

Article

Dual-Tree-Guided Aspect-Based Sentiment Analysis Incorporating Structure-Aware Semantic Refinement and Graph Attention

Xinyu Wang ¹, Yuhang Gao ¹, Lv Xiao ¹, Kun Zhong ², Peisen Tan ¹ and Zhaobin Tan ^{1,*} 

¹ School of Electronic and Control Engineering, North China Institute of Aerospace Engineering, Langfang 065000, China; wxy20020715@163.com (X.W.); gaoyuhang@stumail.nciae.edu.cn (Y.G.); gzdxxl@stumail.nciae.edu.cn (L.X.); peisentan@stumail.nciae.edu.cn (P.T.)

² China Xiongan Group, Xiongan'an 071700, China; zhongkun4454@163.com

* Correspondence: tzb@nciae.edu.cn

Abstract

This work addresses the broken symmetry in syntactic–semantic representations for Aspect-Based Sentiment Analysis, where advancements have been driven by the use of pre-trained language models to achieve contextual understanding and graph neural networks capturing aspect–opinion dependencies using syntactic trees. However, long-distance aspect–opinion pairs pose challenges: the structural noise in dependency trees often causes erroneous associations, while the discrete structure of the constituent trees leads to constituent fragmentation. In this paper, we propose DySynGAT and introduce a Localized Graph Attention Network (LGAT) to fuse bi-gram syntactic and semantic information from both dependency and constituent trees, effectively mitigating interference from dependency tree noise. A dynamic semantic enhancement module efficiently integrates local and global semantics, alleviating constituent fragmentation caused by constituent trees. An aspect–context interaction graph (ACIG), built upon minimal semantic segmentation and jointly enhanced features, filters out noisy cross-clause edges. Spatial reduction attention (SRA) with mean pooling compresses the redundant sequential features, reducing the noise under long-range dependencies. Experiments on foods and beverages, electronics, and user review datasets demonstrate F1 score improvements of 0.55%, 3.55%, and 1.75% over SAGAT-BERT, demonstrating strong cross-domain robustness.

Keywords: aspect-based sentiment analysis; dependency tree; constituent tree; GAT; BERT



Received: 20 July 2025

Revised: 17 August 2025

Accepted: 1 September 2025

Published: 9 September 2025

Citation: Wang, X.; Gao, Y.; Xiao, L.; Zhong, K.; Tan, P.; Tan, Z.

Dual-Tree-Guided Aspect-Based Sentiment Analysis Incorporating Structure-Aware Semantic Refinement and Graph Attention. *Symmetry* **2025**, *17*, 1492. <https://doi.org/10.3390/sym17091492>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Aspect-Based Sentiment Analysis (ABSA) aims to predict the sentiment polarity of specific targets in a text. Its application has significantly improved the accuracy of multidimensional opinion mining, but traditional methods still face significant limitations. Although LSTM and its variant, BiLSTM, are able to model temporal dependencies, they are not capable of explicitly modeling grammatical structures such as phrase boundaries and subject–predicate relations, resulting in weak dynamic associations between aspectual words and opinion terms [1]. CNNs capture local co-occurrence patterns using multi-scale convolutional kernels, but their inability to model complex semantic relations stems from a lack of global syntactic hierarchical information [2]. While an attention mechanism can dynamically focus on the relevant text span, it is susceptible to interference from irrelevant

grammatical relations in long-distance dependency scenarios because it does not explicitly utilize the syntactic structure, leading to weight allocation that deviates from the true affective associations and triggers imbalances in the feature responses [3]. As shown in Figure 1, the traditional method may incorrectly assign weights to the modifier “delicious” because it cannot recognize the dependency association between “slow” and “service”, resulting in an imbalanced feature response.

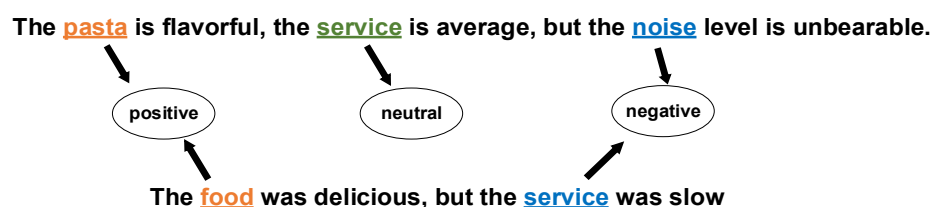


Figure 1. Example ABSA task where each aspect word is categorized according to the corresponding sentiment polarity.

In recent years, graph neural network (GNN) technology has increasingly become a research hotspot in ABSA. Such methods construct graph adjacency matrices based on dependency trees and update vertex representations by aggregating the features of neighboring nodes, thus reinforcing specific semantic associations. Figure 2 shows an example from a Laptop dataset containing a short dependency tree. The dependency relationship directly links the aspect word “keyboard” with the opinion word “responsive”. However, the existing methods face significant limitations in complex scenarios; traditional GCNs assign equal weight to all dependent edges [4,5], which generates cross-clause noise. Although graph attention networks (GATs) allocate weights dynamically using an attention mechanism, their performance heavily relies on the syntactic parser’s accuracy, and a static attention mechanism has difficulties adapting to diverse syntactic structures [6]. In addition, most existing studies neglect the explicit modeling of cross-aspect relations, which makes it difficult to capture the semantic transitions implied by the contrastive conjunction “but”, and the problem of remote dependency interference has not been effectively mitigated [7].

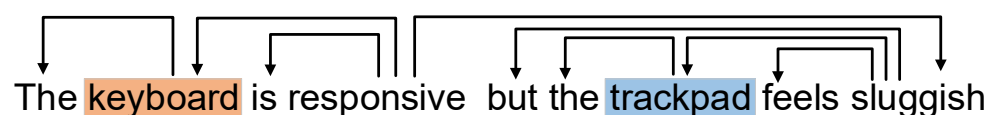


Figure 2. An example Laptop dataset containing a short dependency tree.

In this paper, we propose DySynGAT, a framework that addresses the challenge of multi-level affective semantic fusion by integrating dynamic semantics with syntactic structures. To enhance semantic representation, we employ BiLSTM [8], channel recalibration [9], and multi-scale trend sensing attention (MTA) [10] to capture contextual dependencies, suppress noisy features, and model semantic patterns at multiple granularities. For syntactic modeling, we fuse constituent and dependency trees [11] through Localized Graph Attention Networks (LGAT) to hierarchically aggregate structural information with adaptive weighting. To model interactions among multiple aspects, we construct an aspect–context interaction graph (ACIG) [12] using semantic span segmentation (MSS) and refine it with spatial reduction attention (SRA) [13] and LGAT quadratic aggregation, enabling effective capture of cross-sentence sentiment associations. Experimental results demonstrate that DySynGAT achieves superior performance by synergizing dynamic semantic enhancement with syntax-aware reasoning. The main contributions of this paper are summarized below.

- (1) Designing a dynamic semantic enhancement framework for multi-dimensional semantic fusion through a cascading feature refinement mechanism. The framework

employs BiLSTM to capture timing dependencies, and feature channel weights are adjusted via a channel recalibration module to suppress redundant noise. Introducing multi-scale convolutional structure and combining scale-aware attention to dynamically fuse local semantics of different granularities effectively suppresses the imbalance problem of channel feature response and sentence component fragmentation caused by the discrete structure of constituent trees.

- (2) Designing Localized Graph Attention Networks, LGAT dynamically aggregates multi-level syntactic information through a layer-by-layer attention mechanism. Explicitly inscribing phrase boundaries and hierarchical combinatorial relations achieves deep alignment of syntactic structures with semantic representations as a way to reduce the interference of structural noise.
- (3) The dynamic sparse ACIG is constructed by locating the smallest semantic unit between multiple targets through the MSS algorithm. Combining the SRA mechanism to compress redundant features and using LGAT to explicitly model multi-target affective associations, the long-distance dependent interference problem is effectively solved.

