SpaceGTN: A Time-Agnostic Graph Transformer Network for Handwritten Diagram Recognition and Segmentation

Haoxiang Hu[1, 2], Cangjun Gao[1, 2], Yaokun Li[1, 2], Xiaoming Deng[1, 2], YuKun Lai[3], Cuixia Ma[1, 2*], Yong-Jin Liu[4], Hongan Wang[1, 2]

[1]Beijing Key Laboratory of Human-Computer Interaction, Institute of Software, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences
[3]Cardiff University
[4]Tsinghua University
{huhaoxiang22, gaocangjun23, liyaokun22}@mails.ucas.ac.cn, {xiaoming, cuixia, hongan}@iscas.ac.cn,
yukun.lai@cs.cardiff.ac.uk, liuyongjin@tsinghua.edu.cn

Abstract

Online handwriting recognition is pivotal in domains like note-taking, education, healthcare, and office tasks. Existing diagram recognition algorithms mainly rely on the temporal information of strokes, resulting in a decline in recognition performance when dealing with notes that have been modified or have no temporal information. The current datasets are drawn based on templates and cannot reflect the real free-drawing situation. To address these challenges, we present SpaceGTN, a time-agnostic Graph Transformer Network, leveraging spatial integration and removing the need for temporal data. Extensive experiments on multiple datasets have demonstrated that our method consistently outperforms existing methods and achieves state-of-the-art performance. We also propose a pipeline that seamlessly connects offline and online handwritten diagrams. By integrating a stroke restoration technique with SpaceGTN, it enables intelligent editing of previously uneditable offline diagrams at the stroke level. In addition, we have also launched the first online handwritten diagram dataset, OHSD, which is collected using a free-drawing method and comes with modification annotations.

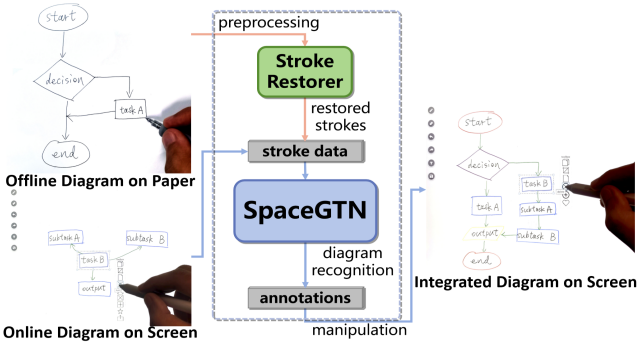


Figure 1: Overview of the proposed Diagram Manipulation System. For offline diagrams, stroke data can be obtained by a stroke restorer. Stroke and symbol annotations are generated by SpaceGTN, facilitating symbol relationship reconstruction. Stroke-level and symbol-level manipulation functions are available to users.

Introduction

In handwriting recognition and machine learning, researchers have been concentrating on extracting structural information from handwritten diagrams to enhance both interaction design and intent comprehension. Methods for recognizing handwritten diagrams can be categorized into offline and online approaches. Offline methods, such as (Herrera-Camara and Hammond 2017; Julca-Aguilar and Hirata 2018; Montellano, Garcia, and Leija 2022), disregard original stroke information, leading to difficulties in editing interactions. Online algorithms, such as (Yun et al. 2022; Bresler, Průša, and Hlaváč 2016), leverage temporal information from strokes to enhance algorithm performance. However, temporal information is not always consistently reliable. For example, temporal data is absent in strokes restored from offline handwritten documents, and user editing actions such as moving, erasing and redrawing can influence temporal features, consequently compromising the precision of recognition and segmentation.

Current handwritten diagram datasets, such as (Yun et al. 2022; Bresler et al. 2014; Awal et al. 2011; Bresler, Průša, and Hlaváč 2016), have serious limitations in applications due to the limited numbers of samples and templates. Additionally, these datasets exhibit consistent drawing styles, achieved through copying or tracing, resulting in minimal deviations such as irregular pauses or alterations. As a consequence, they diverge from the context of real-world usage. Many existing diagram datasets only contain a single category of diagrams, like flowchart datasets (Yun et al. 2022; Awal et al. 2011; Bresler, Průša, and Hlaváč 2016; Gervais et al. 2020) or automaton datasets (Bresler et al. 2014). The lack of type diversity makes it difficult to train models with the diagram classification capability.

There is limited research on the recognition and interaction systems for handwritten diagrams. Jiang et al. (2011) implemented a recognition and interaction system for handwritten Concept Maps, designing various pen-gesture-based interaction methods. Bresler, Průša, and Hlaváč (2016) proposed a recognition system for arrow-connected diagrams,

*indicates corresponding author.

but no interaction on handwritten diagrams is supported. Currently, there is no unified system capable of recognizing and interacting with a diverse set of offline or online handwritten diagrams.

To address the aforementioned issues, we propose an online handwritten diagram recognition method that does not rely on temporal information. Additionally, we propose a comprehensive diagram dataset with abundant samples and closer resemblance to real-world drawing scenarios. Building upon the aforementioned foundations, we develop a complete diagram recognition and interaction pipeline for both offline and online diagrams, as shown in Figure 1.

