Uncertainty-Aware Generation and Decision-Making Under Ambiguity

Anonymous ACL submission

Abstract

Disclaimer: We do not promote automation of the peer-review process but aim to support human reviewers and authors.

With rapidly improving capabilities, Large Language Models (LLMs) are increasingly used in many complex real-world tasks. Beyond requiring in-depth knowledge and reasoning skills, many of these tasks exhibit a high degree of subjectivity and require that the outputs of the model can be trusted. While a lot of progress has been made to train better models, decision-making algorithms have received less attention. In this work, we present and evaluate various uncertainty-aware decision-making algorithms based on Bayesian decision theory and risk-averse decision making on the tasks of tutoring and automatic peer reviewing. Concretely, we take uncertainty over tutoring strategies and review scores into account when generating a tutor response or review and use conformal prediction to provide guarantees over strategy and score. We find empirically that these algorithms can improve the utility of the generations but need to be carefully implemented when ambiguity is high. For example, risk-averse rules can degrade performance by optimizing for generic outputs, while Bayesian methods tend to perform better. Our work uses techniques from decision theory to improve LLM-based decision-making and outlines open challenges for the community.[1]

1 Introduction

By now, LLMs are ubiquitously used, including in applications that were deemed out of reach just years ago because they are sensitive and require expert-level knowledge. One example is tutoring, where LLMs have now already found their way into real-world applications (LearnLM Team et al., 2025). Another example is peer review, where several sub-communities in machine learning research are adopting LLM-based tools to support authors and reviewers. NeurIPS 2026 uses the tool of Cohen-Addad and Woodruff (2025) to provide LLM-generated reviews to authors before submission. The tool is based on Gemini-2.5 (Comanici et al., 2025) and has received overwhelmingly positive feedback in a pilot at the STOC 2026 conference. ACL Rolling Review already uses the Revas tool to support reviewers.[2] Overall, LLMs are increasingly used in many such high-stakes domains.

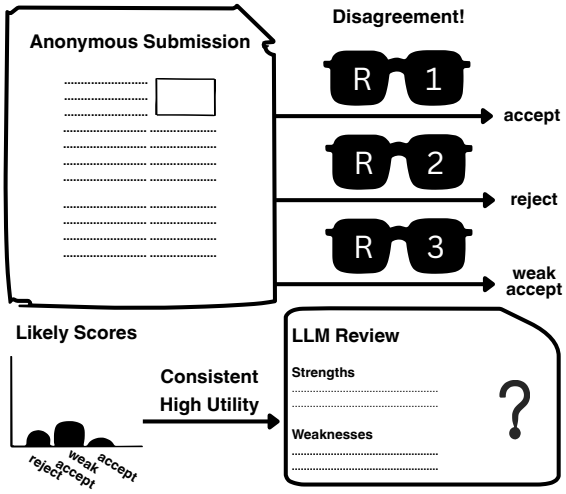


Figure 1: In many real-world tasks there is a large amount of uncertainty over decisions and even humans disagree significantly. For example, in reviewing human reviewers often disagree about a paper's scores. We argue that automatic methods for supporting humans in these tasks should take this uncertainty into account to generate high-utility outputs while narrowing down plausible ranges of intermediate decisions.

These domains bring up a list of challenges that need to be taken care of. For example, if students are told solutions too early their learning can be slower than with active learning ap-

[1] Code will be released publicly.

[2] https://revas.mbzuai.ac.ae/

proaches that allow exploration by the student (Chi and Wylie, 2014). In peer review, incorrect or imprecise feedback might lead authors to add revisions that are against conference guidelines, and misleading scores could lead to authors submitting early versions of works that are very unlikely to be accepted and overload peer review even more.

However, addressing these challenges is often hard for both humans and models. One reason for that is that both these settings involve multiple decision steps. A teacher needs to find out where the student made a mistake, and then pick a strategy to address it. A reviewer needs to assess many nuanced points about a paper and then provide both a review and a score. In addition, both settings are highly ambiguous and even different human teachers or reviewers will pick fundamentally different (but valid) strategies or scores and might even contradict each other (Kumar et al., 2023).

A variety of works have been proposed that change the model, for example, to split tasks into multiple sequential reasoning steps before a final answer generation. In tutor response generation this usually entails finding a student mistake, picking a strategy and responding (Wang et al., 2024; Daheim et al., 2024; Hsu et al., 2026). In reviewing, multiple steps might be identifying novelty (Afzal et al., 2026) and finding suggestions to present (Zhu et al., 2025). The same is true in related domains like NLP for mental health, where intermediate reasoning can help in applications like reframing (Goel et al., 2025). Other works try to improve models via reinforcement (Dinucu-Jianu et al., 2025) or supervised learning (Idahl and Ahmadi, 2025) or use multi-agent systems (Weng et al., 2026). Many of these problems can be formulated as the prediction of a state and response but common decision-making algorithms usually neither account for ambiguity over states nor can provide any guarantees over them. This often limits thir real-world utility, for example, because they can not narrow down plausible score ranges well.

Here, we present decision-making and generation algorithms than can both account for uncertainty and provide statistical guarantees and highlight various open challenges for the community. First, we use a Bayesian approach that maximizes an expected utility for an LLM generation with respect to a distribution over such a state. In tutoring, this state is the tutoring strategy and in reviewing this is the reviewer score. We use conformal prediction to provide a statistical guarantee over this state, which is also used to prune the number of states that are considered for utility estimation. Finally, we use a risk-averse strategy (Kiyani et al., 2025) that maximizes the utility of the worst-case state in this conformal prediction set. These strategies can be used to improve the predictions of any model that predicts both state and response.

We evaluate these methods on the tasks of review generation and scoring as well as on the task of tutor response generation in connection with tutoring strategies. For generation, we use Gemma-3-27B (Team et al., 2025), Qwen3-30B (Yang et al., 2025) and Mistral-3.2-24B (Mistral AI, 2025) for reviewing, and TutorRL-7B for tutoring (Dinucu-Jianu et al., 2025). Across multiple datasets, namely, Review-5K (Weng et al., 2025), NLPEER (Dycke et al., 2023), and MathDial (Macina et al., 2023a) we find that these methods can increase utility of reviews and tutor responses. We also find that Bayesian strategies seem to be more robust to the quality of state distribution and that meaningful prediction sets depend on both classifier performance and how ambiguity is handled. Overall, our work outlines methods for uncertainty-aware decision-making with LLMs as well as open problems for the community.