The rest of this paper is as follows. Section 2 summarizes the related work, in Section 3 the model components are described in detail, Section 4 demonstrates the performance validation of the model as well as the experimental analysis, and in Section 5 the relevant conclusions of this study and directions for future research are summarized.

2. Related Work

2.1. Conventional Methods

Traditional ABSA methods initially relied on LSTM and its bidirectional variant BiLSTM to capture temporal dependencies and to lay the groundwork for the initial association of opinion words with target aspects [14]. CNNs were introduced to capture multi-word co-occurrence patterns to enhance local feature extraction [15]. In recent years, the pre-trained language model BERT has achieved significant performance improvements in single-target scenarios through end-to-end fine-tuning, but its ability to explicitly model syntactic structures is limited, making it difficult to handle remote dependencies and cross-aspect semantic associations in complex sentences [16].

2.2. GAT-Based Approach

The GAT-based ABSA approach focuses on dependency tree-driven syntactic structure modeling with noise suppression. Wang et al. [17] proposed the Relational Graph Attention Network (R-GAT), which explicitly connects target aspect words with opinion words through a dependency tree optimization strategy to enhance the modeling of grammatical association paths. Zhang et al. [18] fused BERT coding with GAT, constructed an aspect–sentiment graph structure based on dependency trees to mitigate noise interference, and designed a dynamic edge weight assignment mechanism to suppress redundant cross-clause connections. For local–global semantic interactions in complex sentences, Chang et al. [19] introduced a dependency type-aware attention mechanism to enhance local context modeling through syntactic role constraints.

2.3. Dependency Tree vs. Constituent Tree

The joint modeling of dependency trees and constituent trees has become an important paradigm for improving the performance of ABSA tasks and as a mechanism to enhance local context modeling through syntactic role constraints. Nguyen et al. [20] proposed goal-dependent binary phrase dependency trees that synergize the fine-grained inter-word relations of dependency trees with the hierarchical phrase structure of constituent trees to

provide a syntactic basis to locate multi-granular semantic spans. Sun et al. [21] designed a dual-tree coding framework to enhance the complementarity of syntactic information across hierarchical levels by modeling bi-grammatical structures through GCNs and using a gating mechanism for feature fusion, respectively. Xu et al. [22] introduced Semantic Role Labeling (SRL) trees to expand the dimension of syntactic representations, constructed multi-view syntactic graphs of dependency trees, constituent trees and SRL trees, and improved the robustness and generalization ability of sentiment inference through heterogeneous syntactic feature interactions.

3. Methodology

As shown in Figure 3, DySynGAT consists of the following five components: an input encoder, dynamic semantic enhancements, syntax-aware graph aggregation, cross-aspect relationship modeling, and Classification of Emotional Polarity. The goal of this paper is to accurately model aspect–opinion word associations through multi-level syntactic–semantic collaborative reasoning. First, a fine-grained contextual representation of the target aspect is generated based on the BERT encoder, and the initial word vectors are obtained through subword aggregation. Second, a dual-channel parallel processing architecture is designed: the syntax-aware graph aggregation module fuses the constituent tree hierarchical phrase partitioning with the dependency tree fine-grained relations, and outputs syntactically enhanced representations by dynamically assigning neighbor weights through the Localized Graph Attention Network (LGAT). The dynamic semantic enhancement module effectively mitigates sentence component fragmentation and feature response due to constituent tree discretization through the synergistic combination of established techniques: BiLSTM for bidirectional sequential context, channel recalibration for adaptive feature channel weighting, and multi-scale convolution attention for capturing semantic patterns at varying granularities. This triple coupling operates by retaining the local context information while suppressing the redundant noise through feature channel weight adjustment and fusing different granularities of semantic patterns using multi-scale convolution. Subsequently, the syntactic and semantic features are weighted and spliced into a joint augmented representation. The joint augmented representation is inputted into the cross-aspect relationship modeling module, followed by locating the association units among multiple targets based on the Minimum Semantic Segmentation (MSS) algorithm and constructing aspect–context interaction graph (ACIG) by combining the joint augmented representation. Finally, the redundant sequence dimensions are compressed by the spatial reduction attention (SRA), and a secondary call to LGAT aggregates cross-clause sentiment associations to output a cross-aspect augmented representation; Ultimately, the cross-aspect augmented representation is adaptively fused with the joint augmented representation, and the sentiment polarity is output by pooling with a softmax classifier. Detailed information is provided below.

3.1. Input Encoder

A pre-trained BERT is used to capture the contextual semantic information of aspectual words in the input text. Specifically, for a given sentence $S = w_1, w_2, \dots, w_n$ and the target aspects to be analyzed a_t , this module follows BERT-SPC [23] to construct BERT-based sequences. The sequence contains special markers [CLS], original sentences, delimiter markers [SEP], aspects of the objectives and other [SEP] markers, formalized as

$$\text{BERT}_{\text{input}}^t = [[\text{CLS}], w_1, w_2, \dots, w_n, [\text{SEP}], a_t, [\text{SEP}]] \quad (1)$$

Among them, a_t is an aspect item consisting of a word or multiple words. The output of BERT is $H^t = \{h_0^t, h_1^t, \dots, h_L^t\}$, where h_0^t corresponds to the aggregated representation

of the [CLS] tag. The remaining h_i^t is the representation of the i -th word or subword in the sequence.

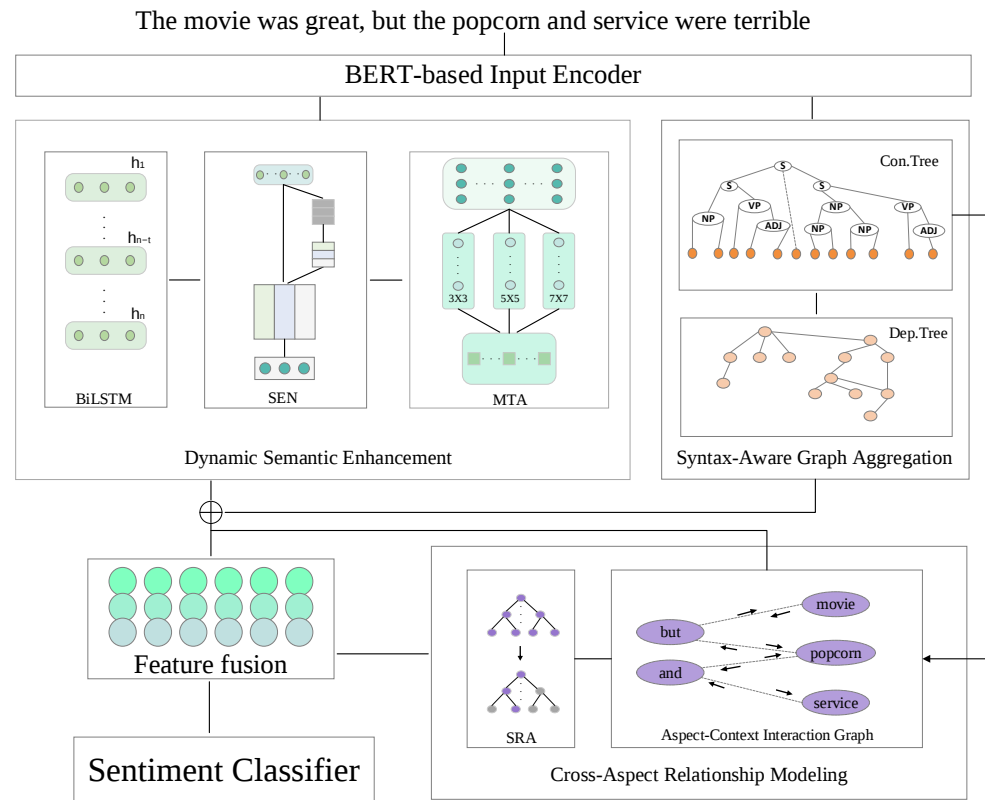


Figure 3. Overall architecture.

