Automated Focused Feedback Generation for Scientific Writing Assistance

Eric Chamoun, Michael Schlichtkrull , Andreas Vlachos
Department of Computer Science, University of Cambridge
{ec806,mss84,av308}@cam.ac.uk

Abstract

Scientific writing is a challenging task, particularly for novice researchers who often rely on feedback from experienced peers. Recent work has primarily focused on improving surface form and style rather than manuscript content. In this paper, we propose a novel task: automated focused feedback generation for scientific writing assistance[1]. We present SWIF^{2}T: a Scientific WrIting Focused Feedback Tool. It is designed to generate specific, actionable and coherent comments, which identify weaknesses in a scientific paper and/or propose revisions to it. Our approach consists of four components - planner, investigator, reviewer and controller - leveraging multiple Large Language Models (LLMs) to implement them. We compile a dataset of 300 peer reviews citing weaknesses in scientific papers and conduct human evaluation. The results demonstrate the superiority in specificity, reading comprehension, and overall helpfulness of SWIF^{2}T's feedback compared to other approaches. In our analysis, we also identified cases where automatically generated reviews were judged better than human ones, suggesting opportunities for integration of AI-generated feedback in scientific writing.

1 Introduction

Peer feedback is essential in the scientific communication process, facilitating the development, interpretation, and dissemination of new findings (Kuhn, 1962). Yet, delivering high-quality feedback is both time-intensive and demanding (Horbach and Halffman, 2018). In response to this challenge, scientists are increasingly utilizing tools such as Grammarly[2], to improve the clarity of their manuscripts (Jourdan et al., 2023). However, these tools focus primarily on enhancing expression clarity rather than refining the research content, and it has been argued that a scientific peer should prioritize scientific value over editorial comments to enhance the quality of a submission (Kelly et al., 2014).

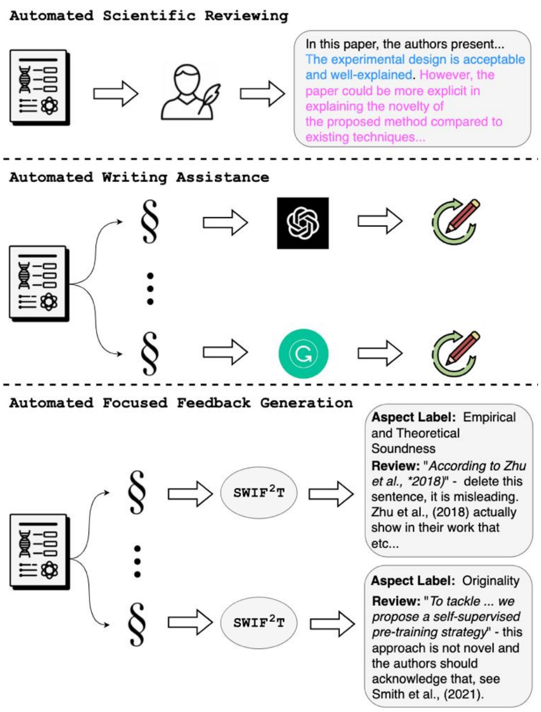


Figure 1: Overview of the proposed task.

Given the importance of peer feedback in the scientific process, predicting peer response would be a powerful assistive tool for writers. Several studies (Wang et al., 2020; Lin et al., 2023) have explored automated scientific reviewing. While the task is similar to scientific writing assistance, these studies focus on extracting high-level information for paper-level reviews, lacking specificity. Additionally, their emphasis is often on predicting conference acceptance rather than revising the manuscript. More recently, Liang et al. (2023) investigated the usefulness of GPT-4-generated comments on entire scientific papers. Their results indicate that, while such feedback benefits researchers, a crucial limitation lies in the models' inability to generate

[1] https://github.com/ericchamoun/FocusedFeedbackGeneration
[2] https://www.grammarly.com

Findings of the Association for Computational Linguistics: ACL 2024, pages 9742–9763
August 11-16, 2024 ©2024 Association for Computational Linguistics

specific and actionable suggestions.

In this paper, we introduce the task of automated focused feedback generation for scientific writing assistance (Figure 1). Focused feedback entails providing specific, actionable, and coherent comments that identify weaknesses in a paper or suggest revisions. To this end, we present SWIF^{2}T, a Scientific WrIting Focused Feedback Tool. This approach leverages multiple LLMs to decompose the process of generating scientific feedback into four components (Figure 2): planner, investigator, reviewer and controller. The planner designs a step-by-step plan involving the other components. It first instructs the investigator to answer questions using the paper or literature to enrich the paragraph under review with relevant context. Then, it directs the reviewer to leverage the gathered context to identify a specific portion requiring review, predict a weakness type (e.g. *Originality* in Figure 1) before generating focused feedback accordingly. The controller manages the plan's progress and coordinates other models' actions. Recognizing the pivotal role of the plan in shaping the review outcome, we also propose a plan re-ranking method aimed at optimizing the coherence, structure, and specificity of the generated plan, thereby enhancing the effectiveness of our scientific feedback generation process.

To assess our system, we compile a test set using human-written feedback derived from peer reviews. It consists of 300 examples where reviews highlight weaknesses in specific paragraphs of scientific papers. We conduct both automatic and human evaluations of SWIF^{2}T, comparing it against robust baselines such as CoVe (Dhuliawala et al., 2023) and GPT-4. The results demonstrate that SWIF^{2}T outperforms other models not only in terms of similarity to human-written reviews, but also in specificity, reading comprehension, and overall helpfulness of its comments. Our analysis reveals instances where SWIF^{2}T -generated comments are preferred over human-written ones, hinting at the potential for integration of AI-generated feedback in the scientific writing process. Finally, our analysis highlights the strengths and weaknesses of tested LLM-based approaches, outlining avenues for future research.

2 Related work

Scientific reviewing In recent years, researchers have made significant contributions by constructing datasets of peer reviews for research (Kang et al., 2018; Lin et al., 2023; Dycke et al., 2023). Review aspects such as *Originality* (Figure 1) have been explored in various studies, ranging from aspect score prediction (Kang et al., 2018) to aspect sentiment prediction (Chakraborty et al., 2020) and aspect-aware review generation (Yuan et al., 2022).

These datasets have facilitated the creation of tools that automatically generate reviews from papers (Wang et al., 2020; Yuan et al., 2022; Lin et al., 2023), although these reviews tend to focus on the high-level information of a scientific paper given that the tools consume entire papers at once. Recently, Robertson (2023) and Liang et al. (2023) investigated the potential of GPT-4 in assisting the peer review process. They found that GPT-4's feedback was comparable to human-written feedback in terms of helpfulness. However, they emphasized its limitations, notably its inability to provide specific, actionable feedback and to review a paper's low-level details. To address the lack of specificity in generated comments, D'Arcy et al. (2024) introduced MARG, a system that uses multiple LLM instances to handle different sections of scientific papers. Their proposed approach reduces generic feedback and increases comment quality. However, their focus is on generating paper-level reviews that emphasize the overall work rather than detailed aspects - which our study aims to address.

Text revision Several works have developed text revision tools to rewrite the intended text in scientific style. A recent survey by Jourdan et al. (2023) has grouped text revision tools into three different categories. The first one is Grammatical Error Correction (GEC), which seeks to detect and rectify grammatically incorrect or misspelled text (Bryant et al., 2023). The second category focuses on more substantial revisions than GEC, such as changing the structure of a sentence and rephrasing for clarity to generate error-free, proofread versions of early-draft sentences from scientific papers (Ito et al., 2019, 2020; Faltings et al., 2021; Du et al., 2022). The third category of text revision tools focuses on enhancing writing abilities through the visualization of the rhetorical structures in text (Knight et al., 2020). While these approaches provide high-quality writing assistance, they do not target providing constructive criticism on the research content.

Scientific paper-review discourse analysis Recently, there has been an increased interest in understanding the interaction between scientific papers and their reviews, fueled by the availability of large open peer review datasets. Shibani et al. (2023)

produced a visualization tool to study the interactions between a writer and GPT-3 when used by the former to produce real-time feedback.

Kuznetsov et al. (2022) presented three tasks: pragmatic tagging, i.e. communicative purpose prediction, for classifying review sentences by whether they highlight a strength, weakness or provide a summary of the paper among other possibilities, linking to establish fine-grained relationships between review and paper sentences, and version alignment to align two revisions of the same text.

More relevant to our research is D'Arcy et al. (2023), which explores the tasks of aligning high-level feedback to specific edits and generating revisions for scientific papers. This work differs from ours in that the presented tasks use reviews as input, whereas we focus on generating the reviews themselves. They discuss as a long-term goal the development of an "ideal writing assistant" that consumes relevant information from the paper and literature when generating an edit, which we implement in our work for scientific feedback generation.

