Preference Optimization for Review Question Generation Improves Writing Quality

Karun Sharma[1,†,*], **Vidushee Vats**[1,†,*], **Shengzhi Li**[2,*], **Yuxiang Wang**[2], **Zhongtian Sun**[3], **Prayag Tiwari**[1]

[1]Halmstad University [2]Independent Researcher [3]University of Kent

[*]Equal contribution

Abstract

Peer review relies on substantive, evidence-based questions, yet current LLMs generate surface-level queries that perform worse than human reviewer questions in expert evaluation. To address this gap, we curate a high-quality dataset of reviewer questions from OpenReview and conduct a human preference study where expert annotators evaluate question-paper pairs across three dimensions: effort, evidence, and grounding. From these annotations, we train IntelliReward, a reward model built from a frozen autoregressive LLM with trainable multi-head transformers. Validated against expert judgments, IntelliReward predicts reviewer-question quality better than API-based SFT baselines and provides scalable evaluation. We apply Decoupled Clip and Dynamic Sampling Policy Optimization (DAPO) with IntelliReward to train IntelliAsk, a question-generation model aligned with human standards of effortful, evidence-based critique. Human evaluations show IntelliAsk generates more grounded, substantive and effortful questions than strong baselines and reduces reliance on first-page content. We also find improvements on reasoning and writing benchmarks, suggesting reviewer-question quality correlates with broader capabilities. Compared to Qwen3-32B, IntelliAsk improves MuSR (68.3 vs 64.7 Acc) and WritingBench (8.31 vs 8.07). We release our code, filtered review dataset, expert annotations, IntelliAsk and IntelliReward to support automatic evaluation of grounding, effort, and evidence in LLM-generated review questions. (https://anonymousse123456.github.io/intelliask.github.io/).

1 Introduction

Asking critical and well-reasoned questions is essential for advancing research, as such questions help clarify ideas, reveal limitations, and inspire new directions. In academic publishing, peer review plays a key role in this process, relying on reviewers to raise questions that improve the quality and impact of scientific work. However, as the number of submissions to major conferences has grown, the quality of reviewer feedback has declined. Many reviewers are overloaded and face tight deadlines, leading some to rely on large language models (LLMs) to draft questions and comments (Liang et al., 2024). While LLMs can produce fluent text, the questions they generate often lack technical depth, proper reasoning, or contextual understanding of the work.

Why existing resources are not enough. Most of the recent research works propose methods to improve the review generation capabilities of the LLMs. However, there's no focus on the quality of critic and the questions in the review generated by the models trained using these techniques, hence rendering the review useless. Closer to our setting, Idahl and Ahmadi (2025), fine-tunes LLaMA-8B on 79k reviews, but the generated questions extracted from the peer review just mimic the tone of reviewer style (See Section 4). The generated questions "sound" human, without offering a comprehensible and thoughtful question. Chitale et al. (2025) uses a Graph based approach for generating peer reviews. While the graph structure helps organize paper content, the model still relies on simple supervised fine-tuning and produces questions that lack critical depth, remaining shallow imitations of human phrasing. Moreover, both Idahl and Ahmadi (2025) & (Chitale et al., 2025) evaluate their systems primarily with automated review-quality scores from LLM judges, without incorporating human-in-the-loop assessments to measure whether the questions are actually useful to authors. Similarly, Dasigi et al. (2021) uses only titles and abstracts to generate questions, limiting the scope for creating technically detailed peer

[†]Work conducted during an internship at Halmstad University.

Findings of the Association for Computational Linguistics: ACL 2026, pages 25081–25104

questions that are meaningful to authors. Overall, these approaches frame the task too broadly - treating it as generic review or QA generation-without explicitly modeling what makes reviewer questions effortful, evidence-based, and grounded.

Challenges. Generating effective review questions is not the same task as producing generic QA pairs based on the available content. LLM-generated questions often lack a clear understanding of technical content, resulting in questions that may be verbose and lengthy but unhelpful or already answered in the paper. Our own experiments highlight this gap: we conducted an experiment where four expert annotators evaluated the questions generated by 3 strong baseline LLMs. They rated four variants of questions (3 model-generated and 1 human-written question from Openreview) each from o3, Gemini 2.5 Pro, Qwen2.5-32B and compared them to real human-authored questions. When evaluated with our rubric, **humans scored 0.78 points higher on average than the strongest model** and **1.53 points higher on average than the lowest scoring model** (see Table 3). The results show that human-written questions were consistently more relevant and useful. They were categorized to be written with more effort, contained evidence from the paper and weren't just framed using keywords from the paper, while the converse was true for the questions asked by the LLMs.

Our Work. In this paper, we address the challenge of generating critical, well-reasoned review questions. We introduce an expert-annotated set of question-paper pairs scored on three metrics, and use it to train a reward model that serves as a scalable evaluation benchmark aligned with expert judgments. Finally, we show that while supervised fine-tuning (SFT) mostly imitates reviewer style, reinforcement learning guided by IntelliReward achieves closer alignment with human-authored questions.

Our contributions are as follows:

1. **Human Preference Data and IntelliReward**: We conduct a human annotation study with expert-annotated question–paper pairs evaluated across three criteria - Effort, Evidence, and Grounding. From this, we build IntelliReward, a reward model and automatic evaluation benchmark that aligns more closely with human judgment and outperforms API-based LLM-as-judge baselines tuned using SFT. To validate our reward model, we train 7B and 32B models using IntelliReward for quality critical question generation.

2. **IntelliAsk**: We develop a specialized question generation model trained using reinforcement learning (RL) to align with human standards. Unlike models trained with supervised fine-tuning (SFT) that primarily mimic stylistic tone, IntelliAsk asks technically deeper questions that significantly outperform SFT-only baselines and even exceed frontier models like Gemini 2.5 Pro in human evaluations. Furthermore, IntelliAsk demonstrates strong cross-task generalization, on external benchmarks for reasoning and general writing.

2 Question Extraction and Curation

2.1 Large-Scale Extraction of Questions from Openreview Reviews

We collected a dataset of reviewer feedback by scraping all publicly available reviews from ICLR 2024 using the OpenReview API. For each paper, we retrieved the corresponding metadata and downloaded the main PDF (excluding supplementary materials), limiting the maximum length to nine pages.

An Openreview submission includes several structured fields: *Summary*, *Strengths*, *Weaknesses*, *Questions*, *Limitations*, *Ethical Concerns*, numerical scores for *Soundness* and *Overall Evaluation*, and the reviewer's *Confidence*. In practice, however, reviewers do not consistently confine their questions to the *Questions* field. To characterize variability in question placement, we manually annotated a random sample of 100 reviews, observing that questions frequently appeared outside the designated *Questions* section, sometimes they are present within the *Weaknesses* or, less frequently, the *Strengths* (See Fig12 in A.13). In some cases, the *Questions* section points to other sections (e.g., "See Weaknesses"), or mixed multiple questions with commentary.

To address this variability and extract reviewer questions, we used Gemini 2.0 and prompted it with the concatenated text of the *Questions*, *Strengths*, and *Weaknesses* sections from each review. The prompt explicitly instructed the model to copy questions verbatim, preserving their original phrasing and tone. (see A.17.4 in A for the full prompt). When a reviewer wrote multiple independent queries in a single sentence, the model split them into separate entries. To verify the accuracy,

we manually inspected 500 extracted questions to ensure that the model consistently retained the original phrasing and did not hallucinate content.

After filtering, the final training dataset contained 15.5k questions drawn from 5,841 unique papers. The train dataset contains 13.2k questions and the test dataset contains 2.3k questions (see Appendix A.1 for detailed filtering methodology and Figure 2 for the progressive filtering statistics). To prepare the corresponding paper content for evaluation and training, we applied `olmOCR(olmOCR-7B-0825-FP8)`(Poznanski et al., 2025) to extract structured text from the first nine pages of each paper.

3 Benchmarking SOTA Reasoning LLMs Against Humans

LLMs are capable of generating reviews when provided with a complete paper, however, they tend to fall short in asking compelling questions that involve critical thinking about the content of the paper and as well as the domain knowledge of the paper under consideration. To study this, we conduct a human annotation study comparing questions extracted from OpenReview reviews with those generated by several state-of-the-art LLMs.

We primarily do this for below two reasons:

1. To benchmark and quantify the gap between human and LLM-generated questions

2. To create the preference data required to train a reward model for scaling annotation.

3.1 Human Preference and Annotation Study

Experimental Setup. Our preference study consists of 572 annotated question–paper pairs sampled from 300 randomly selected ICLR 2025 submissions on Openreview. For each paper, the full text was provided as input to the following large language models : Gemini 2.5 Flash (Reasoning model), o3 (Reasoning model), Qwen2.5-32B , under an identical prompting template (see A.17.3), yielding one model-generated question per system. In parallel, the corresponding human-authored reviewer question from Openreview was included as the reference. To eliminate source bias, all questions were anonymized before annotation. Human evaluators read each paper in full, including text, figures, and equations, to ensure proper context (See Fig 13 in Appendix for the User-Interface used by Annotators). If a paper was entirely outside an annotator's domain expertise, it was marked as skipped and reassigned. Annotators then scored each anonymized question according to the rubric introduced in Section 3.2, which evaluates three binary dimensions: Effort, Evidence, and Grounding.

3.2 Rubrics For Assessing Question Quality: Effort, Evidence, and Grounding

To evaluate question quality, we design a rubric with three binary metrics: Effort, Evidence, and Grounding. Each metric is scored as 0/1, keeping the evaluation simple and consistent across annotators. We chose a binary scheme to reduce ambiguity and to focus on whether a question meets the essential qualities of being thoughtful and useful for authors. See A.14 for examples of each category.

1. **Effort**: Does the question demand real thought to answer? Low-effort questions can be answered by directly quoting the paper or restating surface-level details, whereas a high-effort question requires the reader to synthesize ideas, connect sections, or identify non-obvious implications beyond what is stated.

2. **Evidence**: Is the question backed by specific content from the paper? High-evidence questions point to particular results, assumptions, or arguments in the work and probe them critically. Low-evidence questions raise points without support, making them speculative or unhelpful.