Since BERT's tokenizer may split a single word into multiple subwords (e.g., "tokenization" split into "token" and "ization"), so there is a need to aggregate subword representations of the same original word. To achieve this, for each word w_i in the sentence, its final contextual representation is computed by averaging the subword vectors as follows:

$$\tilde{h}_i^t = \frac{1}{|S(w_i)|} \sum_{k \in S(w_i)} h_k^t \quad (2)$$

Among them, $\tilde{h}_i^t$ indicates the aggregated context representation of word w_i at time step t , $S(w_i)$ indicates the set of indices of the corresponding subwords of word w_i in the input sequence, $|S(w_i)|$ is the number of subwords, and h_k^t is the representation of the k th subword in the sequence at time step t .

This encoding step produces a contextually enriched, word-aligned representation that preserves both syntactic and semantic information from the pre-trained language model.

3.2. Dynamic Semantic Enhancement

This module repairs compositional symmetry by reintegrating fragmented constituents through symmetric feature fusion.

3.2.1. Bidirectional Timing Dependency Modeling

We use BiLSTM to process the contextual representation $\tilde{h}_i^t$ generated by the input encoder, and the forward and backward hidden states are calculated by a bidirectional

recursive unit to obtain an enhanced temporal representation H_{rec} . This process can be formalized as

$$\vec{h}_t = \text{LSTM}(h_t, \vec{h}_{t-1}) \quad (3)$$

$$\overleftarrow{h}_t = \text{LSTM}(h_t, \overleftarrow{h}_{t+1}) \quad (4)$$

where $\overleftarrow{h}_t$ and $\vec{h}_t$ represent the backward and forward hidden states of the t -th word, respectively, and the two-way hidden state after splicing is $H_{\text{rec}} = [\vec{H}; \overleftarrow{H}]$.

This bidirectional recurrent modeling captures rich sequential context from both directions, forming the foundational temporal representation for subsequent dynamic enhancement. It ensures that each word's representation is informed by its entire context, which is crucial for accurate aspect sentiment understanding.

3.2.2. Channel Recalibration

Channel recalibration is used to add weights to key feature channels and suppress redundant and noisy channels. This module is based on the Squeeze and Excitation Network (SEN). This module performs global average pooling on temporal representation H_{rec} to generate channel statistics vector z , and learns channel importance through two fully connected layers:

$$w_c = \sigma(W_2 \cdot \delta(W_1 \cdot z)) \quad (5)$$

Among them, W_1 and W_2 are learnable parameters, δ is the ReLU activation function, and σ is the sigmoid function.

The characteristics after recalibration are shown below, where $\odot$ denotes channel-dimensional broadcast multiplication.

$$H_{\text{sc}} = W_c \odot H_{\text{rec}} \quad (6)$$

The channel recalibration mechanism dynamically adjusts the importance of different feature dimensions based on their global context. By learning attention weights (W_c) for each channel, it amplifies informative features (e.g., sentiment-bearing words) and suppresses less relevant or noisy ones, enabling the model to focus on the most salient semantic signals. This introduces a form of dynamic, feature-level adaptation.

3.2.3. Multi-Scale Trend Sensing Attention

Following channel-wise refinement, the model captures semantic patterns at varying granularities using the multi-scale trend sensing attention module. The multi-scale trend sensing module captures semantic patterns at different granularities. The scale-aware attention dynamically weights the contributions of features from different convolutional kernels, allowing the model to adaptively focus on the most relevant local and global semantic trends for the given context.

A multi-scale convolution is used in conjunction with joint attention to fuse local and global semantic trends. The set of convolution kernels is defined as $\mathcal{K} = \{3, 5, 7\}$, and multi-scale convolution is performed on the recalibrated feature H_{sc} to extract local operations

$$C_k = \text{Conv1D}_k(H_{\text{sc}}) \quad (7)$$

where Conv1D represents a 1D convolution with a kernel width of k . Then, multi-scale features are fused using scale-aware attention weighting

$$\beta_k = \text{Softmax}\left(\frac{(C_k W_Q)(C_k W_K)^T}{\sqrt{d_a}}\right) \quad (8)$$

where W_Q, W_K are the query and key transformation matrices, and $d_a = 25$ is the attention hidden layer dimension. The final dynamic semantic enhancement is expressed as:

$$H_{\text{enhanced}} = \sum_{k \in \mathcal{K}} \beta_k C_k W_V \quad (9)$$

The dynamic semantic enhancement module introduces an integration framework for established components (BiLSTM, SEN, and MTA), where the innovation lies not in redefining these individual techniques but in their systematic combination to address limitations inherent in their isolated use. Specifically, the bidirectional context modeling of BiLSTM ensures temporal coherence, while SEN dynamically recalibrates feature channels to suppress noise. MTA, in turn, enables multi-scale semantic abstraction through attention mechanisms. The synergy between these components—BiLSTM’s sequential dependency capture, SEN’s adaptive filtering, and MTA’s hierarchical pattern extraction—creates a feedback loop that mitigates individual drawbacks (e.g., local bias in BiLSTM and static weighting in SEN). This approach generates a rich, contextually aware, and dynamically weighted semantic representation, which is more robust and discriminative than representations from static or single-mechanism models.

3.3. Syntax-Aware Graph Aggregation

Having obtained the dynamically enhanced semantic representation H_{enhanced} from the previous module, the model now incorporates syntactic structural information to further refine contextual understanding. In this section, we construct a grammatical graph by leveraging constituency parsing and dependency parsing.

3.3.1. Hierarchical Phrase Structure Modeling

The input sentences are parsed into nested phrase units with a hierarchical structure using a constituent tree, including grammatical constituents such as noun phrases (NPs) and verb phrases (VPs). Specifically, through top-down recursive segmentation, sentences are first decomposed into major phrase units such as NPs and VPs. Each phrase unit is then further subdivided into smaller grammatical components (e.g., qualifiers, adjectives, etc.). Based on this hierarchical structure, this module constructs a corresponding adjacency matrix for each syntactic level with the following formula:

$$A_{\text{const}}^{(l)}(i, j) = \begin{cases} 1, & \text{If } w_i, w_j \text{ belong to phrase unit at level } l \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

Constituent trees, through hierarchical phrase segmentation, naturally isolate core semantic units within sentences, avoiding the potential introduction of redundant dependencies inherent in dependency trees. This phrase-level modeling provides the structural constraints essential for subsequent cross-aspect relation modeling.

3.3.2. Conditional Grammar Fusion

To combine the strengths of both constituent and dependency syntactic representations, we introduce a conditional grammar fusion strategy. First, the dependency adjacency matrix A_{dep} is constructed from the dependency tree. Then, based on the information about clause boundaries in the constituent tree clause, the edges across clauses in the dependency adjacency matrix A_{dep} are filtered out to obtain the revised matrix $A_{\text{dep}}^{\text{filtered}}$.

$$A_{\text{dep}}^{\text{filtered}}(i, j) = \begin{cases} A_{\text{dep}}(i, j), & \text{if } w_i, w_j \text{ It is also a clause} \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

Then, the fusion adjacency matrix of level l is constructed. The fusion adjacency matrix is as follows, where $\vee$ indicates a bitwise logical or operation:

$$A_{\text{fuse}}^{(l)} = A_{\text{const}}^{(l)} \vee A_{\text{dep}}^{\text{filtered}} \quad (12)$$

After constructing a modified dependency adjacency matrix to remove edge spanning clauses, this is then fused with the adjacency matrix of the constituent tree as a way of more accurately capturing the grammatical structure in a sentence.

