Tree-CoT-RT: An Explainable Multi-Path Tree-Guided Chain-of-Thought and Reinforcement Learning Framework for Aspect Sentiment Quad Prediction

Hao Zhang[1,2†] **Jiahao Wang**[2] **Zhenke Duan**[3] **Xin Yin**[4] **Haichuan Hu**[5]
Hualong Chen[6] **Yi Su**[7] **Congqing He**[8] **Yike Tan**[9] **Yu-N Cheah**[2†]
[1]XU Exponential University of Applied Sciences, Germany
[2]Universiti Sains Malaysia, Malaysia
[3]Zhongnan University of Economics and Law, China
[4]Zhejiang University, China
[5]Hong Kong Polytechnic University, Hong Kong SAR, China
[6]Capital Medical University, China [7] Xiangtan University, China
[8]Huzhou Normal University, China [9] Carnegie Mellon University, USA
Correspondence: h.zhang@xu-university.de yncheah@usm.my

Abstract

Aspect Sentiment Quad Prediction (ASQP) is a fundamental yet challenging task in fine-grained sentiment analysis, particularly when aspects or opinions are implicit. Existing methods often lack explainability and generalization, making it difficult to justify inference decisions and to detect implicit sentiment across domains and varied expression patterns. To address these limitations, we propose Tree-CoT-RT, an explainable multi-path tree-guided chain-of-thought and reinforcement learning framework specifically designed for ASQP. The core idea is to use sentiment tree structures to design type-specific reasoning templates that guide LLMs in generating explainable chains, including both final sentiment quadruples and intermediate inference steps for transparent implicit reasoning. However, the generated reasoning chains often vary in quality and may contain logical inconsistencies. To mitigate this, we introduce a reinforcement learning strategy with a rule-based reward function to generate high-quality reasoning traces, which are then used to fine-tune the LLM and enable controlled sampling. Experiments on benchmark datasets demonstrate that Tree-CoT-RT substantially outperforms strong baselines, particularly in scenarios involving implicit sentiment analysis. [1]

1 Introduction

Aspect-Based Sentiment Quad Prediction (ASQP) is a fundamental task in fine-grained sentiment analysis that extracts four elements from text: Aspect, Category, Opinion, and Sentiment. It supports applications such as review mining and opinion monitoring.

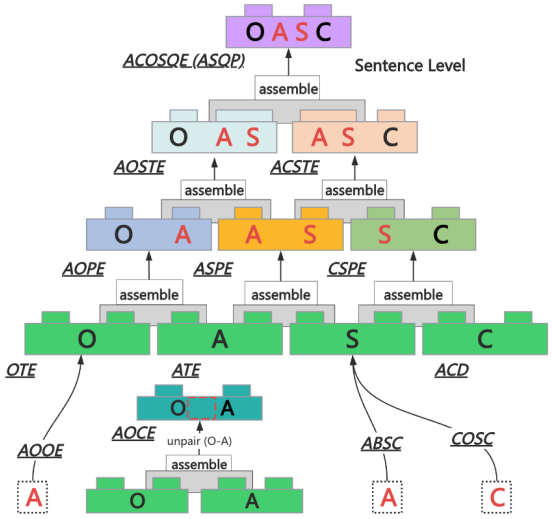

Figure 1: Building ABSA tasks like assembling Lego blocks. Progressing from element level to full ASQP, covering ACOS: Aspect, Category, Opinion, and Sentiment.

Surveys (Zhang et al., 2023a, 2024a) present the first comprehensive overview of ASQP, reclassifying ABSA subtasks, summarizing PLM-based approaches, and examining the role of ChatGPT in sentiment analysis. Recent progress has been largely driven by pre-trained language models (PLMs), with Transformer-based architectures such as BERT (Devlin et al., 2019), BART (Lewis et al., 2020), and T5 (Raffel et al., 2020) achieving state-of-the-art results through fine-tuning strategies. More recently, large language models (LLMs) such as the closed-source GPT-5 (OpenAI, 2025)

[†]Corresponding authors
[1]Our experimental codes and data are available at: https://github.com/sydmou/ASQP-Tree-CoT-RT

Findings of the Association for Computational Linguistics: ACL 2026, pages 16372–16391
July 2-7, 2026

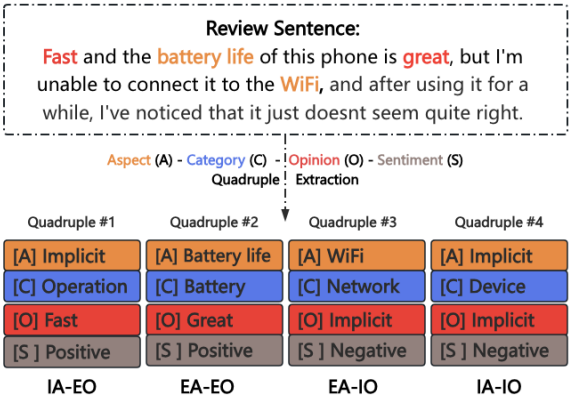

Figure 2: An illustration of the aspect sentiment quad prediction task.

and open-source models like Qwen3 (Qwen, 2025) and LLaMA (Grattafiori et al., 2024) have been applied to ASQP tasks, demonstrating improved generalization and reasoning capabilities in handling implicit sentiment and complex opinion structures. As shown in Figure 1, ABSA tasks can be hierarchically composed from element-level to full quadruple-level structures tasks such as Aspect-Based Sentiment Classification (ABSC)(Wang et al., 2016; Liu and Zhang, 2017), Opinion Term Extraction (OTE), and Category-Oriented Sentiment Classification (COSC) form the foundation of ABSA. These are composed into pair-level tasks including Aspect-Opinion Pair Extraction (AOPE)(Zhao et al., 2020), Aspect-Sentiment Pair Extraction (ASPE)(Cai et al., 2020; Liu et al., 2021), and Category-Sentiment Pair Extraction (CSPE)(Wan et al., 2020; Bu et al., 2021), as well as fusion tasks like Aspect-Oriented Opinion Extraction (AOOE)(Fan et al., 2019) and Aspect-Opinion Co-Extraction (AOCE)(Yin et al., 2016). Triplet-level tasks such as AOSTE (Peng et al., 2020; Xu et al., 2020; Mao et al., 2021; Chen et al., 2021) and ACSTE (Wan et al., 2020; Wu et al., 2021; Zhang et al., 2021b) further increase the structural complexity. At the top, Aspect Sentiment Quad Prediction (ASQP), also referred to as Aspect–Category–Opinion–Sentiment Quadruple Extraction (ACOSQE), unifies ACOS into a full quadruple prediction task, requiring more sophisticated reasoning, especially for implicit sentiment.

Implicit aspects/opinions are a core difficulty in ASQP. Figure 2 shows that explicit cues (e.g., "Fast", "Great") are easy to detect, while implicit evidence like "unable to connect to Wi-Fi" demands contextual inference of sentiment toward unstated elements. Yet current methods remain black-box and unreliable in implicit cases, often yielding inconsistent, weakly generalizable, and sometimes logically invalid outputs.

To improve explainability and generalization in ASQP, especially under implicit sentiment, we propose a unified framework combining structured reasoning, GRPO-based RL for reasoning-chain optimization, and self-consistent inference. We generate multi-path sentiment chains of thought across implicit/explicit settings, use high-quality chains to fine-tune LLMs for stronger implicit understanding, and aggregate multiple reasoning paths at inference to produce robust ACOS quadruples. Experiments on two benchmarks show consistent gains over state-of-the-art methods, with the largest improvements on implicit aspects and opinions.

- A Tree-CoT prompting strategy introduces structured, type-specific explainable reasoning traces to guide LLMs toward explainable and consistent sentiment inference.
- A GRPO-based reinforcement learning method selects high-quality reasoning paths using a rule-based reward, improving logical consistency and diversity.
- The model is fine-tuned on selected reasoning chains and employs self-consistent inference at test time, enhancing its ability to handle implicit sentiment and produce robust predictions.

2 Related Works

2.1 Template-Based Augmentation.

To enhance generalization and address data sparsity in sentiment quadruple extraction, recent studies have proposed structure-aware augmentation strategies. Opinion Tree (Bao et al., 2022) encodes hierarchical sentiment elements as tree paths, producing structured targets that guide the model in capturing complex sentiment relationships. MvP (Gou et al., 2023) employs multi-view prompting to generate diverse reasoning paths, which are aggregated to enrich the training distribution. Reinforcement learning has also been explored to improve augmentation quality. Target-to-Source Augmentation (Zhang et al., 2023b), designed for aspect sentiment triplet extraction (ASTE), trains a generator to produce labeled sentences from given triples and

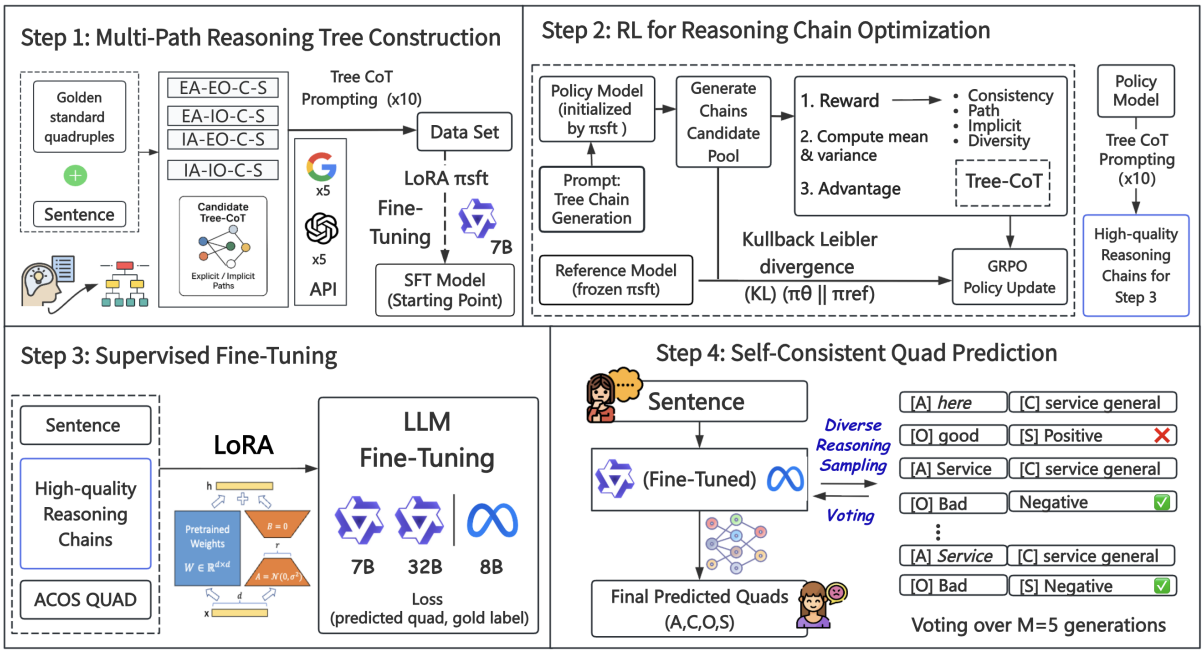

Figure 3: Framework of the Tree-CoT-RT approach. It integrates multi-path reasoning tree generation, RL-based reasoning chain selection, supervised fine-tuning, and self-consistent inference to enhance ASQP performance and explainability.

templates, using fluency and alignment discriminators to assess quality and RL to optimize generation. To tackle long-tail pattern sparsity, ADA (Zhang et al., 2024c) synthesizes rare quadruples based on syntactic templates and class-conditioned constraints, effectively improving model robustness against low-frequency sentiment patterns.

