Usefulness of LLMs as an Author Checklist Assistant for Scientific Papers: Experiment

Abstract

Large language models (LLMs) represent a promising, but controversial, tool in aiding scientific peer review. This study evaluates the usefulness of LLMs in a conference setting as a tool for vetting paper submissions against submission standards. We conduct an experiment where 234 papers were voluntarily submitted to an 201cLLM- based Checklist Assistant.201d This assistant validates whether papers adhere to the author checklist, which includes questions to ensure compliance with research and manuscript preparation standards. Evaluation of the assistant by paper authors suggests that the LLM-based assistant was generally helpful in verifying checklist completion. In post-usage surveys, over 70

1 Introduction

Recent advancements in large language models (LLMs) have significantly enhanced their capabilities in areas such as question answering and text generation. One promising application of LLMs is in aiding the scientific peer-review process. However, the idea of using LLMs in peer review is contentious and fraught with potential issues. LLMs can hallucinate, exhibit biases, and may compromise the fairness of the peer-review process. Despite these potential issues, LLMs may serve as useful analytical tools to scrutinize manuscripts and identify possible weaknesses or inaccuracies that need addressing.

In this study, we take the first steps towards harnessing the power of LLMs in the application of conference peer review. We conduct an experiment at a premier conference in the field of machine learning. While the wider ethical implications and appropriate use cases of LLMs remain unclear and must be a larger community discussion, here, we evaluate a relatively clear-cut and low-risk use case: vetting paper submissions against submission standards, with results shown only to the authors.

Specifically, the peer-review process requires authors to submit a checklist appended to their manuscripts. Such author checklists, utilized in as well as in other peer-review venues, contain a set of questions designed to ensure that authors follow appropriate research and manuscript preparation practices. The Paper Checklist is a series of yes/no questions that help authors check if their work meets reproducibility, transparency, and ethical research standards expected for papers. The checklist is a critical component in maintaining standards of research presented at the conference. Adhering to the guidelines outlined by these checklists helps authors avoid mistakes that could lead to rejection during peer review.

We deploy and evaluate a Checklist Assistant powered by LLMs. This assistant scrutinizes authors2019 responses to the checklist, proposing enhancements for submissions to meet the conference2019s requirements. To prevent any potential bias in the review process, we confine its usage exclusively to the authors of papers, so the checklist assistant is not accessible to reviewers. We then systematically evaluate the benefits and risks of LLMs by conducting a structured study to understand if LLMs can enhance research quality and improve efficiency by helping authors understand if their work meets research standards. Specifically, we administered surveys both before and after use of the Checklist Assistant asking authors about their expectations for and perceptions of the tool. We

.

received 539 responses to the pre-usage survey, 234 submissions the the Checklist Assistant and 78 responses to the post-usage survey. Our main findings are as follows:

(1) Authors generally reported that the LLM-assisted checklist review was a valuable enhancement to the paper submission process.

- The majority of surveyed authors reported a positive experience using the LLM assistant. After using the assistant, over 70
- Authors 2019 expectations of the assistant 2019s effectiveness were even more positive before using it than their assessments after actually using it (Section 4.1.3).
- Among the main issues reported by authors in qualitative feedback, the most frequently cited were inaccuracy (20/52 respondents) and that the LLM was too strict in its requirements (14/52 respon- dents) (Section 4.1.4).

(2) While changes in paper submissions cannot be causally attributed to use of the checklist verification assistant, we find qualitative evidence that the checklist review meaningfully helped some authors to improve their submissions.

- Analysis of the content of LLM feedback to authors indicates that the LLM provided granular feedback to authors, generally giving 4-6 distinct and specific points of feedback per question across the 15 questions (Section 4.2.1).
- Survey responses reflect that some authors made meaningful changes to their submissions 201435 survey respondents described specific modifications they would make to their submissions in response to the Checklist Assistant (Section 4.2.2).
- In 40 instances, authors submitted their paper twice to the checklist verifier (accounting for 80 total paper submissions.) Between these two submissions, authors tended to increase the length of their checklist justifications significantly, suggesting that they may have added content in response to LLM feedback (Section 4.2.3).

Finally, we investigate how LLM-based tools can be easily manipulated 2013 specifically, we find that with AI- assisted re-writing of the justifications, an adversarial author can make the Checklist Assistant significantly more lenient (Section 5.1).