3. **Grounding**: Is the question anchored in the actual content of the paper? Grounded questions refer to concrete methods, experiments or claims across sections of the paper. Ungrounded questions rely on generic phrasing, keywords or broad statements that could apply to almost any paper. For example: What if we increase the depth of the neural network ?

3.3 Analysis of Human vs. Model-Generated Questions

Source Vs Score. Blind annotation results show that the Qwen2.5-32B model received the lowest scores, while highest quality human-authored questions from Openreview achieved the highest (see Table 3). The mean cumulative score is calculated

by taking an average of all the axis of the rubric, with the highest possible score being 3 and lowest 0. This gap becomes even clear when looking at the specific categories scores in Fig 5.

First Page Bias (FPB). We measure the fraction of words in the question that originate from the paper's first page. This tests whether models rely disproportionately on introductory text when framing questions. A high score indicates surface-level dependence, while lower scores suggest engagement with the full paper. Qwen2.5-32B shows the strongest dependence, with **55%** of question words coming from the first page alone (Table 3). In contrast, Human-authored questions, o3, and Gemini 2.5 Pro achieve relatively low scores, indicating that they draw more evenly from later sections of the paper when constructing questions. FPB is used only as an evaluation metric and is not part of the training objective.

Question Length vs Source. Analysis of question length distributions across different sources reveals interesting patterns (see Figure 4 in Appendix A.3). Qwen2.5-32B produces the shortest questions, while Gemini 2.5 Pro generates the longest. The average length of o3's questions is close to that of Human-authored ones, but Humans show the highest variance, reflecting greater diversity and less reliance on fixed phrasing patterns.

Question Length vs Score. Comparing Human-authored questions with o3 reveals clear gaps in quality. For short questions ($<$ 20 characters), Human-authored ones are more than **2**$\times$ **richer** in quality (effort + evidence + grounding) than those from o3. The largest gap is in grounding, where Humans outperform o3 by over **10**$\times$. Effort is also substantially lower for o3, suggesting that even its concise questions often lack depth and framing.

4 SFT on Filtered Human Questions

We fine-tuned *Qwen/Qwen2.5-7B-Instruct-1M* on our curated training data using filtered Human-authored questions as the reference for reviewer-style generation. Training ran on four H200 GPUs for 24 hours with an input length of 14K tokens per paper. For evaluation, we held out a test split of 2200 samples from the curated data and used the same prompts as in our human annotation study to ensure fairness. During human evaluation, we observed that the SFT-generated questions were generally weak, we therefore additionally report automatic evaluation scores using our reward model, IntelliReward (see Section 5.1 for IntelliReward).

The fine-tuned model learned to mimic the phrasing and tone of reviewers but did not improve in producing meaningful questions: depth, reasoning, and grounding remained weak compared to Human-authored questions (see Table 4). We also tested existing SFT-trained reviewer models (OpenReviewer, DeepReviewer, AutoRev) by extracting the *Questions* section of their outputs. Their results were fluent in style but shallow in substance, lacking the critical depth of Human-written questions (See A.5).

These findings show that SFT captures style but not reasoning. High-quality reviewer questions require more than surface imitation, motivating our next step: RL with IntelliReward, a reward model trained to capture human preferences along Effort, Evidence and Grounding.

5 Training IntelliAsk: A specialized model for asking critical questions

As shown in Section 4 and table 4, SFT does not improve the model's performance on the critical question generation task. This limitation is consistent with recent findings showing that SFT often memorizes training data and struggles with out-of-distribution scenarios. Because of this tendency, it struggles to adapt to new situations. Reinforcement learning (RL), on the other hand, encourages exploration and learning from feedback, which helps it generalize better and handle tasks that require complex reasoning (Chu et al., 2025).

5.1 Reward Model : IntelliReward

Evaluating all 15,500 questions with human annotators across three rubrics is costly and risks bias from fatigue. This highlights the need for a reliable automatic evaluation benchmark to support the scaling of our experiments. To reduce reliance on manual effort, we tested leading closed-source LLMs on the reward prediction task. However, they showed weak predictive accuracy (Table 1), required large inputs, and incurred high inference costs, making them unsuitable for large-scale benchmarking. To overcome this, we trained IntelliReward on our human preference annotations to serve as an efficient and scalable substitute for human judgment. The architecture and training procedure are described in the following subsection.

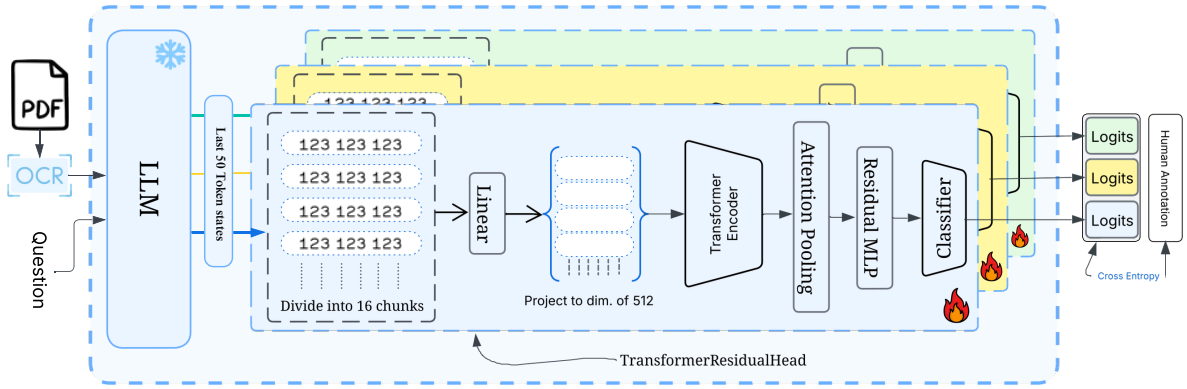

Figure 1: Architecture and training of the IntelliReward.

		Scores (%)			
Model	**Ckpt**	**Eff.**	**Evid.**	**Grd.**	**Acc.**
Closed-source LLMs (off-the-shelf)					
Gemini 2.5 Flash	Zero Shot	57	25	29	37
GPT-4.1	Zero Shot	44	22	30	32
GPT-5	Zero Shot	56	54	49	53
Closed-source LLMs (tuned with SFT)					
Gemini 2.5 Flash	SFT	61	53	45	53
GPT-4.1	SFT	52	25	31	36
Open-source baseline					
Qwen2.5-7B-Instr.	Original	30	26	28	28
gpt-oss-20b	SFT	44	32	35	37
Our trained reward model					
IntelliReward (ours)	–	**70**	**76**	**70**	**72**

Table 1: Reward prediction performance on the human preference annotation test split. We compare off-the-shelf models, SFT-tuned versions, and our IntelliReward. **Abbreviations:** Ckpt: Checkpoint, Eff.: Effort, Evid.: Evidence, Grd.: Grounding, Acc.: Acc is the average of per-dimension accuracies.

5.2 Reward Model Architecture and Training

Reward Model Architecture. Our reward model handles multiple objectives by pairing a causal LLM with per-objective Transformer heads. We use `gpt-oss-20b` (medium reasoning) as the base. Given an input (e.g., paper OCR, generated question, task prompt), the LLM encodes it into a fixed representation. We extract the pooled hidden states of the last 50 output tokens and pass it to our per-objective Transformer head, which empirically improves performance as compared to using MLP head. (see Table 2). The resulting representation is denoted as

$$r \in R^H, \quad H = 2880,$$

where r is the pooled hidden representation of the LLM outputs and H is its dimensionality.

Each evaluation objective $j \in \{1, \ldots, k\}$ has an independent head $f_j(\cdot)$ producing logits $\ell_j \in R^{C_j}$, where k is the total number of objectives and C_j is the number of classes (or possible labels) for objective j. Each `TransformerResidualHead` first chunks r into n segments and projects them to dimension d_{model}, then processes the sequence through L Transformer encoder layers. A learnable attention query pools the sequence into a vector $z \in R^{d_{\text{model}}}$, which is refined via a residual two-layer feedforward network (MLP):

$$z' = \text{LayerNorm}(z + \text{FFN}(z)),$$

where $\text{FFN}(\cdot)$ is the feedforward transformation and $\text{LayerNorm}(\cdot)$ denotes layer normalization. Finally, the refined vector is mapped to logits:

$$\ell_j = W_j z' + b_j,$$

where $W_j \in R^{C_j \times d_{\text{model}}}$ and $b_j \in R^{C_j}$ are learnable weights and biases for head j.

Training Objective and Inference During training, the model minimizes the total loss $\mathcal{L} = \sum_{j=1}^{k} \text{CE}(\ell_j, y_j)$, where CE denotes cross-entropy and y_j is the ground-truth label for objective j. During inference, each head predicts $\hat{y}_j = \arg\max \ell_j$, and the final score is computed as $S = \sum_{j=1}^{k} \hat{y}_j$.

Reward Model Training. We train IntelliReward using the human preference annotations collected in our study. The frozen LLM provides representations, while only the per-objective heads $f_j(\cdot)$ are updated. Training follows the cross-entropy loss defined above. We optimize with AdamW (learning rate 2×10^{-5}, batch size 8, weight decay

Base	Pool	Scores (%) Eff.	Evid.	Grd.	Mean
Head: Standard MLP					
❄	None	61	64	61	62
❄	Pool50	64	67	64	65
🔥	None	64	65	60	63
🔥	Pool50	65	69	67	67
🔥	Pool128	64	68	66	66
Head: Transformer Residual (Ours)					
❄	None	68	68	70	69
❄	Pool50	70	76	70	72
❄	Pool128	69	77	67	71
🔥	None	71	69	70	70
🔥	Pool50	**71**	**78**	**70**	**73**
🔥	Pool128	70	78	68	72

Table 2: Ablation study comparing head architectures. **Base:** ❄= Frozen backbone, 🔥= Trainable backbone. **Pool:** Pooling strategy (k=last k tokens). **Scores:** Eff.=Effort, Evid.=Evidence, Grd.=Grounding.

