EarlySciRev: A Dataset of Early-Stage Scientific Revisions Extracted from LaTeX Writing Traces

Léane Jourdan[1], Julien Aubert-Béduchaud[1], Yannis Chupin[1], Marah Baccari[1] and Florian Boudin[2]
[1]Nantes Université, École Centrale Nantes, CNRS, LS2N, UMR 6004, F-44000 Nantes, France
[2]Inria, LS2N, Nantes Université, France
{firstname.lastname}@univ-nantes.fr

Abstract

Scientific writing is an iterative process that generates rich revision traces, yet publicly available resources typically expose only final or near-final versions of papers. This limits empirical study of revision behaviour and evaluation of large language models (LLMs) for scientific writing. We introduce EarlySciRev, a dataset of early-stage scientific text revisions automatically extracted from arXiv LaTeX source files. Our key observation is that commented-out text in LaTeX often preserves discarded or alternative formulations written by the authors themselves. By aligning commented segments with nearby final text, we extract paragraph-level candidate revision pairs and apply LLM-based filtering to retain genuine revisions. Starting from 1.28M candidate pairs, our pipeline yields 578k validated revision pairs, grounded in authentic early drafting traces. We additionally provide a human-annotated benchmark for revision detection. EarlySciRev complements existing resources focused on late-stage revisions or synthetic rewrites and supports research on scientific writing dynamics, revision modelling, and LLM-assisted editing.

Keywords: text revision, dataset, LLM filtering

1. Introduction

Academic writing is an inherently demanding task, requiring precision, clarity, and conciseness, often under time pressure and frequently in a second language. To support this process, researchers increasingly rely on LLMs, which can rewrite entire passages in response to high-level instructions. These models offer unprecedented assistance for revising drafts, improving readability, and articulating complex ideas with clarity and fluency.

However, assessing the impact of LLMs on scientific writing remains challenging. The key data required for such analysis, *drafts, revisions, and writing traces*, are largely inaccessible. While most scientific papers are publicly available in their final or near-final form, the iterative writing process that produced them typically remains hidden.

Because scientific writing is inherently incremental, authors progressively refine arguments, restructure paragraphs, clarify explanations, and adjust claims. This process naturally generates various intermediate artifacts, including discarded sentences, rewritten paragraphs, and commented alternatives. Yet these traces are rarely preserved in publicly accessible resources. As a result, existing research on revision modelling predominantly relies on late-stage revisions (e.g., between submission versions) or synthetic rewrites. Early drafting stages, where substantial conceptual and structural changes occur, remain underexplored.

The lack of access to early-stage revisions limits our ability to study authentic writing dynamics, measure quality improvements, train models that support in-depth revision, and evaluate the role of LLMs in scientific writing. It also hinders systematic investigation of issues such as stylistic homogenisation, bias propagation, or factual distortion introduced during automated rewriting.

This inaccessibility largely stems from ethical, legal, and ownership concerns: drafts may contain personal information, unpublished ideas, or confidential comments, and they often relate to papers that later fall under publisher copyright.

In this paper, we introduce EarlySciRev, a large-scale dataset of early-stage scientific revisions automatically extracted from arXiv LaTeX source files. Our key insight is that LaTeX comments often preserve discarded or intermediate versions of sentences and paragraphs. By mining these commented segments and aligning them with nearby final text, we recover fine-grained revision pairs that reflect authentic author rewriting. We present a complete pipeline[1] for (i) collecting computer science papers from arXiv, (ii) cleaning and processing LaTeX sources, (iii) extracting candidate revision pairs from commented text, and (iv) filtering genuine revisions using a LLM-based classification. Finally, we report an annotation study that benchmarks several LLMs on the revision detection task, and use the best model to curate the final dataset.

Our contributions are threefold:

- We propose a new method to retrieve early-

[1] https://github.com/JourdanL/EarlySciRev

Proceedings of the 3rd Int. Workshop on Natural Scientific Language Processing (NSLP 2026) @ LREC 2026, pages 119–126
12 May 2026.

stage scientific writing traces by exploiting commented content in LaTeX source files.

