STAR-OPD: Structured Aspect-Cascade-Aware On-Policy Reward Distillation for ABSA Quadruple Extraction

Tong Sun[1], **Mingyang Ma**[1], **Jiayang Yu**[1]
[1]Alibaba International Digital Commerce Group
st422019@alibaba-inc.com
mingyang.mmy@lazada.com
yujiayang.jy@alibaba-inc.com

Abstract

Aspect-based sentiment analysis (ABSA) quadruple extraction requires jointly predicting target, aspect, opinion, and sentiment over reviews that often contain multiple fine-grained sentiment tuples. While large chain-of-thought (CoT) models perform well on this task, distilling them into smaller deployable models remains difficult. We identify a task-specific failure mode in distilled ABSA extraction: student errors at the target–aspect interface create structurally invalid states, such as broken target–aspect bindings and hallucinated targets, which then corrupt downstream predictions. Conventional off-policy distillation is poorly suited to this setting because it trains only on teacher-generated trajectories and provides little supervision on the student-induced structural states that dominate inference. To address this mismatch, we propose STAR-OPD (**ST**ructured **A**spect-cascade-aware **O**n-**P**olicy **R**eward **D**istillation), which builds on generic on-policy distillation and instantiates it for ABSA quadruple extraction with cascade-aware, set-structured rewards. STAR-OPD trains on student rollouts and applies set-structured rewards that directly target binding consistency, target grounding, and fine-grained aspect disambiguation. Experiments on E-ABSA20K and SemEval-2014 show that STAR-OPD consistently outperforms off-policy and general on-policy baselines, reduces target hallucination, and substantially improves performance on structurally hard cases. With Qwen3-4B, STAR-OPD substantially narrows the student–teacher gap while improving inference efficiency, highlighting the importance of on-policy structural correction for distilled ABSA extraction.

1 Introduction

Aspect-based sentiment analysis (ABSA) quadruple extraction aims to identify structured sentiment tuples of the form $(target, aspect, opinion, sentiment)$. Compared with simpler ABSA settings, it is substantially more challenging: each review may contain multiple tuples, targets may refer to products or product-parts, and aspects are selected from fine-grained domain taxonomies. Recent chain-of-thought (CoT) large language models (LLMs) perform strongly on this task, but their inference cost remains prohibitive for high-throughput e-commerce applications, motivating distillation into smaller deployable models.

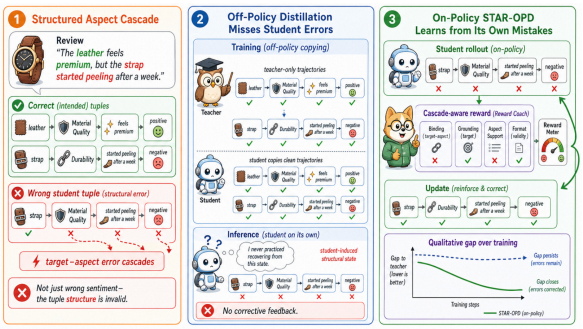

Figure 1: Motivation for STAR-OPD. **Left:** Errors at the target–aspect interface create structurally invalid tuples and corrupt downstream prediction. **Center:** Off-policy distillation trains only on teacher trajectories and misses student-induced cascade failures. **Right:** On-policy training exposes these states and uses cascade-aware rewards to correct them.

We identify a task-specific failure mode in distilled ABSA extraction, which we call *structured aspect cascade*: errors at the target–aspect interface create structurally invalid tuple states, most commonly through broken bindings and hallucinated non-grounded targets. In a pilot study with a SeqKD distillation baseline, we observe a substantial drop from target-only to target–aspect correctness, suggesting that the main bottleneck lies not in target identification alone, but in preserving valid target–aspect structure. For example, in "The leather looks premium, but the strap feels cheap,"

misassigning *strap* to the wrong aspect or hallucinating an unseen product-part can invalidate the entire tuple even if local sentiment words remain plausible.

Conventional *off-policy* distillation is ill-suited to this problem because it supervises the student only on teacher-generated trajectories, which are overwhelmingly structurally valid. At inference time, however, the student must condition on its own predictions, including incorrect target–aspect bindings and hallucinated targets rarely observed during training. Because such errors alter tuple structure rather than only local token accuracy, off-policy imitation provides little direct supervision for recovering from the student-induced states that dominate inference-time failure.

To address this mismatch, we propose STAR-OPD (**ST**ructured **A**spect-cascade-aware **O**n-**P**olicy **R**eward **D**istillation), which builds on generic on-policy distillation and specializes it for ABSA quadruple extraction. STAR-OPD trains on student rollouts rather than teacher-only trajectories and applies cascade-aware rewards that directly target incorrect target–aspect binding and hallucinated non-grounded targets, with lightweight filtering and sampling as stabilizers.

Experiments on E-ABSA20K (Sun et al., 2026) and SemEval-2014 (Pontiki et al., 2014) show that STAR-OPD consistently improves over both off-policy baselines and a strong general on-policy baseline. With Qwen3-4B, it reduces target hallucination from 9.75% to 7.22% and yields the largest gains on structurally hard reviews, while also improving deployment efficiency.

Contributions:

- We identify *structured aspect cascade*, a task-specific failure mode in distilled ABSA extraction centered on target–aspect binding and target hallucination.
- We propose STAR-OPD, an instantiation of generic on-policy distillation for ABSA quadruple extraction with cascade-aware rewards for binding consistency, target grounding, and aspect disambiguation.
- We show consistent gains over off-policy and generic on-policy baselines on E-ABSA20K and SemEval-2014, especially on structurally hard cases.

2 Related Work

Aspect-Based Sentiment Analysis. ABSA has progressed from aspect-level classification (Pontiki et al., 2014) to triplet (Peng et al., 2020) and quadruple extraction (Zhang et al., 2021; Cai et al., 2021). Recent generative and LLM-based methods (Scaria et al., 2024; Wang et al., 2023; Hu et al., 2022) perform well, but existing quadruple extraction work does not study the structural dependency and distillation-specific error propagation issues that arise in multi-quadruple reviews.

LLM Knowledge Distillation. SeqKD (Kim and Rush, 2016) distills teacher outputs off-policy and suffers from train–test mismatch. Recent on-policy methods address this by optimizing on student-generated sequences, including MiniLLM (Gu et al., 2024), GKD (Agarwal et al., 2024), and DistiLLM (Ko et al., 2024). Subsequent work further extends this framework, including G-OPD (Yang et al., 2026), entropy-aware distillation (Jin et al., 2026), and RLKD (Xu et al., 2026); CoT distillation (Magister et al., 2023; Ho et al., 2023) transfers reasoning traces but has not been studied for structured extraction. However, these methods are designed for general text generation and optimize generic sequence-level objectives, without modeling field-level binding constraints or dependency structure in quadruple extraction. Rather than proposing a new generic on-policy distillation principle, our method instantiates this family for ABSA quadruple extraction with set-structured, cascade-aware rewards that provide task-specific structural credit assignment for binding consistency, target grounding, and fine-grained aspect disambiguation.

Error Propagation in Structured Prediction. Prior work on exposure bias (Bengio et al., 2015) and error propagation in structured prediction (Finkel et al., 2006) has shown that training–inference mismatch can amplify upstream mistakes. We extend this perspective to quadruple extraction distillation, where errors at the target–aspect interface act as a structural bottleneck that corrupts downstream fields within a single quadruple.

3 Problem Analysis

3.1 Task Structure and Failure Mode

Given a review text $\mathbf{x}$, the goal of ABSA quadruple extraction is to predict a set of sentiment quadruples $\mathcal{Q}(\mathbf{x}) = \{(t_i, a_i, o_i, s_i)\}_{i=1}^{K}$, where t_i is either

a generic product tag or a target mention grounded in the review text, $a_i \in \mathcal{C}$ is a fine-grained aspect category from a closed taxonomy, o_i is an opinion span, and $s_i \in \{\text{pos}, \text{neu}, \text{neg}\}$ is sentiment polarity. Quadruple extraction is structurally challenging because each review may contain multiple tuples and the fields are semantically interdependent. Among these fields, the target–aspect interface is especially critical because it determines how downstream opinion and sentiment should be interpreted within each tuple. Detailed label definitions, target hierarchy, and dataset statistics are provided in Appendix A.

To understand where distilled models fail on this task, we conduct a pilot diagnostic comparison between our strong CoT teacher and a smaller direct model on E-ABSA20K-Hard.[1] We find that the smaller model is already close to the teacher when evaluation considers target identification alone (91.0% vs. 97.6%), a gap of only 6.6 percentage points. However, once correctness requires preserving each target together with its aspect, the gap widens sharply to 30.5 points (50.6% vs. 81.1%). The same pattern continues downstream: when sentiment must also be correct, the gap further increases to 36.1 points (42.5% vs. 78.6%), and full quadruple correctness remains 36.3 points lower for the smaller model (40.0% vs. 76.3%). Viewed relatively, the smaller model retains 93.2% of teacher performance for target identification alone, but only 62.4% once correct target–aspect binding is required. This indicates that the main bottleneck is not target grounding itself, but maintaining valid target–aspect structure under fine-grained aspect ambiguity.

