Probabilistic Syntax-Aware Joint Span–Sentiment Learning for Aspect-Based Sentiment Analysis

Yuxun Wang and Dunhui Yu(✉)

[1] School of Computer Science, Hubei University, Wuhan 430062, China
yumhy@hubu.edu.cn

Abstract. Aspect-Based Sentiment Analysis (ABSA) aims to identify aspect terms in text and predict their sentiment polarities. The two subtasks are tightly coupled: inaccurate aspect boundaries can mislead sentiment prediction, while sentiment supervision should in turn refine span selection. However, existing methods still suffer from two limitations. First, although boundary detection and sentiment classification are interdependent, gradients between them are often weak, resulting in error propagation, especially in pipeline settings. Second, under complex syntactic phenomena such as negation and contrast, boundary shifts and the mismatch between training on gold spans and inferring on predicted spans cause a train–inference discrepancy and unstable optimization. To address these issues, we propose a Probabilistic Syntax-Aware Joint Span–Sentiment Learning (PSJL) approach for ABSA. PSJL relaxes discrete span selection into a differentiable span distribution and dynamically incorporates syntactic constraints for end-to-end optimization. Specifically, (i) Probabilistic Joint Modeling (PJM) computes expectation-based span representations from a span distribution, enabling sentiment loss to backpropagate to boundary prediction. (ii) Syntax-Constrained Contrastive Learning (SCCL) derives span-level masks from dependency parse trees to contrast syntactically plausible and implausible candidates, regularizing boundary decisions. (iii) Adaptive Training Strategy (ATS) adaptively balances supervision from gold and predicted spans based on a syntax-consistency rate, mitigating train–inference discrepancy and improving training stability. Experiments on three benchmark datasets show that PSJL consistently outperforms state-of-the-art models and strong LLM-based baselines on aspect term extraction and sentiment classification.

Keywords: Aspect-Based Sentiment Analysis, Joint Span–Sentiment Learning, Probabilistic Modeling, Syntax-Aware Learning, Contrastive Learning, Curriculum Learning

1 Introduction

Aspect-Based Sentiment Analysis (ABSA) extracts aspect terms from text and predicts their sentiment polarities [1-3]. Its goal goes beyond assigning a single label to an entire sentence, because sentiment must be grounded on specific targets. In **Fig. 1**, the sentence “The battery is great but the screen is terrible” requires identifying two targets,

"battery" and "screen", and assigning opposite polarities to them. A key difficulty is that boundary detection and sentiment classification depend on each other: contextual sentiment cues help decide target spans, and span accuracy is essential for reliable polarity prediction.

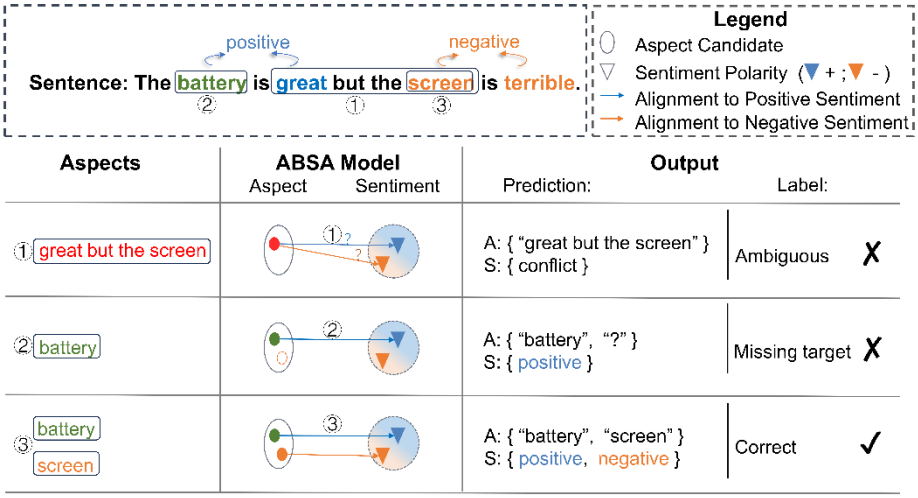

Fig. 1. Illustration of ABSA's core challenge: the coupling between aspect boundary detection and sentiment classification. ① Incorrect span boundaries lead to ambiguous sentiment; ② Missing targets yield incomplete predictions; ③ Desired output with precise aspect scopes and correct sentiment alignment.

Early methods mainly adopted a pipeline architecture, decomposing the task into two independent steps: first identifying candidate aspect terms through sequence labeling or span detection [4-7], and then performing sentiment classification on each candidate aspect term [6], [8-10]. Although this modular design is easy to implement, it severs the dependency between the two subtasks. Specifically, the boundary detection module is optimized based on the cross-entropy loss of boundary labels, while the sentiment classification module is trained independently on gold spans [6], [7], [11], [12]. This separate optimization easily leads to two problems: (1) errors in boundary detection are propagated to the sentiment classification stage; (2) loss signals from sentiment classification are difficult to backpropagate to the boundary detection parameters, making the model unable to fully exploit sentiment information to correct boundary selection [7], [11-13]. Motivated by these limitations, recent research has increasingly shifted toward end-to-end joint modeling to better couple boundary detection and sentiment classification, and has reported encouraging progress [7], [11-13]. A large number of studies use graph neural networks to propagate information on dependency parse trees [14-17], or achieve cross-task knowledge transfer through multi-task learning [11-13], [18]. These methods realize the collaborative optimization of the two subtasks to some extent through shared encoders and interactive attention mechanisms [12], [19-21].