2 Background

2.1 Tutor Response Generation

The goal of tutor response generation is to generate a response $\mathbf{y} = (y_1, \ldots, y_T) \in \mathcal{V}^*$ from a vocabulary $\mathcal{V}$ given a set of inputs that we will refer to as $\mathbf{x} \in \mathcal{V}^*$ for generality. Usually, these inputs include the previous dialog history, supporting knowledge, an incorrect solution step, and a given problem. The amount of information that the LLM has to process in order to solve the task already hints that the task is quite hard to solve directly. Just prompting an LLM with $\mathbf{x}$ as the input and sampling $\mathbf{y} \sim p(\cdot \mid \mathbf{x})$ often leads to the LLM providing incorrect feedback (Macina et al., 2023b) or telling the solution to the student early on, which can be detrimental to their learning. Other strategies should be preferred that provide the student with room to explore possible problem solutions in order to foster learning (Lepper and Woolverton, 2002; Nye et al., 2014), sometimes going as far as letting the student fail proactively in order to increase exploration (Kapur, 2008).

To reduce the amount of incorrect feedback and improper tutoring strategies, many recent works

model tutoring as a multi-step problem. For example, Wang et al. (2024) classify the type of error a student has made given a pre-defined taxonomy of errors and then predict a tutoring strategy and intent. Daheim et al. (2024) propose to first verify the student solution attempt with free-text reasoning created by an LLM that is then used to produce a response. Another example is the work by Puech et al. (2025) who trace student solutions and then classify an intent before generating a response. Such strategies have been adopted in a variety of works (Hsu et al., 2026; Park et al., 2026) but, notably, are prone to propagating errors in initial steps.

Other works introduce training and data collection strategies to improve tutoring models, for example, using reinforcement learning (Dinucu-Jianu et al., 2025) or by collecting synthetic data (LearnLM Team et al., 2025). Finally, methods have been proposed to evaluate how well the responses generated by a tutor follow pedagogical principles (Macina et al., 2025).

2.2 Automated Methods in Peer Review

Similar observations have been made in the area of automated peer review. There, the goal is to generate a review $\mathbf{y} \in \mathcal{V}^*$ given a context $\mathbf{x}$ that usually consists of a research paper and, perhaps accompanying materials. Oftentimes, a review score is also predicted which can be useful, for example, for authors to understand their chances of acceptance. Our aim is to explore such review generation for peer review support, for example, to provide meaningful critiques to authors before submission or help reviewers write better reviews.

Due to its complexity, recent works also split reviewing into multiple steps. For example, Weng et al. (2025) generate multiple reviews sequentially using the same LLM that are then summarized to mimick a senior reviewer. Another example is the work by Zhu et al. (2025), who first identify novelty, then generate and verify reviews, and then generate a meta-review and score. Recently, such multi-step systems have also been modeled with multiple agents that might take on specific tasks and roles (Gao et al., 2025; Weng et al., 2026).

Other works look at some of these steps in isolation. For example, Afzal et al. (2026) try to identify novelty in scientific papers based on a multi-step procedure and Purkayastha et al. (2026) focus on modeling meta reviewing. Recently, there have also been works to evaluate scientific reviews according to a set of pre-defined criteria (Sadallah et al., 2025; Şahinuç et al., 2026).

At the same time, decision-making and generation algorithms have received less attention in both domains. As a consequence, while there are models and approaches to understand the utility of a review, it is not maximized. Furthermore, uncertainty over decisions, for example, over the right score or tutoring strategy is not modeled explicitly. Errors may be propagated and there are no indications of error rates which can endanger user trust (Dhuliawala et al., 2023; Palod et al., 2026). Ideally, we would like a decision-making algorithm that both accounts for uncertainty and provides statistical guarantees over states to increase trust.

2.3 Uncertainty-Aware Methods in NLP

Many works have been proposed to account for utility and uncertainty, and to provide guarantees, but these are often believed to be conflicting. As a consequence, methods usually provide one or the other, but not both at the same time.

Minimum Bayes Risk decoding (Kumar and Byrne, 2004) is one method to account for utility in the sense of a comparison between one output $\mathbf{y}$ and another output $\mathbf{y}'$ via a utility function $u(\mathbf{y}, \mathbf{y}')$ and the following decision rule,

$$\mathbf{y}^* = \arg\max_{\mathbf{y} \in \mathcal{V}^*} \sum_{\mathbf{y}' \in \mathcal{V}^*} p(\mathbf{y}' \mid \mathbf{x}) \cdot u(\mathbf{y}, \mathbf{y}'). \quad (1)$$

The objective is usually approximated by a collection of samples, which we will denote with $\mathcal{H}$, for example, using Monte-Carlo methods (Eikema and Aziz, 2022) or the model probability directly (Jinnai et al., 2024), and can be extended to ensembles (Daheim et al., 2025). However, the choice of u is not easy and the method scales quadratically with the size of $\mathcal{H}$, though methods to reduce this exist (Eikema and Aziz, 2022; Cheng and Vlachos, 2023). To address these, we here adopt a similar strategy but target a different decision problem that does not compare different $\mathbf{y}$. The method is similar to quality-aware decoding (Fernandes et al., 2022) and best-of-n sampling (Stiennon et al., 2020) and a similar formulation has been explored by (Johnson et al., 2023) for code generation.

Split conformal prediction Vovk et al. (2005) provides what is referred to as marginal coverage for a classifier, which means that

$$P(\widehat{\mathbf{y}} \in \mathcal{C}(\mathbf{x})) \geq 1 - \alpha, \quad (2)$$

given a threshold α, ground-truth label $\widehat{\mathbf{y}}$ and prediction set $\mathcal{C}(\mathbf{x})$ for a random (not fixed) $\mathbf{x}$ in a test

set. In words, for exchangeable data (see Campos et al. (2024, Sec. 3.1) for a precise definition), with probability at least $1 - \alpha$ the correct label is contained in $\mathcal{C}(\mathbf{x})$. The prediction set is constructed using a a threshold determined on a separate validation set with the same classifier. The formulation has been adopted and built upon in many general machine learning and NLP works (Angelopoulos and Bates, 2023; Campos et al., 2024).

Historically, there is a lot of discussion about whether Bayesian methods and frequentist methods like conformal prediction are reconcilable, see e.g. Robert (2007). Here, we do not intend to add to this discussion but intend to show that combining such techniques can achieve many real-world goals when applying LLMs to complex tasks.

3 Bayes Decision Rule

We discuss decision problems with two components. The first component is a decision space $\mathcal{V}^*$ that coincides with the space of all strings that can be generated using a vocabulary $\mathcal{V}$ and contains reviews for academic works and responses of tutors. The second component is a finite, discrete, and problem-specific set Ω. For example, in tutoring this is a set of tutoring strategies and in reviewing this could be possible scores according to a scoring guideline of a conference. That is, in reviewing each $\omega \in \Omega$ is a specific score. The decision is taken given $\mathbf{x} \in \mathcal{V}^*$ which is contextual information like the paper in peer review or dialog context in tutoring. In both cases, we aim to find a final output $\mathbf{y}^* \in \mathcal{V}^*$ that should have high utility and account for uncertainty over choices of $\omega \in \Omega$ which often arises due to ambiguity. How utility is defined is problem-specific and, ideally, we would like to empower practitioners to be able to define and use their own criteria.