In summary, the majority of authors found LLM assistance to be beneficial, highlighting the significant potential of LLMs to enhance scientific workflows 2014 whether by serving as direct assistants to authors or helping journals and conferences verify guideline compliance. However, our findings also underscore that LLMs cannot fully replace human expertise in these contexts. A notable portion of users encountered inaccuracies, and the models were also vulnerable to adversarial manipulation.

Our code, LLM prompts, and sample papers used for testing are available at: https://github.com/ihsaan-ullah/neurips-checklist-assistant

2 Related Work

In the following section, we provide background on the Author Checklist (Section 2.1) and on the use of LLMs in the scientific peer review process (Section 2.2).

2.1 The Author Checklist

We provide below the checklist questions used in submission template. We provide only the questions here and give the full version including guidelines in Appendix A. These questions are designed by organizers, not specifically for this study, and questions are carried over from previous years. The authors had to provide a response to each question, comprising 201cYes,201d 2018No201d or 201cNA201d (Not Applicable), along with a justification for their answer.

Claims: Do the main claims made in the abstract and introduction accurately reflect the paper2019s contri- butions and scope?

Limitations: Does the paper discuss the limitations of the work performed by the authors?

Theory Assumptions and Proofs: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Experimental Result Reproducibility: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Open access to data and code: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Experimental Setting/Details: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Experiment Statistical Significance: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Experiments Compute Resources: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Code Of Ethics: Does the research conducted in the paper conform, in every respect, with the Code of Ethics

Broader Impacts: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Safeguards: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Licenses for existing assets: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

New Assets: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Crowdsourcing and Research with Human Subjects: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screen- shots, if applicable, as well as details about compensation (if any)?

Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Sub- jects: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

2.2 Related work

Language models have been used in the scientific peer review process for over a decade. The primary application so far has been in assigning reviewers to papers. Here, a language model first computes a 201csimilarity score 201d between every reviewer-paper pair, based on the text of the submitted paper and the text of the reviewer2019s previously published papers. A higher value of the similarity score indicates that the language model considers this reviewer to have a higher expertise for this paper. Given these similarity scores, reviewers are then assigned to papers using an optimization routine that maximizes the similarity scores of the assigned reviewer-paper pairs.

There have been recent works that design or use LLMs to write the entire review of papers. The outcome measures for evaluating the effectiveness of the LLM- generated reviews are based on ratings sourced from authors or other researchers. It is not entirely clear how these ratings translate to meeting the objectives of peer review in practice namely that of identifying errors, choosing better papers, and providing useful feedback to authors. Moreover, it is also known that evaluation of peer reviews themselves are fraught with biases, and the aggregate effect of such biases on these evaluations of reviews is not clear. Our work focuses on a more concrete task in reviewing papers than generating an end-to-end review, namely validating that papers meet criteria specified in an

Author Checklist. Moreover, we evaluate the efficacy of LLMs in the setting of an actual peer review conference.

Recent work also investigates whether LLMs can identify errors in papers and shows promising initial results. The paper constructs a set of short papers with deliberately inserted errors and asks LLMs to identify errors. GPT-4 does identify the error more than half the time. Another experiment described asks GPT-4 to identify deliberately inserted errors in three full papers. It successfully and consis- tently does so on one paper, partially and occasionally on a second paper, and is consistently unsuccessful on the third. Note that in both experiments, the prompts specifically asked the LLM to find errors rather than generically asking the LLM to review the paper. Moreover, both experiments had small sample sizes in terms of the number of papers. In another set of experiments presented, evaluated the ability of large language models (LLMs) to compare the 201cstrength201d of results between papers, mirroring the goals of conferences and journals in selecting 2018better 2019 papers. The experiment consisted of creating 10 pairs of abstracts, where one abstract in each pair was made 2018 clearly 2019 and objectively stronger than the other. To simulate diverse, yet irrelevant conditions, the language of the abstracts was deliberately varied. In this test, GPT-4 performed no better than random chance in identifying the stronger abstract, underscoring that while LLMs may excel at some complex tasks like scientific error identification, they often struggle with seemingly simpler tasks.

The papers investigate the performance of LLMs in evaluating checklist compliance. These studies, however, were retrospective studies of published papers, whereas our work is deployed live associated to a peer-review venue and helps authors improve their checklist compliance before they make their submission.