- We introduce EarlySciRev, a large-scale dataset of paragraph-level scientific revisions automatically extracted from arXiv papers, capturing authentic early drafting revisions.[2]
- We release a human-annotated benchmark for revision detection in scientific text, enabling systematic evaluation of both LLMs and future revision models.[2]

2. Related Work

A variety of datasets for text revision have been released over the years, reflecting growing interest in modelling writing and rewriting processes. Some of these datasets rely on synthetic revisions, generated either automatically (Ito et al., 2019) or manually by annotators who are not the original authors (Mita et al., 2024). While such resources are useful for controlled experimentation, they do not capture authentic authorial revision behaviour.

Among datasets that feature real author revisions, two primary sources have emerged: arXiv (Tan and Lee, 2014; Du et al., 2022; Jiang et al., 2022) and OpenReview (D'Arcy et al., 2023; Jourdan et al., 2024, 2025). These resources enable the study of revision behaviour in real-world scientific writing contexts.

Despite their usefulness, existing datasets have clear limitations. First, many are restricted in scope. Some focus exclusively on abstracts (Tan and Lee, 2014; Du et al., 2022), while others remain relatively small in scale, ranging from a few hundred (Jiang et al., 2022; Mita et al., 2024; Ruan et al., 2024) to a few thousand papers (Kuznetsov et al., 2022; D'Arcy et al., 2023; Dycke et al., 2023; Lin et al., 2023; Jourdan et al., 2025).

Second, most existing datasets primarily capture late-stage revisions, typically between near-final drafts posted on arXiv or submission platforms. These revisions often reflect already polished manuscripts prepared for public dissemination, rather than the exploratory and formative stages of writing. Early drafting phases, where substantial conceptual, structural, and stylistic changes are made, are therefore largely absent from current resources. To our knowledge, the only dataset that explicitly targets writing traces from the earliest stages of the drafting process is ScholaWrite (Wang et al., 2025). However, it is limited to only five papers, which restricts its applicability beyond exploratory analysis.

As a result, access to early-stage writing traces remains a major bottleneck for studying authentic scientific revision behaviour and for developing models that support in-depth revision. In this work, we hypothesize that such traces can be recovered directly from the LaTeX source files uploaded to arXiv. In practice, authors frequently leave commented-out sentences, paragraphs, or alternative phrasings in the source code (i.e. lines beginning with the "`%`" character). Mining these commented segments offers a promising and underexplored avenue for reconstructing fine-grained, early-stage scientific revisions.

3. Data Creation

This section describes the pipeline used to construct the dataset, from collecting raw data from arXiv to cleaning LaTeX sources and extracting candidate revision pairs.

3.1. Data Collection

We rely on the `arxiv-metadata-oai-snapshot.json`[3] dump downloaded on January 21, 2026. This metadata file contains 2,932,928 arXiv repositories, among which 596,118 are distributed under a permissive licence.[4]

In this work, we focus on computer science (CS) papers, which represent 286,747 with a valid licence. For each of these papers, we downloaded all available source archives (zip files), resulting in approximately 1.2 TB of source data.

3.2. LaTeX Source Processing

We first filter the source files to retain only LaTeX documents that contain potentially meaningful comments. Specifically, we select files that include commented lines that do *not* start with a backslash. This criterion excludes commented-out LaTeX commands, while preserving comments that may contain previous versions of sentences or paragraphs.

We then apply the following cleaning steps:

1. We retain only the content between `\begin{document}` and `\end{document}`.
2. We remove non-textual environments, including `table`, `figure`, `align`, `tikz`, and `algorithm`.
3. We replace each `equation` environment with a special token `[EQUATION]`, as equations may appear within running text.
4. We remove LaTeX commands that do not contain textual content (e.g., `\appendix`, `\vs-`

[2] https://huggingface.co/datasets/taln-ls2n/EarlySciRev

[3] https://www.kaggle.com/datasets/Cornell-University/arxiv

[4] CC BY-NC-SA 4.0, CC BY-SA 4.0, CC BY 4.0, Public Domain List, CC BY-NC-SA 3.0, CC BY 3.0, CC0 1.0

pace), while preserving structural commands such as \section.

These steps ensure that the resulting text primarily consists of natural language content suitable for revision analysis.

3.3. Candidate Revision Pair Extraction

Our objective is to extract candidate revision pairs composed of: i) a block of uncommented text that appears in the compiled document (the *final paragraph*), and (ii) a block of commented text that may correspond to an earlier version (the *commented paragraph*). We define a *block* as a sequence of consecutive lines of the same type (commented or uncommented), not interrupted by an empty line or by a line of the other type.