2.2 Chain-of-Thought Reasoning

Chain-of-Thought (CoT) prompting improves both accuracy and explainability in large language models by making intermediate reasoning steps explicit. Initially explored in large-scale settings (Wei et al., 2022), CoT has shown strong results in math and language tasks, with performance improving as model size increases. Even simple prompts like "Let's think step by step" enable stepwise reasoning across domains (Kojima et al., 2022), and recent surveys summarize its progress and challenges (Chu et al., 2024). In sentiment analysis, especially ABSA, CoT has been applied to tasks involving aspects, opinions, categories, and sentiments. THOR (Fei et al., 2023) adopts a multi-hop chain (aspect → opinion → polarity) for implicit sentiment reasoning. SCRAP (Kim et al., 2024) improves robustness via self-consistency voting, while S^2IT (Chen et al., 2025) further integrates syntactic structure into CoT without modifying model architecture, demonstrating the value of combining linguistic cues with explicit reasoning.

3 Methodology

3.1 ASQP Problem Formulation

We formulate Aspect Sentiment Quad Prediction (ASQP) as a joint extraction task, aiming to identify a set of sentiment quadruples from a given sentence:

$$Q = \{(a_i, c_i, o_i, s_i)\}_{i=1}^{n} \quad (1)$$

Each quadruple consists of an *aspect term* $a_i \in \mathcal{V}_x \cup \{\text{Implicit}\}$, an *opinion term* $o_i \in \mathcal{V}_x \cup \{\text{Implicit}\}$, a *category* $c_i \in \mathcal{C}$ from a predefined set (e.g., *Food*, *Service*), and a *sentiment polarity* $s_i \in \{\text{POS}, \text{NEU}, \text{NEG}\}$. Here, $\mathcal{V}_x$ denotes the sentence vocabulary. We focus on the implicit challenge, where a_i and/or o_i are not explicitly mentioned in the text.

3.2 Framework Overview

Our framework enhances ASQP performance and explainability through a four-Step reasoning-guided pipeline (Figure 3). In Step 1 (S1), we construct type-specific sentiment reasoning trees for the four ASQP types (EAEO/EAIO/IAEO/IAIO). Each tree specifies a step-by-step inference path over (A, C, O, S), with particular emphasis on implicit aspects and/or opinions, and guides LLMs to

Quad Type	Aspect	Opinion	Sentiment	Category	Reasoning Path Summary
EA-EO-C-S	Explicit	Explicit	A + O → S	A → C	Identify A+O → Pair → Sentiment, Category
EA-IO-C-S	Explicit	Implicit	A + IO → S	A → C	A → Infer O → Sentiment, Category
IA-EO-C-S	Implicit	Explicit	IA + O → S	IA → C	O → Infer A → Sentiment, Category
IA-IO-C-S	Implicit	Implicit	IA + IO → S	IA → C	Context → Infer IA + IO → Sentiment, Category

Table 1: Reasoning paths for different quad Types

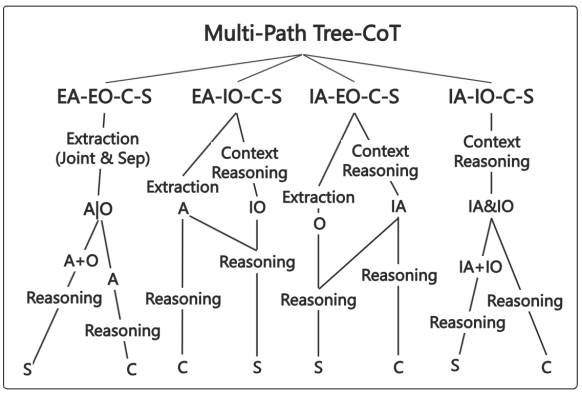

Figure 4: Tree-structured CoT reasoning paths.

generate Tree-CoT reasoning chains paired with gold-standard quadruples. We then perform a lightweight tree-supervised fine-tuning (tSFT) with LoRA to obtain an instruction-following starting model for RL. In Step 2 (S2), reinforcement learning (GRPO) ranks and filters reasoning chains using a rule-based reward that measures consistency, path validity, implicit alignment, and diversity, producing high-quality chains for downstream learning. In Step 3 (S3), the selected chains supervise LLM fine-tuning to generate accurate quadruples together with reasoning traces. In Step 4 (S4), we sample multiple reasoning paths during inference and apply voting to ensure consistent and robust outputs. The following sections elaborate on S1–S4.

3.3 S1: Multi-Path Reasoning Construction

In Step 1, we build a candidate pool of Tree-CoT reasoning chains guided by type-specific sentiment reasoning trees. For the four ASQP types (EAEO, EAIO, IAEO, IAIO), we design Tree-CoT prompts that encode hierarchical, step-by-step inference paths tailored to each type. Given a sentence and its corresponding quad type, the tree determines the reasoning order for inferring Aspect, Category, Opinion, and Sentiment, explicitly covering both explicit and implicit elements. Table 1 outlines the logic of each ASQP type, while Figure 4 shows the corresponding Tree-CoT structures. For each sentence, we generate $K{=}10$ Tree-CoT reasoning chains paired with gold-standard quadruples, forming a supervised dataset for reasoning-guided learning. The chains are generated via LLM APIs (Gemini 2.5 Pro (Google DeepMind, 2024) and ChatGPT-4 (OpenAI, 2023)) and are used solely to construct the tSFT dataset. Following this, we perform **tree-supervised fine-tuning (tSFT)** using LoRA on an instruction-tuned backbone (e.g., Qwen2.5-7B-Instruct), yielding an *SFT starting model* that serves as the initialization point for RL in Step 2.

3.4 S2: RL for Reasoning Chain Optimisation

To refine the reasoning chains generated in Step 1, we apply reinforcement learning (RL) based on Group Relative Policy Optimization (GRPO) (Shao et al., 2024), inspired by DeepSeek R1-Zero (DeepSeek, 2024). This approach enables efficient selection of high-quality reasoning paths for data augmentation, without relying on learned value or reward models.

Model Components. In Step 2, our RL framework is built on Qwen2.5-7B-Instruct and starts from the **tSFT model** obtained in Step 1. It includes three components: (1) a policy model π_θ, initialized from the tSFT starting model and updated via reward-guided GRPO; (2) a frozen reference model π_{ref}, defined as a *snapshot of the initial policy* π_{θ_0} at the start of GRPO and used only for KL regularization; (3) a rule-based reward function that scores each chain by (i) label consistency, (ii) path validity, (iii) implicit alignment, and (iv) diversity. Details of the reward design are provided in Appendix C.1.

Reward Function. Each chain o_i is scored as:

$$R(o_i) = \lambda_1 R_{\text{match}}(o_i) + \lambda_2 R_{\text{path}}(o_i) + \lambda_3 R_{\text{implicit}}(o_i) + \lambda_4 R_{\text{diverse}}(o_i), \quad (2)$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4 \geq 0$.

The weights balance label consistency, path validity, implicit-alignment, and diversity.

Group-Based Advantage Computation. For each input, the policy samples a group of G=10 candidate reasoning chains $\mathcal{G} = \{o_i\}_{i=1}^{G}$. For each $o_i \in \mathcal{G}$:

$$\mu_{\mathcal{G}} = \frac{1}{|\mathcal{G}|} \sum_{o_i \in \mathcal{G}} R(o_i), \tag{3}$$

$$\sigma_{\mathcal{G}}^2 = \frac{1}{|\mathcal{G}|} \sum_{o_i \in \mathcal{G}} (R(o_i) - \mu_{\mathcal{G}})^2, \tag{4}$$

$$A(o_i) = \frac{R(o_i) - \mu_{\mathcal{G}}}{\sqrt{\sigma_{\mathcal{G}}^2 + \epsilon}} \tag{5}$$

$A(o_i)$ measures the relative quality of each chain within its group.

KL-Regularized GRPO Loss. Following the GRPO Policy (Shao et al., 2024), we formulate the final optimization objective as:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q \sim \mathcal{P}(Q),\, \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}} \Bigg[\frac{1}{G} \sum_{i=1}^{G} \frac{1}{|\mathcal{O}_i|} \sum_{t=1}^{|\mathcal{O}_i|} \min \left(\frac{\pi_\theta(o_{i,t})}{\pi_{\theta_{\text{old}}}(o_{i,t})}, \text{clip} \left(\frac{\pi_\theta(o_{i,t})}{\pi_{\theta_{\text{old}}}(o_{i,t})}, 1-\epsilon, 1+\epsilon \right) \right) \hat{A}_{i,t} - \beta \cdot \mathrm{D}_{\text{KL}} \left[\pi_\theta \,\|\, \pi_{\text{ref}} \right] \Bigg] \tag{6}$$

where $\hat{A}_{i,t}$ is the token-level advantage. Clipping stabilizes updates, and β controls regularization strength.