The contributions are summarized as follows:

- We introduce SpaceGTN, a graph transformer network that leverages exclusively the spatial stroke information. Alongside SpaceGTN, a dynamic graph building methodology is developed to augment the representation of handwritten diagram structures. Demonstrated across diverse datasets, our approach achieves state-of-the-art performance.
- We propose a handwritten diagram recognition and interaction pipeline that bridges both offline and online handwritten diagrams. The process includes diagram stroke restoration and tailored interaction designed for various types of diagrams.
- We release a large-scale dataset featuring detailed annotations, closely resembling real writing process, encompassing various types of online handwritten diagrams.

Related Work

In this section, we will review relevant research on handwritten diagram recognition algorithms, particularly focusing on the application of graph neural networks in handwritten diagram recognition. Subsequently, datasets on handwritten diagrams will be revisited. Finally, we will briefly touch upon techniques for restoring online strokes from offline handwritten documents.

Online and Offline Handwritten Diagram Recognition Algorithm. Handwritten diagram recognition algorithms can be categorized into offline and online methods. In offline methods, Herrera-Camara and Hammond (2017) extract axis-aligned scores and other stroke features, employing computer vision techniques to recognize flowcharts. Other studies employ methods based on Faster R-CNN (Montellano, Garcia, and Leija 2022), (Julca-Aguilar and Hirata 2018), or Arrow R-CNN (Schäfer, Keuper, and Stuckenschmidt 2021) for flowchart element recognition. Offline methods discard stroke information, leading to difficulties in interaction design. These methods are commonly used for standardized redrawing (Schäfer, Keuper, and Stuckenschmidt 2021) or code generation (Montellano, Garcia, and Leija 2022; Julca-Aguilar and Hirata 2018). In online methods, some employ machine learning techniques such as data mining (Blagojevic et al. 2010) or traditional classifiers such as SVM (Miyao and Maruyama 2012), yielding relatively lower accuracy. Others leverage graph neural networks (GNNs). For stroke classification, relevant research employs GNN variants like EGAT (Ye et al. 2019) and EPAT (Ye et al. 2021). Instance GNN (Yun et al. 2022) has been employed for handwritten diagram recognition and achieved state-of-the-art performance across various datasets. Graph neural networks have shown excellent performance in handwriting recognition. However, existing methods rely on online stroke temporal information during graph building, feature extraction, and model training processes. Eliminating the dependence on temporal information to enhance the robustness of recognition algorithms in practical scenarios remains a pressing challenge.

Diagram Datasets. Diagram datasets, including flowchart datasets (Yun et al. 2022; Awal et al. 2011; Bresler, Průša, and Hlaváč 2016; Gervais et al. 2020) and automaton datasets (Bresler et al. 2014), are characterized by well-defined structural forms. Most of the diagram datasets (Bresler et al. 2014; Yun et al. 2022; Awal et al. 2011; Bresler, Průša, and Hlaváč 2016) suffer from limited sample sizes, particularly for specific categories of elements, making accurate assessment of model performance challenging. Existing datasets adopt tracing or redrawing techniques for handwritten documents creation, ensuring consistent temporal patterns of stroke creation and pauses. However, real-world sketching involves interruptions of varying durations and modifications at arbitrary time points, which are not accurately captured by existing datasets. Furthermore, a single-type diagram dataset cannot adequately support the training of classification algorithms for diagrams. Therefore, the need arises for a large-scale, realistic and multi-category diagram dataset.

Stroke Restoration. Offline diagram manipulation faces challenges due to the absence of stroke information. Stroke restoration methods reconstruct stroke coordinate sequences from offline diagrams and are primarily utilized for character recognition and structural analysis. Chan (2020) has verified that mathematical formula recognition could benefit from stroke restoration. Recent studies have employed template or reference strokes to restore strokes in Chinese characters (Wang, Jiang, and Liu 2022; Li et al. 2023). Nevertheless, the acquisition of template or reference strokes becomes intricate in free-drawing diagrams rich in stroke semantics and multi-stroke structures. Stroke-level manipulation for offline diagrams remains a challenge to be addressed.

Method

In this section, we introduce our proposed SpaceGTN model. We first give the problem definition. Then, we provide details of dynamic graph building and SpaceGTN. Finally, the stroke restoration is decribed. The framework of the model is shown in Figure 2.

Problem Definition

This research aims to address the challenge of intelligently recognizing handwritten diagrams. We have L online handwritten diagrams labeled as D, where D consists of diagrams $\{G_i|i = 1, ..., L\}$. Each online handwritten diagram is defined as a graph G_i, where each stroke S_u^i is represented as a node N_u^i, and the node set N^i in G_i can be expressed as $\{(S_u^i, C_u^i, I_u^i)|u = 1, ..., n\}$, where n denotes the number of