For each commented block c, we compare it to the five preceding and five following blocks. For each neighbouring block that corresponds to a final paragraph f, we compute a normalised difference ratio based on the Levenshtein distance between the two texts. To account for cases in which a revision affects only part of a paragraph, we compute this distance using a sliding window over the final paragraph. We define this normalised difference ratio as:

$$d_{norm}(f, c) = lev(f, c)/\max(|f|, |c|) \quad (1)$$

where $lev(\cdot, \cdot)$ denotes the Levenshtein distance and $|\cdot|$ the character length. We treat a pair as a candidate revision when $d_{norm} < 0.7$. This threshold was set empirically on a subset of the dataset.

For each final paragraph, we retain all associated candidate commented revisions. Before concatenating uncommented blocks, we remove inline trailing comments to ensure that the resulting text matches the content of the compiled document.

4. Data Annotation

The automatic extraction procedure described above result in 1,269,976 candidate revision pairs. However, not all of these candidates correspond to genuine rewriting instances. We therefore introduce an additional filtering step based on LLMs. To select an appropriate prompting strategy and model, we first construct a human-annotated gold subset.

4.1. Human Annotation

The objective of the annotation campaign is to determine whether a given pair of paragraphs constitutes a genuine revision instance. We randomly sampled 500 candidate pairs, each consisting of a *commented paragraph* (representing a potential original version) and a corresponding *final paragraph*. The annotation was carried out by five annotators: two master's students, two junior researchers, and one senior researcher. All annotators had prior experience with scientific writing and NLP research.

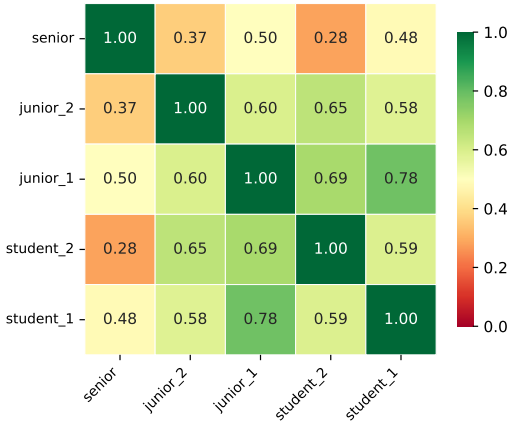

Figure 1: Pairwise Cohen's Kappa (κ_{Cohen}) scores between annotators.

For each paragraph pair, annotators answered the following binary question: *"Can the final paragraph be qualified as a revision of the original one(s)?"* Annotators were provided with detailed guidelines specifying decision criteria (see Appendix A). These guidelines were derived from the revision taxonomy proposed by Jourdan et al. (2025) and adapted to the present task.

Annotation was conducted using *Label Studio*. Paragraph pairs were displayed side by side. To facilitate comparison, identical text spans shared by the two paragraphs were highlighted (Appendix B).

Inter-annotator agreement. Figure 1 shows pairwise Cohen's κ_{Cohen} scores between annotators. Agreement varies across annotator pairs, with lower agreement observed for pairs involving the senior annotator.The senior annotator was more selective than other annotators on the pairs considered as revision.

To assess overall inter-annotator agreement, we apply Fleiss' κ_{Fleiss} under a partially overlapping design, in which each item is annotated by three annotators selected from the pool of five. The resulting $\kappa_{\text{Fleiss}} = 0.54$ indicates moderate agreement, according to the scale proposed by Landis and Koch (1977).

4.2. LLM-Based Filtering

Given the number of the extracted candidate pairs, manual annotation of the full dataset is infeasible. We therefore rely on automatic classification to identify genuine revision pairs. To select the most reliable approach, we evaluate several models and prompting strategies on the human-annotated subset. Since each item in this subset was labelled by

Instructions

```
You will receive two paragraphs P1
and P2. P2 is a final version of a
paragraph written for a scientific
article, and P1 is suspected to be an
 original version of P2 before
revision.
Can the P2 paragraph or a part of it
be qualified as a revision of P1?
If the changes only affect equations
or if here is too much equations it
is not considered a revision.
Answer with only one word "Yes" or "
No".
```

Figure 2: Prompt used in both *-PLUIE and LLM-choice settings.

three annotators, we use majority vote to derive the reference label for evaluation.