KL Divergence Term. We adopt the forward Kullback–Leibler divergence (KL) formulation from the GRPO Policy (Shao et al., 2024):

$$\mathrm{D}_{\text{KL}} \left[\pi_\theta \,\|\, \pi_{\text{ref}} \right] = \frac{\pi_{\text{ref}}(o_{i,t})}{\pi_\theta(o_{i,t})} - \log \frac{\pi_{\text{ref}}(o_{i,t})}{\pi_\theta(o_{i,t})} - 1 \tag{7}$$

This constraint keeps π_θ close to π_{ref}, avoiding drift toward low-quality but high-reward chains.

Selected Chains. After GRPO converges, $\pi_{\theta^\star}$ is used to sample G Tree-CoT chains per sentence (default G=10). Each chain is parsed and scored by $R(\cdot)$ and advantage $A(\cdot)$, after which we filter invalid outputs, keep the top-N chains by $A(\cdot)$, and deduplicate near-duplicates. The selected chains paired with gold quads form the supervision for Step 3 fine-tuning.

3.5 S3: Supervised Fine-Tuning

In Step 3, we fine-tune LLMs using the high-quality reasoning chains and their corresponding ACOS quadruples selected in Step 2. We employ instruction-tuned Qwen-7B and Qwen-32B models as our main backbones for performance comparison with existing baselines. Additionally, Meta's LLaMA-8B model is included in ablation studies to assess generalization under different backbone settings. Each training instance consists of an input sentence x, a filtered reasoning chain o_i, and a target sentiment quadruple $q_i = (a_i, c_i, o_i, s_i)$. The model is trained to generate both the reasoning trace and the final structured output in a unified sequence.

To efficiently adapt large models, we apply Low-Rank Adaptation (LoRA) (Hu et al., 2022a)by injecting trainable low-rank matrices $\Delta W = AB^\top$ into attention and feedforward layers, where $A \in \mathbb{R}^{d \times r}$ and $B \in \mathbb{R}^{r \times d}$ with $r \ll d$. The original weights W are frozen, enabling parameter-efficient fine-tuning. The model is trained to generate both the reasoning chain o_i and the final quadruple $q_i = (a_i, c_i, o_i, s_i)$ as a unified sequence $y^{(i)} = [o_i; a_i; c_i; o_i; s_i]$, optimized by:

$$\mathcal{L}\text{gen} = -\sum i = 1^k \sum_{t=1}^{T_i} \log P_\theta(y_t^{(i)} \mid y_{<t}^{(i)}, x) \tag{8}$$

where θ denotes LoRA parameters and T_i is the sequence length.

3.6 S4: Self-Consistent Quad Prediction

At inference time, the fine-tuned model receives an input sentence x and generates five reasoning chains $\{o_i\}_{i=1}^{5}$ via stochastic decoding (e.g., top-p or temperature sampling). Each chain leads to a predicted sentiment quadruple $q_i = (a_i, c_i, o_i, s_i)$. Due to generation variability, individual outputs may differ in quality. To improve robustness and consistency, we adopt a self-consistency mechanism based on majority voting. Specifically, we select the most frequently predicted quadruple across all sampled outputs:

$$q^* = \arg\max_q \sum_{i=1}^{5} \mathbb{I}[q_i = q] \tag{9}$$

where $\mathbb{I}[\cdot]$ is the indicator function. The final output q^* reflects the consensus prediction among diverse reasoning paths.

	Restaurant	Laptop
#Categories	13	121
#Sentences (S)	2286	4076
#Quads (Q)	3658	5758
#Q/S	1.60	1.42
#EA & EO	2429 (66.40%)	3269 (56.77%)
#IA & EO	530 (14.49%)	910 (15.80%)
#EA & IO	350 (9.57%)	1237 (21.48%)
#IA & IO	349 (9.54%)	342 (5.94%)
#POS	2503	3578
#NEU	151	316
#NEG	1007	1879
#Train	1530	2934
#Dev	171	326
#Test	583	816
#Train (Quads)	2484	4172
#Dev (Quads)	261	440
#Test (Quads)	916	1161

Table 2: Dataset statistics for the ACOS benchmark. Both domains include explicit and implicit aspects and opinions, covering all four ASQP types (EA&EO, EA&IO, IA&EO, IA&IO) with balanced sentiment distributions (#NEG, #NEU, #POS).

4 Experimental Setup

4.1 Datasets

ACOS Dataset We incorporate the Restaurant and Laptop datasets from (Cai et al., 2021), detailed in Table 2. They are divided into training, validation, and testing sets for systematic model training and evaluation.

4.2 Compared Models

We compare our methods with the following two types of previous state-of-the-art methods:

Pipeline models. Double Propagation (DP) (Qiu et al., 2011), adapted by Cai et al. (2021) for ASQP, enhances coverage by exploiting relationships between extracted aspects and opinions. JET (Xu et al., 2020), originally for triplet extraction, was extended to ASQP by combining BERT-based category classification. TAS-BERT (Wan et al., 2020) jointly extracts sentiment tuples, while Extract-Classify (Cai et al., 2021) decomposes the ACOS task into two sequential steps. TAS-BERT-ACOS (Cai et al., 2021) filters invalid aspect-opinion pairs based on predicted category-sentiment labels.

Unified models. BART-based models have been widely applied to unified ABSA. PARAPHRASE-BART (Xiong et al., 2023) performs aspect and sentiment extraction jointly, while GEN-NAT-SCL-BART and BART-CRN (Xiong et al., 2023) incorporate adversarial training or recurrent networks for enhanced robustness. BARTABSA (Hoang et al., 2022) separates sub-tasks within BART, and GAS (Zhang et al., 2021b) formulates ABSA as a generative task. Template-based approaches such as PARAPHRASE (Zhang et al., 2021a) define fixed output formats, while Seq2Path (Mao et al., 2022) and Opinion Tree (Bao et al., 2022) adopt tree-structured generation. Recent T5-based methods include GEN-SCL-NAT (Peper and Wang, 2022), which combines contrastive learning and adversarial training, and UnifiedABSA (Wang et al., 2024), which adopts multi-task instruction tuning. DLO (Hu et al., 2022b) optimizes structure selection based on training set scores. DLO+UAUL (Hu et al., 2023) combine unsupervised adversarial uncertainty learning (UAUL) to improve robustness, while MvP (Gou et al., 2023) fuses multi-view predictions. ASQP-ITSCL (Zhang et al., 2024b) introduces instruction-tuned contrastive learning framework for sentiment quad extraction with implicit aspects and opinions. S^2IT (Chen et al., 2025) enhances structure-aware generation via syntactic injection in a two-Step Qwen-based pipeline.

4.3 Experiment Details

4.3.1 Implementation Details

All models are implemented using the PyTorch framework and the HuggingFace Transformers library. Experiments are conducted on a cluster of four NVIDIA A100 GPUs. We use instruction-tuned LLMs including Qwen2.5-7B-Instruct, Qwen2.5-7B, Qwen2.5-32B, and LLaMA3-8B as backbones, fine-tuned using LoRA. Reasoning chains are generated via the official APIs of ChatGPT 5.1 and Gemini 2.5 Pro. The generation process uses a temperature of 0.7, top-p of 0.95, and a maximum generation length of 2,048 tokens to support multi-step reasoning. Each LLM-based method is run four times with random seeds (444, 555, 666, 777), and we report average results. Detailed training details, hyperparameters, and computational cost are provided in Appendix A.

4.3.2 Evaluation Metrics

The experiment employs F1 scores (F_1) as the main evaluation metric. A sentiment quad prediction is counted as correct if all the predicted elements are the same as the gold labels. The experiment also reports the precision (P) and recall (R) scores.

Method	Model	REST-ACOS			LAPTOP-ACOS		
		P.	R.	F1.	P.	R.	F1.
Double-Propagation (Cai et al., 2021)	RULE	34.67	15.08	21.04	13.0	5.70	8.0
JET-ACOS (Cai et al., 2021)	BERT	59.81	28.94	39.01	44.52	16.25	23.81
TAS-BERT-ACOS (Cai et al., 2021)	BERT	26.29	46.29	33.53	47.15	19.22	27.31
Extract-Classify (Cai et al., 2021)	BERT	38.54	52.96	44.61	45.56	29.48	35.80
PARAPHRASE-BART (Xiong et al., 2023)	BART	43.62	36.19	39.56	36.36	29.63	32.65
GEN-NAT-SCL-BART (Xiong et al., 2023)	BART	48.93	40.51	44.32	37.13	32.44	34.63
BART-CRN (Xiong et al., 2023)	BART	50.84	47.10	48.90	48.16	31.83	38.32
BARTABSA(split) (Hoang et al., 2022)	BART	56.80	51.09	53.45	41.06	37.89	39.41
Seq2Path(Mao et al., 2022)	T5-base	-	-	58.41	-	-	42.97
Muti-Task-IT(Wang et al., 2024)	T5-base	-	-	60.60	-	-	42.58
DLO + UAUL (Hu et al., 2023)	T5-base	-	-	60.78	-	-	43.65
PARAPHRASE (Peper and Wang, 2022)	T5-base	-	-	60.97	-	-	44.08
MvP (Gou et al., 2023)	T5-base	-	-	61.54	-	-	43.92
GEN-SCL-NAT (Peper and Wang, 2022)	T5-large	-	-	62.62	-	-	45.16
ASQP-ITSCL (Zhang et al., 2024b)	T5-base	61.45	60.92	61.18	44.69	44.19	44.43
ASQP-ITSCL (Zhang et al., 2024b)	T5-large	65.56	64.19	64.86	46.31	45.91	46.11
S^2IT-7B (Chen et al., 2025)	Qwen2.5-7B	-	64.90	66.10	-	44.70	45.90
S^2IT-32B (Chen et al., 2025)	Qwen2.5-32B	-	**66.60**	**67.37**	-	**45.40**	**46.70**
Tree-CoT-RT-7B (Ours)	Qwen2.5-7B	71.18	67.70	69.50	48.36	47.54	49.10
Tree-CoT-RT-32B (Ours)	Qwen2.5-32B	**73.34**	**69.52**	**71.22**	**51.22**	**49.72**	**50.25**

Table 3: Performance comparison of different methods on the REST-ACOS and LAPTOP-ACOS datasets.

