AblationBench: Evaluating Automated Planning of Ablations in Empirical AI Research

Talor Abramovich
Tel Aviv University
talora1@mail.tau.ac.il

Gal Chechik
Bar Ilan University
NVIDIA

Abstract

Language model agents are increasingly used to automate scientific research, yet evaluating their scientific contributions remains a challenge. A key mechanism to obtain such insights is through ablation experiments. To this end, we introduce *AblationBench*, a benchmark suite for evaluating agents on ablation planning tasks in empirical AI research. It includes two tasks: *AuthorAblation*, which helps authors propose ablation experiments based on a method section and contains 83 instances, and *ReviewerAblation*, which helps reviewers find missing ablations in a full paper and contains 350 instances. For both tasks, we develop LM-based judges that serve as an automatic evaluation framework. Our experiments with frontier LMs show that these tasks remain challenging, with the best-performing LM system identifying only 45% of the original ablations on average, below human-level performance. We observe an inverse performance trend between the author and reviewer tasks, which we attribute to differences in model grounding. Lastly, we analyze the limitations of current LMs on these tasks, and find that chain-of-thought prompting outperforms an agent-based approach.[1]

1 Introduction

Automating scientific research using language model agents is a fast-growing field. Several groups have shown LM agents, often referred to as *AI co-scientists*, that automate various steps of the scientific process from hypothesis generation (Lu et al., 2024; Si et al., 2024; Gottweis et al., 2025) to experiment execution (Bogin et al., 2024; Tian et al., 2024a; Xiao et al., 2025; Nathani et al., 2025; Starace et al., 2025). Evaluating the efficacy of these agents in generating insightful scientific contributions poses a significant challenge. A key mechanism to obtain such insights is by dissecting a proposed method into its components. The contribution of these components is evaluated by *ablation experiments* (Meyes et al., 2019; Lipton and Steinhardt, 2019).

One can separate the process of ablation experiments into two steps: *planning* these experiments, and *executing* them by writing the code and collecting results. Here, we identify planning as a key step in AI co-scientists (Wei et al., 2022; Wang et al., 2023; Yao et al., 2023a; Huang et al., 2024b). Other studies developed approaches for research code generation, e.g. (Tian et al., 2024a; Xiao et al., 2025; Starace et al., 2025), and this is not the focus of this paper. Planning ablations is challenging because it requires identifying key components of a method, and then proposing meaningful changes that are relevant, feasible, and grounded in domain knowledge. An important first step toward automatic ablation planning was introduced in AbGen (Zhao et al., 2025). AbGen generates ablation plans from manually processed paper content that summarizes the research background, methodology, main experiment and results. An LM is then instructed to propose ablations for a specific component of the method. These ablations are evaluated mainly by human experts who rank their quality. This leaves fully automated ablation planning from raw papers unexplored.

Here, we look into two tasks in automating ablations: (1) Assisting ***authors***, proposing ablation experiments given a written method section; and (2) Assisting ***reviewers***: proposing missing ablation experiments of a given paper. We construct benchmark datasets for developing ablation agents for these two tasks. The first, *AuthorAblation* contains 83 papers, spans across 14 conferences and includes 230 human-annotated ablation experiments from the original papers. The second, *ReviewerAblation* is based on 350 ICLR submissions and their official reviews from 2023-2025 that mention ablations. Also, for each of the two tasks we develop an LM-based judge - an evaluation agent using ground

[1]Data: HuggingFace. Code: GitHub.

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 30612–30651
October 24-29, 2026 ©2026 Association for Computational Linguistics

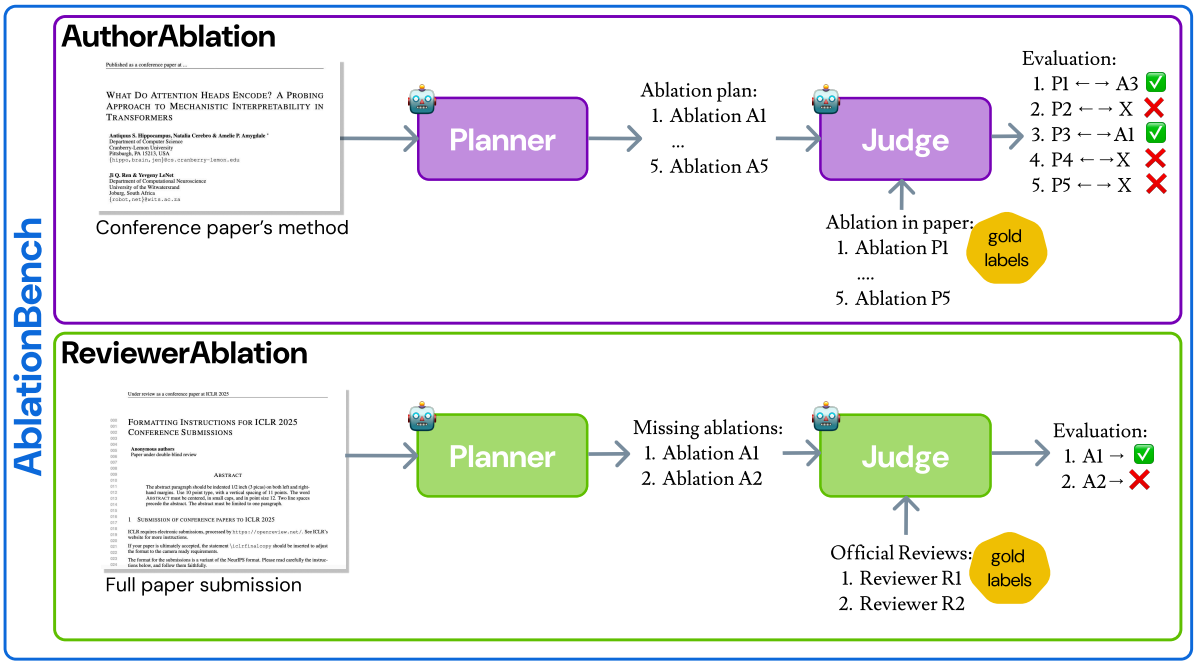


Figure 1: Overview of AblationBench. (**Top**) **AuthorAblation** is a collection of conference papers where the task of the planner is to generate ablation plan based on the paper's method section. The judge evaluates the generated plan using manually extracted gold ablations from the paper. (**Bottom**) **ReviewerAblation** is a collection of ICLR submissions where the task of the planner is to generate missing ablation based on the full submission. The judge evaluates the generated ablations using the official reviews of the submission.

truth data, that can evaluate any author-assistant or reviewer-assistant applied to our benchmarks. Together they provide a complete evaluation resource for developing ablation agents.

We establish baseline levels of performance for our two ablation benchmark tasks (author and reviewer) using two "planners" (LM and Agent). These planners are LM systems that generate ablation plans. The best planner identifies at most 45% of the original ablations across tasks. On a subset of ten AuthorAblation tasks, humans achieve an F1 score of 0.65, compared to 0.42 for the best-performing model. These results highlight the benchmark's difficulty.

In summary this paper makes the following contributions:
(1) *AblationBench*: A benchmark suite for evaluating LMs in ablation planning, containing two tasks:

- *AuthorAblation*: A task of generating an ablation plan based on a paper's method section, consisting of 83 research papers with 230 human-annotated ablations.
- *ReviewerAblation*: A task of identifying missing ablations in paper submissions, consisting of 350 ICLR submissions and their official reviews.

(2) Two LM-based judges for evaluating generated plans against ground truth for each task, together with a quantitative evaluation of the accuracy of the two judges using a held-out evaluation dataset with human annotations.
(3) Two baseline LM planners for the benchmark to assess the capabilities of current frontier LMs on ablation planning.

2 Related Work

AI co-scientists. Several groups have proposed automated pipelines for the scientific research cycle using AI agents (Ifargan et al., 2024; Lu et al., 2024; Bubeck et al., 2025; Tang et al., 2025). Gottweis et al. (2025) introduce a multi-agent system for hypothesis generation in biomedical research that assists human scientists, while Audran-Reiss et al. (2025) analyze the strengths and failure modes of such systems. Our work focuses on isolating and evaluating a single core scientific capability: ablation experiment design.

Co-scientists benchmarks. Several benchmarks evaluate AI systems across stages of scientific re-

	AblationBench	AbGen (Zhao et al., 2025)
Task	Author and reviewer	Author only
Input	Raw paper text	Manually processed research context from a single paper
Output	Full ablation plan	Specified component ablation
Evaluation	LM-based judges	Human
Criteria	Compared to ground truth	Subjective ablation quality
Domain	Empirical AI (ML, CV, NLP)	NLP
# Papers	433	807

Table 1: AblationBench and AbGen comparison.

search. Song et al. (2025) evaluates models on scientific discovery projects in various domains. CiteME (Press et al., 2024), LitSearch (Ajith et al., 2024), and PaperQA2 (Skarlinski et al., 2024) focus on citation accuracy for claims. Si et al. (2024) evaluate AI research idea generation. Many benchmarks focus on experiment replication and code generation from papers (Bogin et al., 2024; Huang et al., 2024a; Tian et al., 2024a; Xiao et al., 2025; Nathani et al., 2025; Starace et al., 2025; Siegel et al., 2024; Seo et al., 2025), or solving scientific tasks like Kaggle challenges (Chan et al., 2025; Wijk et al., 2024; Gu et al., 2024). We build our benchmark on top of prior work on code replication, with a focus on ablation studies in empirical AI research. AbGen (Zhao et al., 2025) is a recent benchmark focused on ablation planning, where an LM is instructed to design ablation plan given a manually processed research context, method and main experiment from a single paper. AblationBench extends this line of work in several important ways. Most notably, it establishes a fully automated pipeline for ablation planning and evaluation, where models must infer full ablation plans directly from raw paper content, rather than from preprocessed or structured inputs. Additionally, AblationBench formulates ablation planning from two complementary perspectives: authors and reviewers, covering multiple conferences and topics in AI research. Full comparison appears in Table 1.

LM agents for scientific research. LM agents for scientific tasks often build on ReAct (Yao et al., 2023b), where the model iteratively generates thoughts, performs actions, and receives observations in a containerized environment. Examples include MLAgentBench (Huang et al., 2024a), AutoKaggle (Li et al., 2024), and AgentK (Grosnit et al., 2024). More recently, MLGym (Nathani et al., 2025) introduced a framework based on SWE-agent (Yang et al., 2024b), an agent built to automate software engineering using specialized agent-computer interfaces. MLGym is tailored for AI research tasks with built-in agent-computer interfaces for that matter.

LMs as evaluation judges. Evaluating model-generated output is inherently ambiguous—different plausible outputs may vary in structure or wording, even if they capture the same underlying idea. To address this, prior work has researched LMs-as-judges (Liu et al., 2023; Zheng et al., 2023b; Fu et al., 2023; Chan et al., 2023; Zhuge et al., 2024), often with chain-of-thought (CoT) prompting (Wei et al., 2023), showing performance on par with or better than human evaluations (Chiang and Lee, 2023; Nasution and Onan, 2024; Ahmed et al., 2025). Despite their strong performance, LM-based judges can exhibit systematic biases, including positional, intra-model, and contextual biases (Wang et al., 2024; Verga et al., 2024; Xu et al., 2024; Zhou et al., 2024). Prior work has proposed mitigation strategies such as using diverse judge models or relying on pairwise comparisons instead of direct ranking. We follow this line of work to design our automatic evaluation system.

3 Overview of AblationBench

AblationBench is designed for two tasks: First, for generating ablation plans based on a paper's method section, we built a component called *AuthorAblation*. Second, to identify missing ablations in a full paper, similar to what a reviewer would do, we built *ReviewerAblation*.

For each task, we start by collecting a set of papers for the task setup. Next, we design *LM planners* to generate ablation plans for each of these two tasks. Finally, we develop *LM judges* to evaluate the quality of the generated plans by comparing them to gold labels. The judges are used to automate the evaluation of our benchmark, inspired by the judge used in PaperBench (Starace et al., 2025). In both tasks, the judges evaluate against a ground-truth (GT): the ablations originally conducted by the authors in AuthorAblation, and the ablations suggested by reviewers in ReviewerAblation. While models may propose new ablation ideas, evaluating such suggestions is challenging, as no GT exists for this setting, and the suggestions must be assessed for both relevance and feasibility, so we leave this as future work. With that said, we argue in the next sections that the ablations proposed by the authors and reviewers provide a meaningful evaluation target, and we stress that current

models still struggle with finding those ablations.

To evaluate how well the judges perform, we also construct a dedicated dataset, *JudgeEval*, consisting of *AuthorEval* and *ReviewerEval*, with manually labeled examples.

We report the overall AblationBench score as the average performance across the two tasks, for a complete view of both the author and reviewer perspectives in ablation planning. Figure 1 shows an overview of AblationBench. We now describe each component in detail.

4 AuthorAblation: Ablation Benchmark for Authors

The first AblationBench component is designed for the task of proposing ablation experiments given a paper including its method section but without the experiments sections. We collect 83 papers and annotate each with the ablations that appear in the full paper, serving as gold labels.

We develop two baseline planners for this task. For a given instance they each generate k proposals for ablation experiments. We also design an automatic judge to evaluate planners. To assess the quality of the judge itself, we created a separate dataset of 63 plans that is human annotated, so we can compare the decisions of the judge with human evaluation of the ablation proposals.

In the following, we describe the construction, planner, and judge for AuthorAblation component.

4.1 Constructing AuthorAblation

AuthorAblation is built from 83 machine learning conference papers that include ablation studies and open-source code. We collect these papers from prior benchmarks—CSR-Bench (100 papers) (Xiao et al., 2025), SUPER-Expert (45 papers) (Bogin et al., 2024), and PaperBench (20 papers) (Starace et al., 2025), and supplement them with 38 best paper awardees from CVPR and ICCV (2020–2024) to increase topic and conference diversity.

We filter papers by checking for the presence of ablation sections, then truncate each to include only content up to the method section. The ablation plan from the full paper is extracted into a structured JSON format to serve as GT. We extract ablations in the order they appear in the paper, using this order as a proxy for their relative importance. Papers are split into 21 for development and 62 for testing. See Appendix A.1 for full construction details.

Since ablation planning is open-ended, there may be many valid and useful ablations that do not appear in the original paper. We further validate the usefulness of author ablations on the development set. We find that among 105 ablation generations of the best model (GPT 5.4), 58 matched the GT (55%), while only 7 ablations (7%) were valuable ablations beyond the GT.

4.2 AuthorAblation Planner

The planner in AuthorAblation is tasked with generating an ablation plan given a paper's title, abstract, and truncated content up to the method section. The output is a structured JSONL file containing up to k ablation entries. Each entry should describe the removal or modification of a component of the proposed method, aiming to assess its contribution.

We implement two planner variants as baselines. The first, LM-PLANNER, is using a single CoT style prompt to an LM for generating ablation plan. The LM receives the title, abstract, and aggregated TEX source of the paper, and outputs k ablations along with its reasoning.

The second, AGENT-PLANNER, is an agent-based planner built on SWE-agent (Yang et al., 2024b), a ReAct-style LM agent. This planner operates in a containerized environment where it has access to the paper files and basic tools for file inspection and command execution. The agent is prompted similarly but can take intermediate steps, perform actions on the environment, and refine its output before submitting the final ablation plan.

4.3 AuthorAblation Judge

We evaluate the quality of ablation plans by checking whether each GT ablation is correctly predicted by the model. For each GT ablation, we assess whether there exists at least one matching ablation in the generated plan. A match means that the proposed ablation captures the same idea as the GT one. This includes targeting the same part of the method and applying a similar type of change. Figure 4 in Appendix A.1 provides examples of the matching criteria.

We report three metrics for this binary classification task. First, *precision@*k measures how many of the top k proposed ablations appear in the GT. Second, *recall@*k measures how many GT ablations are correctly predicted. Lastly, we report the *F1-score@*k which balances between the two.

To support automatic evaluation, we design an LM-based judge that compares the GT ablations to the generated ones and decides whether each has a

match. We implement two scaffoldings for the LM-based judge: (1) LMJUDGE, a CoT-style prompt to an LM; and (2) AGENTJUDGE, a ReAct-based agent allowing multiple steps of reasoning and interaction with an environment before generating the final decision. We address three sources of potential bias. First, *intra-model bias*, where a model may favor its own outputs, is addressed by using an ensemble of three models and taking a majority vote (Verga et al., 2024). Second, *contextual bias*, where a model may prefer the GT over the plan due to contextual cues, is mitigated by removing GT and plan tags and randomly assigning them as Side A and Side B, with the model matching in both directions (Zhou et al., 2024). Lastly, *positional bias*, where a model may favor ablations based on their order, is mitigated by randomly shuffling ablations on both sides, with each model observing a different permutation (Wang et al., 2024).

To evaluate judges we create a dataset, *AuthorEval*, of ablation plans for papers in the development set with correct match labels.

5 ReviewerAblation: Ablation Benchmark for Reviewers

The second AblationBench component focuses on suggesting missing ablations for a full paper submission. The goal is to identify important ablations that were not reported but should have been. We collect 350 ICLR submissions along with their official reviews, serving as the gold labels.

We develop two baseline planners for this task. Given a paper submission they generate k proposals for missing ablations. We also design an automatic judge to evaluate planners. To assess the quality of the judge itself, we created a separate dataset of 60 plans that is human annotated, so we can compare the decisions of the judge with human evaluation of the ablation proposals. In the following, we describe the benchmark, planner, and judge for ReviewerAblation.

5.1 Constructing ReviewerAblation

ReviewerAblation is built from ICLR submissions from 2023 to 2025, where reviewers suggested missing ablations. This setting enables the development of reviewer-assistive models that can detect gaps in a paper's empirical validation.

We apply a three-stage pipeline: (1) collect ∼89,100 reviews from ∼22,800 OpenReview (OpenReview, 2025) submission forums and filter for reviews mentioning "ablat,"; (2) use a CoT-prompted LM to identify reviews with concrete suggestions for new ablations; and (3) retrieve the submitted PDFs and convert them to markdown for standardized model input. This automatic pipeline makes it easier to extend the benchmark to other conferences and years. This process results in 5,960 papers in ReviewerAblation. We randomly sample 50 papers for development and 300 for testing. Full construction details are provided in Appendix A.2.

Since reviewer suggestions may be inconsistent (Cortes and Lawrence, 2021; Beygelzimer et al., 2021), we further validate the usefulness of suggested ablations on the development set. We find that for 41 out of 50 papers (82%), the authors accepted the suggested ablations as valid and incorporated them either in the rebuttal or in later revisions of the paper.

5.2 ReviewerAblation Planner

The ReviewerAblation planner is given a paper's title, abstract, and full preprint, and is tasked with generating k missing ablations—experiments that were not reported in the paper but should have been included. This simulates a reviewer's perspective, where the goal is to identify important omissions in the empirical evaluation.

We implement two planner variants as baselines for this task, both follow the same setup described in Section 4.2: the LM planner uses a single CoT-style prompt to generate up to k missing ablations, while the agent planner uses a ReAct-style agent to iteratively construct its suggestions.

5.3 ReviewerAblation Judge

We evaluate the quality of generated missing ablations by checking whether they match the suggested ablations by reviewers. For each generated ablation, we assess whether there is a matching ablation in one of the reviewer's comments. A match means that the generated ablation captures the same underlying idea as the reviewer's suggestion. The ablated component must align in both, but specific actions (e.g., remove, replace) or precise modification details are only required when they are explicitly stated in the review.

We report three metrics for this binary classification task: *precision@*k, which measures how many of the top k generated ablations match the GT; *recall@*k, which measures how many of the

GT reviewer suggestions are correctly predicted; and the *F1-score@k*, which balances the two.

To enable automatic evaluation, we use two LM-based judge architectures similar to the AuthorAblation Judge introduced in Section 4.3. Each judge compares a generated ablation to the reviews' suggestions and decides whether it has a match. We also address potential biases in a similar way: we use an ensemble of three judges to avoid intra-model bias and we randomly shuffle the order of ablations and reviews to reduce positional bias, so that each judge sees a different permutation.

To evaluate judges we construct a dataset, *ReviewerEval*, of ablation plans for papers in the development set, human annotated with match labels.

6 AblationBench Properties

AblationBench, composed of AuthorAblation and ReviewerAblation, provides a diverse evaluation suite for ablation planning in empirical AI research. AuthorAblation contains 230 ground-truth ablations (4 per paper on average), including both modification (59%) and removal (41%) ablations, with modification ablations often admitting multiple valid alternatives. ReviewerAblation contains on average 2 reviewer-suggested ablations per paper, with nearly half of all ICLR 2023–2025 submissions including at least one review mentioning ablations, highlighting their central role in the review process. The benchmark spans 14 conferences and a broad range of AI topics, including general machine learning, NLP, computer vision, representation learning, generative models, and AI applications, enabling evaluation across diverse research domains. See Appendix B for full details.