Equation (11) filters cross-clause dependency edges, effectively mitigating noise introduced by long-range dependencies inherent in dependency trees. The logical OR operation in Equation (12) retains both the phrase-level adjacency relations from constituent trees and the local dependencies from dependency trees, thereby forming a multi-grained syntactic graph structure. This design preserves the structured advantages of constituent trees while harnessing the fine-grained dependency information from dependency trees.

3.3.3. Localized Graph Attention Network

A fused multi-grained syntactic graph $A_{\text{fuse}}^{(l)}$ is next constructed by integrating both dependency and constituent trees to serve as the backbone for contextual information propagation within LGAT. This integration enables LGAT to restore relational symmetry through a neighbor-weight-equilibration mechanism, which prioritizes syntactically adjacent nodes over global dependencies to mitigate the over-smoothing issue inherent in traditional GATs. Specifically, LGAT progressively aggregates multi-level syntactic information from local phrases to global clauses by stacking multi-layer graph attention networks (GATs). Each layer updates node features using the output of the previous layer, enabling the model to capture both fine-grained dependencies (from dependency trees) and structured phrase-level coherence (from constituent trees). First, the neighbor weight is assigned, and the attention coefficient is calculated for node w_i and neighbor w_j . W is the learnable projection matrix, a is the attention parameter vector, and $\parallel$ represents the splicing operation. The attention coefficient is calculated using the following formula:

$$e_{ij} = \text{LeakyReLU}\left(a^T (Wh_i \parallel Wh_j)\right) \quad (13)$$

Then, dynamic weight normalization is performed and invalid connections are masked based on $A_{\text{fuse}}^{(l)}$, where $\mathcal{N}(i)$ is the set of neighbors of w_i defined by $A_{\text{fuse}}^{(l)}$, as follows:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}(i)} \exp(e_{ik})} \quad (14)$$

Finally, feature aggregation is performed to aggregate the neighbor information by weighting and updating the node representation

$$h'_i = \text{GELU}\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} W_v h_j\right) \quad (15)$$

where h'_i denotes the updated representation of node i at the current layer, GELU is the activation function, h_j represents the feature vector of neighboring node j , and α_{ij} signifies the attention coefficient between nodes i and j .

Each layer GAT takes the output features of the previous layer as the input, and the input of layer l ($1 \leq l \leq L$) is the node feature matrix $H^{(l-1)}$. The initial context

representation $\tilde{h}_i^t$ is the input $H^{(0)}$ of the first layer, and the node representation is updated layer by layer according to Equation (15):

$$H^{(l)} = \text{LGAT}\left(H^{(l-1)}, A_{\text{fuse}}^{(l)}\right) \quad (16)$$

The final output syntax enhancement is represented as $G_{\text{syntax}} = H^{(l)}$. Syntactic enhancement means G_{syntax} , and semantic enhancement means H_{enhanced} . The two are combined by weighted concatenation to form joint enhancement H_{joint} .

$$H_{\text{joint}} = \beta G_{\text{syntax}} + (1 - \beta) H_{\text{enhanced}} \quad (17)$$

The LGAT employs dynamic weight assignment via Equations (13)–(15) to prioritize syntactically adjacent nodes over global dependencies. For instance, in the sentence “The food was delicious, but the service was slow”, LGAT focuses on the local phrase structures “food-delicious” and “service-slow”, rather than the inter-clausal connective “but”. This design mitigates the over-smoothing issue inherent in traditional graph attention networks while progressively aggregating hierarchical syntactic information from phrases to clauses through multi-layer stacking in Equation (16).

3.4. Cross-Aspect Relationship Modeling

3.4.1. Phrase Segmentation

Building upon the hierarchical structure provided by the constituent tree, the Minimum Semantic Span (MSS) algorithm extracts the smallest semantic unit connecting the two in its subtree by locating the lowest common ancestor node, LCA, that covers both target aspects. The specific process is as follows. First identify the LCA nodes, and subsequently analyze their subtree structure: if there are explicit connectives (e.g., “but”, “and”) in the direct child nodes, they are extracted directly; if not, it is traversed layer by layer toward finer-grained child nodes until a connective is found or a child node is reached. For example, in the sentence “The movie is great but the cinema is bad”, LCA is the root node S , and its direct child node contains “but”, so the split phrase s_{ij} is “but”.

3.4.2. Aspect–Context Interaction Graph

Based on the MSS results, the joint enhancement feature H_{joint} is used as the initial feature of the nodes to construct a dynamic sparse graph structure $\mathcal{G} = (\mathcal{V}, \varepsilon)$, and node set $\mathcal{V}$ contains all target aspects and their associated split phrases s_{ij} . Edge set ε defines the adjacency matrix A_{aci} .

$$A_{\text{aci}}(u, v) = \begin{cases} 1 & \text{if } u \in \{a_i, a_j\} \text{ and } v = s_{ij} \\ 1 & \text{if } u = s_{ij} \text{ and } v \in \{a_i, a_j\} \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

Based on the above construction process, the dynamic sparse graph structure output by ACIG contains the node feature matrix $\{h_{a_i}, h_{s_{ij}}\}$ and the adjacency matrix A_{aci} .

3.4.3. Spatial Reduction Attention

The characteristic of the fused nodes feature $\{h_{a_i}, h_{s_{ij}}\}$ is serialized and an enhanced eigenmatrix H_{fuse} is obtained. The sequence dimension is then compressed by mean pooling with a step size of 2 to achieve local convolutional dimensionality reduction. The formula is as follows:

$$H_{\text{pool}} = \text{AvgPool}_{\text{sr}=2}(H_{\text{fuse}}) \quad (19)$$

Subsequently, cavity convolution with a kernel width of 3 is used to capture multi-span local patterns as a way to achieve cavity convolution feature extraction, and residual concatenation and normalization are performed afterwards:

$$H_{\text{dilated}} = \text{DConv1}_{k=3, \text{dilation}=2}(H_{\text{pool}}) \quad (20)$$

$$H_{\text{osr}} = \text{LayerNorm}(H_{\text{dilated}} + H_{\text{pool}}) \quad (21)$$

ACIG enforces interaction symmetry through minimal symmetric units (MSS), while SRA maintains dimensional symmetry via isotropic compression.

3.4.4. Localized Graph Attention Network Secondary Aggregation

A cross-aspect augmented representation is generated by combining the adjacency matrix information of the aspect–context interaction graph with the spatiotemporal reduction features output by the SRA module, and then performing a second aggregation using LGAT (refer to Section 3.3.3). This aggregation focuses on cross-clause dependency modeling, with an input feature of H_{osr} and an adjacency matrix of A_{aci} .