5 Results and Discussions

5.1 Main Performance Results

Table 3 summarizes the overall results on the REST-ACOS and LAPTOP-ACOS datasets. The proposed Tree-CoT-RT framework achieves state-of-the-art performance on both benchmarks. Tree-CoT-RT-32B attains the highest F1 scores of 71.22 on REST-ACOS and 50.25 on LAPTOP-ACOS, outperforming the previous best model, S^2IT-32B, by 3.85 and 3.55 points, respectively. Notably, with a smaller Qwen2.5-7B backbone, Tree-CoT-RT-7B still achieves strong F1 scores of 69.50 and 49.10, exceeding all other non-32B baselines. These results demonstrate both the effectiveness and scalability of our structured reasoning framework. Compared with prior T5- and BART-based approaches (e.g., ASQP-ITSCL), Tree-CoT-RT yields consistent gains, indicating that explicit reasoning over complex sentiment structures leads to more accurate quad prediction.

5.2 Explicit and Implicit Sentiment Analysis

Table 4 shows F1 scores for four ACOS subtypes on REST-ACOS and LAPTOP-ACOS, revealing consistent patterns. EAEO is the easiest, as both aspect and opinion are explicit, while IAIO is the most difficult due to the absence of surface cues. The partially implicit types, IAEO and EAIO, fall in between, reflecting increased reasoning difficulty. Tree-CoT-RT-32B consistently outperforms baselines across all subtypes, showing strong generalization under different levels of implicitness. It slightly improves over prior models on EAEO and achieves notable gains on IAEO and EAIO, especially on the more challenging LAPTOP dataset. On IAIO, it delivers substantial improvement, confirming its strength in complex implicit reasoning. Notably, the lighter Tree-CoT-RT-7B variant remains competitive and often surpasses larger T5-based models, suggesting that the performance gains stem primarily from the reasoning framework rather than model size.

6 Ablation Experiments

6.1 Effect of Removing Tree-CoT

To examine the role of tree-guided reasoning, we remove the tree-structured path constraints and use a plain step-by-step CoT prompt to generate reasoning chains. This variant preserves CoT-style explanations but no longer enforces type-specific inference order for EAEO/EAIO/IAEO/IAIO.

Plain Step-by-Step CoT Template.

Let's think step by step. Given a sentence [Target Sentence], first identify and extract the relevant aspects. Determine the appropriate category for each extracted aspect. Based on the extracted opinions, infer the sentiment polarity. Finally,

Method	REST-ACOS (F1.)				LAPTOP-ACOS (F1.)			
	EAEO	IAEO	EAIO	IAIO	EAEO	IAEO	EAIO	IAIO
Double-Propagation (Cai et al., 2021)	26.0	N/A	N/A	N/A	9.8	N/A	N/A	N/A
JET-ACOS (Cai et al., 2021)	52.3	N/A	N/A	N/A	35.7	N/A	N/A	N/A
TAS-BERT-ACOS (Cai et al., 2021)	33.6	31.8	14.0	39.8	26.1	41.5	10.9	21.2
Extract-Classify (Cai et al., 2021)	45.0	34.7	23.9	33.7	35.4	39.0	16.8	18.6
PARAPHRASE-BART (Xiong et al., 2023)	38.6	37.8	16.7	38.5	31.3	38.9	21.1	35.6
GEN-NAT-SCL-BART (Xiong et al., 2023)	46.9	30.5	20.5	37.6	35.9	40.7	20.9	30.2
BART-CRN (Xiong et al., 2023)	54.1	50.6	18.9	42.9	38.9	54.3	24.5	40.7
BARTABSA(split) (Hoang et al., 2022)	58.5	43.9	20.0	42.9	39.9	52.8	23.4	29.8
PARAPHRASE (Peper and Wang, 2022)	65.4	53.3	45.6	45.6	45.7	51.0	33.0	39.6
GEN-SCL-NAT (Peper and Wang, 2022)	66.5	56.5	46.2	50.7	45.8	54.0	34.3	39.6
ASQP-ITSCL (T5-base) (Zhang et al., 2024b)	69.8	51.2	31.9	45.6	46.4	59.1	30.3	40.0
ASQP-ITSCL (T5-large) (Zhang et al., 2024b)	71.8	53.2	44.4	52.2	47.2	61.3	34.4	39.7
Tree-CoT-RT-7B (Ours)	73.9	55.8	46.4	54.6	49.0	63.9	35.7	43.6
Tree-CoT-RT-32B (Ours)	**74.2**	**57.6**	**47.5**	**56.1**	**50.5**	**65.4**	**38.1**	**45.8**

Table 4: Comparison of explicit/implicit sentiment analysis (EA, EO, IA, IO) on REST-ACOS and LAPTOP-ACOS. N/A denotes unsupported types.

Method	REST (F1)	LAPTOP (F1)
GEN-SCL-NAT	62.62	45.16
ASQP-ITSCL (T5-large)	64.86	46.11
S^2IT-32B	67.37	46.70
Tree-CoT-RT-LLaMA3-8B	68.42	48.38
Tree-CoT-RT-Qwen2.5-7B	69.50	49.10
Tree-CoT-RT-Qwen2.5-32B	**71.22**	**50.25**
Ablation Variants (Qwen)		
w/o Tree-CoT	68.54 (↓2.68)	46.56 (↓3.69)
w/o GRPO-RL	67.30 (↓3.92)	48.91 (↓1.34)
w/o CoT-RL	65.35 (↓5.87)	46.30 (↓3.95)
Ablation Variants (LLaMA)		
w/o Tree-CoT	67.52 (↓0.90)	46.15 (↓2.23)
w/o GRPO-RL	67.10 (↓1.32)	47.56 (↓0.82)
w/o CoT-RL	65.41 (↓3.01)	47.06 (↓1.32)

Table 5: Ablation of Tree-CoT-RT with LLMs.

generate the (aspect, category, opinion, sentiment) quadruples.

As shown in Table 5, performance consistently drops on both REST and LAPTOP, highlighting the importance of Tree-CoT path constraints for implicit ASQP.

6.2 Effect of Removing RL

To evaluate the effect of GRPO-based reinforcement learning, the Step 2 RL module is removed, and unfiltered reasoning chains from Step 1 are directly used for supervised fine-tuning. Table 5 shows a clear performance drop, confirming the necessity of reward-guided filtering for selecting high-quality reasoning paths.

6.3 Effect of Removing CoT and RL

To assess the impact of reasoning-based augmentation, both Step 1 (reasoning chain generation) and Step 2 (reinforcement learning) are removed. The model is directly fine-tuned on original human-annotated data without any generated reasoning supervision. As shown in Table 5, this leads to the most significant performance drop, highlighting the essential role of reasoning-guided data synthesis in improving generalization and robustness.

Our gains stem from structured reasoning rather than proprietary LLM outputs. While LLMs are used to initialize candidate chains, Tree-CoT supervision under GRPO enforces alignment with ASQP logic. Removing Tree-CoT significantly degrades performance under the same supervision, indicating that improvements arise from structural reasoning rather than data scaling. Importantly, our contribution lies in formalizing structured reasoning supervision at the task–reasoning interface for implicit ASQP, rather than introducing a new optimizer.

6.4 Ablation on Model Backbone

We compare Tree-CoT-RT using two instruction-tuned LLMs: Qwen2.5-32B and LLaMA3-8B. As shown in Table 5, Qwen2.5-32B achieves the best results across both datasets, slightly outperforming LLaMA3-8B. The strong performance of both variants confirms that Tree-CoT-RT is robust and that the gains primarily stem from the reasoning strategy rather than the backbone.

Reward Component	REST (F1)	LAPTOP (F1)
Full Reward (GRPO-RL)	**71.22**	**50.25**
w/o Consistency	68.55 (↓2.67)	47.24 (↓3.01)
w/o Validity	68.33 (↓2.89)	46.58 (↓3.67)
w/o Implicit Signal	65.83 (↓5.39)	45.25 (↓5.00)
w/o Diversity	67.32 (↓3.90)	47.34 (↓2.91)

Table 6: Impact of removing RL reward components.

6.5 Ablation on Reward Components

We ablate each reward term in the GRPO RL framework by removing one component at a time. As shown in Table 6, performance consistently drops across both datasets, confirming that all four components (consistency, reasoning validity, implicit alignment, and diversity) are essential for improving reasoning quality and overall performance. Notably, the removal of the implicit signal term results in the most significant performance drop among all components, underscoring its critical importance in modeling implicit sentiment.

7 Conclusions

This paper introduces Tree-CoT-RT, a novel framework for ASQP that addresses the core challenges of explainability and implicit sentiment understanding. Our method leverages four sentiment tree types to generate structured reasoning chains via tree-guided prompts, enabling LLMs to internalize both explicit and implicit inference patterns. Through a reinforcement learning mechanism (GRPO-RL) with multi-dimensional rewards, we effectively filter and refine reasoning samples to enhance data quality. The fine-tuned models not only predict sentiment quadruples accurately but also expose their underlying explainable reasoning process. Experiments on REST-ACOS and LAPTOP-ACOS show state-of-the-art results, especially on implicit cases.

8 Limitations

The study has the following limitations:

- **Single Turn Scope:** The current method is limited to single-sentence inputs. Extending sentiment trees to multi-turn dialogues could better capture context-dependent and dynamic emotional cues.
- **Reward Design and Alignment:** The reinforcement learning module relies on manually designed reward functions. Future work could explore data-driven or human aligned approaches, such as inverse reinforcement learning or reinforcement learning with human feedback (RLHF), to improve reasoning quality and diversity.
- **Lack of Multimodal Integration:** Our method only considers textual input. Incorporating visual or multimodal signals could enhance the model's ability to analyze sentiment in richer contexts such as image or video-based reviews.

Ethics Statement

We conducted all experiments on publicly available, ethically sourced, and anonymized datasets widely used in prior sentiment analysis research. Our work adheres to the highest ethical standards in AI and NLP, with a focus on fairness, transparency, and reproducibility. The proposed methods are designed solely for academic use and aim to improve ASQP performance without enabling misuse or harm. We used AI assistants such as ChatGPT for writing assistance. All AI-generated content was verified by the authors.