3 Focused feedback generation

We define focused feedback as feedback that comprehends the analysed paper (*reading comprehension*), clearly indicates what specific elements in the paper need revision (*specificity*), and communicates to the author which actions they can take to address the issues (*actionability*). These ultimately contribute to the *overall helpfulness* of the feedback, i.e. the accuracy of the points discussed in the review and the extent to which they could assist the author(s) in revising their paper.

The envisioned model we aim to create concentrates on specific paragraphs p within a paper P to ensure no detail is missed. Given that a paragraph on its own may lack context, the model requires access to the entire paper and relevant literature. This access enables the model to provide more accurate and contextualized feedback. The desired output is comments highlighting weaknesses in the paragraph and/or suggesting revisions to the paper.

Prior studies (Chakraborty et al., 2020; Yuan et al., 2022; Ghosal et al., 2022) utilize aspects inspired by conference reviewing guidelines[3] to classify review sentences according to the aspects they address. In our study, we employ these aspects as weakness types to capture the focal points

[3] Example with definitions: https://acl2018.org/downloads/acl_2018_review_form.html

Figure 2: Illustration of our approach.

of reviews and exclude review sentences that do not constitute focused feedback during dataset compilation (Section 5.1). Specifically, we focus on the following five aspects[3]: *Replicability*, *Originality*, *Empirical and Theoretical Soundness*, *Meaningful Comparison*, and *Substance*. We exclude remaining aspects because they focus on refining the surface form (*Readability* and *Clarity*), are inherently subjective (*Impact*, *Motivation*) or do not constitute feedback (*Summary*, *Appropriateness*).

4 SWIF^{2}T

4.1 Description

Our system (Figure 2) relies on four components: planner, investigator, reviewer and controller. Given a paragraph in a paper, the planner initially formulates relevant questions and instructs the investigator to address them using information from the rest of the paper or literature to retrieve relevant context (Figure 2). For questions related to the literature, we use a search engine and feed retrieved pages into the investigator's QA pipeline.

To answer a question utilizing the content of the paper or a retrieved document, this pipeline first splits the document into segments of equal lengths before creating embeddings for each. A similarity search is then performed to retrieve the top 5 text chunks that are most similar to the question. Finally, our abstractive question-answering LLM generates an answer based on the retrieved paragraphs, responding with "I don't know." when questions cannot be answered from the provided

```
1- Investigator: Answer using the paper "What are the main objectives of this study?"
...
9- Investigator: Answer using the literature "What is the encoder/decoder model proposed by Sutskever et al. (2014)?"
...
11- Reviewer: Write a review using the gathered context.
```

Figure 3: Example of a plan generated by SWIF^{2}T . We include an example of an entire run in Appendix D.5.

documents. Then, once enough information has been gathered, the reviewer component utilizes it to identify a section of text requiring revision, predict an aspect to concentrate the feedback on before generating the focused feedback. After each step in the plan (Example in Figure 3), the controller decides whether the next step should be executed or replaced, enabling it to adapt the plan if needed.

4.2 Plan re-ranking

The success of our approach relies on the quality of the generated plan according to three criteria. Firstly, the steps within the plan must adhere to a correct format to ensure the program's successful execution. For instance, deviations involving undefined components would lead to program failure. Secondly, the plan must exhibit coherence, ensuring that its execution results in the incorporation of relevant context from the paper and literature, thus preparing for the generation of focused feedback. Lastly, the plan steps should be specific to the input paragraph, exploring relevant concepts within it.

We produce plans meeting this criteria by training a re-ranker to score plans w.r.t. the input paragraph. To generate training data, we select a subset $\boldsymbol{S}$ from our paragraph-review pairs training dataset (described in Section 5.1), where human-written reviews identify weaknesses in corresponding paragraphs. Given a paragraph and its review, an LLM is instructed to produce a plan facilitating the generation of the review based on the input paragraph. Two instances of this LLM are configured with distinct prompts: M_{struct}, instructed to generate a structured plan with correctly formatted steps for successful program execution, and $M_{unstruct}$ given no structuring instruction. For each paragraph $p_i \in \boldsymbol{S}$, our objective is to generate a plan optimizing the structure, coherence, and specificity criteria. Additionally, we aim to create three alternative plans, each lacking in one of these criteria.

Our strategy involves training a re-ranker to favor the optimal plan over alternatives, as it optimizes all criteria, including the one absent in each alternative plan. We leverage M_{struct} and $M_{unstruct}$ to generate four plans for each p_i:

- **Optimal**: M_{struct} is given as input p_i and its corresponding gold review r_i and generates a structured, coherent and specific plan to reach the review r_i from p_i.
- **Lacking coherence**: M_{struct} is given as input p_i and the human-written review r_j of another paragraph p_j from $\boldsymbol{S}$ and generates a structured, incoherent and specific plan to reach the review r_j from p_i.
- **Lacking structure**: $M_{unstruct}$ is fed p_i and its corresponding human-written review r_i and generates an unstructured, coherent and specific plan to reach the review r_i from p_i.
- **Lacking specificity**: For each of our five weakness types, we formulate a generic comment that addresses it (e.g. for *Originality*: "The ideas discussed in this passage are incremental as similar ideas have already been explored in the literature."). These comments do not reference any concept from the paragraph and, consequently, are suitable for any paragraph that might exhibit such a weakness. From this set, we extract the comment with the same weakness type as r_i and label it r_g. M_{struct} receives p_i and r_g as input, generating a structured, coherent and general plan to reach the review r_g from p_i.

In the training phase, the re-ranker receives a pair of inputs: one comprising a paragraph p_i and the optimal plan generated for it, and the other comprising p_i and one of the plans lacking in one of the desired characteristics. The re-ranker learns to score the former higher than the latter.

During inference, the planner generates multiple plans, and the system selects the one with the highest score through pointwise inference using our trained re-ranker.

5 Implementation

5.1 Dataset compilation

To construct our dataset, we utilize several existing peer review datasets that contain complete reviews and paper content: NLPeer (Dycke et al., 2023), F1000rd (Kuznetsov et al., 2022), PeerRead (Kang et al., 2018), MOPRD (Lin et al., 2023), the dataset by Wang et al. (2020), ASAP-Review (Yuan et al., 2022) and ARIES (D'Arcy et al., 2023). We extract linked paragraph-review pairs from ARIES and a subset of F1000rd. For the remaining datasets without annotated links, we automatically retrieve all paragraph-review pairs where the review quotes a sentence within the corresponding paragraph.

We implement a two-step data filtering process to ensure our reviews (1) identify weaknesses and/or propose revisions in a paragraph while (2) providing feedback on research and technical aspects. The first step involved actionability detection: we trained a model on the subset of *F1000Research* containing reviews annotated with communicative purposes (*Recap*, *Strength*, *Todo*, *Weakness*, *Structure*, *Other*), applied this model to our collected reviews, and only retained those indicating weaknesses (*Weakness*) or suggesting recommendations (*Todo*). Next, aspect annotation was performed on these actionable reviews. Here, we trained a model on ASAP-Review to predict the aspect addressed in our reviews from the following list: ***Replicability***, ***Originality***, ***Empirical and Theoretical Soundness***, ***Meaningful Comparison***, and ***Substance***, *Summary*, *Clarity* and *Motivation* before then only retaining those annotated with one of the five aspects highlighted in bold as motivated in Section 3. By fine-tuning RoBERTa models, we achieve high accuracy in pragmatic tagging (84.40% on test set, six labels) and aspect annotation (89.20% on test set, eight labels), enabling effective data filtering (implementation details in Appendix B). The resulting dataset comprises 2,581 paragraphs paired with human-written reviews.

For our test set, we first randomly selected 300 examples from our dataset. Manual quality assurance was then conducted to ensure that our reviews are actionable, relevant and address one of the five aspects we focus on (Section 3). We replaced inadequate reviews with randomly sampled instances from the dataset that met the criteria. For instance, reviews such as *Required revision in the abstract 'To our knowledge...standard manner'* (lacking specificity about the revision needed) or *The statement 'Types of ...' is not clear.* (misclassified as Soundness instead of Clarity) were excluded. However, we abstained from evaluating the helpfulness of individual reviews, recognizing the subjectivity of assessing this criterion, which could introduce bias into the selected dataset.

In the final test set, 51.3% of examples were sourced from ARIES, 20.3% from ASAP-Review and the remainder from other discussed datasets (dataset composition details in Appendix A).

5.2 Modelling details

We provide model and selection details in Appendix C, and SWIF^{2}T prompts in Appendix D.