However, existing joint modeling methods still have room for improvement in two key aspects. First, most of them essentially retain a pipeline-style training paradigm: boundary detection and sentiment classification are optimized with their respective loss functions [13], and sentiment classifiers are trained on gold spans constructed offline [6], [7], [11]. This discrete coupling makes it difficult for the gradients of the sentiment classifier to be effectively propagated back to the boundary detection parameters, easily causing limited gradient propagation and even gradient blocking between the two subtasks [11-13]. Second, existing methods generally suffer from a train–inference discrepancy: they rely on gold spans for sentiment classification during training but use predicted spans during inference, making it difficult for the model to fully adapt to boundary errors at test time [6], [7], [11], [13]. Meanwhile, purely data-driven implicit alignment is easily limited when dealing with complex linguistic structures [14], [17], [19], [20]. For example, in sentences containing negation structures such as "I do not like the decoration here" or contrastive relations such as "Although the price is high, the quality is good", the model tends to include function words or incomplete modifiers in the scope of aspect terms, leading to boundary misalignment and unstable training behavior [5], [6], [22], [23]. Recent studies have also explored span-level and contrastive formulations for ABSA and related sentiment extraction tasks. Zhang et al. proposed a table-filling framework for span-level ABSA, which models aspect-sentiment structures through pairwise token interactions [33]. Qiu et al. modeled inter-aspect relations with clause-level structure and contrastive learning to better capture dependencies among multiple aspects [34]. Wu et al. further introduced multi-view contrastive learning for sentiment triplet extraction, showing the usefulness of contrastive objectives in structured sentiment prediction [35]. Different from these studies, PSJL focuses on span-level train–inference discrepancy by combining probabilistic span scoring, syntax-guided contrastive regularization, and adaptive predicted-span supervision within a unified span–sentiment framework.

To address these challenges, we propose a novel **Probabilistic Syntax-Aware Joint Span–Sentiment Learning (PSJL)** framework. PSJL tackles the above issues through three complementary modules: (1) the **Probabilistic Joint Modeling (PJM)** module softens discrete span selection into a continuous probability distribution and uses an expectation-based span representation, enabling the sentiment loss to be propagated back to boundary detection parameters and thus achieving end-to-end joint optimization of boundaries and sentiment; (2) the **Syntax-Constrained Contrastive Learning (SCCL)** module uses dependency parse trees to construct structured masks that guide contrastive learning, constraining the linguistic rationality of candidate spans and reducing boundary shifts under complex syntactic constructions; (3) the **Adaptive Training Strategy (ATS)** dynamically adjusts the weights of gold spans and predicted-span supervision based on syntax-consistency rate, allowing the model to gradually adapt to the error distribution patterns in the inference stage and alleviating training instability caused by the train–inference discrepancy. Experiments show that the F1 scores of PSJL on the Laptop, Restaurant, and Tweets datasets are improved by 3.9%, 3.3%, and 8.2%, respectively, compared with existing state-of-the-art baselines.

Our main contributions are summarized as follows:

- We propose a probabilistic joint sentiment-span modeling mechanism within the **PSJL** framework, which establishes a differentiable joint distribution between boundary detection and sentiment classification through expectation-based span representations, realizing effective feedback of sentiment loss on boundary selection and mitigating the problem of limited gradient propagation.
- We design a syntax-constrained contrastive learning module that uses dependency parse trees to construct structured masks for sampling positive and negative pairs, constraining the linguistic rationality of candidate spans and improving robustness to boundary shifts under complex syntactic structures.
- We introduce an adaptive training strategy that dynamically balances the gold-span loss and the predicted-span loss through a curriculum driven by syntax-consistency rate, alleviating the train-inference discrepancy and training instability.
- We conduct extensive experiments on three benchmark datasets and empirically demonstrate state-of-the-art performance for both aspect term extraction and sentiment classification. Code is available at https://github.com/sentiment-anon/text-sentiment.

2 Method

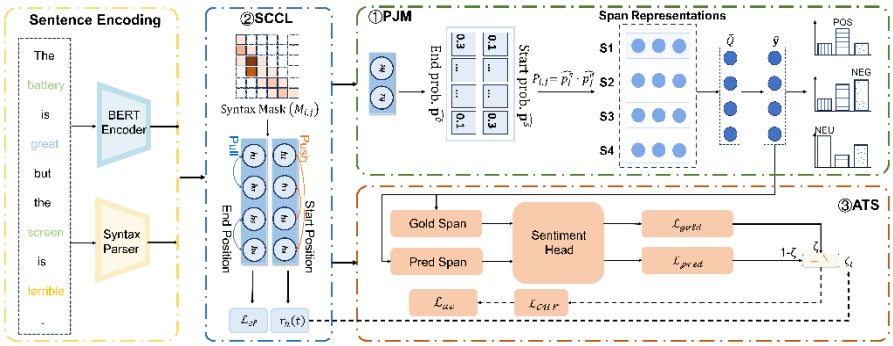

Fig. 2. Overall framework of the proposed PSJL model. PJM predicts start/end probabilities $\hat{p}_s, \hat{p}_e$ 3 and forms span scores $P_{i,j} = \hat{p}_i^s \hat{p}_j^e$ 4; a syntax mask $M_{i,j}$ induces a multiplicative bias $Q_{i,j} = P_{i,j}(1 + \omega M_{i,j})$ 5, and candidate spans are obtained by thresholding and top-K pruning on Q: $\mathcal{C} = TopK(\{(i,j) \mid (i,j) \in \mathcal{S}_{h_{train}}, Q_{i,j} \geq \delta\}, \mathrm{K})$ 6. For each $(i,j) \in C$, we compute a span representation $z_{i,j}$ (10) and the sentiment head outputs $\hat{y}_{i,j} = softmax\left(f_\theta(z_{i,j})\right)$; a soft predicted-span representation $\bar{z}_b = \Sigma_{(i,j)\in C} Q_{i,j} z_{i,j}$ (13) enables differentiable predicted-span sentiment supervision L_pred 14. SCCL applies dependency-guided contrastive regularization 16 and computes the syntax-consistency rate $r_h(t)$ (17). ATS uses $r_h(t)$ to update the curriculum weight ζ_t 19 and mixes $\mathcal{L}_{gold}$ and $\mathcal{L}_{pred}$ (18).