3.5. Classification of Emotional Polarity

The affective polarity classification module takes the combined representation of the cross-aspect enhancement representation V_{cross} and the joint enhancement feature H_{joint} as input and generates the final prediction through adaptive feature fusion and pooling operations. The feature fusion formula is as follows, where α is the learnable fusion weight:

$$H_{\text{final}} = \alpha \cdot H_{\text{joint}} + (1 - \alpha) \cdot V_{\text{cross}} \quad (22)$$

Based on the feature-integrated H_{final} , the set of word positions a_t corresponding to each target aspect $S(a_t)$ is aggregated using mean pooling to obtain its contextual information:

$$h_{a_t} = \text{MeanPool}(\{H_{\text{final}}[i] \mid i \in S(a_t)\}) \quad (23)$$

The pooled representation is fed into the fully connected layer to generate the sentiment polarity probability distribution via softmax

$$P(y \mid a_t) = \text{Softmax}(W_c h_{a_t} + b_c) \quad (24)$$

where W_c is the classification weight matrix and b_c is the bias term. During model training, the model parameters are optimized using cross-entropy loss:

$$\mathcal{L} = -\frac{1}{M} \sum_{m=1}^M \sum_{k=1}^K y_{m,k} \log P(y \mid a_t) \quad (25)$$

4. Experimental

4.1. Experimental Dataset

This experiment validates the model performance on three publicly available English datasets: Restaurant (SemEval2014 Task4), Laptop (SemEval2014 Task4) [24], Twitter [25] and MAMS [26], and the dataset information is shown in Table 1.

Table 1. The detail of domain datasets.

Datasets	Positive		Neural		Negative	
	Train	Test	Train	Test	Train	Test
Restaurant	2164	728	637	196	807	196
Laptop	994	341	464	169	870	128
Twitter	1561	173	3127	346	1560	173
MAMS	3380	400	2764	607	2764	329

4.2. Experimental Details

In this paper, a pre-trained BERT-base model is fine-tuned, with a hidden layer dimension of 768 and its output dimension mapped to 200 dimensions through a fully connected layer. The model parameters are initialized by Xavier uniform distribution. Set the input dropout rate to 0.1, the batch size to 12, and the regularization coefficient to 1×10^{-5} . The optimizer AdamW is used, and its learning rate is set to 2×10^{-5} . Learning rates for the Restaurant, Laptop, and Twitter datasets are kept uniform to avoid domain bias. CRF Selection Parser is used to obtain the constituent tree and Biaffine Parser is used to obtain the dependency tree. The number of LGAT layers is set to 2. In this paper, accuracy and F1-score are used as the core evaluation metrics.

In our study, models were implemented using Python 3.10 and the deep learning framework PyTorch 2.5.0. The computational experiments were executed on a system equipped with an NVIDIA RTX 3060 GPU.

4.3. Experimental Program

In this paper, the model is compared with some classical ABSA models.

TD-LSTM [27]: LSTM is used to learn the contextual features of words, and they are finally spliced together for classification.

ATAE-LSTM [1]: The attention mechanism calculates the semantic relationship between words and the LSTM hidden state.

MemNet [28]: This relies on a multi-layered memory network to keep context and aspect information up to date.

IAN [29]: Interactive learning of aspect terms and sentence context information is achieved through two LSTMs and an attention mechanism.

ASGCN [30]: ABSA's syntactic dependency tree and graph convolutional network are combined.

AOA [31]: Attention mechanisms are used to complete sentences and interactively learn about aspects.

TD-GAT [32]: A directed acyclic graph (DAG) of the dependency network is constructed to obtain aspect word dependency information.

BERT-PT [33]: BERT is trained again in the client dataset to make it more suitable for the ABSA task.

AEN-BERT [23]: BERT encodes contextual words and aspect words, and then obtains classification features through a complex attention layer.

SAGAT-BERT [34]: BERT's output is modeled through GAT, and then classification features are obtained through a multi-layer attention mechanism and pooling operations.

DualGCN [35]: The discrete output of the dependency parser is replaced with a probability matrix of all dependency arcs, and two regularization factor constraints are proposed to capture semantic dependencies.

EK-GCN [36]: Sentiment lexicon and part of speech knowledge are integrated into a graph convolutional network, complemented by a word-sentence interaction

module that reinforces aspect–context interplay, achieving state-of-the-art results on four benchmark corpora.

FITE-GAT [37]: An explicit dependency relationship between aspect words and emotional words is established by combining the FT-RoBERTa induction tree and the graph attention network.

BERT-SPC [23]: The ABSC task is reformulated as a sentence-pair classification problem by adopting the input schema “[CLS] sentence [SEP] aspect [SEP]”, and employs the final hidden state corresponding to [CLS] for prediction.

T-GCN [4]: Label information from the dependency tree is incorporated into the graph convolution, thereby enabling the model to selectively emphasize semantically relevant label contexts. In this paper, we evaluate the performance of the model described herein and the baseline model on the ABSA task by two metrics: accuracy and F1-score.

4.4. Experimental Analysis

Here, DySynGAT is compared with other baseline models on the Restaurant, Laptop, Twitter, and MAMS datasets, and a detailed analysis of the experimental data is performed, leading to the following conclusions.

As shown in Table 2, the model DySynGAT has certain advantages on both the Restaurant and Laptop datasets. Although the accuracy (76.36%) on the Twitter dataset is second only to DualGCN (76.41%), the F1 value (75.58%) is still the best. The results show that DySynGAT effectively captures the information of emotional words related to aspect words, and suppresses the problems of parsing errors and cross-sentence noise interference.

Table 2. Performance of different models on three datasets, with the best result in bold and the second best result underlined. Some results are obtained by reproducing the original results and are marked with *.

Model	Laptop		Restaurant		Twitter		MAMS	
	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1
TD-LSTM	70.45	64.78	78.73	68.75	69.85	67.85	—	—
ATAE-LSTM	71.90	67.97	79.44	69.15	70.28	68.51	—	—
MemNet	72.64	65.17	79.61	69.64	71.48	69.90	—	—
IAN	72.05	67.38	79.26	70.09	72.50	70.81	76.60	—
AOA	72.62	67.52	79.97	70.42	72.31	70.25	—	—
ASGCN	75.29	70.67	80.80	71.51	71.56	70.40	—	—
DualGCN *	<u>79.61</u>	76.12	85.22	79.10	76.41	<u>74.92</u>	—	—
BERT-PT *	77.88	74.86	83.79	76.36	—	—	—	—
BERT-SPC	77.62	74.06	83.05	76.44	73.21	71.56	<u>82.82</u>	81.90
TGCN-BERT *	79.28	75.21	86.24	79.55	75.24	74.09	82.58	<u>82.06</u>
AEN-BERT *	78.35	74.62	83.03	74.21	73.88	72.56	—	—
TD-GAT	74.67	71.76	80.93	71.86	71.78	70.83	—	—
SAGAT-BERT *	79.24	74.56	84.08	78.63	74.32	73.83	—	—
EK-GCN	78.46	<u>76.54</u>	83.96	74.93	75.84	74.57	—	—
FITE-GAT	76.96	73.23	83.43	73.78	74.46	73.06	—	—
DySynGAT	81.17	78.11	<u>85.79</u>	<u>79.18</u>	<u>76.36</u>	75.58	82.85	82.68

Although ASGCN models grammatical relations by introducing syntactic dependency trees, its static dependency structure is vulnerable to syntactic parsing errors and difficult to adapt to dynamic semantic changes in complex sentence structures. In contrast, DySynGAT integrates complementary syntactic perspectives from both constituent and dependency trees, thereby constructing a syntactically enriched representation that enhances structural robustness and linguistic comprehensiveness. Furthermore, the Localized Graph Attention Network module dynamically assigns attention weights to syntactic edges within localized

neighborhoods, which not only mitigates the over-smoothing problem prevalent in standard GAT, but also reduces noise propagation from irrelevant or erroneous dependencies. This design enables DySynGAT to better preserve local structural integrity and adaptively focus on salient syntactic paths.