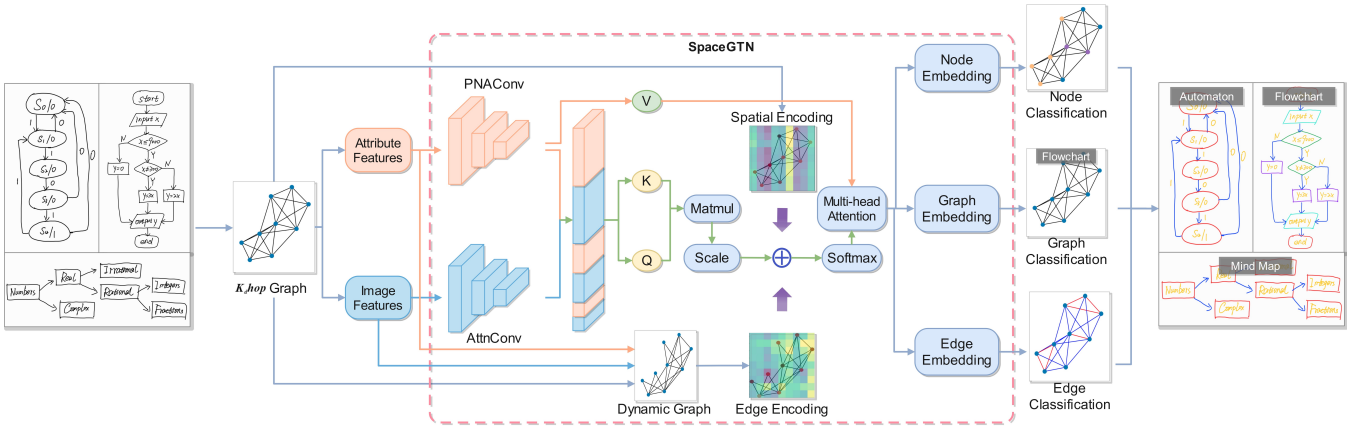


Figure 2: Overview of the processing pipeline based on SpaceGTN. The system generates the $k_d hop$ graph from the input online handwritten diagram, subsequently extracting both attribute and image features. Within the SpaceGTN model, the PNAConv processes the attribute features, while the AttnConv manages the image features. Features from each layer are then integrated to derive the key and query matrices K, Q. The K, Q matrices along with the V matrix which is sourced from the attribute feature, implement the attention mechanism. The spatial encoding from the stroke position information and the edge encoding derived from the dynamic graph are added on the attention matrix. The edge embedding is the average of the node embeddings it connects. The graph embedding is the mean of all the node embeddings within the graph. In conclusion, the system yields node, edge, and graph classification results, ultimately producing an online handwritten diagram with strokes and symbol annotations.

nodes in G_i. Each node has a class label C_u^i, and a serial number I_u^i of the symbol it belongs to. The spatial relationships between strokes are encoded as an edge set E^i. Each edge is classified as positive or negative. Positive edges represent connections between nodes within the same symbol, while negative edges represent connections between nodes from different symbols. With the result of edge classification, the symbol segmentation task can be completed. Each edge set E^i is expressed as $\{(E_v^i, C_v^i)|v = 1, ..., m\}$, where m denotes the number of edges in G_i. Therefore, the graph G_i can be represented as: $G_i = (N^i, E^i, C^i)$, where C^i is the category of graph G_i. Our nodes capture only the (x, y) coordinates of each stroke, without requiring additional information such as time, pressure, pen state, or other stroke specifics, making it easier for data capture. However, omitting such vital information greatly increases the difficulty of the research. For online diagrams, our goal is to group strokes into symbols and predict labels for both the symbols and the entire graph. For offline diagrams, we aim to convert non-editable offline strokes into editable online strokes and process them in the same manner as online diagrams.

Feature Extraction and Fusion Module

In our experiments, we have found that both the attribute features of the strokes and the deep features of stroke images have a significant impact on classification. Common stroke feature extraction techniques, such as methods combining CNN (LeCun et al. 1998) with LSTM (Yao et al. 2018) and methods that directly extract stroke attribute features, primarily rely on the temporal information of strokes. However, the absence or modification of temporal information can severely affect the LSTM and the temporal attributes of strokes. Therefore, we propose a dual-channel feature extraction method: the first channel extracts the geometric and contextual attribute features of strokes; the second channel extracts deep image features using neural networks. Through our experiments, PNAConv (Corso et al. 2020) surpasses GCN (Kipf and Welling 2017) and GAT in spatial feature aggregation of strokes. We chose PNAConv to aggregate attribute features X^s, and combine convolution with attention (Mnih et al. 2014) to construct the feature extractor F for finely extracting the image features of the strokes X^p. In order to mitigate the computational complexity introduced by the transformer architecture, we employ feature fusion during self-attention computation. During the fusion stage, attribute features and deep features are concatenated at various depths, simultaneously reducing feature dimensionality. For the calculation of V(value) matrix in the attention model, the feature fusion module is omitted, effectively addressing problems introduced by the self-attention mechanism:

$$X_l = \text{softmax}(F_l(X_l^p) \oplus PNAConv_l(X_l^s)) \quad (1)$$

where X_l is the aggregated feature at layer l.

Dynamic Graph Building

Existing methods for determining if nodes are connected based on time proximity are only suitable for fluent writing processes without modifications. However, our dataset proves that instances of intermittent drawing and modifications frequently occur during the drawing process, leading to wrong connections in the graph. In handwritten diagrams, there can be numerous strokes within the same category which are contained by different symbols. These strokes should be able to convey information among themselves. However, the method in (Yun et al. 2022) fails to effectively utilize the stroke information from symbols of the same category that are located farther apart. To address these challenges, we introduce a method for dynamic graph