To address limited gradient propagation between boundary detection and sentiment classification, as well as the train–inference discrepancy caused by training on gold spans but inferring on predicted spans, we propose **PSJL** (**P**robabilistic **S**yntax-Aware **J**oint Span–Sentiment **L**earning). PSJL is a closed-loop framework that integrates three

components: Probabilistic Joint Modeling (**PJM**), Syntax-Constrained Contrastive Learning (**SCCL**), and Adaptive Training Strategy (**ATS**), as shown in **Fig. 2**.

2.1 Problem Definition and Notation

Given a sentence $x = (w_1, \dots, w_n)$ of length n, ABSA aims to extract an aspect span (s, e) and predict its sentiment polarity $y \in \{1, \dots, C\}$. We follow an instance-wise training formulation: each gold annotation (s^*, e^*, y) in x forms one training instance (multiple aspects yield multiple instances sharing the same x).

BERT encoding. We encode x with a BERT encoder to obtain contextual token representations $H = [h_1; \dots; h_n] \in \mathbb{R}^{\mathbb{n} \times \mathbb{d}}$, where $h_i \in \mathbb{R}^{\mathbb{d}}$.

Candidate spans. We use two span-length bounds: h_{train}. for training (`max_span_train_len`) and h_{test}. for inference (`max_answer_length`). The candidate span set under bound h is

$$S_h = \{(i, j) \mid 1 \le i \le j \le n, j - i + 1 \le h\} \tag{1}$$

Dependency mask. Let $G = (\mathcal{V}, \mathcal{E})$ be the dependency tree of x over tokens $\mathcal{V} = \{1, \dots, n\}$. With dependency radius r (`dep_radius`), we define

$$M_{i,j} = \mathbb{I}[dist_G(i, j) \le r], \tag{2}$$

where $\mathrm{dist}_G(i, j)$ is the shortest-path length on G and $\mathbb{I}[\cdot]$ is the indicator function.

2.2 Probabilistic Joint Modeling (PJM)

Discrete span selection weakens gradient flow from sentiment supervision to boundary parameters. PJM relaxes span selection into differentiable probabilities and couples boundary and sentiment learning via an expected-span representation. Specifically, $\widehat{\boldsymbol{p}^s}, \widehat{\boldsymbol{p}^e}$ induce span scores $P_{i,j}$, which are syntax-biased into $Q_{i,j}$ to form candidates C; normalizing Q over C yields $\bar{z}$ for differentiable predicted-span sentiment learning.

Span scoring. We compute start/end distributions:

$$\widehat{\mathbf{p}^s} = \mathrm{softmax}(\mathbf{H}\mathbf{w_s}), \widehat{\mathbf{p}^e} = \mathrm{softmax}(\mathbf{H}\mathbf{w_e}), \tag{3}$$

where $w_s, w_e \in \mathbb{R}^{\mathbb{d}}$ are learnable parameters. For $(i, j) \in \mathcal{S}_{h_{train}}$, we score spans by

$$P_{i,j} = \widehat{p_i^s} \cdot \widehat{p_j^e}. \tag{4}$$

Syntax-biased candidates. We bias span scores with $M_{i,j}$ 2:

$$Q_{i,j} = P_{i,j}\big(1 + \omega M_{i,j}\big), (i, j) \in S_{h_{train}} \tag{5}$$

We retain candidates by thresholding and top-K pruning:

$$C = TopK\big(\{(i, j) \in S_{h_{train}} \mid Q_{i,j} \ge \delta\}, K\big) \tag{6}$$

Differentiability clarification. Thresholding and top-K pruning in Eq. 6 are non-differentiable operations used for tractable candidate selection. The differentiable learning signal is propagated through probabilistic span scoring and the expected-span representation over the retained candidates in Eqs. 12–14.

Here ω is the syntax bias weight (`dep_weight`), and δ and K correspond to `logit_threshold` and `n_best_size`, respectively. Note that $P_{(i,j)}$ is a probability product, and the multiplicative form keeps $Q_{(i,j)}$ on a comparable scale, so δ remains effective for filtering low-confidence spans.

Expected-span joint training. For a mini-batch of B instances indexed by b, each instance has gold boundaries (s_b^*, e_b^*) and sentiment label y_b. We supervise boundaries with start/end negative log-likelihood:

$$\mathcal{L}_{start} = -\frac{1}{B}\sum_{b=1}^{B} \log \widehat{p^{s}_{b,s_b^*}}, \mathcal{L}_{end} = -\frac{1}{B}\sum_{b=1}^{B} \log \widehat{p^{e}_{b,e_b^*}}, \tag{7}$$

and a span-level loss by normalizing span scores over $\mathcal{S}_{h_{train}}$:

$$\widetilde{P_{i,j}} = \frac{P_{i,j}}{\sum_{(u,v)\in S_{h_{train}}} P_{u,v}}, L_{spn} = -\frac{1}{B}\sum_{b=1}^{B} \log \widetilde{P_{b,s_b^*,e_b^*}} \tag{8}$$

We combine them as

$$\mathcal{L}_{ae} = \lambda_s \mathcal{L}_{start} + \lambda_e \mathcal{L}_{end} + \lambda_{spn} \mathcal{L}_{spn}, \tag{9}$$

where λ_s , λ_e , λ_{spn} are loss weights (corresponding to `weight_start`, `weight_end`, and `weight_span`).