Recent work has highlighted the prevalence of the use of LLMs both in preparation of scientific paper manuscripts and in the generation of scientific peer reviews. For example, estimates that as of January 2024, 17.5

3 Methodology

We design an LLM-based tool (Checklist Assistant) to assist authors in ensuring their submitted checklists are thoroughly answered. Our platform interfaced with a third-party LLM (GPT-4 from OpenAI), using simple prompt engineering with these hyper-parameters: temperature = 1, topp = 1, and n = 1. For each checklist question, the LLM is provided with the author2019s checklist response and justification, alongside the complete paper and any appendices. The LLM2019s role is to assess the accuracy and thoroughness of each response and justification, offering targeted suggestions for improvement. Each checklist item is treated as an individual task, i.e., an API call with only one question, its answer and justification by the author, and the paper and appendices. The API call returns a review and score for the submitted question.

Figure 1 illustrates examples of feedback provided by the Checklist Assistant for two different papers. In these examples, green indicates that the tool found 201cno significant concerns201d, while orange signals 201cneeds improvement201d with the Paper Checklist standards. Authors are encouraged to carefully review any orange feedback, validate the identified issues, and make the necessary revisions to align with the checklist requirements.

3.1 Deployment

We deployed the Checklist Assistant on Codabench.org. We configured 15 Google Cloud CPU workers, integrated with Codabench, to handle multiple paper submissions concurrently. The bulk of the computations were carried out by the LLM third-party software (GPT-4 from OpenAI) via API calls (one call per question, and additional calls in case of failure).

Participation was fully voluntary, and participants were recruited through a blog post that was released 8 days before the abstract submission deadline. Interested participants were asked to register though a Google form. Participants who submitted registration requests through the Google form were then given access to the Assistant on the Codabench platform. The submissions were entirely optional and completely separate from the paper submission system and the review process. The papers had to be formatted as specified in the call for papers (complete with appendices and checklist). Information provided in external links was not taken into account by the assistant. We asked submitters to fill out

the checklist to the best of their abilities. Submissions made via the Codabench landing page were processed as follows:

Checklist Assistant: The paper was parsed using a PDF-to-text parser, then screened for any problems such as the format of the paper or checklist, etc. Each answered question in the checklist was processed by an LLM using an API.

Result Compilation: LLM responses were combined for all questions and formatted in an HTML document with proper colors and structure for readability and user-friendliness.

We encountered several parsing issues with both paper texts and checklists. Initially, our parser struggled with subsections and titles, prompting code improvements to handle sections accurately. Checklist parsing also faced issues due to spacing and incomplete checklists, which we addressed by refining the code. Special characters, especially merged letters like 201cfi 201d and 201cfl 201d in the submitted PDFs required further parsing updates.

3.2 Prompt engineering

In this section we discuss design of a prompt given to the LLM, tasked to behave as Checklist Assistant. We provide the full prompt in Appendix B.

While preparing the Checklist Assistant, we experimented with various prompt styles. Tuning was carried out using a dozen papers. Some checklists were filled out with our best effort to be correct, and others included deliberately planted errors to verify robustness and calibrate the scores. We observed that the LLM performed better with clear, step-by-step instructions.

Our final prompt provided a sequence of instructions covering different aspects of the required review, designed as follows: first, the context is set by indicating that the paper is under review for the conference. Next, the main goal is clarified, specifying that the LLM2019s primary task is to assist the author in responding to the checklist question. The LLM is then directed to review the author2019s answer and justification, identifying any discrepancies with the paper based on the specific guidelines of the question. It is instructed to provide itemized, actionable feedback according to the guidelines, offering suggestions for improvement, with clear examples for responses such as 201cYes, 201d 201cNo, 201d or 201cNA. 201d At the end of the review, the LLM is asked to assign a score: Score=1 for no issues, Score=0.5 for minor improvements, and Score=0 for critical issues. Finally, the LLM is provided with the checklist question, the author 2019s answer, justification, the relevant guidelines, and the paper content.

Before prompt adjustments, LLM responses often mixed the review with the score. To fix this, we specified that the score should be returned on a separate line at the end of the review. For long papers exceeding 35 pages (or 15,000 words), we processed only the first 15,000 words and notified authors with a warning.

We hypothesized that users might find the LLM responses overly strict, vague, and lengthy (which was indeed later confirmed), so we added prompt instructions like 201cuse 0 score sparingly 201d, 201cprovide itemized, actionable feedback 201d, and 201cfocus on significant improvements. 201d Although the Checklist Assistant returned scores of 0, 0.5, and 1, we combined the 0 and 0.5 scores to indicate that improvement was needed, rather than differentiating between two levels of severity (with red for 0 and orange for 0.5). This decision was made due to concerns that the LLM 2019s evaluations might be too harsh. User feedback on LLM strictness and other issues is analyzed in Section 4.