We argue that Bayesian decision theory provides a rule that satisfies these desiderata for $\mathbf{y}^*$. The rule is to choose the output $\mathbf{y}^*$ that minimizes the Bayes risk or, equivalently, maximizes expected utility (DeGroot, 2004, Sec. 8.2), as follows,

$$\mathbf{y}^* = \arg\max_{\mathbf{y} \in \mathcal{V}^*} \sum_{\omega \in \Omega} u(\mathbf{y}, \omega) \cdot p(\omega \mid \mathbf{x}). \quad (3)$$

The corresponding $\mathbf{y}^*$ is then called Bayes decision against the distribution $p(\omega \mid \mathbf{x})$. The rule (in expectation) guarantees optimality wrt. u and $p(\omega \mid \mathbf{x})$ but does not guarantee optimal decisions for another utility function. It is important to choose it wisely to avoid metric bias (Kovacs et al., 2024).

Dissecting Eq. (3) shows that $u(\mathbf{y}, \omega)$ is a function that depends on both a specific choice $\omega \in \Omega$ and a specific output $\mathbf{y} \in \mathcal{V}^*$ which can be used to encode both quality and consistency criteria. Concretely, in tutoring we can give a score that indicates for each tutor response $\mathbf{y}$ whether it follows sound pedagogical principles and whether it is consistent with a specific tutoring strategy ω. Responses that are consistent with likely tutoring strategies are preferred, and so are responses that follow sound pedagogical principles. Crucially, we do not make a hard decision according to a specific tutoring strategy but actively reduce uncertainty over them via marginalization.

In Eq. (3), there is no notion of output sequence probabilities but these can be incorporated by using Bayes theorem. For example, we may rewrite

$$p(\omega \mid \mathbf{x}) \propto \sum_{\mathbf{y} \in \mathcal{V}^*} p(\omega \mid \mathbf{y}, \mathbf{x}) \cdot p(\mathbf{y} \mid \mathbf{x}) \quad (4)$$

to include $p(\mathbf{y} \mid \mathbf{x})$. Many choices can be made by the practitioner that can have different advantages in practice. These choices are important, because *Eq.* (3) can not be calculated exactly due to expectations and/or maximization over the countably-infinite space $\mathcal{V}^*$ and corresponding estimators do not need to yield the same decisions, even if they estimate equivalent rewritings. One option is to evaluate $p(\omega \mid \mathbf{x})$ directly and sample $\mathcal{H} = (\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(M)} \mid \mathbf{y}^{(i)} \sim p(\cdot \mid \mathbf{x}))$ to restrict the maximization to a promising subset,

$$\widehat{\mathbf{y}}^* = \arg\max_{\mathbf{y} \in \mathcal{H}} \sum_{\omega \in \Omega} u(\mathbf{y}, \omega) \cdot p(\omega \mid \mathbf{x}). \quad (5)$$

The formulation is similar to n-best sampling (Stiennon et al., 2020) but uses a smoothed n-best calculation that reduces uncertainty over an additional state space Ω via $p(\omega \mid \mathbf{x})$. Another option is to use conditional distributions by either conditioning the LLM on a categorical output or conditioning the model of Ω on outputs. For example, we could generate responses for specific tutoring strategies. These could use a Monte-Carlo approximation of the sum in Eq. (4) but it is also possible to use the LLM probabilities directly (Jinnai et al., 2024).

The method is clearly related to Minimum Bayes Risk decoding (Goel and Byrne, 2000) but both aim to minimize the Bayes risk of fundamentally different decision problems. Usually, in MBR the space Ω is chosen as $\Omega = \mathcal{V}^*$ and as a consequence the utility function compares two different output

strings. While this is valid and also successful in practice for domains like machine translation (Freitag et al., 2022), there are also issues. It is often hard to define $u(\mathbf{y}, \mathbf{y}')$ and, indeed, multiple works instead use pointwise utility functions that do not account for uncertainty via marginalization anymore (Fernandes et al., 2022). These can also be used to prune collections $\mathcal{H}$ as MBR has, in general, a quadratic complexity in $|\mathcal{H}|$. Eq. (3) has much lower complexity if $|\Omega| << \mathcal{H}$. Our method is also inspired by Johnson et al. (2023), who use a decision rule like Eq. (3) to find good code suggestions. We build upon this work and show further connections to other uncertainty-aware methods and expand the discussion to general problems.

4 Conformal Prediction & Risk Aversion

The rule in Eq. (3) tells us how to choose $\mathbf{y}^*$ but is perhaps less informative about Ω. The Bayes decision for selecting ω^* wrt. classification accuracy is to pick the maximizer of $p(\omega \mid \mathbf{x})$ but presenting only the corresponding ω^* does not provide a practitioner with an uncertainty estimate over the entire space and the distribution $p(\omega \mid \mathbf{x})$ might or might not be informative. For example, we found that Qwen3-14B assigned probabilities close to 1 for one class and close to 0 for other classes when prompted zero-shot in review score prediction. Ideally, an author submitting their paper to an automatic review before conference submission would like to know plausible intervals of scores to understand the acceptance chances of their paper. Such a method would also help the community, as papers that are likely to receive lower scores might be less likely to be submitted which could reduce the overall number of submissions. For tutoring, it would help, for example, to understand common patterns in tutoring strategies over a full dialog.

Conformal Prediction provides such a method which produces prediction sets $\mathcal{C}(\mathbf{x})$ that, measured over a random test set, cover the true labels with probability at least $1-\alpha$. In split conformal prediction (Vovk et al., 2005), this is achieved by splitting the data into a training set to obtain a classifier over Ω and a validation set that is used to determine a threshold τ based on a non-conformity score, for which we use $1 - p(\omega \mid \mathbf{x})$. Still, there are some issues with split conformal prediction in our setting, because it assumes that data is exchangeable and that one ground-truth label exists. These assumptions are violated, for example, when data distributions drift and data is not identically distributed. Importantly, the assumptions also may not hold when the label is ambiguous (Stutz et al., 2023) under human label variation (Plank, 2022).

Arguably, this is the case in all of the applications we discuss: different reviewers score the same paper differently and different teachers prefer different but similarly effective tutoring strategies. In such a case, standard conformal prediction may not provide the same $1-\alpha$ coverage. Instead, Stutz et al. (2023) propose Monte-Carlo conformal prediction which uses an augmented set for calibration, where the score $1 - p(\omega \mid \mathbf{x})$ is calculated multiple times for same input $\mathbf{x}$ but for different ω. These can either be based on an existing set of labelings or on samples from an estimated label distribution with a slight modification of the quantile definition. Then, coverage can only be guaranteed wrt. this distribution and theoretical coverages are widened to $1-2\alpha$, though empirically a tighter $1-\alpha$ is often observed. We choose this strategy whenever multiple labels are available and otherwise fall back to using one label. We emphasize this as an important problem for the community. Instead of only a single label, multiple labels should be collected to be able to explicitly consider label variation.