For sentiment, we compute span representations

$$\mathbf{z_{i,j}} = \frac{1}{j-i+1}\sum_{u=i}^{j} \mathbf{h_u}, \tag{10}$$

where $\boldsymbol{z_{i,j}} \in \mathbb{R}^{\mathbb{d}}$. Let f_θ be the sentiment classifier and define $\widehat{y_{i,j}} = \mathrm{softmax}\left(f_\theta(z_{i,j})\right) \in \mathbb{R}^{\mathbb{C}}$. For a mini-batch of size B indexed by b, $\mathrm{CE}_\epsilon(\cdot)$ denotes label-smoothed cross-entropy with smoothing factor ϵ. We train with gold-span supervision

$$\mathcal{L}_{gold} = \frac{1}{B}\sum_{b=1}^{B} \mathrm{CE}_\epsilon\left(\widehat{\boldsymbol{y}_{s_b^*,e_b^*}}, y_b\right), \tag{11}$$

as well as a differentiable predicted-span loss by normalizing $Q_{(i,j)}$ over $\mathcal{C}$:

$$\widetilde{Q_{i,j}} = \frac{\exp(Q_{i,j})}{\sum_{(u,v)\in\mathcal{C}} \exp(Q_{u,v})}, (i,j) \in \mathcal{C}, \tag{12}$$

$$\bar{z}_b = \sum_{(i,j)\in C} \widetilde{Q_{i,j}}\, z_{i,j} \tag{13}$$

$$\mathcal{L}_{pred} = \frac{1}{B}\sum_{b=1}^{B} \mathrm{CE}_{\epsilon}\big(\mathrm{softmax}\big(f_\theta(\overline{\mathbf{z}_b})\big), y_b\big). \tag{14}$$

This makes sentiment supervision differentiable w.r.t. $Q_{(i,j)}$ and thus w.r.t. $\widehat{p^s}, \widehat{p^e}$ in 3.

2.3 Syntax-Constrained Contrastive Learning (SCCL)

While PJM enables end-to-end coupling, boundary predictions can still drift to syntactically implausible spans under negation or contrast. SCCL regularizes boundary pairing with dependency-guided contrastive learning and outputs a syntax-consistency signal to drive ATS.

Dependency Scope and Negative Sampling. We reuse the dependency proximity indicator $M_{i,j}$ in 2 and regard a boundary pair (i, j) as syntactically plausible if $M_{i,j}$ = 1. For each instance b with gold boundaries (s_b^*, e_b^*), we treat (s_b^*, e_b^*) as the positive pair. We construct negatives by fixing the start boundary and sampling ends outside its mask:

$$\mathcal{N}(s_b^*) = \{j \in \mathcal{V} \mid M_{s_b^*,j} = 0\}, \tag{15}$$

We then draw a subset $\widetilde{\mathcal{N}}(s_b^*) \subseteq \mathcal{N}(s_b^*)$ for efficiency.

Syntax-Constrained Contrastive Loss. We apply a projection head $g(\cdot)$ to obtain $u_i = g(h_i)$ and optimize an InfoNCE-style objective:

$$L_{cl} = -\frac{1}{B}\sum_{b=1}^{B} \log\left(\frac{exp\left(sim\left(u_{s_b^*}, u_{e_b^*}\right)\right)}{exp\left(sim\left(u_{s_b^*}, u_{e_b^*}\right)\right) + \sum_{j\in\widetilde{N}(s_b^*)} exp\left(sim\left(u_{s_b^*}, u_j\right)\right)}\right). \tag{16}$$

Here $sim(\cdot,\cdot)$ denotes cosine similarity. We weight this term by λ (`cl_weight`) in the final objective 20 [12].

Syntax-consistency signal. At step t, we take $\widehat{s_b} = \arg\max_i \widehat{p^s_{b,i}}$ and $\widehat{e_b} = \arg\max_j \widehat{p^e_{b,j}}$ and compute

$$r_h(t) = \frac{1}{B}\sum_{b=1}^{B} \mathbb{I}\big[M_{\widehat{s_b},\widehat{e_b}} = 1\big] \tag{17}$$

$r_h(t)$ is the `syntax-consistency rate`, i.e., the fraction of predicted boundary pairs that satisfy the dependency constraint. We use $r_h(t)$ as the syntax-consistency feedback to update the curriculum weight ζ_t in ATS.

2.4 Adaptive Training Strategy (ATS)

Training sentiment on gold spans is stable but creates a train–inference mismatch, whereas using predicted spans too early is noisy. ATS mitigates this issue by mixing gold-span and predicted-span supervision with an adaptive curriculum weight ζ_t driven by syntax consistency.

Curriculum mixing. We combine the two sentiment losses as follows:

$$\mathcal{L}_{cur}(t) = w_{ac}\big(\zeta_t \mathcal{L}_{gold} + (1-\zeta_t)\mathcal{L}_{pred}\big), \tag{18}$$

where w_{ac} is `weight_ac` and $\zeta_t \in [0,1]$ controls the gold/predicted supervision ratio.

Feedback-driven update. Given the target syntax-consistency level ξ (`scope_target`), we update ζ_t using $r_h(t)$:

$$\zeta_{t+1} = (1-\alpha)\zeta_t + \alpha \cdot clip\left(1 - \frac{r_h(t)}{\xi}, 0,1\right) \tag{19}$$

where α is `scope_adapt_alpha` and $\text{clip}(x, 0,1) = \min(\max(x, 0), 1)$.

Training and inference. The final objective is

$$\mathcal{L}(t) = \mathcal{L}_{ae} + \mathcal{L}_{cur}(t) + \lambda \mathcal{L}_{cl}, \tag{20}$$

where λ is `cl_weight` and L_{ae} is the boundary loss defined in PJM. At inference, we enumerate spans in $\mathcal{S}_{h_{test}}$, form C by thresholding (δ) and top-K pruning, and predict sentiment for each $(i, j) \in C$.

3 Experiments

3.1 Datasets and Evaluation Metrics

Experiments are carried out on three established ABSA benchmarks: the SemEval Laptop and Restaurant domains [1-3]and the Tweets dataset [10]. Each corpus provides sentence-level annotations of aspect spans together with sentiment polarities. Summary statistics are given in **Table 2**.