We also tested whether the LLM was consistent in generating answers for reiterations of the same input. As a sanity check, we test for each question, whether the variation of the output scores for multiple runs on the same paper is comparable to the variation across papers. We find that the variation in scores for multiple runs on the same paper is significantly lower than variation across papers ($p < 0.05$; based on a one sided permutation test after BH correction) for all but one question. The only question that had a comparable variance within and across papers was the question on ethics (Q9; $p > 0.4$).

3.3 Anonymity, confidentiality, and consent

The authors could retain their anonymity by registering to Codabench with an email that did not reveal their identity, and by submitting anonymized papers. The papers and LLM outputs were kept confidential and were not be accessible to reviewers, meta reviewers, and program chairs. It is important to note that while authors retained ownership of their submissions, the papers were sent to the API of an LLM service, and treated under their conditions of confidentiality.

This study was approved by the Carnegie Mellon University Institutional Review Board (IRB). The participants gave written documentation of informed consent to participate.

4 Experiments

In our evaluations, we seek to address two main questions regarding the use of an LLM-automated Author Checklist Assistant:

(1) Do authors perceive an LLM Author Checklist Assistant as a valuable enhancement to the paper sub- mission process?

(2) Does the use of an Author Checklist Assistant meaningfully help authors to improve their paper sub- missions?

In order to understand author experience using the provided Author Checklist Assistant, we surveyed authors before and after submitting to the Author Checklist Assistant. Additionally, we analyzed the content and submission patterns of author 2019s checklists and the LLM responses. A summary of our main findings is given in Section 1. In this subsequent section we provide detailed analyses of survey responses and usage of the Checklist Assistant. In Section 4.1, we give results on author perception and experience and in Section 4.2 we analyze changes made by authors to their submissions after using the Author Checklist Assistant.

4.1 Author Perception and Experience

First, we analyze the authors 2019 usage patterns and perceptions of the Author Checklist Assistant, as captured through surveys. In Section 4.1.1, we provide an overview of how authors filled out the checklist and the responses given by the LLM on their checklists. In Section 4.1.2, we detail the survey methodology used to understand author experience and in Section 4.1.3, we analyze results of the survey. Finally, in Section 4.1.4, we overview the main challenges identified by authors when using the Author Checklist Assistant.

4.1.1 Overview of Checklist Usage and Responses

A total of 234 papers, each accompanied by a checklist, were submitted to the assistant. For each checklist question, authors could respond with Yes, No, NA, or TODO. As illustrated in Figure 2a, most questions received a Yes response, indicating that the authors confirmed their paper met the corresponding checklist criteria. However, for the questions on Theory, Impacts, Safeguards, Documentation, Human Subjects, and Risks, a significant portion of authors selected NA. Additionally, a notable number of authors responded No to the questions on Code and Data, and Error Bars.

In response to the authors 2019 checklists, the LLM provided written feedback, with green indicating 2018No Concerns 2019 and orange indicating 2018Needs improvement 2019. Figure 2b illustrates the distribution of LLM feedback for each checklist question. For most questions, the majority of feedback suggested that the checklist or manuscript could be improved. However, for the questions on Theory, Human Subjects, and Risks, many NA responses were deemed appropriate, leading the LLM to respond with 2019No Concerns. 2019 This likely reflects the LLM 2019s confidence in confirming that certain papers did not include theory, human subjects research, or clear broader risks, making those checklist items irrelevant. In Figure 3, we show the distribution of LLM evaluations per submission. All submissions received several 2018Needs improvement 2019 ratings, with each being advised to improve on 8 to 13 out of the 15 checklist questions.

4.1.2 Survey Methodology

To assess authors 2019 perceptions of the usefulness of the Author Checklist Assistant, we conducted a survey with all participants both at registration (pre-usage) and immediately after using the Author Checklist Assistant (post-usage). We provide the content of the surveys in Figure 4. Both surveys contained the same four questions, with the pre-usage survey focusing on expectations and the post-usage survey on actual experience. Responses were recorded on a four-point Likert scale, ranging from strongly disagree to strongly agree. In the post-usage survey, we also asked authors to provide freeform feedback on (1) any changes they planned to make to their paper, and (2) any issues they encountered while using the Checklist Assistant.