building. We first construct graph connections based on spatial proximity. If one node is spatially adjacent to another one, an edge connecting them is added to the edge set. Subsequently, nodes with their degrees smaller than a threshold will be connected to their K_dhop nodes. Experiments show that K_dhop connection can effectively connect non-neighboring similar stroke nodes, but K_dhop may also connect too many different types of nodes. To solve this problem, we use the similarity scores between nodes to dynamically modify K_dhop connected edges. We dynamically update the features of the strokes, as well as the similarity score weights of the image features and attribute features of the strokes during the training process, so as to obtain the most accurate node similarity scores. We design a method for computing similarity based on stroke image features. Different from the common method (Zhang et al. 2018) of extracting complex image features of AlexNet (Krizhevsky, Sutskever, and Hinton 2012), we consider the details of simple images. Our feature extractor F adjusts the size of the convolution kernel and adds the attention mechanism to extract the image features of the current stroke N_p^i and the adjacent stroke N_q^i respectively. To quantify stroke similarity, we define an image feature similarity score S_{perc}:

$$S_{\text{perc}} = \exp(-\frac{1}{H_j W_j} \sum_{h=1}^{H_j} \sum_{w=1}^{W_j} (F_{h,w}^j(p) - F_{h,w}^j(q))^2) \quad (2)$$

where $F_{h,w}^j$ is the image features on the map of the jth layer. We further employ shape context score (Belongie, Malik, and Puzicha 2000) and curvature loss methods based on stroke coordinate points. Because of the rotation invariance of the stroke structure in the diagram, we use the Shape Context score S_{SC} to compare strokes S_p and S_q:

$$S_{sc} = \exp(-\sum_{(i,j)\in M} \sum_{k,l} |C(P_{p,i})[k,l] - C(P_{q,j})[k,l]|) \quad (3)$$

where M is the optimal match of the strokes, and C is the shape context of the strokes.

Observing that the curvatures of strokes in the same category of symbols have similarities, we propose a curvature similarity score S_c of strokes:

$$S_c = \exp(-\sum_{i=1}^{n-2} |\alpha_i - \beta_j|) \quad (4)$$

where α_i and β_j represent the vector angles between adjacent points in strokes S_p and S_q respectively. We use the following formula to calculate the total similarity score S_{total}:

$$S_{\text{total}} = w_1 \cdot S_{\text{perc}} + w_2 \cdot S_{sc} + w_3 \cdot S_{\text{c}} \quad (5)$$

where w_1, w_2, w_3 are the weights of scores.

SpaceGTN Model

Many diagram recognition techniques primarily rely on GAT(Ye et al. 2019, 2021; Yun et al. 2022). However, GAT is limited to focusing on a node's first-order neighborhood. When GAT incorporates higher-order neighborhoods, its classification performance deteriorates significantly (Zhang et al. 2019). In hand-written diagrams, strokes of the same category might lack direct connections, irrespective of whether they belong to the same symbol. This prevents nodes from effectively aggregating all pertinent features. To address this issue, we integrate the GTN network (Ying et al. 2021) into handwritten diagram recognition for the first time. This approach enables each node to collect features across the entire diagram, facilitating learning for distant nodes of the same category. Additionally, we introduce two structure encoding methods tailored for handwritten diagrams, as explained in detail below.

Edge Encoding In the recognition of handwritten diagrams, edge features describe the relationships between nodes, serving as a crucial component in graph representation. The majority of existing studies have incorporated edge features into node features. Due to inaccuracies in edge relationships and the constraints of the GAT network, edge information has not been effectively utilized to guide computations of global correlations. To address this problem, our dynamic graph building module plays a pivotal role in edge encoding. This module sorts nodes based on their dynamic similarity scores S_{total}, resulting in a more precise adjacency matrix E_{opt}:

$$E_p^{\text{opt}} = \begin{cases} 1 & \text{if } S_{\text{total}} \text{ ranks top } k \text{ for node p} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

where S_{total} is the similarity score between nodes.

However, solely relying on E_{opt} to guide the attention matrix does not capture the characteristics of connected edges. We aggregate the edge features F_E, assign them as weights to the adjacency matrix, and use this matrix as a bias term A_E in the attention matrix calculation:

$$A_E = \text{Attn}(F_E) \times E_{\text{opt}} \quad (7)$$

Spatial Encoding For handwritten diagrams, the positional information of stroke nodes holds significant contextual relevance. The effectiveness of absolute position information for strokes in handwritten diagrams is limited by substantial variations across different samples, and it inadequately emphasizes the relative interrelationships between strokes. Consequently, we employ the technique of encoding strokes through relative positional information. Specifically, given any graph G_i with a node set N^i, we define $h(N_u^i) = p(N_u^i)_h$ and $w(N_u^i) = p(N_u^i)_w$, where h represents the vertical coordinate and w represents the horizontal coordinate. This yields a coordinate vector:

$$coords(N_u^i) = \begin{bmatrix} h(N_u^i) \\ w(N_u^i) \end{bmatrix} \quad (8)$$

For any two nodes $N_{u_a}^i$ and $N_{u_b}^i$, their relative coordinates (RC) are defined as follows:

$$RC(N_{u_a}^i, N_{u_b}^i) = coords(N_{u_a}^i) - coords(N_{u_b}^i) \quad (9)$$

The relative position encoding index (R) is expressed as:

$$R(N_{u_a}^i, N_{u_b}^i) = RC(N_{u_a}^i, N_{u_b}^i)_h + RC(N_{u_a}^i, N_{u_b}^i)_w \quad (10)$$

Finally we get the attention matrix A_G of the graph G:

$$A_G = \text{softmax}\left(\frac{XW_q(XW_k)^T}{\sqrt{d_v}} + A_E W_e + R_N W_r\right) \quad (11)$$

where X is the characteristic matrix of the graph, W means learnable weights, and R_N is the relative position encoding index of the graph.