References

Xiaoyi Bao, Z Wang, Xiaotong Jiang, Rong Xiao, and Shoushan Li. 2022. Aspect-based sentiment analysis with opinion tree generation. *IJCAI 2022*, pages 4044–4050.

Jiahao Bu, Lei Ren, Shuang Zheng, Yang Yang, Jingang Wang, Fuzheng Zhang, and Wei Wu. 2021. Asap: A chinese review dataset towards aspect category sentiment analysis and rating prediction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2069–2079.

Hongjie Cai, Yaofeng Tu, Xiangsheng Zhou, Jianfei Yu, and Rui Xia. 2020. Aspect-category based sentiment analysis with hierarchical graph convolutional network. In *Proceedings of the 28th international conference on computational linguistics*, pages 833–843.

Hongjie Cai, Rui Xia, and Jianfei Yu. 2021. Aspect-category-opinion-sentiment quadruple extraction with implicit aspects and opinions. 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)*, pages 340–350, Online. Association for Computational Linguistics.

Bingfeng Chen, Chenjie Qiu, Yifeng Xie, Boyan Xu, Ruichu Cai, and Zhifeng Hao. 2025. $\mathcal{S}^2$IT: Stepwise syntax integration tuning for large language models in aspect sentiment quad prediction. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 6799–6806, Albuquerque, New Mexico. Association for Computational Linguistics.

Shaowei Chen, Yu Wang, Jie Liu, and Yuelin Wang. 2021. Bidirectional machine reading comprehension for aspect sentiment triplet extraction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 12666–12674.

Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Tao He, Haotian Wang, Weihua Peng, Ming Liu, Bing Qin, and Ting Liu. 2024. Navigate through enigmatic labyrinth a survey of chain of thought reasoning: Advances, frontiers and future. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1173–1203, Bangkok, Thailand. Association for Computational Linguistics.

DeepSeek. 2024. Deepseek r1: Generalist language agent with reasoning and action. https://deepseekcoder.github.io/blog/deepseek-r1. Accessed: 2025-05-19.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhifang Fan, Zhen Wu, Xinyu Dai, Shujian Huang, and Jiajun Chen. 2019. Target-oriented opinion words extraction with target-fused neural sequence labeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2509–2518.

Hao Fei, Bobo Li, Qian Liu, Lidong Bing, Fei Li, and Tat-Seng Chua. 2023. Reasoning implicit sentiment with chain-of-thought prompting. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1171–1182, Toronto, Canada. Association for Computational Linguistics.

Google DeepMind. 2024. Gemini 1.5 technical report. https://deepmind.google/technologies/gemini/#gemini-15. Accessed: 2025-05-19.

Zhibin Gou, Qingyan Guo, and Yujiu Yang. 2023. MvP: Multi-view prompting improves aspect sentiment tuple prediction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4380–4397, Toronto, Canada. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Cao Duy Hoang, Quang Vinh Dinh, and Ngoc Hong Tran. 2022. Aspect-category-opinion-sentiment extraction using generative transformer model. In *2022 RIVF International Conference on Computing and Communication Technologies (RIVF)*, pages 1–6. IEEE.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022a. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Mengting Hu, Yinhao Bai, Yike Wu, Zhen Zhang, Liqi Zhang, Hang Gao, Shiwan Zhao, and Minlie Huang. 2023. Uncertainty-aware unlikelihood learning improves generative aspect sentiment quad prediction. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13481–13494, Toronto, Canada. Association for Computational Linguistics.

Mengting Hu, Yike Wu, Hang Gao, Yinhao Bai, and Shiwan Zhao. 2022b. Improving aspect sentiment quad prediction via template-order data augmentation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7889–7900, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jieyong Kim, Ryang Heo, Yongsik Seo, SeongKu Kang, Jinyoung Yeo, and Dongha Lee. 2024. Self-consistent reasoning-based aspect-sentiment quad prediction with extract-then-assign strategy. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 7295–7303, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Jian Liu, Zhiyang Teng, Leyang Cui, Hanmeng Liu, and Yue Zhang. 2021. Solving aspect category sentiment analysis as a text generation task. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4406–4416, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jiangming Liu and Yue Zhang. 2017. Attention modeling for targeted sentiment. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 572–577, Valencia, Spain. Association for Computational Linguistics.

Yue Mao, Yi Shen, Jingchao Yang, Xiaoying Zhu, and Longjun Cai. 2022. Seq2Path: Generating sentiment tuples as paths of a tree. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2215–2225, Dublin, Ireland. Association for Computational Linguistics.

Yue Mao, Yi Shen, Chao Yu, and Longjun Cai. 2021. A joint training dual-mrc framework for aspect based sentiment analysis. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 13543–13551.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

OpenAI. 2025. Introducing GPT-5. https://openai.com/index/introducing-gpt-5/. Accessed: 2026-01-06.

Haiyun Peng, Lu Xu, Lidong Bing, Fei Huang, Wei Lu, and Luo Si. 2020. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8600–8607.

Joseph Peper and Lu Wang. 2022. Generative aspect-based sentiment analysis with contrastive learning and expressive structure. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6089–6095, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion word expansion and target extraction through double propagation. *Computational linguistics*, 37(1):9–27.

Qwen. 2025. Qwen3 technical report. https://arxiv.org/abs/2505.09388. *Preprint*, arXiv:2505.09388. Accessed: 2026-01-06.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Hai Wan, Yufei Yang, Jianfeng Du, Yanan Liu, Kunxun Qi, and Jeff Z. Pan. 2020. Target-aspect-sentiment joint detection for aspect-based sentiment analysis. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9122–9129.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 606–615, Austin, Texas. Association for Computational Linguistics.

Zengzhi Wang, Rui Xia, and Jianfei Yu. 2024. Unified absa via annotation-decoupled multi-task instruction tuning. *IEEE Transactions on Knowledge and Data Engineering*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Chao Wu, Qingyu Xiong, Hualing Yi, Yang Yu, Qiwu Zhu, Min Gao, and Jie Chen. 2021. Multiple-element joint detection for aspect-based sentiment analysis. *Knowledge-Based Systems*, 223:107073.

H. Xiong, Z. Yan, C. Wu, et al. 2023. Bart-based contrastive and retrospective network for aspect-category-opinion-sentiment quadruple extraction. *International Journal of Machine Learning and Cybernetics*, 14:3243–3255.

Lu Xu, Hao Li, Wei Lu, and Lidong Bing. 2020. Position-aware tagging for aspect sentiment triplet extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2339–2349, Online. Association for Computational Linguistics.

Yichun Yin, Furu Wei, Li Dong, Kaimeng Xu, Ming Zhang, and Ming Zhou. 2016. Unsupervised word and dependency path embeddings for aspect term extraction. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 2979–2985.

Hao Zhang, Yu-N Cheah, Osamah Mohammed Alyasiri, and Jieyu An. 2023a. A survey on aspect-based sentiment quadruple extraction with implicit aspects and opinions. Research Square, Preprint.

Hao Zhang, Yu-N Cheah, Osamah Mohammed Alyasiri, and Jieyu An. 2024a. Exploring aspect-based sentiment quadruple extraction with implicit aspects, opinions, and chatgpt: a comprehensive survey. *Artificial Intelligence Review*, 57(2):17.

Hao Zhang, Yu-N Cheah, Congqing He, and Feifan Yi. 2024b. An instruction tuning-based contrastive learning framework for aspect sentiment quad prediction with implicit aspects and opinions. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7698–7714, Miami, Florida, USA. Association for Computational Linguistics.

Wenxuan Zhang, Yang Deng, Xin Li, Yifei Yuan, Lidong Bing, and Wai Lam. 2021a. Aspect sentiment

quad prediction as paraphrase generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9209–9219, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2021b. Towards generative 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 2: Short Papers)*, pages 504–510, Online. Association for Computational Linguistics.

Wenyuan Zhang, Xinghua Zhang, Shiyao Cui, Kun Huang, Xuebin Wang, and Tingwen Liu. 2024c. Adaptive data augmentation for aspect sentiment quad prediction. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11176–11180.

Yice Zhang, Yifan Yang, Meng Li, Bin Liang, Shiwei Chen, and Ruifeng Xu. 2023b. Target-to-source augmentation for aspect sentiment triplet extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12165–12177, Singapore. Association for Computational Linguistics.

He Zhao, Longtao Huang, Rong Zhang, Quan Lu, and Hui Xue. 2020. Spanmlt: A span-based multi-task learning framework for pair-wise aspect and opinion terms extraction. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 3239–3248.

A Training Details, Hyperparameters, and Computational Cost

We provide detailed implementation settings for Step 1 (multi-path chain generation and tSFT), Step 2 (GRPO-based reinforcement learning), and Step 3 (reasoning-aware supervised fine-tuning). All local model updates are parameter-efficient via LoRA (Hu et al., 2022a) with a cosine learning-rate scheduler. Unless otherwise specified, the maximum sequence length is 2048, the learning rate is 5×10^{-5}, and we use NVIDIA A100 GPUs (80GB for Qwen2.5-32B, and 40GB for Qwen2.5-7B and LLaMA3-8B).

A.1 Step 1 (S1): Multi-Path Chain Generation and tSFT

S1-a: Multi-Path Chain Generation (External LLM APIs). For each sentence, we generate $K{=}10$ candidate reasoning chains using type-specific Tree-CoT prompts (EAEO/EAIO/IAEO/IAIO), together with prompt paraphrases and different random seeds to induce diversity. We use the same decoding configuration across prompt variants: temperature 0.7, top-p 0.95, and maximum generation length 2048 tokens. These API-generated chains are used *only* to construct the supervision data for tSFT and GRPO filtering.

Table 7: Hyperparameters for Step 1 tSFT (Qwen2.5-7B-Instruct).

