A Frequency-Channel Enhanced Dynamic GCN for Aspect-Based Sentiment Analysis

Shaokun Liu[1,2,3,4], Mieradilijiang Maimaiti[1,2,3,4,*], and Wushour Silamu[1,2,3,4]
[1]School of Computer Science and Technology, Xinjiang University, Urumqi, China
[2]Xinjiang Laboratory of Multi-Language Information Technology, Xinjiang University, Urumqi, China
[3]Xinjiang Multilingual Information Technology Research Center, Urumqi, China
[4]Joint International Research Laboratory of Silk Road Multilingual Cognitive Computing, Urumqi, China
Email: liushaokun@stu.xju.edu.cn,{miradeljan51,wushour}@xju.edu.cn

***Abstract*—Aspect-based Sentiment Analysis (ABSA) is a fundamental task in natural language processing. Most existing methods rely on graph-based approaches with dependency trees, which often suffer from parsing noise and rigid static graph structures. To address these issues, we propose a Frequency-Channel Enhanced Dynamic Graph Convolutional Network (FCD-GCN). Unlike static models, our approach introduces a dynamic graph mechanism that iteratively updates edge weights across layers, enabling adaptive capture of aspect-context associations. Our method consists of three steps: layer-wise dynamic graph construction, frequency-channel feature enhancement, and sentiment prediction. Additionally, a dual-attention strategy combines a frequency-domain module using Fourier transforms to model long-range dependencies and a channel-aware self-attention module to reduce feature redundancy. Extensive experiments on benchmark datasets demonstrate that FCD-GCN consistently outperforms strong baselines, validating the effectiveness of dynamic graph refinement with frequency and channel-level modeling. The code is publicly available[1].**

***Index Terms*—Aspect-Based Sentiment Analysis, Dynamic Graph Learning, Frequency Attention, Channel Self-Attention, Graph Neural Networks.**

I. Introduction

Aspect-based Sentiment Analysis (ABSA) is a fine-grained sentiment analysis task that aims to identify the polarity of the sentiment (e.g., positive, negative, or neutral) of specific aspect terms within a sentence [1]–[3]. Unlike sentence-level sentiment classification, which assigns a single label to an entire document, ABSA must disentangle the sentiments of multiple aspects that may coexist in the same context [4], [5]. This is particularly challenging in real-world scenarios where users often express contrasting opinions toward different aspects of an entity in a single sentence [6]. To accurately predict the sentiment of a specific aspect, the model must effectively capture the semantic dependency between the aspect term and its corresponding opinion words while ignoring irrelevant context.

However, in many cases, the aspect and its opinion word are separated by a long distance in the surface sequence, making it difficult for traditional sequence-based models (such as LSTM or standard attention mechanisms) to capture their interactions [7]. Syntactic dependency trees offer a solution by providing structural connections that shorten the distance between related words [8]. Consider the example sentence shown in Figure 1: *"The food is delicious but the service is terrible."* As illustrated in Figure 1, although the aspect "food" and the negative adjective "terrible" appear in the same sentence, they are structurally distant in the dependency tree. In contrast, there is a direct syntactic path connecting "food" to its positive opinion word "delicious" (e.g., through the *nsubj* relation). By performing graph convolutions over such dependency structures, Graph Neural Networks (GNNs) can explicitly guide the information flow from opinion words to the target aspect, thereby reducing the interference from irrelevant context and improving classification accuracy.

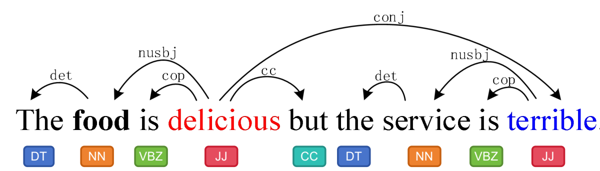


Fig. 1: An example of a dependency tree for a multi-aspect sentence.

Despite the effectiveness of GNN-based approaches, representative models such as ASGCN [9], RGAT [10], and DGEDT [11] encounter limitations in complex scenarios. A primary drawback is their heavy reliance on static dependency trees generated by off-the-shelf parsers, which inevitably propagate noise and erroneous connections when processing informal texts. Furthermore, as these spatial-domain models stack layers to capture long-range dependencies, they often suffer from the over-smoothing problem intrinsic to GCNs, washing out high-frequency signals critical for distinguishing sentiment shifts. Consequently, they lack the capability to take advantage of the features in the frequency-domain that are robust against such local structural noise.

To address these limitations, we propose a novel framework named Frequency-Channel Dynamic Graph Convolutional Network (FCD-GCN). Instead of relying on a fixed and potentially noisy static graph, FCD-GCN incorporates a dynamic structure learning mechanism that adaptively refines the adjacency matrix during training. This allows the model to prune noisy edges and enhance connections relevant to the

*Corresponding author.

[1]https://github.com/xjulsk/FCDGCN

target aspect. Moreover, to tackle the over-smoothing issue and capture richer semantic features, we introduce a Frequency Attention module based on the discrete Fourier transform. By transforming features into the spectral domain, our model can effectively capture global dependencies and preserve discriminative high-frequency details that are often lost by spatial convolutions. We further integrate a Channel Self-Attention mechanism to explicitly model the inter-dependencies between feature channels, adaptively reweighting the feature subspaces to suppress redundant information.