The smaller model also exhibits a noticeably higher target hallucination rate, showing that the problem is not merely choosing the wrong aspect for an existing target, but also entering structurally invalid states with fabricated non-grounded targets. We refer to this failure pattern as *structured aspect cascade*. It is centered on the target–aspect interface and manifests primarily as broken target–aspect bindings and hallucinated targets. Once such an invalid state is entered, downstream fields are conditioned on an incorrect structural interpretation of the tuple, which reduces full-quadruple correctness even when local sentiment expressions remain plausible.

A second pilot with a SeqKD baseline exhibits the same pattern: target identification remains relatively strong, but performance drops sharply once target–aspect correctness is required, and 8.1% of correctly identified target–aspect pairs still receive an incorrect sentiment label.

Together, these observations suggest that the dominant failure mode of distilled ABSA extraction is not generic sequence noise, but structural corruption at the target–aspect interface. All values above are micro-averaged F1 unless otherwise specified; detailed statistics and metric definitions are given in Appendix B.

3.2 Why Off-Policy Distillation Is Insufficient

Off-policy distillation supervises the student on teacher-generated trajectories, which are overwhelmingly structurally valid. At inference time, however, the student conditions on its own predictions and may enter states with incorrect target–aspect bindings or hallucinated targets. Because these student-induced structural states are largely absent during training, off-policy imitation provides little direct signal for correcting them. This mismatch is especially harmful in ABSA quadruple extraction, where an early structural mistake changes the semantic interpretation of the remaining fields rather than merely degrading local token accuracy. An effective solution therefore needs both to expose the student to its own trajectories and to provide feedback at the level of tuple structure, especially around the target–aspect interface.

4 Method

4.1 Overview

To address the off-policy mismatch identified in Section 3.2, we propose STAR-OPD, which builds on generic on-policy distillation by training on student rollouts and applying structured aspect-cascade-aware rewards at the tuple level, especially for incorrect target–aspect bindings and hallucinated targets.

In practice, we first prepare a high-quality CoT teacher for pseudo-label generation and filtering, then optimize the student on-policy using student rollouts, reverse-KL distillation, and structure-aware rewards, with difficulty-aware sampling and replay as lightweight stabilizers (Appendix D, E).

[1] The teacher here is the same high-quality 32B model later used for pseudo-label generation and distillation; preparation details are given in Appendix D. The smaller direct model is a 4B direct baseline trained without on-policy distillation.

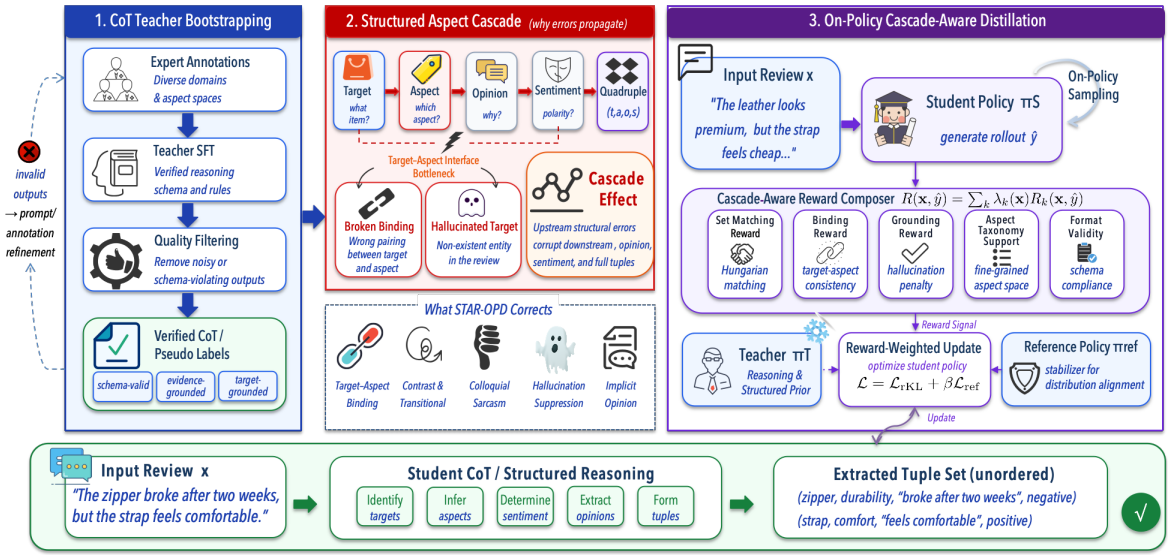

Figure 2: Overview of STAR-OPD. We first prepare a high-quality CoT teacher and generate filtered pseudo-labels for quadruple extraction. We then train the student on-policy under its own rollouts, using cascade-aware rewards to correct target–aspect binding errors and hallucinated targets.

4.2 On-Policy Cascade-Aware Distillation

Let π_T denote the teacher policy, π_S the student policy, and π_{ref} the frozen reference policy initialized from the student before on-policy updates. Given an input review $\mathbf{x}$, standard off-policy distillation optimizes the student only on teacher-generated trajectories. In STAR-OPD, following generic on-policy distillation, the student is optimized on its own sampled outputs:

$$\hat{y} \sim \pi_S(\cdot \mid \mathbf{x}).$$

This exposes training to the same student-induced and often structurally invalid states that occur at inference time, including incorrect target–aspect bindings and hallucinated targets.

We optimize the student with a reward-weighted on-policy distillation objective:

$$\mathcal{L} = \mathcal{L}_{\mathrm{rKL}} + \beta\mathcal{L}_{\mathrm{ref}}, \tag{1}$$

where $\alpha(\mathbf{x},\hat{y}) = w(R(\mathbf{x},\hat{y}))$ denotes the reward-derived rollout weight,

$$\mathcal{L}_{\mathrm{rKL}} = \mathbb{E}_{\hat{y}\sim\pi_S}\left[\alpha(\mathbf{x},\hat{y}) \log \frac{\pi_S(\hat{y} \mid \mathbf{x})}{\pi_T(\hat{y} \mid \mathbf{x})}\right], \tag{2}$$

and

$$\mathcal{L}_{\mathrm{ref}} = \mathbb{E}_{\hat{y}\sim\pi_S}\left[\log \frac{\pi_S(\hat{y} \mid \mathbf{x})}{\pi_{\mathrm{ref}}(\hat{y} \mid \mathbf{x})}\right]. \tag{3}$$

Here $R(\mathbf{x},\hat{y})$ is a cascade-aware structural reward, and $w(\cdot)$ maps the batch-normalized reward to a non-negative rollout weight (the exact transformation is given in Appendix E.3), and β controls the strength of reference regularization. Intuitively, higher-reward rollouts receive stronger teacher-aligned updates, while low-reward rollouts contribute less. The reverse-KL term favors structurally valid teacher-supported outputs, and the reference regularizer stabilizes on-policy updates.

On-policy exposure alone, however, does not specify which structural property of a rollout should be corrected. We therefore define a cascade-aware reward over student outputs:

$$R(\mathbf{x},\hat{y}) = \sum_k \lambda_k(\mathbf{x}) R_k(\mathbf{x},\hat{y}), \tag{4}$$

where each component reward targets a distinct structural failure mode of quadruple extraction. In practice, we normalize the raw reward within each batch and transform it into a non-negative rollout weight. This design turns on-policy exposure into targeted structural correction: the student is trained on its own trajectories, but updates are biased toward rollouts that better preserve target–aspect consistency and target grounding.

4.3 Cascade-Aware Rewards

Our reward design follows the structural failures identified in Section 3. Because ABSA quadruple extraction is evaluated as an unordered set prediction problem, rewards should reflect tuple-set structure rather than generation order. We therefore compute all reward components after optimal bipartite

matching between predicted and gold quadruples, making reward computation invariant to generation order and better aligned with set-level evaluation. The reward focuses on three dominant issues: incorrect target–aspect binding, hallucinated targets, and ambiguity among fine-grained aspect categories.

Base matching reward. We first define a set-level alignment reward that measures overall structural compatibility between predicted and gold quadruples. Let $\mathcal{Q}^* = \{q_i^*\}_{i=1}^{K^*}$ denote the gold quadruples and $\hat{\mathcal{Q}} = \{\hat{q}_j\}_{j=1}^{\hat{K}}$ the predicted quadruples. For each gold–prediction pair, we define

$$S_{ij} = \tfrac{1}{3}\Big[\mathbf{1}[t_i^* = \hat{t}_j] + \mathbf{1}[a_i^* = \hat{a}_j] + \mathbf{1}[s_i^* = \hat{s}_j]\Big], \quad (5)$$

and obtain an optimal one-to-one alignment $\mathcal{M}$ using the Hungarian algorithm. The resulting reward is

$$R_{\text{base}}(\mathbf{x}, \hat{y}) = \frac{1}{K^*} \sum_{(i,j)\in\mathcal{M}} S_{ij}. \quad (6)$$

This term provides a global set-level signal for overall tuple quality.