Approaches. We compare two LLM-based approaches: i) **Standard prompting**, where the model directly answers the binary question posed in the prompt, and the decision is extracted from the generated output. ii) ***-PLUIE** (Lemesle et al., 2026), a perplexity-based LLM-as-a-judge method that estimates the model's confidence without generating free-form text. In the *-PLUIE setting, the model estimates its confidence by computing the perplexity of candidate responses ("Yes" vs. "No") given the prompt. Positive values favour the revision hypothesis, negative values the opposite, and a binary decision is obtained by thresholding the score (default threshold 0, optimised a posteriori when annotations are available).

Models and setup. We evaluate several LLMs of varying sizes, including Qwen3 (4B,14B), phi-4 (14B), Olmo-3-7B-Instruct, and Llama-3.1-8B-Instruct. All models are run in a chat-completion setting using `bfloat16` precision. Sampling is disabled during inference (`do_sample=False`) to ensure deterministic outputs. The same prompt (Figure 2) is used across both standard prompting and *-PLUIE configurations, enabling a direct comparison between generation-based and perplexity-based classification for each model.

Results and model selection. Results are reported in Table 1. Across both approaches, Qwen3 (14B) achieves the best overall performance, with accuracy exceeding 80%. Balancing classification performance and computational efficiency, we select the *-PLUIE configuration with Qwen3 (14B) to filter the full dataset, using the optimal threshold determined on the annotated subset.

5. Dataset Statistics

Applying the selected LLM-based filtering strategy retain 578,440 revision pairs out of the initial 1,2M candidates (approximately 45.55%). These validated revisions correspond to 523,932 distinct final paragraphs. Among them, 46,192 paragraphs are associated with more than one revision candidate, reflecting cases where authors experimented with multiple alternative formulations before settling on a final version or cases where multiple previous paragraphs were merged into one. At the document level, the revisions are distributed across 104,023 articles.Those characteristics and more are summarised in Table 2.

Qualitative inspection suggests that revisions captured in EarlySciRev range from local fluency edits to more substantial restructuring and clarification of scientific arguments, to draft ideas left as to do with their fully written version.

The 500 paragraphs used for human annotation are also included and filtered at this step. Both the human annotated dataset and large LLM filtered one are openly available.[5]

	Model	Acc.	P.	R.	time
*-PLUIE	Qwen3 (4B) (thr=0)	0.77	0.77	0.76	22h
	Qwen3 (4B) (thr=-4.75)	0.78	0.75	0.81	
	Olmo-3 (7B) (thr=0)	0.74	0.73	0.76	35h
	Olmo-3 (7B) (thr=0.60)	0.74	0.78	0.67	
	Llama-3.1 (8B) (thr=0)	0.70	0.65	0.85	35h
	Llama-3.1 (8B) (thr=0.85)	0.73	0.75	0.67	
	phi-4 (14B) (thr=0)	0.77	0.71	0.92	55h
	phi-4 (14B) (thr=2.15)	0.79	0.75	0.85	
	Qwen3 (14B) (thr=0)	0.80	0.75	0.90	62h
	Qwen3 (14B) (thr=5.55)	**0.82**	0.80	0.86	
LLM-choice	Qwen3 (4B)	0.59	0.55	0.95	29h
	Llama-3.1 (8B)	0.59	0.55	**0.97**	46h
	Olmo-3 (7B)	0.68	**0.81**	0.45	45h
	phi-4 (14B)	0.78	0.80	0.75	81h
	Qwen3 (14B)	0.80	0.78	0.83	82h

Table 1: Alignment of LLM-based classifier to human majority vote. *Thr.* is the threshold used to binarise *-PLUIE values, *Acc.* the accuracy, *P.* the precision, *R.* the recall and *time* the estimated time to classify all the data. **Bold** values indicate the best results, and underlined values indicate the second-best results.

#rev	#paper	#rev/§	#words/§	% words diff
578,440	104,023	1.10	82.42	56.85

Table 2: Characteristics of EarlySciRev. In this order: number of revision pairs, number of articles, average number of commented paragraphs per final paragraph, average number of words per paragraph (final version), average percentage of difference in words per revision pair

[5] https://huggingface.co/datasets/taln-ls2n/EarlySciRev

6. Conclusion

We introduced EarlySciRev, a dataset of paragraph-level scientific revisions extracted from LaTeX writing traces in arXiv source files, together with a human-annotated benchmark for revision detection. By focusing on early drafting stages, this resource makes visible revision phenomena that are typically inaccessible in existing datasets.