Hyperparameter	Value
LoRA Rank	8
LoRA Alpha	16
LoRA Dropout	0.05
Batch Size	4
Gradient Accumulation	2
Max Sequence Length	2048
Learning Rate	5e-5
Training Steps	15
Optimizer	AdamW
Precision	bf16

S1-b: Tree-Supervised Fine-Tuning (tSFT). We perform a lightweight tree-supervised fine-tuning (tSFT) on Qwen2.5-7B-Instruct to obtain an instruction-following starting model for GRPO in Step 2. We freeze the backbone and update LoRA adapters only. Table 7 reports the hyperparameters.

A.2 Step 2 (S2): GRPO-Based Reinforcement Learning

In Step 2, we apply GRPO optimisation starting from the tSFT model obtained in Step 1. The policy model is LoRA-enabled and updated with group-relative advantages; the reference model is a frozen snapshot of the initial policy for KL regularisation. We use group size $G{=}10$ and run $K{=}10$ GRPO steps (following Section 3.4).

GRPO-Specific Settings. To avoid notation conflict with the reward weights, we denote the KL coefficient as λ_{KL}. Table 8 summarises the GRPO-specific settings.

A.3 Step 3 (S3): Reasoning-Aware Supervised Fine-Tuning

In Step 3, we fine-tune base models on the filtered high-quality reasoning chains and their gold-standard ACOS quadruples. We use Qwen2.5-7B, Qwen2.5-32B, and LLaMA3-8B as backbones and apply LoRA for parameter-efficient adaptation. Hyperparameters are reported in Tables 9–11.

Table 8: GRPO settings for Step 2 (Qwen2.5-7B-Instruct; LoRA policy).

Hyperparameter	Value
Group size G	10
GRPO steps K	10
Clip ϵ	0.2
KL coefficient λ_{KL}	0.01
Reward weights $(\lambda_{\text{cons}}, \lambda_{\text{valid}}, \lambda_{\text{impl}}, \lambda_{\text{div}})$	0.25
Policy LoRA Rank / Alpha	8 / 16
Policy LoRA Dropout	0.05
Batch Size	4
Gradient Accumulation	2
Max Sequence Length	2048
Learning Rate	5e-5
Optimizer	AdamW
Precision	bf16

Table 9: Hyperparameters for Step 3 (Qwen2.5-7B Base).

Hyperparameter	Value
LoRA Rank	32
LoRA Alpha	32
LoRA Dropout	0.1
Batch Size	4
Gradient Accumulation	2
Max Sequence Length	2048
Learning Rate	5e-5
Training Steps	10
Optimizer	AdamW
Precision	bf16

A.4 Computational Cost

Tree-CoT-RT introduces additional computation due to multi-path chain generation (S1), GRPO-based optimisation (S2), and self-consistent inference (S4). To quantify the overhead, we report wall-clock time and GPU usage for each stage under our main experimental setting (Section 4.3.1). Specifically, we measure: (1) **S1 Chain Generation:** average latency to generate $K{=}10$ chains per sentence, and the total number of API calls; (2) **S2 GRPO Training:** total GPU-hours, peak GPU memory, and training wall-clock time (LoRA updates only); (3) **S3 Fine-tuning:** total GPU-hours, peak GPU memory, and wall-clock time for LoRA fine-tuning on selected chains; (4) **S4 Inference:** per-sentence latency for single-sample decoding and self-consistent voting with $M{=}5$ samples.

Sampling protocol. All timings are averaged over 200 sentences randomly sampled from the *training set* and repeated with four seeds (444/555/666/777), following our main protocol. Table 12 summarises the cost breakdown and compares inference-time overhead against a strong baseline under the same backbone size.

Table 10: Hyperparameters for Step 3 (Qwen2.5-32B Base).

Hyperparameter	Value
LoRA Rank	32
LoRA Alpha	32
LoRA Dropout	0.1
Batch Size	2
Gradient Accumulation	4
Max Sequence Length	2048
Learning Rate	5e-5
Training Steps	10
Optimizer	AdamW
Precision	bf16

Table 11: Hyperparameters for Step 3 (LLaMA3-8B Base).

Hyperparameter	Value
LoRA Rank	16
LoRA Alpha	32
LoRA Dropout	0.1
Batch Size	4
Gradient Accumulation	2
Max Sequence Length	2048
Learning Rate	5e-5
Training Steps	10
Optimizer	AdamW
Precision	bf16

Practicality of Tree-CoT-RT. We clarify the latency and deployment trade-offs from three perspectives:

Flexible configuration (High-Precision vs. Fast Mode). The reported 12.0s latency corresponds to an upper-bound setting in High-Precision mode (M=5) using the Qwen2.5-32B model, optimized for maximum accuracy. In contrast, Fast Mode (M=1) reduces latency to 2.4s, which is comparable to strong baselines and suitable for real-time applications.

Impact of model scale (32B vs. 7B). The reported latency is based on a 32B model to push SOTA performance. For higher-throughput scenarios, the framework is equally compatible with smaller models (e.g., 7B), which can significantly reduce inference time to sub-second levels.

Teacher–generator role for distillation. We position the 32B Tree-CoT-RT as a reasoning teacher that generates high-quality reasoning traces offline, which can be distilled into smaller models. This enables compact models to inherit tree-guided reasoning capabilities while achieving millisecond-level inference speed.

Stage	Metric	7B	32B	Notes
S1-a (Gen.)	sec/sent (K=10)	18–22	19–23	API latency varies with output length and provider load.
S1-a (Gen.)	#calls/sent	10	10	One API call per chain; total fixed to K=10.
S1-b (tSFT)†	GPU-hours	6	6	Qwen2.5-7B-Instruct; LoRA only.
S1-b (tSFT)†	peak VRAM (GB)	36	36	A100 40GB; max len 2048.
S2 (GRPO)†	GPU-hours	24	24	LoRA policy on 7B-Instruct; 4×A100 parallel.
S2 (GRPO)†	peak VRAM (GB)	36	36	A100 40GB.
S2 (Select)†	sec/sent	0.03	0.03	Parse+reward+dedup on CPU; negligible GPU.
S3 (SFT)	GPU-hours	14	56	LoRA only; backbone-specific fine-tuning.
S3 (SFT)	peak VRAM (GB)	30	74	A100 40/80GB; max len 2048.
S4 (Infer)	sec/sent (M=1)	0.7	2.4	Single decode (reasoning + quad).
S4 (Infer)	sec/sent (M=5)	3.6	12.0	5 samples + voting.
Baseline	sec/sent (Infer)	0.4	1.6	Same backbone; single decode; no self-consistency.

Table 12: Computational cost breakdown of Tree-CoT-RT. "sec/sent" reports average latency per sentence. † indicates stages executed once on the 7B-Instruct pipeline (tSFT/GRPO/selection) and shared by both 7B and 32B backbones in Step 3.

B Multi-Path Reasoning and GRPO Optimisation Algorithm

Step 1-a (S1-a): API-based multi-path Tree-CoT chain generation. To construct diverse yet type-consistent reasoning traces, we generate multiple Tree-CoT chains per sentence using a type-specific tree prompt template and stochastic variations (e.g., paraphrased instructions and decoding seeds). Given an input sentence s and its ASQP type t, we call an external LLM API to sample K candidate chains, and parse each chain to extract its predicted quadruple set. This forms a candidate pool $(\mathcal{C}, \mathcal{Q})$ for subsequent tree-supervised fine-tuning and GRPO optimisation.

Algorithm 1 Step 1-a (S1-a): API-based Tree-CoT Prompting for Candidate Chain Generation

Require: Input sentence s, ASQP type t, number of chains K (default 10)
Ensure: Candidate chain pool $\mathcal{C}$ and parsed quads $\mathcal{Q}$

Initialize $\mathcal{C} \leftarrow \emptyset$, $\mathcal{Q} \leftarrow \emptyset$
Construct a type-specific Tree-CoT prompt template $\mathcal{P}_t$
for $i = 1$ to K **do**
  Instantiate prompt $p_i \leftarrow \textsc{Instantiate}(\mathcal{P}_t; \text{seed}_i)$
  Generate chain $c_i \leftarrow \textsc{LLMApiGenerate}(s, p_i)$ ▷ call external LLM API
  Parse quads $q_i \leftarrow \textsc{ExtractQuad}(c_i)$
  Append c_i to $\mathcal{C}$ and q_i to $\mathcal{Q}$
end for
return $\mathcal{C}, \mathcal{Q}$

Step 1-b (S1-b): Tree-supervised fine-tuning (tSFT) for RL initialisation. To obtain a stable and task-aligned starting policy for GRPO, we perform a lightweight tree-supervised fine-tuning (tSFT) on an instruction-tuned backbone (e.g., Qwen2.5-7B-Instruct). We build a supervised dataset from the candidate pool in Step 1-a by keeping chains that are (i) parsable into quads and (ii) structurally valid under the type-specific Tree-CoT constraints, and pairing them with the gold quadruple set. We then fine-tune LoRA adapters while freezing the backbone weights, producing an initial policy π_{θ_0} used to start GRPO.

Algorithm 2 Step 1-b (S1-b): Tree-Supervised Fine-Tuning (tSFT) for RL Initial Policy

Require: Training set $\mathcal{D} = \{(s, t, Y)\}$; candidate pool generator from Alg. 1; instruct backbone M
Ensure: Initial policy π_{θ_0} for GRPO

Initialize supervised set $\mathcal{D}_{\text{tSFT}} \leftarrow \emptyset$
for each $(s, t, Y) \in \mathcal{D}$ **do**
  Generate $(\mathcal{C}, \mathcal{Q}) \leftarrow \textsc{GenPool}(s, t)$ using Alg. 1
  for each chain $o \in \mathcal{C}$ **do**
    $\hat{Y} \leftarrow \textsc{ExtractQuad}(o)$
    if $\hat{Y}$ is parsable **and** $\textsc{CheckPathValidity}(o, t) = 1$ **then**
      Add training pair $\big((s, t) \rightarrow (o,\ Y)\big)$ to $\mathcal{D}_{\text{tSFT}}$
    end if
  end for
end for
Fine-tune LoRA adapters on M with $\mathcal{D}_{\text{tSFT}}$ (freeze backbone) to obtain π_{θ_0}
return π_{θ_0}

Step 2 (S2): GRPO optimisation and offline chain selection. Starting from the tSFT policy

π_{θ_0}, we apply GRPO to optimise the policy towards producing chains that are consistent with gold quads, structurally valid under Tree-CoT paths, and informative for implicit inference. During GRPO, we maintain a frozen reference policy π_{ref} (a snapshot of π_{θ_0}) and regularise updates via KL divergence. After training, we perform an offline selection step: for each sentence, we sample a group of chains from the optimised policy, compute rewards, keep top-N valid chains, and deduplicate them to form the final filtered chain set for Step 3 fine-tuning.