Here, we use conformal prediction to find $\mathcal{C}(\mathbf{x}) \subseteq \Omega$ that can be used to reduce the number of utility calculations in Eq. (3) if $|\mathcal{C}(\mathbf{x})| < \Omega$ and provide a set of plausible scores or tutoring strategies that can be used for further insights. This is similar to Cheng and Vlachos (2023) but uses calibrated scores. Applied to Eq. (3) this means,

$$\mathbf{y}^* = \arg\max_{\mathbf{y} \in \mathcal{V}^*} \sum_{\omega \in \mathcal{C}(\mathbf{x})} u(\mathbf{y}, \omega) \cdot p(\omega \mid \mathbf{x}). \quad (6)$$

The objective also bridges to the risk-averse decision rule from Kiyani et al. (2025, Eq. 8) if $p(\omega \mid \mathbf{x})$ is chosen to be a Dirac distribution that places all mass on the worst-case $\omega \in \mathcal{C}(\mathbf{x})$ in the sense of the utility u. The corresponding minmax decision is given by the decision rule

$$\mathbf{y}^* = \arg\max_{\mathbf{y} \in \mathcal{V}^*} \min_{\omega \in \mathcal{C}(\mathbf{x})} u(\mathbf{y}, \omega) \quad (7)$$

and can be seen as maximizing a worst-case utility. For example, in reviewing this maximizes the utility of a worst-case score to reduce the chance of presenting a poor-quality review to the user. Again, the same estimators as discussed in Sec. 3 can be used to narrow down promising examples for the maximization over $\mathcal{V}^*$.

Method	ICLR Actionable	ICLR Verifiable	ICLR Helpful	ICLR Grounded	NLPEER Actionable	NLPEER Verifiable	NLPEER Helpful	NLPEER Grounded
				Gemma3-27B				
Sampling	1.99$\pm$0.008	2.44$\pm$0.007	2.80$\pm$0.002	3.02$\pm$0.005	1.89$\pm$0.018	2.36$\pm$0.013	2.74$\pm$0.004	2.92$\pm$0.011
Bayes	2.13$\pm$0.009	2.48$\pm$0.005	**2.84$\pm$0.003**	3.06$\pm$0.004	**2.15$\pm$0.020**	**2.45$\pm$0.013**	**2.82$\pm$0.012**	**3.05$\pm$0.011**
+ Conformal	**2.14$\pm$0.005**	2.48$\pm$0.007	**2.84$\pm$0.002**	**3.07$\pm$0.005**	2.13$\pm$0.018	**2.45$\pm$0.011**	**2.82$\pm$0.008**	3.04$\pm$0.010
Minmax	2.04$\pm$0.014	**2.49$\pm$0.010**	2.83$\pm$0.013	**3.07$\pm$0.011**	1.93$\pm$0.017	2.39$\pm$0.012	2.77$\pm$0.008	3.00$\pm$0.019
				Mistral3-24B				
Sampling	1.92$\pm$0.017	2.37$\pm$0.012	2.77$\pm$0.003	2.68$\pm$0.007	1.79$\pm$0.030	2.30$\pm$0.016	2.70$\pm$0.014	2.63$\pm$0.010
Bayes	**2.26$\pm$0.021**	**2.53$\pm$0.010**	2.86$\pm$0.015	**2.77$\pm$0.007**	**2.18$\pm$0.015**	**2.53$\pm$0.012**	**2.85$\pm$0.008**	**2.77$\pm$0.016**
+ Conformal	**2.26$\pm$0.019**	**2.53$\pm$0.005**	**2.87$\pm$0.014**	**2.77$\pm$0.006**	2.14$\pm$0.019	2.51$\pm$0.009	2.84$\pm$0.010	2.76$\pm$0.015
Minmax	2.13$\pm$0.024	2.44$\pm$0.015	2.82$\pm$0.013	**2.77$\pm$0.009**	1.93$\pm$0.035	2.30$\pm$0.016	2.69$\pm$0.022	2.65$\pm$0.009
				Qwen3-30B				
Sampling	1.59$\pm$0.030	2.84$\pm$0.009	2.77$\pm$0.011	3.28$\pm$0.018	1.48$\pm$0.009	2.77$\pm$0.038	2.77$\pm$0.033	3.09$\pm$0.020
Bayes	**1.69$\pm$0.014**	2.88$\pm$0.011	2.80$\pm$0.005	3.32$\pm$0.016	**1.55$\pm$0.025**	2.84$\pm$0.049	**2.86$\pm$0.037**	**3.16$\pm$0.020**
+ Conformal	1.68$\pm$0.015	2.87$\pm$0.011	2.79$\pm$0.003	3.31$\pm$0.015	1.53$\pm$0.016	2.84$\pm$0.060	**2.86$\pm$0.046**	3.15$\pm$0.036
Minmax	1.67$\pm$0.006	**2.96$\pm$0.004**	**2.82$\pm$0.002**	**3.36$\pm$0.015**	1.35$\pm$0.029	**2.89$\pm$0.010**	2.84$\pm$0.009	3.05$\pm$0.006

Table 1: We compare Eq. (3) (Bayes) with and without a pruning via a conformal prediction set to Eq. (7) (Minmax) on ICLR and ARR reviews. We use the reward models from Şahinuç et al. (2026) that assign a score from 1-4 to each aspect. We find that both methods provide improvements, with Bayes usually performing better, except for the case of Qwen3, where Minmax provides stronger improvements in some criteria. Averaged over 3 seeds.

5 Experiments

We conduct experiments on review generation and scoring as well as on tutor response generation and tutoring strategy prediction. Both times, we compare sampling against the Minimum Bayes Risk objective from Eq. (3) with and without a conformal set pruning and the minmax objective from Eq. (7).

We use MathDial (Macina et al., 2023a) for tutoring and NLPEER (Dycke et al., 2023) and ICLR submissions from Review-5k (Weng et al., 2025) for reviewing. We use google/gemma-3-27b-it (Team et al., 2025), Qwen/Qwen3-30B-A3B-Instruct-2507 (Yang et al., 2025), and mistralai/Mistral-Small-3.2-24B-Instruct-2506 (Mistral AI, 2025) to generate reviews conditioned on the paper and the scoring guidelines from ICLR and ACL Rolling Review. For tutoring, we use TutorRL-7B (Dinucu-Jianu et al., 2025) for generation and prompt it with the previous dialog history, problem, student solution attempt, and ground-truth solution.

Due to finding poor zero-shot prompting performance, we train meta-llama/Llama-3.2-3B-Instruct (Grattafiori et al., 2024) to predict a distribution over Ω. We use $\alpha = 0.05$ for conformal prediction. For u, we use Qwen3-14B as an LLM judge with 5-point Likert-scale ratings defined in App. A. We use an expectation over rubric scores as concurrently proposed by Kwok et al. (2026). We use the reward model from Macina et al. (2025) to evaluate tutor responses and the reward model from Şahinuç et al. (2026) to evaluate reviews. Both are different from the model used to calculate utility and have been shown to outperform prompted LLMs.