The main contributions of this paper are summarized as follows:

- We propose FCD-GCN, a dynamic architecture that reduces reliance on static parsing tools by learning to refine the graph structure adaptively.
- We introduce a novel Frequency Attention mechanism that captures global semantic patterns in the spectral domain, offering a new perspective to mitigate the over-smoothing problem in GCNs.
- We conduct extensive experiments on three benchmark datasets. The results demonstrate that FCD-GCN consistently outperforms state-of-the-art baselines, proving its effectiveness in handling complex sentiment dependencies.

II. Related Work

Early neural approaches for Aspect-Based Sentiment Analysis primarily focused on attention-based sequence modeling to capture the semantic correlation between aspect terms and their contexts [12]. Wang et al. [13] first integrated aspect embeddings in Long Short-Term Memory (LSTM) networks, using an attention mechanism to highlight aspect-specific keywords. Recognizing the need for bidirectional interaction, Ma et al. [14] designed an interactive attention network to iteratively learn representations for both the target and the context. To capture finer-grained interactions, Huang et al. [15] introduced an "Attention-over-Attention" module, while Ma et al. [16] incorporated the affective common sense knowledge of SenticNet to enhance sentiment identification. Additionally, Li et al. [17] used a CNN-based transformation network to generate target-specific representations, reducing the reliance on external parsers. However, treating sentences as linear sequences often restricts the ability to capture long-range dependencies, prompting a shift towards syntax-aware methods. Zhang et al. [9] pioneered the use of Graph Convolutional Networks over dependency trees, using syntactic paths to bridge the gap between distant aspect-opinion pairs. Extending this, Wang et al. [10] argued that dependency types carry crucial information and explicitly proposed encoding relational labels, while Tang et al. [11] enhanced feature expressiveness by combing flat Transformer representations with graph-based structural features in a dual-stream architecture.

Despite the success of syntax-based models, relying solely on static dependency trees introduces noise from parsing errors. Consequently, recent research has focused on dynamic structures and multi-view learning to enhance robustness. Li et al. [18] constructed a dual-graph architecture to simultaneously capture syntactic structures and semantic correlations using orthogonal regularizers. To mitigate the impact of noise, Zhang et al. [19] introduced aspect-aware attention combined with syntactic mask matrices. Furthermore, Zhao et al. [20] and Lu et al. [21] explored feature aggregation strategies, leveraging multi-scale features and semantic preserving mechanisms to alleviate information loss in deep networks. More recently, Zheng et al. [22] proposed transferring sentiment knowledge across aspects to dynamically update features, while Yin et al. [23] integrated graph-guided structural bias into Transformers to combine global modeling with structural constraints. Additionally, Liu et al. [24] proposed iteratively refining the graph structure during training to capture evolving dependencies. Our work advances this trajectory by introducing frequency-domain spectral filtering to address the over-smoothing problem inherent in these dynamic graph approaches.

III. Methodology

In this section, we present the proposed FCD-GCN model in detail. The overall framework is illustrated in Fig. 2.

A. Attention Layer

We design a dual-stream attention module that integrates aspect-aware attention and self-attention to capture both target-specific semantics and global contextual dependencies. We employ p parallel attention heads to enrich the representation subspace.

1) Aspect-aware Attention: Given the fine-grained nature of ABSA, modeling dependencies conditioned on specific aspects is crucial. We formulate the aspect representation as a query to evaluate its relevance against each context token:

$$A^i_{\text{asp}} = \tanh\Big(H_a W^a \times \big(KW^k\big)^\top + b\Big), \tag{1}$$

where $K = H$ denotes the contextual representation of the encoder and W^a, W^k are learnable weights. $H_a \in \mathbb{R}^{n\times d}$ is constructed by replicating the mean-pooled aspect vector h_a n times. This operation yields aspect-aware matrices $\{A^i_{\text{asp}}\}$ for p heads.

2) Self-Attention: Complementarily, to model global interactions among all tokens, we incorporate multi-head self-attention (Vaswani et al., 2017). The attention scores for the i-th head are computed as:

$$A^i_{\text{self}} = \frac{QW^Q \times \big(KW^K\big)^\top}{\sqrt{d}}, \tag{2}$$

where $Q = K = H$, and W^Q, W^K are projection parameters.

3) Head-wise Fusion: We integrate the aspect-aware and self-attention scores to form a combined adjacency matrix for each head:

$$A^i = A^i_{\text{asp}} + A^i_{\text{self}}, \tag{3}$$

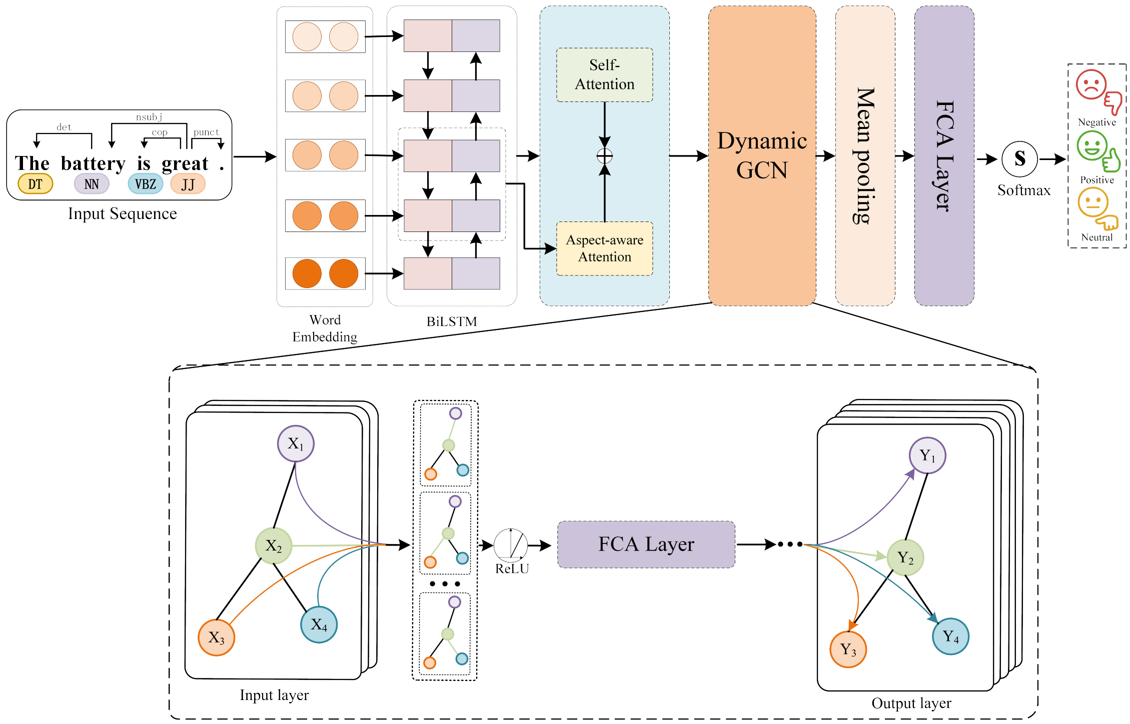


Fig. 2: Overall architecture of the proposed Frequency-Channel Enhanced Dynamic Graph Convolutional Network.

Here, $A^i \in \mathbb{R}^{n\times n}$ represents the fully connected attention graph of the i-th head. Subsequently, the p matrices $\{A^i\}_{i=1}^{p}$ are aggregated into a unified structure:

$$A^{(0)} = \text{Fuse}(A^1, A^2, \cdots, A^p), \tag{4}$$

where $\text{Fuse}(\cdot)$ denotes an aggregation operation (e.g., linear projection). The resulting $A^{(0)}$ serves as the initial topology for the subsequent Dynamic GCN Layer.

B. Frequency–Channel Attention Layer

The Frequency–Channel Attention Layer provides a refinement module that is repeatedly called inside the graph propagation process to enhance node representations. as shown in Fig. 3.

Given a sequence of hidden states $H \in \mathbb{R}^{n\times d}$ from a GCN layer, we first reshape it into a 2D feature map $X \in \mathbb{R}^{d\times 1\times n}$ by treating the feature dimension as channels and the token positions as a one-dimensional spatial axis. On top of X, we stack a channel self-attention branch and a frequency-domain attention branch, and then fuse their outputs through a gated residual connection.

1) Channel Self-Attention: The channel self-attention branch aims to model correlations among different feature channels and reweight informative subspaces. We first apply three learnable 1×1 convolutions (or linear layers) on X to obtain channel-wise query, key and value tensors:

$$Q_{\text{ch}} = f_q(X), \quad K_{\text{ch}} = f_k(X), \quad V_{\text{ch}} = f_v(X), \tag{5}$$

where $Q_{\text{ch}}, K_{\text{ch}}, V_{\text{ch}} \in \mathbb{R}^{d\times 1\times n}$ shares the same shape as X. For each channel, we flatten the spatial dimension and compute channel affinities:

$$A_{\text{ch}} = \text{softmax}\left(Q_{\text{ch}} K_{\text{ch}}^{\top}\right), \tag{6}$$

where $A_{\text{ch}} \in \mathbb{R}^{d\times d}$ measures how one channel can be reconstructed from the others. The channel-refined features are obtained by

$$Z_{\text{ch}} = A_{\text{ch}} V_{\text{ch}}, \tag{7}$$

which are then reshaped to the same size as X.

2) Frequency Attention: In parallel, the frequency attention branch captures long-range and multi-scale patterns along the sequence dimension. We reuse the projected tensors and apply a 2D Fast Fourier Transform along the spatial axes:

$$\widehat{Q} = \mathcal{F}(Q_{\text{ch}}), \quad \widehat{K} = \mathcal{F}(K_{\text{ch}}), \quad \widehat{V} = \mathcal{F}(V_{\text{ch}}), \tag{8}$$

where $\widehat{Q}, \widehat{K}, \widehat{V}$ are complex-valued frequency-domain representations. Attention is then computed in the frequency domain:

$$\widehat{A} = \text{softmax}\left(\widehat{Q}\,\widehat{K}^{\text{H}}\right), \tag{9}$$

and the attended response is given by

$$\widehat{Z}_{\text{freq}} = \widehat{A}\,\widehat{V}. \tag{10}$$

Finally, we apply the inverse Fourier transform to obtain the frequency-enhanced features in the original domain:

$$Z_{\text{freq}} = \mathcal{F}^{-1}(\widehat{Z}_{\text{freq}}). \tag{11}$$

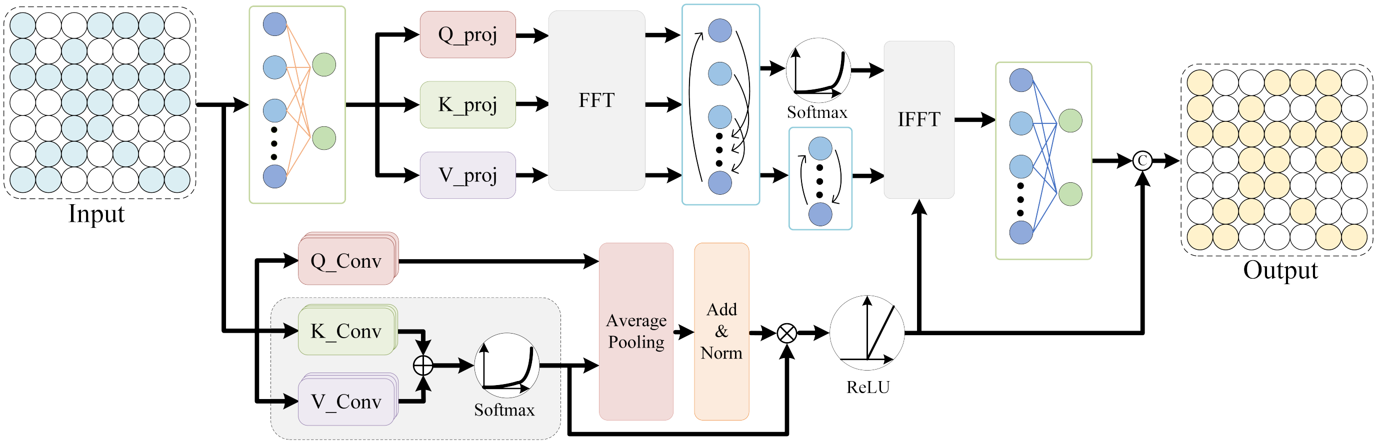


Fig. 3: Structure of the Frequency-Channel Attention (FCA) module. FFT and IFFT denote the Fast Fourier Transform and its inverse. $\oplus$ and $\otimes$ represent element-wise addition and multiplication, and ⓒ denotes concatenation.

3) Fusion: The outputs of the two branches are combined with the original features X via a gated residual mechanism:

$$X' = X + \beta_1\big(Z_{\text{ch}} - X\big) + \beta_2\big(Z_{\text{freq}} - X\big), \tag{12}$$

where β_1 and β_2 are learnable scalars. The refined sequence representation returned by the Frequency–Channel Attention Layer (FCA) is then obtained by reshaping X' back to the sequence form:

$$\text{FCA}(H) = \text{reshape}^{-1}(X') \in \mathbb{R}^{n\times d}. \tag{13}$$

This layer is later applied after each graph convolution layer and in the post-processing stage, providing consistent frequency–channel enhancement for the evolving node features.

C. Dynamic GCN Layer

Initialized with token representations $H^{(0)} \in \mathbb{R}^{n\times d}$ and the attention-based adjacency matrix $A^{(0)} \in \mathbb{R}^{n\times n}$, the Dynamic GCN Layer iteratively updates both node features and graph topology in steps L. Each layer l proceeds in three stages:

First, we pass the spatial message. The node features are aggregated through the current adjacency matrix and transformed through a shared linear layer with SELU activation:

$$\widetilde{H}^{(l)} = A^{(l-1)} H^{(l-1)}, \tag{14}$$

$$H^{(l)} = \sigma\Big(W_g\, \widetilde{H}^{(l)}\Big), \tag{15}$$

where W_g is a learnable weight matrix.

Second, to mitigate over-smoothing, we refine the representations using the FCA module. A learnable gate α_l dynamically fuses spatial features with frequency-enhanced features:

$$\widehat{H}^{(l)} = (1-\alpha_l)\, H^{(l)} + \alpha_l\, \text{FCA}(H^{(l)}). \tag{16}$$

The refined state $\widehat{H}^{(l)}$ serves as the input for both the next layer's propagation and the structural update module.

Third, we adaptively update the graph structure based on the refined node states. For each pair of nodes (i, j), the new edge weight is derived from the previous weight and the current endpoint features through a transformation function $\phi(\cdot)$:

$$G^{(l)}_{i,j} = \text{concat}\big(A^{(l-1)}_{i,j},\ \widehat{H}^{(l)}_i,\ \widehat{H}^{(l)}_j\big), \tag{17}$$

$$A^{(l)}_{i,j} = \phi\big(G^{(l)}_{i,j}\big). \tag{18}$$

This iterative co-evolution of features and topology allows the model to progressively strengthen aspect-relevant dependencies while pruning noisy connections. Finally, the output $H^{(L)}$ undergoes a final projection and FCA refinement to produce the representation of the pool.

D. Feature Fusion and Pooling

Upon concluding the propagation and refinement of the dynamic graph, we obtain the final token representation matrix $H \in \mathbb{R}^{n\times d}$, derived from the projected output $H^{(L)}$. To extract specific features of the target, we utilize a binary aspect mask $m \in \{0,1\}^n$ and perform mask-guided average pooling:

$$H_{\text{aspect}} = \frac{\sum_{i=1}^{n} H_i \cdot m_i}{\sum_{i=1}^{n} m_i}. \tag{19}$$

This operation aggregates token representations strictly within the aspect span, yielding a compact vector H_{aspect} that is subsequently fed into the output classifier.

E. Output Layer

The aspect-level sentiment prediction is generated by passing the pooled feature vector through a linear classifier followed by a softmax function:

$$p(a) = \text{Softmax}(W H_{\text{aspect}} + b), \tag{20}$$

where W and b are trainable parameters. The resulting probability distribution $p(a)$ corresponds to the sentiment polarity of the target aspect.

F. Training

The model is trained in a supervised manner using cross-entropy loss over all sentence–aspect pairs:

$$L(\theta) = -\sum_{(s,a)\in D}\sum_{c\in C}\log p(a), \tag{21}$$

where D denotes the training corpus, C is the sentiment label set, and θ represents all the learnable parameters. The objective encourages the model to assign a higher probability to the correct sentiment class for each aspect instance.