7 Experiments and Results

We first describe the experimental setup for the judges and the planners in AuthorAblation and ReviewerAblation. We then present the results of the judges and analyze their results. Finally, we report and analyze the main results of the baseline planners across both tasks, together with a human evaluation study performed on AuthorAblation. Additional experiments and results are presented in Appendix C, and full prompts are provided in Appendix D.

7.1 Experimental Setup: Judges

Agent judges. For the agent judges, we use SWE-agent (Yang et al., 2024b) with several frontier LMs: GPT-5.4 (Singh et al., 2025), o3-mini (OpenAI, 2025) (high reasoning), GPT OSS 120B (OpenAI et al., 2025), MiniMax M2.5 (MiniMax, 2026) and Qwen 3.5 122B (Team, 2026). The agents use tools only for file creation and editing, and are also provided a tool to generate the submission file in the required format, with a submit command to verify evaluation formatting. Memory is managed using SWE-agent's default context handling, which keeps the last five observations and collapses earlier ones. Each agent is supplied with a demonstration trajectory illustrating how to perform matching and use the interface, following in-context learning (Brown et al., 2020). Prompts are configured per benchmark's task, based on the matching criteria in Sections 4.3 and 5.3, to create separate judges for each task.

LM judges. For the LM judges, we use the same LMs as in the agent judges. Each LM is prompted to output a decision for every ablation, along with a reasoning field explaining its choice. The judges are also provided with an example illustrating how to perform the matching, similar to the demonstration used for the agent judges.

Metrics. We report precision, recall, and F1-score for the judges. We also report the Cohen's kappa (κ) agreement between human annotations and our judges' annotations. Evaluation is done on AuthorEval and ReviewerEval for the AuthorAblation judge and ReviewerAblation judge, respectively. The overall JudgeEval score is calculated as the average performance across the two tasks.

7.2 Experimental Setup: Planners

Number of generated ablations (k). We set $k = 5$ for AuthorAblation and $k = 2$ for ReviewerAblation, based on the benchmarks statistics presented in Table 4.

Agent planners. For the agent planners, we use SWE-agent with several frontier LMs: GPT-5.4, Llama 3.1 405B Instruct (Meta, 2024), GPT OSS 120B, MiniMax M2.5 and Qwen 3.5 122B. SWE-agent is configured with default tools for file inspection, editing, and command execution. We provide it with a command to submit the ablation plan and verify that it matches the required format. Prompts are tailored to both benchmark task, as described in Sections 4.2 and 5.2, resulting in two planners.

	Precision	Recall	F1	κ
LMJUDGE				
GPT-5.4	0.77	0.81	0.76	0.56
o3-mini	0.76	**0.86**	0.78	0.58
GPT-OSS-120B	0.70	0.75	0.63	0.45
MiniMax-M2.5	0.74	0.80	0.73	0.51
Qwen3.5-122B-A10B	**0.79**	0.81	0.77	0.57
Majority Vote	0.78	0.84	**0.79**	**0.62**
AGENTJUDGE				
GPT-5.4	0.70	0.74	0.69	0.45
o3-mini	**0.78**	0.78	0.74	0.53
GPT-OSS-120B	0.71	0.75	0.70	0.49
MiniMax-M2.5	0.74	**0.80**	0.73	0.49
Qwen3.5-122B-A10B	0.73	0.77	0.72	0.48
Majority Vote	**0.78**	0.79	**0.76**	**0.57**

Table 2: **Evaluating judges:** Macro-average results of the two judge scaffoldings for the JudgeEval. The majority vote is based on the three best models (in terms of κ) used in each scaffolding.

LM planners. For the LM planners, we use the same LMs as in the agent planners, with two additional smaller, reasoning-oriented models: o3-mini and Gemini 2.5 Flash (Google, 2025). Each LM is prompted to output an ablation plan based on the task definition, along with a reasoning field that explains its planning process.

Metrics. We report macro-average precision@k, recall@k, and F1-score@k. AblationBench score is calculated as the average metrics of the two tasks.

7.3 Experimental Results: Judges

Table 2 reports results for the judges on JudgeEval. AGENTJUDGE does not outperform LMJUDGE. This suggests that in short contexts, a single CoT step works better than multi-step reasoning. Below, we report agreement with human annotations and provide an error analysis to support this choice.

Majority vote better aligns with human annotations. We build a majority-vote ensemble using all three model sets per scaffolding. We then select the best ensemble by κ. For LMJUDGE, the best ensemble uses GPT-5.4, o3-mini, and MiniMax M2.5. For AGENTJUDGE, it uses o3-mini, GPT OSS 120B, and MiniMax M2.5. These ensembles also achieve the best classification scores. We adopt majority voting as it mitigates potential self-model biases and achieves better performance than the best individual LM, consistent with prior work (Verga et al., 2024).

Impact of inputs and models on judge performance. Tables 9 and 10 (Appendix C) analyzes errors of the best and worst LMJUDGE models on JudgeEval (o3-mini vs. GPT OSS 120B). On AuthorAblation, o3-mini mainly makes false negatives (51% of its errors). GPT OSS shows more surface matches (30%), suggesting reliance on lexical cues. On ReviewerAblation, o3-mini often misses ablations (34%). GPT OSS shows higher rates (38%), indicating sensitivity to ambiguous input. These trends explain the overall results (Tables 5 and 6): performance is higher on AuthorEval due to structured inputs, and lower on ReviewerEval due to unstructured review text.

	Precision@k	Recall@k	F1@k
LM-PLANNER			
GPT-5.4	**0.33**	**0.45**	**0.35**
o3-mini	0.26	0.33	0.27
Gemini 2.5 Flash	0.29	0.35	0.29
Llama 3.1 405B Instruct	0.24	0.33	0.26
GPT-OSS-120B	0.28	0.40	0.31
MiniMax-M2.5	0.29	0.37	0.30
Qwen3.5-122B-A10B	0.28	0.39	0.30
AGENT-PLANNER			
GPT-5.4	**0.33**	**0.44**	**0.35**
Llama 3.1 405B Instruct	0.24	0.26	0.23
GPT-OSS-120B	0.16	0.20	0.17
MiniMax-M2.5	0.18	0.24	0.19
Qwen3.5-122B-A10B	0.29	0.37	0.30

Table 3: **Evaluating planners:** Macro-average results of the two planner scaffoldings for the AblationBench.

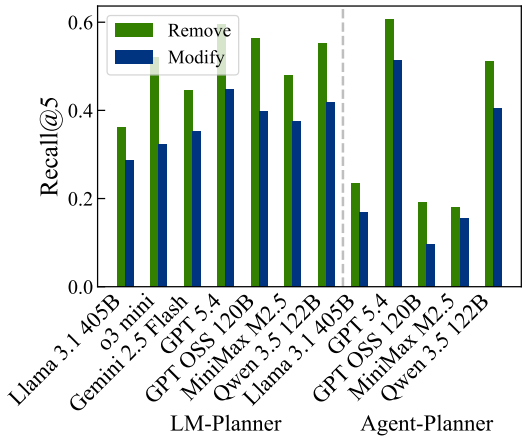


Figure 2: Recall@5 performance for AuthorAblation of each planner approach across different LMs, separated by ablation type.

7.4 Experimental Results: Planners

Table 3 presents the main results for AblationBench. Below, we analyze the performance of the planners, highlighting where they succeed and what limits their current capabilities.

Models are more likely to identify ablations that completely remove key components of the

method, rather than those that modify them. In Figure 2 we observe that recall@5 is substantially higher for removal-type ablations across planners and models, suggesting that LMs are more effective at identifying core components of a method. In contrast, modification-type ablations require not only identifying the component but also proposing relevant and feasible substitutions, which remains a challenge for current models. Our qualitative analysis (Appendix C) shows that modification type ablations across models often emphasizes hyperparameter variations rather than suggesting meaningful replacements for core components. Models also tend to produce a fixed number of ablations, without prioritizing the most informative ones.

Model grounding impacts performance across tasks. We qualitatively analyze ten randomly selected test papers, five per task, using the best- and worst-performing models, extracted from the results per each task. We observe a trade-off: models that perform well on one task tend to perform poorly on the other. Our analysis suggests that this behavior is associated with differences in model grounding. More strongly grounded models, such as Qwen, perform well on AuthorAblation by closely adhering to the method description and generating instruction-compliant ablation plans. However, this grounding appears to limit performance on ReviewerAblation as illustrated in Figure 3, where identifying missing ablations often requires creativity beyond the experiments explicitly described in the paper. In contrast, less grounded models such as Llama overlook contextual details, which hurts performance on AuthorAblation, but can be advantageous for ReviewerAblation. This observation aligns with prior findings on hallucination and factuality (Bang et al., 2025; Jacovi et al., 2025), particularly in long-context settings, which characterize both tasks. See Appendix C for a detailed analysis.

Performance varies by conference domain. When comparing performance across conference categories—General ML, CV, and NLP—we observe that F1 score is higher on CV papers and lower on NLP papers (0.45 vs. 0.30). Our qualitative analysis suggests this gap is due to differences in paper structure: CV papers typically follow a more standardized format with clearly defined components and ablations derived directly from this decomposition, while NLP papers are more variable and often less explicitly decompose components,

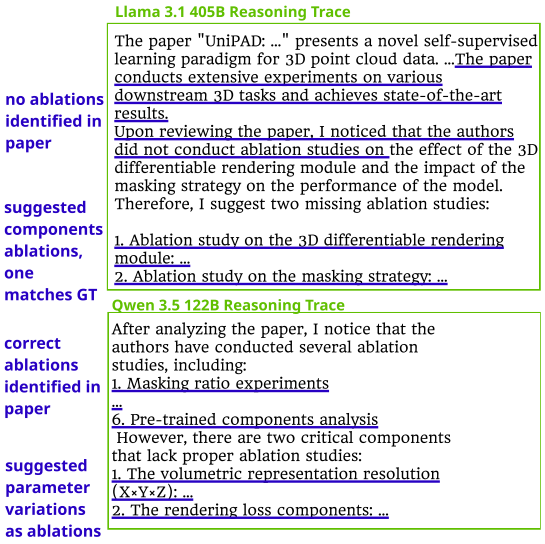


Figure 3: Qualitative analysis highlighting differences in the reasoning and grounding of Llama 3.1 405B and Qwen 3.5 122B on a ReviewerAblation instance.

making ablation planning harder (Figure 10).

LM-PLANNER VS. AGENT-PLANNER. Overall, LM-PLANNER achieves slightly better classification metrics than AGENT-PLANNER, outperforming it for Llama 3.1, GPT OSS, and MiniMax-M2.5. For GPT-5.4, the best-performing model, LM-PLANNER attains an F1 score on par with AGENT-PLANNER. We also observe no correlation between agent trajectory length and F1 score (Spearman: 0.07). These results suggest that, despite their ability to iterate over files and perform multi-step reasoning, the agent scaffold used does not provide a clear advantage over a single LM call with full context and one CoT step. To test whether this finding depends on the specific scaffolding used, we additionally evaluate GPTResearcher (Elovic, 2023) on AuthorAblation using its default configuration, while restricting retrieval from external sources containing the paper name to avoid contamination. GPTResearcher achieves an F1@5 of 0.24, further supporting our finding that current agent scaffolds do not improve performance on this task.

7.5 Human Baseline Performance

We establish a human baseline for AuthorAblation using 10 PhD students in machine learning, each selecting one of 15 randomly sampled papers and generating ablation plans from truncated versions of the papers (details in Appendix C.3). Humans achieve substantially higher performance than GPT-

5.4, with F1@5 of 0.66 vs. 0.43 and $\times 1.82$ precision, indicating a more selective and accurate identification of ablations. These results support the validity of the GT ablations and show that AblationBench remains a challenging, non saturated task with a clear gap for improvement.

8 Conclusion

We introduced **AblationBench**, a suite for evaluating ablation planning in empirical AI research. It includes **AuthorAblation**, which tasks models with generating ablations from a paper's method, and **ReviewerAblation**, which focuses on suggesting missing ablations from full submissions—capturing both author and reviewer perspectives. Each benchmark includes a baseline planner, and an LM judge for automatic evaluation of planners. CoT prompting outperforms agent-based methods in both accuracy and cost. The best model achieve an F1 score of 0.35 on AblationBench, leaving substantial room for improvement.

Limitations

Our current evaluation framework checks whether models generate ablations found in the paper or suggested in reviews. However, it has some limitations in the following.

LM-based judges achieve moderate human-model agreement. Our automatic evaluation framework relies on LM-based judges, which achieve strong classification performance but only moderate agreement with human annotators. While this level of performance is sufficient for reliable evaluation, improving human–model agreement could further increase the stability and robustness of the judges.

GT covers only ablations proposed by authors and reviewers. Models may generate valid and feasible ablations that are not present in the GT. We stress throughout this work that our GT as evaluation target is meaningful and that current models struggle in the task defined by AblationBench. Our qualitative analysis in Appendix C shows that, in most cases, GT ablations are preferred over those proposed by models, as model-generated plans often focus on hyperparameter variations rather than core methodological ablations. However, as models become stronger, they are expected to produce more meaningful ablation designs that go beyond the current GT, and such capabilities will be important to benchmark once AblationBench becomes saturated.

Potential data contamination. AblationBench uses published conference papers, ICLR submissions and their official reviews, which frontier LMs may have seen during training. However, current models still struggle substantially with the task, suggesting that memorization is not a major driver of performance. Further, across the six models with disclosed knowledge cutoffs, F1 on instances published before vs. after the cutoff has only a weak correlation (Pearson: 0.35). While contamination may become more relevant on newer models, our fully automated data construction pipeline makes it easy to add newly published papers and construct temporally held-out evaluations for future models.

We hope AblationBench will support future research toward developing stronger models capable of creative ablation planning.

References

Hadi Abdullah. 2025. Boosting membership inference attacks with upstream modification.

Toufique Ahmed, Premkumar Devanbu, Christoph Treude, and Michael Pradel. 2025. Can llms replace manual annotation of software engineering artifacts? *Preprint*, arXiv:2408.05534.

Anirudh Ajith, Mengzhou Xia, Alexis Chevalier, Tanya Goyal, Danqi Chen, and Tianyu Gao. 2024. Litsearch: A retrieval benchmark for scientific literature search. *Preprint*, arXiv:2407.18940.

Alexis Audran-Reiss, Jordi Armengol-Estapé, Karen Hambardzumyan, Amar Budhiraja, Martin Josifoski, Edan Toledo, Rishi Hazra, Despoina Magka, Michael Shvartsman, Parth Pathak, Justine T Kao, Lucia Cipolina-Kun, Bhavul Gauri, Jean-Christophe Gagnon-Audet, Emanuel Tewolde, Jenny Zhang, Taco Cohen, Yossi Adi, Tatiana Shavrina, and Yoram Bachrach. 2025. What does it take to be a good ai research agent? studying the role of ideation diversity. *Preprint*, arXiv:2511.15593.

Yejin Bang, Ziwei Ji, Alan Schelten, An-419 thony Hartshorn, Tara Fowler, Cheng Zhang, Nicola Cancedda, and Pascale Fung. 2025. Hallulens: Llm hallucination benchmark. *ArXiv*, abs/2504.17550.

Alina Beygelzimer, Yann Dauphin, Percy Liang, and Jennifer Wortman Vaughan. 2021. The NeurIPS 2021 Consistency Experiment.

Ben Bogin, Kejuan Yang, Shashank Gupta, Kyle Richardson, Erin Bransom, Peter Clark, Ashish Sabharwal, and Tushar Khot. 2024. Super: Evaluating

agents on setting up and executing tasks from research repositories. *Preprint*, arXiv:2409.07440.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Sébastien Bubeck, Christian Coester, Ronen Eldan, Timothy Gowers, Yin Tat Lee, Alexandru Lupsasca, Mehtaab Sawhney, Robert Scherrer, Mark Sellke, Brian K. Spears, Derya Unutmaz, Kevin Weil, Steven Yin, and Nikita Zhivotovskiy. 2025. Early science acceleration experiments with gpt-5. *Preprint*, arXiv:2511.16072.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shan Zhang, Jie Fu, and Zhiyuan Liu. 2023. Chateval: Towards better llm-based evaluators through multi-agent debate. *ArXiv*, abs/2308.07201.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, Lilian Weng, and Aleksander Mądry. 2025. Mle-bench: Evaluating machine learning agents on machine learning engineering. *Preprint*, arXiv:2410.07095.

Die Chen, Zhiwen Li, Mingyuan Fan, Cen Chen, Wenmeng Zhou, Yanhao Wang, and Yaliang Li. 2025. Growth inhibitors for suppressing inappropriate image concepts in diffusion models. In *The Thirteenth International Conference on Learning Representations*.

Xiang Chen, Ningyu Zhang, Lei Li, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022. Good visual guidance make a better extractor: Hierarchical visual prefix for multimodal entity and relation extraction. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1607–1618, Seattle, United States. Association for Computational Linguistics.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631, Toronto, Canada. Association for Computational Linguistics.

Corinna Cortes and Neil D. Lawrence. 2021. Inconsistency in conference peer review: Revisiting the 2014 neurips experiment. *Preprint*, arXiv:2109.09774.

Vijay Prakash Dwivedi and Xavier Bresson. 2021. A generalization of transformer networks to graphs. *Preprint*, arXiv:2012.09699.

Assaf Elovic. 2023. gpt-researcher.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire. In *North American Chapter of the Association for Computational Linguistics*.

Luke Gessler, Shabnam Behzad, Yang Janet Liu, Siyao Peng, Yilun Zhu, and Amir Zeldes. 2021. DisCoDisCo at the DISRPT2021 shared task: A system for discourse segmentation, classification, and connective detection. In *Proceedings of the 2nd Shared Task on Discourse Relation Parsing and Treebanking (DISRPT 2021)*, pages 51–62, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Google. 2025. Start building with gemini 2.5 flash. https://developers.googleblog.com/en/start-building-with-gemini-25-flash/.

Google Research. 2025. arXiv LaTeX Cleaner.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Anil Palepu, Petar Sirkovic, Artiom Myaskovsky, Felix Weissenberger, Keran Rong, Ryutaro Tanno, Khaled Saab, Dan Popovici, Jacob Blum, Fan Zhang, Katherine Chou, Avinatan Hassidim, Burak Gokturk, Amin Vahdat, Pushmeet Kohli, and 15 others. 2025. Towards an ai co-scientist. *Preprint*, arXiv:2502.18864.

Antoine Grosnit, Alexandre Maraval, James Doran, Giuseppe Paolo, Albert Thomas, Refinath Shahul Hameed Nabeezath Beevi, Jonas Gonzalez, Khyati Khandelwal, Ignacio Iacobacci, Abdelhakim Benechehab, Hamza Cherkaoui, Youssef Attia El-Hili, Kun Shao, Jianye Hao, Jun Yao, Balazs Kegl, Haitham Bou-Ammar, and Jun Wang. 2024. Large language models orchestrating structured reasoning achieve kaggle grandmaster level. *Preprint*, arXiv:2411.03562.

Ken Gu, Ruoxi Shang, Ruien Jiang, Keying Kuang, Richard-John Lin, Donghe Lyu, Yue Mao, Youran Pan, Teng Wu, Jiaqian Yu, Yikun Zhang, Tianmai M. Zhang, Lanyi Zhu, Mike A. Merrill, Jeffrey Heer, and Tim Althoff. 2024. Blade: Benchmarking language model agents for data-driven science. *Preprint*, arXiv:2408.09667.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. 2020. Gpt-gnn: Generative pre-training of graph neural networks. *Preprint*, arXiv:2006.15437.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. 2024a. Mlagentbench: Evaluating language agents on machine learning experimentation. *Preprint*, arXiv:2310.03302.

Xu Huang, Weiwen Liu, Xiaolong Chen, Xingmei Wang, Hao Wang, Defu Lian, Yasheng Wang, Ruiming Tang, and Enhong Chen. 2024b. Understanding the planning of llm agents: A survey. *CoRR*, abs/2402.02716.

Tal Ifargan, Lukas Hafner, Maor Kern, Ori Alcalay, and Roy Kishony. 2024. Autonomous llm-driven research from data to human-verifiable research papers. *Preprint*, arXiv:2404.17605.

Alon Jacovi, Andrew Wang, Chris Alberti, Connie Tao, Jon Lipovetz, Kate Olszewska, Lukas Haas, Michelle Liu, Nate Keating, Adam Bloniarz, Carl Saroufim, Corey Fry, Dror Marcus, Doron Kukliansky, Gaurav Singh Tomar, James Swirhun, Jinwei Xing, Lily Wang, Madhu Gurumurthy, and 7 others. 2025. The facts grounding leaderboard: Benchmarking llms' ability to ground responses to long-form input. *Preprint*, arXiv:2501.03200.