Plan re-ranking The subset ***S*** of paragraph-review pairs, used for data generation, consists of 600 examples from our training set. As for optimization, let f be a scoring function with parameters θ that takes in a paragraph p and a plan i and outputs a relevance score $f_\theta(p,i)$ for i w.r.t. p according to our defined criteria. Let J_{p^+} be the set of optimal plans w.r.t. $p \in S$ and J_{p^-} be the set of plans lacking in one of the desired criteria w.r.t. p (see Section 4.2). During training, the implemented re-ranker takes in a triplet consisting of paragraph p and plans i and j, and estimates the probability $P(f_\theta(p,i) > f_\theta(p,j))$ that f assigns a higher score $f_\theta(p,i)$ to plan i w.r.t p, compared to $f_\theta(p,j)$ to plan j according to the defined criteria. We train the model with the following cross-entropy loss (based on Nogueira et al. (2019)):

$$L(\theta,(S,J)) = \sum_{p\in S;\{J_{p^+},J_{p^-}\}\in J} - \Bigg[\sum_{i\in J_{p^+}, j\in J_{p^-}} \log(P(f_\theta(p,i) > f_\theta(p,j))) + \log(1 - P(f_\theta(p,j) > f_\theta(p,i))) \Bigg] \quad (1)$$

We include details about models, performance and hyperparameter search in Appendix B. In 72% of the examples on the test set, the plan that optimizes the structure, coherence and specificity criteria is ranked the highest by our re-ranker, over plans lacking in one of these criteria.

Investigator The Google API is used to search the web. When PDFs are retrieved, Langchain's[4] PyPDF loader is used to extract the textual content. The LangChain API is used to split the documents into chunks of equal lengths and create an embedding for each chunk. To prevent our model from retrieving actual human reviews for the paragraphs – which would not be available for a newly-written paragraph – we block access to peer review websites such as OpenReview, PeerJ and F1000Research. Furthermore, given that F1000Research papers include reviews as part of the manuscripts, we manually extract relevant

[4] https://api.python.langchain.com/en

pages from these to avoid providing an unfair advantage to SWIF^{2}T in our evaluation. Our abstractive question answering model relies on providing the question-context pairs to GPT-4. It is tasked with responding with "I don't know" if the question cannot be answered from the given context.

Reviewer The GPT-4 based reviewer model uses in-context learning, wherein a representative training example for each weakness type is carefully chosen and incorporated into the prompt.

6 Evaluation

6.1 Evaluation setup

Human evaluation is conducted on a subset of 100 Machine Learning and NLP examples from the test set. Each model we consider is fed a paragraph and directed to identify a portion of text containing a weakness, predict a weakness type then generate a focused review. To guide our approach to human evaluation, we conducted several pilot studies.

In shaping our evaluation criteria, insights from these pilot studies and previous research guided our selection. Liang et al. (2023) highlighted the lack of specificity in LLM-generated reviews as a major weakness, emphasizing generic comments like "add more experiments on more datasets." To address this, we compare reviews based on their **specificity and actionability**, offering an indication of how often a system provides precise suggestions that allow authors to understand the necessary actions. Secondly, pilot studies revealed a tendency for models to hallucinate in instances where context is lacking, e.g., suggesting the addition of information present in another paragraph of the paper or making unfounded assumptions. The **reading comprehension** criterion captures such instances. The third criterion assesses **overall helpfulness**, considering the accuracy of the point and potential impact on helping the author revise the paper.

Additionally, the pilot studies revealed that the task's inherent complexity posed challenges for annotators in providing absolute review rankings. As a result, we opted for pairwise comparisons to mitigate subjectivity. No information about the approach generating a particular review was disclosed, and the order of the reviews was randomized. As there were four systems and three criteria, one example involved 18 blind pairwise comparisons - one for each criterion and each pair. In reporting our results, we opted to utilize dominance scores, which represent the sum of the instances where one approach was ranked superior to another, enhancing transparency regarding the frequency of such rankings based on specified criteria.

Furthermore, we considered the task's difficulty and estimated the time required to annotate an example. Based on the pilot studies, we found that completing one example, i.e., understanding the paragraph and reviews using the paper and literature before performing 18 pairwise comparisons, would take an average of 15 minutes.

We recruited 11 researchers from our research group, ranging from PhD students to associate professors, all with experience in reviewing and publishing at top ML/NLP conferences. Each annotator dedicated 2.5 to 3 hours to evaluate 10 examples, performing 18 pairwise comparisons for each example. The study therefore focused on a dataset of 100 examples, with 10 being annotated twice to measure inter-annotator agreement. Dominance scores were computed based on the results of the 1,980 undertaken comparisons. The inter-annotator agreement was computed using labels for 180 comparisons from 10 examples. To ensure consistency, detailed guidelines, criteria definitions, and examples were provided to annotators through the interface (examples in Appendix F).

	Reading comprehension	**Specificity**	**Helpfulness**
Tie %	50.15	31.08	30.63

Table 1: Tie percentage in human evaluation test set. Reading comprehension is the criterion that annotators have found most difficult to determine.

Including ties, the inter-annotator agreement is 0.3927 in Cohen κ, but rises to 0.7793 when considering only cases where one system is preferred over another. This is indicative of subjectivity in close cases, which is unsurprising given the nature of the task, but substantial agreement occurs when annotators prefer one system over another.

6.2 Baselines

We consider three baselines in our experiments:

GPT-4 (OpenAI, 2023) Similarly to Liang et al. (2023) and Robertson (2023), we explore the potential of GPT-4 in providing feedback on scientific papers. However, rather than operating at paper level, it takes a paragraph as input. This model mirrors the reviewer LLM in SWIF^{2}T but is missing the planner, investigator and controller components.

CoVe (Dhuliawala et al., 2023) We adapt CoVe, a system designed to mitigate hallucinations. We modify the **Factor+Revise** version of CoVe to generate plans for collecting information as part of its broader goal of writing focused feedback. Similar to the original implementation by Dhuliawala et al. (2023), we employ GPT-4 for planning, answering questions, cross-checking QA pairs, and generating the review. CoVe is essentially SWIF^{2}T without plan re-ranking and retrieval, except it includes a step to cross-check QA pairs before feeding them into the reviewer model. This step does not constitute a major difference given that the Controller model filters out unanswered QA pairs.

Human-written reviews We compare against human-written reviews. Importantly, these reviews are not treated as a "gold standard" but rather as a strong baseline. This is because we are comparing isolated snippets with model-generated feedback, which is specifically tailored for our task. The reviews were written under different conditions – specifically, peer reviewing rather than feedback writing. Furthermore, the passages were extracted in a manner that may strip comments of their original context.

7 Results

7.1 Human evaluation

The human evaluation reveals that SWIF^{2}T outperforms other baselines across all three criteria.

	Winners			
	SWIF^{2}T	**GPT-4**	**CoVe**	**Human**
SWIF^{2}T	-	22.75	28.90	29.50
GPT-4	61.75	-	48.00	45.75
CoVe	50.00	31.50	-	40.25
Human	58.75	43.00	50.25	-
Total	**170.50**	97.25	126.25	115.50

Table 2: Dominance table resulting from the pairwise comparisons in our human evaluation for the *specificity and actionability* criterion. A cell with coordinates (system$_a$, system$_b$) is the sum of the number of times system b was preferred over system a (+1) and those when these systems were rated equally (+0.5). Scores for examples that are annotated twice are averaged.

For specificity and actionability, LLM-based approaches are instructed through their prompts to extract a portion of text from the paragraph and focus the review on it. This explains the high specificity of reviews generated by LLM-based approaches, in contrast to human-written reviews that are typically shorter and more direct.

	Winners			
	SWIF^{2}T	**GPT-4**	**CoVe**	**Human**
SWIF^{2}T	-	24.50	23.25	32.25
GPT-4	52.00	-	41.50	44.75
CoVe	46.00	30.00	-	41.75
Human	45.50	34.25	39.25	-
Total	**143.50**	88.75	104.00	118.75

Table 3: Dominance table for *reading comprehension.*

In reading comprehension, SWIF^{2}T and human reviews perform best thanks to their access to the paper and literature. The brevity of human reviews, discussed in Section 7.3, poses challenges in gauging reading comprehension compared to LLM-generated reviews, which frequently integrate their knowledge within the review.

	Winners			
	SWIF^{2}T	**GPT-4**	**CoVe**	**Human**
SWIF^{2}T	-	24.00	28.00	30.75
GPT-4	60.50	-	52.50	48.50
CoVe	53.25	29.00	-	42.00
Human	58.00	39.50	47.00	-
Total	**171.75**	92.50	127.50	121.25

Table 4: Dominance table for *overall helpfulness*.