IV. Experiment

A. Setup

We performed our experiments on three widely used benchmark datasets for ABSA, including Restaurant and Laptop reviews from SemEval 2014 Task 4[2], and the Twitter dataset[3] [25], [26]. These datasets classify aspect sentiments into positive, negative, or neutral categories, with detailed statistics summarized in Table I.

TABLE I: Statistics of the experimental datasets.

Dataset	Division	Positive	Negative	Neutral
Laptop	Train	976	851	455
	Test	337	128	167
Restaurant	Train	2164	807	637
	Test	727	196	196
Twitter	Train	1507	1528	3016
	Test	172	169	336

All experiments are implemented within the PyTorch framework and are conducted on a single NVIDIA A40 GPU. To ensure reproducibility, the random seed is fixed at 1000. Syntactic dependency trees are generated using the Stanford CoreNLP toolkit[4] [27]. In terms of embeddings, the standard FCD-GCN utilizes 300-dimensional pre-trained GloVe vectors [28], while the FCD-GCN-BERT variant is initialized with the `bert-base-uncased` model[5]. A comprehensive summary of the specific hyper-parameter settings for both variants is provided in Table II.

B. Baselines

We also compare the model performance of the proposed approach with the strong baselines:

- **ATAE-LSTM** [13]: Attention-based LSTM that appends aspect embeddings to word embeddings.
- **IAN** [14]: Interactive Attention Network using two LSTMs to interactively learn aspect and context representations.
- **AOA** [15]: Attention-over-Attention mechanism for jointly modeling aspect-context interaction.

[2]https://alt.qcri.org/semeval2014/task4/
[3]https://github.com/songyouwei/ABSA-PyTorch
[4]https://stanfordnlp.github.io/CoreNLP
[5]https://github.com/huggingface/transformers

TABLE II: Hyper-parameter settings for FCD-GCN and FCD-GCN-BERT.

Parameter	FCD-GCN	FCD-GCN-BERT
Embedding Dimension	300	768
GCN Hidden Dimension	50	768
Optimizer	Adam	AdamW
Learning Rate	0.002	2e-5
Dropout Rate	0.7	0.3
Max Sequence Length	85	85
Random Seed	1000	1000

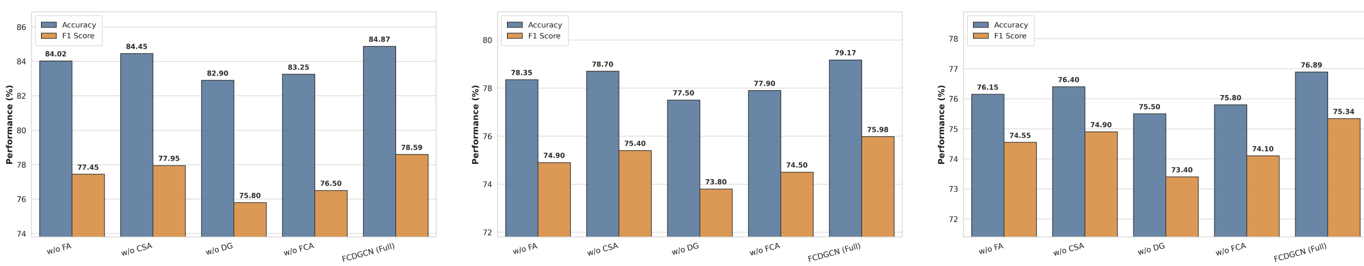
- **Sentic LSTM** [16]: Attentive LSTM incorporating SenticNet common sense knowledge for affective information.
- **TNet** [17]: Target-Specific Transformation Network using CNNs to generate aspect-specific representations without dependency parsing.
- **ASGCN** [9]: Pioneering GCN model applied over dependency trees for syntactic dependency capture in ABSA.
- **R-GAT** [10]: Graph Attention Network extended with relational information from dependency trees.
- **DGEDT** [11]: Integrates a dual-transformer structure and bidirectional GCNs to jointly model flat textual representations and dependency graph knowledge.
- **DualGCN** [18]: Dual-graph architecture with syntactic and semantic correlation graphs for complementary features.
- **SSEGCN** [19]: GCN enhanced with aspect-aware attention and syntactic mask matrices for semantic-syntactic fusion.
- **AGCN** [20]: Aggregated GCN leveraging multi-scale features and multi-layer aggregation.
- **SPGCN** [21]: Semantic-Preserving GCN with multi-graph perception and sentiment interaction.
- **SEGCN** [22]: Mechanism for storing, sharing, and transferring sentiment knowledge.
- **TextGT** [23]: Double-View Graph Transformer integrating graph-guided structural bias into the Transformer.
- **IDGNN** [24]: Iterative Dynamic GNN that updates the graph structure during training to capture evolving semantic dependencies.

C. Results

Table III presents a comprehensive comparison of the performance of the model. In the standard GloVe embedding setting, FCD-GCN exhibits superior performance against all baseline models across the three benchmark datasets. It notably surpasses competitive baselines such as TextGT and SSEGCN in terms of both Accuracy and F1 metrics, especially on the challenging Laptop and Twitter domains. This empirical evidence validates the efficacy of our dynamic graph construction and frequency-domain attention in mitigating noise and capturing long-range dependencies that remain elusive for static graph approaches.

TABLE III: Performance Comparison on Restaurant, Laptop, and Twitter Datasets. The best results are in **bold**, and the second-best results are underlined. "†" indicates results reproduced using open source code.