Loss Function We choose the cross-entropy function (CE) to calculate the loss for the node, edge and graph classification problems.

$$\mathrm{CE}(P, Q) = -\sum_{i=1}^{c} P(i) \log(Q(i)) \tag{12}$$

where $P(i)$ is the true probability of sample i derived from labels, $Q(i)$ is the predicted probability of sample i by the model, and c is the number of categories.

Stroke Restoration

Current stroke restoration methods mainly focus on junction detection and stroke segment merging. The primary challenge lies in potential inaccuracies during the stroke segment merging process. In this work, we improve the junction detection method to facilitate segments merging. For every pair of adjacent segments, we collect two point sequences P_1, P_2 and vectors v_1, v_2 near the junction, separately calculate their connectivity c, and then sequentially connect two strokes with the biggest connectivity. The connectivity c is calculated as:

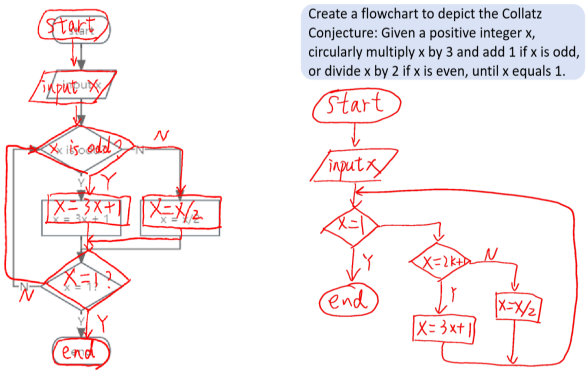
$$c = \eta \frac{|\sum(x_i - \bar{x}) \sum(y_i - \bar{y})|}{\sqrt{\sum(x_i - \bar{x})^2} \sqrt{\sum(y_i - \bar{y})^2}} + (1 - \eta) \frac{v_1 \cdot v_2}{|v_1| \cdot |v_2|} \tag{13}$$

where $(x_i, y_i) \in P_1 \cup P_2$, $\bar{x} = \frac{\sum x_i}{n}$, $\bar{y} = \frac{\sum y_i}{n}$. The first term in the formula represents the linear dependence, and the second term represents the consistency of direction.

Unlike other approaches, we split the merged strokes at their corners, which is more congruent with stroke-level manipulations. It offers a more intuitive representation for geometrical primitives which are better depicted by multiple individual lines rather than a single continuous polyline.

OHSD Dataset

In this work, we release the first large-scale online handwritten structure diagram dataset (OHSD), including 10,000 diagrams with 3 diagram types, a total of 4900 flowcharts, 3100 mind maps and 2000 automata. Figure 3 shows the difference between template drawing and free drawing. For free drawing, we provided the writers with prompt texts that can be expressed in structural diagrams and asked them to draw the corresponding diagram according to their own understanding. Compared with template drawing, free drawing better matches real drawing scenarios. Each stroke in the dataset contains point coordinates, timestamps, a category label, a symbol label and stroke modification annotations. Statistics indicate that about 40% of the symbols and 70% of the diagrams contain modified strokes, and the number of modified strokes for each free drawing diagram is 20%-30% more than that of a template drawing diagram. It can be seen that the chaos of stroke timestamp exists commonly in real drawing scenarios, which further confirms the superiority of our method that does not depend on temporal information.



(a) Template drawing. (b) Free drawing.

Figure 3: Comparison of template drawing and free drawing. Both diagrams describe the Collatz conjecture. (a) Flowchart drawn following a template. (b) Flowchart drawn based on prompt text.

Datasets	Types	Diagrams	Templates	Symbols	Strokes
FA	1	300	36	8261	14976
FC_A	1	419	28	9331	39051
FC_B	1	672	84	14880	70684
CASIA-OHFC	1	2957	1200	90376	849719
OHSD	**3**	**10000**	**3900**	**319038**	**2757801**

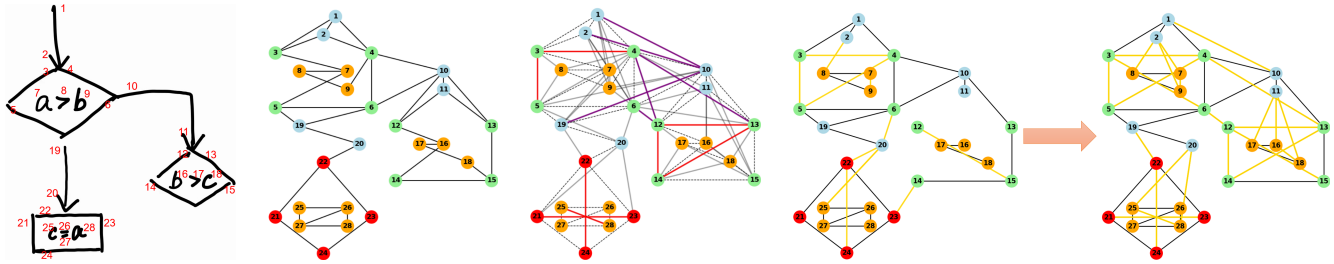
Table 1: Overview of online diagram datasets. The prompt texts in OHSD are counted in the 'Templates' column.