Algorithm 3 Step 2 (S2): GRPO for Tree-CoT Reasoning Chain Optimisation and Selection

Require: Dataset $\mathcal{D} = \{(s, t, Y)\}$; init policy π_{θ_0} (Alg. 2); ref policy $\pi_{\text{ref}} = \text{Freeze}(\pi_{\theta_0})$; group size G; optimisation steps T; keep top-N

Ensure: Optimised policy $\pi_{\theta^\star}$ and selected chains $\mathcal{G}^\star$

$\pi_\theta \leftarrow \pi_{\theta_0}$ ▷ LoRA-enabled policy
for $j = 1$ to T **do**
  Sample $(s, t, Y) \sim \mathcal{D}$ and build Tree-CoT prompt p
  Sample group $\mathcal{G} = \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot \mid p)$
  for $i = 1$ to G **do**
    $\hat{Y}_i \leftarrow \textsc{ExtractQuad}(o_i)$
    $R_i \leftarrow \textsc{Reward}(o_i, \hat{Y}_i, Y, \mathcal{G}, t)$
  end for
  Compute group-normalized advantages $\hat{A}_i$ from $\{R_i\}_{i=1}^{G}$
  Update π_θ by GRPO objective with clipping and $\text{KL}(\pi_\theta \,\|\, \pi_{\text{ref}})$
  $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$
end for
$\pi_{\theta^\star} \leftarrow \pi_\theta$
Offline selection: for each (s, t), sample G chains from $\pi_{\theta^\star}$, keep top-N valid by reward/advantage, deduplicate $\Rightarrow \mathcal{G}^\star$
return $\pi_{\theta^\star}, \mathcal{G}^\star$

C Reward Computation

This section is organized into two parts: we first define each reward component in detail, and then analyze the robustness of the reward design under different weight configurations.

C.1 Reward Formulation

To improve reproducibility, we provide explicit definitions for each reward component in Eq. (2), using the terminology in Table 6. Let Y denote the gold quadruple set for sentence s, and let $\hat{Y}(o)$ be the quadruple set parsed from a candidate chain o.

Consistency (R_{cons}). We score element-wise label alignment between $\hat{Y}(o)$ and Y over (A, C, O, S). For the single-quad case, we compute:

$$R_{\text{cons}}(o) = \sum_{e \in \{A,C,O,S\}} w_e \cdot \mathbb{I}[\hat{y}_e(o) = y_e], \quad \sum_e w_e = 1. \tag{10}$$

where we use $w_A{=}w_C{=}w_O{=}w_S{=}0.25$ by default. For multi-quad sentences, we match predicted and gold quads by maximum bipartite matching (exact match on (A,C,O,S)) and average the per-quad scores.

Validity (R_{valid}). We verify whether the reasoning steps in o conform to the type-specific Tree-CoT path of its ASQP type (Table 1 and Figure 4). We implement a rule-based checker that enforces: (i) required step coverage (e.g., identifying an explicit anchor before inferring implicit elements), (ii) allowable step order for the given type, and (iii) absence of contradictions (e.g., assigning two different sentiments to the same quadruple). $R_{\text{valid}}(o)$ is the normalized count of satisfied constraints, mapped to $[0, 1]$.

Implicit Signal (R_{impl}). For types involving implicit elements (EAIO/IAEO/IAIO), we reward chains that justify implicit inference using sentence-grounded cues. Concretely, we require that the chain explicitly links at least one textual cue span (e.g., complaint events, negative outcomes, or service-failure descriptions) to the inferred implicit aspect/opinion. $R_{\text{impl}}(o)$ is the normalized count of satisfied cue-link rules. For EAEO, we set $R_{\text{impl}}(o){=}0$.

Diversity (R_{div}). To discourage degenerate near-duplicate chains, we measure dissimilarity between o and the template-chain set T_{temp} generated in Step 1. We compute

$$R_{\text{div}}(o) = 1 - \max_{u \in T_{\text{temp}}} \textsc{Sim}(o, u), \tag{11}$$

where $\textsc{Sim}(\cdot)$ can be token-level similarity (e.g., normalized BLEU) or embedding cosine similarity. This encourages stylistic variation while maintaining type-consistent reasoning.

Final Reward. We unify Eq. (2) with Table 6 as:

$$R(o) = \lambda_{\text{cons}} R_{\text{cons}}(o) + \lambda_{\text{valid}} R_{\text{valid}}(o) + \lambda_{\text{impl}} R_{\text{impl}}(o) + \lambda_{\text{div}} R_{\text{div}}(o), \quad \sum \lambda_\cdot = 1. \tag{12}$$

The weights balance label accuracy, structural validity, implicit reasoning strength, and diversity.

C.2 Sensitivity Analysis of Reward Weights

To evaluate the robustness of the reward design, we conduct a sensitivity analysis on the reward weights under the constraint $\sum \lambda = 1$. Starting from the default setting where all weights are uniformly assigned ($\lambda_{\text{cons}} = \lambda_{\text{valid}} = \lambda_{\text{impl}} = \lambda_{\text{div}} = 0.25$), we vary each component by $\pm 10\%$ and proportionally redistribute the remaining weights to maintain the normalization constraint.

Table 13 reports the results on the REST-ACOS dataset. Overall, the framework demonstrates strong robustness, with only minor fluctuations in F1 score. Among all components, R_{impl} exhibits the highest sensitivity, which is consistent with its critical role in modeling implicit sentiment reasoning. However, even in this case, the performance drop remains limited (maximum ΔF1 = 1.38), indicating stable behavior.

These results suggest that the proposed reward formulation is not overly sensitive to specific weight configurations and can generalize well without requiring fine-grained manual tuning.

Component	Variation	F1	ΔF1
Default	All = 0.25	71.22	–
Consistency	$\pm 10\%$	70.81 / 71.32	-0.41 / +0.10
Validity	$\pm 10\%$	70.75 / 71.29	-0.47 / +0.07
Implicit	$\pm 10\%$	69.84 / 71.48	-1.38 / +0.26
Diversity	$\pm 10\%$	70.96 / 71.34	-0.26 / +0.12

Table 13: Sensitivity analysis of reward weights on the REST-ACOS dataset.

D Reasoning Chain Examples for Sentiment Quadruples

To better illustrate how our Tree-CoT framework performs explainable reasoning for different sentiment quadruple types, Table 14 provides representative examples of reasoning chains for four ASQP configurations: EAEO (Explicit Aspect, Explicit Opinion), EAIO (Explicit Aspect, Implicit Opinion), IAEO (Implicit Aspect, Explicit Opinion), and IAIO (Implicit Aspect, Implicit Opinion).

For each case, the table shows the input sentence, the final extracted quadruple, and a step-by-step reasoning chain derived from the corresponding sentiment tree. These reasoning steps reflect how explicit or implicit elements are progressively identified or inferred, and how they lead to the construction of the complete quadruple.

This structured approach demonstrates the model's ability to explain its predictions, offering transparency for both explicit and implicit sentiment elements.

E Error Analysis and Case Study

To better understand the limitations of our Tree CoT RT framework, we conduct a detailed error analysis across the four ASQP types on the REST ACOS dataset. As shown in Figure 5, most errors are concentrated in types involving implicit elements, with IAIO being the most error prone. To further illustrate these trends, Table 15 presents four representative case studies, one for each ASQP type, highlighting common failure patterns such as missed implicit elements, sentiment misclassification, and incomplete extraction in multi quad scenarios. These analyses shed light on the challenges Tree CoT RT faces when processing structurally complex or sentiment rich inputs.

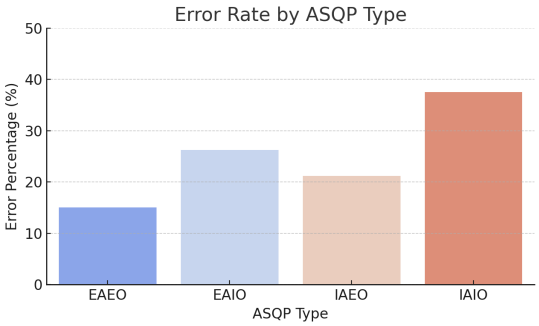

Figure 5: Error distribution by ASQP type. IAIO accounts for the highest proportion of errors, highlighting the challenge of jointly inferring implicit aspects and opinions.

F CoT-RL Reward Scoring Examples

Table 16 presents a comparative analysis of Tree-CoT reasoning chains for an IAIO-type sentence, showcasing both supervised and unsupervised generation strategies. Each reasoning chain is evaluated across four reward dimensions: Match, Logic, Implicit, and Diversity.

The supervised chain (Chain 2) is guided by the gold-standard label and successfully identifies the implicit aspect and opinion, producing a complete and accurate quadruple. It achieves strong scores in all dimensions, particularly in Logic (1.4), indicating that the reasoning steps are coherent and well-aligned with the input semantics. Notably, in the Implicit dimension, Chain 2 scores 1.4, reflecting its ability to reason beyond surface cues and reconstruct the implicit sentiment trajectory.

While generated under supervision, the reasoning is not a mere copy of the gold label—it justifies why the sentiment is negative by referencing the cumulative service failures such as "no greeting" and "long wait," showing good explainability in natural language.

However, its Match score (1.3) is slightly lower than that of Chain 4 (unsupervised), which achieves 1.5. This is because supervised chains often follow templated forms that, while accurate, may not align as naturally with the sentence's surface structure. In contrast, high-quality unsupervised chains like Chain 4 tend to generate more flexible and semantically attuned reasoning paths, sometimes scoring higher in matching even without explicit label guidance.

Chain 4 also performs well in the Implicit dimension (1.5), reinforcing its ability to capture latent aspects and opinions. Additionally, it receives a Diversity score of 1.5 compared to 1.3 for Chain 2, indicating that its reasoning structure is more varied and expressive—an important factor for improving model generalization in training.