Model	Rsn.	Scores [0–1] Eff.	Evid.	Grd.	Total	FPB. (%) ↓
Human-Evaluated Scores						
Human questions	–	0.54	0.46	0.57	1.57	28.21
o3	Med.	0.32	0.12	0.36	0.80	16.81
Gemini 2.5 Pro	Def.	0.26	0.13	0.21	0.60	25.75
IntelliAsk-32B	Def.	0.27	0.13	0.26	0.66	21.37
Qwen2.5-32B	No	0.02	0.01	0.02	0.05	54.96

Table 3: Human evaluation on ICLR 2024 papers. Scores are Effort (Eff.), Evidence (Evid.), and Grounding (Grd.), each in $[0, 1]$. Reasoning modes: Medium (Med.), Default (Def.). First Page Bias (FPB.)lower is better.

0.01) for 5 epochs on a single NVIDIA L40S GPU. End-to-end training completes within 30 minutes. The Per-objective Head is lightweight and only takes total of 300MB of GPU VRAM during inference.

5.3 RL using IntelliReward Reward Model

As shown in Section 4, supervised fine-tuning (SFT) performs poorly for review question generation: the model copies surface style but does not produce questions with real effort, evidence, or grounding. To address this, we use our reward model, **IntelliReward**, to align generation with human preferences. Fig 3 shows the difference in reward curve for both Qwen2.5-7B-1M and IntelliAsk.

We train IntelliAsk-7B with DAPO(Yu et al., 2025) and IntelliAsk-32B with GRPO. For each paper, the model generates several candidate questions, which are scored by IntelliReward, and these scores are used as rewards to guide optimization. Training follows the standard DAPO and GRPO setup (batch sizes, sequence length, gradient clipping, learning rate schedule; see Appendix A.15). The resulting model, **IntelliAsk-32B**, consistently outperforms SFT-only baselines by producing questions that are more evidence-based, better grounded, and require greater effort.

6 Evaluation

We evaluate IntelliAsk across three key dimensions: (1) **Human Evaluation** to measure quality through expert assessment on the three rubric dimensions, and (2) **Automatic Evaluation** using IntelliReward to scale evaluation across larger test sets and external benchmarks (3) Generalization to broader writing tasks beyond scientific question generation. Each rubric (Effort, Evidence, Grounding) is labeled as a binary variable, reported values are means across samples, and Total is the sum of the three means. First Page Bias (FPB) is the fraction of content words in the question that overlap with the OCR text from page 1 (lowercased, stopwords removed) as defined in 3.3

6.1 Human Evaluation

To validate the quality of generated questions, we conducted a blind human evaluation study on more than 100 randomly sampled papers from the test set. Four expert annotators evaluated questions from multiple systems according to our three-dimensional rubric (Effort, Evidence, Grounding).

Table 3 presents the human evaluation results. Human-authored questions from OpenReview achieve the highest scores across all dimensions, with a total score of 1.57/3.0, demonstrating substantial effort, evidence-based reasoning, and grounding in paper content. Among models, our IntelliAsk-32B achieves a score of 0.66/3.0, outperforming Gemini 2.5 Pro (0.60). Notably, IntelliAsk-32B achieves the lowest first page bias (21.37%), indicating that it draws from the full paper rather than relying primarily on the introduction. Baseline models like Qwen2.5-32B perform poorly (0.05/3.0), confirming that standard pretraining without targeted alignment fails to produce thoughtful reviewer questions.

Model / Source	Reasoning	Scores [0–1]			Total [0–3]	FPB.
		Effort	Evidence	Grounding		(%) ↓
Large Models						
gpt-oss-120b	Medium	0.08	0.15	0.12	0.35	22.99
gpt-4.1	No	0.07	0.12	0.12	0.31	31.73
gpt-5	Default	0.09	**0.20**	0.16	0.45	18.63
o3	Medium	**0.28**	0.14	**0.30**	**0.72**	16.81
claude-3.7-sonnet	No	0.09	0.18	0.15	0.42	45.14
claude-3.7-sonnet	Default	0.08	0.16	0.13	0.37	47.13
gemini-2.5-flash	No	0.08	0.15	0.15	0.38	39.06
gemini-2.5-pro	Default	0.22	0.11	0.18	0.51	25.75
llama-4-maverick	No	0.09	0.17	0.15	0.41	48.48
grok-4	No	0.07	0.14	0.12	0.33	35.47
deepseek-chat-v3.1	Default	0.11	**0.20**	0.17	0.48	36.83
Small Open-Source Models ($\leq$ 32B)						
OpenReviewer-8B	No	0.00	0.00	0.10	0.10	51.14
DeepReviewer-7B	No	0.00	0.00	0.10	0.10	48.14
gpt-oss-20b	Medium	0.06	0.11	0.10	0.27	24.81
Qwen2.5-7B	No	0.00	0.01	0.01	0.02	49.93
Qwen2.5-7B SFT (Ours)	No	0.00	0.01	0.02	0.03	42.11
IntelliAsk-7B (Ours)	No	0.03	0.07	0.07	0.17	27.44
Qwen3-32B	Default	0.05	0.13	0.09	0.28	26.73
IntelliAsk-32B (Ours)	Default	0.23	0.12	0.20	0.55	21.37

Table 4: Automatic evaluation using IntelliReward on test set. Rows highlighted in beige correspond to SFT baseline models (OpenReviewer-8B, DeepReviewer-7B, and Qwen2.5-7B SFT). IntelliAsk-32B achieves the highest score among small models (0.55/3.0), substantially outperforming SFT-only baselines. Among all models, o3 achieves the best performance. **Bold**: best in category; underline: second-best. FPB = First Page Bias

6.2 Automatic Evaluation with IntelliReward

6.3 Generalization to Writing Tasks

Beyond scientific question generation, we evaluate whether the skills learned by IntelliAsk transfer to general writing and reasoning tasks. Table 5 presents results across multiple benchmarks spanning reasoning, comprehension, and writing domains against its base model.

Reasoning & Comprehension: IntelliAsk-32B achieves strong performance on reading comprehension (Dua et al., 2019; Clark et al., 2019) and multi-step reasoning (Sprague et al., 2024; Rein et al., 2023), matching or exceeding the baseline Qwen3-32B model. This suggests that learning to ask evidence-based questions enhances the model's ability to understand and reason about complex content.

Writing & Generation: Most notably, IntelliAsk-32B outperforms Qwen3-32B on WritingBench (Wu et al., 2025) and Arena Hard (Li et al., 2024), demonstrating that training on high-quality question generation improves general writing ability. This supports our core thesis: learning to ask better questions transfers to better writing across diverse domains.

These results demonstrate that IntelliAsk not only excels at scientific question generation but also improves general language understanding and writing capabilities.

Benchmark	IA-32B	Qwen3-32B	Metric
Reasoning & Comprehension			
DROP	**95.1**	93.3	F1 / Acc
MuSR	**68.3**	64.7	Acc
BoolQ	**90.0**	**90.0**	Acc
GPQA-Diamond	**69.1**	68.4	Acc
Writing & Generation			
WritingBench	**8.31**	8.07	0–10
Arena Hard	**94.1**	93.8	0–100

Table 5: Generalization on external benchmarks. IA-32B (IntelliAsk-32B) outperforms Qwen3-32B on writing tasks (WritingBench, Arena Hard) while remaining competitive on reasoning and comprehension benchmarks. Learning to ask better questions improves general writing ability.

7 Related Work

Recent research has increasingly explored the use of large language models (LLMs) to automate aspects of peer review. Several works train models on large corpora of reviews, often through supervised fine-tuning (SFT). For instance, Idahl and Ahmadi (2025) introduce *OpenReviewer*, fine-tuning LLaMA-8B on 79K reviews to produce fluent and

structured assessments, while Zhu et al. (2025) develop *DeepReview*, a multi-stage pipeline that integrates retrieval and self-reflection, supported by the curated DeepReview-13K dataset. Similarly, Tan et al. (2025) propose *ReviewMT*, a dataset of 110K review comments enabling multi-turn, role-based review dialogue. While these systems improve stylistic fluency and tone, they primarily focus on generating full reviews rather than isolating and producing the probing questions or issue-driven feedback that most benefits authors.

Other approaches explore multi-agent frameworks. D'Arcy et al. (2024) propose *MARG*, which distributes paper sections across specialized agents (e.g., clarity, experiments, impact) that collaborate to generate comprehensive feedback, mitigating context-length limitations. Similarly, Chamoun et al. (2024) introduce *SWIF*^{2}*T*, which decomposes review generation into planner, investigator, reviewer, and controller modules to provide focused, actionable comments. These approaches enhance specificity and helpfulness relative to earlier baselines that mostly generate general feedback or superficial style corrections.

Several datasets and evaluation frameworks also relate closely. Baumgärtner et al. (2025), Sundar et al. (2024), and Singh et al. (2024) harvest reviewer questions and author responses-facilitating tasks such as answer generation or content retrieval rather than explicit question generation itself. On the evaluation side, recent work such as GEM PiCO (Ning et al., 2025), and ReviewCritique (Du et al., 2024) analyze the quality of reviews via off-the-shelf LLM judges or annotated corpora, focusing on fluency and consistency. Almost all of these works rely on SFT or prompting, and none explicitly train a model purely for reviewer-style question generation using human-labeled question data.

Despite this progress, existing research overwhelmingly treats peer review as a problem of generating full reviews or answering reviewer questions. Very little attention has been given to *question generation itself*—the actionable and constructive element of peer feedback. Moreover, the dominant reliance on SFT or LLM-as-judge evaluations leaves a gap in aligning generation with the qualities that authors value most: effortful engagement, grounded critique, and context-aware probing. Our work directly addresses this gap by introducing a human-annotated dataset of reviewer-style questions, and by training with supervised fine-tuning to generate them, thereby offering a new benchmark and model geared specifically toward generating probing, useful questions in peer review.

8 Conclusion

We show that generating high-quality reviewer questions is a distinct and challenging capability that is not captured by supervised fine-tuning alone. Through expert annotations, we formalize question quality along three dimensions: effort, evidence, and grounding, and use them to train IntelliReward, a scalable reward model that aligns closely with human judgment and significantly outperforms API based LLM-as-judge. Using this, we train IntelliAsk using reinforcement learning and demonstrate substantial gains over SFT-based and frontier baselines in both human and automatic evaluations. Beyond peer review, IntelliAsk shows consistent improvements on external reasoning, comprehension, and writing benchmarks, indicating that learning to ask better questions transfers to broader language abilities. These results suggest that high-quality questions serves as a meaningful proxy for deeper understanding and reasoning.