OHSD is the first dataset collected using a free drawing method and comes with modification annotations. Besides, OHSD contains the most free drawing themes and templates covering a wide range of application fields, a variety of drawing styles, and structures including tree, timeline, one-way and organizational structure, etc. In contrast to other datasets, OHSD has a more balanced distribution of different symbol types. In summary, OHSD manifests a higher level in terms of complexity, standardization and diversity. Table 1 shows the comparison of OHSD, FA (Awal et al. 2011), FC_A (Awal et al. 2011), FC_B (Bresler, Průša, and Hlaváč 2016) and CASIA-OHFC (Yun et al. 2022).

Experiments

Experiment Design

We employed datasets listed in Table 1 to evaluate our model. The experiments mainly encompass comparative experiments and ablation experiments. For the comparative experiments, we implement four offline methods, DETR (Carion et al. 2020), Deformable DETR (Zhu et al. 2021), Mask R-CNN (He et al. 2017), Faster R-CNN(Ren et al. 2015) along with three online methods, Inst-GNN(Yun et al. 2022), ORSAD (Bresler, Průša, and Hlaváč 2016) and EGAT (Ye et al. 2019). We tested these methods across various datasets and compared them with SpaceGTN. In the context of ablation experiments, we systematically examine the impact of



(a) Handwritten diagram example. (b) Graph built based on spatial distance. (c) Graph built based on K_dhop. (d) Dynamic graph building.

Figure 4: Visualization of graph building approaches for handwritten diagram (a). (b) builds the graph solely based on spatial distance. (c) introduces K_dhop edges, enhancing connectivity within symbols (highlighted in red), and reinforcing communication among symbols of the same category (highlighted in purple). (d) is built by calculating the dynamic similarity score S_{total} throughout the training process. The connections in the graph are dynamically adjusted. The graph significantly strengthens connections between strokes that are spatially distant and similar in shape (highlighted in gold).

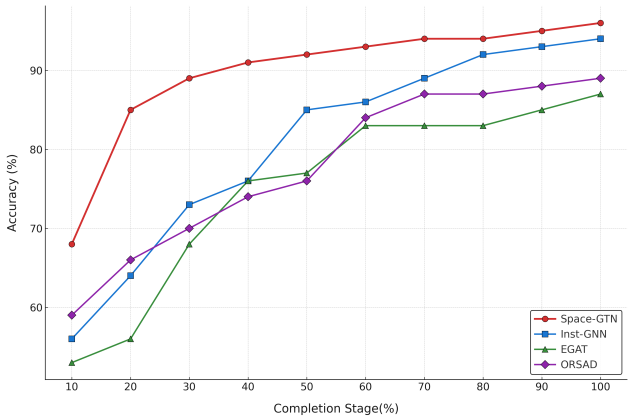


Figure 5: Diagram recognition accuracy of different degrees of completion using different methods.

four modules, K_dhop graph building, dynamic graph building, edge encoding, and spatial encoding on model accuracy. Furthermore, we also test the graph recognition accuracy at varying completion stages of diagrams. Experiments are carried out on NVIDIA A40 GPUs.

Evaluation Metrics

We employ Stroke Classification Accuracy, Stroke Classification Precision and Symbol Recognition F1-Score to measure the performance of the model.
(1) Stroke Classification Accuracy (SCA).

$$SCA = \frac{\sum_{i=1}^{N} C_i}{\sum_{i=1}^{N} T_i} \tag{14}$$

where C_i is the number of strokes correctly classified in category i, T_i is the total number of strokes in category i and N is the number of categories.
(2) Stroke Classification Precision (SCP).

$$SCP = \sum_{i=1}^{N} \frac{T_i}{\sum_{i=1}^{N} T_i} \times P_i \tag{15}$$

where P_i is the precision of class i.
(3) Symbol Recognition F1-Score (SRF).
F1-score is the harmonic mean of precision (P) and recall (R), given by:

$$F1 = \frac{2 \times P \times R}{P + R} \tag{16}$$

where P is the fraction of correctly identified symbols out of all predicted symbols, and R is the fraction of correctly identified symbols out of all actual symbols. In the task of diagram segmentation and recognition, a symbol is considered to be correctly identified only if both its strokes and symbol category are correct.

Results and Analysis

Temporal Information Dependence of Methods To study the dependency of online methods on temporal information, we replicate projects whose codes are not publicly available, and test the performance of methods without temporal information. Table 3 shows the stroke classification results of these algorithms. The decreased accuracy in NUT condition confirms other approaches' high dependence on temporal information. However, our approach utilizes spatial information and remarkably surpasses the performance of the state-of-the-art methodologies in UT condition.