Binding reward. Our core reward directly targets the bottleneck at the target–aspect interface:

$$R_{\text{bind}}(\mathbf{x}, \hat{y}) = \tfrac{1}{|\mathcal{M}|+\varepsilon} \textstyle\sum_{(i,j)\in\mathcal{M}} \mathbf{1}[t_i^* = \hat{t}_j]\,\mathbf{1}[a_i^* = \hat{a}_j]. \quad (7)$$

Unlike R_{base}, R_{bind} gives credit only when target and aspect are jointly correct, directly targeting the main structural bottleneck under student rollouts.

Hallucination penalty. To suppress fabricated non-PRODUCT targets, we define a grounded hallucination reward:

$$R_{\text{hall}}(\mathbf{x}, \hat{y}) = \mathbb{1}[|\hat{\mathcal{Q}}| > 0] \cdot \left(1 - \frac{\sum_{\hat{q}_j\in\hat{\mathcal{Q}}} \text{Hall}_t(\hat{q}_j, \mathbf{x})}{|\hat{\mathcal{Q}}|}\right), \quad (8)$$

where $\mathbb{1}[\cdot]$ is the indicator function, and $\text{Hall}_t(\hat{q}_j, \mathbf{x}) \in \{0, 1\}$ indicates that the predicted quadruple $\hat{q}_j$ contains a non-PRODUCT target whose entity span cannot be grounded (i.e., is absent) in the review text $\mathbf{x}$.

Crucially, the indicator $\mathbb{1}[|\hat{\mathcal{Q}}| > 0]$ prevents the student model from exploiting a trivial reward-hacking shortcut—specifically, generating empty predictions to mathematically bypass the hallucination penalty. This formulation ensures that R_{hall} acts as a rigorous grounding constraint, assigning a reward of 1.0 only when the student generates a non-empty set of predictions that are entirely grounded in the source text.

Category reward. Hard matching alone provides limited guidance when several aspect labels are semantically adjacent. To preserve fine-grained teacher supervision within the aspect taxonomy, we add a teacher-support reward over the student-predicted aspect category:

$$R_{\text{cat}}(\mathbf{x}, \hat{y}) = \tfrac{1}{|\mathcal{M}|+\varepsilon} \textstyle\sum_{(i,j)\in\mathcal{M}} P_T(\hat{a}_j \mid \mathbf{x}, \text{ctx}_j), \quad (9)$$

where $\hat{a}_j$ is the student-predicted aspect in the aligned tuple, ctx_j is the student decoding context at which the aspect for the j-th predicted tuple is produced, and $P_T(\cdot)$ is the teacher distribution over the aspect taxonomy. In practice, $P_T(\cdot \mid \mathbf{x}, \text{ctx}_j)$ is obtained by a lightweight teacher scoring step under the student rollout context at the aspect prediction step, after restricting and normalizing the logits over the domain taxonomy; details are given in Appendix E. This term complements the discrete binding reward with a softer signal over nearby aspect categories.

Additional stabilizers. We further include a lightweight format-validity reward and use input-adaptive weighting, with stronger emphasis on binding consistency and hallucination suppression for reviews identified as likely product-part cases based on teacher pseudo-labels. The full cascade-aware reward is therefore

$$R(\mathbf{x}, \hat{y}) = \textstyle\sum_{k\in\{\text{base, bind, hall, cat, fmt}\}} \lambda_k(\mathbf{x}) R_k(\mathbf{x}, \hat{y}). \quad (10)$$

where the input-dependent coefficients $\lambda_k(\mathbf{x})$ control the relative contribution of each component.

Among these rewards, R_{bind} and R_{hall} are the most important. The former directly targets the target–aspect interface where structured aspect cascade is centered, while the latter suppresses hallucinated targets that invalidate the entire tuple. On-policy exposure and cascade-aware rewards are complementary: the former lets the student visit its own erroneous structural states, while the latter provides explicit credit assignment over which structural property of the rollout should be corrected. Detailed reward weights and training settings are given in Appendix E.

Because the reward is rule-based, a natural concern is shortcut behavior; we discuss this further in Appendix E.4.

5 Experiments

We design our experiments to answer three questions: (Q1) Does STAR-OPD improve overall dis-

tilled ABSA extraction quality over off-policy and generic on-policy baselines? (Q2) Does it specifically reduce the structural failure modes identified in Section 3, namely the target–aspect bottleneck and target hallucination? (Q3) Are the gains largest on reviews where structural ambiguity is strongest?

5.1 Setup

Datasets. We evaluate on E-ABSA20K (Sun et al., 2026), a 20K-review benchmark spanning four e-commerce domains, and SemEval-2014 (Pontiki et al., 2014) (Restaurant and Laptop) for cross-domain validation. We adopt E-ABSA20K as the primary benchmark for this study, as its longer reviews and higher tuple density make target–aspect binding errors and target hallucination substantially more frequent than in earlier ABSA settings.

Baselines. Unless otherwise specified, all student methods use Qwen3-4B. We compare against five baselines. **Direct-SFT** fine-tunes the student on filtered teacher-produced pseudo-labels as ordinary supervision targets. **SeqKD** (Kim and Rush, 2016) is the classical off-policy sequence-distillation baseline, training the student to imitate complete teacher-generated output sequences as hard targets. Direct-SFT treats teacher outputs as ordinary supervised targets, whereas SeqKD is reported as the canonical sequence-level off-policy distillation baseline.

GKD (Agarwal et al., 2024) is a general distillation baseline that combines off-policy and on-policy learning, training the student on both fixed target sequences and self-generated outputs with teacher feedback.

MiniLLM (Gu et al., 2024) is a strong general on-policy reverse-KL distillation method without task-specific structural rewards. We also report the **Teacher** as reference.

G-OPD (Yang et al., 2026) is a generalized on-policy distillation framework that extends standard OPD with a reward scaling factor and a flexible reference model, enabling stronger student updates and, in some settings, performance beyond the teacher.

GKD, MiniLLM, and G-OPD all represent generic on-policy distillation baselines without ABSA-specific structural reward design.

Evaluation metrics. Our primary metric is **Quad-F1**, with Hungarian matching over target, aspect, and sentiment, followed by opinion evaluation using a fuzzy character-level criterion (CharF1 with threshold τ=0.5) to avoid over-penalizing minor boundary deviations.

We additionally report **T-F1**, **TA-F1**, **TAS-F1**, and **T-Hall** for structural diagnosis (Figure 3). Formal definitions of all metrics are given in Appendix B.

Implementation details. Detailed optimization settings, reward-weighting details, hardware configuration, and runtime statistics are reported in Appendix E.

5.2 Main Results

Table 1 reports the overall results on E-ABSA20K and SemEval-2014.

STAR-OPD consistently improves over the off-policy baselines and also outperforms the generic on-policy baselines, including MiniLLM and G-OPD, on the more structurally demanding E-ABSA20K benchmark.

On SemEval-2014, the gains over the generic on-policy baselines are smaller, and statistical significance is less consistent. We attribute this to the simpler review structure and lower tuple density of SemEval, which leave less room for structured cascade and therefore reduce the advantage of task-specific structural rewards.

For Qwen3-4B, STAR-OPD substantially narrows the gap to the 32B teacher while retaining the efficiency advantage of the smaller model, and the same trend transfers to the 1.7B student.

5.3 Reducing Structured Cascade

To test whether STAR-OPD reduces the central failure mode identified in Section 3, Figure 3 reports field-level performance together with target hallucination rate.

Across all students, target-only performance is already strong, but the much larger drop from T-F1 to TA-F1 confirms that the main bottleneck lies at the target–aspect interface, with further declines to TAS-F1 and Quad-F1 showing downstream effects on sentiment and opinion prediction.

Compared with SeqKD, STAR-OPD improves every stage of this progression, with the largest gain at the target–aspect interface, substantially reducing the T–TA gap.

Figure 3(b) further shows that STAR-OPD lowers target hallucination from 9.75% to 7.22%, suggesting improved structural validity rather than

Table 1: Main results on E-ABSA20K and SemEval-2014 (Micro F1). Best student in **bold**; runner-up underlined. †: significantly better than MiniLLM ($p<0.05$, paired bootstrap).