EarlySciRev supports empirical study of scientific writing dynamics and provides a foundation for developing and evaluating revision models, including LLM-based systems. To facilitate reproducibility and further development, we release the full extraction and filtering framework, enabling updates as new papers become available.[6]

A future step could be to label all data with a revision intention and see how the distribution compare to dataset focusing on late stage revision.

7. Limitations

The current pipeline is restricted to computer science papers, as we only process licensed CS articles from arXiv. Writing practices and revision behaviours may differ across disciplines, limiting the generalizability of our findings. Extending the approach to other domains is a natural direction for future work.

Additionally, we do not explicitly control for the language of the paper. A part of papers submitted to arXiv are written in languages other than English. Our LLM-based filtering relies on prompts written in English, which may affect classification reliability for non-English texts. Adapting prompts to the language of each document or incorporating language identification into the pipeline could improve robustness.

Acknowledgements

This work was partly supported the AID-CNRS NaviTerm project (convention 2022 65 0079 CNRS Occitanie Ouest).

8. Bibliographical References

Mike D'Arcy, Alexis Ross, Erin Bransom, Bailey Kuehl, Jonathan Bragg, Tom Hope, and Doug Downey. 2023. Aries: A corpus of scientific paper edits made in response to peer reviews.

Wanyu Du, Zae Myung Kim, Vipul Raheja, Dhruv Kumar, and Dongyeop Kang. 2022. Read, revise, repeat: A system demonstration for human-in-the-loop iterative text revision. In *Proceedings of the First Workshop on Intelligent and Interactive Writing Assistants (In2Writing 2022)*, pages 96–108, Dublin, Ireland. Association for Computational Linguistics.

Nils Dycke, Ilia Kuznetsov, and Iryna Gurevych. 2023. NLPeer: A unified resource for the computational study of peer review. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5049–5073, Toronto, Canada. Association for Computational Linguistics.

Takumi Ito, Tatsuki Kuribayashi, Hayato Kobayashi, Ana Brassard, Masato Hagiwara, Jun Suzuki, and Kentaro Inui. 2019. Diamonds in the rough: Generating fluent sentences from early-stage drafts for academic writing assistance. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 40–53, Tokyo, Japan. Association for Computational Linguistics.

Chao Jiang, Wei Xu, and Samuel Stevens. 2022. arXivEdits: Understanding the human revision process in scientific writing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9420–9435, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Léane Jourdan, Florian Boudin, Richard Dufour, Nicolas Hernandez, and Akiko Aizawa. 2025. ParaRev : Building a dataset for scientific paragraph revision annotated with revision instruction. In *Proceedings of the First Workshop on Writing Aids at the Crossroads of AI, Cognitive Science and NLP (WRAICOGS 2025)*, pages 35–44, Abu Dhabi, UAE. International Committee on Computational Linguistics.

Léane Jourdan, Florian Boudin, Nicolas Hernandez, and Richard Dufour. 2024. CASIMIR: A corpus of scientific articles enhanced with multiple author-integrated revisions. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2883–2892, Torino, Italia. ELRA and ICCL.

Ilia Kuznetsov, Jan Buchmann, Max Eichler, and Iryna Gurevych. 2022. Revise and resubmit: An intertextual model of text-based collaboration in peer review. *Computational Linguistics*, 48(4):949–986.

J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

[6] https://github.com/JourdanL/EarlySciRev

Quentin Lemesle, Léane Jourdan, Daisy Munson, Pierre Alain, Jonathan Chevelu, Arnaud Delhay, and Damien Lolive. 2026. *-pluie: Personalisable metric with llm used for improved evaluation.

Jialiang Lin, Jiaxin Song, Zhangping Zhou, Yidong Chen, and Xiaodong Shi. 2023. MOPRD: A multidisciplinary open peer review dataset. *Neural Computing and Applications*, 35(34):24191–24206.

Masato Mita, Keisuke Sakaguchi, Masato Hagiwara, Tomoya Mizumoto, Jun Suzuki, and Kentaro Inui. 2024. Towards automated document revision: Grammatical error correction, fluency edits, and beyond. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 251–265, Mexico City, Mexico. Association for Computational Linguistics.