Model	Restaurant		Laptop		Twitter	
	Acc	F1	Acc	F1	Acc	F1
ATAE-LSTM (2016)	77.20	67.02	68.70	63.93	-	-
IAN (2017)	79.26	70.09	72.05	67.38	72.50	70.81
AOA (2018)	79.97	70.42	72.62	67.52	72.30	70.20
Sentic LSTM (2018)	79.43	70.38	70.88	67.19	70.66	67.87
TNet (2018)	80.69	71.27	76.54	71.75	74.90	73.60
ASGCN (2019)	80.77	72.02	75.55	71.05	72.15	70.40
R-GAT (2020)	83.30	76.08	77.42	73.76	75.57	73.82
DGEDT (2020)	83.90	75.10	76.80	72.30	74.80	73.40
DualGCN (2021)	84.27	78.08	78.48	74.74	75.92	74.29
AGCN (2022)	80.02	71.02	75.07	70.96	73.98	72.48
SPGCN (2022)	83.16	74.91	77.90	73.86	74.86	72.95
SSEGCN† (2022)	84.32	77.43	79.11	75.97	76.03	74.78
SEGCN (2023)	84.38	77.88	78.21	74.82	76.16	74.68
TextGT† (2024)	84.65	78.42	78.52	74.78	76.43	75.15
IDGNN (2024)	83.40	76.37	78.07	74.17	75.94	74.37
FCD-GCN (Ours)	**84.87**	**78.59**	**79.17**	**75.98**	**76.89**	**75.34**
AGCN+BERT (2022)	82.77	73.29	79.94	76.52	75.43	74.11
R-GAT+BERT (2020)	86.68	80.92	80.94	78.20	76.28	75.25
DualGCN+BERT (2022)	87.13	81.16	81.80	78.10	77.40	76.02
SSEGCN+BERT† (2022)	86.68	80.50	80.85	77.93	77.56	76.20
SEGCN+BERT (2023)	86.94	81.34	**82.76**	79.03	77.17	75.26
TextGT+BERT† (2024)	87.02	81.87	80.93	77.98	77.63	76.43
IDGNN+BERT (2024)	**87.25**	81.16	81.12	77.73	76.72	75.92
FCD-GCN+BERT (Ours)	87.20	**82.21**	81.78	**79.14**	**77.59**	**76.52**



(a) Restaurant (b) Laptop (c) Twitter

Fig. 4: Visual comparison of ablation study results. The bar charts illustrate the performance drop across three datasets when specific modules are removed, corresponding to the numerical data in Table IV.

The integration of pre-trained BERT embeddings yields substantial performance gains across all models. In this robust setting, FCD-GCN+BERT demonstrates remarkable stability and balance. Although obtaining the second-highest accuracy in the Restaurant and Laptop datasets, our model consistently secures the state-of-the-art F1 score in all domains, specifically reaching 82.21% in the Restaurant dataset and 76.52% on Twitter. These results confirm that our frequency-channel enhanced architecture functions orthogonally to pre-trained language models, effectively leveraging the rich semantic representations of BERT to maximize classification robustness.

D. Ablation Study

To validate the contribution of individual components within FCD-GCN, we conducted an ablation study on the Restaurant, Laptop, and Twitter datasets, with detailed results presented in Table IV and Figure 4. We evaluated variants by removing Frequency Attention, Channel Self-Attention, the Dynamic Graph mechanism, and the entire FCA module.

TABLE IV: Ablation Study of FCD-GCN. "w/o" denotes removing a specific module. **Bold** indicates the best performance.

Model	Restaurant		Laptop		Twitter	
	Acc	F1	Acc	F1	Acc	F1
w/o Frequency Attn	84.02	77.45	78.35	74.90	76.15	74.55
w/o Channel Attn	84.45	77.95	78.70	75.40	76.40	74.90
w/o Dynamic Graph	82.90	75.80	77.50	73.80	75.50	73.40
w/o FCA Module	83.25	76.50	77.90	74.50	75.80	74.10
FCD-GCN (Full)	**84.87**	**78.59**	**79.17**	**75.98**	**76.89**	**75.34**

The full model consistently outperforms all the variants ablated. Most notably, removing the Dynamic Graph mechanism induces the sharpest performance decline, exemplified by a 2.79% drop in F1 score on the Restaurant dataset. This

TABLE V: Case Study comparison on real examples. 😊, 🙁, and 😐denote Positive, Negative, and Neutral predictions, respectively. ✓ indicates correct prediction; ✗ indicates incorrect prediction.

Review Sentences with Targets	R-GAT	DualGCN	SSEGCN	TextGT	Ours
1. If you are a Tequila fan you will not be disappointed.	🙁✗	🙁✗	😊✓	🙁✗	😊✓
2. The falafal was rather over cooked..., but the chicken was fine.	🙁✓, 🙁✗	🙁✓, 😐✗	🙁✓, 🙁✗	🙁✓, 😊✓	🙁✓, 😊✓
3. Windows 8 was not to my liking but the touch screen ... were cool.	🙁✓, 🙁✗	🙁✓, 😊✓	🙁✓, 😊✓	🙁✓, 😊✓	🙁✓, 😊✓
4. I love my iPhone but hate the AT&T service and battery.	😊✓, 😊✗, 🙁✓	😊✓, 🙁✓, 😐✗	😊✓, 🙁✓, 🙁✓	😊✓, 🙁✓, 🙁✓	😊✓, 🙁✓, 🙁✓

empirical evidence underscores the critical role of layer-wise graph refinement. Furthermore, the performance degradation observed upon removal of frequency attention and Channel Self-Attention validates the necessity of spectral modeling and channel-wise reweighting.