	E-ABSA20K				SemEval-2014	
Method	WomenBags	Dresses	Makeup	Furniture	Restaurant	Laptop
Teacher	0.756	0.762	0.757	0.812	0.746	0.516
Student: Qwen3-4B						
Direct-SFT	0.685	0.691	0.692	0.720	0.685	0.470
SeqKD	0.687	0.695	0.699	0.724	0.715	0.468
GKD	0.689	0.699	0.704	0.727	0.719	0.471
MiniLLM	0.705	0.711	0.705	0.738	0.728	0.481
G-OPD	0.711	0.713	0.711	0.741	0.728	0.480
STAR-OPD	**0.739**†	**0.750**†	**0.726**†	**0.771**†	**0.732**	**0.487**
Student: Qwen3-1.7B						
Direct-SFT	0.657	0.661	0.637	0.685	0.553	0.402
SeqKD	0.666	0.664	0.647	0.702	0.570	0.405
GKD	0.671	0.673	0.652	0.705	0.580	0.412
MiniLLM	0.694	0.695	0.660	0.716	0.608	0.421
G-OPD	0.699	0.702	0.662	0.717	0.607	0.421
STAR-OPD	**0.702**†	**0.707**†	**0.673**†	**0.721**†	**0.614**†	**0.425**

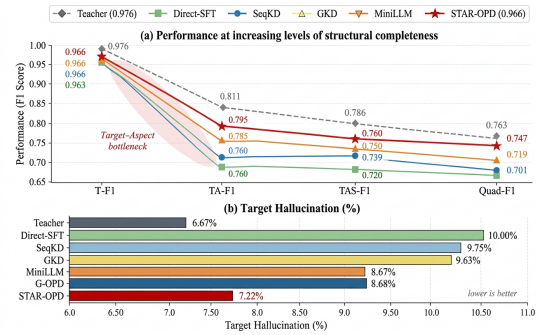

Figure 3: Reducing structured aspect cascade on E-ABSA20K Test. (a) Performance at increasing levels of structural completeness: T-F1, TA-F1, TAS-F1, and Quad-F1. (b) Target hallucination rate (T-Hall; lower is better). STAR-OPD improves all stages of structural correctness, with the largest gain at the target–aspect interface, while reducing hallucinated targets.

merely surface-level imitation.

5.4 On-Policy and Reward Ablations

Table 2 evaluates the contributions of on-policy exposure and cascade-aware rewards.

Removing on-policy training causes the largest drop, confirming that student rollouts are essential for covering the structurally invalid states absent from off-policy teacher supervision. Removing reward shaping and retaining only plain on-policy distillation also leads to a clear degradation, showing that on-policy exposure alone is not sufficient without explicit structural credit assignment.

We further compare against a collapsed scalar reward variant that merges the main structural signals into a single outcome-level score without component-specific weighting (Appendix E.2). Its weaker performance indicates that decomposing binding, grounding, and category support provides more effective structural credit assignment than a single undifferentiated reward.

Among reward terms, removing the binding reward R_{bind} causes the largest performance drop, showing that correcting target–aspect binding errors is central to the method. Removing R_{hall}, R_{cat}, R_{fmt}, or R_{adap} also degrades performance, indicating that suppressing non-grounded targets, preserving fine-grained aspect supervision, stabilizing for-

Table 2: Ablation of on-policy design choices and cascade-aware rewards on E-ABSA20K Test. F1 denotes micro Quad-F1 over the four E-ABSA20K test domains. ΔF1 is relative to full **STAR-OPD**.

Variant	**F1**	**ΔF1**
STAR-OPD (full)	0.747	—
On-policy mechanism		
w/o On-Policy	0.702	−0.045
w/o Rewards	0.716	−0.031
Reward decomposition		
Collapsed scalar reward	0.734	−0.013
Cascade-aware rewards		
w/o R_{bind}	0.729	−0.018
w/o R_{hall}	0.737	−0.010
w/o R_{cat}	0.738	−0.009
w/o R_{fmt}	0.735	−0.012
w/o R_{adap}	0.737	−0.010
Matching strategy		
Greedy (vs. Hungarian)	0.735	−0.012

Table 3: Hard-sample F1 on E-ABSA20K-Hard. PP: product-part mentions; Sarc.: colloquial sarcasm; Cont.: contrastive sentences; Impl.: implicit opinion expressions.

Method	PP	Sarc.	Cont.	Impl.
Teacher	0.687	0.717	0.722	0.723
Direct-SFT	0.632	0.664	0.657	0.679
SeqKD	0.633	0.678	0.671	0.680
GKD	0.635	0.678	0.672	0.683
MiniLLM	0.643	0.689	0.687	0.699
G-OPD	0.644	0.692	0.686	0.709
STAR-OPD	**0.665**	**0.705**	**0.699**	**0.710**

mat validity, and emphasizing structurally difficult cases are all beneficial.

Finally, replacing Hungarian matching with greedy matching also degrades performance, showing that order-invariant set-level alignment is important for assigning structurally meaningful rewards.

These results show that on-policy exposure and structure-aware rewards are complementary: the former exposes student error states, and the latter provides explicit structural credit assignment. This is consistent with the gap between MiniLLM and STAR-OPD in Table 1.

5.5 Performance on Hard Cases

We further evaluate on E-ABSA20K-Hard, where structural ambiguity most strongly amplifies cascade failures. Table 3 shows that STAR-OPD achieves the best student performance on all hard subsets, with the largest gains on product-part cases and consistent improvements on contrastive and implicit-opinion reviews. Generic on-policy baselines such as GKD and G-OPD also improve over off-policy distillation on these subsets, but remain less effective than STAR-OPD on the most structure-sensitive cases.

Because aggregate scores do not reveal how different distillation strategies fail, Figure 4 presents a representative product-part example: SeqKD preserves the grounded part target but drifts in target–aspect binding, MiniLLM generates a plausible yet non-grounded target, and STAR-OPD restores the intended grounded target together with its fine-grained aspect.

The qualitative example in Figure 4 highlights how the baselines fail differently under product-part ambiguity. SeqKD preserves the observed part mention but drifts to a neighboring aspect, whereas MiniLLM produces a plausible yet non-grounded part target. In contrast, STAR-OPD recovers the intended grounded target together with its part-specific aspect, illustrating why its gains are largest on hard subsets where target granularity and aspect assignment are tightly coupled.

Qualitative case: product-part (womenbags)	
Review	The bag looks elegant, but the zipper got stuck on the second day.
Gold	(`PRODUCT:bag`, `Design Aesthetic`, *looks elegant*, `pos`); (`PRODUCT_PART:zipper`, `Hardware Quality`, *got stuck*, `neg`)
SeqKD	(`PRODUCT_PART:zipper`, `Ease of Use`, *got stuck*, `neg`) [binding drift]
MiniLLM	(`PRODUCT_PART:handle`, `Hardware Quality`, *got stuck*, `neg`) [hallucinated target]
STAR-OPD	(`PRODUCT_PART:zipper`, `Hardware Quality`, *got stuck*, `neg`) [correct grounding + binding]

Figure 4: Representative qualitative case from E-ABSA20K-Hard. SeqKD preserves the grounded target but drifts in aspect binding, whereas MiniLLM generates a plausible yet non-grounded part target. STAR-OPD restores both correct grounding and part-specific aspect binding. Additional cases are given in Appendix G.

6 Conclusion

We studied why distillation is brittle for ABSA quadruple extraction and found that the main challenge is not teacher imitation alone, but recovery from student-induced structural states. In this task, the most damaging failures are centered on the target–aspect interface, where broken bindings and hallucinated targets degrade downstream tuple correctness. STAR-OPD addresses this mismatch by building on generic on-policy distillation and adding cascade-aware rewards for binding consistency and target grounding. Across E-ABSA20K and SemEval-2014, this design consistently improves distilled students, reduces hallucination, and yields the largest gains on structurally hard reviews.

More broadly, our results suggest that distillation for structured extraction should be designed around inference-time student states rather than teacher-only trajectories. Our rewards operate at the structured outcome level rather than as token-level process rewards, which is natural for unordered quadruple extraction. Exploring finer-grained process-style rewards is an interesting direction for future work.

7 Limitations

Our study has several limitations. First, although we evaluate on both E-ABSA20K and SemEval-

2014, the analysis is centered on ABSA quadruple extraction, especially in e-commerce reviews, so the generality of STAR-OPD to other structured extraction settings remains to be validated. Second, our reward design operates at the structured outcome level after generation rather than as a token-level or process-level reward, which may limit earlier intervention during decoding. Third, the method depends on the quality of the teacher and pseudo-label filtering pipeline, and our hard-subset analyses are intended as diagnostic evidence of failure modes rather than the sole basis for broad statistical claims.

Potential Risks. We do not identify severe direct risks beyond those typical of sentiment analysis systems, but deployment may amplify annotation biases or residual errors in target–aspect binding and target grounding. In practical settings, such errors could lead to incorrect summaries of user opinions or mischaracterization of product attributes, so the method is better viewed as a research and assistive analysis tool rather than a fully autonomous high-stakes decision system.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. 2024. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

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 (ACL-IJCNLP 2021)*, pages 340–350, Online. Association for Computational Linguistics.