Comparison with State-of-the-Art We compare our model with state-of-the-art methods on multiple datasets. Notably, temporal information is not utilized by any of the methods. As illustrated in Table 2, the absence of temporal information leads to low accuracy for the aforementioned online methods across various datasets. Moreover, our algorithm relies on the ultra-high accuracy of edge and node classification. We discard the complicated post-processing required in the segmentation process by other methods, thus simplifying the process and making it more suitable for real-time interaction. The strokes restored from offline diagrams lack temporal information, and modifications or interrupted drawing can cause confusion in the time data. In such cases, other online methods fail to identify them accurately, while our method remains unaffected. Extensive experiments on

Method	FC_A			FC_B			CASIA-OHFC			OHSD		
	SCA	SCP	SRF	SCA	SCP	SRF	SCA	SCP	SRF	SCA	SCP	SRF
DETR	47.89	53.75	40.83	68.87	65.03	59.52	50.76	52.94	47.83	48.02	47.48	43.53
DeDETR	55.64	56.27	51.72	64.07	68.10	59.68	52.17	59.75	49.10	56.72	62.30	52.91
FasterRCNN	71.66	68.89	52.45	74.61	79.68	52.19	71.56	74.50	58.44	67.75	69.04	53.62
MaskRCNN	77.57	74.62	61.88	78.70	72.56	63.72	72.26	70.74	56.52	76.30	71.95	53.21
ORSAD	96.30	93.04	84.20	97.92	95.83	92.16	87.90	83.23	68.26	85.36	82.34	64.63
EGAT	97.28	96.46	-	98.32	98.25	-	89.71	90.46	-	90.12	89.51	-
InstGNN	98.44	98.07	95.04	98.56	98.71	97.82	92.86	93.18	74.29	95.52	94.86	88.32
Ours	**99.02**	**98.67**	**95.32**	**99.12**	**99.32**	**98.05**	**98.13**	**97.93**	**80.47**	**99.78**	**99.32**	**95.54**

Table 2: Quantitative comparisons across multiple datasets with diverse offline and online methods (%), all conducted without incorporating temporal information.

Method	CASIA-OHFC				OHSD			
	UT		NUT		UT		NUT	
	SCA	SCP	SCA	SCP	SCA	SCP	SCA	SCP
ORSAD	91.31	91.04	87.90	83.23	86.65	84.36	85.36	82.34
EGAT	92.76	92.01	89.71	90.46	92.82	90.46	90.12	89.51
I-GNN	95.81	95.42	92.86	93.18	96.89	95.44	95.52	94.86
Ours	-	-	**98.13**	**97.93**	-	-	**99.78**	**99.32**

Table 3: Online methods are applied to datasets containing temporal information. The performance (%) of these methods under both scenarios: utilizing (UT) and not utilizing temporal information (NUT). "I-GNN" stands for Inst-GNN and "Ours" represents SpaceGTN.

multiple datasets (Table 2) show that our method consistently outperforms existing methods and achieves state-of-the-art performance.

Graph Classification Result As shown in Figure 5, diagram recognition accuracy using SpaceGTN can achieve more than 85% under 20% strokes, and about 90% under 30% strokes. The accuracy of the proposed method is higher than that of other methods at different completion degrees, especially in the low completion degree situation. This implies that we can accurately identify the graph category when the user initially draws, and promptly provide corresponding interactive functions.

Ablation Study

We conduct ablation studies using the CASIA-OHFC dataset to assess the influence of different components within our proposed method on performance.

Effect of Dynamic Graph Building We investigate the influence of various graph building blocks on performance. We conduct experiments using the K_dhop method and compute the similarity score S_{total} separately and simultaneously. The experimental results shown in Table 4 prove that method for edge expansion relying only on the K_dhop is not optimal. It introduces many irrelevant nodes and compromises the aggregation of node features. However, the combined approach of K_dhop and Dy_G is significantly better than

Exp	K_dhop	Dy_G	$Edge_{enc}$	Sp_{enc}	SCA	SCP
1	×	×	×	×	94.06	94.28
2	✓	×	×	×	94.56	94.44
3	×	✓	×	×	96.13	95.57
4	×	×	✓	×	95.90	95.83
5	×	×	×	✓	95.02	94.82
6	✓	✓	×	×	97.42	97.38
7	✓	✓	✓	×	97.81	97.63
8	✓	✓	×	✓	97.76	97.68
9	✓	✓	✓	✓	**98.13**	**97.93**

Table 4: Ablation experiments conducted to investigate the impact of modules K_dhop Graph Building (K_dhop), Dynamic Graph Building (Dy_G), Edge Encoding ($Edge_{enc}$), and Spatial Encoding (Sp_{enc}) (%).

using either module alone. The visualization of graph building is shown in Figure 4.

Effect of Structure Encoding As presented in Table 4, activating either edge encoding or spatial encoding in SpaceGTN, the performance enhancement is suboptimal. Concurrent activation of both modules markedly boosts the accuracy, which underscores the pivotal role of structural encoding in node classification.

Conclusion

The recognition of online handwritten diagrams is converted into problems of graph, node and edge classification. We introduce SpaceGTN, a graph transformer network that operates independently of temporal information. Through the integration of proposed modules, we attained an accuracy of 98.13% on the CASIA-OHFC dataset and 99.78% on the OHSD dataset in node classification task. Our approach maintains state-of-the-art performance even when benchmarked against methods that incorporate temporal information. Furthermore, we have designed an integrated prototype system that harmoniously bridges offline and online diagrams. In addition, we will publicly release the OHSD dataset, the pioneering large-scale online handwritten diagram dataset with freely drawn strokes and stroke modification annotations.

Acknowledgements

This work was supported by National Key R&D Program of China (2022ZD0117900), the National Natural Science Foundation of China under Grant 62272447 and 62332019, and Beijing Natural Science Foundation under Grant 4212029 and L222008.