The DySynGAT model in this paper explicitly models multi-level syntactic associations such as phrase boundaries and subject–predicate relationships through a dynamic semantic enhancement module and a syntactic awareness graph aggregation module. This effectively suppresses the fragmentation of sentence components and significantly reduces the impact of structural noise on model performance. In addition, the model designs a cross-aspect relationship modeling module to improve the accuracy of sentiment analysis in complex sentence structures. Experimental results show that this model improves the F1 score of SAGAT-BERT by 3.55%, 0.55%, and 1.75% on the Laptop, Restaurant, and Twitter datasets, respectively.

Compared with DualGCN, DySynGAT improves the F1 score by 2.99%, 0.08%, and 0.66% on the Laptop, Restaurant, and Twitter datasets, respectively. This consistent gain, especially the significant improvement on the Laptop dataset, underscores the effectiveness of our LGAT in noise suppression and the value of fusing dual syntactic trees. Compared to TD-GAT, which utilizes only a single dependency tree, our LGAT innovatively fuses a multi-level adjacency matrix derived from both constituent and dependency trees, enabling a more holistic understanding of sentence structure.

In summary, DySynGAT achieves better performance through a synergistic integration of three complementary mechanisms: (1) dynamic semantic enhancement suppresses structural noise and component fragmentation by combining bidirectional sequential modeling, channel recalibration, and multi-scale trend sensing, ensuring context-aware, dynamically weighted representations; (2) syntax-aware graph aggregation leverages LGAT to fuse adjacency matrices from constituent and dependency trees, capturing both localized syntactic dependencies and global semantic relationships while overcoming the limitations of single-tree approaches; and (3) cross-aspect relationship modeling explicitly encodes long-range affective interactions across sentences, critical for handling discourse-rich contexts with contrasting opinions or contextual shifts. These mechanisms collectively address critical shortcomings of prior methods—such as static feature representations and single-tree biases—by enabling context-aware noise suppression, syntactic coherence, and cross-sentence decoupling. Experimental results on benchmarks like the Laptop dataset demonstrate consistent improvements, validating the effectiveness of this holistic design in modeling fine-grained dependencies and global semantic coherence in complex scenarios.

4.5. Ablation Experiments

A systematic ablation study was conducted to evaluate the contribution of each core component in the DySynGAT model for the sentiment classification task, with the results presented in Table 3. Several variants of DySynGAT were constructed sequentially.

Disabling the semantic enhancement module (DySynGAT-w/o-Semantic) led to F1 score reductions of 2.67%, 1.02%, 1.38%, and 1.51% on the Laptop, Restaurant, Twitter, and MAMS datasets, respectively. This performance degradation indicates that the semantic enhancement module plays a critical role in integrating local and global contextual information, thereby enhancing the representation capability of aspect-related semantics. To further validate the rationality and effectiveness of this module, ablation experiments were also conducted on its internal components. First, removing the BiLSTM component (DySynGAT-w/o-BiLSTM) resulted in a significant performance drop, with an F1 decrease of 2.94% on the Twitter dataset, highlighting the importance of sequential modeling in capturing contextual dependencies. Next, eliminating the SENet component caused F1 score drops

of 2.17%, 3.03%, 1.98%, and 0.86% across the four datasets, confirming its effectiveness in adaptively strengthening important feature responses through channel-wise attention. Subsequently, removing the multi-scale trend sensing attention module led to F1 reductions of 2.90%, 1.06%, 0.95%, and 0.49% on the four datasets, respectively, demonstrating its particularly significant contribution on the Laptop dataset.

Table 3. Ablation results for this model; “w/o” indicates the removal of a portion of the model structure.

Model	Laptop		Restaurant		Twitter		MAMS	
	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1
DySynGAT	81.17	78.11	85.79	79.18	76.36	75.58	82.85	82.68
DySynGAT-w/o-Fusion	80.01	76.08	84.15	77.93	76.17	75.21	82.31	82.54
DySynGAT-w/o-Semantic	78.95	75.44	84.89	78.16	75.18	74.20	81.81	81.17
DySynGAT-w/o-BiLSTM	80.96	77.61	84.36	77.41	74.29	72.64	82.78	82.25
DySynGAT-w/o-SEN	79.27	75.94	84.71	76.15	75.33	73.60	82.13	81.82
DySynGAT-w/o-MTA	78.79	75.21	85.25	78.12	75.18	74.63	82.70	82.19
DySynGAT-w/o-CrossAspect	79.74	76.13	84.09	76.41	74.29	72.63	81.30	80.95
DySynGAT-w/o-SRA	80.22	77.04	85.07	77.72	75.03	74.09	82.48	82.17
DySynGAT-w/o-LGAT	80.06	76.66	85.43	77.11	75.48	73.85	82.04	81.31

The removal of key modules, and their performances, were evaluated on four benchmark datasets: Laptop, Restaurant, Twitter, and MAMS.

Removing the SRA module resulted in a noticeable performance decline, with F1 score reductions of 1.07%, 1.46%, 1.49%, and 0.51% on the four datasets, respectively. This demonstrates that the SRA module effectively filters out syntactic noise and suppresses irrelevant cross-clause dependencies, thereby enhancing the model’s focus on sentiment-bearing phrases. Furthermore, the significant performance degradation observed when the LGAT module was removed confirms its effectiveness in modeling syntactic dependencies through the dynamic fusion of constituent trees and dependency trees, enabling more accurate capture of the relationships between aspect terms and their modifiers.

Disabling the feature fusion module (DySynGAT-w/o-Fusion) and instead adopting a concatenation-based approach for feature integration led to F1 score drops of 2.03%, 1.25%, 0.37%, and 0.14% on the four datasets, respectively. Moreover, this modification doubled the model training time, indicating that effective feature fusion is crucial for constructing robust semantic representations and improving training efficiency.

Finally, removing the cross-aspect relationship modeling module (DySyn-GAT-w/o-CrossAspect) led to a significant performance degradation, with F1 score reductions of 1.98%, 2.77%, 2.95%, and 1.73% on the respective datasets. This clearly demonstrates the indispensable role of this module in modeling interactive relationships among multiple aspects and enhancing the model’s holistic sentiment understanding capability.

In summary, the significant performance degradation observed upon removing any core module confirms the unique and effective contribution of each component to the overall capability of the DySynGAT model, and further validates the effectiveness of the proposed architecture.

4.6. Effect of LGAT Layer Number

An important parameter affecting the performance of the DySynGAT model is the number of LGAT layers L . We observe the impact of changing the number of layers on the model’s performance on the three datasets, as shown in Figure 4. When the number of layers in the LGAT module is $L = 2$, this model performs well on the three datasets. When $L < 2$, the model can only capture shallow grammatical dependencies of local phrases, and cannot effectively model nested structures or long-distance modification relationships,

resulting in misjudgments. When $L > 2$, the node features are smoothed after multiple aggregations, and the high-level features lose local details, resulting in worse results. When $L = 5$, the model's performance on some dataset rebounds, indicating that the model mitigates the oversmoothing problem through residual connections and an adaptive gating mechanism. In addition, the structure $L = 5$ may be more suitable for specific complex sentence patterns that exist in the data, improving its performance in specific scenarios.