Jenny Rose Finkel, Christopher D. Manning, and Andrew Y. Ng. 2006. Solving the problem of cascading errors: Approximate Bayesian inference for linguistic annotation pipelines. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, pages 618–626, Sydney, Australia. Association for Computational Linguistics.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. Minillm: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2023. Large language models are reasoning teachers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL 2023)*, pages 14852–14882, Toronto, Canada. Association for Computational Linguistics.

Mengting Hu, Yike Wu, Hang Gao, Yinhao Bai, and Shiwan Zhao. 2022. 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.

Woogyeol Jin, Taywon Min, Yongjin Yang, Swanand Ravindra Kadhe, Yi Zhou, Dennis Wei, Nathalie Baracaldo, and Kimin Lee. 2026. Entropy-aware on-policy distillation of language models. In *The 1st Workshop on Scaling Post-training for LLMs*.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. 2024. Distillm: towards streamlined distillation for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, page 611–626, New York, NY, USA. Association for Computing Machinery.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2023. Teaching small language models to reason. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1773–1781, Toronto, Canada. Association for Computational Linguistics.

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. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI 2020)*, volume 34, pages 8864–8871, New York, USA. AAAI Press.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics.

Kevin Scaria, Himanshu Gupta, Siddharth Goyal, Saurabh Sawant, Swaroop Mishra, and Chitta Baral. 2024. InstructABSA: Instruction learning for aspect based sentiment analysis. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 720–736, Mexico City, Mexico. Association for Computational Linguistics.

Tong Sun, Mingyang Ma, and Cheng Yu. 2026. E-ABSA20k: A dataset and propose-and-verify for aspect-based sentiment analysis in long e-commerce reviews. In *The 64th Annual Meeting of the Association for Computational Linguistics*.

Yanshan Wang, Sunyang Fu, Chen Sun, Ying Li, Changlin Gong, Xun Chen, Liwei Wang, Fengjun Li, and Sunghwan Sohn. 2023. Is ChatGPT a good sentiment reasoner? a preliminary study. *Preprint*, arXiv:2304.09582.

Shicheng Xu, Liang Pang, Yunchang Zhu, Jia Gu, Zihao Wei, Jingcheng Deng, Feiyang Pan, Huawei Shen, and Xueqi Cheng. 2026. Rlkd: Distilling llms' reasoning via reinforcement learning. In *AAAI*, pages 34151–34159.

Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. 2026. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation. *CoRR*, abs/2602.12125.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2021. 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 (ACL-IJCNLP 2021)*, pages 504–510, Online. Association for Computational Linguistics.

A Task and Dataset Details

A.1 Full Task Definition

Given a review text $\mathbf{x}$, the goal of ABSA quadruple extraction is to predict a set of sentiment quadruples

$$\mathcal{Q}(\mathbf{x}) = \{q_i\}_{i=1}^{K}, \qquad q_i = (t_i, a_i, o_i, s_i),$$

where each quadruple consists of a target t_i, an aspect a_i, an opinion span o_i, and a sentiment label s_i.

The target field follows a structured hierarchy:

$$t_i \in \{\texttt{PRODUCT},\ \texttt{PRODUCT:}xx,\ \texttt{PRODUCT\_PART:}xx\}. \tag{11}$$

where the entity xx must appear verbatim in the review text. The aspect field $a_i \in \mathcal{C}$ is selected from a closed domain taxonomy with up to $|\mathcal{C}|$=42 fine-grained categories depending on the domain. The opinion field o_i is a minimal contiguous span in the input, and the sentiment field $s_i \in \{\text{pos}, \text{neu}, \text{neg}\}$.

Compared with simpler ABSA formulations, quadruple extraction is structurally more challenging because each review may contain multiple quadruples and each quadruple combines several heterogeneous prediction subproblems, including entity identification for targets, taxonomy classification for aspects, span extraction for opinions, and polarity classification for sentiment. In particular, the target–aspect interface forms the key structural bottleneck linking target grounding to downstream sentiment and opinion interpretation, making quadruple extraction especially vulnerable to structural error propagation.

A.2 Dataset Statistics

We conduct experiments on E-ABSA20K (Sun et al., 2026), a 20K-review benchmark spanning four e-commerce domains, and on SemEval-2014 (Pontiki et al., 2014) for cross-domain validation. The E-ABSA20K benchmark is substantially more challenging than earlier ABSA datasets because reviews are longer and contain multiple sentiment quadruples. On average, each review in E-ABSA20K contains 6.0 quadruples, creating substantially more opportunities for target confusion, aspect ambiguity, and structural binding errors than shorter benchmark settings.

We additionally construct E-ABSA20K-Hard for fine-grained diagnosis. This subset contains reviews exhibiting at least one structurally difficult phenomenon, including product-part mentions, sarcasm cases, contrastive sentences, implicit opinion expressions. These phenomena are particularly useful for analyzing the failure modes of distilled models because they amplify aspect ambiguity, target–aspect binding difficulty, and target hallucination.

A.3 Construction of E-ABSA20K-Hard

We define E-ABSA20K-Hard as the subset of reviews that contain at least one challenging phenomenon known to increase structural ambiguity

in quadruple extraction. The subset includes the following categories:

- **Product-part mentions**: reviews containing fine-grained product-part targets, which require the model to preserve target identity at a more specific level than the generic product tag.
- **Colloquial sarcasm**: reviews whose sentiment is expressed indirectly or ironically, often weakening the reliability of surface lexical cues.
- **Contrastive sentences**: reviews containing discourse structures such as *but*, where sentiment orientation may reverse across clauses and must be resolved in an aspect-sensitive way.
- **Implicit opinion expressions**: reviews in which polarity must be inferred without explicit sentiment words, increasing reliance on aspect semantics and context.

These categories are used only for diagnostic evaluation and hard-case analysis. They are not required by the proposed method, but help reveal where structured aspect cascade is most severe. Because these subsets are smaller and partially overlapping, we treat the corresponding results as diagnostic rather than as the primary basis for significance claims.

B Evaluation Details

B.1 Field-Level Matching Metrics

To analyze structural errors at different levels of completeness, we report four cumulative matching metrics: target-only F1 (T-F1), target–aspect F1 (TA-F1), target–aspect–sentiment F1 (TAS-F1), and quadruple F1 (Quad-F1).

Let $\mathcal{Q}^* = \{q_i^*\}_{i=1}^{K^*}$ denote the gold quadruples and $\hat{\mathcal{Q}} = \{\hat{q}_j\}_{j=1}^{\hat{K}}$ denote the predicted quadruples, where each quadruple has the form $q = (t, a, o, s)$.

T-F1. A prediction is counted as correct under T-F1 if its target field matches a gold target after one-to-one alignment between predicted and gold quadruples. This metric measures target identification in isolation.

TA-F1. A prediction is counted as correct under TA-F1 if both the target and aspect fields match after alignment. This metric directly reflects the quality of target–aspect binding and is therefore central to diagnosing the bottleneck at the target–aspect interface.

TAS-F1. A prediction is counted as correct under TAS-F1 if the target, aspect, and sentiment fields all match after alignment. This metric captures structural correctness after downstream sentiment resolution.

Quad-F1. A prediction is counted as correct under Quad-F1 if the target, aspect, sentiment, and opinion fields all match after alignment, where opinion matching is evaluated with a fuzzy character-level criterion described below. This is the primary evaluation metric used in the main experiments.

All F1 scores are reported as micro-averaged set-level F1 across the evaluation corpus.

B.2 Hungarian Matching for Set Prediction

Because each review may contain multiple predicted and gold quadruples, evaluation is performed with one-to-one bipartite matching rather than by comparing outputs in generation order. We compute an optimal alignment between gold and predicted quadruples using the Hungarian algorithm.

For a gold quadruple q_i^* and a predicted quadruple $\hat{q}_j$, we define the structural similarity

$$S_{ij} = \frac{1}{3}\Big(\mathbf{1}[t_i^* = \hat{t}_j] + \mathbf{1}[a_i^* = \hat{a}_j] + \mathbf{1}[s_i^* = \hat{s}_j]\Big), \tag{12}$$

and obtain an optimal one-to-one matching $\mathcal{M} \subseteq \mathcal{Q}^* \times \hat{\mathcal{Q}}$ that maximizes total similarity.

After matching, precision, recall, and F1 are computed under the relevant correctness criterion: target-only matching for T-F1, joint target–aspect matching for TA-F1, joint target–aspect–sentiment matching for TAS-F1, and full quadruple matching for Quad-F1.