Yasamin Jafarian and Hyun Soo Park. 2021. Learning high fidelity depths of dressed humans by watching social media dance videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12753–12762.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446.

Ziming Li, Qianbo Zang, David Ma, Jiawei Guo, Tuney Zheng, Minghao Liu, Xinyao Niu, Yue Wang, Jian Yang, Jiaheng Liu, Wanjun Zhong, Wangchunshu Zhou, Wenhao Huang, and Ge Zhang. 2024. Autokaggle: A multi-agent framework for autonomous data science competitions. *Preprint*, arXiv:2410.20424.

Shanchuan Lin, Andrey Ryabtsev, Soumyadip Sengupta, Brian Curless, Steve Seitz, and Ira Kemelmacher-Shlizerman. 2020. Real-time high-resolution background matting. *Preprint*, arXiv:2012.07810.

Zachary C. Lipton and Jacob Steinhardt. 2019. Troubling trends in machine learning scholarship: Some ml papers suffer from flaws that could mislead the public and stymie future research. *Queue*, 17(1):45–77.

Yang Liu, Dan Iter, Yichong Xu, Shuo Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Conference on Empirical Methods in Natural Language Processing*.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. 2024. The ai scientist: Towards fully automated open-ended scientific discovery. *Preprint*, arXiv:2408.06292.

Baorui Ma, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. 2021. Neural-pull: Learning signed distance functions from point clouds by learning to pull space onto surfaces. *Preprint*, arXiv:2011.13495.

Mikel Malagón, Josu Ceberio, and Jose A. Lozano. 2025. Self-composing policies for scalable continual reinforcement learning. *Preprint*, arXiv:2506.14811.

Meta. 2024. Introducing llama 3.1: Our most capable models to date. https://ai.meta.com/blog/meta-llama-3-1/.

Richard Meyes, Melanie Lu, Constantin Waubert de Puiseau, and Tobias Meisen. 2019. Ablation studies in artificial neural networks. *Preprint*, arXiv:1901.08644.

MiniMax. 2026. Minimax m2.5: Built for real-world productivity. https://www.minimax.io/news/minimax-m25.

Arbi Haza Nasution and Aytuğ Onan. 2024. Chatgpt label: Comparing the quality of human-generated and llm-generated annotations in low-resource language nlp tasks. *IEEE Access*, 12:71876–71900.

Deepak Nathani, Lovish Madaan, Nicholas Roberts, Nikolay Bashlykov, Ajay Menon, Vincent Moens, Amar Budhiraja, Despoina Magka, Vladislav Vorotilov, Gaurav Chaurasia, Dieuwke Hupkes, Ricardo Silveira Cabral, Tatiana Shavrina, Jakob Foerster, Yoram Bachrach, William Yang Wang, and Roberta Raileanu. 2025. Mlgym: A new framework and benchmark for advancing ai research agents. *Preprint*, arXiv:2502.14499.

OpenAI, :, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, and 108 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

OpenAI. 2025. Openai o3-mini. https://openai.com/index/openai-o3-mini/.

OpenReview. 2025. OpenReview.net.

Vik Paruchuri. 2025. Marker.

Ori Press, Andreas Hochlehnert, Ameya Prabhu, Vishaal Udandarao, Ofir Press, and Matthias Bethge. 2024. CiteME: Can language models accurately cite scientific claims? In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Paul Roetzer and Florian Bernard. 2024. Spidermatch: 3d shape matching with global optimality and geometric consistency. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Minju Seo, Jinheon Baek, Seongyun Lee, and Sung Ju Hwang. 2025. Paper2code: Automating code generation from scientific papers in machine learning. *Preprint*, arXiv:2504.17192.

Yijia Shao, Yucheng Jiang, Theodore A. Kanell, Peter Xu, Omar Khattab, and Monica S. Lam. 2024. Assisting in writing wikipedia-like articles from scratch with large language models. *Preprint*, arXiv:2402.14207.

Jia Shi, Gautam Gare, Jinjin Tian, Siqi Chai, Zhiqiu Lin, Arun Vasudevan, Di Feng, Francesco Ferroni, and Shu Kong. 2024. Lca-on-the-line: Benchmarking out-of-distribution generalization with class taxonomies. *Preprint*, arXiv:2407.16067.

Alexander Shypula, Aman Madaan, Yimeng Zeng, Uri Alon, Jacob Gardner, Milad Hashemi, Graham Neubig, Parthasarathy Ranganathan, Osbert Bastani, and Amir Yazdanbakhsh. 2024. Learning performance-improving code edits. *Preprint*, arXiv:2302.07867.

Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. 2024. Can llms generate novel research ideas? a large-scale human study with 100+ nlp researchers. *Preprint*, arXiv:2409.04109.

Zachary S. Siegel, Sayash Kapoor, Nitya Nagdir, Benedikt Stroebl, and Arvind Narayanan. 2024. Core-bench: Fostering the credibility of published research through a computational reproducibility agent benchmark. *Preprint*, arXiv:2409.11363.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, Akshay Nathan, Alan Luo, Alec Helyar, Aleksander Madry, Aleksandr Efremov, Aleksandra Spyra, Alex Baker-Whitcomb, Alex Beutel, Alex Karpenko, and 465 others. 2025. Openai gpt-5 system card. *Preprint*, arXiv:2601.03267.

Michael D. Skarlinski, Sam Cox, Jon M. Laurent, James D. Braza, Michaela Hinks, Michael J. Hammerling, Manvitha Ponnapati, Samuel G. Rodriques, and Andrew D. White. 2024. Language agents achieve superhuman synthesis of scientific knowledge. *Preprint*, arXiv:2409.13740.

Zhangde Song, Jieyu Lu, Yuanqi Du, Botao Yu, Thomas M. Pruyn, Yue Huang, Kehan Guo, Xiuzhe Luo, Yuanhao Qu, Yi Qu, Yinkai Wang, Haorui Wang, Jeff Guo, Jingru Gan, Parshin Shojaee, Di Luo, Andres M Bran, Gen Li, Qiyuan Zhao, and 37 others. 2025. Evaluating large language models in scientific discovery. *Preprint*, arXiv:2512.15567.

Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, Johannes Heidecke, Amelia Glaese, and Tejal Patwardhan. 2025. Paperbench: Evaluating ai's ability to replicate ai research. *Preprint*, arXiv:2504.01848.

Viacheslav Surkov, Chris Wendler, Mikhail Terekhov, Justin Deschenaux, Robert West, and Caglar Gulcehre. 2024. Unpacking SDXL turbo: Interpreting text-to-image models with sparse autoencoders.

Jiabin Tang, Lianghao Xia, Zhonghang Li, and Chao Huang. 2025. Ai-researcher: Autonomous scientific innovation. *Preprint*, arXiv:2505.18705.

Qwen Team. 2026. Qwen3.5-omni technical report. *Preprint*, arXiv:2604.15804.

Minyang Tian, Luyu Gao, Shizhuo Dylan Zhang, Xinan Chen, Cunwei Fan, Xuefei Guo, Roland Haas, Pan Ji, Kittithat Krongchon, Yao Li, Shengyan Liu, Di Luo, Yutao Ma, Hao Tong, Kha Trinh, Chenyu Tian, Zihan Wang, Bohao Wu, Yanyu Xiong, and 11 others. 2024a. Scicode: A research coding benchmark curated by scientists. *Preprint*, arXiv:2407.13168.

Yuchuan Tian, Hanting Chen, Chao Xu, and Yunhe Wang. 2024b. Image processing gnn: Breaking rigidity in super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24108–24117.

Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. 2024. Replacing judges with juries: Evaluating llm generations with a panel of diverse models. *Preprint*, arXiv:2404.18796.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Annual Meeting of the Association for Computational Linguistics*.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450, Bangkok, Thailand. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Hjalmar Wijk, Tao Lin, Joel Becker, Sami Jawhar, Neev Parikh, Thomas Broadley, Lawrence Chan, Michael Chen, Josh Clymer, Jai Dhyani, Elena Ericheva, Katharyn Garcia, Brian Goodrich, Nikola Jurkovic, Megan Kinniment, Aron Lajko, Seraphina Nix, Lucas Sato, William Saunders, and 3 others. 2024. Re-bench: Evaluating frontier ai r&d capabilities of language model agents against human experts. *Preprint*, arXiv:2411.15114.

Yijia Xiao, Runhui Wang, Luyang Kong, Davor Golac, and Wei Wang. 2025. Csr-bench: Benchmarking llm agents in deployment of computer science research repositories. *Preprint*, arXiv:2502.06111.

Wenda Xu, Guanglei Zhu, Xuandong Zhao, Liangming Pan, Lei Li, and William Wang. 2024. Pride and prejudice: LLM amplifies self-bias in self-refinement. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15474–15492, Bangkok, Thailand. Association for Computational Linguistics.

Honghui Yang, Sha Zhang, Di Huang, Xiaoyang Wu, Haoyi Zhu, Tong He, SHIXIANG TANG, Hengshuang Zhao, Qibo Qiu, Binbin Lin, Xiaofei He, and Wanli Ouyang. 2024a. UniPAD: A universal pre-training paradigm for autonomous driving.

John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik R Narasimhan, and Ofir Press. 2024b. SWE-agent: Agent-computer interfaces enable automated software engineering. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Yuzhe Yang, Kaiwen Zha, Ying-Cong Chen, Hao Wang, and Dina Katabi. 2021. Delving into deep imbalanced regression. *Preprint*, arXiv:2102.09554.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023b. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

Ailing Zeng, Lei Yang, Xuan Ju, Jiefeng Li, Jianyi Wang, and Qiang Xu. 2022. Smoothnet: A plug-and-play network for refining human poses in videos. *Preprint*, arXiv:2112.13715.

Ningyu Zhang, Luoqiu Li, Xiang Chen, Shumin Deng, Zhen Bi, Chuanqi Tan, Fei Huang, and Huajun Chen. 2022. Differentiable prompt makes pre-trained language models better few-shot learners. In *International Conference on Learning Representations*.

Zhihao Zhang, Alan Zhu, Lijie Yang, Yihua Xu, Lanting Li, Phitchaya Mangpo Phothilimthana, and Zhihao Jia. 2024. Accelerating retrieval-augmented language model serving with speculation.

Bowen Zhao, Hannaneh Hajishirzi, and Qingqing Cao. 2024. Apt: Adaptive pruning and tuning pretrained language models for efficient training and inference. *Preprint*, arXiv:2401.12200.

Yilun Zhao, Weiyuan Chen, Zhijian Xu, Manasi Patwardhan, Chengye Wang, Yixin Liu, Lovekesh Vig, and Arman Cohan. 2025. AbGen: Evaluating large language models in ablation study design and evaluation for scientific research. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12479–12491, Vienna, Austria. Association for Computational Linguistics.

Junhao Zheng, Qianli Ma, Shengjie Qiu, Yue Wu, Peitian Ma, Junlong Liu, Huawen Feng, Xichen Shang, and Haibin Chen. 2023a. Preserving commonsense knowledge from pre-trained language models via causal inference. *Preprint*, arXiv:2306.10790.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023b. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Han Zhou, Xingchen Wan, Lev Proleev, Diana Mincu, Jilin Chen, Katherine Heller, and Subhrajit Roy. 2024. Batch calibration: Rethinking calibration for in-context learning and prompt engineering. *Preprint*, arXiv:2309.17249.

Mingchen Zhuge, Changsheng Zhao, Dylan Ashley, Wenyi Wang, Dmitrii Khizbullin, Yunyang Xiong, Zechun Liu, Ernie Chang, Raghuraman Krishnamoorthi, Yuandong Tian, Yangyang Shi, Vikas Chandra, and Jürgen Schmidhuber. 2024. Agent-as-a-judge: Evaluate agents with agents. *Preprint*, arXiv:2410.10934.

A AblationBench Construction

A.1 Constructing AuthorAblation

We construct the AuthorAblation benchmark using papers from previous datasets designed for scientific reasoning over machine learning papers. Specifically, we aggregate papers from CSR-Bench (Xiao et al., 2025) (100 papers), SUPER-Expert (Bogin et al., 2024) (45 papers), and PaperBench (Starace et al., 2025) (20 papers). These sources were selected due to their size, high-quality paper selection process, availability of code, diversity of conferences and recency.

To further increase topical and conference diversity, we supplement these with best paper awardees from major computer vision conferences (CVPR, ICCV) from 2020 to 2024 (38 papers).

All selected papers should satisfy two criteria: they should include ablation studies and should have publicly released code, enabling future extensions of the benchmark to code development and execution settings.

Based on these criteria we curate the benchmark using a three-stage pipeline:

1. Filtering for papers with ablations. We collect the TEX source of all candidate papers and discard those that do not include ablation studies. This filtering is performed via keyword search ("ablat") and manual validation.

2. Partial paper creation. For each remaining paper, we create a truncated version containing only the sections up to and including the method. To ensure that only content relevant to the model's task is included, we apply the arXiv LATEX cleaner utility (Google Research, 2025) to remove comments and any unreferenced files, such as figures or tables associated with later sections. This setup ensures that the model generates ablations based solely on the method description, without access to experimental results or analysis.

3. Ground-truth (GT) extraction. The ablation plan described in the full paper is manually extracted into a structured JSON format, which serves as the GT output for evaluation. The instructions for annotators are given in Figure 5. All extracted annotations are manually inspected against the original papers to verify correctness and consistency.

AuthorAblation - Instructions for Annotators

We are building a new benchmark to evaluate the scientific research capabilities of language models (LLMs). Our current focus is on asking LLM agents to plan the ablation study of a paper, given its method section.
Your task is to identify and record ablation studies from a research paper.
Please follow the exact instructions to make our ground-truth exact and valid.
Please do NOT use any GenAI for the task.

What to include:

- Extract only experiments that are explicitly described by the authors as ablation studies.
- The paper must use the term "ablation" to describe the experiment.
- Ablation studies may appear anywhere in the paper, including the appendix.

What to exclude:

- Do not include experiments that are not labeled as ablations by the authors, even if they look similar.
- Specifically exclude:
  Hyperparameter tuning experiments.
  Baseline or model comparisons.
  Any experiment not explicitly called an ablation.

How to record each ablation:
For each ablation study, create a JSON entry with the following fields:

1. `name`: an indicative name provided by the authors.
2. `ablated_part`: the component being ablated.
3. `action`: one of `REMOVE`, `REPLACE`, or `ADD`, indicating the modification applied to the component.
4. `replacement`: a list of components used as substitutes when the action is `REPLACE` or `ADD`.
5. `metrics`: a list of evaluation metrics reported for the ablation experiment.

Figure 5: The instructions for the annotators of AuthorAblation.

Through the process of benchmark curation, the original 203 papers are filtered down to the 83 papers which comprise AuthorAblation. We further split the 83 instances into two splits using random selection – development split containing 21 papers and test split containing 62 papers.

A.2 Constructing ReviewerAblation

Open review policies adopted by conferences such as ICLR provide data to study the peer review pro-

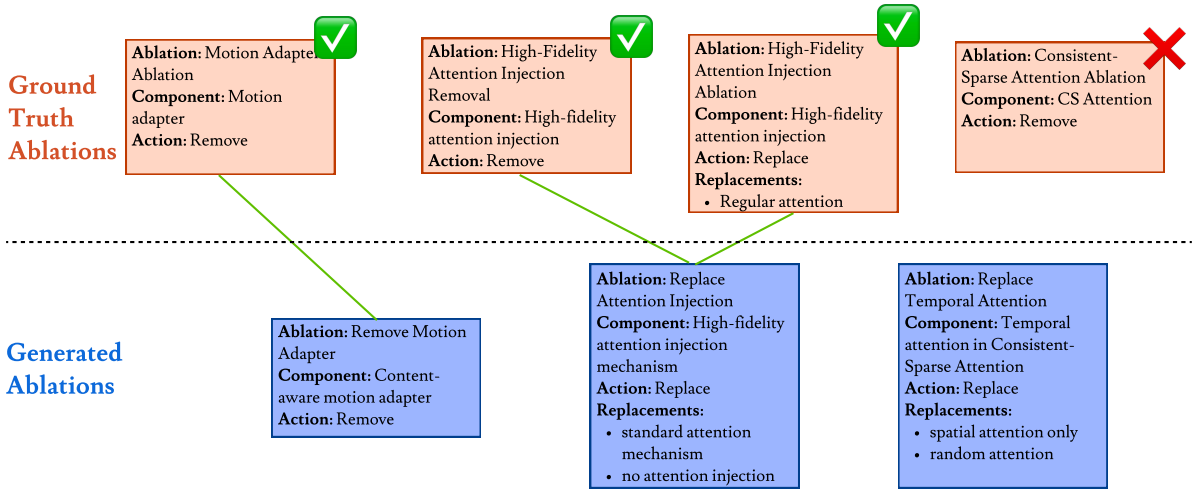


Figure 4: An example of the matching criteria in AuthorAblation instance, showing alignment between GT (gold, top) and generated (blue, bottom) ablations. Matching is done semantically, and is represented as a bipartite graph: nodes correspond to GT and generated ablations and edges indicate valid matches. Evaluation metrics are computed with respect to the GT ablations; in this example, both precision and recall equal 0.75.

cess at scale, including reviewer-initiated suggestions for additional ablations. To construct the ReviewerAblation benchmark, we use submissions to the ICLR conference from 2023 to 2025, focusing on cases where reviewers proposed ablation experiments that were not included in the original submission. This setup supports the development of reviewer-assistive models that can automatically identify potential gaps in a paper's empirical evaluation.

We curate the benchmark using three-stages pipeline:

1. Submission collection and basic filtering. We collect all $\sim 89,100$ reviews associated with $\sim 22,800$ ICLR submissions from 2023 to 2025 and filter for those containing the substring "ablat".

2. Filtering for reviews suggesting new ablations. We apply an LM (GPT-4o) with CoT prompting (Figure 6) to identify reviews that explicitly suggest new ablation experiments not already included in the paper. We calibrate this filter on the development set by manually annotating the number of missing ablation suggestions appearing in each review and comparing them to the model predictions. On this set, the CoT-based filter achieves 88% accuracy in predicting the number of suggested ablations, with a mean absolute error of 0.46.

3. Paper collection. For each selected submission, we retrieve the submitted preprint PDF and convert it to markdown format using the `marker` tool (Paruchuri, 2025). This produces a standardized and machine-readable version of the full paper used as input to models.

Instance Prompt

You are given a peer review of a machine learning paper submitted to ICLR. Your task is to determine how many new ablation study experiments the reviewer is suggesting that the paper is missing.

DEFINITION:
A missing ablation refers to an experiment that the reviewer believes should have been conducted, in which a specific component, module, feature, or design choice is removed, replaced, or altered in order to assess its impact on the model's performance. The review must clearly indicate that such an ablation study is missing or should be added.

You should answer with a number > 0 only if:
1. The review explicitly mentions the word ablation or ablate, and
2. It refers to a study that the paper did not perform, and
3. The reviewer is asking for that study to be added or noting that it is missing.

You should answer with a number = 0 if:
1. The review only discusses existing ablations.
2. The review mentions comparisons to baselines, datasets, or methods but not as ablation studies.
3. The word "ablation" appears in a different context (e.g., praising existing ablations).
4. The reviewer mentions the lack of ablation study but doesn't give any specific suggestions.

EXAMPLES:
1. "Lack of Ablation Study: An ablation study investigating the impact of different components of the advanced memory module (e.g., summary length, specific aspects included) would provide a deeper understanding of its effectiveness. This would be an interesting result to many. AFAIK, lots and lots of memory mechanisms have been tested in many different contexts, but very few have of-

fered insights on how memory mechanisms should be constructed. ." → The reviewer specifically mentions that the paper lacks an ablation study regarding the impact of different components of the advanced memory module, and mentions two possible components of it that should be ablated. → 2
2. "Lack of ablation study." → The reviewer mentions the lack of ablation studies but doesn't suggest any specific ones. → 0
3. "The improvement over GeoDiff is not significant to me. Could the author provide more ablation study about the f_σ function in Eq7, which can help to verify the importance of the proposed MB diffusion distribution." → The reviewer specifically mentions that the paper lacks an ablation study regarding the f_σ function in Eq7, and mentions that it is important to verify the importance of the proposed MB diffusion distribution. → 1
4. "The experiments are limited. A more comprehensive evaluation is needed to trust the effectiveness of the proposed method." → The reviewer mentions the lack of experiments but doesn't refer to ablation study. → 0

Figure 6: The instance prompt for filtering ICLR reviews to get the number of ablation suggestion per each review (if any) for ReviewerAblation.