Finally, in terms of helpfulness, SWIF^{2}T clearly outperforms other baselines while CoVe reviews are slightly preferred to human-written ones.

7.2 Automatic evaluation

We performed an automatic evaluation on the full test set consisting of 300 examples to assess the significance of each component in our experiments. Ablations were conducted, comparing the performance of SWIF^{2}T to SWIF^{2}T$_{-RR}$ (SWIF^{2}T minus plan re-ranking) to evaluate the importance of plan re-ranking, CoVe to analyze the importance of information retrieval from the paper and literature, and vanilla GPT-4 to examine the importance of plans when compared to CoVe.

Table 5 shows the aspect label accuracy and F1 per category for each model w.r.t gold labels.

		GPT-4	CoVe	SWIF^{2}T $_{-RR}$	SWIF^{2}T
LA		24.00	30.70	33.00	**34.00**
Aspect F1	Soundness	30.60	**36.10**	33.51	31.40
	Meaningful comparison	8.70	24.40	31.25	**38.00**
	Replicability	41.30	30.90	**45.30**	42.90
	Originality	6.60	33.80	35.50	**38.90**
	Substance	9.20	24.50	22.40	**25.37**

Table 5: Aspect label accuracy and F1 per category

Aspect label accuracy results suggest that adding all the components together lead to better capturing the major weakness type of the paragraph. Notably, SWIF^{2}T and SWIF^{2}T $_{-RR}$, which incorporate retrieval, are more effective in predicting the *Meaningful comparison*, *Originality* labels. This suggests that the pipeline effectively retrieves relevant context from the paper and literature, enhancing the model's awareness. Without planning, we can also see that GPT-4 performs poorly on weakness types suggesting incorporating more information or comparing to other studies or methods. In contrast, it is effective in predicting the *Soundness* and *Replicability* weakness types, indicating that these are less dependent on planning and retrieval.

Model	METEOR	BLEU@4	ROUGE-L
GPT-4	18.13	28.50	18.37
CoVe	18.76	29.07	19.62
SWIF^{2}T $_{-RR}$	19.17	29.77	19.39
SWIF^{2}T	**20.04**	**30.06**	**20.44**

Table 6: Similarity of generated reviews to human reviews per model

Furthermore, Table 6 shows that SWIF^{2}T -generated comments exhibit highest similarity to human-written ones, with a progressive increase as components are added across most metrics.

7.3 Analysis

We manually analyze instances of our test set used in the human evaluation to shed light on the results, utilizing examples presented in Table 7 to explain our findings.

Human vs LLM-generated reviews In examining examples where LLM-generated reviews were preferred to human-written ones, we aimed to examine the potential of using LLM-based approaches for scientific writing assistance. Two key observations emerged. First, instances were identified where the point made by SWIF^{2}T differed but was preferred over the one in the human review (Example 1, Table 7). In these cases, neither was definitively superior, and the higher-scoring review was determined based on annotator preferences, showcasing the encouraging potential of LLMs in a blind comparison setup. Secondly, when the SWIF^{2}T-generated and human-written reviews conveyed the same point, annotators often preferred the former, appreciating the richer content that enabled higher specificity and exhibiting contextual knowledge as opposed to the more direct nature of human-written reviews (Example 2, Table 7). On average, comments of SWIF^{2}T and CoVe consisted of 61 and 58 tokens, respectively, while GPT-4 and human-written reviews were considerably shorter at averages of 52 and 43 tokens.

However, an important limitation of SWIF^{2}T is its imprecision regarding studies in the literature. The literature retrieval pipeline relies on summarized information in papers to answer questions about other studies rather than on the studies themselves. In cases where human-written reviews are specific about other studies whereas SWIF^{2}T -generated comments are vague ("other studies" in Example 3, Table 7), the former were consistently preferred. This highlights an area for future work to further enhance SWIF^{2}T 's performance.

Another scenario where human reviews were favored over those generated by LLM-based approaches was in cases of hallucinations (Example 4 in Table 7). While SWIF^{2}T demonstrated lower susceptibility to producing generic hallucinations, it still exhibited instances of hallucinating when lacking weaknesses to discuss. This highlights the need for a weakness detection model preceding review generation to avoid such cases.

Comparing different LLM-based approaches We delve into the factors influencing preferences among comments generated by our different systems. First, laying out a plan significantly enhances the likelihood of generating focused feedback. Even CoVe, which relies solely on internal knowledge, benefits from articulating a plan as it provides background, but also a roadmap and reasoning for the review (Example 5, Table 7). However, the lack of access to the paper and literature harms the performance of CoVe and GPT-4, as simple notations can confuse these models (Example 6, Table 7). For this reason, SWIF^{2}T , which incorporates planning and has access to both the paper and literature, outperforms these models.

Observation	Example
1. System makes a different point than human-written reviews that is preferred	**Human-written:** It's not clear which dataset was used in the experiment. **SWIF^{2}T :** The uneven distribution of individual voices, as indicated by the statement 'Individual voices are unevenly distributed, accounting for from 15 minutes to over 51 hours of recorded speech', may lead to potential biases in the training process. The authors should discuss this issue and, if possible, consider how they might address it.
2. Human-written review lacking specificity compared to system review making the same point	**Human-written:** In the introduction I'm not sure that the statement "RL methods optimize for short-term returns" is a fair statement. **SWIF^{2}T :** The authors claim that 'RL methods optimize for short-term returns instead of accounting for the complete learning process, they may get stuck in local minima and fail to explore the full search space'. However, not all RL methods are short-sighted. Some methods, for example, MCTS and DQN, do take the long-term reward into consideration when making decisions.
3. Lack of preciseness of SWIF^{2}T regarding studies from literature.	**Investigator:** *What criticisms or weaknesses have been pointed out in other studies regarding the use of kernel method-based theorems in infinitely overparameterized regimes of neural networks?* According to https://proceedings.neurips.cc/paper/2020/file/9afe487de556e59e6db6c862adfe25a4-Paper.pdf, " Other studies have pointed out that kernel method-based theorems require that the network weights stay very close to their initialization throughout training, and cannot handle regularizers or gradient noises." **SWIF^{2}T :** "While the authors claim that 'kernel method-based theorems...have non-vacuous guarantees on generalization', they fail to address the criticisms mentioned in other studies, particularly about the inability of such theorems to handle regularizers or gradient noises.",
4. Confident hallucinations	**GPT-4:** The claim 'we can see in Table 8 that our proposed system correctly disambiguate its causal meaning' needs supporting evidence. **CoVe:** The statement 'Our experimental results show that PriorityCut outperforms state-of-the-art models in identity preservation.' lacks substantial evidence or data to support this claim.
5. Usefulness of the plans	Relevant steps of plan **Investigator**: Answer "*Can you explain how the concept of the PDE solver works?*" **Investigator**: Answer "*What does it mean to integrate the latent space over time? How does a PDE solver achieve this?*" **CoVe**:While there is a good discussion on the utilization of latent spaces in PDE solutions; such as 'With an autoencoder network architecture, for example, the latent space can be used for discovering interpretable, low-dimensional dynamical models and their associated coordinates from high-dimensional data'. The paragraph would benefit from a discussion or explanation of how the features or dimensions compressed into the latent space are selected or prioritized for different types of problems or models.
6. Lack of context for CoVe and GPT-4	**CoVe:** The sentence 'Although rigorous model selection on L is necessary to draw scientific conclusions from the results, the purpose of this experiment is to illustrate the utility of the CSM kernel in this application.' lacks clarity in regards to what 'L' stands for. This missing context could make it hard for readers to fully comprehend the content of this sentence.
7. SWIF^{2}T applied to our study (v1)	**Human-written:** [Section 5.1] The paper mentions compiling a dataset of 300 peer reviews citing weaknesses but provides limited details on the dataset's size, diversity, and sources. A more thorough explanation of the dataset's composition of the broader context. **SWIF^{2}T :** [Section 5.1] While it is clear that several existing datasets are used in construction of a new dataset, the exact specifics of how these comparative datasets are actually combined is not transparent enough in "... " More details on the pragmatic tagging and aspect annotation process would also help in ensuring replicability of the experiment. **Human-written:** [Section 6.1] The paper mentioned that 11 experts were hired, with each one evaluating 10 samples. What are the criteria? How to guarantee the criteria in each group are the same? How challenging is the task? **SWIF^{2}T :** [Section 6.1] There might be room for providing more details about the annotators, such as their level of expertise. A more precise description of the expertise levels of the annotators is beneficial for the reader to assess the validity of the results.