We intentionally exclude opinion spans from the matching score used by the Hungarian algorithm. Because opinion extraction is evaluated with a fuzzy character-level criterion, incorporating it directly into the alignment objective may introduce unstable matches under minor boundary variation. Using target, aspect, and sentiment for alignment yields a more stable structural correspondence between predicted and gold quadruples, while opinion correctness is still enforced afterward in Quad-F1 through the CharF1 threshold. Thus, opinion is excluded from alignment but not from final quadruple evaluation.

B.3 Opinion Matching

Opinion spans are evaluated with character-level F1 (CharF1), which is more robust than exact span

matching to minor boundary deviations. For a gold opinion span o_i^* and a predicted opinion span $\hat{o}_j$, we compute

$$\mathrm{CharF1}(o_i^*, \hat{o}_j),$$

the character-level F1 score between the two spans after normalization.

An aligned prediction is counted as opinion-correct if its opinion CharF1 exceeds a fixed threshold τ. Accordingly, a matched pair $(q_i^*, \hat{q}_j)$ is counted as correct under Quad-F1 if

$$t_i^* = \hat{t}_j, \quad a_i^* = \hat{a}_j, \quad s_i^* = \hat{s}_j,$$

and

$$\mathrm{CharF1}(o_i^*, \hat{o}_j) \geq \tau.$$

Unless otherwise specified, we use $\tau = 0.5$ in all experiments.

B.4 Target Hallucination Rate (T-Hall)

Target hallucination rate (T-Hall) measures the proportion of predicted quadruples whose non-PRODUCT target entity does not appear in the input review text. This metric is designed to capture a structurally severe failure mode in distilled quadruple extraction, where the model fabricates a target that cannot be grounded in the source review.

Formally, let $\hat{\mathcal{Q}} = \{\hat{q}_j\}_{j=1}^{\hat{K}}$ be the predicted quadruples for input $\mathbf{x}$, and let $\hat{t}_j$ denote the target field of $\hat{q}_j$. We define an indicator

$$\mathrm{Hall}_t(\hat{q}_j, \mathbf{x}) = \begin{cases} 1, & \hat{t}_j \neq \texttt{PRODUCT} \\ & \wedge\ \mathrm{ent}(\hat{t}_j) \notin \mathrm{text}(\mathbf{x}), \\ 0, & \text{otherwise.} \end{cases}$$

where $\mathrm{ent}(\hat{t}_j)$ extracts the entity span from the predicted target and $\mathrm{text}(\mathbf{x})$ denotes the normalized review text.

The target hallucination rate is then

$$\text{T-Hall}(\mathbf{x}) = \frac{|\{\hat{q}_j : \mathrm{Hall}_t(\hat{q}_j, \mathbf{x}) = 1\}|}{|\hat{\mathcal{Q}}| + \varepsilon}, \quad (13)$$

where ε is a small constant to avoid division by zero. Corpus-level T-Hall is obtained by averaging over the evaluation set.

For targets of the form PRODUCT:*xx* or PRODUCT_PART:*xx*, the entity span *xx* is extracted and checked against the review text using exact substring matching after lowercasing and whitespace normalization. A lower T-Hall indicates that predicted targets are better grounded in the review text.

C Additional Analysis of Structural Failure Modes

We observe that structured aspect cascade in distilled quadruple extraction is centered on the target–aspect interface and manifests primarily in two ways: broken target–aspect bindings and hallucinated targets.

Broken target–aspect bindings and downstream corruption. Because the fields in a quadruple are structurally interdependent, an incorrect aspect assignment changes how the target is interpreted and can degrade downstream sentiment and opinion prediction. This explains why the largest performance drop occurs at the transition from target-only to target–aspect matching, and why additional degradation remains at the TAS-F1 and Quad-F1 levels even after partial structural correctness is achieved.

Hallucinated targets. In addition to misclassifying existing targets, small distilled models may fabricate non-PRODUCT entities that do not appear in the review text, especially at the product-part level. Such hallucinations are structurally severe: once the target itself is invalid, the entire quadruple becomes spurious and no downstream field can be correct. In the SeqKD baseline, the target hallucination rate reaches 9.75%, compared with 6.67% for the teacher (see Appendix B for the formal definition of T-Hall).

Taken together, these observations show that distilled quadruple extraction fails in a distinctly structured way. The main issue is not generic sequence noise, but student errors that create invalid structural states at the target–aspect interface and thereby corrupt downstream tuple interpretation.

D Teacher Preparation and Quality Control

D.1 CoT Teacher Fine-Tuning

We use a Qwen3-32B model as the teacher. To improve the reliability of pseudo-label generation, we fine-tune the teacher with chain-of-thought (CoT) supervision so that it explicitly reasons over target identification, aspect assignment, opinion extraction, and sentiment resolution before producing the final quadruples. Only the final quadruple outputs are used as task targets for distillation; teacher reasoning traces are not distilled. Teacher distributions may still be queried during training for reward computation (e.g., aspect-category support).

The teacher is trained only on manually annotated training data. Development annotations are used for model selection, while test annotations are never used for teacher adaptation, pseudo-label generation, reward tuning, or student training.

The resulting model is referred to as **Teacher** in the main paper.

D.2 Agreement with Expert Annotations

To ensure that the teacher provides reliable supervision, we compare teacher predictions against human expert annotations on a held-out subset of E-ABSA20K-Hard. The subset was independently annotated by two domain annotators who were blind to model outputs. We report teacher–human agreement against the adjudicated labels. Human–human agreement on the same subset was $\kappa = 0.89$, and disagreements were resolved by discussion with a third annotator.

Table 4 reports agreement at the field level as well as overall Cohen's κ. The **Teacher** achieves substantially higher agreement than a smaller direct model and reaches $\kappa = 0.92$, which we treat as sufficient quality for pseudo-label generation.

These results support the use of the CoT teacher as the source of pseudo-label supervision in STAR-OPD.

Table 4: Teacher quality against human expert annotations. κ denotes Cohen's Kappa.

Field	32B CoT	4B Direct
Target agreement	97.5%	94.2%
Aspect agreement	93.8%	81.6%
Opinion agreement	94.1%	88.3%
Sentiment agreement	95.3%	86.7%
Overall κ	0.92	0.81

D.3 Pseudo-Label Generation and Filtering

Using the validated teacher, we generate pseudo-labels for the student training split. Because teacher outputs may still contain occasional formatting errors or structurally invalid extractions, we apply lightweight filtering before distillation.

We retain only instances satisfying the following constraints:

- **Well-formed quadruples:** each prediction must follow the required output schema and contain all four fields.
- **Valid aspect labels:** the predicted aspect must belong to the domain-specific taxonomy $\mathcal{C}$.
- **Valid sentiment labels:** the predicted sentiment must be one of $\{\texttt{pos}, \texttt{neu}, \texttt{neg}\}$.
- **Grounded non-`PRODUCT` targets:** when the predicted target is not `PRODUCT`, the corresponding entity span must appear in the review text.
- **Duplicate removal:** repeated identical quadruples for the same review are collapsed into a single instance.

This filtering step is intended only to remove structurally invalid supervision. The correction of student-induced structural errors is still handled during on-policy training with cascade-aware rewards. After filtering, we retain 70% of teacher-generated training instances overall. The domain-wise retention rates are broadly similar, remaining within 66%–73% across the four domains rather than concentrating on a single domain or review type. Pseudo-labels are generated only for the student training split. No evaluation annotations from the validation or test sets are used in pseudo-label generation or filtering. The E-ABSA20K-Hard subsets are used only for diagnostic evaluation and never for teacher or student training.

E Training Details

E.1 Reward Weights and Input-Adaptive Weighting

The total reward is defined as

$$R(\mathbf{x}, \hat{y}) = \sum_k \lambda_k(\mathbf{x}) R_k(\mathbf{x}, \hat{y}),$$

where the component rewards include base matching, target–aspect binding, hallucination suppression, category supervision, and format validity.

In practice, we assign larger relative weight to the binding reward and the hallucination penalty, since these two terms directly target the dominant structural failures identified in Section 3. The category reward is given moderate weight to preserve fine-grained aspect supervision, while the format reward receives only a small weight as a stabilizing term. All reward weights are selected on a held-out validation set.

We selected these weights on the validation set by first fixing a stable default configuration and then varying each coefficient within a narrow range while monitoring Quad-F1, TA-F1, and T-Hall. We found the method to be relatively stable under moderate (± 0.05) perturbations around the final values. We use the same default reward weights across

Table 5: Reward weights used in STAR-OPD. Default weights are used unless the input is identified as a likely PRODUCT_PART case based on teacher pseudo-labels.

Reward term	Default	PRODUCT_PART case
λ_{base}	0.20	0.15
λ_{bind}	0.30	0.35
λ_{hall}	0.25	0.30
λ_{cat}	0.20	0.15
λ_{fmt}	0.05	0.05

all domains rather than tuning them separately per domain.