Metrics are computed under the SemEval evaluation scheme [1-3]. For AE and TSA, we report Precision, Recall, and F1 based on an exact boundary match of predicted spans. For TSA specifically, a hit requires agreement with the gold label on both the span boundaries and the polarity. For SP, we measure sentiment accuracy on the subset of correctly extracted aspects [6], [7], defined as $Acc_{SP} = TP_{SP}/\text{T}P_{AE}$.

3.2 Experimental Setup

PSJL is built with PyTorch (v1.11.0) and all runs are executed on a single NVIDIA RTX 4090. To mitigate variance from random initialization, we repeat each experiment three times with different seeds and report the mean score. **Significance test**. Following

the three-run setting, we use a paired t-test and tag our method as significant when it outperforms the best baseline at $p \leq 0.05$.

3.2.1 Implementation Details.

We use BERT-Large (uncased) as the backbone encoder and cap the input length at 85 tokens. Training uses Adam with $(\beta 1, \beta 2)$ = (0.9, 0.999) and a learning rate of 2 × 10^-5, together with a 0.1 warmup proportion. Models are trained for 80 epochs with batch size 16. Dropout is set to 0.1 to regularize training.

3.2.2 Loss Weights.

The loss weight configuration is as follows: `weight_start` = 1.0, `weight_end` = 1.0, and `weight_span` = 0.05. The sentiment classification loss weight is searched in `weight_ac` ∈ {1.0, 1.2, 1.4, 1.6, 1.8}.

Table 1. We report overall performance on three real-world datasets. "†" indicates numbers taken from prior work; "‡" indicates reproduced results. Bold denotes the best score. "*" marks significance over the best baseline with $p < 0.05$.

Model		Laptop			Restaurant			Tweets		
		Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Pipe-line	CRF†	0.5969	0.4754	0.5293	0.5228	0.5101	0.5164	0.4297	0.2521	0.3173
	NN-CRF†	0.5772	0.4932	0.5319	0.6009	0.6193	0.6100	0.4371	0.3712	0.4006
	TAG†	0.6584	0.6719	0.6651	0.7166	0.7645	0.7398	0.5424	0.5437	0.5426
	SPAN†	0.6946	0.6672	0.6806	0.7614	0.7334	0.7492	0.6072	0.5502	0.5769
End-to-End	SPJM†	0.6140	0.5820	0.5976	0.7620	0.6820	0.7198	0.5484	0.4844	0.5144
	SPAN-Joint†	0.6741	0.6199	0.6459	0.7232	0.7261	0.7247	0.5703	0.5269	0.5455
	S-AESC†	0.6687	0.6492	0.6588	0.7826	0.7050	0.7418	0.5586	0.5374	0.5473
	HI-ASA†	0.6796	0.6625	0.6709	0.7915	0.7621	0.7765	0.5732	0.5622	0.5676
	DCS†	0.6812	0.6640	0.6725	0.7835	0.7751	0.7793	0.5862	0.5835	0.5848
	MiniConGTS†	0.7206	0.6725	0.6957	0.7926	0.7952	0.7939	0.6164	0.5728	0.5938
	PDGN†	0.7025	0.6812	0.6921	0.8036	0.7985	0.8010	0.6235	0.5924	0.6076
	AIFI†	0.7105	0.6915	0.7009	0.7925	0.8034	0.7979	0.6342	0.5911	0.6119
	FCKT‡	0.7110	0.6560	0.6824	0.7958	0.7698	0.7826	0.5860	0.5650	0.5753
LLM-based	GPT-3.5-turbo Zero-Shot†	0.3462	0.4065	0.3739	0.6221	0.6605	0.6407	0.3750	0.2868	0.3250
	GPT-3.5-turbo Few-Shot†	0.3389	0.4452	0.3855	0.5847	0.6605	0.6203.	0.3812	0.2841	0.3256
	GPT-3.5-turbo CoT†	0.3430	0.4581	0.3923	0.6624	0.6420	0.6520	0.4233	0.2752	0.3335
	GPT-3.5-turbo CoT + Few-Shot†	0.3532	0.4581	0.3989	0.6215	0.6790	0.6490	0.3752	0.3012	0.3342
	GPT-4o Zero-Shot†	0.3214	0.4065	0.3590	0.6242	0.5741	0.5981	0.2326	0.3704	0.2857
	GPT-4o Few-Shot†	0.3299	0.4194	0.3693	0.6571	0.5679	0.6093	0.2532	0.3631	0.2984
	GPT-4o CoT†	0.3371	0.3806	0.3576	0.6643	0.5741	0.6159	0.2823	0.3891	0.3272
	GPT-4o CoT + Few-Shot†	0.3622	0.4323	0.3941	0.6842	0.6420	0.6624	0.4022	0.3842	0.3930
Ours:PSJL		**0.7376***	**0.6830***	**0.7093***	**0.8152***	**0.8017***	**0.8084***	**0.6460***	**0.6010***	**0.6227***

4 Module Hyperparameters.

We introduce three key modules and their hyperparameter settings as follows:

- Probabilistic Joint Modeling (PJM) module: dependency radius dep_radius ∈ {1, 2, 3}; dependency weight dep_weight ∈ {0.3, 0.4, 0.5}; maximum aspect length

max_answer_length ∈ {8, 10, 12, 14}; candidate threshold logit_threshold ∈ [0.024, 0.035]; number of retained candidates n_best_size ∈ {8, 10, 12, 14}; maximum training span length max_span_train_len ∈ {6, 8, 10, 12}.

- Syntax-Constrained Contrastive Learning (SCCL) module: contrastive learning weight cl_weight ∈ {0.2, 0.3, 0.4}.
- Adaptive Training Strategy (ATS): label smoothing coefficient ac_label_smoothing ∈ {0.00, 0.05, 0.10}; scope target scope_target ∈ {0.7, 0.8, 0.85, 0.9}; adaptive step size scope_adapt_alpha ∈ [0.15, 0.35].