Table 7: Observations and examples from our manual analysis.

Practical use case Finally, to motivate the potential use of our task, we apply SWIF^{2}T on sections 5.1 and 6.1 of the previous version of our paper. The generated comments (Example 7 in Table 7) align with some of the feedback received across the peer reviews, which highlights the potential of our tool in providing revisions to a paper ahead of its submission to human reviewers.

8 Conclusion

In this study, we introduced the task of automated focused feedback generation for scientific writing assistance. It involves generating specific, actionable and coherent comments which identify weaknesses in a paper and/or suggest revisions. We presented SWIF^{2}T , a system comprising planner, investigator, reviewer, and controller components. It is designed to gather context for a given paragraph and generate focused feedback. Through automatic and human evaluations, SWIF^{2}T demonstrated superiority over other LLM-based approaches in similarity to human-written reviews, specificity, reading comprehension and overall helpfulness of its comments.

Limitations

In this study, we introduced SWIF^{2}T, a system for automated focused feedback in scientific writing assistance. While our model outperforms other LLM-based approaches and shows promise in aiding the feedback process, it is crucial to acknowledge limitations in real-world applications.

Efficiency A major drawback is its cost and time intensity, relying on GPT-4 and Google queries. This limits accessibility. While the overall runtime depends on various factors including number of steps in the plan and complexity of the Google queries, SWIF^{2}T runs have averaged 7 minutes and 11 seconds across the 300 examples of our dataset.

Usage of Google for literature search Using the first few pages of Google searches to answer questions about the literature has some limitations that are worth highlighting. First, using the Google API may lead to answering questions from non-scientific sources. Nonetheless, an analysis of our retrieval system's top sources indicates a predominantly scientific focus (See Appendix E).

Additionally, this methodology might miss studies that are not well-cited or well-known, or focus on works from prestigious universities that are written in English. As a result, relevant related studies might be missed, which leads to a bias. An area of future work could involve developing a more sophisticated approach for literature-augmented question answering and context retrieval.

Limitations of LLMs for feedback generation LLMs are biased by their training data, narrowing the scope of generated reviews. Furthermore, variability exists in the models' generations, which can lead to favorable or less favorable suggestions.

Intended uses and ethics

Our envisioned use case for our system is to aid writers in refining drafts of scientific papers. Writers input a specific section into the system, and within minutes, receive constructive suggestions, along with structured plans and QA pairs for context and explanation. This iterative process continues until the writer is satisfied. Similar to feedback received from supervisors or peers, we expect the writer to exercise judgment in deciding whether a suggestion is worth implementing. Importantly, we want to emphasize that our tool is not intended to replace human reviewing. Peer review serves multiple purposes beyond providing suggestions, including spotting problems and preventing manuscripts that should not be published from reaching publication stage. The aim of our tool is to assist writers in submitting the best possible paper version to a conference or journal.

References

Christopher Bryant, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe. 2023. Grammatical Error Correction: A Survey of the State of the Art. *Computational Linguistics*, 49(3):643–701.

Souvic Chakraborty, Pawan Goyal, and Animesh Mukherjee. 2020. Aspect-based Sentiment Analysis of Scientific Reviews. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*, JCDL '20, page 207–216, New York, NY, USA. Association for Computing Machinery.

Mike D'Arcy, Tom Hope, Larry Birnbaum, and Doug Downey. 2024. Marg: Multi-agent review generation for scientific papers.

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.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. Chain-of-Verification Reduces Hallucination in Large Language Models.

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.

Felix Faltings, Michel Galley, Gerold Hintz, Chris Brockett, Chris Quirk, Jianfeng Gao, and Bill Dolan. 2021. Text Editing by Command. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5259–5274, Online. Association for Computational Linguistics.

Tirthankar Ghosal, Sandeep Kumar, Prabhat Kumar Bharti, and Asif Ekbal. 2022. Peer review analyze: A novel benchmark resource for computational analysis of peer reviews. *PLOS ONE*, 17(1):1–29.

Serge P.J.M. Horbach and Willem Halffman. 2018. The changing forms and expectations of peer review. *Research Integrity and Peer Review*, 3.

Takumi Ito, Tatsuki Kuribayashi, Masatoshi Hidaka, Jun Suzuki, and Kentaro Inui. 2020. Langsmith: An Interactive Academic Text Revision System. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 216–226, Online. 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.

Léane Jourdan, Florian Boudin, Richard Dufour, and Nicolas Hernandez. 2023. Text revision in Scientific

Writing Assistance: An Overview. *13th International Workshop on Bibliometric-enhanced Information Retrieval (BIR 2023)*, pages 22–36.

Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. 2018. A Dataset of Peer Reviews (PeerRead): Collection, Insights and NLP Applications. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1647–1661, New Orleans, Louisiana. Association for Computational Linguistics.

Jacalyn Kelly, Tara Sadeghieh, and Khosrow Adeli. 2014. Peer Review in Scientific Publications: Benefits, Critiques, & A Survival Guide. *EJIFCC*, 25:227–43.

Simon Knight, Sophie Abel, Antonette Shibani, Yoong Goh, Rianne Conijn, Andrew Gibson, Sowmya V., Elena Cotos, Agnes Sandor, and Simon Buckingham Shum. 2020. Are You Being Rhetorical? A Description of Rhetorical Move Annotation Tools and Open Corpus of Sample Machine-Annotated Rhetorical Moves. *Journal of Learning Analytics*, 7:138–154.

Thomas S. Kuhn. 1962. *The Structure of Scientific Revolutions*. University of Chicago Press, Chicago.

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.

Weixin Liang, Yuhui Zhang, Hancheng Cao, Binglu Wang, Daisy Ding, Xinyu Yang, Kailas Vodrahalli, Siyu He, Daniel Smith, Yian Yin, Daniel McFarland, and James Zou. 2023. Can large language models provide useful feedback on research papers? A large-scale empirical analysis. In *arXiv preprint arXiv:2310.01783*.

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.

Rodrigo Frassetto Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with BERT. *CoRR*, abs/1910.14424.

OpenAI. 2023. GPT-4 Technical Report.

Zachary Robertson. 2023. GPT4 is Slightly Helpful for Peer-Review Assistance: A Pilot Study.

Antonette Shibani, Ratnavel Rajalakshmi, Faerie Mattins, Srivarshan Selvaraj, and Simon Knight. 2023. Visual representation of co-authorship with GPT-3: Studying human-machine interaction for effective writing. In *Proceedings of the 16th International Conference on Educational Data Mining*, pages 183–193. International Educational Data Mining Society.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Qingyun Wang, Qi Zeng, Lifu Huang, Kevin Knight, Heng Ji, and Nazneen Fatema Rajani. 2020. ReviewRobot: Explainable paper review generation based on knowledge synthesis. In *Proceedings of the 13th International Conference on Natural Language Generation*, Dublin, Ireland. Association for Computational Linguistics.

Weizhe Yuan, Pengfei Liu, and Graham Neubig. 2022. Can we automate scientific reviewing? *J. Artif. Int. Res.*, 75.

A Test dataset statistics

A.1 Details

	Full test set	Manual evaluation sample
Description	One datapoint consists of the following entries: unique ID, paper_id, paragraph, human_review, human_review_aspect.	
Size	300	100
Total number of papers	288	96
Datapoints distribution across papers	- 1 paper contributed 3 datapoints - 10 papers contributed 2 datapoints each - 277 papers contributed one datapoint each	- 4 papers contributed 2 datapoints each - 92 papers contributed 1 datapoint each

A.2 Composition

		Full test set		Sample	
Dataset	**Sources**	**Domains**	**Datapoints from dataset**	**Domains**	**Datapoints from dataset**
ARIES (D'Arcy et al., 2023)	Computer Science	OpenReview	154	ML and NLP	66
ASAP-Review (Yuan et al., 2022)	ML and NLP	ICLR and NeurIPS	61	ML and NLP	25
NLPeer (Dycke et al., 2023)	ARR, COLING, CONLL, F1000Research excluding F1000Rd annotated sample	ML and NLP Medical, Biology, Metascience, Public Health	35	ML and NLP	3
MOPRD (Lin et al., 2023)	PeerJ	Biology, Computer Science, Medical, Environment, Chemistry	25	ML and NLP	3
F1000Rd Annotated Sample (Kuznetsov et al., 2022)	F1000Research	Medical, Biology, Metascience, Public Health	16	N/A	0
Dataset from Wang et al. (2020)	ICLR, NeurIPS, ACL	ML and NLP	8	ML and NLP	3
PeerRead (Kang et al., 2018)	ACL	NLP	1	N/A	0

A.3 Extra information:

As the human evaluation was conducted by researchers from our group specializing in Machine Learning and NLP, we selected a subset of 100 datapoints within the same domain as utilized in the human evaluation. To achieve this, we initially randomly extracted 100 datapoints from our full dataset. Subsequently, we manually filtered out datapoints from domains other than Machine Learning and NLP, and randomly selected suitable replacements from the remaining dataset.