Through the process of benchmark curation, the original set of 22,800 ICLR submissions was filtered down to the 5,960 papers included in ReviewerAblation-AblationBench. Among these, 1,844 submissions were accepted (31%) and 4,116 were rejected (69%). We randomly select from the 5,960 papers a development set of 50 papers, and a test set of 300 papers.

A.3 Licenses

AuthorAblation consists of papers from datasets with permissive licenses. CSR-Bench (Xiao et al., 2025) includes only papers with permissive licenses as stated in their paper, SUPER-Expert (Bogin et al., 2024) is released under the Apache-2.0 license, and PaperBench (Starace et al., 2025) uses the MIT license. Additional papers from CVPR and ICCV are licensed under CC-BY-4.0. For the creation of AuthorAblation, papers were modified through truncation, as permitted by their licenses. Papers were not anonymized, since authors can often be identified from the paper title and content.

For ReviewerAblation, all selected papers and reviews are sourced from OpenReview and released under the CC-BY-4.0 license, in accordance with OpenReview's terms and conditions. Papers were collected in their anonymized versions whenever available, while reviews are always anonymized under ICLR's double-blind review policy. Papers and reviews were used as-is without modification.

	Mean	Median
(a) # ablations in GT		
AuthorAblation	3.7	3
ReviewerAblation	1.8	1
(b) # TEX files in truncated paper	4.2	2
(c) # figures		
AuthorAblation	6.8	4
ReviewerAblation	9.4	8

Table 4: Mean and median of attributes for a task instance in AblationBench test set. Row (b) is only relevant for AuthorAblation.

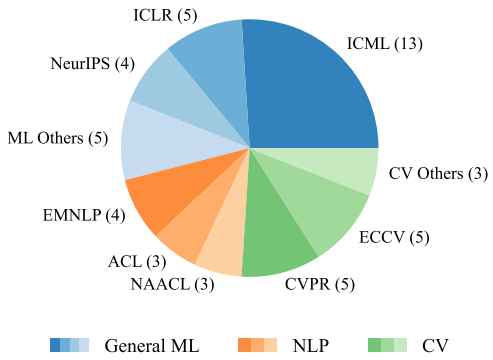


Figure 7: Conference distribution of AuthorAblation test set.

B Properties of AblationBench

AblationBench, composed of AuthorAblation and ReviewerAblation, offers a comprehensive evaluation suite for ablation planning in empirical AI research, covering diverse settings, topics, and ablation types. Key statistics of the test sets and benchmark characteristics are presented below.

Diversity of ablations. Table 4(a) shows the number of GT ablations in each benchmark. AuthorAblation includes 230 ablations (4 per paper on average), with 59% involving modifications and 41% removals. Modification ablations often requiring multiple valid alternatives (3.3 on average). In ReviewerAblation, papers contain on average 2 reviewer-suggested ablations, with over half containing only one.

Importance of ablation studies in review process. In ReviewerAblation, nearly half (48%) of all ICLR submissions between 2023 and 2025 include at least one review mentioning the keyword "ablat", with about one out of five reviews mention ablations explicitly. This indicates that ablations are a frequent point of feedback throughout the review process. Among the 300 submissions that

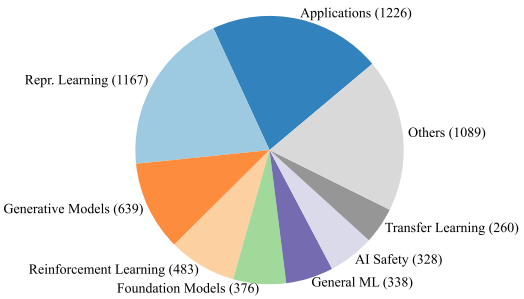


Figure 8: Topic distribution of ReviewerAblation full test set, extracted from OpenReview.

comprise the test set of the benchmark, 31% were accepted—matching the overall ICLR acceptance rate—suggesting that reviewer attention to ablations appears independent of final decisions.

Conferences and topics. AuthorAblation spans across 14 conferences in multiple areas of AI research, including general machine learning conferences, NLP, and computer vision. The distribution of conferences is shown in Figure 7. ReviewerAblation covers a diverse set of topics (Figure 8) such as AI applications, representation learning, and generative models. This diversity across conferences and research areas enables AblationBench to support evaluation of ablation planning across a wide range of AI domains.

	Precision	Recall	F1	κ
LMJUDGE				
w/ GPT-5.4	0.83	0.89	0.83	0.55
w/ o3-mini	0.81	0.89	0.81	0.57
w/ GPT-OSS-120B	0.81	0.88	0.79	0.43
w/ MiniMax-M2.5	0.79	0.86	0.76	0.47
w/ Qwen3.5-122B-A10B	0.85	0.84	0.80	0.51
Majority Vote	**0.86**	**0.91**	**0.86**	**0.66**
AGENTJUDGE				
w/ GPT-5.4	0.79	0.88	0.79	0.43
w/ o3-mini	**0.86**	0.85	0.80	0.51
w/ GPT-OSS-120B	0.84	**0.92**	**0.84**	0.53
w/ MiniMax-M2.5	0.77	0.82	0.74	0.38
w/ Qwen3.5-122B-A10B	0.81	0.91	0.80	0.42
Majority Vote	0.85	0.88	0.82	**0.54**

Table 5: **Evaluating AuthorAblation Judge:** Macro-average results of the two judge scaffoldings measured on AuthorEval.

	Precision	Recall	F1	κ
LMJUDGE				
w/ GPT-5.4	0.70	0.72	0.69	0.56
w/ o3-mini	0.71	**0.82**	**0.74**	0.59
w/ GPT-OSS-120B	0.58	0.62	0.47	0.47
w/ MiniMax-M2.5	0.69	0.74	0.69	0.54
w/ Qwen3.5-122B-A10B	**0.72**	0.77	0.73	**0.63**
Majority Vote	0.69	0.77	0.71	0.57
AGENTJUDGE				
w/ GPT-5.4	0.60	0.59	0.58	0.46
w/ o3-mini	0.69	0.71	0.68	0.55
w/ GPT-OSS-120B	0.57	0.57	0.55	0.45
w/ MiniMax-M2.5	**0.71**	**0.77**	**0.72**	0.59
w/ Qwen3.5-122B-A10B	0.64	0.63	0.63	0.53
Majority Vote	**0.71**	0.70	0.69	**0.59**

Table 6: **Evaluating ReviewerAblation Judge:** Macro-average results of the two judge scaffoldings measured on ReviewerEval.

C Experiments and Results

This section presents additional results and analysis from the experiments in Section 7, including per-task results, error analysis for both judges, and qualitative analysis of the planners. We observe that models that perform well on one task often perform poorly on the other: high-level reasoning is advantageous in ReviewerAblation, allowing models like Llama to suggest potentially useful ablations even without deeply analyzing the paper, while fine-grained, detailed reasoning benefits AuthorAblation, where precise analysis of method components is crucial. Overall, this illustrates how reasoning style influences planner performance across tasks.

	Precision@5 ↑	Recall@5 ↑	F1@5 ↑	nDCG@5 ↑
LM-PLANNER				
w/ GPT-5.4	**0.35**	**0.52**	**0.39**	**0.53**
w/ o3-mini	0.31	0.42	0.33	0.45
w/ Gemini 2.5 Flash	0.32	0.39	0.32	0.41
w/ Llama 3.1 405B Instruct	0.23	0.33	0.25	0.34
w/ GPT-OSS-120B	0.33	0.49	0.37	0.51
w/ MiniMax-M2.5	0.30	0.41	0.32	0.41
w/ Qwen3.5-122B-A10B	0.33	0.48	0.36	0.50
AGENT-PLANNER				
w/ GPT-5.4	**0.37**	**0.52**	**0.40**	**0.55**
w/ Llama 3.1 405B Instruct	0.23	0.22	0.21	0.23
w/ GPT-OSS-120B	0.10	0.14	0.11	0.16
w/ MiniMax-M2.5	0.12	0.18	0.13	0.19
w/ Qwen3.5-122B-A10B	0.33	0.45	0.35	0.47

Table 7: **Evaluating AuthorAblation Planners:** Macro-average results of the two planner scaffoldings.

	Precision@2 ↑	Recall@2 ↑	F1@2 ↑
LM-PLANNER			
w/ GPT-5.4	**0.31**	**0.37**	**0.31**
w/ o3-mini	0.20	0.24	0.20
w/ Gemini 2.5 Flash	0.25	0.30	0.26
w/ Llama 3.1 405B Instruct	0.25	0.32	0.27
w/ GPT-OSS-120B	0.23	0.31	0.25
w/ MiniMax-M2.5	0.27	0.32	0.28
w/ Qwen3.5-122B-A10B	0.23	0.29	0.24
AGENT-PLANNER			
w/ GPT-5.4	**0.29**	**0.35**	**0.30**
w/ Llama 3.1 405B Instruct	0.24	0.29	0.24
w/ GPT-OSS-120B	0.21	0.25	0.22
w/ MiniMax-M2.5	0.24	0.29	0.25
w/ Qwen3.5-122B-A10B	0.25	0.29	0.25

Table 8: **Evaluating ReviewerAblation Planners:** Macro-average results of the two planner scaffoldings.

C.1 AuthorAblation

Varying the number of generated ablations (k) has a minor effect on performance. Figure 9 shows the results for different values of k using the best-performing model. As expected, precision decreases as k increases, while recall improves. Overall, $k = 4, 5$ offers the best trade-off between precision and recall.

Models tend to recover important ablations first. For AuthorAblation, we additionally report *nDCG@k* (Järvelin and Kekäläinen, 2002), which evaluates ranking quality in search algorithms. Since GT ablations are ordered by their appearance in the paper, we use nDCG@k as a proxy for how well a model recovers more important ablations, under the assumption that the order of ablations in the paper roughly reflects their relative importance. In our setting, we define relevance for nDCG@k as a binary indicator of whether the ablation appears in the paper. Table 7 indicates a strong correlation between nDCG@5 and Recall@5 (Pearson correlation of 0.97 on average across all models). This indicates that when models successfully recover ablations, they tend to prioritize those that appear earlier in the paper. Interpreting earlier appearance as a proxy for ablation importance, this suggests that models preferentially recover more central ablations before less prominent ones.

C.1.1 Qualitative Analysis

In the following, we qualitatively analyze the LM-PLANNER using Qwen 3.5 and Llama 3.1, on five randomly selected papers. This analysis examines the models' reasoning traces to understand how they plan ablations, which types of ablations they propose, and whether they suggest ablations that go beyond or improve upon those reported by the original authors. We provide reasoning traces for all papers as part of the supplementary material.

GPT-GNN: Generative Pre-Training of Graph Neural Networks (Hu et al., 2020). GPT-GNN proposes a framework to initialize GNNs by generative pre-training, by factorizing the likelihood of graph generation into two components: (1) attribute generation and (2) edge generation. The GT evaluation includes six ablations:

1. Removing the attribute generation component.
2. Removing the edge generation component.
3. Removing both attribute generation and node separation.

Error Type	Description	o3-mini (%)	GPT OSS 120B (%)
False Negative	The judge marked it as "not found" even though the ablation appears in the prediction.	51	17
Replacement Mismatch	The correct component is identified, but the suggested replacements are not aligned with the GT.	9	20
Surface-Level Match	The match is based on similar wording, but the actual meaning does not align.	7	30
Partial Match	The predicted ablation is either broader or narrower than the GT.	16	26
Action Type Mismatch	The action (e.g., remove vs. modify) in the predicted ablation differs from the GT.	12	5
Other	Miscellaneous errors that do not fall into the above categories (e.g., formatting errors).	5	2

Table 9: Analysis of prediction errors for AuthorAblation judge, using LMJUDGE powered by o3-mini and GPT OSS, measured on AuthorEval. Percentage indicates the percentage of all errors.

Error Type	Description	o3-mini (%)	GPT OSS 120B (%)
Misinterpreted due to review breadth	The review includes an ablation suggestion, but the model draws an incorrect conclusion due to vague or broad review suggestions.	25	38
Ablation not identified	The model fails to identify the ablation suggestion in the review (although it appears).	34	37
Misinterpreted due to contextual understanding	The model identifies the ablation but misinterprets it in the context of the paper.	17	5
Incorrectly flags a non-ablation experiment	The model mistakenly classifies an unrelated experiment suggestion as a suggested ablation.	19	18
Other	Miscellaneous errors that do not fall into the above categories (e.g., formatting errors).	5	2

Table 10: Analysis of prediction errors for ReviewerAblation judge, using LMJUDGE powered by o3-mini and GPT OSS, measured on ReviewerEval. Percentage indicates the percentage of all errors.

4. Removing edge generation together with the adaptive queue.

5. Replacing the GNN architecture (GCN, GAT, RGCN, HAN).

6. Varying the amount of labeled data during fine-tuning.

Both models successfully identify the main components of the method: attribute generation, edge generation, node separation, and adaptive queue—which correspond to the core mechanisms targeted in the first four GT ablations. Qwen's reasoning trace (Figure 11) shows a clear methodological analysis: it first identifies the main contributions and then proposes ablations based on this analysis. The model successfully recovers the first four GT ablations, all of these correspond to removal-type ablations, categories where models also perform best quantitatively (Figure 2). It additionally suggests fine-grained modifications, such as changes to masking strategies or sampling procedures. While these ablations are relevant to the method, they are closer to hyperparameter tuning than true ablation studies, as they do not directly attribute performance to the paper's central contributions. Llama, in contrast, produces a less structured reasoning trace (Figure 12) that mixes component identification with ablation suggestions. It favors removal-type ablations and recovers the main removal ablations for attribute generation, edge generation, and the adaptive queue. However, some of its proposals, such as removing the entire pretraining stage, do not isolate specific contributions of the method, providing limited insight into the mechanisms responsible for GPT-GNN's performance. Overall, Llama identifies the main components but designs ablations that are less precise and less aligned with the goals of a proper ablation study.

Reasoning - Qwen 3.5

Based on the paper, GPT-GNN's main contributions are:
1. A novel attributed graph generation task that factorizes into Attribute Generation and Edge Generation
2. An efficient framework to conduct generative pre-training
3. Handling of large-scale graphs using sub-graph sam-

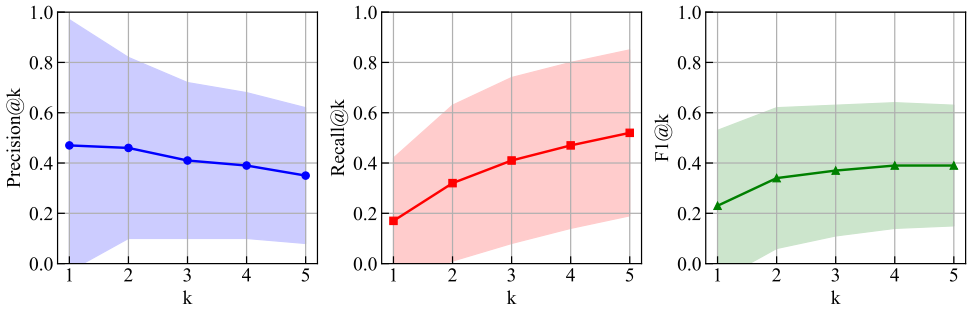


Figure 9: Evaluation metrics for AuthorAblation using the LM-PLANNER with GPT-5.4, across varying values of k, number of generated ablations. Standard error presented across papers.

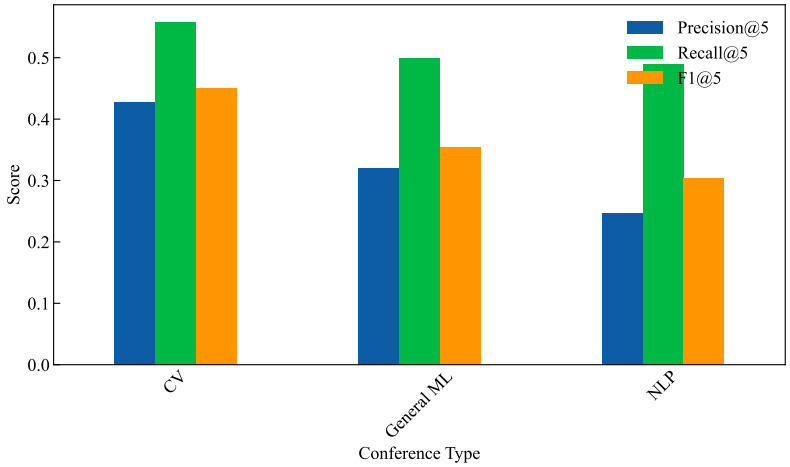


Figure 10: Metrics@5 per conference type measured on AuthorAblation using the LM-PLANNER with GPT-5.4. We categorize the conferences in the following manner: (1) *General ML:* ICML, NeurIPS, ICLR, AAAI, IJCAI, KDD; (2) *CV:* CVPR, ECCV, ICPR, IEEE, ICCV; and (3) *NLP:* ACL, EMNLP, NAACL.

pling and adaptive embedding queue
Key ablation studies should focus on:
1. The importance of the dual-generation approach (attributes and edges)
2. The effectiveness of the adaptive queue for negative samples
3. The impact of separating nodes into Attribute Generation and Edge Generation types
4. The sampling strategy for observed vs. masked edges
5. The sub-graph sampling approach for large graphs
I'll rank these by importance and formulate specific ablation experiments:

Figure 11: The reasoning of LM-PLANNER with Qwen 3.5 in generating the ablation plans for GPT-GNN.

Reasoning - Llama 3.1 405B

To design ablation studies for the GPT-GNN framework, we need to consider the key components of the method and how they contribute to its overall performance. The GPT-GNN framework is designed for generative pre-training of graph neural networks, and it involves several important components, including the use of a self-supervised attributed graph generation task, the factorization of the graph generation objective into two components (Attribute Generation and Edge Generation), and the use of an adaptive node embedding queue to mitigate the inaccurate loss brought by negative sampling.
Based on these components, we can design several ablation studies to investigate their impact on the performance of the GPT-GNN framework.
1. **Ablation of the self-supervised attributed graph generation task**: In this ablation study, we can remove the self-supervised attributed graph generation task and instead use a traditional supervised learning approach to train the GNN model. This will allow us to investigate the importance of the self-supervised task in the GPT-GNN framework.
2. **Ablation of the Attribute Generation component**: In this ablation study, we can remove the Attribute Generation component and only use the Edge Generation component to generate the graph. This will allow us to investigate the importance of modeling the attributes of the nodes in the graph.
3. **Ablation of the Edge Generation component**: In this ablation study, we can remove the Edge Generation component and only use the Attribute Generation component to generate the graph. This will allow us to investigate the importance of modeling the edges between the nodes in the graph.
4. **Ablation of the adaptive node embedding queue**: In this ablation study, we can remove the adaptive node embedding queue and instead use a traditional negative sampling approach. This will allow us to investigate

the importance of the adaptive node embedding queue in mitigating the inaccurate loss brought by negative sampling.
5. **Ablation of the pre-training task**: In this ablation study, we can remove the pre-training task and instead train the GNN model from scratch on the downstream task. This will allow us to investigate the importance of the pre-training task in the GPT-GNN framework.
By conducting these ablation studies, we can gain a better understanding of the importance of each component in the GPT-GNN framework and how they contribute to its overall performance.

Figure 12: The reasoning of LM-PLANNER with Llama 3.1 405B in generating the ablation plans for GPT-GNN.

SmoothNet: A Plug-and-Play Network for Refining Human Poses in Videos (Zeng et al., 2022). SmoothNet introduces a temporal-only refinement network designed to reduce jitter in video-based human pose estimation, for rarely seen or occluded actions. The paper evaluates its design through seven GT ablations that examine both architectural and training choices:

1. Replacing the FCN-based temporal model with alternative temporal models (Gaussian filtering, TCNs, and Transformers).
2. Replacing the proposed motion-aware temporal network with a simpler residual FCN.
3. Varying the temporal window size.
4. Removing the acceleration loss.
5. Removing the pose loss.
6. Replacing the motion modality representation.
7. Replacing the normalization strategy.

Both models identify the key components of the method, including the motion-aware temporal network and temporal modeling choices. Qwen demonstrates strong alignment with the paper's main contributions (Figure 13) and successfully recovers the majority of GT ablations, including the FCN-based temporal model, motion-aware network, window size, and loss-related ablations. It occasionally proposes deviations from the GT, such as joint-wise versus independent processing instead of the pose-loss removal; these ablations are still relevant but reflect a finer-grained focus on design choices rather than the high-level attribution of performance to main components. Llama, by contrast, exhibits a less structured reasoning trace (Figure 14), combining identification of components with ablation suggestions in a higher-level and imprecise manner. It confuses components and metrics, frames some ablations vaguely, and proposes ideas (e.g., transferability) that are not considered ablation studies. Consistent with the previous paper, Llama favors removal-type ablations and suggests broadly relevant research ideas rather than targeted experiments that isolate the contributions of specific components.