4.1 Baseline Models

In **Table 1**, † denotes results reported from original papers, while ‡ denotes results reproduced under our implementation setting. We follow prior ABSA studies and keep the same datasets, evaluation metrics, and task definitions for comparability.We evaluate PSJL against three baseline categories: (i) two-stage pipelines that extract aspect terms and then predict sentiment; (ii) unified end-to-end joint models; and (iii) prompting-based LLM baselines.

Pipeline baselines: CRF-Pipeline [10], NN-CRF-Pipeline [8], TAG-Pipeline [6], and SPAN-Pipeline [6].

End-to-end baselines: SPJM [7], SPAN-Joint [6], S-AESC [5], HI-ASA [11], DCS [24], MiniConGTS [25], PDGN [26], AIFI [12], and FCKT [13].

LLM prompting baselines: We test GPT-3.5-turbo and GPT-4o under four prompting setups: zero-shot, few-shot, CoT, and CoT+few-shot.

Dataset	Sentences	Aspects	Positive	Negative	Neutral
Laptop	1869	2936	1326	900	620
Restaurant	3900	6603	4134	1538	931
Tweets	2350	3243	703	274	2266

Table 2. Dataset statistics for Laptop, Restaurant, and Tweets.

4.2 Comparative Experiments

Table 1 reports the results on all three datasets. PSJL delivers the strongest overall performance in terms of Precision, Recall, and F1. The gains are consistent when compared with strong baselines built on dependency-graph modeling and sequence labeling, including AIFI, PDGN, DCS, and MiniConGTS. This pattern suggests that our design improves both span boundary quality and sentiment reliability, thanks to better candidate selection and more effective cross-context discrimination.

Furthermore, the cross-domain performance is consistent: PSJL demonstrates more significant advantages on Laptop and Restaurant datasets with more regular syntax, while remaining leading on the Tweets dataset with noisier and shorter texts. This reflects the robustness of the method across different text styles and length distributions.

In addition, we compare our method with LLM baselines equipped with Few-Shot/CoT capabilities. The results show that even with enhanced prompting and reasoning strategies, the TSA performance of GPT-4o (CoT+Few-Shot) is still significantly lower than that of PSJL, while its computational and reasoning overhead is much higher. This phenomenon indicates that in fine-grained TSA scenarios, task-specific structural priors and contrastive consistency learning can better capture aspect-level fine-grained clues than general large language models, achieving higher cost-effectiveness and deployability. These results demonstrate the superiority of the PSJL model and validate the effectiveness of dependency triggering and curriculum-based training scheduling.

Table 3. Method-wise performance on aspect-term extraction measured by F1, and sentiment prediction measured by accuracy.

Task	Method	Laptop	Restaurant	Tweets
AE	HI-ASA(COLING22)	0.8424	0.8511	0.7546
	DCS(EMNLP23)	0.8455	0.8462	0.7543
	MiniConGTS(EMNLP24)	0.8374	0.8432	0.7514
	PDGN(ACL24)	0.8485	0.8546	0.7593
	GPT-4o CoT + Few-Shot	0.5120	0.6010	0.4724
	AIFI(AAAI24)	0.8511	0.8631	0.7634
	FCKT(IJCAI25)	0.8550	0.8532	0.7540
	PSJL	**0.8780**	**0.8784**	**0.7790**
SP	HI-ASA(COLING22)	0.8531	0.9257	0.8451
	DCS(EMNLP23)	0.8467	0.9212	0.8429
	MiniConGTS(EMNLP24)	0.8541	0.9194	0.8349
	PDGN(ACL24)	0.8519	0.9224	0.8496
	GPT-4o CoT + Few-Shot	0.7045	0.7538	0.6323
	AIFI(AAAI24)	0.8594	0.9305	0.8496
	FCKT(IJCAI25)	0.8390	0.9222	0.8220
	PSJL	**0.8654**	**0.9384**	**0.8560**

4.3 Ablation Experiments and Analysis

To assess PSJL effectiveness, we conduct ablation studies over its three components on the TSA task. As shown in **错误!未找到引用源。**, removing **PJM** (row 2) decreases TSA F1 by **0.99%** on average, confirming the benefit of probabilistic joint span-sentiment modeling. Dropping **ATS** (row 3) leads to a larger **1.84%** average drop, especially on the noisy Tweets dataset, showing that the adaptive training strategy is crucial for stabilizing learning under imperfect spans. Removing **SCCL** (row 4) causes a **1.57%** average degradation, indicating that syntax-constrained contrastive learning effectively

filters unreasonable spans. When two modules are disabled simultaneously, the performance drops further (up to **3.48%**), and turning off all three components results in the largest decline of **4.63%**, demonstrating that PJM, ATS, and SCCL are complementary and jointly responsible for the strong performance of PSJL.

Table 4. Results of ablation studies on the TSA task. "✓/✗ PJM" indicates whether the Probabilistic Joint Modeling module is used, "✓/✗ ATS" denotes the inclusion or exclusion of the Adaptive Training Strategy, and "✓/✗ SCCL" represents whether the Syntax-Constrained Contrastive Learning module is applied.