All experimental details are in App. A, where we also report the exact prompts that are used for generation, scoring of categorical labels, and utility estimation. Unless otherwise stated, we use $|\mathcal{H}| = 64$ for reviewing and $|\mathcal{H}| = 32$ for tutoring.

6 Results

6.1 Main Results on Reviewing

We first show our main results on the task of review generation and scoring in Tab. 1 on Review-5k and NLPEER, where we use the subset of papers with more than one review for testing. We compare the Bayes decision rule from Eq. (3) to using a conformal pruning as in Eq. (6) and using the minmax rule from Eq. (7). $\mathcal{H}$ is sampled from the LLM with ancestral sampling and temperature 1 and we disable all truncations of token-level distributions.

We use a finetuning of Llama-3.2-3B-Instruct on the training set of Review-5k and papers with one review in NLPEER, as we found that zero-shot prompting of larger models like Qwen3-14B was insufficient for score prediction and, for example, only obtained an accuracy of ca. 24% on NLPEER. In both cases we split 90% for training and 10% for calibration. For reviewing, the score

Method	Reward	Win (Sampl.)	Win (GT)
Sampling	0.858 $\pm$ 0.002	-	71.6 $\pm$ 0.01
Bayes	**0.892** $\pm$ **0.002**	**54.3** $\pm$ **0.09**	**75.4** $\pm$ **0.04**
+ Conformal	**0.892** $\pm$ **0.002**	**54.3** $\pm$ **0.01**	**75.2** $\pm$ **0.04**
Minmax	0.809 $\pm$ 0.003	39.3 $\pm$ 0.03	62.2 $\pm$ 0.07

Table 2: We measure the scalar reward from Macina et al. (2025) as well as the win-rate over sampling and ground-truth responses (GT) on MathDial with TutorRL-7B. Bayes provides improvements in all settings. Minmax shows degradations due to optimizing for the tutoring strategies *generic* and *telling* too often.

Generator	Avg. $\lvert\mathcal{C}\rvert/\lvert\Omega\rvert$	Δ-Minmax	Δ-Bayes
	MathDial		
TutorRL-7B	0.766	-0.049	+0.034
	ICLR		
Gemma-3	0.641	+0.05	+0.15
Mistral-3	0.641	+0.21	+0.34
Qwen-3	0.641	+0.08	+0.09
	NLPEER		
Gemma-3	0.470	+0.04	+0.24
Mistral-3	0.470	+0.14	+0.35
Qwen-3	0.470	-0.13	+0.05

Table 3: We compare the compression ratio of conformal sets $\mathcal{C}$ over the number of possible labels $|\Omega|$ with downstream improvements over sampling in terms of actionability for Eq. (7) (Δ-Minmax) and Eq. (6) (Δ-Bayes). Conformal prediction sets are often wide for highly ambiguous tasks which can degrade performance when using minmax strategies.

distribution $p(\omega \mid \mathbf{x})$ is estimated by predicting the scores of each generated review and averaging all resulting score distributions for the same paper. The resulting classifiers obtained 32.6% accuracy on the Review-5k validation set and 30.1% on the NLPEER validaton set which highlights the large amount of ambiguity and might also be due to the shift from human reviews to generated reviews during prediction. We have also tried using an oracle based on the human reviews which achieved better results (58.6% on Review-5k) but downstream generation metrics did not differ significantly.

In our results, we find that both methods generally give improvements over just sampling responses. Overall, the improvements with the Bayes decision rule seem to be the largest, especially in terms of actionability. We believe that this is due to the prompt used for utility estimation that names actionability first and which might thus be favored. Conformal pruning seems to mostly retain performance at the benefit of requiring fewer utility estimations. The savings are shown in Tab. 3. The minmax rule performs similarly. Actionability is often lower but, interestingly, on Qwen3 in terms of other criteria it performs best on Review-5k.

When comparing models, we find that Gemma3 and Mistral3 perform similarly. Qwen3 is not as good in actionability but better in other criteria which might be due to the post-training of Qwen3 favoring these criteria. Our results show many open avenues for future work. First, future work can look into better LLMs for sampling and better utility estimation. In addition, future work should look into better score prediction methods that might further improve performance.

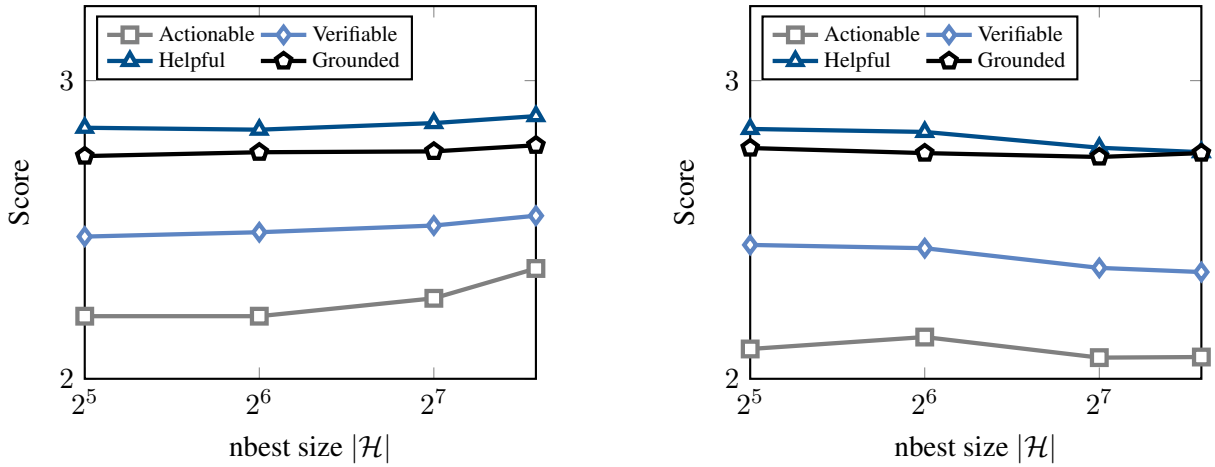
6.2 Main Results on Tutoring

Next, we show results on tutor response generation, again using ancestral sampling for $\mathcal{H}$ without conditioning on a tutoring strategy and using $p(\omega \mid \mathbf{x})$ determined by a finetuned Llama-3.2-Instruct model that we train on 90% of the MathDial training data. The remaining 10% are used for calibration, where the model achieves an accuracy of 47.3% which again shows the high degree of ambiguity. Since we do not have multiple annotations for the same input we use standard split conformal prediction. We think ambiguity should be taken into account in future datasets by collecting multiple plausible tutoring strategies and responses for the same input.

Results are shown in Tab. 2 and show that, this time, only Bayes decision rule improves results over sampling, though all methods improve over the human ground-truth in terms of reward due to the RL training of TutorRL. We have analyzed these results in more detail and attribute these to the fact that tutoring strategies that do not always follow pedagogical principles, namely, telling of the solution and generic answers are contained in the prediction sets over 50% of times. As a consequence, minmax seeks to find the highest reward sequence wrt. these strategies, as these minimize utility. We believe this can be mitigated with stronger strategy predictors but caution that minmax utility seems to depend a lot on the structure of Ω.