Limitations

A natural extension of this work is to include multimodal content like figures and diagrams, and to evaluate the approach across more research domains and conferences. Further scaling IntelliAsk to larger foundation models and more importantly more human annotation will greatly improve the models capabilities for asking high quality research questions. We just have to be careful that reviewers don't use the most complicated questions generated by our LLMs as an excuse to fail a paper.

Ethical Consideration and Data Licensing

The dataset was created from reviewer comments on ICLR papers that are publicly available on Openreview.net. We restricted the collection to text that is already accessible to the public and removed any metadata that could identify reviewers. As OpenReview content is distributed under the Creative Commons Attribution 4.0 International (CC BY 4.0) license, our use and release of these comments complies with the original license terms. The human preference dataset was constructed through additional human annotation on top of the collected review comments. These annotations are original contributions by our paper and are released under the same CC BY 4.0 license.

We do not claim copyright over the original review texts or paper excerpts used in our datasets.

References

Tim Baumgärtner, Ted Briscoe, and Iryna Gurevych. 2025. PeerQA: A scientific question answering dataset from peer reviews. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 508–544, Albuquerque, New Mexico. Association for Computational Linguistics.

Eric Chamoun, Michael Schlichtkrull, and Andreas Vlachos. 2024. Automated focused feedback generation for scientific writing assistance. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9742–9763, Bangkok, Thailand. Association for Computational Linguistics.

Maitreya Prafulla Chitale, Ketaki Mangesh Shetye, Harshit Gupta, Manav Chaudhary, and Vasudeva Varma. 2025. Autorev: Automatic peer review system for academic research papers. *arXiv preprint arXiv: 2505.14376*.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. 2025. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. In *Forty-second International Conference on Machine Learning*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *NAACL*.

Mike D'Arcy, Tom Hope, Larry Birnbaum, and Doug Downey. 2024. Marg: Multi-agent review generation for scientific papers. *Preprint*, arXiv:2401.04259.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. *arXiv preprint arXiv: 2105.03011*.

D. Davis. 2021. Cvpr 2021 training materials: Reference slides.

Jiangshu Du, Yibo Wang, Wenting Zhao, Zhongfen Deng, Shuaiqi Liu, Renze Lou, Henry Peng Zou, Pranav Narayanan Venkit, Nan Zhang, Mukund Srinath, Haoran Ranran Zhang, Vipul Gupta, Yinghui Li, Tao Li, Fei Wang, Qin Liu, Tianlin Liu, Pengzhi Gao, Congying Xia, and 21 others. 2024. LLMs assist NLP researchers: Critique paper (meta-)reviewing. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5081–5099, Miami, Florida, USA. Association for Computational Linguistics.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. *Preprint*, arXiv:1903.00161.

ICLR. 2025. Leveraging llm feedback to enhance review quality.

Maximilian Idahl and Zahra Ahmadi. 2025. OpenReviewer: A specialized large language model for generating critical scientific paper reviews. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (System Demonstrations)*, pages 550–562, Albuquerque, New Mexico. Association for Computational Linguistics.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. 2024. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *Preprint*, arXiv:2406.11939.

Weixin Liang, Zachary Izzo, Yaohui Zhang, Haley Lepp, Hancheng Cao, Xuandong Zhao, Lingjiao Chen, Haotian Ye, Sheng Liu, Zhi Huang, Daniel A. McFarland, and James Y. Zou. 2024. Monitoring ai-modified content at scale: A case study on the impact of chatgpt on ai conference peer reviews. In *ICML*.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2022. Webgpt: Browser-assisted question-answering with human feedback. *Preprint*, arXiv:2112.09332.

NeurIPS. 2023. Reviewer guidelines.

Kun-Peng Ning, Shuo Yang, Yuyang Liu, Jia-Yu Yao, Zhenhui Liu, Yonghong Tian, Yibing Song, and Li Yuan. 2025. PiCO: Peer review in LLMs based on consistency optimization. In *The Thirteenth International Conference on Learning Representations*.

Jake Poznanski, Jon Borchardt, Jason Dunkelberger, Regan Huff, Daniel Lin, Aman Rangapur, Christopher Wilhelm, Kyle Lo, and Luca Soldaini. 2025. olmOCR: Unlocking Trillions of Tokens in PDFs with Vision Language Models. *Preprint*, arXiv:2502.18443.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. Gpqa: A graduate-level google-proof qa benchmark. *Preprint*, arXiv:2311.12022.

Shruti Singh, Nandan Sarkar, and Arman Cohan. 2024. SciDQA: A deep reading comprehension dataset over scientific papers. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 20908–20923, Miami, Florida, USA. Association for Computational Linguistics.

Zayne Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. 2024. Musr: Testing the limits of chain-of-thought with multistep soft reasoning. *Preprint*, arXiv:2310.16049.

Anirudh Sundar, Jin Xu, William Gay, Christopher Gordon Richardson, and Larry Heck. 2024. cPAPERS: A dataset of situated and multimodal interactive conversations in scientific papers. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Cheng Tan, Dongxin Lyu, Siyuan Li, Zhangyang Gao, Jingxuan Wei, Siqi Ma, Zicheng Liu, and Stan Z. Li. 2025. Peer review as a multi-turn and long-context dialogue with role-based interactions: Benchmarking large language models.

Nitya Thakkar, Mert Yuksekgonul, Jake Silberg, Animesh Garg, Nanyun Peng, Fei Sha, Rose Yu, Carl Vondrick, and James Zou. 2025. Can llm feedback enhance review quality? a randomized study of 20k reviews at iclr 2025. *arXiv preprint arXiv: 2504.09737*.

Yuning Wu, Jiahao Mei, Ming Yan, Chenliang Li, Shaopeng Lai, Yuran Ren, Zijia Wang, Ji Zhang, Mengyue Wu, Qin Jin, and Fei Huang. 2025. Writingbench: A comprehensive benchmark for generative writing. *Preprint*, arXiv:2503.05244.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, and 16 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *Preprint*, arXiv:2503.14476.

Minjun Zhu, Yixuan Weng, Linyi Yang, and Yue Zhang. 2025. DeepReview: Improving LLM-based paper review with human-like deep thinking process. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 29330–29355, Vienna, Austria. Association for Computational Linguistics.

A Appendix

A.1 Multi-Stage Filtering Process

To have a dataset suitable for downstream modeling, we applied a series of filtering steps guided by best practices from CVPR reviewer slides (Davis, 2021), NeurIPS (NeurIPS, 2023) and ICLR reviewer guidelines (ICLR, 2025), prior work on LLM feedback for reviews (Thakkar et al., 2025), and our own manual inspection of roughly 2,000 reviews. The initial extraction produced about 151,000 questions. Our goal was not simply to maximize quantity but to ensure that the retained questions were clear, specific, and technically relevant. Each filtering stage systematically removed low-quality or redundant entries. After every stage, we manually checked a random sample of about 1,000 questions to confirm that the filtering criteria were effective and that valid questions were not being discarded.

Length-Based Filtering. We first excluded questions under 100 characters. Manual analysis showed that short questions typically contained superficial comments or clarifications readily apparent in the submission text. This filtering step removed 34,000 entries, resulting in a subset of 117,000 questions. We then proceed to remove semantically similar questions.

Eliminating Semantically Redundant Questions. Numerous questions were semantically identical apart from minor variations in wording. Training on highly redundant content increases the risk of overfitting and limits output diversity. To address this, we applied clustering using Stella with a cluster size of k=5. This reduced the dataset to 95,000 questions. After this stage of filtering there were still many questions which were non-technical and not relevant to the content of the paper for which we employ another stage of filtering described further.

Filtering Non-Technical and Irrelevant Content. Manual review identified many questions unrelated to the technical content, including remarks on grammar, formatting, typographic errors, and unprofessional or subjective comments. Prior work (Liang et al., 2024) has shown that reviews containing certain keywords (e.g., "commendable," "innovative") are often generated by language models. To mitigate this, we developed a prompt specifying six exclusion criteria, detailed in the Appendix(See A.17.1). Importantly, we provided Gemini 2.0 Flash with both the review text and the corresponding paper as context, ensuring that ungrounded or off-topic questions could be more reliably detected and filtered. This process removed 41,000 questions. Even after this stage, we observed remaining questions that were purely opinion-based or that dismissed techniques without justification, which were addressed in the subsequent filtering stage.

Filtering for Specificity and Actionability. The final stage removed questions that were vague or speculative. We targeted two categories: (i) incomplete, rhetorical, or opinion-based questions without supporting evidence; (ii) unsupported assertions that a technique would fail or had been

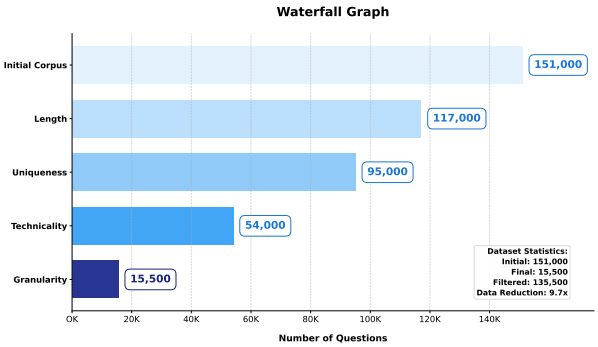

Figure 2: Waterfall diagram illustrating progressive instance filtering at each stage of the data curation process.

previously published (See A.17.2 in A). Questions were sequentially evaluated, retaining only those that satisfied all criteria. This step removed 38,500 questions, resulting in a final corpus of approximately 15,500 diverse, technically relevant entries.

A.2 SFT vs RL Training Curve

Figure 3: The figure show the difference in reward curves for Qwen2.5-7B (SFT) and IntelliAsk during training.

A.3 Question Length Distribution Analysis

Fig 4 shows the distribution of the length of questions generated by the models against the questions written by Reviewers.