Task	PJM	ATS	SCCL	Laptop	Restaurant	Tweets	Avg.Drop
TSA	✓	✓	✓	**0.7093**	**0.8084**	**0.6227**	-
	✗	✓	✓	0.7023	0.8031	0.6144	▽ 0.99%
	✓	✗	✓	0.6954	0.8002	0.6068	▽ 1.84%
	✓	✓	✗	0.6953	0.8012	0.6111	▽ 1.57%
	✗	✗	✓	0.6948	0.7955	0.5935	▽ 2.78%
	✗	✓	✗	0.6924	0.7980	0.6099	▽ 1.91%
	✓	✗	✗	0.6540	0.7912	0.6196	▽ 3.48%
	✗	✗	✗	0.6694	0.7873	0.5875	▽ 4.63%

4.4 Hyperparameter Experiments

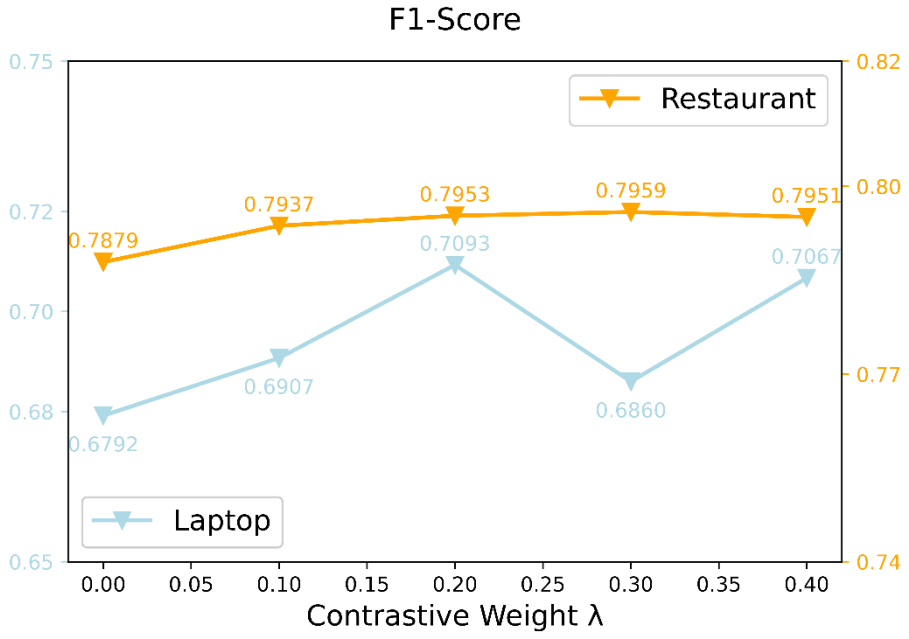

Fig. 3. Performance variation with contrastive weight λ across datasets.

As shown in **Fig. 3**, λ = 0.3 balances representation learning and task-specific optimization. Meanwhile, **Fig. 4** indicates that ξ = 0.8 achieves the best trade-off between gold supervision and predicted boundary reliance. We analyze the effects of two hyperparameters, λ and ξ, as shown in **Fig. 3** and **Fig. 4**.

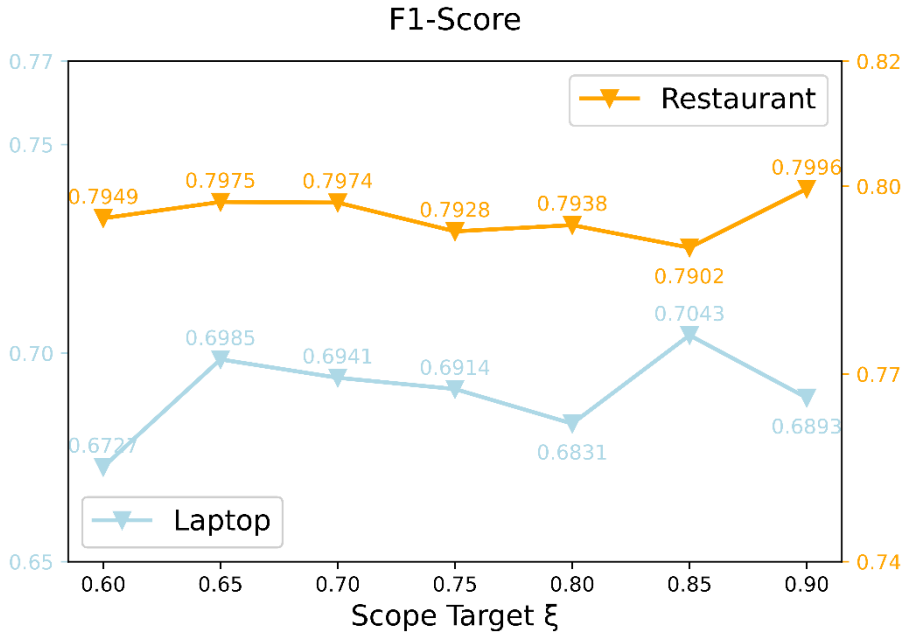

Fig. 4. Impact of scope target ξ on adaptive training stability.

5 Conclusion

This paper proposes PSJL to address weak cross-task feedback between boundary detection and sentiment classification, as well as the train–inference mismatch caused by training on gold spans but inferring on predicted spans in aspect-based sentiment analysis. PSJL comprises three complementary modules: Probabilistic Joint Modeling (PJM), which relaxes discrete span selection into differentiable span distributions so that sentiment supervision can directly refine boundary detection in an end-to-end manner; Syntax-Constrained Contrastive Learning (SCCL), which regularizes boundary pairing with dependency-based syntactic priors via contrastive learning to improve syntactic plausibility; and Adaptive Training Strategy (ATS), which dynamically mixes gold-span and predicted-span supervision to stabilize optimization and reduce train–inference discrepancy. Extensive experiments on three public benchmarks, namely Laptop, Restaurant, and Tweets, demonstrate that PSJL consistently improves aspect

extraction, sentiment prediction, and targeted sentiment analysis performance over representative end-to-end baselines and strong LLM-based prompting methods, and ablation studies further verify the effectiveness and complementarity of each component. As a limitation, PSJL relies on dependency parses to provide syntactic cues; improving robustness to noisy parses and reducing overhead are promising directions. Future work will explore extending PSJL to more challenging ABSA settings such as implicit aspects and domain-shifted scenarios, and investigate its integration with large language models.