B Implementation

Model optimization was performed on the validation sets using the Huggingface hyperparameter search for communicative purpose prediction (for actionability detection) and aspect annotation [5].

	Communicative purpose prediction	Aspect annotation	Plan re-ranking
Task	Classification	Classification	Re-ranking
Evaluation	Accuracy	Accuracy	Recall@1
Majority baseline performance	30.10%	12.50%	25.0%
Model performance	84.40%	89.20%	72.0%

Table 8: Performance of different models on respective tasks.

Parameter	Communicative purpose prediction	Aspect annotation	Plan re-ranking
Checkpoint	roberta-large	roberta-large	bert-base-cased
Learning rate	1.8e-05	2.3e-05	{5e-05, 1e-06 **5e-06**, 1e-07}
Batch size	16	32	{**8**,16,32}
Epochs	6	10	**7**[6]
adam-epsilon	1e-08	1e-08	1e-08
Dropout	0.1	0.1	{0,**0.1**}
Weight decay	0.01	0.01	{**0**,0.01}

Table 9: Hyperparameters used during training. For plan re-ranking, tested values are reported and those leading to the best performance are in bold.

C Model

All components in SWIF^{2}T are based on GPT-4 (version: gpt-4-0613).

Initially, we tested GPT-3.5 and various open-source Large Language Models (LLMs), such as Alpaca (Taori et al., 2023). Similarly to Robertson (2023) who use GPT-4 in their study because they find that "only the most powerful language models are capable of producing coherent reviews", we find the overall performance of other LLMs in our initial experiments unsatisfactory. Additionally, SWIF^{2}T relies on multi-agent LLMs, which facilitates rapid error propagation. This affects overall effectiveness and introduces bugs. Therefore, considering that our paper introduces a new task, we utilized the most effective model available to showcase the potential of AI tools designed for focused feedback generation.

[5] https://huggingface.co/docs/transformers/hpo_train

[6] Max 10 with early stopping

D Prompts

D.1 Planner

System message You are the ReviewGPT Planner, a world class scientific reviewing assistant. You create plans using the Investigator and Reviewer AI agents to review paragraphs. You will ask the Investigator to gather context from both the web and the paper in the first few steps, then, at the end, the action you ask the Reviewer agent will be exactly “Write a review based on the gathered context.”. DO NOT add a single word to this sentence. Your output MUST be formatted as a numbered list. NEVER write a step that does not involve an action for the Investigator or the Reviewer agents.
User message You will be given a paragraph. Your task is to point out the weaknesses of this paragraph, i.e. ask questions to gather context from the paper and the web before reasoning over it and the paragraph to identify the weaknesses of the passage. Thinking step by step, break the process of scientific reviewing down into small, simple tasks. These should involve gathering context for the paragraph, i.e., gathering information from the paper (such that the paragraph can be understood, verified and criticized without requiring any access to the paper.) and from the literature (such that the Reviewer AI agent can understand cited studies, compare the paper againt other related studies, evaluate its originality and soundness and be aware of criticisms and limitations, all without needing any access to the literature. The questions should be self-contained and formulated to facilitate effective Google searches). The gathered information should allow the Reviewer to comment on the soundness, originality, replicability, meaningfulness of the comparison or the substance of the information discussed in the paragraph. As you make plans for other AI tools, each step should be solvable using one of the following actions:

1. Actor: Investigator | Action: Answer question using the paper | Parameters: question | Description: Answer the provided question from the provided paper. It is important that the query that is searched is a question ending in ‘?’.

2. Actor: Investigator | Action: Answer question using Google | Parameters: question | Description: Use google search to try to answer the provided question. It is important that the query that is searched on Google is a self-contained question ending in ‘?’.

3. Actor: Reviewer | Action: Write review | Parameters: | Description: Write a review that only points out the weaknesses and areas of improvement of a passage based on the plan so far. Can only be called once context has been gathered by another agent.

Your plan should be a numbered list. Steps should be in simple language, and mention which agent should do them.
I will give you now the golden rules by which you NEED to abide. It is of upmost important that none of these rules is broken:
Rule #1: Each step involves requesting the Investigator or the Reviewer to perform an action.
Rule #2: The plan begins with the Investigator answering questions using the paper. Each of these steps should start with “Search the paper to understand”. The next steps should request the Investigator to answer questions by searching the web. These should start with “Search the web to understand” and should be only about one idea. Questions answerable from the web should be self-contained such that they are understandable without referring to another step or the paper. Finally, the last step should be EXACTLY “Reviewer: Write a review based on the gathered context.”
Rule #3: The questions to the Investigator should ONLY ask about one concept at a time. For example, “Search the paper to understand what Attention is ” is valid but “Search the paper to understand what Attention is and how it works” is NOT.
IT IS IMPORTANT TO RESPECT THE THREE GOLDEN RULES I JUST GAVE YOU. Now, the paragraph you will review is:
{*paragraph*}

D.2 Controller

System message You are the ReviewGPT Controller, a helpful scientific reviewing assistant. You manage several other AI agents, passing directions to them from the user. You communicate directly with the other AI agents, and as such your answers MUST be ONLY valid json.
User message You are currently following an overall plan to point out the weaknesses in the paragraph:
{*paragraph*}
This is a log of your progress so far:
{*progress*}
The remaining steps are:
{*steps*}
The next step is:
{*next step*}
You will be given a list of actions. Your task is to decide what the best action to take is to accomplish the next step. Each action has several fields, separated by a vertical line (|). These are the actor who takes the action, the name of the action, the parameters that action requires, and a short description of the action. The options are:

- Actor: Investigator | Action: Answer question using the paper | Parameters: question | Description: Answer the provided question from the provided paper. It is important that the query that is searched is a question ending in '?'.
- Actor: Investigator | Action: Answer question using Google | Parameters: question | Description: Use google search to try to answer the provided question. It is important that the query that is searched on Google is a question ending in '?'.
- Actor: Reviewer | Action: Write review | Parameters: | Description: Write a review that only points out the weaknesses and areas of improvement of a passage based on the plan so far. Can only be called once context has been gathered by another agent.
- Actor: Controller | Action: Skip this step | Parameters: | Description: Skip the current step if it is unnecessary or impossible

Provide the best action to take. Your answer must be valid JSON. It should be a JSON object with four entries, "explanation", "actor", "action", "parameters". Actors and actions should be strings, parameters should be another JSON object. Explanations should be a string containing a step-by-step description of why you chose this action. Remember, necessary parameters can be found between curly brackets in the commands. Output JUST the command.

D.3 Investigator

User message Answer the question based on the context below. IF the question cannot be answered based on the context, return exactly 'I don't know'.
{*context*}
Question: {*question*}
Answer:

D.4 Reviewer

System message You are the ReviewGPT Reviewer, a world class AI assistant for scientific reviewers. You write a review that highlights the weaknesses and areas of improvements of a paragraph based on context given to you, and return results as valid JSON. You need to make sure that the review addresses a specific portion of the paragraph, that it is not generic and that it is constructive. If you think that no review is needed, then you can also say this. Also, make sure to use the given context to generate a review: a review that points out the absence of an information in the paragraph should not be made if this information is present in another paragraph of the paper, be careful this is very important!
User message You will be given a paragraph with the following context:
{*context*}
There are five possible review labels: Empirical and Theoretical Soundness, Meaningful Comparison, Substance, Originality, Replicability. Write a review that:

1. Selects and quotes a substring from the given paragraph.
2. Chooses the appropriate review label.
3. Writes a review sentence using the quoted substring, the review label and the context. It is IMPORTANT that you use the provided context to generate a sensible review.
4. Generates a JSON object with the keys "reasoning", "label" and "review". Below are examples that follow all these rules, use them as inspiration + {in-context learning examples}

D.5 Example of SWIF^{2}T run

Example

Paragraph [7]: The GF-SVM approach was deployed for the first time in the prostate cancer detection dataset, and it demonstrated a significant performance improvement over the existing models in this domain. The full features of the dataset were conducted in the experiments.
Gold review: The authors state (lines 205,206) that GF-SVM "demonstrated a significant performance improvement over the existing models in this domain." However, the authors have failed to perform any sort of statistical analysis to demonstrate that their method is significantly better, or even different from, the established methods (e.g. see [1]).