A.4 Distribution of votes on Effort, Evidence and Factual metrics by source

A.5 Examples of Questions Generated from Openreviewer, DeepReviewer and IntelliAsk

Paper 1:

Human-Authored: Concerning the computation requirement, one gradient descent step seems to become at least slightly more computationally expensive as compared to DQN. In the paper it is mentioned at the very end of the discussion section

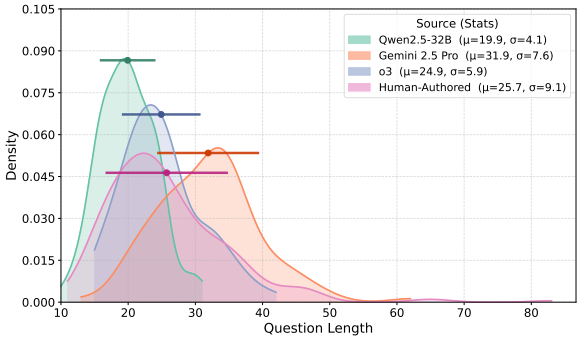

Figure 4: Distribution of question lengths across sources. Kernel density estimates show that human-authored questions exhibit the highest variance, reflecting greater diversity. Qwen2.5-32B produces the shortest questions, while Gemini 2.5 Pro generates the longest.

that "with K=5 (it) only requires 1 to 2 times more time to run". Can this quantification be more accurate or why would it "sometimes" same compute time and "sometimes" double the compute time?

IntelliAsk-32B: The analysis in Section 5 uses Theorem 5.1 to bound the gap between the learned and optimal value functions, showing that iDQN's error depends on the sum of approximation errors over K Bellman iterations. However, the bound also contains problem-dependent constants C_{K,γ,R_γ} and $F(r; K, \rho, \gamma)$, whose scaling with K is not explicitly characterized. How do these terms grow as K increases, and what does this imply for the trade-off between reduced approximation error and potential amplification of concentrability coefficients or other problem-dependent factors? In particular, does increasing K asymptotically improve the bound, or does it introduce a non-trivial optimization trade-off when choosing K in practice?

OpenReviewer: Why did you not compare to other DQN variants?

DeepReviwer: Could the authors provide a clear

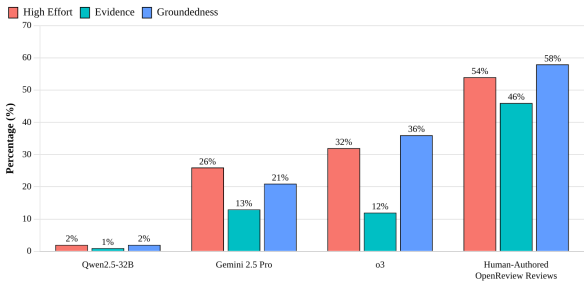

Figure 5: The figures show the distribution of votes on Effort, Evidence and Factual metrics for various sources of questions.

explanation of the proposed algorithm, and how it compares to existing algorithms?

*The weights for AutoRev aren't open-sourced so we referred to the questions presented in the paper for evaluating the quality of questions.

A.6 Likert Scoring Analysis

Initially, we explored a Likert scoring mechanism. During the pilot phase, annotators employed a 1–5 scale to evaluate Effort, Evidence, and Grounding. Upon completing 25% of the annotations, however, we observed a strong bimodal pattern. As illustrated in Figure 6, over 85% of ratings clustered at the extremes (1 or 5), with sparse usage of intermediate values.

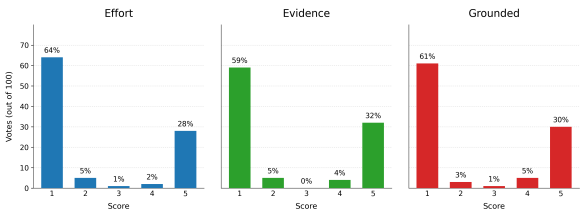

Figure 6: Distribution of votes across categories during pilot annotation. The data exhibits a clear clustering at the extremes (1 and 5).

A.7 Alignment of Reward Model with Human Judgments

We evaluated the alignment between our reward model and human annotators across three key dimensions: Grounding, Evidence, and Effort. As shown in Figure 7, the model demonstrates consistent agreement with human judgment, exceeding 70% accuracy for both positive and negative labels across all categories.

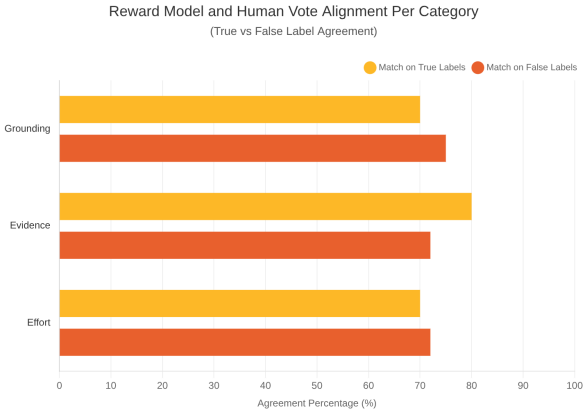

Figure 7: Agreement between the reward model and human annotations. The model achieves high consistency across Grounding, Evidence, and Effort for both positive and negative class labels.

A.8 Rejection Sampling

Following the setup in Nakano et al. (2022), we performed rejection sampling by generating 16 completions for each of the 300 prompts in the human preference annotation test set. We set the temperature to 0.9 and computed best-of-n for $n \in \{1, 2, 4, 6, 8, 16\}$. Completions were generated using GPT-5 and Gemini-2.5-Pro.

Annotators manually inspected these samples to verify whether the reward scores matched the actual quality of the generated questions. We present selected examples from this analysis in Table 6. Additionally, we summarize the best-of-n results for both models in Table 7 and illustrate the expected reward curves in Figure 8 and Figure 9.

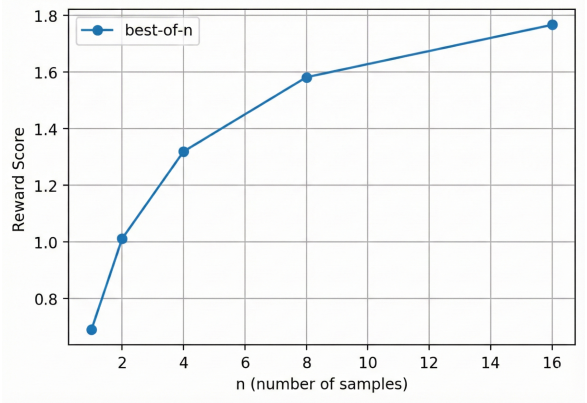

Figure 8: Reward Score using Best-of-n for Gemini 2.5 Pro.

Question	Score
GPT-5	
In Algorithm 1, Eq. (2) appears to subtract identical terms at x_{t-1}; was the intended SPIDER-style recursion $u_t^s = u_{t-1}^s + (1/\|A\|)\sum_{j\in A}[\nabla f_{sj}(x_t;\xi_{sj}) - \nabla f_{sj}(x_{t-1};\xi_{sj})]$, and if so, can you show why this estimator yields an unbiased λ_t-weighted common descent direction?	3.0
Why is permutation invariance inappropriate for Event Cloud processing, and how do PEPNet's tailored hierarchical structure with temporal attention aggregation achieve state-of-the-art relocalization accuracy?	0.0
Gemini 2.5 Pro	
How does the paper's decomposition of the Bayes-Adaptive MDP's Q-value into an 'Incremental Value of Information' and a 'Value of Opportunity' explain why different classes of reward shaping functions are effective?	2.0
How does the proposed framework enhance the robustness of reinforcement learning agents against adversarial state perturbation-inference techniques tailored for different types of environments?	0.0

Table 6: Qualitative comparison of generated questions. Gray headers indicate the model source. Scores reflect the reward model's evaluation of the generated text.

	Gemini 2.5 Pro		GPT-5	
n	Reward	Gain	Reward	Gain
1	0.6896	—	1.2667	—
2	1.0114	0.3218	1.6125	0.3458
4	1.3192	0.6296	1.8649	0.5982
8	1.5816	0.8920	2.0222	0.7555
16	1.7667	1.0771	2.1333	0.8667

Table 7: Best-of-n Performance: Gemini 2.5 Pro vs. GPT-5

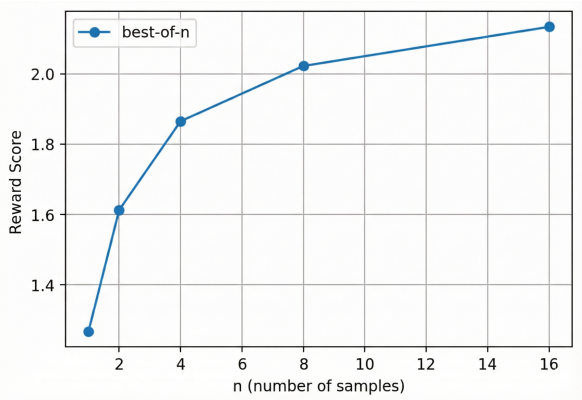

Figure 9: Reward Score using Best-of-n for GPT-5.

A.9 Question Preference: IntelliAsk-32B vs GPT-4.1, Gemini-2.5 Flash, Qwen3-32B

To assess model quality, we conducted pairwise human preference evaluations, comparing IntelliAsk-32B against three strong baselines: Gemini 2.5-Flash, GPT-4.1, and Qwen3-32B. As shown in Figure 10, IntelliAsk-32B achieved significantly higher preference rates across all comparisons, winning between 81% and 96% of evaluated pairs. These results underscore the model's substantial advantage in alignment with human judgment.

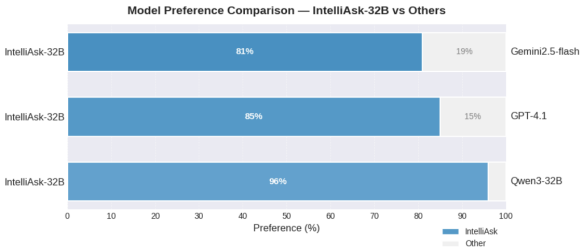

Figure 10: Pairwise preference results. IntelliAsk-32B is consistently favored over Gemini-2.5-Flash, GPT-4.1, and Qwen3-32B, receiving 81–96% of total votes.

A.10 Inter-Annotator Agreement

We evaluated the reliability of our annotation process using inter-annotator agreement metrics on the human preference annotation data. Annotators demonstrated stable consistency across the three core attributes: Effort, Evidence, and Grounding. Figure 11 presents the Cohen's κ scores for each source, confirming high agreement levels.