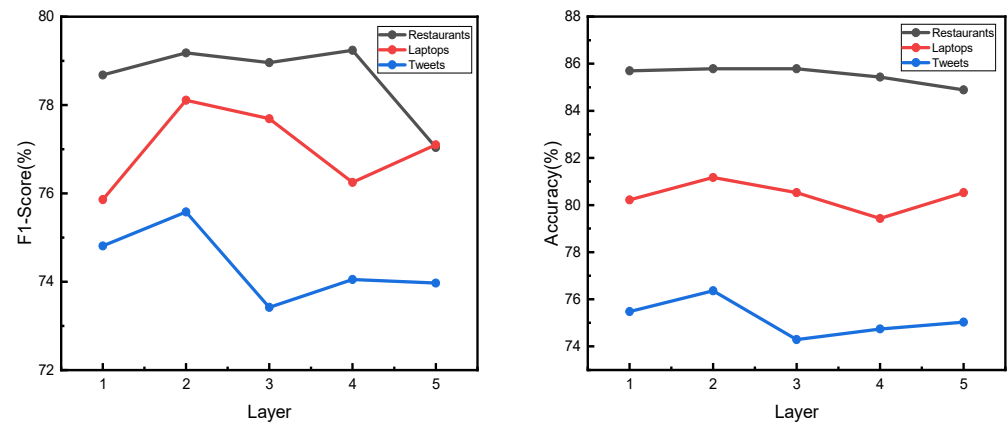


Figure 4. Experimental results of different LGAT layers.

4.7. Case Study

We conducted a case study of the original model and two variants and used two example sentences for discussion, the results of which are shown in Table 4. In the first sentence, because the semicolon separates the context into independent clauses, the dependency tree parser incorrectly connects “screen display” with “charging speed” after the semicolon, resulting in an incorrect polarity judgment in the w/o-Semantic model. The original model and the w/o-LGAT variant were recalibrated through channel recalibration, adaptively reducing the fusion weight of the syntactic channel and increasing the weight of the semantic channel during training, thereby accurately predicting the negative polarity of aspect words. In the analysis of the second sentence, due to the deep dependency tree structure caused by the contrastive structure introduced by the logical conjunction “Although”, the w/o-LGAT model failed to capture the dependency path due to the lack of a syntactic aggregation module, incorrectly associating “steak” with the irrelevant word “side dishes”, resulting in a neutral prediction. The original model correctly identifies the negative polarity of “steak” by aggregating multi-level graph attention through the LGAT module and dynamically integrating the phrase segmentation of the constituent tree and the fine-grained relationships of the dependency tree. Comparative experiments show that variants without semantic modules (w/o-Semantic) are unable to model the contrastive meaning of logical connectives, incorrectly associating “restaurant” with the negative word “hard,” thereby validating the necessity of semantic enhancement.

For the sentence “I’ve had it for about 2 months now and found no issues with software or updates,” DySynGAT effectively disentangled the explicit negation (“no issues”) from the implicit positive inference, whereas w/o-Semantic failed to reconcile the contradiction between the absence of problems and the implied endorsement. Meanwhile, w/o-LGAT achieved partial accuracy by focusing on semantic cues. In the second case (“No installation disk is included”), DySynGAT accurately treated the statement as neutral by isolating the factual assertion from potential user expectations, while w/o-Semantic erroneously projected a negative sentiment by conflating the negation with dissatisfaction.

Table 4. Pos, neg, and neu stand for positive, negative, and neutral, respectively. Aspects are shown in bold, with incorrect predictions indicated by a × and correct predictions indicated by a √.

Sentence	Aspects	DySynGAT		w/o-Semantic		w/o-LGAT	
The screen display effect is very delicate and the colors are bright; But the battery life is extremely poor and the charging speed is also disappointing.	screen display	Pos	√Pos	Pos	×Neu	Pos	√Pos
	battery life	Neg	√Neg	Neg	×Neu	Neg	×Neu
	charging speed	Neg	√Neg	Neg	×Neu	Neg	×Neu
Although the restaurant has an elegant environment and friendly waiters, the main steak is hard, and the side dishes are nothing new.	restaurant	Pos	√Pos	Pos	×Neu	Pos	√Pos
	steak	Neg	√Neg	Neg	√Neg	Neg	×Neu
	side dishes	Neg	√Neg	Neg	√Neg	Neg	√Neg
I’ve had it for about 2 months now and found no issues with software or updates .	software	Pos	√Pos	Pos	×Neu	Pos	√Pos
	updates	Pos	√Pos	Pos	×Neu	Pos	√Pos
No installation disk is included.	installation disk	Neu	√Neu	Neg	×Neg	Neu	√Neu

5. Conclusions

This study proposes DySynGAT, a dual-tree-guided framework for Aspect-Based Sentiment Analysis (ABSA), designed to address the challenges of fine-grained sentiment polarity prediction in multi-aspect contexts. By integrating constituent and dependency trees, DySynGAT explicitly models syntactic–semantic interactions, effectively suppressing structural noise and mitigating long-distance dependency interference. The dynamic semantic enhancement module captures contextual cues around aspect terms, while the syntactic perceptual graph aggregation module refines structural representations. A cross-aspect relationship modeling module further constructs dynamic sparse graphs to resolve contradictions among co-occurring aspects. Experimental results on four domain datasets demonstrate that DySynGAT outperforms traditional models, particularly in scenarios with complex emotional expressions and multi-aspect conflicts.

In future work, the team aims to improve the model’s ability to process long texts. We plan to improve the input encoder to better capture the semantic information of long texts.

Author Contributions: Conceptualization, K.Z. and Z.T.; Data curation, Y.G.; Formal analysis, L.X.; Validation and Supervision, P.T.; Writing—original draft, X.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on demand from the corresponding author at tzb@nciae.edu.cn.

Conflicts of Interest: Author Kun Zhong was employed by China Xiongan Group. The remaining authors declare that this research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

LGAT	Localized Graph Attention Network
ACIG	Aspect–Context Interaction Graph
SRA	Spatial Reduction Attention

References

1. Wang, Y.; Huang, M.; Zhu, X.; Zhao, L. Attention-based LSTM for aspect-level sentiment classification. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 606–615.
2. Li, D.; Yao, A.; Chen, Q. PSConv: Squeezing Feature Pyramid into One Compact Poly-Scale Convolutional Layer. In Proceedings of the 16th European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; Lecture Notes in Computer Science. Springer: Berlin/Heidelberg, Germany, 2020; Volume 12347, pp. 615–632. [[CrossRef](#)]