In contrast, Chains 1 and 3 exhibit common failure patterns. Although neither generates the correct quadruple, both receive low Match scores (0.4), reflecting serious semantic misalignment. Chain 1 mistakenly focuses on "waited" and "over 15 minutes" as explicit surface cues, misinterpreting the sentence as commentary on service speed. Chain 3 erroneously extracts "door" and "no one" and links them to "service ambiance," showing disconnection from the actual service-related dissatisfaction.

In summary, this analysis demonstrates that the CoT-RL filtering mechanism is capable of evaluating the structural validity, semantic depth, and expressive variation of reasoning chains. It enables selection of high-quality examples for training, particularly in the challenging task of implicit sentiment extraction.

Table 14: Examples of reasoning chains for different ASQP types based on the Sentiment Tree-CoT logic.

ASQP Type	Reasoning Chain Generation Example
EAEO	**Sentence:** Judging from previous posts this used to be a good place, but not any longer. **Quadruple:** ["place", "restaurant general", "negative", "not any longer"] **Reasoning:** Explicit Aspect Identification: The sentence explicitly mentions the aspect "place". Explicit Opinion Identification: The phrase "not any longer" directly expresses an opinion about the "place", especially following "used to be a good place". Sentiment Derivation: The contrast "used to be a good place, but not any longer" clearly indicates a negative sentiment towards the current state of the "place". Category Derivation: The aspect "place" in a restaurant review context belongs to the "restaurant general" category. Therefore, the quadruple (Aspect: "place", Category: "restaurant general", Sentiment: "negative", Opinion: "not any longer") is extracted.
IAIO	**Sentence:** They never brought us complimentary noodles, ignored repeated requests for sugar, and threw our dishes on the table. **Quadruple:** ["NULL", "service general", "negative", "NULL"] **Reasoning:** Contextual Understanding for Implicit Aspect: The series of actions – "never brought us complimentary noodles", "ignored repeated requests for sugar", and "threw our dishes on the table" – collectively describe poor service. Although no specific aspect term is given for this quadruple (Aspect is "NULL"), the context points to an unstated subject related to the overall service experience (e.g., the staff, the way service was conducted). Contextual Understanding for Implicit Opinion: The described negative actions (e.g., "ignored", "threw dishes") implicitly convey a strong negative opinion about the service. No explicit opinion phrase is given (Opinion is "NULL"); the opinion is inferred from these actions. Sentiment Derivation: The implicitly understood poor service (aspect) and the implicitly conveyed negative treatment (opinion) result in a negative sentiment. Category Derivation: The context of these service-related failures points to the "service general" category. Therefore, the quadruple (Aspect: "NULL", Category: "service general", Sentiment: "negative", Opinion: "NULL") is extracted.
IAEO	**Sentence:** After all that, they complained to me about the small tip. **Quadruple:** ["NULL", "service general", "negative", "complained"] **Reasoning:** Explicit Opinion Identification: The sentence explicitly states the opinion "complained". Implicit Aspect Inference: The action "complained to me about the small tip" is performed by "they". In a restaurant context, "they" implicitly refers to the staff or service personnel. Thus, the aspect (who performed the action of complaining) is implicit. Sentiment Derivation: The opinion "complained" (by staff about a tip) expresses a negative sentiment from the customer's viewpoint regarding the service interaction. Category Derivation: The implicit aspect (staff/service) and the explicit opinion ("complained" about tip) relate to the "service general" category. Therefore, the quadruple (Aspect: "NULL", Category: "service general", Sentiment: "negative", Opinion: "complained") is extracted.
EAIO	**Sentence:** I've asked a cart attendant for a lotus leaf wrapped rice and she replied back rice and just walked away. **Quadruple:** ["cart attendant", "service general", "negative", "NULL"] **Reasoning:** Explicit Aspect Identification: The sentence explicitly mentions the aspect "cart attendant". Implicit Opinion Inference: The cart attendant's actions are described as "she replied back rice and just walked away" after a request. This behavior, in a service context, implicitly conveys an unhelpful or dismissive opinion about the attendant's performance. The opinion word itself is "NULL", but the negative nature of the service interaction is inferred from her actions. Sentiment Derivation: The explicit aspect "cart attendant" combined with the contextually inferred negative actions/opinion results in a negative sentiment. Category Derivation: The aspect "cart attendant" is a specific instance of service personnel, falling under the "service general" category. Therefore, the quadruple (Aspect: "cart attendant", Category: "service general", Sentiment: "negative", Opinion: "NULL") is extracted.

Table 15: Representative Failure Cases Across ASQP Types

ASQP Type: IAIO	**Input:** They left us waiting for 20 minutes, didn't apologize, and threw the bill on the table. **Ground Truth:** (Aspect: `NULL`, Opinion: `NULL`, Category: `service general`, Sentiment: `negative`) **Predicted:** (Aspect: "waiting", Opinion: "20 minutes", Category: `service general`, Sentiment: `neutral`) **Error:** Misinterpreted "20 minutes" as a neutral time mention; failed to capture negative cues from "didn't apologize" and "threw the bill".
ASQP Type: EAIO	**Input:** The waiter kept interrupting us while we were ordering. **Ground Truth:** (Aspect: "waiter", Opinion: `NULL`, Category: `service general`, Sentiment: `negative`) **Predicted:** (Aspect: "waiter", Opinion: "interrupting", Category: `service general`, Sentiment: `neutral`) **Error:** Although "interrupting" implies poor service, model failed to infer negative sentiment; reveals difficulty in interpreting implicit opinions.
ASQP Type: IAEO	**Input:** They charged extra without telling us. Really unfair. **Ground Truth:** (Aspect: `NULL`, Opinion: "unfair", Category: `price general`, Sentiment: `negative`) **Predicted:** (Aspect: "charge", Opinion: "unfair", Category: `service general`, Sentiment: `neutral`) **Error:** Although the opinion "unfair" is explicitly expressed, the model incorrectly anchors it to the verb "charge" as an explicit aspect, missing the fact that the user's dissatisfaction is implicitly targeted at the overall pricing policy. This leads to incorrect aspect detection and sentiment misclassification.
ASQP Type: EAEO	**Input:** The pasta was hot and fresh, but the sauce was bland. **Ground Truth:** 1. (Aspect: "pasta", Opinion: "hot and fresh", Category: `food quality`, Sentiment: `positive`) 2. (Aspect: "sauce", Opinion: "bland", Category: `food quality`, Sentiment: `negative`) **Predicted:** (Aspect: "pasta", Opinion: "hot and fresh", Category: `food quality`, Sentiment: `positive`) **Error:** The model correctly captured the first quadruple but failed to extract the second one, despite the opinion term "bland" and its aspect "sauce" being explicit. This demonstrates a typical failure in multi-quad scenarios, where the model tends to focus on the first opinion and overlook subsequent sentiment expressions, especially in contrastive clauses.

Table 16: Component-wise Tree-CoT Reasoning Chains and Reward Scores for IAIO Type

Input Sentence	No one greeted us at the door, and we waited over 15 minutes before getting water.
Ground Truth	(Aspect: `NULL`, Opinion: `NULL`, Category: `service general`, Sentiment: `negative`)
Target Type	IAIO (Implicit Aspect, Implicit Opinion)
Chain 1	**Generated Quadruple:** (Aspect: "waited", Opinion: "over 15 minutes", Category: `service speed`, Sentiment: "neutral") **Reasoning:** *Aspect Identification:* The model treats "waited" as an explicit aspect. *Opinion Identification:* "Over 15 minutes" is interpreted as a descriptive opinion. *Sentiment Derivation:* The duration is considered neutral. *Category Derivation:* Mapped to the "service speed" category based on surface semantics. *Conclusion:* The model captures only shallow cues, resulting in a semantically misaligned quadruple. **Reward Scores:** Match: 0.4, Logic: 0.5, Implicit: 0.4, Diversity: 0.3 **Total: 1.6**
Chain 2	**Generated Quadruple:** (Aspect: `NULL`, Opinion: `NULL`, Category: `service general`, Sentiment: `negative`) **Reasoning:** *Implicit Aspect Inference:* The lack of greeting and extended waiting time imply poor service behavior. *Implicit Opinion Inference:* No opinion words appear, but the customer experience conveys clear dissatisfaction. *Sentiment Derivation:* The cumulative negative events indicate a strong negative sentiment. *Category Derivation:* Contextually aligned with the "service general" category. *Conclusion:* All elements match the gold label, forming a high-quality reasoning chain. **Reward Scores:** Match: 1.3, Logic: 1.4, Implicit: 1.4, Diversity: 1.0 **Total: 5.1**
Chain 3	**Generated Quadruple:** (Aspect: "door", Opinion: "no one", Category: `service ambiance`, Sentiment: "neutral") **Reasoning:** *Aspect Identification:* Extracts "door" from the phrase "at the door" as the aspect. *Opinion Identification:* Treats "no one" as a negated presence opinion. *Sentiment Derivation:* Interprets lack of greeting as neutral due to absence of emotion-bearing terms. *Category Derivation:* Links to "service ambiance" based on location. *Conclusion:* A plausible chain with surface-level reasoning that misses implicit cues. **Reward Scores:** Match: 0.4, Logic: 0.6, Implicit: 0.5, Diversity: 0.3 **Total: 1.8**
Chain 4	**Generated Quadruple:** (Aspect: `NULL`, Opinion: `NULL`, Category: `service general`, Sentiment: `negative`) **Reasoning:** *Implicit Aspect Inference:* The sentence describes two sequential service failures: lack of initial acknowledgment and prolonged delay. These collectively imply poor service experience. *Implicit Opinion Inference:* The negative implication arises not from emotional words but from cumulative neglect—waiting, being ignored, lacking attentiveness. *Sentiment Derivation:* The combination of these events evokes frustration and dissatisfaction, thus a negative sentiment is justified. *Category Derivation:* Since both events refer to staff behavior and hospitality quality, the category is assigned as "service general." *Conclusion:* Though unsupervised, the model produces a logically coherent, structurally diverse reasoning path while arriving at the correct quadruple. **Reward Scores:** Match: 1.5, Logic: 1.3, Implicit: 1.5, Diversity: 1.5 **Total: 5.8**