6.3 Conformal Prediction Sets

Overall, we find in Tab. 3 that the prediction sets in a task like tutoring, where uncertainty about the correct strategy is high, are often large. While these could be narrowed down by allowing a larger error margin, this would also result in less reliable prediction sets. In reviewing, the prediction sets



(a) Bayes

(b) Minmax

Figure 2: We provide initial results from scaling the size of $\mathcal{H}$ using Mistral3 on ICLR and Eq. (3) (left) and Eq. (7) (right). We find that all criteria are improved for Bayes as $|\mathcal{H}|$ increases, especially towards the largest size tried ($|\mathcal{H}| = 192$), but can not find the opposite trend for minmax which degrades as $\mathcal{H}$ grows.

show a larger compression than in tutoring with cost savings of up to ca. 50%, though especially on NLPEER there is a slight performance degradation when using conformal pruning. There, results using minmax are better, indicating that the quality of prediction sets is key for the method to work well.

6.4 Scaling Behavior

Finally, we evaluate the scaling behavior of Eq. (3) and Eq. (7) in terms of $|\mathcal{H}|$ on Review-5k using Mistral3. We use sizes ranging from 16 to 192 and find that the Bayes decision rule provides consistent improvements as $\mathcal{H}$ grows. The same is not true for minmax decisions, where performance even seems to degrade, potentially due to interactions with too wide prediction sets. Altogether, the experiment provides support for approximations of Bayes decision rule as a method for test-time scaling, which are consistent with prior literature (Daheim et al., 2025; Cong et al., 2025).

7 Conclusion & Future Work

In this work, we have presented uncertainty-aware methods for decision-making, where decisions consist of an LLM generation, for example, a review or a tutor response, and a categorical decision, for example, a reviewer score or a tutoring strategy. These methods either use a Bayesian motivation and average over a distribution over the categorical score to maximize expected utility or take a risk-averse approach and maximize utility with respect to a worst-case score within a conformal prediction set, which can also be used to reduce the sum in the Bayesian case to only important examples and can be useful, for example, to understand an expected score distribution or to how AI tutors navigate students through complex problems. We evaluate these strategies in the domains of automatic reviewing and tutoring using various LLMs. We find that the Bayesian approach always gives improvements while the risk-averse approach can lead to, for example, overly generic responses if the prediction set is wide. Finally, both can empower pracitioners to define own notions of utility.

Overall, there are many open problems for the community to solve towards making these methods work more reliably, even though we find encouraging first results. Current LLMs do not always perform well on hard categorical prediction tasks like review score predictions and are prone to providing distributions that do not directly agree with human label variation. It is also important to account for such label variation when using conformal prediction but this requires actively collecting multiple labelings of the same example, which are not often found in existing datasets. The methods are also expected to benefit from better LLM judges that, among others, can account for long contexts and judge complex user-defined criteria. Finally, we believe that there is still work to be done on making LLMs good samplers for such tasks that also reflect plausible diversity of humans. Our work is therefore best read as a call for action to the community to tackle such hard problems and focus on actively reducing uncertainty during prediction.

Limitations

One limitation of our study is that we do not directly know whether any of the data we use for testing was used for training the models used in this study. The ICLR data is from 2024 and the NLPEER data is from version 1 of the dataset which was released in 2023. All models that were used are newer than this data and, perhaps, have seen at least some of the reviews during training. Therefore, we can at least not guarantee that our results are expected to generalize to new papers that are submitted to, say, a conference in 2026. At the same time, it is necessary to use newer models to ensure that the models perform sufficiently well, otherwise conclusions about a good generation algorithm or decision rule might be driven more by a lack of model capabilities than actual qualities of the algorithms used on top of it.

We do believe that the relevance of our work grows as it is applied to more capable systems than can, perhaps, function as better and more diverse samplers of high-quality reviews. Unfortunately, though, systems like the one used in NeurIPS are proprietary but we call on the community to also focus on generation algorithms and decision rules.

Finally, our study does not clearly evaluate real-world impact, as there is no field study, for example, on providing feedback before real conference submissions. Such studies might be interesting but require careful thought. Similarly, we do not know long-lasting consequences of AI tutor use with a different decision rule, as we could not directly apply our method on real students, which, again, requires careful considerations.

Ethics Statement

In this work, we use publicly available research data from the ACL Anthology, OpenReview, and data that was made publicy available through NLPEER via an explicit consent in the ACL Rolling Review. None of our experiments involve any personal data and only data that has been used extensively in previous research studies. None of the tutoring experiments involve data from any real students as all students in Mathdial were simulated by LLMs and teachers were crowdworkers who gave explicit consent. Our work provides methods that can be used to improve automated methods in both peer review and tutoring. These techniques can bring various benefits to the communities, for example, because they decrease the load of human reviewers, can give authors valuable feedback, and can provide personalized feedback to students. However, we caution that any adoption of these methods must be carefully prepared and designed. We, in particular, do not advocate for using them to automate the peer review process but only to use them as support systems that are followed by human supervision.

References

Osama Mohammed Afzal, Preslav Nakov, Tom Hope, and Iryna Gurevych. 2026. Beyond "not novel enough": Enriching scholarly critique with LLM-assisted feedback. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2648–2671, Rabat, Morocco. Association for Computational Linguistics.

Anastasios N. Angelopoulos and Stephen Bates. 2023. Conformal prediction: A gentle introduction. *Found. Trends Mach. Learn.*, 16(4):494–591.

Margarida Campos, António Farinhas, Chrysoula Zerva, Mário A. T. Figueiredo, and André F. T. Martins. 2024. Conformal prediction for natural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 12:1497–1516.

Julius Cheng and Andreas Vlachos. 2023. Faster minimum Bayes risk decoding with confidence-based pruning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12473–12480, Singapore. Association for Computational Linguistics.

Michelene T. H. Chi and Ruth Wylie. 2014. The icap framework: Linking cognitive engagement to active learning outcomes. *Educational Psychologist*, 49(4):219–243.

Vincent Cohen-Addad and David Woodruff. 2025. Gemini provides automated feedback for theoretical computer scientists at stoc 2026. https://research.google/blog/gemini-provides-automated-feedback-for-theoretical-computer-scientists-at-stoc-2026/. Google Research Blog, accessed May 26, 2026.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3416 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

Bai Cong, Nico Daheim, Yuesong Shen, Rio Yokota, Mohammad Emtiyaz Khan, and Thomas Möllenhoff. 2025. Improving lora with variational learning. *Preprint*, arXiv:2506.14280.

Nico Daheim, Jakub Macina, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2024. Stepwise verification and remediation of student reasoning errors with large language model tutors. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8386–8411, Miami, Florida, USA. Association for Computational Linguistics.

