoders. *arXiv* **2021**, arXiv:2110.00210.
56. Wang, H.; Wan, R.; Wen, C.; Li, S.; Jia, Y.; Zhang, W.; Wang, X. Author name disambiguation on heterogeneous information network with adversarial representation learning. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 238–245.
57. Yan, Y.; Zhang, S.; Tong, H. Bright: A bridging algorithm for network alignment. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; pp. 3907–3917.
58. Yang, C.; Li, J.; Wang, R.; Yao, S.; Shao, H.; Liu, D.; Liu, S.; Wang, T.; Abdelzaher, T.F. Hierarchical overlapping belief estimation by structured matrix factorization. In Proceedings of the 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), The Hague, The Netherlands, 7–10 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 81–88.
59. Lu, Y.; Wang, X.; Shi, C.; Yu, P.S.; Ye, Y. Temporal network embedding with micro-and macro-dynamics. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 469–478.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.