Plan

1. Investigator: Answer question using the paper: "What is the GF-SVM approach?"
2. Investigator: Answer question using the paper: "How is the GF-SVM approach applied to the prostate cancer detection dataset?"
3. Investigator: Answer question using the paper: "What improvements did the GF-SVM approach bring to the existing models in prostate cancer detection?"
4. Investigator: Answer question using the paper: "What are the full features of the dataset used in the experiments?"
5. Investigator: Answer question using Google: "What are the usual performance measures in prostate cancer detection models?"
6. Investigator: Answer question using Google: "Are there any specific considerations or challenges in handling datasets for cancer detection?"
7. Reviewer: Write a review based on the gathered context.

Retrieved context

What is the GF-SVM approach?

[7]Paragraphs vary in lengths from couple of sentences to full sections. However, in this case, a short paragraph lacking context was chosen to better illustrate context retrieval.

* The GF-SVM approach is an algorithm that combines the Genetic Folding (GF) algorithm with Support Vector Machines (SVM) for classifying patients with prostate cancer. It is a hybrid model that uses the SVM classifier with various conventional kernels to achieve high accuracy in classification.

How is the GF-SVM approach applied to the prostate cancer detection dataset?
* The GF-SVM approach is applied to the prostate cancer detection dataset by using the SVM classifier with several conventional kernels such as linear, polynomial, and RBF kernels. The performance of the GF-SVM approach is evaluated and compared to other ML approaches, and it is found to have superior accuracy.

What improvements did the GF-SVM approach bring to the existing models in prostate cancer detection?
* The GF-SVM approach brought superior accuracy performance compared to the six ML approaches in prostate cancer detection.

What are the full features of the dataset used in the experiments?
* The full features of the dataset used in the experiments are Radius, Texture, Perimeter, Area, Smoothness, Compactness, Symmetry, Fractal_dimension, and Diagnosis.

What are the usual performance measures in prostate cancer detection models?
* The usual performance measures in prostate cancer detection models are sensitivity, specificity, and the area under the ROC curve.

Are there any specific considerations or challenges in handling datasets for cancer detection?
* Yes, there are specific considerations and challenges in handling datasets for cancer detection, such as batch effects and the need to consider vital biological information.

Output
{"review": "The paragraph states, 'The GF-SVM approach was deployed for the first time in the prostate cancer detection dataset, and it demonstrated a significant performance improvement over the existing models in this domain.' This sentence attributes a significant performance improvement to the GF-SVM model compared to existing models but fails to provide specific performance metrics or a measure of significance. ", "label": "Substance", "reasoning": "The claim of significant improvement by the GF-SVM model over existing models lacks empirical support in terms of specific metrics (sensitivity, specificity, AUC-ROC) used in this domain, as mentioned in the given context. "}

E Sources retrieved by SWIF^{2}T

aclanthology.org	pubmed.ncbi.nlm.nih.gov
nature.com	proceedings.mlr.press
mdpi.com	proceedings.neurips.cc
arxiv.org	machinelearningmastery.com
link.springer.com	towardsdatascience.com
en.wikipedia.org	ncbi.nlm.nih.gov
journalofbigdata.springeropen.com	

Table 10: Sources retrieved at least 10 times by SWIF^{2}T

F Human evaluation interface example

Review 1 vs Review 4

Paragraph: We focus on the intersection of these two branches of research, and propose Egocentric Spatial Memory Networks (ESMN), a parameter-free module that encodes the memory in an ego-sphere around the agent. We emphasize the suitability of an ego-centric mapping computation graph for propagating gradients over long time horizons in an end-to-end manner, enabling an ego-centric memory of learned features, see Figure 1 . We demonstrate the superiority of learning features through the ESMN module, on both drone target shape reaching and object segmentation tasks.

Review 1: The claim that 'We demonstrate the superiority of learning features through the ESMN module, on both drone target shape reaching and object segmentation tasks.' needs additional substantiation. Comparative analysis with other approaches, the specific methods of evaluation, and the results should be explicitly described.

Review 4: The text mostly focuses on improvement, and hence it is hard to judge which parts are novel and what the contribution is.

Comparing the two above reviews regarding:

1. **Reading comprehension**: has the reviewer read the paper well?
2. **Specificity and Actionability**: how specific the review is about the problem in the paragraph and how actionable the review is for the reader.
3. **Overall helpfulness**: does implementing the feedback improve the quality of the submission?

Figure 4: Each comparison involves a paragraph and two reviews. Reviews are randomized at the start of the annotation and no information about the source of the review is provided to prevent bias.


Under review as a conference paper at ICLR 2021

EGOCENTRIC SPATIAL MEMORY NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Spatial memory, or the ability to remember and recall specific locations and objects, is central to autonomous agents' ability to carry out tasks in real environments. A key challenge here is dealing with partial observability (PO) as cameras and other sensors can only observe the parts of the world that are visible from their specific vantage point. The most common approach for dealing with PO is to aggregate observations in an external memory. However, most existing external memory modules do not leverage any spatial inductive bias. In this paper, we propose Egocentric Spatial Memory Networks (ESMN), which encodes the memory in an ego-sphere around the agent, enabling expressive 3D representations. ESMN can be trained via either imitation or reinforcement learning and improves both training efficiency and final performance against other memory baselines on both drone and manipulator visuomotor control tasks. The explicit ego-centric geometry also enables us to seamlessly combine the learned controller with other non-learned modalities, such as local obstacle avoidance. We further show applications to semantic segmentation on the ScanNet dataset, where ESMN naturally combines image-level and map-level inference modalities. Through our broad set of experiments, we show that ESMN provides a general computation graph for embodied spatial reasoning, and the module forms a bridge between real-time mapping systems and differentiable memory architectures.

1 INTRODUCTION

Egocentric spatial memory is central to our understanding of spatial reasoning in biology (Klatzky, 1998; Burgess, 2006), where an embodied agent constantly carries with it a local map of its surrounding geometry. Such representations have particular significance for action selection and motor control (Wang et al., 2018; Hinman et al., 2019). For robotics and embodied AI, the benefits of a persistent local spatial memory are also clear. Such a system has the potential to run indefinitely, and bypass both the memory and runtime complexities of large scale world-centric mapping. Some pieces of work in the mapping literature have already utilized this ego-centric formulation, with applications to drone obstacle avoidance (Fragoso et al., 2018), mono-to-depth from video (Liu et al., 2019), and planar robot mapping (Fankhauser et al., 2014).

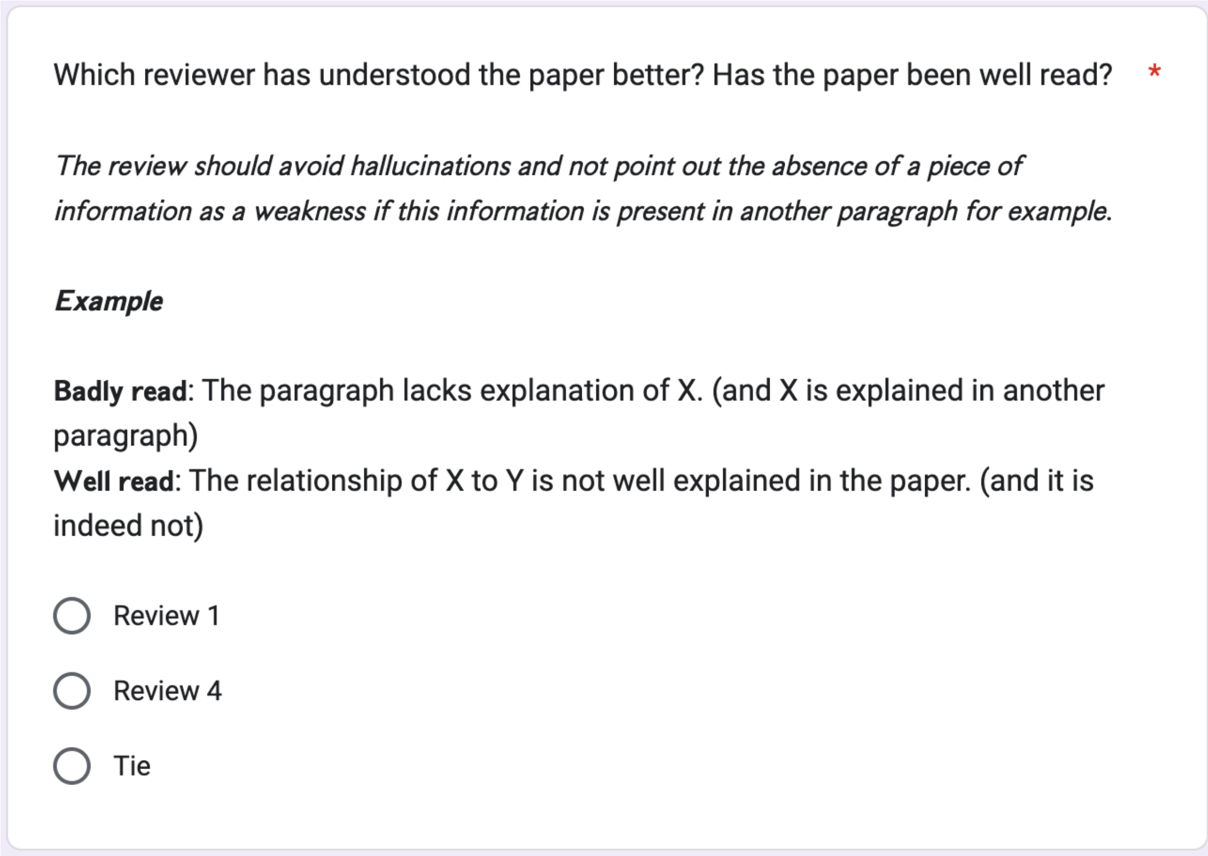
In parallel with these explicit ego-centric mapping systems, a new paradigm of differentiable memory architectures has arisen, where an explicit memory bank is augmented to a neural network which can then learn read and write operations (Weston et al., 2014; Graves et al., 2014; Sukhbaatar et al., 2015). When compared to Recurrent Neural Networks (RNNs), the persistent memory circumvents issues of vanishing or exploding gradients, enabling solutions to complex long-horizon tasks. These have also been successfully applied to visuomotor control and navigation tasks (Wayne et al., 2018), surpassing baselines such as the ubiquitous Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997).