Qian Ruan, Ilia Kuznetsov, and Iryna Gurevych. 2024. Re3: A holistic framework and dataset for modeling collaborative document revision. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4635–4655, Bangkok, Thailand. Association for Computational Linguistics.

Chenhao Tan and Lillian Lee. 2014. A corpus of sentence-level revisions in academic writing: A step towards understanding statement strength in communication. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 403–408, Baltimore, Maryland. Association for Computational Linguistics.

Linghe Wang, Minhwa Lee, Ross Volkov, Luan Tuyen Chau, and Dongyeop Kang. 2025. Scholawrite: A dataset of end-to-end scholarly writing process.

A. Annotation Guidelines

Annotation Guidelines for Revision Detection

1. Introduction
The goal of this annotation campaign is to detect text revisions amid paragraphs originating from computer science scientific papers. A paragraph level revision is defined as a paragraph that is substantially modified for clarity, simplicity, style and other aspects. To that end, some final paragraphs have been selected and each one of them was provided with one or more original paragraphs that were under comment in the latex file. A final paragraph is a paragraph that is not commented and is suspected to be a revision of an original paragraph(s). In this task, we aim to characterize the final paragraphs' relationship with the suspected original paragraph(s), so that they can be classified as revisions or not down the line.

2. Annotation Task
Annotators are presented with a pair of paragraphs: an original version composed of one or several paragraphs and a final version. Their task is to answer the following question: *Can the final paragraph be qualified as a revision of the original one(s)?*
Annotators must select one of the following labels:
- **YES**: The final paragraph constitutes a revision of the original paragraph.
- **NO**: The final paragraph does not constitute a revision (e.g., different scientific content, the idea developed is not the same, introduces too much new information, or does not change the text).

As several original candidates are proposed, the annotator can answer Yes for multiple paragraphs (e.g. in cases of paragraph merging or iterative revision).

2.1 Positive example

Original Paragraph	Final paragraph
Therefore, the generalization rapidly decreases after augmentationinterrupted when training with a single background because the learning direction toward generalization about various backgrounds is not helpful to train. On the other hand, the training can have helpwhen their difculty is solved by augmentation, such as Figure 2(b) and Figure 2(c).	Therefore, the generalization rapidly decreases after augmentation is interrupted during training with a single background because the learning direction toward generalization about various backgrounds is not helpful to train. In contrast, the training can help when their difficulty is solved by augmentation (Figure 2(b), 2(c)).

2.2 Negative example

Original Paragraph	Final paragraph
In future research, the multi-mode characteristics will be studied to improve the representativeness of degradation features and the trendability of HI, and transfer learning approaches will be investigated to improve the generalization ability of the proposed framework and extend it to different systems.	Based on the ablation study, it can be concluded that the proposed SkipAE, inner HI-prediction block, and the HI-generating module jointly improve the ability of HI for reliable and accurate prognostics.

2.3 Annotation Procedure
For each pair of paragraphs (original and final), annotators must proceed as follows:
1. Read the final paragraph carefully to understand its scientific content and intent.
2. Read the original paragraph to identify any differences with respect to the final version.
3. Assess whether each original is rephrased in the final paragraph considering aspects such as: grammatical correctness, clarity and readability, fluency and coherence, appropriateness of scientific style.
4. Determine whether the scientific meaning of the paragraph is preserved in the final version.
5. Assign a label (YES or NO) according to the decision rules defined below.

Annotators should base their decision solely on the information contained in the paragraph pair and should not rely on external context. Also, annotators are prohibited to invent things.

2.4 Decision Rules
Annotators must apply the following rules when assigning labels:
Assign YES if at least one of the following conditions are met for a part or the whole paragraph:
1. The final paragraph is a revised version of the original paragraph, incorporating changes ranging from minor edits to substantial rephrasing.
2. The final version has been modified through the addition, the substitution or the deletion of ideas or facts.
3. The revision expands on the same idea with additional or withdrawn details.
4. The differences between the original and final paragraphs indicate the correction of document processing errors (e.g., parsing issues, segmentation errors, or misaligned paragraphs).

Assign NO if:
1. None of the above conditions are met.
2. If the annotator is unsure whether the revised paragraph constitutes a valid paragraph-level revision.
3. If there are only equations or code.
4. If the two paragraphs are the exact same.