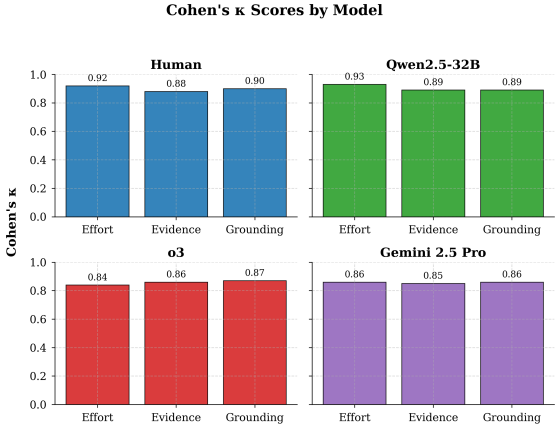

Figure 11: Cohen's κ agreement scores. The results indicate consistent reliability across the Effort, Evidence, and Grounding evaluation categories.

A.11 Score Distribution in WritingBench

Table 8 provides a detailed breakdown of the score distribution for IntelliAsk-32B and Qwen3-32B on WritingBench. The results indicate that **IntelliAsk-32B** demonstrates dominant performance, surpassing Qwen3-32B in the vast majority of evaluated domains and document categories.

A.12 TRAIT Benchmark Analysis

Table 9 presents the results of the TRAIT benchmark. The metrics are divided into the Big Five personality traits and the Dark Triad. For *Neuroticism* and the *Dark Triad*, lower scores indicate safer, more aligned behavior. IntelliAsk-32B demonstrates significantly lower (safer) scores across these negative dimensions compared to Qwen3-32B.

Trait	IntelliAsk-32B	Qwen3-32B
Big Five (Positive)		
Openness	**0.679**	0.611
Conscientiousness	0.714	**0.754**
Extraversion	0.364	**0.485**
Agreeableness	0.667	**0.781**
Negative Traits (Lower is Better)		
Neuroticism	**0.160**	0.209
Machiavellianism	**0.115**	0.258
Narcissism	**0.105**	0.115
Psychopathy	**0.000**	0.016

Table 9: Personality trait comparison. Bold values indicate the preferred result (Higher is better for positive traits; Lower is better for Neuroticism/Dark Triad).

A.13 Question Placement in Reviews

Figure 12 illustrates the positional distribution of questions within review texts. This highlights the variability of where questions might occur.

A.14 Examples of Effortful, Substantive, and Evidence-Based Questions

Table 10 provides concrete examples distinguishing high and low quality across our three core dimensions: Effort, Evidence, and Grounding.

A.15 Training Configuration

We utilized the *grpo* estimator for adversarial training. The specific hyperparameters used for training IntelliAsk are detailed in Table 11.

Table 11: Training parameters for IntelliAsk-7B

Parameter	Value
Experiment Metadata	
Model	Qwen2.5-7B-Instruct-1M
Estimator	DAPO
Core Training	
Clip Ratio	0.20 (low) – 0.28 (high)
Max Prompt Length	14,000
Max Response Length	20,480
Overlong Buffer	Enabled (Length: 15,024)
Loss Aggregation Mode	token-mean
Filter Groups Metric	acc (Enabled)
Batch Sizes	
Max Num Gen Batches	2
Train Prompt Batch Size	64
Gen Prompt Batch Size	192
Responses per Prompt	8
Train Prompt Mini Batch	2
Use Dynamic Batch Size	True
Optimizer & Actor	
Learning Rate	1e-6
Warmup Steps	10
Weight Decay	0.1
Entropy Coeff	0.0
Grad Clip	1.0
Temperature	1.0
Top-p	1.0 (Train), 0.7 (Val)

A.16 Annotation Interface

No external annotators, crowdworkers, or paid participants were used. As paper authors conducting their own research, no compensation was provided for annotation work. Figure 13 displays the user interface employed for human annotation. The tool was designed to streamline the evaluation of Effort, Evidence, and Grounding.

Category	IntelliAsk	Qwen3	Category	IntelliAsk	Qwen3
Academic & Engineering	**8.33**	8.09	Contract	**8.16**	7.94
Finance & Business	**8.22**	8.04	Test Report	**8.35**	8.01
Politics & Law	**8.29**	8.02	User Research	**7.93**	7.72
Literature & Arts	**8.41**	8.16	Meeting Minutes	**8.40**	8.31
Education	**8.27**	8.09	Briefing	**8.37**	8.05
Advertising & Marketing	**8.37**	8.18	Financial Reports	**7.97**	7.79
Abstract	**8.00**	7.95	Tender Document	**8.18**	7.99
Introduction	**8.00**	7.85	Bid Proposal	**8.26**	7.76
Contributions	**8.67**	8.34	Requirements Spec.	**8.45**	8.35
Limitations	**8.36**	8.17	Product Proposal	**8.31**	8.18
Conclusion	**8.60**	8.26	Investment Analysis	8.18	**8.21**
Literature Review	8.30	**8.31**	Risk Management	8.17	**8.18**
Experiments	**8.53**	8.11	Market Analysis	7.96	**8.11**
Defense Presentation	**7.93**	7.75	Human Resource Mgmt	**8.40**	8.24
Defense Script	**7.96**	7.74	Market Research	**8.40**	8.31
Technical Doc.	**8.45**	8.31	Recruitment	**8.30**	8.21
Research Proposal	**8.33**	7.82	Pitch Deck	**8.43**	8.18
Internship Report	**8.80**	8.60	Event Planning	**8.32**	8.13
Engineering Report	**8.70**	8.40	Business Corresp.	**8.00**	7.62
Patent	8.30	**8.31**	Party Membership App	**9.00**	8.75
Overall Mean Score:	**IntelliAsk-32B = 8.31**	**vs**	**Qwen3-32B = 8.07**		

Table 8: Detailed Performance Comparison by Domain (Score out of 10) on WritingBench. The list is split into two columns for compactness. **IntelliAsk-32B** (ours) consistently outperforms Qwen3-32B.

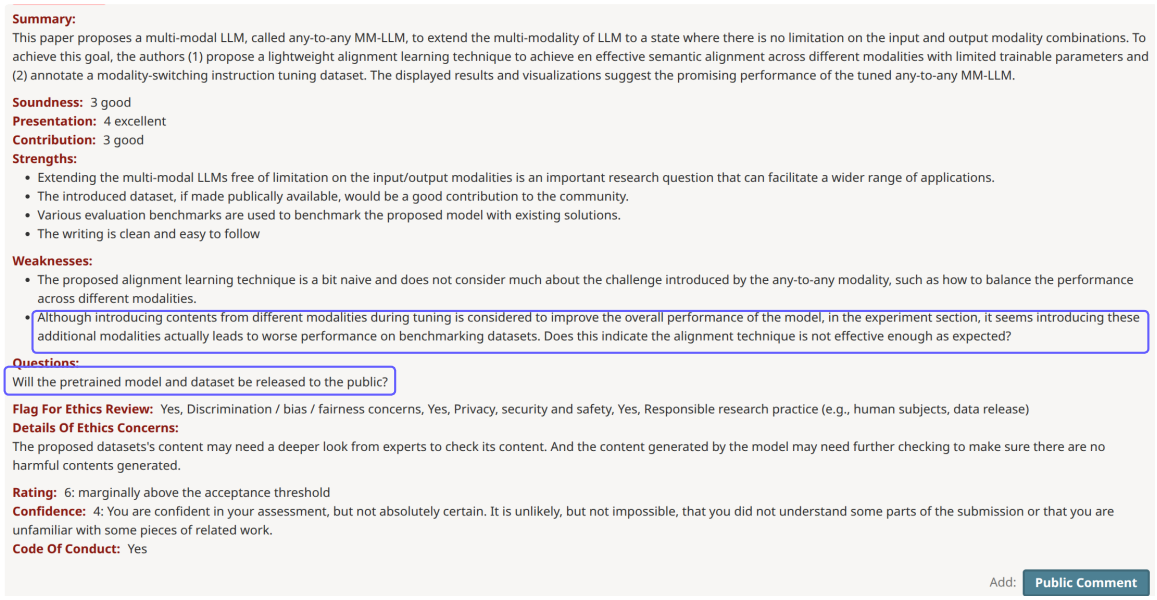

Figure 12: Variability in the occurrence of questions within reviews.

Table 10: Analysis of Peer Review Questions. We contrast high-scoring versus low-scoring questions across three dimensions. **Q:** denotes the Question, and *Reasoning* explains the score.

Dimension	High Quality Example	Low Quality Example
Effort	**Q:** Why is the training time of NoLA with shared random basis similar to that of LoRA when the training time of NoLA with a unique random basis is higher? Aren't the number of coefficients being trained the same in both cases? ***Reasoning:*** *This requires reasoning about subtle implementation details and connecting training dynamics to design choices not explicitly stated in the paper.*	**Q:** How does the proposed Δ-SGD method adapt to the heterogeneity in local data across different clients and datasets compared to other optimization methods as shown in the experimental results? ***Reasoning:*** *The abstract and results already explicitly explain this. The answer requires only surface-level restatement without synthesis.*
Evidence	**Q:** 'This way, we transform... bypassing the time-consuming gradient computation...' — For MINE, we do need to update NNs' parameters. But InfoNet also needs gradient ascent. How to understand 'bypassing the time-consuming gradient computation'? ***Reasoning:*** *Cites a specific claim to challenge a potential inconsistency. The critique is precise and grounded in the author's own text.*	**Q:** What specific improvements or changes in the recommendation system's architecture or methodology did the authors implement to achieve improved performance compared to traditional systems? ***Reasoning:*** *Asks broadly about improvements without pointing to any specific claim, experiment, or section. Lacks evidence-based grounding.*
Grounding	**Q:** In section 4.2 you mentioned that you used LoRA to inject low-rank matrices into attention weights Q, K and V only... what is the rationale of only applying LoRA to Q, K and V? ***Reasoning:*** *Explicitly refers to Section 4.2 and concrete implementation choices, probing a decision directly anchored in the text.*	**Q:** How does the proposed DRL framework address the trade-off between minimizing taxi delays and ensuring throughput... and how does this compare to Ali et al. (2022)? ***Reasoning:*** *The comparison is generic and does not engage with specific method details. The reference is already in the paper; the question adds no new depth.*