To avoid over-tuning a five-way reward decomposition, we used a constrained local grid around a stable default configuration rather than an exhaustive combinatorial search. Specifically, we first fixed a default profile consistent with the structural diagnosis in Section 3, then varied one coefficient at a time while monitoring Quad-F1, TA-F1, and T-Hall on the validation set. This narrow search strategy was chosen to reduce the risk of overfitting to a particular validation distribution. In practice, we did not observe severe instability from interactions among reward terms; the main trade-off was between stronger hard-constraint rewards ($R_{\text{bind}}, R_{\text{hall}}$) and softer category supervision (R_{cat}). We intentionally kept the final weights simple and low-dimensional so that the method remains interpretable and easy to reproduce.

We further use lightweight input-adaptive weighting. For reviews identified as likely product-part cases based on teacher pseudo-labels, we increase the relative emphasis on binding consistency and hallucination suppression, since these cases are especially prone to target–aspect binding failures and fabricated part-level targets.

E.2 Collapsed Scalar Reward Variant

To test whether explicit reward decomposition is necessary, we additionally consider a collapsed scalar reward variant that merges the main structural reward components into a single outcome-level score without component-specific weighting. We use a simple uniform average so that the comparison isolates the value of explicit reward decomposition and component-specific weighting.

Specifically, we replace the decomposed reward in Eq. (10) with

$$R_{\text{collapsed}}(\mathbf{x}, \hat{y}) = \frac{1}{4}\Big(R_{\text{base}}(\mathbf{x}, \hat{y}) + R_{\text{bind}}(\mathbf{x}, \hat{y}) + R_{\text{hall}}(\mathbf{x}, \hat{y}) + R_{\text{cat}}(\mathbf{x}, \hat{y})\Big). \tag{14}$$

In this variant, we remove component-specific weighting and input-adaptive weighting, while keeping the rest of the on-policy optimization setup unchanged, including student rollouts, reverse-KL distillation, reference regularization, reward normalization, and the reward-to-weight transformation $w(\cdot)$. The format-validity reward is omitted from this collapsed variant so that the comparison focuses on whether decomposing the main structural signals for binding, grounding, and category supervision provides better credit assignment than a single undifferentiated reward.

This variant preserves access to the same underlying structural information as STAR-OPD, but discards the explicit decomposition used to bias updates toward specific failure modes. As reported in Table 2, the weaker performance of this variant suggests that separating binding consistency, hallucination suppression, and category support yields more effective structural credit assignment than collapsing them into a single scalar outcome reward.

E.3 Reward-to-Weight Transformation

For each student rollout, STAR-OPD first computes the raw cascade-aware reward. For brevity, $R_i = R(\mathbf{x}_i, \hat{y}_i)$ denotes the raw reward of the i-th rollout in the current batch. Because the absolute reward scale may vary across batches, we normalize rewards within each batch before converting them into rollout weights.

Specifically, let μ_R and σ_R denote the mean and standard deviation of raw rewards in the current batch. We compute the normalized reward

$$\tilde{R}_i = \frac{R_i - \mu_R}{\sigma_R + \epsilon}, \tag{15}$$

where ϵ is a small constant for numerical stability.

The rollout weight is then defined as

$$\alpha_i = w(R_i) = \text{clip}\Big(\exp(\tilde{R}_i/\tau_w), \alpha_{\min}, \alpha_{\max}\Big), \tag{16}$$

where τ_w is a temperature parameter controlling the sharpness of reward weighting, and $\alpha_{\min}, \alpha_{\max}$ bound the rollout weight to avoid overly small or overly dominant updates.

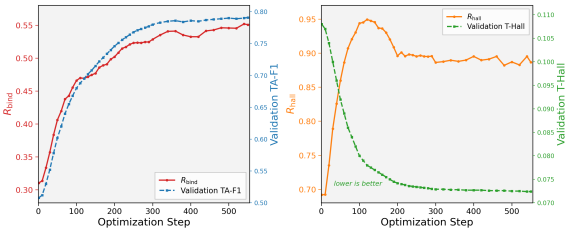

Figure 5: Training dynamics of the main reward components and corresponding validation structural metrics during STAR-OPD optimization. Left: R_{bind} and validation TA-F1. Right: R_{hall} and validation T-Hall. The reward curves and validation metrics move in directions consistent with improved structural behavior and do not show obvious pathological late-stage divergence.

Unless otherwise specified, we use $\tau_w = 2.0$, $\alpha_{\min} = 0.2$, $\alpha_{\max} = 3.0$, and $\epsilon = 10^{-6}$ in all experiments. This transformation preserves the ordering of rollout quality, gives higher weight to better-structured student outputs, and improves training stability compared with using raw rewards directly.

E.4 Training Dynamics of Reward Components

To examine whether the rule-based rewards remain aligned with the intended structural objectives during training, we track the dynamics of the two main reward components together with their corresponding validation metrics. Figure 5 plots R_{bind} alongside validation TA-F1, and R_{hall} alongside validation T-Hall.

The increase in R_{bind} is accompanied by improved validation TA-F1, which is consistent with better target–aspect structural consistency. Similarly, R_{hall} rises sharply early in training and later fluctuates within a stable high-reward range, while validation T-Hall continues to decrease, indicating improved suppression of non-grounded target predictions.

We do not observe obvious late-stage reward inflation or instability together with metric degradation. Although this analysis does not formally exclude shortcut optimization, it provides evidence that the reward terms remain broadly aligned with the structural behaviors they are intended to encourage. This interpretation is also consistent with the strongest gains of STAR-OPD on product-part and other structurally difficult subsets in Table 3.

E.5 Difficulty-Aware Sampling and Replay

To improve hard-case coverage under on-policy training, we use a lightweight difficulty-aware sampling strategy. Each training instance is assigned a difficulty score

$$d(x) = \mathbf{1}[\text{PP}] + \mathbf{1}[\text{Sarc}] + \mathbf{1}[\text{Cont}] + \mathbf{1}[\text{Impl}], \quad (17)$$

where each indicator is derived from teacher pseudo-labels and lightweight heuristic signals corresponding to the four hard phenomena; no diagnostic evaluation labels are used for training. Sampling probability is then defined as

$$p(x) \propto 1 + \gamma \cdot d(x), \quad (18)$$

with $\gamma = 0.5$ by default. Harder instances are sampled more frequently than easy ones.

We maintain a replay buffer of low-reward student rollouts to revisit unresolved structural failures. A rollout $(x_i, \hat{y}_i)$ is added to the buffer if its normalized reward $\tilde{R}_i < -0.5$. The buffer stores up to 4,096 rollouts in a FIFO manner. During training, 20% of each batch is drawn from the replay buffer when available, and the remaining 80% is sampled from the training set.

These replayed rollouts typically correspond to unresolved structural failures, such as incorrect target–aspect bindings or hallucinated targets, and help focus updates on the student states where structure-aware feedback is most needed.

E.6 Curriculum

To stabilize optimization, we use a simple curriculum on hard-sample exposure. During the first 30% of training steps, the probability of sampling from the difficulty-aware distribution is increased linearly from 0.3 to 0.7; afterward, it is fixed at 0.7 for the remainder of training.

This curriculum reduces early optimization noise while ensuring that the model eventually receives sufficient coverage of structurally difficult cases.

E.7 Optimization Hyperparameters

Unless otherwise specified, all student methods use Qwen3-4B as the default student and Qwen3-1.7B for scale analysis. On-policy distillation is run for 2K update iterations.

Unless otherwise specified, each input contributes one student rollout per update step. We use a per-device training batch size of 4 with gradient accumulation over 2 steps on 4 GPUs, resulting in an effective batch size of 32 training instances per parameter update. The maximum generation length for student rollouts is set to 1024 new tokens.

We use AdamW with a peak learning rate of 1×10^{-5}, weight decay of 0.01, and a cosine decay schedule with 5% warmup. The reference-policy regularization coefficient in Eq. (1) is set to $\beta = 0.1$.

For rollout generation, we use temperature 0.8 and top-p sampling with $p = 0.95$. Unless otherwise noted, all experiments use seed 42. For R_{cat}, teacher scoring is performed only at the aspect prediction step for the sampled student rollout, rather than over the full decoded sequence, to limit additional training overhead.

E.8 Hardware and Runtime

Teacher fine-tuning is conducted on 4×A100 80G GPUs for approximately 14 hours. On-policy student distillation is conducted on 4×A100 80G GPUs for about 24 hours, with teacher inference served by vLLM (Kwon et al., 2023). These numbers are reported to provide an approximate training-cost reference for STAR-OPD; inference-time efficiency comparisons are given separately in Appendix E.9.

For evaluation, all reported numbers are averaged over three runs when applicable, and statistical significance is tested with paired bootstrap resampling at $p < 0.05$.