Reasoning - Qwen 3.5

Based on the paper, SmoothNet is a temporal-only refinement network that aims to reduce jitters in human pose estimation. The key components that warrant investigation through ablation studies are:
1. The motion-aware architecture - The paper introduces velocity and acceleration branches which should be validated.
2. The temporal modeling approach - The authors chose FCN over TCN and Transformer, which needs validation.
3. The loss function components - The paper uses both pose and acceleration losses.
4. The temporal window size - This is a critical hyperparameter that affects the model's ability to capture long-range dependencies.
5. The independence of joint processing - The model processes each axis independently.
I'll suggest ablation studies to validate these key design choices, focusing on the most important aspects first.
The metrics used in the paper are:
- MPJPE (Mean Per Joint Position Error) for accuracy
- Accel (Mean Per Joint Acceleration Error) for smoothness

Figure 13: The reasoning of LM-PLANNER with Qwen 3.5 in generating the ablation plans for SmoothNet.

Reasoning - Llama 3.1 405B

To design ablation studies for the SmoothNet paper, we need to identify the key components of the method and analyze their impact on the overall performance. The SmoothNet is a plug-and-play network for refining human poses in videos, which uses a temporal-only refinement approach to mitigate jitters.
One potential ablation study is to analyze the effect of the motion-aware fully-connected network on the overall performance. This can be done by comparing the results of the basic SmoothNet with the motion-aware SmoothNet.
Another potential ablation study is to investigate the impact of the long-range temporal receptive fields on the performance of the SmoothNet. This can be done by varying the length of the input sequence and analyzing the effect on the results.
Additionally, we can also analyze the effect of the loss function on the performance of the SmoothNet. This can be done by comparing the results of the SmoothNet trained with different loss functions, such as the mean per joint position error (MPJPE) and the mean per joint acceleration error (Accel).
Finally, we can also investigate the transferability of the SmoothNet across different datasets and estimators. This can be done by training the SmoothNet on one dataset

and testing it on another dataset, or by using different estimators as input to the SmoothNet.

Figure 14: The reasoning of LM-PLANNER with Llama 3.1 405B in generating the ablation plans for SmoothNet.

Image Processing GNN: Breaking Rigidity in Super-Resolution (Tian et al., 2024b). Image Processing GNN (IPG) proposes a graph-based model for image super-resolution reconstruction that aims to break rigidity that dominates previous super-resolution methods, by leveraging degree flexibility in graphs, pixel node sets instead of patch nodes and local and global information. The paper evaluates its claims using four ground-truth ablations:

1. Replacing pixel-level nodes with patch-based nodes.
2. Replacing degree-flexible graphs with KNN graphs.
3. Removing the local graph construction.
4. Removing the global graph construction.

Qwen demonstrates strong alignment with the paper's method (Figure 15), explicitly identifying all three key components and successfully recovering all four GT ablations. Since the planner is prompted to produce up to five ablations, Qwen also proposes an additional ablation on auxiliary components, such as the detail-rich indicator and relative position encoding. While these go beyond the GT set, they remain relevant to the method's internal mechanisms and coherent with the research. Overall, Qwen generates a well-scoped ablation plan that directly attributes performance to the primary components responsible for breaking rigidity. Llama, in contrast, exhibits a higher-level and less detailed reasoning trace (Figure 16). It correctly identifies the main components and proposes ablations broadly aligned with the GT, particularly for degree flexibility, pixel versus patch nodes, and local versus global graphs. However, the ablations are described less concretely, and the model favors removal-type or conceptual ablations over precise experimental designs. As in prior cases, Llama occasionally suggests additional relevant modifications, such as graph aggregation and positional encoding, which are coherent with the research.

Reasoning - Qwen 3.5

The IPG model introduces several novel components that contribute to "breaking rigidity" in super-resolution. The main innovations are:
1. Degree Flexibility: Using a detail-rich indicator (D_F) to assign different node degrees to pixels based on their reconstruction needs.
2. Pixel-level Nodes: Using pixels as graph nodes instead of patches to avoid misalignment issues.
3. Dual-scale Graph Construction: Using both local and global sampling strategies to construct graphs at different scales.
Based on these key components, I suggest the following ablation studies:
1. Test the importance of degree flexibility by comparing against fixed degree for all nodes
2. Validate the benefit of pixel-level nodes vs patch-level nodes
3. Evaluate the contribution of dual-scale graphs vs single-scale
4. Examine the effectiveness of the detail-rich indicator metric
5. Test the importance of relative position encoding
I'll prioritize the most critical ones that directly validate the paper's main claims about breaking rigidity.

Figure 15: The reasoning of LM-PLANNER with Qwen 3.5 in generating the ablation plans for IPG.

Reasoning - Llama 3.1 405B

The paper proposes a novel method for image super-resolution using a graph neural network (GNN) approach, called Image Processing GNN (IPG). The IPG model leverages the flexibility of graphs to break the rigidity of traditional super-resolution methods. The authors identify three types of rigidity in existing methods: degree-rigidity, patch-rigidity, and space-rigidity. To address these limitations, the IPG model uses a degree-flexible graph solution, pixel nodes instead of patches, and local and global sampling strategies.
To further investigate the effectiveness of the IPG model, I suggest the following ablation studies:
1. **Degree flexibility**: Remove the degree flexibility component and use a fixed degree for all nodes. This will help to understand the impact of degree flexibility on the model's performance.
2. **Pixel nodes vs. patch nodes**: Replace pixel nodes with patch nodes to see how it affects the model's performance. This will help to understand the importance of using pixel nodes in the IPG model.
3. **Local and global sampling**: Remove the local sampling strategy and only use global sampling. This will help to understand the impact of local sampling on the model's performance.
4. **Graph aggregation**: Replace the edge-conditioned aggregation with a different aggregation method, such as max-pooling. This will help to understand the impact of the aggregation method on the model's performance.
5. **Relative position encoding**: Remove the relative position encoding component and see how it affects the model's performance. This will help to understand the importance of relative position encoding in the IPG model.
These ablation studies will help to understand the contribution of each component to the IPG model's perfor-

mance and provide insights for future improvements.

Figure 16: The reasoning of LM-PLANNER with Llama 3.1 405B in generating the ablation plans for IPG.

Delving into Deep Imbalanced Regression (Yang et al., 2021). This paper proposes two complementary techniques for imbalanced regression: Label Distribution Smoothing (LDS) and Feature Distribution Smoothing (FDS), which exploit the similarity between nearby targets in both label and feature space. The paper includes four GT ablation studies that focus on design choices within these components:

1. Replacing the kernel type used in LDS and FDS (e.g., Gaussian, Laplacian, Triangular).
2. Replacing the regression loss function used jointly with LDS and FDS.
3. Varying kernel hyperparameters (kernel size and standard deviation).
4. Testing robustness under different skewed label distributions.

Both models correctly identify LDS and FDS as the core contributions of the paper, and they recover the kernel-type replacement ablation present in the GT. This shows that both models can capture the primary components and the corresponding ablation space. Beyond this, the models diverge in their approach: Qwen (Figure 17) proposes additional high-level removal or combination ablations, isolating LDS or FDS, which are relevant for analyzing component importance but do not appear in the GT. Llama (Figure 18), in contrast, produces a less structured reasoning trace and frequently combines component identification with ablation suggestions. Its proposed ablations include removal-type studies, baseline comparisons, and alternative calibration methods that fall outside the intended scope of ablation studies, reflecting broader evaluation ideas rather than component-level analysis.

Reasoning - Qwen 3.5

The paper introduces two main components to handle imbalanced regression:
1. Label Distribution Smoothing (LDS) - which uses kernel density estimation to learn the effective imbalance in datasets with continuous targets
2. Feature Distribution Smoothing (FDS) - which performs distribution smoothing on the feature space to calibrate potentially biased feature estimates
Based on these components, I'll suggest ablation studies focusing on:
1. The impact of kernel choice in LDS since it's fundamental to how the label distribution is smoothed
2. The effectiveness of FDS's momentum update mechanism
3. The necessity of feature calibration in FDS
4. The importance of bin size selection for grouping features
5. The combined effect of LDS and FDS versus using them individually
These ablations will help understand which components contribute most to the method's performance and how sensitive it is to different design choices.

Figure 17: The reasoning of LM-PLANNER with Qwen 3.5 in generating the ablation plans for Delving into Deep Imbalanced Regression.

Reasoning - Llama 3.1 405B

The paper "Delving into Deep Imbalanced Regression" presents a novel approach to handling imbalanced data in regression tasks. The authors propose two techniques: Label Distribution Smoothing (LDS) and Feature Distribution Smoothing (FDS). LDS involves convolving a symmetric kernel with the empirical label density to estimate the effective label density distribution, while FDS performs distribution smoothing on the feature space to calibrate the potentially biased estimates of feature distribution.
To further investigate the effectiveness of these techniques, I suggest the following ablation studies:
1. **Remove LDS**: Train the model without LDS to see how much it contributes to the overall performance. This will help understand the impact of LDS on the model's ability to handle imbalanced data.
2. **Replace FDS with a different feature calibration method**: Replace FDS with a different feature calibration method, such as batch normalization or instance normalization, to see if FDS is the most effective method for feature calibration.
3. **Add a different kernel to LDS**: Add a different kernel, such as a Gaussian or Laplacian kernel, to LDS to see if the choice of kernel affects the performance of the model.
4. **Apply FDS to a different layer**: Apply FDS to a different layer of the network, such as the input layer or a hidden layer, to see if the location of FDS affects the performance of the model.
5. **Compare with other imbalanced regression methods**: Compare the performance of the proposed method with other imbalanced regression methods, such as SMOTE or cost-sensitive learning, to see if the proposed method is more effective.
These ablation studies will help understand the contributions of each component of the proposed method and provide insights into how to further improve its performance.

Figure 18: The reasoning of LM-PLANNER with Llama 3.1 405B in generating the ablation plans for Delving into Deep Imbalanced Regression.

Neural-Pull: Learning Signed Distance Functions from Point Clouds by Learning to Pull Space onto Surfaces (Ma et al., 2021). Neural-

Pull proposes learning signed distance functions directly from point clouds by iteratively pulling query locations toward the surface using predicted SDF values and gradients. The paper evaluates the method using three GT ablations that target core design choices:

1. Removing the geometric network initialization (GNI).
2. Replacing the query location sampling strategy with random sampling over the full 3D space.
3. Adding a gradient constraint that enforces unit-norm gradients.

Both models correctly identify the main components of Neural-Pull, including the pulling operation, query location sampling strategy, and geometric network initialization, and they recover several of the GT ablations, such as the GNI removal and space-sampling replacement. Beyond these shared observations, the models differ in their handling of gradient-related ablations: Qwen (Figure 19) fails to recover the gradient-constraint ablation exactly as in the GT, instead proposing alternative optimization-focused ablations such as modifying the loss function or gradient computation, which remain relevant and coherent. Llama (Figure 20) partially overlaps with the GT on gradient and sampling ablations but reframes the gradient-constraint ablation as removing gradient information entirely. As in previous cases, Llama's proposals are generally higher-level, less structured, and focus on removal-type ablations.

Reasoning - Qwen 3.5

Based on the paper, Neural-Pull's main contribution is learning SDFs directly from point clouds by pulling query locations to their closest points on the surface. The key components are:
1. The pulling operation that uses both predicted signed distance values and gradients 2. The adaptive sampling strategy for query locations using isotropic Gaussian functions 3. The geometric network initialization (GNI) to approximate sphere SDFs 4. The loss function based on minimizing distance between pulled locations and nearest neighbors
I'll suggest ablations that investigate these key components:
1. Test the importance of the pulling operation by comparing with direct distance prediction 2. Evaluate different query location sampling strategies 3. Assess the impact of the geometric network initialization 4. Examine alternative loss formulations 5. Test different gradient computation approaches
I'll rank these by importance based on their potential impact on the method's performance.

Figure 19: The reasoning of LM-PLANNER with Qwen 3.5 in generating the ablation plans for Neural-Pull.

Reasoning - Llama 3.1 405B

The paper "NeuralPull: Learning Signed Distance Functions from Point Clouds by Learning to Pull Space onto Surfaces" presents a novel approach to learning signed distance functions (SDFs) from raw point clouds without requiring ground truth signed distance values. The method, called Neural-Pull, trains a neural network to pull query 3D locations to their closest points on the surface using the predicted signed distance values and the gradient at the query locations. To investigate the effectiveness of Neural-Pull, we suggest the following ablation studies:
1. **Removing the gradient component**: We will remove the gradient component from the pulling operation and only use the predicted signed distance values to move the query locations. This will help us understand the importance of the gradient information in the Neural-Pull method.
2. **Replacing the pulling operation with a different transformation**: We will replace the pulling operation with a different transformation, such as a simple translation or scaling, to see if the Neural-Pull method is sensitive to the choice of transformation.
3. **Adding noise to the point cloud**: We will add noise to the point cloud and evaluate the performance of Neural-Pull. This will help us understand the robustness of the method to noisy data.
4. **Using a different loss function**: We will replace the mean squared error loss function used in Neural-Pull with a different loss function, such as mean absolute error or cross-entropy loss, to see if the choice of loss function affects the performance of the method.
5. **Evaluating the effect of query location sampling**: We will evaluate the effect of different query location sampling strategies on the performance of Neural-Pull. This will help us understand the importance of the sampling strategy in the method.

Figure 20: The reasoning of LM-PLANNER with Llama 3.1 405B in generating the ablation plans for Neural-Pull.

Preserving Commonsense Knowledge from Pre-trained Language Models via Causal Inference (Zheng et al., 2023a). The paper proposes Causal Effect Tuning (CET) to preserve pre-trained models knowledge during fine-tuning and avoid overfitting. The GT includes a single ablation study focused on the KNN-based sample selection and search method, replaced with alternative selection strategies and similarity metrics, evaluated on average accuracy over commonsense QA datasets.

Both models correctly identify the main components of the method, including the KNN module, the causal framework, similarity threshold, and weighting scheme. They also recover aspects of the GT ablation space to some extent, particularly the

KNN-based interventions. Beyond this shared observation, Qwen (Figure 21) consistently proposes five ablations, even when the space of meaningful GT ablations is smaller, and tends to focus on fine-grained parameter tuning (e.g., the number of neighbors or the similarity threshold). These ablations are coherent and informative but do not fully target the methodologically critical component of replacing KNN for causal effect estimation. Llama (Figure 22) exhibits similar patterns as in previous papers, combining component identification with ablation proposals, with its suggestions being more naive (e.g., uniform estimation replacing KNN) and generally emphasizing hyperparameter-type changes rather than ablations that test the method's core functionality. As in previous papers, Llama produces ideas that are relevant to the field but not strictly aligned with the GT or classical ablation study goals.

Reasoning - Qwen 3.5

Based on the paper, the main components of the method are:
1. The causal graph framework for fine-tuning that introduces colliding effects between pre-trained data and target data.
2. The K-Nearest Neighbors (KNN) approach to estimate causal effects, where neighbors are selected based on similarity of gold answers.
3. The unified objective that combines vanilla fine-tuning with causal effects preservation.
4. The similarity threshold Θ that controls which samples get KNN treatment.
5. The weighting parameter W_0 that balances between anchor sample and its neighbors.
I'll suggest ablation studies focusing on these key components, ranked by importance:
1. First, we should verify the effectiveness of the causal framework by removing it completely.
2. The KNN approach and similarity threshold are critical design choices that should be tested.
3. The weighting scheme between anchor and neighbors is important for balancing knowledge preservation.
4. Alternative similarity metrics for finding neighbors could be explored.
The metrics used in the paper are primarily accuracy on commonsense QA datasets, so we'll use those.

Figure 21: The reasoning of LM-PLANNER with Qwen 3.5 in generating the ablation plans for Preserving Commonsense Knowledge from Pre-trained Language Models via Causal Inference.

Reasoning - Llama 3.1 405B

To design ablation studies for the research titled "Preserving Commonsense Knowledge from Pre-trained Language Models via Causal Inference," we need to identify the key components of the proposed method, Causal Effect Tuning (CET), and analyze their impact on the model's performance. Here are five ablation study suggestions, ranked by their importance:
1. **Ablation of the causal objective**: Remove the causal objective from the overall objective function and fine-tune the model using only the vanilla fine-tuning objective. This ablation study will help us understand the contribution of the causal objective in preserving commonsense knowledge.
2. **Ablation of the KNN-based colliding effect estimation**: Replace the KNN-based colliding effect estimation with a random or uniform estimation method. This ablation study will help us understand the importance of using KNNs to estimate the colliding effect.
3. **Ablation of the similarity threshold**: Remove the similarity threshold (Θ) and use a fixed number of nearest neighbors for all samples. This ablation study will help us understand the impact of the similarity threshold on the model's performance.
4. **Ablation of the weight controlling the strength of colliding effects (W0)**: Set W0 to 0 or 1 and analyze the impact on the model's performance. This ablation study will help us understand the importance of controlling the strength of colliding effects.
5. **Ablation of the heuristic approximation for estimating the causal effect on commonsense QA**: Replace the heuristic approximation with a different method, such as using the euclidean distance or cosine similarity between the average last hidden states of PLMs. This ablation study will help us understand the effectiveness of the proposed heuristic approximation.

Figure 22: The reasoning of LM-PLANNER with Llama 3.1 405B in generating the ablation plans for Preserving Commonsense Knowledge from Pre-trained Language Models via Causal Inference.

C.2 ReviewerAblation

Models perform better on rejected papers. We test whether planner recall correlates with paper acceptance. Most models achieve higher recall on rejected papers (Table 12). This is expected: lower-quality papers often lack rigor and validation, making missing ablations easier to identify.

Results by year. Table 11 compares recall@2 on submissions from before 2025 and from 2025. Results are similar across both splits and show no consistent trend across models. This suggests that model performance is not strongly affected by the submission year, indicating generalization beyond the training data.

Results per acceptance status. Table 12 reports recall@2 by acceptance status. Most models perform better on rejected papers than accepted ones. This suggests performance correlates with paper quality, and that proposing ablations is easier for lower-quality papers.

Year	2023-2024	2025
LM-PLANNER		
w/ GPT-5.4	0.33	0.39
w/ Llama 3.1 405B Instruct	0.35	0.30
w/ o3-mini	0.21	0.26
w/ Gemini 2.5 Flash	0.30	0.30
w/ GPT OSS 120B	0.30	0.32
w/ MiniMax M2.5	0.33	0.32
w/ Qwen 3.5 122B	0.27	0.31
AGENT-PLANNER		
w/ GPT-5.4	0.38	0.33
w/ Llama 3.1 405B Instruct	0.29	0.29
w/ GPT OSS 120B	0.25	0.25
w/ MiniMax M2.5	0.29	0.29
w/ Qwen 3.5 122B	0.30	0.29

Table 11: Macro-average recall@2 results of the two planner scaffoldings for the ReviewerAblation split by year.

Decision	Accept	Reject
LM-PLANNER		
w/ GPT-5.4	0.32	0.39
w/ Llama 3.1 405B Instruct	0.31	0.33
w/ o3-mini	0.25	0.23
w/ Gemini 2.5 Flash	0.21	0.34
w/ GPT OSS 120B	0.26	0.33
w/ MiniMax M2.5	0.31	0.33
w/ Qwen 3.5 122B	0.28	0.30
AGENT-PLANNER		
w/ GPT-5.4	0.27	0.39
w/ Llama 3.1 405B Instruct	0.27	0.30
w/ GPT OSS 120B	0.21	0.27
w/ MiniMax M2.5	0.21	0.32
w/ Qwen 3.5 122B	0.28	0.30

Table 12: Macro-average recall@2 results of the two planner scaffoldings for the ReviewerAblation split by decision.

C.2.1 Qualitative Analysis

In the following, we qualitatively analyze the LM-PLANNER using Llama 3.1 and Qwen 3.5, on five randomly selected papers. This analysis examines the models' reasoning traces to understand how they find missing ablations, and which types of ablations they propose.

Boosting Membership Inference Attacks with Upstream Modification (Abdullah, 2025). The paper improves Membership Inference Attacks (MIAs) by modifying dataset partitioning and focusing attacks on outliers, achieving higher true positive rates and near-perfect AUC. No formal ablation studies are included. Llama correctly identifies this gap (Figure 23) and proposes two high-level ablations: isolating the effect of reduced drop rate and evaluating the impact of using only outliers, effectively targeting the main method components, although not appearing in any review. Qwen, in contrast, produces a structured reasoning trace (Figure 24) that first identifies the method's components and then incorrectly treats parameter sweeps (e.g., drop rates, number of shadow models, outlier detection settings) as ablations. Subsequently, it proposes more granular ablations focusing on hyperparameters, rather than the main modifications that determine the attack's effectiveness. Qwen's fine-grained analysis, while coherent, fails to capture the essence of ablation studies.