3. Zhao, T.; Du, J.; Shao, Y.; Li, A. Aspect-Based Sentiment Analysis Using Local Context Focus Mechanism with DeBERTa. In Proceedings of the IEEE International Conference on Web Services (ICWS), Tianjin, China, 22–24 September 2023; IEEE: New York, NY, USA, 2023; pp. 123–130. [\[CrossRef\]](#)
4. Tian, Y.; Chen, G.; Yan, S. Aspect-Based Sentiment Analysis with Type-Aware Graph Convolutional Networks and Layer Ensemble. In Proceedings of the North American Chapter of the Association for Computational Linguistics, Mexico City, Mexico, 6–11 June 2021; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA, 2021; pp. 292–302. [\[CrossRef\]](#)
5. Feng, J.; Cai, S.; Li, K.; Chen, Y.; Cai, Q.; Zhao, H. Fusing Syntax and Semantics-Based Graph Convolutional Network for Aspect-Based Sentiment Analysis. *Int. J. Data Warehous. Min.* **2023**, *19*, 1–15. [\[CrossRef\]](#)
6. Wang, P.; Zhang, S.; Li, Z.; Hou, J. Enhancing Ancient Chinese Understanding with Derived Noisy Syntax Trees. In Proceedings of the ACL 2023 Short Research Workshops (ACL-SRW), Toronto, CA, USA, 9–14 July 2023; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA, 2023; pp. 150–158. [\[CrossRef\]](#)
7. Kovtunen, I.V. Rhetorical relations of contrast in the blog text: Marking means, semantics, functions. *Rudn. J. Russ. Foreign Lang. Res. Teach.* **2018**, *16*, 63–90. [\[CrossRef\]](#)
8. Yang, G.; Xu, H. A Residual BiLSTM Model for Named Entity Recognition. *IEEE Access* **2020**, *8*, 1. [\[CrossRef\]](#)
9. Xia, L.; Yu, J.; Jiang, K.; Hu, Z.; Xie, Y. Digital Camouflage Generation Based on an Improved CycleGAN Network Model. *Int. J. Adv. Netw. Monit. Control.* **2024**, *9*, 89–99. [\[CrossRef\]](#)
10. Chen, X.; Li, D.; Liu, M.; Jia, J. CNN and Transformer Fusion for Remote Sensing Image Semantic Segmentation. *Remote Sens.* **2023**, *15*, 4455. [\[CrossRef\]](#)
11. Zhou, J.; Zhang, S.; Zhao, H. Concurrent Parsing of Constituency and Dependency. *arXiv* **2019**, arXiv:1908.06379. [\[CrossRef\]](#)
12. Jaime, J.A.S.; Robert, R.W.; Tom, L.; Scott, B. Fracture Mapping Using Offset Vector Tile Technology. In Proceedings of the IV Simpósio Brasileiro de Geofísica (4SIMBGF), Brasilia, Brazil, 14–17 November 2010; Society of Exploration Geophysicists (SEG): Houston, TX, USA, 2010; pp. 18–25. [\[CrossRef\]](#)
13. Yao, G.; Zhu, S.; Zhang, L.; Qi, M. HP-YOLOv8: High-Precision Small Object Detection Algorithm for Remote Sensing Images. *Sensors* **2024**, *24*, 4858. [\[CrossRef\]](#)
14. Zhu, Y.; Gao, X.; Zhang, W.; Liu, S.; Zhang, Y. A Bi-Directional LSTM-CNN Model with Attention for Aspect-Level Text Classification. *Future Internet* **2018**, *10*, 116. [\[CrossRef\]](#)
15. Lu, H.Y.; Zhang, M.; Liu, Y.Q.; Ma, S.P. Convolution neural network feature importance analysis and feature selection enhanced model. *J. Softw.* **2017**, *28*, 2879–2890.
16. Sun, C.; Huang, L.; Qiu, X. Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. *arXiv* **2019**, arXiv:1903.09588. [\[CrossRef\]](#)
17. Wang, K.; Shen, W.; Yang, Y.; Quan, X.; Wang, R. Relational Graph Attention Network for Aspect-based Sentiment Analysis. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Seattle, WA, USA, 5–10 July 2020. [\[CrossRef\]](#)
18. Zhang, J.; Cui, Z.; Park, H.J.; Noh, G. BHGAttn: A Feature-Enhanced Hierarchical Graph Attention Network for Sentiment Analysis. *Entropy* **2022**, *24*, 1691. [\[CrossRef\]](#)
19. Chang, B.; Lee, I.; Kim, H.; Kang, J. “Killing Me” Is Not a Spoiler: Spoiler Detection Model using Graph Neural Networks with Dependency Relation-Aware Attention Mechanism. *arXiv* **2021**, arXiv:2101.05972.
20. Nguyen, T.H.; Shirai, K. Phrasernn: Phrase recursive neural network for aspect-based sentiment analysis. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 2509–2514.
21. Sun, K.; Zhang, R.; Mensah, S.; Mao, Y.; Li, X. Aspect-Level Sentiment Analysis via Convolution over Dependency Tree. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), Hong Kong, China, 3–7 November 2019; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA, 2019; pp. 5678–5689. [\[CrossRef\]](#)
22. Xu, H.; Liu, S.; Wang, W.; Deng, L. RAG-TCGCN: Aspect Sentiment Analysis Based on Residual Attention Gating and Three-Channel Graph Convolutional Networks. *Appl. Sci.* **2022**, *12*, 12108. [\[CrossRef\]](#)
23. Song, Y.; Wang, J.; Jiang, T.; Liu, Z.; Rao, Y. Attentional encoder network for targeted sentiment classification. *arXiv* **2019**, arXiv:1902.09314. [\[CrossRef\]](#)
24. Pontiki, M.; Galanis, D.; Papageorgiou, H.; Androutopoulos, I.; Manandhar, S.; Al-Smadi, M.; Al-Ayyoub, M.; Zhao, Y.; Qin, B.; De Clercq, O.; et al. SemEval-2016 Task 5: Aspect Based Sentiment Analysis. In Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), San Diego, CA, USA, 16–17 June 2016; pp. 19–30. [\[CrossRef\]](#)
25. Li, D.; Furu, W.; Chuanqi, T.; Duyu, T.; Ming, Z.; Ke, X. Adaptive Recursive Neural Network for Target-Dependent Twitter Sentiment Classification. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014), Baltimore, MD, USA, 22–27 June 2014; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA, 2014; Volume 2, pp. 55–60. [\[CrossRef\]](#)

26. Jiang, Q.; Chen, L.; Xu, R.; Ao, X.; Yang, M. A challenge dataset and effective models for aspect-based sentiment analysis. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), Hong Kong, China, 3–7 November 2019; pp. 6280–6285.
27. Tane, D.; Qin, B.; Feng, X.; Liu, T. Effective LSTMs for Target-Dependent Sentiment Classification. *arXiv* **2015**, arXiv:1512.01100. [[CrossRef](#)]
28. Tang, D.; Qin, B.; Liu, T. Aspect level sentiment classification with deep memory network. *arXiv* **2016**, arXiv:1605.08900. [[CrossRef](#)]
29. Ma, D.; Li, S.; Zhang, X.; Wang, H. Interactive Attention Networks for Aspect-Level Sentiment Classification. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017; pp. 4068–4074. [[CrossRef](#)]
30. Zhang, C.; Li, Q.; Song, D. Aspect-based Sentiment Classification with Aspect-specific Graph Convolutional Networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019. [[CrossRef](#)]
31. Huang, B.; Ou, Y.; Carley, K.M. Aspect level sentiment classification with attention-over-attention neural networks. In Proceedings of the Social, Cultural, and Behavioral Modeling: 11th International Conference, SBP-BRIMS 2018, Washington, DC, USA, 10–13 July 2018; Springer International Publishing: Cham, Switzerland, 2018. Proceedings 11. pp. 197–206.
32. Huang, B.; Carley, K.M. Syntax-aware aspect level sentiment classification with graph attention networks. *arXiv* **2019**, arXiv:1909.02606. [[CrossRef](#)]
33. Xu, H.; Liu, B.; Shu, L.; Yu, P.S. BERT post-training for review reading comprehension and aspect-based sentiment analysis. *arXiv* **2019**, arXiv:1904.02232.
34. Huang, L.; Sun, X.; Li, S.; Zhang, L.; Wang, H. Syntax-Aware Graph Attention Network for Aspect-Level Sentiment Classification. In Proceedings of the 28th International Conference on Computational Linguistics (COLING 2020), Barcelona, Spain, 8–13 December 2020; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 700–709. [[CrossRef](#)]
35. Li, R.; Chen, H.; Feng, F.; Ma, Z.; Wang, X.; Hovy, E. Dual Graph Convolutional Networks for Aspect-Based Sentiment Analysis. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL 2021), Bangkok, Thailand, 1–6 August 2021; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA, 2021; pp. 6441–6452. [[CrossRef](#)]
36. Gu, T.; Zhao, H.; He, Z.; Li, M.; Ying, D. Integrating External Knowledge into Aspect-Based Sentiment Analysis Using Graph Neural Network. *Knowl. Based Syst.* **2022**, *256*, 110025. [[CrossRef](#)]
37. Fan, M.; Kong, M.; Wang, X.; Hao, F.; Zhang, C. FITE-GAT: Enhancing aspect-level sentiment classification with FT-RoBERTa induced trees and graph attention network. *Expert Syst. Appl.* **2024**, *264*, 125890. [[CrossRef](#)]

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.