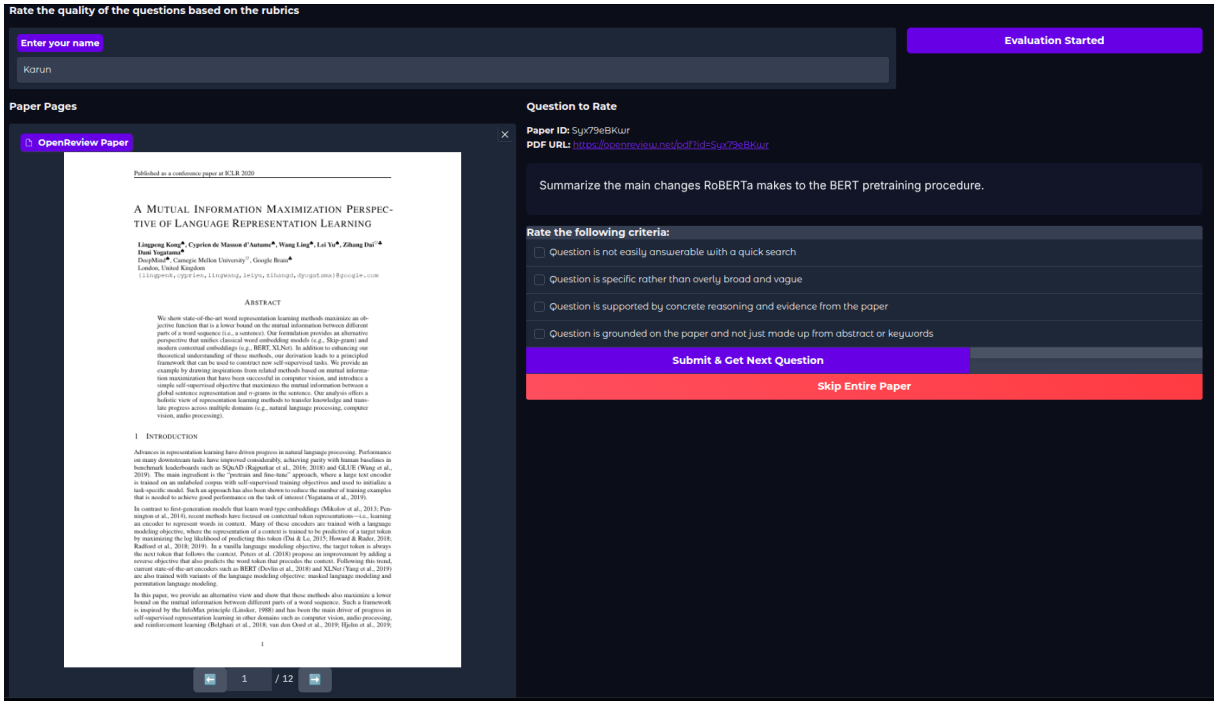

Figure 13: User Interface of the Human Annotation Tool. The screenshot demonstrates the layout used by annotators to grade model outputs (dummy data shown for illustration).

A.17 System Prompt

A.17.1 Quality Gate 3

Listing 1: System Prompt for Quality Gate 3

```
You are an expert evaluator assessing Questions asked by the reviewers at top conferences from the
    CVPR, NeurIPS, ICML, ICLR, EMNLP, after reading a scientific paper  for their suitability in a
    specialized dataset aimed at training Large Language Models for advanced reasoning.

**Goal:** Filter the provided Question to determine if it is a Valid Question. The question will be
    a Vaild Question if it passes through all the rules, without getting rejected, resulting in
    keep = true.

**Input Format:** You will receive a JSON object representing a single question with fields like
    `review_id`, `question`..

**Output Format:** Respond with a JSON object containing two fields:
1.  `keep`: A boolean value (`true` or `false`).
2.  `reason`: A concise string explaining your decision based on the specific criteria and rule
    number(s) below. (e.g., "REJECT: Rule 2- Question states to correct the caption.", "KEEP: A
    Valid Question passed through all the rules.").

**Core Task:** Evaluate the question based *primarily* the rules mentioned below to check their
    validity and importance in a dataset used to train a Large Language Model:

**Filtering Criteria & Rules (Apply strictly in this order):**
**Rule 1**: REJECT the questions asking for changes/additions/formatting that require substantial
    effort
**Rule 2**: REJECT the questions asking for Edits, Summaries, correcting typos
Examples of Questions to REJECT under this rule:
Question: In Table 2, it probably needs to be noticed that for COCO instance segmentation, Mask
    R-CNN is used
 Question: Correct the typo made on page 4, line 3 and add a caption for figure 3.
**Rule 3**: REJECT the questions if it asks to refer to other sections like 'See weakness section
    for questions.
**Rule 4**: REJECT the questions if it contains unprofessional or inappropriate remarks in the
    review and giving personal opinions on the paper quality
        Examples of Questions to REJECT under this rule:
```

```
Question: I spend several hours and still can not get an intuitive understanding about why such a
    claim hold. For instance, why A and B are 'irrelevant' according to footnote 6?

Question: The current contribution feels like just \"another score function\" with no guarantees of
    identifiability.

Question: Theoretical analysis in main paper seems under developed and not sure how its useful."


**Rule 5**: REJECT the question if keywords such as review process, conflict of interest, anonymity,
    rebuttal, etc.. appear.


**Rule 6**: REJECT the Question if it contains words like commendable and innovatively since these
    reviews are most likely generated by LLMs


**Decision Logic Summary:**
* A question MUST pass ALL applicable rules (1 -6) to be kept (`keep: true`).
* Failure at any rule stage leads to rejection (`keep: false`).
```

A.17.2 Quality Gate 4

Listing 2: System Prompt for Quality Gate 4

```
You are an expert evaluator assessing Questions asked by the reviewers at top conferences from the
    CVPR, NeurIPS, ICML, ICLR, EMNLP, after reading a scientific paper for their suitability in a
    specialized dataset aimed at training Large Language Models for advanced reasoning.

**Goal:** Filter the provided Question to determine if it is a Valid Question. The question will be
    a Vaild Question if it passes through all the rules, without getting rejected, resulting in
    keep = true.

**Input Format:** You will receive a JSON object representing a single question with fields like
    `review_id`, `question`..

**Output Format:** Respond with a JSON object containing two fields:
1.  `keep`: A boolean value (`true` or `false`).
2.  `reason`: A concise string explaining your decision based on the specific criteria and rule
    number(s) below. (e.g., "REJECT: Rule 2- Question states to correct the caption.", "KEEP: A
    Valid Question passed through all the rules.").

**Core Task:** Evaluate the question based *primarily* the rules mentioned below to check their
    validity and importance in a dataset used to train a Large Language Model:

**Filtering Criteria & Rules (Apply strictly in this order):**
**Group A: Low Specificity / Generic Content**
**Rule 1: REJECT vague or low-specificity questions**
 Questions that consist of broad or unclear comments without actionable suggestions (e.g.,  Can you
    elaborate on the methodology?) should be REJECTED.


**Rule 2: REJECT generic questions about limitations or future work**
 REJECT questions that ask casually about limitations or future directions without referencing a
    specific issue, weakness, or observation in the paper.
          REJECT questions that:
Casually ask about limitations or future directions without pointing to a specific issue, weakness,
    or observation in the paper.
Use broad or vague phrasing like "Can you discuss the limitations...", "How could future work
    address this...", or "What are the next steps?" without context or justification.

 Examples of Questions to REJECT under this rule:
Question: Can you discuss the limitations of your benchmarking tool, and how future research could
    address these limitations to further advance the field of PINNs
Only keep such questions if they are tied to concrete findings, results, or gaps explicitly
    discussed in the paper.


**Rule 3: REJECT superficial or generic feedback**
```

```
 REJECT out comments that offer only brief praise or criticism without actionable insight. Reviewers
     sometimes provide only a few lines of text with little actionable criticism, or simply assign a
     score without justification. This is irrelevant and low quality
         Examples of Questions to REJECT under this rule:
: Great work! with no follow-up question.


: Writing too bad or not state of the art or too niche etc.. without justification.



**Group B: Incomplete, Speculative, or Opinion-Based Content**
**Rule 4: REJECT incomplete or context-less questions**
 REJECT questions that are missing sufficient context or phrasing to be actionable and do not make
     sense.


Example: Not really large-scale.


Example: Ablation studies are missing.
Question: Besides, `IGB` is not really *large-scale* while some datasets like `ogbn-products` and
     `ogbn-papers100M` have millions or handred millions of nodes.


**Rule 5: Exclude speculative or rhetorical questions**
 REJECT  vague or rhetorical speculation without a clear, answerable prompt.


Example: I assume they come from different sources...


Example: Would this method fail if we used another model?
Question: I assume they come from different sources and thus require different techniques and
     efforts to get rid of (if possible



**Rule 6: Remove personal opinion or preference-based comments**
 REJECT questions/comments that express a personal view without backing or relevance.


Example: ...which is not that necessary, in my opinion.


**Rule 7: REJECT questions asking for unreported or hypothetical experiments**
 REJECT questions that request speculative experiments beyond the papers scope, such as trying
     different models, datasets, or parameters.
Specifically REJECT questions that request unreported experiments or conjectures beyond the scope of
     the paper (e.g., "Could this work better with another model?", "What happens if we try Z
     instead?").

Examples of Questions to REJECT under this rule:
Question: Compared to Hits@10, Hits@1 could be more critical in the real-world applications,
     especially for tail nodes with very few neighbors. I wonder if the authors can also provide the
     Hits@1 performance.
Question: Would the method fail if using a non-contrastive pre-trained model?
The paper mainly focuses on 4-bit and 5-bit quantization, leaving questions about the performance
     and relevance of other bit quantizations



**Rule 8: Exclude questions framed as unsupported suggestions**
 REJECT questions like Did you consider X? if they are isolated and not grounded in the papers
     content, especially if surrounded by uninformative praise or vague critique.




Make sure to be strict so that no poor quality question passes through.
```

```
**Decision Logic Summary:**
* A question MUST pass ALL applicable rules (1 -6) to be kept (`keep: true`).
* Failure at any rule stage leads to rejection (`keep: false`).
```

A.17.3 Question Generation

The prompt shown below was used uniformly across all models for question generation.