Reasoning - Llama 3.1 405B

The paper "Boosting Membership Inference Attacks with Upstream Modification" presents a novel approach to improve the accuracy of membership inference attacks (MIAs) on machine learning models. The authors identify two limitations in the existing MIA framework: sampling bias and attack aggregation. They propose modifications to address these limitations, which involve reducing the drop rate during dataset partitioning and running the MIA against only the outliers.
The authors demonstrate the effectiveness of their modifications through experiments on various datasets, including CIFAR-10, CIFAR-100, and Tiny-Imagenet. The results show that their modified attacks outperform the original ones in every case, achieving near-perfect AUC and TPR at zero FPR.
However, the paper lacks ablation studies to further investigate the impact of each modification on the attack success. Specifically, it would be interesting to see how the attack performance changes when only one of the modifications is applied.
To address this, I suggest two missing ablation studies:
1. **Ablation A:** Evaluate the impact of reducing the drop rate during dataset partitioning on the attack success, without modifying the attack aggregation step. This will help understand the contribution of the sampling bias modification to the overall improvement in attack accuracy.
2. **Ablation B:** Evaluate the impact of using only the outliers for the MIA, without modifying the sampling bias. This will help understand the contribution of the attack aggregation modification to the overall improvement in attack accuracy.
These ablation studies will provide a more detailed understanding of the effects of each modification and help identify potential areas for further improvement.

Figure 23: The reasoning of LM-PLANNER with Llama 3.1 405B in generating missing ablations for Boosting Membership Inference Attacks with Upstream Modification.

Reasoning - Qwen 3.5

The paper proposes modifications to the upstream stages of Membership Inference Attacks (MIAs), specifically focusing on two main components:
1. Sampling bias modification (Step 2): They reduce the drop rate during dataset partitioning from 50

2. Attack aggregation modification (Step 4): They use outlier detection methods (specifically ApB - Accuracy per Batch) to identify vulnerable points instead of using the entire dataset.
The paper already includes ablations for: - Different drop rates (50%, 40%, 30%, 20%, 10%)
- Different outlier detection methods (SPP, MD, ApB)
- Different number of shadow models (1 vs 100)
However, there are two important missing ablations:
1. The effect of the number of vulnerable points selected: The paper fixes this to 5,000 points but doesn't ablate this hyperparameter. This is crucial since it directly affects both the computational cost and attack effectiveness.
2. The robustness of the ApB outlier detection method: While ApB performs best, the paper doesn't ablate its key parameter - the frequency of accuracy measurement (currently done after each batch). This is important since it affects both computational overhead and outlier detection accuracy.
These ablations would provide important insights into the method's sensitivity to these key parameters.

Figure 24: The reasoning of LM-PLANNER with Qwen 3.5 in generating missing ablations for Boosting Membership Inference Attacks with Upstream Modification.

Accelerating Retrieval-augmented Language Model Serving with Speculation (Zhang et al., 2024). RaLMSpec accelerates retrieval-augmented language model serving using caching, batched verification, and speculation stride optimization. Both models miss an ablation that already appears in the paper (cache size variation). Llama's reasoning (Figure 25) describes the paper but ignores the existing ablations and proposes two experiments, one of which coincides with an existing ablation and the other aligns with a reviewer suggestion. Qwen provides a less structured reasoning (Figure 26), noting only that extensive experiments were performed. Its suggested ablations are sensible but either duplicate experiments already in the paper (e.g., cache size variations) or focus on parameter changes rather than altering components to yield insight into the method's behavior.

Reasoning - Llama 3.1 405B

The paper "Accelerating Retrieval-augmented Language Model Serving with Speculation" proposes a speculation-inspired framework called RaLMSpec to accelerate the serving of retrieval-augmented language models (RaLM). RaLMSpec uses speculative retrieval and batched verification to reduce the serving latency of iterative RaLM approaches while preserving the same model outputs. The framework also incorporates prefetching, optimal speculation stride scheduler, and asynchronous verification to further boost the speculation performance.
The paper presents extensive evaluations of RaLMSpec over three language models on four downstream QA datasets, demonstrating that RaLMSpec can achieve a speed-up ratio of 1.75-2.39×, 1.04-1.39×, and 1.31-1.77× when the retriever is an exact dense retriever, approximate dense retriever, and sparse retriever, respectively, compared with the baseline.
To further investigate the effectiveness of RaLMSpec, we suggest two missing ablation studies:
1. **Ablation study on the impact of speculation stride on the performance of RaLMSpec**: The paper mentions that the speculation stride is a crucial parameter that affects the trade-off between speculation overhead and latency saving. However, it does not provide a detailed analysis of how different speculation strides affect the performance of RaLMSpec. An ablation study on the impact of speculation stride can provide valuable insights into the optimal speculation stride for different scenarios.
2. **Ablation study on the effect of cache size on the performance of RaLMSpec**: The paper mentions that RaLMSpec uses a local cache to store retrieved documents, but it does not investigate the effect of cache size on the performance of RaLMSpec. An ablation study on the effect of cache size can help understand the optimal cache size for different scenarios and provide insights into the trade-off between cache size and performance.

Figure 25: The reasoning of LM-PLANNER with Llama 3.1 405B in generating missing ablations for Accelerating Retrieval-augmented Language Model Serving with Speculation.

Reasoning - Qwen 3.5

After analyzing the paper, I notice that while the authors conducted extensive experiments to validate their method, there are some important ablations missing that could provide deeper insights into the effectiveness of RaLMSpec. Here are two key missing ablations:
1. Cache Size Impact: While the paper uses a local cache for speculative retrieval, there is no systematic study of how the cache size affects performance. The authors only test prefetch sizes of 20 and 256 documents, but don't explore the fundamental trade-off between cache size, hit rate, and memory overhead. This is crucial since the cache is a core component of their speculation mechanism.
2. Verification Batch Size: The paper relies heavily on batched verification but doesn't analyze how different batch sizes affect the overall performance across different retrievers. While they show that batched retrieval is more efficient than sequential retrieval, they don't explore the optimal batch size for different scenarios. This is important since the batch size directly impacts both latency and resource utilization.
These ablations would help better understand the method's sensitivity to key hyperparameters and provide practical guidance for deployment.
The metrics used in the paper for evaluating performance are primarily speed-up ratio and latency (in seconds), so I'll include these in the suggested ablations.

Figure 26: The reasoning of LM-PLANNER with Qwen 3.5 in generating missing ablations for Accelerating Retrieval-augmented Language Model Serving with Speculation.

Unpacking SDXL Turbo: Interpreting Text-to-Image Models with Sparse Autoencoders (Surkov et al., 2024). The paper studies SDXL Turbo's internal representations using sparse autoencoders (SAEs) to learn interpretable features. Both models struggle at identifying the ablations already presented in the paper, as the paper does not present them in a dedicated form. Llama's reasoning (Figure 27) describes the general method and then proposes two ablations: one aligns with reviewer suggestions and the other explores hyperparameters of the SAE, which could be interpreted as an ablation. Qwen (Figure 28), in contrast, identifies various analyses in the paper but mistakenly treats them as ablations and then suggests additional hyperparameter-focused experiments rather than altering components to produce insights about the method.

Reasoning - Llama 3.1 405B

The paper "Unpacking SDXL Turbo: Interpreting Text-to-Image Models with Sparse Autoencoders" presents a study on the interpretability of text-to-image models, specifically the SDXL Turbo model. The authors use sparse autoencoders (SAEs) to learn interpretable features from the intermediate representations of the SDXL Turbo model. They find that the learned features are highly interpretable, causally influence the generation process, and reveal specialization among the blocks of the model.
The authors also present a case study on the most active features for a given prompt and find that the features learned by the SAEs correspond to specific objects, textures, and styles in the generated images. They also find that the features learned by the SAEs can be used to manipulate the generated images in a controlled manner.
The paper also discusses the limitations of the study, including the fact that the SAEs are trained on a limited dataset and that the features learned by the SAEs may not be universally applicable.
Overall, the paper presents a promising approach to interpreting text-to-image models and demonstrates the potential of SAEs in revealing the internal workings of these models.
Based on the paper, I suggest two missing ablation studies:
1. Ablating the number of layers in the SAEs: The authors use a fixed number of layers in the SAEs, but it would be interesting to see how the performance of the SAEs changes when the number of layers is increased or decreased.
2. Ablating the type of activation function used in the SAEs: The authors use the ReLU activation function in the SAEs, but it would be interesting to see how the performance of the SAEs changes when different activation functions are used.
These ablation studies could provide further insights into the workings of the SAEs and the SDXL Turbo model, and could potentially lead to improvements in the performance of the SAEs.

Figure 27: The reasoning of LM-PLANNER with Llama 3.1 405B in generating missing ablations for Unpacking SDXL Turbo: Interpreting Text-to-Image Models with Sparse Autoencoders.

Reasoning - Qwen 3.5

The paper investigates using Sparse Autoencoders (SAEs) to interpret SDXL Turbo's internal representations. The authors train SAEs on updates performed by transformer blocks within the model's denoising U-net and analyze the learned features.
The paper already includes several analyses:
1. Qualitative analysis of learned features through visualization
2. Quantitative analysis of feature interpretability using CLIP similarity
3. Causality analysis comparing empty prompt interventions with feature captions
4. Sensitivity analysis checking if features activate on relevant contexts
5. Analysis of texture and color properties of different transformer blocks
However, I notice two important missing ablation studies:
1. The choice of which intermediate representations to analyze - the authors chose to train SAEs on the updates performed by transformer blocks, but did not justify this choice compared to alternatives like training on the full residual stream or individual attention heads.
2. The architecture of the SAE itself - while they followed standard practices, they did not ablate key architectural choices like the number of features (nf) or sparsity level (k) to demonstrate their impact on feature interpretability and specificity.
These ablations would help validate the key methodological choices and strengthen the paper's conclusions.

Figure 28: The reasoning of LM-PLANNER with Qwen 3.5 in generating missing ablations for Unpacking SDXL Turbo: Interpreting Text-to-Image Models with Sparse Autoencoders.

Growth Inhibitors for Suppressing Inappropriate Image Concepts in Diffusion Models (Chen et al., 2025). The paper introduces GIE, a method for eliminating NSFW content in diffusion models by injecting reweighted "growth inhibitors" into attention maps, guided by a suppression-scale adapter. Llama's reasoning (Figure 29) does not analyze the ablations already performed in the paper, which makes it easy for the model to generate ablation ideas from scratch; in some cases, these suggestions hit missing ablations that the reviewers also suggested. Qwen (Figure 30), in contrast, correctly identifies the experiments already performed by the authors, although these are not explicitly labeled as ablations, and additionally proposes two missing ablations from the experimental section that are genuinely relevant, one of them also appears in the reviewer suggestion.

Reasoning - Llama 3.1 405B

The paper "Growth Inhibitors for Suppressing Inappropriate Image Concepts in Diffusion Models" presents a novel approach to suppressing inappropriate features in the image space using growth inhibitors without fine-tuning. The method, called GIE, injects growth inhibitors into the attention map group of the prompt to guide inappropriate features into appropriate ones during the diffusion process. The paper also proposes a scheme for training an adapter to infer the suppression scale of GIE based on the intermediate values of the cross-attention layers.
The paper evaluates the performance of GIE against several baselines for concept erasure in diffusion models and demonstrates its effectiveness in erasing NSFW content, styles, and specific common objects with little effect on unrelated concepts. The paper also shows that GIE preserves the quality and semantics of the generated images.
To further investigate the effectiveness of GIE, we suggest two missing ablation studies:
1. **Ablation of the adapter**: The paper proposes using an adapter to infer the suppression scale of GIE. However, it would be interesting to see how GIE performs without the adapter, using a fixed suppression scale instead. This ablation study would help to understand the importance of the adapter in the GIE framework.
2. **Ablation of the attention map group injection position**: The paper injects the growth inhibitor into the attention map group at the position preceding m[EOT]. However, it would be interesting to see how GIE performs when injecting the growth inhibitor at different positions in the attention map group. This ablation study would help to understand the importance of the injection position in the GIE framework.
These ablation studies would provide further insights into the GIE framework and help to identify potential areas for improvement.

Figure 29: The reasoning of LM-PLANNER with Llama 3.1 405B in generating missing ablations for Growth Inhibitors for Suppressing Inappropriate Image Concepts in Diffusion Models.

Reasoning - Qwen 3.5

After analyzing the paper, I notice that the authors have conducted several ablation studies, including: 1. Testing different positions for injecting growth inhibitors (Appendix B)
2. Comparing fixed suppression scales vs adaptive scales via the adapter (Appendix C)
3. Testing impact of prompt lengths (Appendix E)
4. Testing effectiveness across different Stable Diffusion models (Appendix G)
However, there are two critical missing ablation studies that would help better understand the method's key components:
1. The most important missing ablation is on the feature extraction process. The authors extract features from the attention maps corresponding to target tokens, but don't validate if this is the optimal approach. Alternative feature extraction methods could potentially improve performance.
2. The second important missing ablation is on the adapter architecture. The authors use a 2-layer MLP with specific dimensions (64->32), but don't justify this choice. Testing different architectures could reveal if this is optimal.
These ablations would help understand the contribution of key design choices to the method's performance.

Figure 30: The reasoning of LM-PLANNER with Qwen 3.5 in generating missing ablations for Growth Inhibitors for Suppressing Inappropriate Image Concepts in Diffusion Models.

UniPAD: A Universal Pre-training Paradigm for Autonomous Driving (Yang et al., 2024a). UniPAD introduces a self-supervised learning paradigm applying 3D volumetric differentiable rendering in autonomous driving, to facilitating the reconstruction of continuous 3D shape structures and the intricate appearance characteristics of their 2D projections. Llama's reasoning (Figure 31) ignores the ablations already present in the paper, continuing the trend from previous papers where it starts from scratch. This makes it easy for the model to suggest potentially meaningful ablations. Its reasoning is not fully coherent: for example, it proposes replacing components in the reasoning trace but then suggests removing them in the actual ablation plan. Qwen (Figure 32) correctly identifies the ablations performed in the paper but then focuses on low-level hyperparameter variations, such as voxel resolution and loss weighting, rather than ablations that test major methodological components; these experiments are feasible but do not provide high-level insights about the method.

Reasoning - Llama 3.1 405B

The paper "UniPAD: A Universal Pre-training Paradigm for Autonomous Driving" presents a novel self-supervised learning paradigm for 3D point cloud data. The method uses 3D differentiable rendering to reconstruct the missing geometry on the projected 2D depth image. The paper conducts extensive experiments on various downstream 3D tasks and achieves state-of-the-art results.
Upon reviewing the paper, I noticed that the authors did not conduct ablation studies on the effect of the 3D differentiable rendering module and the impact of the masking strategy on the performance of the model. Therefore, I suggest two missing ablation studies:
1. Ablation study on the 3D differentiable rendering module: This study would involve removing or replacing the 3D differentiable rendering module with a different rendering method to assess its impact on the model's performance.
2. Ablation study on the masking strategy: This study would involve varying the masking ratio or using different masking strategies to assess their impact on the model's performance.
These ablation studies would provide valuable insights into the effectiveness of the 3D differentiable render-

ing module and the masking strategy, which are crucial components of the UniPAD paradigm.

Figure 31: The reasoning of LM-PLANNER with Llama 3.1 405B in generating missing ablations for UniPAD: A Universal Pre-training Paradigm for Autonomous Driving.

Reasoning - Qwen 3.5

After analyzing the paper, I notice that the authors have conducted several ablation studies, including: 1. Masking ratio experiments
2. Decoder depth and width studies
3. Different rendering techniques comparison
4. Ray sampling strategies evaluation
5. Feature projection analysis
6. Pre-trained components analysis
However, there are two critical components that lack proper ablation studies:
1. The volumetric representation resolution (X×Y×Z): The paper uses a fixed voxel resolution of 180×180×5, but doesn't investigate how this choice affects performance. This is crucial since the voxel resolution directly impacts both the computational cost and the representation quality.
2. The rendering loss components: The paper uses a combination of RGB and depth losses (Eq. 4), but doesn't investigate the relative importance of each loss term or explore different loss weightings (λ_{RGB} and λ_{depth}). Understanding the contribution of each loss component is essential for the method's effectiveness.
These missing ablations would help better understand the method's sensitivity to these important design choices and potentially improve its performance.

Figure 32: The reasoning of LM-PLANNER with Qwen 3.5 in generating missing ablations for UniPAD: A Universal Pre-training Paradigm for Autonomous Driving.

C.3 Human Evaluation

This section describes the instructions and processing of the human evaluation results in Section 7.5.

To conduct the evaluation, we begin by randomly selecting 15 test-set papers, 5 from each broad category: Computer Vision, NLP, and General Machine Learning, and truncating each to include all sections up to and including the method section. We recruited 15 participants currently enrolled in a PhD in machine learning. Each participant is given a questionnaire, where he/she chooses one paper based on their expertise and generated an ablation plan following AuthorAblation's formulation (Figure 33) using the truncated paper. Participants were instructed to spend at least one hour on the task and explicitly not to use any generative AI tools to ensure a fair comparison.

We manually inspect all submissions and exclude 5 responses where participants did not demonstrate a clear understanding of the task or the paper, for example by proposing general experiments instead of ablations or by leaving placeholder content in required fields. The remaining 10 submissions are evaluated against the GT ablations using the same matching and scoring criteria applied to model generations. We report the same evaluation metrics as in AuthorAblation and compare human performance to the best-performing model on this task, Qwen 3.5, evaluated on the same subset of papers.

The responses of all participants are supplied as part of the supplementary material. The papers that were part of the survey by category are given below.

Computer Vision:

- Real-Time High-Resolution Background Matting (Lin et al., 2020).
- SpiderMatch: 3D Shape Matching with Global Optimality and Geometric Consistency (Roetzer and Bernard, 2024).
- Learning High Fidelity Depths of Dressed Humans by Watching Social Media Dance Videos (Jafarian and Park, 2021).
- Good Visual Guidance Makes A Better Extractor: Hierarchical Visual Prefix for Multimodal Entity and Relation Extraction (Chen et al., 2022).
- Image Processing GNN: Breaking Rigidity in Super-Resolution (Tian et al., 2024b).

NLP:

- DisCoDisCo at the DISRPT2021 Shared Task: A System for Discourse Segmentation, Classification, and Connective Detection (Gessler et al., 2021).
- Preserving Commonsense Knowledge from Pre-trained Language Models via Causal Inference (Zheng et al., 2023a).
- Assisting in Writing Wikipedia-like Articles From Scratch with Large Language Models (Shao et al., 2024).
- Differentiable Prompt Makes Pre-trained Language Models Better Few-shot Learners (Zhang et al., 2022).

- Learning Performance-Improving Code Edits (Shypula et al., 2024).

General Machine Learning:

- Self-Composing Policies for Scalable Continual Reinforcement Learning (Malagón et al., 2025).
- APT: Adaptive Pruning and Tuning Pretrained Language Models for Efficient Training and Inference (Zhao et al., 2024).
- LCA-on-the-Line: Benchmarking Out-of-Distribution Generalization with Class Taxonomies (Shi et al., 2024).
- A Generalization of Transformer Networks to Graphs (Dwivedi and Bresson, 2021).

Human Evaluation - Questionnaire

We are building a new benchmark to evaluate the scientific research capabilities of language models (LLMs).
Our current focus is on asking LLM agents to plan the ablation study of a paper, given its method section.
We ask you to create ablation study plan given a paper's method section only.
Please follow the exact instructions to make our evaluation valid and reliable.
It should take about approximately 1 hour to participate - mostly by reading and understanding the paper.
Thanks for your help!

Question 1:
Choose your expertise domain:

1. Computer Vision.
2. NLP.
3. General Machine Learning.

Question 2:
Please choose one paper with the following rules:

- A paper the you are NOT familiar with.
- Do not search the paper online, use the provided link to get the paper PDF (which includes all the section up to and including the method section).

List of papers (please open the link now of the paper you choose):
[List of papers is presented here, per category]

Questions 3-7:
Your task now is to generate UP TO 5 ablations (ordered by relevance - from high to low) of the paper you have chosen.
Keep in mind that ablation studies aim to attribute the method's performance to its major components.
Please do NOT use any GenAI in your ablation studies plan.
Each ablation in the plan should contain:

1. **Ablated part/component:** A high-level description of the part or component of the method that you think should be ablated.
2. **Action:** The action to perform on the ablated part (one of: Remove, Replace, Add). If the action is Replace or Add, please include in the 'replacement' a list of all replacements/additions you think are relevant to test.
3. **Replacement:** Include this field only if the action is Replace or Add. Provide a list of all the replacements you think are relevant to test.
4. **Metrics:** A list of metrics to evaluate the ablation study.
5. **Rationale:** Why you think this ablation is relevant in the context of the paper?