We received 539 responses to the pre-usage survey and 234 papers submitted. However, we received only 78 responses to the post-usage survey, representing 63 unique participants (due to multiple submissions for the same paper). While completing the pre-registration survey was mandatory for all participants, the post-usage survey was optional. As a result, all participants in the post-usage survey had also completed the pre-registration survey.

4.1.3 Survey Responses

Figure 5 presents the survey responses collected before and after using the checklist verification tool. We include responses from authors who completed both surveys (n=63). In cases where authors submitted the survey multiple times for the same paper, we included only the earliest post-usage response. Including the duplicated responses made a negligible difference, with the proportion of positive responses changing by less than 0.02 across all questions.

Overall, the majority of authors responded positively regarding their experience with the Checklist Assis- tant. 70

It is notable that authors were even more positive before using the tool. Comparing pre- and post-usage responses, there was a statistically significant drop in positive feedback on the 201cUseful 201d and 201cExcited to Use 201d questions 2014we run a permutation test with 50,0000 permutations to test whether the difference between proportion of positive responses pre and post-usage is non-zero, which gives Benjamini-Hochberg adjusted p-values of 0.007 and 0.013 for 201cExcited to Use 201d and 201cUseful 201d respectively with effect sizes of 22120.23 and 22120.2.

We also assessed the correlation between post-usage survey responses and the number of 2018needs improve- ment 2019 scores given by the LLM to authors. In Figure 6, we show mean number of needs improvement scores for authors responding positively or negatively to each survey question. We find no substantial ef- fect of number of 2018needs improvement 2019 scores on survey responses. This may reflect that the number of 2018needs improvement 2019 scores was less important in author 2019s perception than the written content of the LLM 2019s evaluation.

Finally, we examined potential selection bias due to the drop-off in participation in the post-usage survey by analyzing the pre-usage survey responses across different groups. As noted earlier, only a portion of the 539 participants who completed the pre-usage survey went on to submit papers (234 Submitters), and an even smaller group responded to the post-usage survey (78 Post-Usage Respondents). In Figure 7, we compare the pre-usage survey responses between Submitters and Non-Submitters, as well as between Post- Usage Respondents and Non-Respondents. No substantial differences in rates of positive responses were found (using a permutation test for the difference in mean response, gave p-values of > 0.3 for all questions before multiple testing correction), suggesting there is no significant selection bias.

4.1.4 Challenges in Usage

In addition to the structured survey responses, 52 out of the 78 post-usage survey submissions included freeform feedback detailing issues with the Checklist Assistant 2019s usage. We manually categorized the reported issues from these responses and identified the following primary concerns, listed in order of decreasing frequency (summarized in Figure 8):

Inaccurate: 20 authors reported that the LLM was inaccurate. Note that it is not possible to tell from the responses how many inaccuracies participants found in individual questions since the survey did not ask about individual checklist questions. Many participants noted specific issues, in particular that the LLM overlooked content in the paper, requesting changes to either the checklist or the paper

for elements that the authors believed were already addressed. Additionally, some authors reported more nuanced accuracy issues. For instance, one author mentioned that the LLM misinterpreted a 201cthought experiment 201d as a real experiment and incorrectly asked for more details about the experimental setup. Another author reported that the LLM mistakenly assumed human subjects were involved due to a discussion of 201cinterpretability 201d in the paper.

Too strict: 14 authors reported that the LLM was too strict.

Infeasible to make changes due to page limits: 5 authors felt that they received useful feedback, but it would not be possible to incorporate due to their papers already being at the page limit.

Too generic: 4 authors reported that the feedback they received was not specific enough to their paper.

Insufficient LLM capabilities: 4 authors complained that the LLM could not handle content over the (LLM assistant 2019s) page limit or that it was not multimodal and hence ignored figures.

Feedback inconsistent across submissions: 3 authors reported that the LLM feedback changed across multiple submissions to the server even though the paper and checklist content did not change.

Desire for full paper review: 3 authors reported that they would like feedback on the entire paper, not just on checklist items.

Bad at theory (mathematical) papers: 2 authors wrote that the LLM seemed bad at theory (mathematical) papers.

Too verbose: 2 authors wrote that the LLM 2019s feedback was too wordy.