E. Effect of Activation Functions

To identify the optimal non-linearity for the Frequency-Channel Attention module, we compare different activation functions as shown in Table VI. Among the variants, ReLU consistently achieves the best performance in all datasets. This superiority is attributed to ReLU's ability to induce sparsity and mitigate the vanishing gradient problem in high-dimensional spectral features. Although PReLU and Tanh offer competitive results, Sigmoid underperforms due to gradient saturation, hindering effective convergence. Consequently, ReLU is adopted as the default activation.

TABLE VI: Effect of Activation Functions in the FCA Module. **Bold** indicates the best performance.

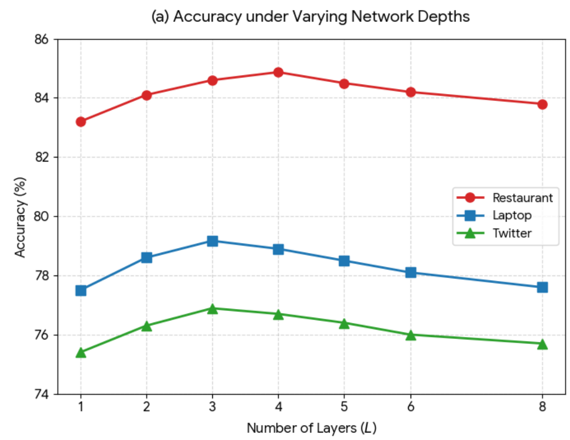
Function	Restaurant		Laptop		Twitter	
	Acc	F1	Acc	F1	Acc	F1
Tanh	84.07	77.52	78.21	75.07	75.96	74.58
PReLU	84.36	77.93	78.52	75.34	76.13	74.85
SELU	83.94	77.41	78.04	74.92	75.77	74.36
Sigmoid	83.76	76.80	77.95	74.85	75.32	73.96
ReLU	**84.87**	**78.59**	**79.17**	**75.98**	**76.89**	**75.34**

F. Case Study

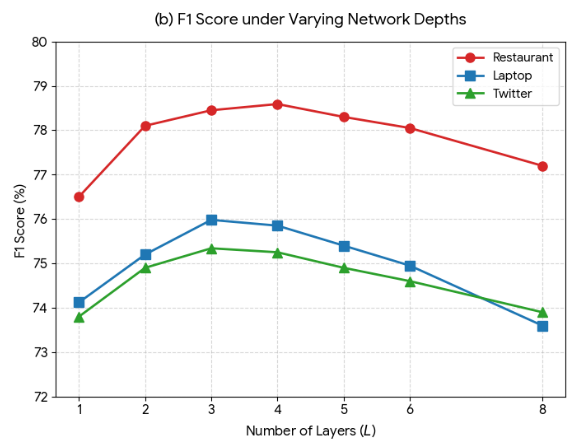
To evaluate the capability of FCD-GCN in resolving complex semantic patterns, we analyze representative test samples, comparing our model with strong baselines in Table V.

Cases 1 and 3 involve negation and conditional logic. In Case 1 ("not be disappointed"), baselines like R-GAT often misinterpret negative keywords as negative sentiment, while FCD-GCN correctly resolves the double negation to predict Positive. This is attributed to the Frequency Attention mechanism, which captures linguistic inversions in the spectral domain. Similarly, in Case 3, FCD-GCN accurately detects polarity shifts, overcoming limitations of rigid static graph structures.

Cases 2 and 4 examine mixed sentiment disentanglement in multi-aspect sentences. Case 2 presents a structural imbalance where a long negative clause precedes a short positive one. Traditional graph models often suffer from sentiment bleeding,



(a)



(b)

Fig. 5: Impact of network depth (L) on (a) Accuracy and (b) F1 Score across three datasets. FCD-GCN exhibits stable performance even at deeper layers without significant degradation.

incorrectly propagating negative sentiment to the positive aspect, whereas FCD-GCN effectively isolates aspect-specific sentiments through dynamic structural refinement.

G. Impact of GCN Layers

To investigate the impact of network depth on model performance, we evaluated FCD-GCN on varying numbers of layers ($L \in \{1, \ldots, 8\}$). As illustrated in Figure 5, the model effectively aggregates contextual information, reaching a peak of 4 layers for the well-structured Restaurant dataset and 3 layers for the noisy Laptop and Twitter domains. In particular, unlike standard GCNs that often suffer from rapid performance degradation in deeper architectures, FCD-GCN exhibits superior stability, maintaining competitive results even at $L = 8$. This robustness confirms that our Frequency Attention preserves feature distinctiveness in the spectral domain, while the Dynamic Graph refinement actively mitigates the accumulation of structural noise as the network deepens.

V. Conclusion

In this paper, we propose FCD-GCN, a unified framework addressing static dependency parsing limitations in aspect-based sentiment analysis. Integrating dynamic graph refinement and Frequency-Channel Attention, our model adapts to diverse semantic dependencies while capturing global patterns and discriminative spectral signals. Evaluated on three benchmarks, FCD-GCN surpasses state-of-the-art baselines, verifying its superiority in disentangling complex sentiment interactions. Future work will optimize the dynamic module for real-time deployment and extend the spectral paradigm to more graph-based NLP tasks.

Acknowledgment

We sincerely thank our collaborators for their valuable support and anonymous reviewers for their constructive comments and suggestions on this work. This work was supported by the National Natural Science Foundation of China (Grant No.62406316, 201704041014, and 201404041254), the Xinjiang 'Tianchi Talent' Recruitment and Introduction Program (awarded to Mieradilijiang Maimaiti).

References

[1] W. Zhang, X. Li, Y. Deng, L. Bing, and W. Lam, "A survey on aspect-based sentiment analysis: Tasks, methods, and challenges," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 11, pp. 11 019–11 038, 2022.