After each ablation you add, you have the option to terminate the survey or to suggest another ablation.

Figure 33: The instructions for the human evaluation survey.

D Prompts

This section provides the prompt for all the LM systems in AblationBench including: the two planners and two judges of AuthorAblation and the two planners and two judges of ReviewerAblation.

D.1 AuthorAblation

D.1.1 LMJUDGE

System Prompt

SETTING: You are an autonomous computer science researcher, an expert at analyzing machine learning papers and their ablation studies.

You are given a matching task between two sets of ablations: those presented in a research paper and those suggested in an ablation plan, but you do not know which set corresponds to the paper and which corresponds to the plan. A match means: The experiment described by an ablation in side A is allowed or included as a possible option within an ablation in side B (and vice versa).

You need to format your output using two fields: discussion and predictions. The discussion field should contain a clear explanation of your reasoning, and the predictions field should contain the final output. The predictions field should be in a strict JSONL format, with each line containing a JSON object representing the final output.

Each JSON object should have the following fields:

1. "name_in_A": name(s) of the matching ablations in the ablations given in side A under `<ablations_in_A></ablations_in_A>`.

2. "name_in_B": name(s) of the matching ablations in the ablations given in side B under `<ablations_in_B></ablations_in_B>`.

Your output should always include _one_ discussion and _one_ predictions field EXACTLY as in the following example:

<discussion>
The ablations in side A contain three different ablations, and the ablations in side B contain four different ablations. We analyze each of them separately.

Side A:

1. Ablation A: This ablation is about the model architecture. The ablations in side B contain a similar ablation (Ablation X) with the same ablated part and action, so we consider it a match.

2. Ablation B: This ablation is about the training data. The ablations in side B contain a similar ablation with a different name (Ablation Y), but the ablated part, action, and replacement content match exactly, so we consider it a match.

3. Ablation C: This ablation is about the evaluation metric. The ablations in side B do not contain any similar ablation, so we cannot consider it a match.

Side B:

1. Ablation X: was matched to Ablation A from side A.

2. Ablation Y: was matched to Ablation B from side A.

3. Ablation Z: there is no matching ablation in side A.

4. Ablation W: there is no matching ablation in side A.
</discussion>

<predictions>
{"name_in_A": "Ablation A", "name_in_B": "Ablation X"}
{"name_in_A": "Ablation B", "name_in_B": "Ablation Y"}
{"name_in_A": "Ablation C", "name_in_B": null}
{"name_in_A": null, "name_in_B": "Ablation Z"}
{"name_in_A": null, "name_in_B": "Ablation W"}
</predictions>

Figure 34: The system prompt for LMJUDGE for AuthorAblation.

Instance Prompt

SETTING: We're currently reviewing an ablation studies plan for the research paper {{paper_title}}. Here's the research abstract:

ABSTRACT: {{abstract}}

INSTRUCTIONS: Below, you will find the ablations performed in the paper and the suggested ablation plan. We do not provide any information which ablations belong to which source, so you need to carefully analyze each ablation in both paper and plan and determine if there is a match. Your task is to find a match between ablations in side A and ablations in side B, and vice versa.

A match is considered valid only if all of the following are true:

1. The "ablated_part" (i.e., the component or mechanism being ablated) must refer to the same component.
2. The "action" in one ablation must be explicitly allowed by an ablation in the other side.
3. If the action is REPLACE or ADD, the "replacement" content must have at least one valid option match.

Examples:

- REMOVE X → REPLACE X with [remove, Y] MATCH
- REPLACE X with Y → REMOVE X NO MATCH
- REMOVE X → REMOVE X MATCH
- REPLACE X with [Y, Z] → REPLACE X with [Z, W] MATCH
- ADD Y to X → ADD [Y, Z] to X MATCH
- REPLACE (X+Y) with [X, Y, Z] → [REMOVE X, REMOVE Y] MATCH

Be conservative: if it is unclear whether one ablation allows the other ablation, do NOT match.

The match can contain multiple ablations from either side, as long as the match meets the criteria above. For example, if an ablation in side A is split into two ablations in side B, but both of them together meet the matching criteria, then it is still considered a valid match.

What You Need to Do:

Fill in the following fields for each entry:

{"name_in_A": [name of the ablation in side A OR a list of ablation names in side A OR null if there is no matching ablation for the ablation in side B], "name_in_B": [name of the matching ablation from side B OR a list of ablation names in side B OR null if there is no matching ablation for the ablation in side A]}

GENERAL IMPORTANT TIPS:

1. The "name_in_A" and "name_in_B" should match exactly to the ablation names in side A and side B, respectively.

2. You must go over ALL the ablations in side A and all ablations in side B, _each one of them should be in your prediction_.
3. It is OK if one ablation from side A is split into multiple ablations in side B, or merged several side A ablations into one — as long as the matching criteria (per ablation in side A) are met, the match is still valid. This holds to the other direction as well (side B to side A).

STRATEGY:

1. For each ablation in side A, you should decide based on the criteria above if there is a valid match in side B, and vice versa.
2. Please make sure to include in the discussion field the reasoning behind your decision for each ablation.
3. After that, you should have the information to fill in the "name_in_B" / "name_in_A" fields for each ablation in side A and side B, respectively. If the ablation is considered a match to one (or more) of the ablations in the other side, you should fill the appropriate ablation(s) in "name_in_B" / "name_in_A", otherwise it should remain null.

Here are the ablations in side A and side B:

<ablations_in_A> {{side_A}} </ablations_in_A>

<ablations_in_B> {{side_B}} </ablations_in_B>

Remember to include all the ablations in side A and side B in your predictions!

BELOW IS AN EXAMPLE THAT WILL HELP YOU UNDERSTAND THE FORMATTING AND THE TASK BETTER. PLEASE USE IT FOR YOUR REFERENCE ONLY AND DO NOT REPEAT IT IN YOUR FINAL ANSWER.

Example:

<ablations_in_A> {"name": "HybrIK Variant Comparison", "ablated_part": "Adaptive HybrIK mechanism", "action": "REPLACE", "replacement": ["Naive HybrIK (using original parent joints)", "Adaptive HybrIK (using reconstructed parent joints)", "Iterative global optimization (non-differentiable baseline)"], "metrics": ["MPJPE (mm)", "PVE (mm)", "Error accumulation along kinematic tree"]}
{"name": "Shape Parameter Study", "ablated_part": "SMPL shape parameter prediction", "action": "REPLACE", "replacement": ["Mean shape parameters", "PCA-based shape space reduction", "Direct vertex offset prediction"], "metrics": ["MPJPE (mm)", "PVE (mm)", "Shape error metrics"]}
{"name": "Twist-Swing Ablation", "ablated_part": "Twist-and-swing decomposition of rotations", "action": "REPLACE", "replacement": ["Direct rotation regression (3-DoF) as in previous works", "Only swing component (2-DoF)", "Only twist component (1-DoF)"], "metrics": ["MPJPE (mm)", "PVE (mm)", "Reconstruction error"]}
{"name": "3D Keypoint Estimation Study", "ablated_part": "3D keypoint estimation component", "action": "REPLACE", "replacement": ["Direct regression without volumetric representation", "Different heatmap resolutions", "Alternative backbone architectures"], "metrics": ["MPJPE (mm)", "PVE (mm)", "Per-joint accuracy"]}
{"name": "Twist Angle Prediction", "ablated_part": "Twist angle prediction network", "action": "REPLACE", "replacement": ["Fixed twist angles (0 degrees)", "Constrained range predictions", "Multi-head prediction network"], "metrics": ["MPJPE (mm)", "PVE (mm)", "Physical plausibility score"]}
</ablations_in_A>

<ablations_in_B> {"name": "Analysis of the twist rotation", "ablated_part": "Twist angles in the twist-and-swing decomposition", "action": "REPLACE", "replacement": ["Random values in [-π, π]", "Zero twist angles", "Network-estimated twist angles"], "metrics": ["Mean error of reconstructed 24 SMPL joints", "Mean error of 14 LSP joints", "Mean error of body mesh", "Mean error of twist angle"]}
{"name": "Error correction capability of HybrIK", "ablated_part": "Apply different algorithms on the predicted pose and compare it to HybrIK", "action": "REPLACE", "replacement": ["SMPLify"], "metrics": ["MPJPE"]}
{"name": "Effect of shape parameters (β)", "ablated_part": "Shape parameters (β) in SMPL model", "action": "REPLACE", "replacement": ["Ground-truth β", "Zero β"], "metrics": ["MPJPE", "PVE"]}
{"name": "Robustness of HybrIK to noise", "ablated_part": "Comparison between Naive HybrIK to Adaptive HybrIK with different input joints", "action": "REPLACE", "replacement": ["GT joints", "10mm noise", "20mm noise", "30mm noise"], "metrics": ["MPJPE"]}
</ablations_in_B>

<discussion>
Let's go over all ablations in side A and side B to determine the matches based on the criteria provided.

Side A:

1. HybrIK Variant Comparison: This ablation replaces Adaptive HybrIK mechanism with several variants. In side B, the ablation "Robustness of HybrIK to noise" deals with the adaptive HybrIK, it does replace it with one variant (GT joints) that matches side A variant (Naive HybrIK (using original parent joints)). Therefore, there is a match.

2. Shape Parameter Study: This ablation replaces SMPL shape parameter prediction with different methods. In side B, "Effect of shape parameters (β)" replaces shape parameters with ground-truth and zero β, which does not align with the replacement criteria. Therefore, no match.

3. Twist-Swing Ablation: This ablation replaces the twist-and-swing decomposition with other methods. In side B, "Analysis of the twist rotation" is the closest one, though it focuses on twist angles specifically. Therefore, no match.

4. 3D Keypoint Estimation Study: This ablation replaces the 3D keypoint estimation component with various alternatives. None of the ablations in side B address this component, so no match.

5. Twist angle prediction: This ablation replaces the twist angle prediction network with different methods. In side B, "Analysis of the twist rotation" again focuses on twist angles with one replacement (zero twist angles) that matches. Therefore, we have a match here.

Side B:

1. Analysis of the twist rotation: was matched to "Twist angle prediction" from side A.

2. Error correction capability of HybrIK: There is no matching ablation in side A, as none of the ablations in side A address error correction on the predicted pose.

3. Effect of shape parameters (β): There is no matching ablation in side A, the closest one is "Shape Parameter Study", but the replacements do not align.

4. Robustness of HybrIK to noise: was matched to "HybrIK Variant Comparison" from side A.
</discussion>

<predictions>

{"name_in_A": "HybrIK Variant Comparison", "name_in_B": "Robustness of HybrIK to noise"}
{"name_in_A": "Shape Parameter Study", "name_in_B": null}
{"name_in_A": "Twist-Swing Ablation", "name_in_B": null}
{"name_in_A": "3D Keypoint Estimation Study", "name_in_B": null}
{"name_in_A": "Twist Angle Prediction", "name_in_B": "Analysis of the twist rotation"}
{"name_in_A": null, "name_in_B": "Error correction capability of HybrIK"}
{"name_in_A": null, "name_in_B": "Effect of shape parameters (β)"}
</predictions>

Figure 35: The task instance prompt for LMJUDGE for AuthorAblation.

D.1.2 AGENTJUDGE

System Prompt

SETTING: You are an autonomous computer science researcher, and you're working directly in the command line of a Linux container with a special interface.

You are given a matching task between two sets of ablations: those presented in a research paper and those suggested in an ablation plan, but you do not know which set corresponds to the paper and which corresponds to the plan. A match means: The experiment described by ablation in side A is allowed or included as a possible option within ablation in side B (and vice versa).

The special interface consists of a file editor that shows you {{WINDOW}} lines of a file at a time. In addition to typical bash commands, you can also use specific commands to help you navigate within files. To call a command, you need to invoke it with a function call/tool call.

RESPONSE FORMAT: Your shell prompt is formatted as follows:

```
(Open file: <path>)
(Current directory: <cwd>)
bash-$
```

First, you should _always_ include a general thought about what you're going to do next. Then, for every response, you must include exactly _ONE_ tool call/function call.

Remember, you should always include a _SINGLE_ tool call/function call and then wait for a response from the shell before continuing with more discussion and commands. Everything you include in the DISCUSSION section will be saved for future reference. If you'd like to issue two commands at once, PLEASE DO NOT DO THAT! Please instead first submit just the first tool call, and then after receiving a response you'll be able to issue the second. Note that the environment does NOT support interactive session commands (e.g. python, vim), so please do not invoke them.

Figure 36: The system prompt for SWE-agent judge for AuthorAblation.

Instance Prompt

SETTING: You are in an empty repository root directory '/repo'. We're currently want to review an ablation studies plan for the research paper {{paper_title}}. Here's the research abstract:

ABSTRACT: {{abstract}}

INSTRUCTIONS: Below, you will find the ablations performed in the paper and the suggested ablation plan. We do not provide any information which ablations belong to which source, so you need to carefully analyze each ablation in both paper and plan and determine if there is a match. Your task is to find a match between ablations in side A and ablations in side B, and vice versa.

A match is considered valid only if all of the following are true:
1. The 'ablated_part' (i.e., the component or mechanism being ablated) must refer to the same component.
2. The 'action' in one ablation must be explicitly allowed by an ablation in the other side.
3. If the action is REPLACE or ADD, the 'replacement' content must have at least one valid option match.

Examples:
- REMOVE X → REPLACE X with [remove, Y] MATCH
- REPLACE X with Y → REMOVE X NO MATCH
- REMOVE X → REMOVE X MATCH
- REPLACE X with [Y, Z] → REPLACE X with [Z, W] MATCH
- ADD Y to X → ADD [Y, Z] to X MATCH
- REPLACE (X+Y) with [X, Y, Z] → [REMOVE X, REMOVE Y] MATCH

Be conservative: if it is unclear whether one ablation allows the other ablation, do NOT match.
The match can contain multiple ablations from either side, as long as the match meets the criteria above. For example, if an ablation in side A is split into two ablations in side B, but both of them together meet the matching criteria, then it is still considered a valid match.

What You Need to Do:
Create the file '/repo/final_score.jsonl' using the command 'create_final_score' and then fill in the following fields for each entry:

{"name_in_A": [name of the ablation in side A OR a list of ablation names in side A OR null if there is no matching ablation for the ablation in side B], "name_in_B": [name of the matching ablation from side B OR a list of ablation names in side B OR null if there is no matching ablation for the ablation in side A]}

GENERAL IMPORTANT TIPS:
1. The file should be initialized using the command 'create_final_score', after that you just need to edit it and fill in the 'name_in_B' or 'name_in_A' (where applicable) field.
2. The 'name_in_A' and 'name_in_B' should match exactly to the ablation names in side A and side B, respectively.
3. You must go over ALL the ablations in side A and all ablations in side B, _each one of them should be in your prediction_.
4. It is OK if one ablation from side A is split into multiple ablations in side B, or merged several side A ablations into one — as long as the matching criteria (per ablation in side A) are met, the match is still valid. This holds to the other direction as well (side B to side A).
5. Please note that all information is in the following text and not in any file or directories in the environment, so

you should base your decision only using the information below.
6. Any operations on the environment besides creating the ‘/repo/final_score.jsonl‘ file and editing it are not necessary.

STRATEGY:
1. First call the ‘create_final_score‘ command to create the file.
2. Then, for each ablation in side A, you should decide based on the criteria above if there is a valid match in side B, and vice versa.
3. After that, you should have the information to fill in the ‘name_in_B‘ / ‘name_in_A‘ fields for each ablation in side A and side B, respectively. If the ablation is considered a match to one (or more) of the ablations in the other side, you should fill the appropriate ablation(s) in ‘name_in_B‘ / ‘name_in_A‘, otherwise it should remain null.
4. Submit your final score using the ‘submit‘ command. PLEASE NOTE THAT THE SUBMISSION FILE NEEDS TO BE INITIALIZED USING THE ‘create_final_score‘ COMMAND. When you are done, run the ‘submit‘ command to confirm. You have access to the terminal session, so take your time and be precise.

Here are the ablations in side A and side B:
<ablations_in_A> {{side_A}} </ablations_in_A>
<ablations_in_B> {{side_B}} </ablations_in_B>

Remember to include all the ablations in side A and side B in your predictions!

```
(Open file: {{open_file}})
(Current directory: {{working_dir}})
bash-$
```

Figure 37: The task instance prompt for SWE-agent judge for AuthorAblation.

D.1.3 LM-PLANNER

System Prompt

SETTING: You are an autonomous computer science researcher, an expert at analyzing machine learning papers and their ablation studies.
You need to format your output using two fields; discussion and predictions. The discussion field should contain a clear explanation of your reasoning, and the predictions field should contain the final output. The predictions field should be in a strict JSONL format, with each line containing a JSON object representing the final output.

Each JSON object should have the following fields:
1. "name": name of the ablation experiment.
2. "ablated_part": high-level description of the part of the method in the research that you want to ablate.
3. "action": the action you want to take on the ablated part (REMOVE, REPLACE, ADD). If the action is REPLACE or ADD please add a field named "replacement" and specify a list of possible replacements/additions, for example if you want to change the value of a parameter please specify a list of values to test.
4. "metrics": a list of metrics to report of the ablation experiment. Please pay special attention and use the metrics that are also used in the paper.
Your output should always include _one_ discussion and _one_ predictions field EXACTLY as in the following example:

<discussion>
Output here the step by step reasoning of the ablation plan you suggest.
</discussion>

<predictions>
{"name": "Ablation A", "ablated_part": "description of the ablated part", "action": "REMOVE", "metrics": ["metric1", "metric2"]}
{"name": "Ablation B", "ablated_part": "description of the ablated part", "action": "REPLACE", "replacement": ["replacement1", "replacement2"], "metrics": ["metric3"]}
</predictions>

Figure 38: The system prompt for LM planner for AuthorAblation.

Instance Prompt

We’re currently want to suggest ablation studies for the research titled {{paper_title}}. Here’s the research abstract:

ABSTRACT:
{{problem_statement}}
The paper source is provided below, after all of the instructions.

INSTRUCTIONS: Now, you’re going to suggest UP TO {{num_ablations}} ablation studies on your own, in a JSONL format. You need to rank the output ablation studies by their importance, and you should only include the most important ones. Each suggestion should include the following fields in a separate JSON:
1. "name": name of the ablation experiment.
2. "ablated_part": high-level description of the part of the method in the research that you want to ablate.
3. "action": the action you want to take on the ablated part (REMOVE, REPLACE, ADD). If the action is REPLACE or ADD please add a field named "replacement" and specify a list of possible replacements/additions, for example if you want to change the value of a parameter please specify a list of values to test.
4. "metrics": a list of metrics to report of the ablation experiment. Please pay special attention and use the metrics that are also used in the paper.
When you’re satisfied with your ablation studies plan, you can submit your plan.

GENERAL IMPORTANT TIPS:
1. The paper source is provided below, after all of the instructions.
2. Less is more - don’t aim to change completely the method, take the important parts and investigate them.

STRATEGY:
1. Read the paper sections, especially the method section, to understand the method.
2. Look at the main components of the method and think about what would happen if you change them.
3. Keep in mind that ablation studies aim to attribute the method’s performance to its major components.
5. Remember that ablation studies does not necessarily mean removing parts of the method, but also changing them.

<paper_source>
{{paper_source}}
</paper_source>

Figure 39: The task instance prompt for LM planner for AuthorAblation.

D.1.4 AGENT-PLANNER

The agent planner system prompt is identical to the agent judge system prompt given in Figure 36.

Instance Prompt

You are in an empty repository root directory '/repo'. The paper TeX source is located in the '/paper' read-only directory. We're currently want to suggest ablation studies for the research titled {{paper_title}}. Here's the research abstract:

ABSTRACT:
{{problem_statement}}

INSTRUCTIONS:
Now, you're going to suggest UP TO {{num_ablations}} ablation studies on your own, in a JSONL file '/repo/ablations_plan.jsonl'. You need to rank the output ablation studies by their importance, and you should only include the most important ones. Each suggestion should include the following fields in a separate JSON:
1. "name": name of the ablation experiment.
2. "ablated_part": high-level description of the part of the method in the research that you want to ablate.
3. "action": the action you want to take on the ablated part (REMOVE, REPLACE, ADD). If the action is REPLACE or ADD please add a field named "replacement" and specify a list of possible replacements/additions, for example if you want to change the value of a parameter please specify a list of values to test.
4. "metrics": a list of metrics to report of the ablation experiment. Please pay special attention and use the metrics that are also used in the paper.