Listing 3: Prompt for Question Generation

```
{"role": "system", "content": "You are expert at asking unique questions based on the OCR text of a
    research paper. So given the text, generate one high quality question now."},
{"role": "user", "content": f"Here's the text of the complete research paper and now generate a
    question based on it. \n{ocr_output}"}
```

A.17.4 Extraction of Questions

Listing 4: System Prompt for Question Extraction

```
"""You are a highly experienced professor from Stanford University with extensive experience in
    reviewing and publishing research papers. You will be provided with a peer review containing a
    heading called Questions and another section called Mixed Content. The Questions section
    contains multiple questions without any indication/ separator for a new question and the Mixed
    Content has a mix of questions that might not have a ? to indicate a question. It can simply be
    a suggestion, an edit, a clarification required from the author etc.


Task: Your Primary task is to Extract Questions first from the Questions section and then from the
    Mixed Content section. Perform verbatim extraction. I.e. Word-for-Word
By Questions I mean all the questions  explicitly or implicitly asked  that the author needs to
    answer the reviewer based on the review text.


1) Extract all the questions from the Questions section in a way all the sentences are retained. Do
    not miss any sentence or words from the original content in the section and output multiple
    Questions you have found, you need to break the Questions properly. If someone concatenates the
    multiple questions you have formed, they must get the Questions section as it is.
2) While breaking the questions from the Question section, you might encounter nested questions. If
    both the parts are related keep them as a single question but if one part is an independent
    question, make them as separate questions.
3) Extract all the questions that are present in the Mixed Content section. The questions might not
    be direct, it might include the reviewer telling what made him arrive at this question and then
    pose the question. It can also be some clarification he/she needs from a content in the paper.
    So include the complete context and dont simply output just the question.
4) In some cases, the Questions section will direct you to refer the Mixed Content section by asking
    you to refer the weakness. That simply is your hint to find questions in the Mixed Content
    section.
5) The "Mixed Content" section might have general observations or weaknesses of the paper, so only
    pick up questions,reviewer's suggestion for edits, reviewer seeking clarification BUT dont
    include general observations. This is the rule for "Mixed Content" section.


Note: The Questions section will always have question present in it until unless it is blank or only
    asking you to refer to the weakness. The Mixed Content section might or might not have
    questions in it, so check very carefully. Learn from the zero-shot example below.
Note 2: Important: When the questions that you form from "Questions" section are concatenated, it
    should form the original and complete content of the "Questions" section. This rule of
    concatenation is important and ONLY for "Questions" Section ONLY.

Remember: Your task is just extraction of Questions and Not Rephrasing.


###Output
Questions: [
{
\"Paper_id\": <ID of Paper>,
\"review_id\": <ID of review>,
\"Q_Number\": <Index of generated question>,
```

```
\"Question\": <Extracted question>
},
]




Example 1:


Input:


Paper Id: : Asdho34
Review Id: ioedh45
Questions :
I have questions about the learning process of the 11 conv layer in equation (5). How is it exactly
    trained? And is it sensitive to the training sample size?\
- Will instance normalization also work in text-to-image tasks? It will be interesting to see if it
    could generate higher fidelity images with semantic meaning more aligned with the provided text
    prompts


Mixed Content :
The proposed method is a systematic approach for image translation tasks incorporating different
    components. A potential drawback is its inference speed. It would be beneficial if the authors
    could compare inference speed with other image translation tasks.\
- The comparison with methods like SDEdit, Prompt2Prompt, and InstructPix2Pix is somehow unfair
    since they do not require an additional segmentation network.\
- The quantitative evaluation is only the proposed dataset, which contains fine-grained edit
    instructions. The effectiveness of DVP could be further proved by evaluating simple or even
    ambiguous instructions


Overall, the paper is well-organized and easy to follow. The figures and tables are informative.\
\
- The performance of the proposed method is promising. Figures 4, 6 clearly demonstrate the
    superiority of DVP.\
\
- The ablation study and system analysis are clear and informative, making it easy to see the
    effectiveness of different parts, such as instance normalization, and prompte.




Output:
{
Questions: [
{
\"Paper_id\": Asdho34,
\"review_id\": ioedh45,
\"Q_Number\": 1,
\"Question\":  I have questions about the learning process of the 11 conv layer in equation (5). How
    is it exactly trained? And is it sensitive to the training sample size?
},


{
\"Paper_id\": Asdho34,
\"review_id\": ioedh45,
\"Q_Number\": 2,
\"Question\":  Will instance normalization also work in text-to-image tasks? It will be interesting
    to see if it could generate higher fidelity images with semantic meaning more aligned with the
    provided text prompts
},
{
\"Paper_id\": Asdho34,
\"review_id\": ioedh45,
\"Q_Number\": 3,
```

\"Question\": A potential drawback is its inference speed. It would be beneficial if the authors could compare inference speed with other image translation tasks
},

{
\"Paper_id\": Asdho34,
\"review_id\": ioedh45,
\"Q_Number\": 4,
\"Question\": The quantitative evaluation is only the proposed dataset, which contains fine-grained edit instructions. The effectiveness of DVP could be further proved by evaluating simple or even ambiguous instructions

}
]

}

Example 2

Input:

Paper Id: : Asdho34
Review Id: ioedh45
Questions :
Please comment on the weaknesses outlined above.\
- Figures 10 and 11, right: Why is adaptation slower for OC-GFN than GFN in the first few thousand iterations? This is surprising since one would hope pretraining helps bootstrap downstream performance as in vision / language / RL. If its an exploration phase, did you validate it and is there a way to side-step it?

Mixed Content :
There should be a discussions of assumptions behind the OC-GFNs pretraining. Namely, that transfer is only possible when the reward function changes but not if the action-space or the state-space change. Moreover, the goal-conditioning requires a well specified set of outcomes Y presumably not all states s are terminal states which makes the proposed method not truly unsupervised. These limitations (together with the applicability mentioned at the end of A.2) could be stated explicitly in the main text, and left to future work.\
- While there are enough benchmarks, I believe none include continuous action/state spaces. Moreover, the experiments only one GFN variant the detailed-balance one, which is also used for OC-GFN. It would help validate the generality of OC if we had experiments showing it worked on these different settings. Moreover, Id be curious to know how other pretrained amortized sampling baselines (eg, VAEs, normalizing flows) fare against OC-GFN \xa0and what about pretraining a GFN on task A (without OC) and fine-tuning it on task B?\
- (minor) The second and fourth paragraphs of Section 4.2 mention the reasoning potential of GFNs, and that intractable marginalization leads to slow thinking. Are these anthropomorphisms really needed for this paper?\
- (minor) I wished the preliminaries (Section 2) included a training objective like Eq. 5 & 9, and that these more clearly specified which are the optimization variables.\
- Some typos, there maybe more:\
- p. 3: multi-objective what?\
- p. 4: given a reward R a posterior as a function\
- p. 4: autotelicly autotelically?\
- p. 5: in log-scale obtained from Eq. (5) should be Eq. 4?'

The exposition is generally clear, and I enjoyed reading the paper. The authors first present the goal-conditioning idea and how it applies to GFNs, then walk the reader through their derivation and assumptions for amortized adaptation. I especially appreciated Section 2 which gave a clear and concise background.\
- The paper tackles an impactful problem for GFNs. While the pretraining solution is not particularly novel, its a neat application of goal-condition RL to an amortized sampling problem. The authors also figured out how to make it work on a wide range of problems, and

```
    provide several ablations in the main text and the appendix.\
- The insight that a new sampling policy can be readily obtained from an outcome-conditioned flow is
    neat and, as far as I can tell, novel. This could spawn interest in outcome-conditioned flows
    and different ways to amortize Eq. 6.




Output:
{
Questions: [
{
\"Paper_id\": Asdho34,
\"review_id\": ioedh45,
\"Q_Number\": 1,
\"Question\":  Please comment on the weaknesses outlined above.\
- Figures 10 and 11, right: Why is adaptation slower for OC-GFN than GFN in the first few thousand
    iterations? This is surprising since one would hope pretraining helps bootstrap downstream
    performance as in vision / language / RL. If its an exploration phase, did you validate it and
    is there a way to side-step it?

},


{
\"Paper_id\": Asdho34,
\"review_id\": ioedh45,
\"Q_Number\": 2,
\"Question\":  There should be a discussions of assumptions behind the OC-GFNs pretraining. Namely,
    that transfer is only possible when the reward function changes but not if the action-space or
    the state-space change
},
{
\"Paper_id\": Asdho34,
\"review_id\": ioedh45,
\"Q_Number\": 3,
\"Question\":  These limitations (together with the applicability mentioned at the end of A.2) could
    be stated explicitly in the main text, and left to future work.
},


{
\"Paper_id\": Asdho34,
\"review_id\": ioedh45,
\"Q_Number\": 4,
\"Question\":  While there are enough benchmarks, I believe none include continuous action/state
    spaces. Moreover, the experiments only one GFN variant  the detailed-balance one, which is also
    used for OC-GFN. It would help validate the generality of OC if we had experiments showing it
    worked on these different settings
},







{
\"Paper_id\": Asdho34,
\"review_id\": ioedh45,
\"Q_Number\": 5,
\"Question\":  Id be curious to know how other pretrained amortized sampling baselines (eg, VAEs,
    normalizing flows) fare against OC-GFN \xa0and what about pretraining a GFN on task A (without
    OC) and fine-tuning it on task B?
},


{
\"Paper_id\": Asdho34,
\"review_id\": ioedh45,
\"Q_Number\": 6,
```

```
\"Question\":  The second and fourth paragraphs of Section 4.2 mention the reasoning potential of
    GFNs, and that intractable marginalization leads to slow thinking. Are these anthropomorphisms
    really needed for this paper?
},
{
\"Paper_id\": Asdho34,
\"review_id\": ioedh45,
\"Q_Number\": 7,
\"Question\":  I wished the preliminaries (Section 2) included a training objective like Eq. 5 & 9,
    and that these more clearly specified which are the optimization variables},


{
\"Paper_id\": Asdho34,
\"review_id\": ioedh45,
\"Q_Number\": 8,
\"Question\":  Some typos, there maybe more:\
- p. 3: multi-objective what?\
- p. 4: given a reward R a posterior as a function\
- p. 4: autotelicly  autotelically?\
- p. 5: in log-scale obtained from Eq. (5) should be Eq. 4?
},




]


}"""
```