References

1. 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), 2014, pp. 27–35.
2. M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, and I. Androutsopoulos, "SemEval-2015 task 12: Aspect based sentiment analysis," in Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), 2015, pp. 486–495.
3. M. Pontiki et al., "SemEval-2016 task 5: Aspect based sentiment analysis," in Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), 2016, pp. 19–30.
4. M. Zhang, Y. Zhang, and D.-T. Vo, "Gated neural networks for targeted sentiment analysis," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30, no. 1, 2016.
5. Y. Lv, F. Wei, Y. Zheng, C. Wang, C. Wan, and C. Wang, "A span-based model for aspect terms extraction and aspect sentiment classification," Neural Computing and Applications, vol. 33, no. 8, pp. 3769–3779, 2021.
6. M. Hu, Y. Peng, Z. Huang, D. Li, and Y. Lv, "Open-domain targeted sentiment analysis via span-based extraction and classification," in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 537–546.
7. Y. Zhou, L. Huang, T. Guo, J. Han, and S. Hu, "A span-based joint model for opinion target extraction and target sentiment classification," in Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, 2019, pp. 5485–5491.
8. M. Zhang, Y. Zhang, and D.-T. Vo, "Neural networks for open domain targeted sentiment," in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 612–621.
9. 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.
10. M. Mitchell, J. Aguilar, T. Wilson, and B. Van Durme, "Open domain targeted sentiment," in Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013, pp. 1643–1654.
11. W. Chen, J. Du, Z. Zhang, F. Zhuang, and Z. He, "A hierarchical interactive network for joint span-based aspect-sentiment analysis," in Proceedings of the 29th International Conference on Computational Linguistics, 2022, pp. 7013–7019.
12. W. Chen, Y. Liu, Z. Zhang, F. Zhuang, and J. Zhong, "Modeling adaptive inter-task feature interactions via sentiment-aware contrastive learning for joint aspect-sentiment prediction,"

in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, no. 16, 2024, pp. 17 781–17 789.
13. W. Chen, Z. Zhang, M. Yuan, K. Xu, and F. Zhuang, "Fckt: Fine-grained cross-task knowledge transfer with semantic contrastive learning for targeted sentiment analysis," in Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25, 2025, pp. 2731–2739.
14. B. Huang and K. Carley, "Syntax-aware aspect level sentiment classification with graph attention 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), 2019, pp. 5469–5477.
15. C. Zhang, Q. Li, and D. Song, "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), 2019, pp. 4568–4578.
16. K. Sun, R. Zhang, S. Mensah, Y. Mao, and X. Liu, "Aspect-level sentiment analysis via convolution over dependency tree," 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), 2019, pp. 5679–5688.
17. K. Wang, W. Shen, Y. Yang, X. Quan, and R. Wang, "Relational graph attention network for aspect-based sentiment analysis," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 3229–3238.
18. H. Cai, H. Ma, J. Yu, and R. Xia, "A joint coreference-aware approach to document-level target sentiment analysis," in Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2024, pp. 12 149–12 160.
19. 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.
20. 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.
21. S. Liang, W. Wei, X.-L. Mao, F. Wang, and Z. He, "BiSyn-GAT+: Bi-syntax aware graph attention network for aspect-based sentiment analysis," in Findings of the Association for Computational Linguistics: ACL 2022, 2022, pp. 1835–1848.
22. S. Yang et al., "Single ground truth is not enough: Adding flexibility to aspect-based sentiment analysis evaluation," in Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), 2025, pp. 12 071–12 096.
23. C. Shen, W. Wei, D. Wang, and Z.-H. Wang, "Zero-shot cross-domain aspect-based sentiment analysis via domain-contextualized chain-of-thought reasoning," in Findings of the Association for Computational Linguistics: EMNLP 2025, 2025, pp. 4558–4573.
24. P. Li, P. Li, and K. Zhang, "Dual-channel span for aspect sentiment triplet extraction," in Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 2023, pp. 248–261.
25. Q. Sun, L. Yang, M. Ma, N. Ye, and Q. Gu, "MiniConGTS: A near ultimate minimalist contrastive grid tagging scheme for aspect sentiment triplet extraction," in Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, 2024, pp. 2817–2834.

26. L. Zhu et al., “Pinpointing diffusion grid noise to enhance aspect sentiment quad prediction,” in Findings of the Association for Computational Linguistics: ACL 2024, 2024, pp. 3717–3726.
27. J. Barnes, R. Kurtz, S. Oepen, L. Øvrelid, and E. Velldal, “Structured sentiment analysis as dependency graph parsing,” 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. 3387–3402.
28. 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.
29. Q. Zhao, F. Yang, D. An, and J. Lian, “Modeling structured dependency tree with graph convolutional networks for aspect-level sentiment classification,” Sensors, vol. 24, no. 2, p. 418, 2024.
30. S. Poria, N. Ofek, A. Gelbukh, A. Hussain, and L. Rokach, “Dependency tree-based rules for concept-level aspect-based sentiment analysis,” in Semantic Web Evaluation Challenge. Springer, 2014, pp. 41–47.
31. K. Peng et al., “T-t: Table transformer for tagging-based aspect sentiment triplet extraction,” in Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25, 2025, pp. 8222–8230.
32. S. Tang, L. Li, X. Tao, L. Zhong, and Q. Xie, “MATO: A model-agnostic training optimization for aspect sentiment triplet extraction,” in Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), 2025, pp. 1648–1662.
33. M. Zhang, Y. Zhu, Z. Liu, Z. Bao, Y. Wu, X. Sun, and L. Xu, “Span-level aspect-based sentiment analysis via table filling,” in Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2023, pp. 9273–9284.
34. Z. Qiu, K. Chen, Y. Xue, Z. Ma, and Z. Zhang, “Modeling inter-aspect relations with clause and contrastive learning for aspect-based sentiment analysis,” IEEE Transactions on Computational Social Systems, vol. 11, no. 2, pp. 2833–2842, 2024.
35. W. Wu, D. Wang, M. Wang, S. Feng, and Y. Zhang, “Sentiment triplet extraction with multi-view contrastive learning,” IEEE Transactions on Affective Computing, vol. 16, no. 3, pp. 1526–1542, 2025.