Your terminal session has started and you're in the repository's root directory. You can use any bash commands or the special interface to help you. The environment is already configured. YOU DON'T NEED TO INSTALL ANYTHING IN THE ENVIRONMENT. Remember, YOU SHOULD ALWAYS INCLUDE EXACTLY ONE TOOL CALL/FUNCTION CALL PER RESPONSE. When you're satisfied with your ablation studies plan, you can submit your plan to the code base by simply running the submit command. Note however that you cannot use any interactive session commands (e.g. python, vim) in this environment, but you can write scripts and run them. E.g. you can write a python script and then run it with the python command.

GENERAL IMPORTANT TIPS:
1. To view the paper you should first run 'cd /paper'.
2. Less is more - don't aim to change completely the method, take the important parts and investigate them.
3. Don't try to run extensive operations, remember that we just want to create the plan and to execute it for now.
4. If you run a command and it doesn't work, try running a different command. A command that did not work once will not work the second time unless you modify it!
5. If you open a file and need to get to an area around a specific line that is not in the first 100 lines, say line 583, don't just use the scroll_down command multiple times. Instead, use the goto 583 command. It's much quicker.
6. Always make sure to look at the currently open file and the current working directory (which appears right after the currently open file). The currently open file might be in a different directory than the working directory! Note that some commands, such as 'create', open files, so they might change the current open file.

STRATEGY:
1. Read the paper sections, especially the method section, to understand the method.
2. Look at the main components of the method and think about what would happen if you change them.
3. Keep in mind that ablation studies aim to attribute the method's performance to its major components.
5. Remember that ablation studies does not necessarily mean removing parts of the method, but also changing them.
(Open file: {{open_file}}) (Current directory: {{working_dir}}) bash-$

Figure 40: The task instance prompt for SWE-agent planner for AuthorAblation.

D.2 ReviewerAblation

D.2.1 LMJUDGE

System Prompt

SETTING: You are an autonomous computer science researcher, an expert at analyzing machine learning papers and their ablation studies.

You need to format your output using two fields; discussion and predictions. The discussion field should contain a clear explanation of your reasoning, and the predictions field should contain the final output. The predictions field should be in a strict JSONL format, with each line containing a JSON object representing the final output.

Each JSON object should have the following fields:
1. "name_in_plan": name of the ablation in the plan.
2. "appears_in_review": true/false based on the criteria given to you.

Your output should always include _one_ discussion and _one_ predictions field EXACTLY as in the following example:

<discussion>
The review contains the following new ablations suggestions:
1. Ablation X: The reviewers explicitly suggests that ablation about the attention component is missing.
2. Ablation Y: The reviewers explicitly suggests that other LoRA configuration should be used for training as part of the ablation study.
The ablation in plan contains three different ablations, we will analyze each of them separately:
1. Ablation A: This ablation is about changing the rank as part of the LoRA configuration. This is a match with Ablation Y suggested by the reviewers.
2. Ablation B: This ablation is about the training data. The reviewers did not suggest any ablation related to the training data, so this is not a match.
3. Ablation C: This ablation is about the evaluation metric. The reviewers did not mention any similar ablation, so we cannot consider it a match.
</discussion>

<predictions>
{"name_in_plan": "Ablation A", "appears_in_review": true}
{"name_in_plan": "Ablation B", "appears_in_review": false}
{"name_in_plan": "Ablation C", "appears_in_review": false}

</predictions>

Figure 41: The system prompt for LM judge for ReviewerAblation.

Instance Prompt

SETTING: You are currently reviewing an ablation studies plan for the research paper {{paper_title}}. Here's the research abstract:

ABSTRACT: {{abstract}}

The paper source is provided below, after all of the instructions. You should interpret the review(s) of the paper in the context of the paper source, which may contain important information about the ablation studies performed in the paper, and can help you better understand the ablation suggestions in the review.

INSTRUCTIONS: Below, you will find official review(s) of the paper referring to the ablation studies performed in the paper, alongside the suggested ablation plan. Your task is to determine, for each ablation in the suggested ablation plan, whether it appears in the review(s).

An ablation in plan is considered to appear in the review if all of the following are true:
1. The `ablated_part` (i.e., the component or mechanism being ablated) matches exactly to one of the ablation suggestions in the review text.
2. The `action` in one ablation must match exactly (if it exists in the review text ablation suggestion).
3. If the action is REPLACE or ADD, the `replacement` content must match (if it exists in the review text ablation suggestion). * If the field `"replacement"` contains multiple valid options, your match is valid if at least one correct replacement is identified in the suggested plan.

Be conservative: if it is unclear whether one ablation matches another, do not match. However, because the review text may be incomplete, an ablation is still considered a match if the `ablated_part` matches exactly (Criterion 1) and the action or replacement information (Criteria 2 and 3) are not mentioned in the review text.

What You Need to Do:
Fill in the following fields for each suggested ablation in the plan:

{"name_in_plan": [name of the ablation in the plan], "appears_in_review": [true/false based on the criteria above]}

GENERAL IMPORTANT TIPS:
1. The `name_in_plan` should match exactly to the ablations in the suggested plan.
2. You must go over all the ablations in the suggested plan.
3. Please note that all review information is in the following text, so you should base your decision only using the information below.
4. You should treat the review _ONLY FOR NEW ABLATION SUGGESTIONS_, and not for existing ones or other aspects of the paper or the review. For example, the review text may contain suggestions for other experiments which are NOT ablations.
5. You are given the paper source — use it to better understand the context of the review and the ablation suggestions. For example, the paper source contains already the ablation studies performed in the paper, which can help you understand better the ablation suggestions in the review.

STRATEGY:
1. You should extract from each review the missing ablation that the reviewer suggests to add. A missing ablation refers to an experiment that the reviewer believes should have been conducted, in which a specific component, module, feature, or design choice is removed, replaced, or altered in order to assess its impact on the model's performance. The review must clearly indicate that such an ablation study is missing or should be added.
2. After that, you should have the information to fill in the `appears_in_review` field for each ablation in the suggested plan. If the ablation is considered a match to one of the reviews, you should set `appears_in_review` to true, otherwise false.

Here are the review(s) of the paper VS. suggested plan:

<official_review> {{official_reviews}} </official_review>

<suggested_plan> {{problem_statement}} </suggested_plan>

<paper_source> {{paper_source}} </paper_source>

BELOW IS AN EXAMPLE THAT WILL HELP YOU UNDERSTAND THE FORMATTING AND THE TASK BETTER. PLEASE USE IT FOR REFERENCE ONLY AND DO NOT REPEAT IT IN YOUR FINAL ANSWER.

<official_review> **summary**: This paper proposes EG-SAT, an equivariant graph self-attention transformer for modeling 3D molecular structures, introducing Attention-based Atom-Centered Symmetry Functions (AACSFs) to capture higher-order geometric interactions. The authors showed the performance of proposed model on QM9 and MD17 datasets. While the paper presents some interesting ideas around combining attention mechanisms with ACSFs, there are several critical limitations that need to be addressed.
soundness: 1
presentation: 3
contribution: 2
strengths: 1. The integration of attention mechanisms with ACSFs is interesting. 2. the framework is applicable to multiple molecular property prediction tasks 3. the paper discusses the theoretical foundation of symmetry and group representation in detail
weaknesses: 1. The paper omits several recent works in molecular property prediction, making the comparisons less relevant. 2. The authors didn't conduct ablation studies to evaluate the contribution of different components in the framework. 3. There is no computational efficiency analysis, what's more, the claims of improved scalability are unsupported by any experiments. 4. The motivation for incorporating angular information lacks clear examples where angular information provides benefits. 5. There's no proof that the attention mechanism preserves chemical validity
questions: 1. Can the authors provide experiment for the claimed scalability improvements over traditional ACSFs? 2. How does the computational complexity scale with the number of atoms and chemical elements compared to existing methods? 3. What is the memory footprint of the attention mechanism for larger molecular systems? 4. Can the authors provide ablation studies showing the specific benefits of angular information integration? 5. How sensitive is the model to hyperparameter choices, particularly the attention and gating parameters?

flag_for_ethics_review: ['No ethics review needed.']
rating: 3
confidence: 4
code_of_conduct: Yes </official_review>

<discussion>
The review text contains several weaknesses and questions. While in the weaknesses section, the reviewer mentions the lack of ablation studies, it does not specify which ablations are missing. However, in the questions section, the reviewer explicitly asks for ablation studies showing the specific benefits of angular information integration. The reviewer also suggests the need for experiments to validate the claimed scalability improvements, which could imply a missing ablation, but since we are being conservative we won't consider this an ablation suggestion.
Therefore, the ablation suggestions that can be extracted from the review text are:
1. Angular information integration ablation: The reviewer explicitly requests ablation studies to demonstrate the benefits of incorporating angular information, indicating that this is a missing ablation.
Now let's analyze the ablation in the suggested plan:
1. Remove Radial ACSFs: This ablation focuses on removing the radial component of ACSFs. The review does not mention anything about radial ACSFs, so this is not a match.
2. Remove Angular ACSFs: This ablation involves removing the angular component of ACSFs. This directly corresponds to the reviewer's request for ablation studies on angular information integration. Although the reviewer does not mention the specific action of removal, the ablated part matches exactly, so we consider this a match.
3. Replace Gating Mechanism: This ablation suggests replacing the gating mechanism in ACSFs with alternatives like simple summation or weighted average. The review does not mention anything about the gating mechanism, so this is not a match.
</discussion>

<predictions>
{"name_in_plan": "Remove Radial AACSFs", "appears_in_review": false}
{"name_in_plan": "Remove Angular AACSFs", "appears_in_review": true}
{"name_in_plan": "Replace Gating Mechanism", "appears_in_review": false}
</predictions>

Figure 42: The task instance prompt for LM judge for ReviewerAblation.

D.2.2 AGENTJUDGE

System Prompt

SETTING: You are an autonomous computer science researcher, an expert at analyzing machine learning papers and their ablation studies, and you're working directly in the command line of a linux container with a special interface.

The special interface consists of a file editor that shows you {{WINDOW}} lines of a file at a time. In addition to typical bash commands, you can also use specific commands to help you navigate within files. To call a command, you need to invoke it with a function call/tool call.

RESPONSE FORMAT: Your shell prompt is formatted as follows:
`(Open file: <path>)`
`(Current directory: <cwd>)`
`bash-$`

First, you should _always_ include a general thought about what you're going to do next. Then, for every response, you must include exactly _ONE_ tool call/function call.

Remember, you should always include a _SINGLE_ tool call/function call and then wait for a response from the shell before continuing with more discussion and commands. Everything you include in the DISCUSSION section will be saved for future reference. If you'd like to issue two commands at once, PLEASE DO NOT DO THAT! Please instead first submit just the first tool call, and then after receiving a response you'll be able to issue the second. Note that the environment does NOT support interactive session commands (e.g. python, vim), so please do not invoke them.

Figure 43: The system prompt for SWE-agent judge for ReviewerAblation.

Instance Prompt

SETTING: You are in an empty repository root directory `/repo`. We're currently reviewing an ablation studies plan for the research paper {{paper_title}}. Here's the research abstract:

ABSTRACT: {{abstract}}

The paper source is provided in the `/paper` read-only directory. You should interpret the review(s) of the paper in the context of the paper source, which may contain important information about the ablation studies performed in the paper, and can help you better understand the ablation suggestions in the review.

INSTRUCTIONS: Below, you will find official review(s) of the paper referring to the ablation studies performed in the paper, alongside the suggested ablation plan. Your task is to determine, for each ablation in the suggested ablation plan, whether it appears in the review(s).

An ablation in plan is considered to appear in the review if all of the following are true:
1. The `ablated_part` (i.e., the component or mechanism being ablated) matches exactly to one of the ablation suggestions in the review text.
2. The `action` in one ablation must match exactly (if it exists in the review text ablation suggestion).
3. If the action is REPLACE or ADD, the `replacement` content must match (if it exists in the review text ablation suggestion). * If the field `"replacement"` contains multiple valid options, your match is valid if at least one correct replacement is identified in the suggested plan.

Be conservative: if it is unclear whether one ablation matches another, do not match. However, because the review text may be incomplete, an ablation is still considered a match if the `ablated_part` matches exactly (Criterion 1) and the action or replacement information (Criteria 2 and 3) are not mentioned in the review text.

What You Need to Do:
Create the file `/repo/final_score.jsonl` using the command `create_final_score` and then fill in the fol-

lowing fields for each entry:

{"name_in_plan": [name of the ablation in the plan], "appears_in_review": [true/false based on the criteria above]}

GENERAL IMPORTANT TIPS:

1. The file should be initialized using the command `create_final_score`, after that you just need to edit it and fill in the `appears_in_review` (where applicable) field.
2. DO NOT modify the field `name_in_plan` at all.
3. You must go over all the ablations in the suggested plan.
4. Please note that all review information is in the following text, so you should base your decision only using the information below.
5. You should treat the review _ONLY FOR NEW ABLATION SUGGESTIONS_, and not for the existing ones or other aspects of the paper or the review. For example, the review text may contain suggestions for other experiments which are NOT ablations.
6. You are given the paper source in the `/paper` directory — use it to better understand the context of the review and the ablation suggestions. For example, the paper source contains already the ablation studies performed in the paper, which can help you understand better the ablation suggestions in the review.

STRATEGY:

1. First call the `create_final_score` command to create the file.
2. Then, you should extract from each review the missing ablation that the reviewer suggests to add. A missing ablation refers to an experiment that the reviewer believes should have been conducted, in which a specific component, module, feature, or design choice is removed, replaced, or altered in order to assess its impact on the model's performance. The review must clearly indicate that such an ablation study is missing or should be added.
3. After that, you should have the information to fill in the `appears_in_review` field for each ablation in the suggested plan. If the ablation is considered a match to one of the reviews, you should set `appears_in_review` to true, otherwise false.
4. Submit your final score using the `submit` command.

PLEASE NOTE THAT THE SUBMISSION FILE NEEDS TO BE INITIALIZED USING THE `create_final_score` COMMAND WHICH INITIALIZES ALL VALUES WITH FALSE AS THE `appears_in_review`, YOU SHOULD CHANGE IT _FOR EACH_ OF THE LINES IF THERE IS A VALID MATCH TO THE REVIEW. When you are done, run the `submit` command to confirm. You have access to the terminal session, so take your time and be precise.

Here are the review(s) of the paper VS. suggested plan:

<official_review> {{official_reviews}} </official_review>

<suggested_plan> {{problem_statement}} </suggested_plan>

(Open file: {{open_file}})
(Current directory: {{working_dir}})
bash-$

Figure 44: The task instance prompt for SWE-agent judge for ReviewerAblation.

D.2.3 LM-PLANNER

The LM planner system prompt for ReviewerAblation is identical to the LM planner system prompt for AuthorAblation given in Figure 38.

Instance Prompt

We're currently want to suggest missing ablation studies for the research titled {{paper_title}}. Here's the research abstract:

ABSTRACT:
{{problem_statement}}

The paper source is provided below, after all of the instructions.

INSTRUCTIONS:
Now, you're going to suggest UP TO {{num_ablations}} missing ablation studies in the given paper on your own, in a JSONL format. You need to rank the output ablation studies by their importance, and you should only include the most important ones. A missing ablation refers to an experiment that you believe should have been conducted, in which a specific component, module, feature, or design choice is removed, replaced, or altered in order to assess its impact on the method's performance. You must clearly indicate that such an ablation study is missing from the given paper. Each suggestion should include the following fields in a separate JSON:
1. "name": name of the ablation experiment.
2. "ablated_part": high-level description of the part of the method in the research that you want to ablate.
3. "action": the action you want to take on the ablated part (REMOVE, REPLACE, ADD). If the action is REPLACE or ADD please add a field named "replacement" and specify a list of possible replacements/additions, for example if you want to change the value of a parameter please specify a list of values to test.
4. "metrics": a list of metrics to report of the ablation experiment. Please pay special attention and use the metrics that are also used in the paper.
When you're satisfied with your ablation studies plan, you can submit your plan.

GENERAL IMPORTANT TIPS:
1. The paper source is provided below, after all of the instructions.
2. Remember that we are looking for _MISSING IMPORTANT ABLATIONS_, and as such they should attribute to the method's major components.
3. Less is more - don't aim to change completely the method, take the important parts and investigate them.

STRATEGY:
1. Read the paper sections, especially the method section, to understand the method, and pay attention to the ablation section which is usually under the experiment section or in the appendix.
2. You must first understand and extract the ablation studies that already exist in the paper (use them in your discussion field to better reason about missing ablations).
3. Then, look at the main components of the method and think about what would happen if you change them.
4. Keep in mind that ablation studies aim to attribute the method's performance to its major components, so think about what the ablation studies in the paper are currently missing.
5. Remember that ablation studies does not necessarily mean removing parts of the method, but also changing them.

<paper_source>
{{paper_source}}
</paper_source>

Figure 45: The task instance prompt for LM planner for ReviewerAblation.

D.2.4 AGENT-PLANNER

The agent planner system prompt for ReviewerAblation is identical to the agent judge system prompt given in Figure 36.

Instance Prompt

You are in an empty repository root directory '/repo'. The paper markdown is located in the '/paper' read-only directory. We're currently want to suggest missing ablation studies for the research titled {{paper_title}}. Here's the research abstract:

ABSTRACT: {{problem_statement}}

INSTRUCTIONS:
Now, you're going to suggest UP TO {{num_ablations}} missing ablation studies in the given paper on your own, in a JSONL file '/repo/ablations_plan.jsonl'. You need to rank the output ablation studies by their importance, and you should only include the most important ones. A missing ablation refers to an experiment that you believe should have been conducted, in which a specific component, module, feature, or design choice is removed, replaced, or altered in order to assess its impact on the method's performance. You must clearly indicate that such an ablation study is missing from the given paper. Each suggestion should include the following fields in a separate JSON:

1. "name": name of the ablation experiment.
2. "ablated_part": high-level description of the part of the method in the research that you want to ablate.
3. "action": the action you want to take on the ablated part (REMOVE, REPLACE, ADD). If the action is REPLACE or ADD please add a field named "replacement" and specify a list of possible replacements/additions, for example if you want to change the value of a parameter please specify a list of values to test.
4. "metrics": a list of metrics to report of the ablation experiment. Please pay special attention and use the metrics that are also used in the paper.

Your terminal session has started and you're in the repository's root directory. You can use any bash commands or the special interface to help you. The environment is already configured. YOU DON'T NEED TO INSTALL ANYTHING IN THE ENVIRONMENT. Remember, YOU SHOULD ALWAYS INCLUDE EXACTLY ONE TOOL CALL/FUNCTION CALL PER RESPONSE. When you're satisfied with your ablation studies plan, you can submit your plan to the code base by simply running the submit command. Note however that you cannot use any interactive session commands (e.g. python, vim) in this environment, but you can write scripts and run them. E.g. you can write a python script and then run it with the python command.

GENERAL IMPORTANT TIPS:
1. To view the paper you should first run 'cd /paper'.
2. Remember that we are looking for _MISSING IMPORTANT ABLATIONS_, and as such they should attribute to the method's major components.
3. Less is more - don't aim to change completely the method, take the important parts and investigate them.
4. Don't try to run extensive operations, remember that we just want to get the missing ablations plan for now.
5. If you run a command and it doesn't work, try running a different command. A command that did not work once will not work the second time unless you modify it!
6. Please note that in order to open a file that contains spaces you MUST use single quotes like this: 'open 'file that contains spaces.md''. YOU SHOULD NOT USE ANY DOUBLE QUOTES TO OPEN THE FILE AT ALL.
7. If you open a file and need to get to an area around a specific line that is not in the first 100 lines, say line 583, don't just use the scroll_down command multiple times. Instead, use the goto 583 command. It's much quicker.
8. Always make sure to look at the currently open file and the current working directory (which appears right after the currently open file). The currently open file might be in a different directory than the working directory! Note that some commands, such as 'create', open files, so they might change the current open file.

STRATEGY:
1. Read the paper sections, especially the method section, to understand the method, and pay attention to the ablation section which is usually under the experiment section or in the appendix.
2. You must first understand and extract the ablation studies that already exist in the paper (don't save them to a file, just use them in your process to generate new ablations).
3. Then, look at the main components of the method and think about what would happen if you change them.
4. Keep in mind that ablation studies aim to attribute the method's performance to its major components, so think about what the ablation studies in the paper are currently missing.
5. Remember that ablation studies does not necessarily mean removing parts of the method, but also changing them.

(Open file: {{open_file}})
(Current directory: {{working_dir}})
bash-$

Figure 46: The task instance prompt for LM planner for ReviewerAblation.

E AI Assistant Usage

AI assistants were used solely for coding assistance (GitHub Copilot) and for grammatical corrections and sentence rephrasing during the writing process (ChatGPT). AI tools were not used for the ideation of this research project, the creation of the dataset, or any other decisions regarding the research directions.