If presented with multiple commented paragraphs for the same final paragraph, one or more commented paragraph can independently be considered as a revision. Classifying a commented paragraph as a revision does not disqualify the other proposed candidates. The same goes with the negative label : all the commented paragraphs may not qualify as a revision.

Table 3: Detailed annotation guidelines provided to the annotators for the revision detection task.

B. Annotation Examples

Reference	Candidate	Revised
We observed that AR2VP demonstrates superior entity perception outcomes, achieving the highest overall perception performance. This analysis underscores that current V2X technologies rarely rely on RSUs to expand perception horizons. In contrast, AR2VP harnesses the latent strengths of RSUs to address intra-scene changes, which enhances the vehicle's ability to adapt to dynamic scenes, consequently elevating the overall perception capabilities. However, AR2VP does exhibit a performance drawback in pedestrian detection, implying a particular challenge in detecting small targets.	We find that AR2VP shows superior entity perception results, with overall perception performance being the best. This analysis suggests that existing V2X technologies merely utilize RSU to extend perception horizons. In contrast, AR2VP leverages the latent advantages of RSU to model intra-scene changes, further enhancing vehicle adaptability to scene dynamics, thereby augmenting overall perception capabilities. However, in comparison to V2V, AR2VP exhibits a performance disadvantage in pedestrian detection, indicating a certain discrepancy in detecting small targets.	yes
Under \cref{ass:runtimecomplexity} and following \cref{tab:setops}, the outer approximative Minkowski sum from \cref{prop:minkSum_polyzono}, the Minkowski difference, and the linear map in the computation of the outer approximation $\outerBRSAE{-t}$ are all $\bigO{n^3}$, while the computation of the inner approximation $\innerBRSAE{-t}$ is dominated by the conversion to a constrained zonotope, which is $\bigO{n^4}$.	Under \cref{ass:runtimecomplexity}, the computation of the outer approximation $\outerBRSAE{-t}$ is marginally dominated by the over-approximative Minkowski sum, which is $\bigO{(\cons{}+2n)n, since the Minkowski difference and linear map are at most $\bigO{(\cons{}+2n)n \steps{}n}$ and $\bigO{(\cons{}+2n)n^2}$, respectively; all these operations are essentially $\bigO{n^3}$ for $n \gg \steps{}$ and under \cref{ass:runtimecomplexity}.	yes
$$\sum_{t=1}^T \sum_{y \in \{0, 1\}} p_t^y \hat{\ell}_t(y) - \inf_{j \in [N]}\sum_{t=1}^T \hat{\ell}_t(\mathcal{E}_t^{j}) \leq \frac{\ln N}{\eta} + \eta \sum_{t=1}^T \hat{\ell}_t(1) + \eta\sum_{t=1}^T p_t^1(1 - p_t^1) \hat{\ell}_t(0)^2 + \eta\sum_{t=1}^T p_t^1 \hat{\ell}_t(1)^2.$$ \end{lemma}	$$\sum_{t=1}^T \sum_{y \in \{0, 1\}} p_t^y \hat{\ell}_t(y) - \sum_{t=1}^T \sum_{y \in \{0, 1\}} \hat{\ell}_t(\mathcal{E}_t^{j}) \leq \frac{\ln N}{\eta} + \eta \sum_{t=1}^T \hat{\ell}_t(1) + \eta\sum_{t=1}^T \sum_{y \in \{0, 1\}} p_t^y \hat{\ell}_t(y)^2.$$	no
\iuhead{Date of fault-triggering test creation and modification:} We identified all the commits that are associated with the fault-triggering tests and analyzed when the commits happened (e.g., before or after the bug was reported/fixed). We used the git command "\inlineCode{git log -L:[funcname]:[file]}" to identify the list of commits that modified the fault-triggering test and the modification date.	We collected the date and time information provided in the output of this command to track the development activities associated with each fault-triggering test. This allowed us to better understand how the fault was identified and resolved over time with respect to the changes in fault-triggering tests. \peter{maybe remove this, I don't quite get this detail}Note that if a test is inherited from a parent class, we perform our analysis directly on the parent class since any changes would only occur in that class.	no

Table 4: Sample of annotated paragraph pairs, identical chain of texts are in green.