E.9 Inference Efficiency

To quantify the deployment advantage of distilled students, we measure inference efficiency on E-ABSA20K Test under the same decoding configuration for teacher and student models. Table 6 reports the average end-to-end latency per review and review throughput on a single A100 80G GPU.

The Qwen3-4B student distilled with STAR-OPD processes reviews 2.18× faster than the Qwen3-32B teacher, while substantially narrowing the performance gap. The smaller 1.7B student is even faster, confirming the practical quality–efficiency advantage of distilled models for large-scale deployment.

Table 6: Inference efficiency on E-ABSA20K Test under the same decoding configuration. All models are evaluated on a single A100 80G GPU.

Model	Latency	Reviews/s	Speedup
Teacher†	2.119	0.47	1.0×
STAR-OPD (4B)‡	0.971	1.03	2.18×
STAR-OPD (1.7B)‡	0.290	3.45	7.31×

† Qwen3-32B teacher model. ‡ Qwen3 student models distilled via STAR-OPD.

Algorithm 1 STAR-OPD: On-Policy Cascade-Aware Distillation

Require: Teacher policy π_T, initial student policy π_S, reference policy $\pi_{\text{ref}} \leftarrow \pi_S$, training set $\mathcal{D}$
Initialize replay buffer $\mathcal{B} \leftarrow \emptyset$
for each training iteration **do**
  Sample a batch $\mathcal{S}$ from $\mathcal{D}$ with difficulty-aware sampling
  Optionally mix in low-reward samples from $\mathcal{B}$
  for each input $\mathbf{x}_i \in \mathcal{S}$ **do**
    Sample a student rollout $\hat{y}_i \sim \pi_S(\cdot \mid \mathbf{x}_i)$
    Compute cascade-aware reward R_i using $R_{\text{base}}, R_{\text{bind}}, R_{\text{hall}}, R_{\text{cat}}, R_{\text{fmt}}$
    if R_i is low **then**
      Add $(\mathbf{x}_i, \hat{y}_i)$ to $\mathcal{B}$
    end if
  end for
  Normalize rewards within the batch
  Convert normalized rewards to rollout weights $\alpha_i = w(R_i)$
  Compute the reward-weighted on-policy objective

$$\mathcal{L} = \mathcal{L}_{\text{rKL}} + \beta \mathcal{L}_{\text{ref}}$$

  Update student parameters
end for
return π_S

F Training Algorithm

G Additional Qualitative Cases

We provide additional qualitative examples to complement the hard-case results in the main text. The cases below cover all four hard phenomena used in E-ABSA20K-Hard: product-part mentions, colloquial sarcasm, contrastive sentences, and implicit opinion expressions. Targets follow the same normalized schema used throughout the paper: `PRODUCT` denotes an implicit product-level target, `PRODUCT:xx` denotes an explicitly mentioned product target, and `PRODUCT_PART:xx` denotes an explicitly mentioned product-part.

Across these cases, SeqKD typically remains locally plausible but drifts in target–aspect structure once generation departs from teacher states. MiniLLM is more tolerant to local deviations, yet often backs off to coarser target/aspect structures or produces plausible but non-grounded targets. In contrast, STAR-OPD more reliably preserves target grounding, fine-grained aspect bindings, and complete tuple structure.

These examples mirror the quantitative hard-case trends in the main text. Product-part cases and contrastive sentences highlight structural target–aspect failures, including binding drift, target collapse, and error propagation across tuples. The

(a) product-part (womenbags)	
Review	The bag looks elegant, but the zipper got stuck on the second day.
Gold	(PRODUCT:bag, Design Aesthetic, *looks elegant*, pos); (PRODUCT_PART:zipper, Hardware Quality, *got stuck*, neg)
SeqKD	(PRODUCT_PART:zipper, Ease of Use, *got stuck*, neg) [binding drift]
MiniLLM	(PRODUCT_PART:handle, Hardware Quality, *got stuck*, neg) [hallucinated target]
STAR-OPD	(PRODUCT_PART:zipper, Hardware Quality, *got stuck*, neg) [correct grounding + binding]

(b) Contrast (furniture)	
Review	The chair looks modern and fits my desk perfectly, but the seat feels hard and the wheels keep squeaking.
Gold	(PRODUCT:chair, Appearance & Design, *looks modern*, pos); (PRODUCT:chair, Fit & Space Compatibility, *fits my desk perfectly*, pos); (PRODUCT_PART:seat, Comfort, *feels hard*, neg); (PRODUCT_PART:wheels, Noise Performance, *keep squeaking*, neg)
SeqKD	(PRODUCT_PART:seat, Ergonomics, *feels hard*, neg); (PRODUCT_PART:wheels, Ergonomics, *keep squeaking*, neg) [error propagation]
MiniLLM	(PRODUCT:chair, Comfort, *feels hard*, neg); (PRODUCT:chair, Noise Performance, *keep squeaking*, neg) [target collapse]
STAR-OPD	(PRODUCT_PART:seat, Comfort, *feels hard*, neg); (PRODUCT_PART:wheels, Noise Performance, *keep squeaking*, neg) [full structure preserved]

Figure 6: Additional qualitative cases: product-part and contrastive reviews.

(c) Sarcasm (makeup)	
Review	Love how this "long-lasting" lipstick vanishes before I finish my coffee.
Gold	(PRODUCT:lipstick, Longevity, *vanishes before I finish my coffee*, neg)
SeqKD	(PRODUCT:lipstick, Longevity, *long-lasting*, pos) [surface cue imitation]
MiniLLM	(PRODUCT:lipstick, Overall, *vanishes before I finish my coffee*, neg) [coarse aspect fallback]
STAR-OPD	(PRODUCT:lipstick, Longevity, *vanishes before I finish my coffee*, neg) [fine-grained recovery]

(d) Implicit opinion (dresses)	
Review	It zipped up fine, but I couldn't sit down in it.
Gold	(PRODUCT, Ease of Use, *zipped up fine*, pos); (PRODUCT, Comfort, *couldn't sit down in it*, neg)
SeqKD	(PRODUCT, Fit, *couldn't sit down in it*, neg) [neighboring aspect confusion]
MiniLLM	(PRODUCT, Overall Satisfaction, *couldn't sit down in it*, neg) [coarse aspect fallback]
STAR-OPD	(PRODUCT, Comfort, *couldn't sit down in it*, neg) [correct implicit mapping]

Figure 7: Additional qualitative cases: sarcasm and implicit opinion reviews.

sarcasm and implicit opinion expressions further show that the gains of STAR-OPD do not come only from better local sentiment prediction, but also from more stable recovery of fine-grained aspect structure under ambiguous evidence.

H Full Prompt Template

```
You are an expert in fine-grained sentiment
analysis for e-commerce product reviews.

Task: Extract all aspect sentiment quadruples.
Format: (target, aspect_category, opinion, sentiment)

Target hierarchy:
  PRODUCT          - product in general
  PRODUCT:XX       - specific product type
  PRODUCT_PART:XX  - specific product part
  !! Do NOT invent targets absent from text.

Aspect: one from the predefined list.
Opinion: minimal exact phrase from the review.
Sentiment: positive | neutral | negative

Rules:
1. Extract every applicable (target,aspect) pair.
2. One quadruple per (target,aspect) combination.
3. Keep opinion as shortest faithful text span.
4. Handle negation: do not assign positive
   sentiment to negated phrases.
5. Handle concession: assign separate quadruples
   with correct polarity to each clause.
6. Multi-product: assign each opinion to
   the correct target product.
7. Verify: all targets exist in review text;
   all aspects from predefined list only.

<thinking>
Step 1: List all target entities in the review.
Step 2: Map each target to aspect categories.
Step 3: Locate minimal opinion span for each.
Step 4: Resolve sentiment under negation/concession.
Step 5: Verify no invented targets;
        all aspects in predefined list.
</thinking>

<quadruples>
[one quadruple per line]
</quadruples>

Review: "{{review_text}}"
Aspect list: {{category_list}}
```

I Aspect Taxonomy Example

We provide the women's bags taxonomy as a representative example of the fine-grained aspect space used in E-ABSA20K. The full flat list is provided to the model without grouping information.

Table 7: Aspect taxonomy for women's bags (34 categories).

Theme	**Aspect Categories**
Overall	Overall Satisfaction
Material	Material Type, Material Quality, Material Accuracy, Water Resistance, Odor, Cleaning & Maintenance
Appearance	Design Aesthetic, Color Accuracy, Color Preference, Appearance
Size	Size Accuracy, Weight, Capacity, Shape Retention
Hardware	Hardware Quality, Hardware Design, Security Features
Strap	Strap Adjustability, Comfort
Interior	Interior Organization
Build	Construction Quality, Durability
Commerce	Price, Value for Money, Customer Service, Shipping Speed, Delivery Experience, Packaging, Return & Refund Policy
Context	Occasion Suitability, Brand Authenticity, Sizing Guidance Clarity, Accessories Included