We focus on the intersection of these two branches of research, and propose Egocentric Spatial Memory Networks (ESMN), a parameter-free module that encodes the memory in an ego-sphere around the agent. We emphasize the suitability of an ego-centric mapping computation graph for propagating gradients over long time horizons in an end-to-end manner, enabling an ego-centric memory of learned features, see Figure 1. We demonstrate the superiority of learning features through the ESMN module, on both drone target shape reaching and object segmentation tasks.

For other visuomotor control tasks, we show that even without learning features through the module, ESMN is able to consistently outperform baselines approaches. Through these experiments, we show

1

Figure 5: A link is provided to the annotator for convenient access to the version of the paper with the paragraph under review.


Which reviewer has understood the paper better? Has the paper been well read? *

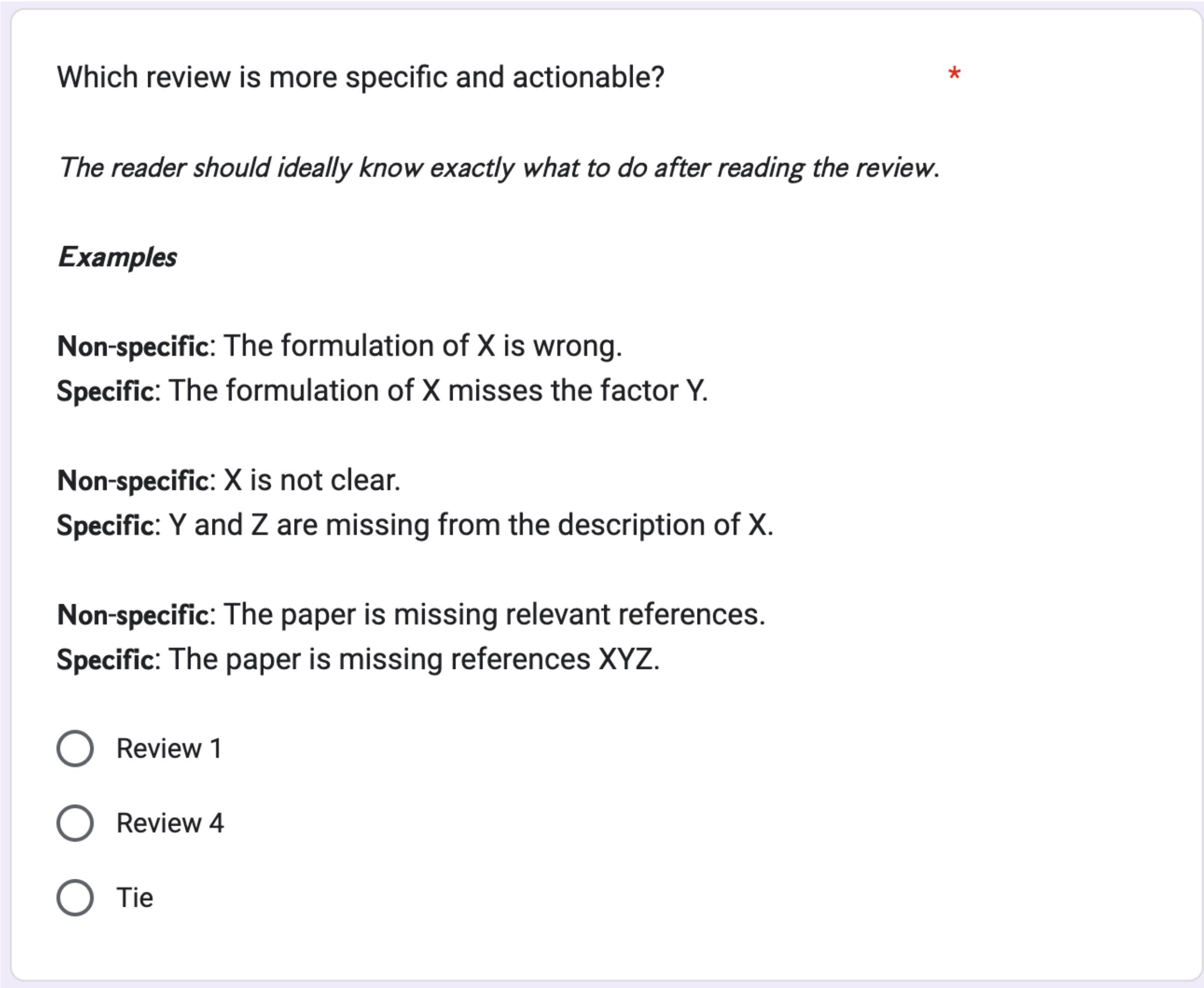
The review should avoid hallucinations and not point out the absence of a piece of information as a weakness if this information is present in another paragraph for example.

Example

Badly read: The paragraph lacks explanation of X. (and X is explained in another paragraph)
Well read: The relationship of X to Y is not well explained in the paper. (and it is indeed not)

- Review 1
- Review 4
- Tie

Figure 6: Comparison guidelines for the "*Reading comprehension*" criterion.



Which review is more specific and actionable? *

The reader should ideally know exactly what to do after reading the review.

Examples

Non-specific: The formulation of X is wrong.
Specific: The formulation of X misses the factor Y.

Non-specific: X is not clear.
Specific: Y and Z are missing from the description of X.

Non-specific: The paper is missing relevant references.
Specific: The paper is missing references XYZ.

- Review 1
- Review 4
- Tie

Figure 7: Comparison guidelines for the "*Specificity and actionability*" criterion.

Which review contributes more to the improvement of the submission? *

Here, there are two factors to consider. First and most importantly, ***the accuracy of the point*** *made in the review: if the point made in the review is valid. Second, if both points are valid, then* ***which point is more important****?*

Examples

Factor 1:

Not helpful: The paper should test the model on more datasets such as FEVER. (*and the paper is about machine translation. If the comment about more testing on more datasets is unfounded and it is more of a 'hallucination' then the review is not helpful*)
Helpful The authors should provide more information about the hyperparameters used for replicability. (*and the authors have really not provided any information about hyperparameters, making the implementation less replicable*)

> *The point made in this example is that* ***even though*** *the helpful review constitutes a* ***small change****, the fact it is* ***accurate*** *makes it* ***more helpful than a wrong review mentioning a bigger change****.*

Factor 2:

Less helpful:The authors should provide more information about the hyperparameters used for replicability. (*and the authors have really not provided any information about hyperparameters, making the implementation less replicable*)

More helpful:The authors claim that they achieve SOTA performance by comparing against models A and B. However, this claim cannot be made if they do not compare against X, Y and Z, which are the current SOTA models in the field.(*and the information is accurate, comparison against X,Y and Z is indeed missing*)

> *The point made in this example is that* ***if both*** *of the mentioned* ***weaknesses are correct****, then we should go with the one that* ***improves the submission the most****.*

- ◯ Review 1
- ◯ Review 4
- ◯ Tie

Figure 8: Comparison guidelines for the "*Overall helpfulness*" criterion.