4.2 Changes to Submissions in Response to Feedback

In the following analysis, we integrate an assessment of the LLM 2019s feedback with the authors 2019 checklist answers, to better understand whether the Checklist Assistant helped authors make concrete and meaningful changes to their papers. In Section 4.2.1, we analyze the types of feedback given by the LLM to authors. In Section 4.2.2, we overview the changes to their papers that authors self-reported making in survey responses. Lastly, in Section 4.2.3, we analyze changes made in multiple submissions of the same paper to the Author Checklist Assistant.

4.2.1 Characterization of LLM Feedback by Question

For authors to make meaningful changes to their papers, the Author Checklist Assistant must provide concrete feedback. In this section, we analyze the type of feedback given by the Checklist Assistant to determine whether it is specific to the checklist answers or more generic.

Given the large volume of feedback, we employed an LLM to extract key points from the Checklist Assistant 2019s responses for each question on the paper checklist and to cluster these points into overarching categories. Specifically, for each of the 15 questions across the 234 checklist submissions, we used GPT-4 to identify the main points of feedback provided to authors. We manually inspected that the main points extracted by GPT-4 matched the long-form feedback on 10 randomly selected submitted paper checklists and found that GPT-4 was highly accurate in extracting these key feedback points. We then passed the names and descriptions of these feedback points to GPT-4 to hierarchically cluster them into broader themes.

The most frequently identified feedback themes for 4 questions are shown in Figure 9. Here are our key observations from this analysis.

The LLM identified many granular types of feedback within each checklist question. We illustrate with examples of responses to four questions in Figure 9. For instance, the LLM gave granular feedback within the Experimental settings/details question on optimizer configuration details, implementation code availability, and explicit mention of non-traditional experiments.

The LLM tended to provide 4-6 distinct points of feedback per question (for each of the 15 questions).

The LLM is capable of giving concrete and specific feedback for many questions. For example, on the 201cClaims 201d question, the LLM commented on consistency and precision in documenting claims on 50 papers, including feedback like matching the abstract and introduction and referencing appendices. On the 201cCompute resources 201d question the LLM commented specifically on detailing compute / execution time of methods.

The LLM tends to provide some generic boilerplate for each question. The most common category of feedback for each question is a generic commentary on enhancing general aspects of the question.

There are certain topics that appear across many questions, in particular discussion of limitations and improved documentation.

The LLM often expands the scope of checklist questions. For example, the LLM brings up reproducibility as a concern in feedback to the code of ethics question and brings up anonymity quite frequently in the code and data accessibility question.

We provide a full list of the summarized main themes of feedback in Appendix C. In summary, our analysis of the feedback given by the LLM suggests that the LLM gave concrete and actionable feedback to authors that they could potentially use to modify their paper submissions. Our analysis also suggests that a more detailed checklist could be developed to provide more granular feedback, based on the rubrics covered by the Author Checklist Assistant. Such a detailed checklist could be processed automatically by an LLM to systematically identify specific, commonly overlooked issues in scientific papers and flag concrete issues for authors to resolve.

4.2.2 Authors2019 Descriptions of Submission Changes

We obtain additional evidence of changes made by authors in response to the Checklist Assistant through the post-usage survey. In the survey, we asked authors to detail in freeform feedback any changes they had made or planned to make in responses to feedback from the LLM. Of the 78 survey responses, 45 provided feedback to this question. Of these 45 responses, 35 actually described changes they would make (the remainder used this freeform feedback to describe issues that they had in using the assistant). Based on manual coding of the comments, we identified the main themes in changes they planned to make:

14 authors said that they would improve justifications for their checklist answers by including more detail and/or references to paper sections.

6 authors said that they would add more details about experiments, datasets, or compute.

2 authors said they would change an answer to the checklist that they filled out incorrectly.

2 or fewer authors mentioned improving the intro/abstract, discussion of limitations, and discussion of standard errors.

Overall, these responses indicate that some authors were motivated to modify their submissions due to feedback from the checklist verification.

4.2.3 Analysis of Re-submissions

Finally, we analyze changes made between submissions to the Checklist Assistant when authors submitted multiple times. There were 40 instances where an author submitted the same paper to the checklist verification multiple times (out of 184 total distinct paper submissions to the checklist verification). In this analysis, we assess changes made to the paper checklist between the first and second submission to our checklist verifier in order to understand whether authors made substantive changes to their checklists and/or paper manuscripts in response to feedback from the checklist verification.