[2] A. Nazir, Y. Rao, L. Wu, and L. Sun, "Issues and challenges of aspect-based sentiment analysis: A comprehensive survey," *IEEE Transactions on Affective Computing*, vol. 13, no. 2, pp. 845–863, 2020.

[3] Y. C. Hua, P. Denny, J. Wicker, and K. Taskova, "A systematic review of aspect-based sentiment analysis: domains, methods, and trends," *Artificial Intelligence Review*, vol. 57, no. 11, p. 296, 2024.

[4] G. Brauwers and F. Frasincar, "A survey on aspect-based sentiment classification," *ACM Computing Surveys*, vol. 55, no. 4, pp. 1–37, 2022.

[5] W. Zhang, X. Li, Y. Deng, L. Bing, and W. Lam, "A survey on aspect-based sentiment analysis: Tasks, methods, and challenges," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 11, pp. 11 019–11 038, 2022.

[6] D. Nath and S. K. Dwivedi, "Aspect-based sentiment analysis: approaches, applications, challenges and trends," *Knowledge and Information Systems*, vol. 66, no. 12, pp. 7261–7303, 2024.

[7] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 1422–1432.

[8] H. Liu, C. Xu, and J. Liang, "Dependency distance: A new perspective on syntactic patterns in natural languages," *Physics of life reviews*, vol. 21, pp. 171–193, 2017.

[9] C. Zhang, Q. Li, and D. Song, "Aspect-based sentiment classification with aspect-specific graph convolutional networks," *arXiv preprint arXiv:1909.03477*, 2019.

[10] K. Wang, W. Shen, Y. Yang, X. Quan, and R. Wang, "Relational graph attention network for aspect-based sentiment analysis," *arXiv preprint arXiv:2004.12362*, 2020.

[11] H. Tang, D. Ji, C. Li, and Q. Zhou, "Dependency graph enhanced dual-transformer structure for aspect-based sentiment classification," in *Proceedings of the 58th annual meeting of the association for computational linguistics*, 2020, pp. 6578–6588.

[12] J. Wang, B. Xu, and Y. Zu, "Deep learning for aspect-based sentiment analysis," in *2021 International conference on machine learning and intelligent systems engineering (MLISE)*. IEEE, 2021, pp. 267–271.

[13] Y. Wang, M. Huang, X. Zhu, and L. Zhao, "Attention-based lstm for aspect-level sentiment classification," in *Proceedings of the 2016 conference on empirical methods in natural language processing*, 2016, pp. 606–615.

[14] D. Ma, S. Li, X. Zhang, and H. Wang, "Interactive attention networks for aspect-level sentiment classification," *arXiv preprint arXiv:1709.00893*, 2017.

[15] B. Huang, Y. Ou, and K. M. Carley, "Aspect level sentiment classification with attention-over-attention neural networks," in *Social, Cultural, and Behavioral Modeling: 11th International Conference, SBP-BRiMS 2018, Washington, DC, USA, July 10-13, 2018, Proceedings 11*. Springer, 2018, pp. 197–206.

[16] Y. Ma, H. Peng, and E. Cambria, "Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[17] X. Li, L. Bing, W. Lam, and B. Shi, "Transformation networks for target-oriented sentiment classification," *arXiv preprint arXiv:1805.01086*, 2018.

[18] R. Li, H. Chen, F. Feng, Z. Ma, X. Wang, and E. Hovy, "Dual graph convolutional networks for aspect-based sentiment analysis," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 6319–6329.

[19] Z. Zhang, Z. Zhou, and Y. Wang, "Ssegcn: Syntactic and semantic enhanced graph convolutional network for aspect-based sentiment analysis," in *Proceedings of the 2022 conference of the North American Chapter of the association for computational linguistics: human language technologies*, 2022, pp. 4916–4925.

[20] M. Zhao, J. Yang, J. Zhang, and S. Wang, "Aggregated graph convolutional networks for aspect-based sentiment classification," *Information Sciences*, vol. 600, pp. 73–93, 2022.

[21] Q. Lu, X. Sun, R. Sutcliffe, Y. Xing, and H. Zhang, "Sentiment interaction and multi-graph perception with graph convolutional networks for aspect-based sentiment analysis," *Knowledge-Based Systems*, vol. 256, p. 109840, 2022.

[22] Y. Zheng, X. Li, and J.-Y. Nie, "Store, share and transfer: Learning and updating sentiment knowledge for aspect-based sentiment analysis," *Information Sciences*, vol. 635, pp. 151–168, 2023.

[23] S. Yin and G. Zhong, "Textgt: A double-view graph transformer on text for aspect-based sentiment analysis," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 17, 2024, pp. 19 404–19 412.

[24] P. Wang, L. Tao, M. Tang, L. Wang, Y. Xu, and M. Zhao, "Incorporating syntax and semantics with dual graph neural networks for aspect-level sentiment analysis," *Engineering Applications of Artificial Intelligence*, vol. 133, p. 108101, 2024.

[25] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar, "SemEval-2014 task 4: Aspect based sentiment analysis," in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, P. Nakov and T. Zesch, Eds. Dublin, Ireland: Association for Computational Linguistics, Aug. 2014, pp. 27–35. [Online]. Available: https://aclanthology.org/S14-2004

[26] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, and K. Xu, "Adaptive recursive neural network for target-dependent twitter sentiment classification," in *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 2: Short papers)*, 2014, pp. 49–54.

[27] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit," in *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, 2014, pp. 55–60.

[28] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.