Nico Daheim, Clara Meister, Thomas Möllenhoff, and Iryna Gurevych. 2025. Uncertainty-aware decoding with minimum bayes risk. In *The Thirteenth International Conference on Learning Representations*.

Morris H DeGroot. 2004. *Optimal statistical decisions*. John Wiley & Sons.

Shehzaad Dhuliawala, Vilém Zouhar, Mennatallah El-Assady, and Mrinmaya Sachan. 2023. A diachronic perspective on user trust in AI under uncertainty. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5567–5580, Singapore. Association for Computational Linguistics.

David Dinucu-Jianu, Jakub Macina, Nico Daheim, Ido Hakimi, Iryna Gurevych, and Mrinmaya Sachan. 2025. From problem-solving to teaching problem-solving: Aligning LLMs with pedagogy using reinforcement learning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 272–292, Suzhou, China. Association for Computational Linguistics.

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

Bryan Eikema and Wilker Aziz. 2022. Sampling-based approximations to minimum Bayes risk decoding for neural machine translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10978–10993, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Patrick Fernandes, António Farinhas, Ricardo Rei, José G. C. de Souza, Perez Ogayo, Graham Neubig, and Andre Martins. 2022. Quality-aware decoding for neural machine translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1396–1412, Seattle, United States. Association for Computational Linguistics.

Markus Freitag, David Grangier, Qijun Tan, and Bowen Liang. 2022. High quality rather than high model probability: Minimum Bayes risk decoding with neural metrics. *Transactions of the Association for Computational Linguistics*, 10:811–825.

Xian Gao, Jiacheng Ruan, Zongyun Zhang, Jingsheng Gao, Ting Liu, and Yuzhuo Fu. 2025. Reviewagents: Bridging the gap between human and ai-generated paper reviews. *Preprint*, arXiv:2503.08506.

Anmol Goel, Nico Daheim, Christian Montag, and Iryna Gurevych. 2025. Socratic reasoning improves positive text rewriting. In *Proceedings of the 10th Workshop on Computational Linguistics and Clinical Psychology (CLPsych 2025)*, pages 140–156, Albuquerque, New Mexico. Association for Computational Linguistics.

Vaibhava Goel and William J. Byrne. 2000. Minimum bayes-risk automatic speech recognition. *Comput. Speech Lang.*, 14(2):115–135.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Wei-Ling Hsu, Yu-Chien Tang, and An-Zi Yen. 2026. MathEDU: Feedback generation on problem-solving processes for mathematical learning support. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2883–2901, Rabat, Morocco. Association for Computational Linguistics.

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.

Yuu Jinnai, Tetsuro Morimura, Ukyo Honda, Kaito Ariu, and Kenshi Abe. 2024. Model-based minimum Bayes risk decoding for text generation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 22326–22347. PMLR.

Daniel D. Johnson, Daniel Tarlow, and Christian Walder. 2023. R-u-SURE? Uncertainty-aware code suggestions by maximizing utility across random user intents. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 15262–15306. PMLR.

Manu Kapur. 2008. Productive failure. *Cognition and Instruction*, 26(3):379–424.

Shayan Kiyani, George J. Pappas, Aaron Roth, and Hamed Hassani. 2025. Decision theoretic foundations for conformal prediction: Optimal uncertainty quantification for risk-averse agents. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net.

Geza Kovacs, Daniel Deutsch, and Markus Freitag. 2024. Mitigating metric bias in minimum Bayes risk decoding. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1063–1094, Miami, Florida, USA. Association for Computational Linguistics.

Sandeep Kumar, Tirthankar Ghosal, and Asif Ekbal. 2023. When reviewers lock horns: Finding disagreements in scientific peer reviews. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16693–16704, Singapore. Association for Computational Linguistics.

Shankar Kumar and William Byrne. 2004. Minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 169–176, Boston, Massachusetts, USA. Association for Computational Linguistics.

Jacky Kwok, Shulu Li, Pranav Atreya, Yuejiang Liu, Marco Pavone, Ion Stoica, and Azalia Mirhoseini. 2026. Llm-as-a-verifier: A general-purpose verification framework. Notion Blog.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

LearnLM Team, Abhinit Modi, Aditya Srikanth Veerubhotla, Aliya Rysbek, Andrea Huber, Brett Wiltshire, Brian Veprek, Daniel Gillick, Daniel Kasenberg, Derek Ahmed, Irina Jurenka, James Cohan, Jennifer She, Julia Wilkowski, Kaiz Alarakyia, Kevin R. McKee, Lisa Wang, Markus Kunesch, Mike Schaekermann, and 27 others. 2025. Learnlm: Improving gemini for learning. *Preprint*, arXiv:2412.16429.

Mark R. Lepper and Maria Woolverton. 2002. Chapter 7 - the wisdom of practice: Lessons learned from the study of highly effective tutors. In Joshua Aronson, editor, *Improving Academic Achievement*, Educational Psychology, pages 135–158. Academic Press, San Diego.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Jakub Macina, Nico Daheim, Sankalan Chowdhury, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023a. MathDial: A dialogue tutoring dataset with rich pedagogical properties grounded in math reasoning problems. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5602–5621, Singapore. Association for Computational Linguistics.

Jakub Macina, Nico Daheim, Ido Hakimi, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2025. MathTutorBench: A benchmark for measuring open-ended pedagogical capabilities of LLM tutors. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 204–221, Suzhou, China. Association for Computational Linguistics.

Jakub Macina, Nico Daheim, Lingzhi Wang, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023b. Opportunities and challenges in neural dialog tutoring. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2357–2372, Dubrovnik, Croatia. Association for Computational Linguistics.

Mistral AI. 2025. Introducing mistral 3. `https://mistral.ai/news/mistral-3`. Accessed: May 26, 2026.

Benjamin D. Nye, Arthur C. Graesser, and Xiangen Hu. 2014. Autotutor and family: A review of 17 years of natural language tutoring. *Int. J. Artif. Intell. Educ.*, 24(4):427–469.

Vardhan Palod, Upasana Biswas, and Subbarao Kambhampati. 2026. Evaluating the false trust engendered by llm explanations. *Preprint*, arXiv:2605.10930.

Jieun Park, KyungTae Lim, and Joon-ho Lim. 2026. Beyond accuracy: Alignment and error detection across languages in the bi-GSM8K math-teaching benchmark. In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 1678–1704, Rabat, Morocco. Association for Computational Linguistics.

Barbara Plank. 2022. The "problem" of human label variation: On ground truth in data, modeling and evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10671–10682, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Romain Puech, Jakub Macina, Julia Chatain, Mrinmaya Sachan, and Manu Kapur. 2025. Towards the pedagogical steering of large language models for tutoring: A case study with modeling productive failure. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 26291–26311,

Vienna, Austria. Association for Computational Linguistics.

Sukannya Purkayastha, Nils Dycke, Anne Lauscher, and Iryna Gurevych. 2026. Decision-making with deliberation: Meta-reviewing as a document-grounded dialogue. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3424–3465, Rabat, Morocco. Association for Computational Linguistics.