References

Awal, A.-M.; Feng, G.; Mouchère, H.; and Viard-Gaudin, C. 2011. First experiments on a new online handwritten flowchart database. In *Document Recognition and Retrieval XVIII*, volume 7874, 81–90. SPIE.

Belongie, S.; Malik, J.; and Puzicha, J. 2000. Shape context: A new descriptor for shape matching and object recognition. *Advances in neural information processing systems*, 13.

Blagojevic, R.; Plimmer, B.; Grundy, J.; and Wang, Y. 2010. Building Digital Ink Recognizers Using Data Mining: Distinguishing between Text and Shapes in Hand Drawn Diagrams. In *Trends in Applied Intelligent Systems*, 358–367.

Bresler, M.; Phan, T. V.; Prusa, D.; Nakagawa, M.; and Hlavac, V. 2014. Recognition System for On-Line Sketched Diagrams. In *2014 14th International Conference on Frontiers in Handwriting Recognition*, 563–568.

Bresler, M.; Průša, D.; and Hlaváč, V. 2016. Online recognition of sketched arrow-connected diagrams. *International Journal on Document Analysis and Recognition*, 19(3): 253–267.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *European conference on computer vision*, 213–229.

Chan, C. 2020. Stroke extraction for offline handwritten mathematical expression recognition. *IEEE Access*, 8: 61565–61575.

Corso, G.; Cavalleri, L.; Beaini, D.; Liò, P.; and Velickovic, P. 2020. Principal Neighbourhood Aggregation for Graph Nets. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*.

Gervais, P.; Deselaers, T.; Aksan, E.; and Hilliges, O. 2020. The DIDI dataset: Digital Ink Diagram data. *CoRR*, abs/2002.09303.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 2961–2969.

Herrera-Camara, J.-I.; and Hammond, T. 2017. Flow2Code: From Hand-Drawn Flowcharts to Code Execution. In *Proceedings of the Symposium on Sketch-Based Interfaces and Modeling*.

Jiang, Y.; Tian, F.; Zhang, X. L.; Dai, G.; and Wang, H. 2011. Understanding, Manipulating and Searching Hand-Drawn Concept Maps. *ACM Transactions on Intelligent Systems and Technology*, 3(1): 1–21.

Julca-Aguilar, F. D.; and Hirata, N. S. T. 2018. Detection in Online Handwritten Graphics Using Faster R-CNN. In *2018 13th IAPR International Workshop on Document Analysis Systems*, 151–156.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Li, M.; Yu, Y.; Yang, Y.; Ren, G.; and Wang, J. 2023. Stroke Extraction of Chinese Character Based on Deep Structure Deformable Image Registration. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37: 1360–1367.

Miyao, H.; and Maruyama, R. 2012. On-Line Handwritten flowchart Recognition, Beautification and Editing System. In *2012 International Conference on Frontiers in Handwriting Recognition*, 83–88.

Mnih, V.; Heess, N.; Graves, A.; et al. 2014. Recurrent models of visual attention. *Advances in neural information processing systems*, 27.

Montellano, C. D. B.; Garcia, C. O. F. C.; and Leija, R. O. C. 2022. Recognition of Handwritten Flowcharts using Convolutional Neural Networks. *International Journal of Computer Applications*, 184(1): 37–41.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Schäfer, B.; Keuper, M.; and Stuckenschmidt, H. 2021. Arrow R-CNN for Handwritten Diagram Recognition. *Int. J. Doc. Anal. Recognit.*, 24(1–2): 3–17.

Wang, T.-Q.; Jiang, X.; and Liu, C.-L. 2022. Query pixel guided stroke extraction with model-based matching for offline handwritten Chinese characters. *Pattern Recognition*, 123: 108416.

Yao, H.; Tang, X.; Wei, H.; Zheng, G.; Yu, Y.; and Li, Z. J. 2018. Modeling Spatial-Temporal Dynamics for Traffic Prediction. *ArXiv*, abs/1803.01254.

Ye, J.-Y.; Zhang, Y.-M.; Yang, Q.; and Liu, C.-L. 2019. Contextual Stroke Classification in Online Handwritten Documents with Graph Attention Networks. In *2019 International Conference on Document Analysis and Recognition*, 993–998.

Ye, J.-Y.; Zhang, Y.-M.; Yang, Q.; and Liu, C.-L. 2021. Joint stroke classification and text line grouping in online handwritten documents with edge pooling attention networks. *Pattern Recognition*, 114: 107859.

Ying, C.; Cai, T.; Luo, S.; Zheng, S.; Ke, G.; He, D.; Shen, Y.; and Liu, T.-Y. 2021. Do Transformers Really Perform Badly for Graph Representation? In *Advances in Neural Information Processing Systems*, volume 34, 28877–28888.

Yun, X.-L.; Zhang, Y.-M.; Yin, F.; and Liu, C.-L. 2022. Instance GNN: A Learning Framework for Joint Symbol Segmentation and Recognition in Online Handwritten Diagrams. *IEEE Transactions on Multimedia*, 24: 2580–2594.

Zhang, K.; Zhu, Y.; Wang, J.; and Zhang, J. 2019. Adaptive structural fingerprints for graph attention networks. In *International Conference on Learning Representations*.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 586–595.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2021. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In *International Conference on Learning Representations*.