Christian P Robert. 2007. *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer.

Abdelrahman Sadallah, Tim Baumgärtner, Iryna Gurevych, and Ted Briscoe. 2025. The good, the bad and the constructive: Automatically measuring peer review's utility for authors. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 28991–29021, Suzhou, China. Association for Computational Linguistics.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. 33:3008–3021.

David Stutz, Abhijit Guha Roy, Tatiana Matejovicova, Patricia Strachan, Ali Taylan Cemgil, and Arnaud Doucet. 2023. Conformal prediction under ambiguous ground truth. *Transactions on Machine Learning Research*.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. 2005. *Algorithmic learning in a random world*. Springer.

Rose Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2024. Bridging the novice-expert gap via models of decision-making: A case study on remediating math mistakes. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2174–2199, Mexico City, Mexico. Association for Computational Linguistics.

Yixuan Weng, Minjun Zhu, Guangsheng Bao, Hongbo Zhang, Jindong Wang, Yue Zhang, and Linyi Yang. 2025. Cycleresearcher: Improving automated research via automated review. In *The Thirteenth International Conference on Learning Representations*.

Yixuan Weng, Minjun Zhu, Qiujie Xie, Zhiyuan Ning, Shichen Li, Panzhong Lu, Zhen Lin, Enhao Gu, Qiyao Sun, and Yue Zhang. 2026. Deepreviewer 2.0: A traceable agentic system for auditable scientific peer review. *Preprint*, arXiv:2604.09590.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

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.

Furkan Şahinuç, Subhabrata Dutta, and Iryna Gurevych. 2026. Reward modeling for scientific writing evaluation. *Preprint*, arXiv:2601.11374.

A Experimental Details

A.1 Reviewing Experiments

We first detail the exact prompts that we use for our work for reproducibility. The prompts for generating reviews on ICLR and ACL are found in Fig. 3 and Fig. 3, respectively. These prompts directly follow the guidelines that are posted on OpenReview for human reviewers in order to stay objective. The models used for generation are exactly detailed in the main body of the text with their corresponding identifier on the huggingface hub (`https://huggingface.co/models`). We use vllm (Kwon et al., 2023) with ancestral sampling and, in particular, do not use any kind of truncation sampling but rather use the full token-level distributions.

The prompts for scoring and utility estimation are given in Fig. 4, Fig. 5, Fig. 6, Fig. 7. We did not extensively tune the prompts for utility estimation and, in particular, did not optimize them for validation results using the reward model that we use for evaluation. For both cases, we transform the used models to a classifier in vllm that outputs a distribution over tokens, for example $0-4$ for 5-point-Likert utility estimation by taking the output embeddings of the corresponding tokens and removing all other output embeddings. We do this by throwing away all output embeddings but the ones for relevant tokens and then calculating a softmax only over these. We calculate the final score via an expectation over this classifier. Denoting it with $p(\cdot \mid \mathbf{x})$ and possible scores with $\mathcal{S} \in \mathbb{N}^{|\mathcal{S}|}$ we therefore use

$$s(\mathbf{y}, \omega) = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} p(s \mid \mathbf{x}) \cdot s. \tag{8}$$

This is consistent with a Bayesian approach wher unknowns are marginalized. The approach has concurrently been dubbed LLM-as-a-verifier (Kwok et al., 2026) and shown empirical success which we attribute to its inherent uncertainty reduction.

For scoring, we found that zero-shot LLMs without training did not perform well and almost always preferred scores that are between reject and accept or borderline. We therefore trained a smaller Llama3.2-3B model to predict scores. On ICLR data, we used the training set that is also provided by Weng et al. (2025). On NLPEER, we use the set of papers with one review in v1 from Dycke et al. (2023). We split off the first 90% of data for training and the remaining 10% for validation.

On ICLR, this amounts to 3,770 training and 419 calibration papers, on NLPEER to 770 training and 86 calibration papers. The test set on ICLR contains 781 papers and on NLPEER it contains 673 papers. We train the model for 4 epochs on ICLR and 6 epochs on NLPEER with a batch size of 32 achieved via gradient accumulation. For both we use AdamW (Loshchilov and Hutter, 2019), a learning rate of $1e-5$, a small weight decay of 0.000001, $(\beta_1, \beta_2) = (0.9, 0.99)$, 100 linear warmup steps for the learning rate which then is annealed to 0 using cosine decay. In both training and prediction we use a maximum input length of $32,768$ tokens for all models.

The evaluation is done by directly using the reward model and scores as defined in Şahinuç et al. (2026). We use the UKPLab/SciRM-Ref-7B model after splitting the weaknesses section of the paper into individual points using simple rules and apply the model to each weakness and average over them. A limitation of the approach is that strengths are not included in the rating due to the focus of Şahinuç et al. (2026). The scaling experiment in Sec. 6.4 follows the exact same set-up but varies the size of $\mathcal{H}$ as described.

All experiments are conducted on NVIDIA A100, RTX Pro 6000 and H200 GPUs with 80GB, 96GB, and 141GB memory. The experiments take around one day to run for the full pipeline of generation, utility estimation, decision, and evaluation on one of the datasets. The Llama3.2-3B-Instruct model was trained in ca. 2 hours on one A100 GPU with 80GB memory.

None of the data used contains personally-identifiable information. We also note that our usage of the data is in accordance with the licenses from Weng et al. (2025) and Dycke et al. (2023). We only use the data for research purposes and only advocate for usage of the methods within this scope. We do not promote any use of automated peer review methods in a malicious way or to automate the peer review process, or replace human decision-making. Rather, we clearly emphasize our goal as helping human reviewers and authors with automated methods to foster a responsible research environment and help all who are included in the peer review process.

A.2 Tutoring Experiments

For the tutoring experiments we use TutorRL-7B which is available on the huggingface hub under the identifier eth-nlped/TutorRL-7B. We do not alter

the system prompt but directly use the one defined in the chat template, as this was also used during model training. We again use ancestral sampling, no truncations of any sort, and use vllm.

We classify tutoring strategies into focus, probing, generic, and telling using a finetuned Llama3.2-3B model on MathDial using the prompt in Fig. 8. We again use AdamW, a learning rate of $1e-5$, a small weight decay of 0.000001, $(\beta_1, \beta_2) = (0.9, 0.99)$, 100 linear warmup steps for the learning rate which then is annealed to 0 using cosine decay and train for 5 epochs.

For finetuning, we use 90% of the tutor responses from the training set, where a conversation has at least had one utterance, yielding 9,162 utterances. We calibrate the conformal threshold on the remaining 10% which are 1,018 utterances. The test set contains 3100 examples.

Utility is calculated with a Qwen3-14B model that again is transformed to a classifier. We use the prompt in Fig. 6 and again use the LLM-as-a-verifier approach.

For evaluation, we directly use the approach from Macina et al. (2025) using the finetuned model found under eth-nlped/Qwen2.5-1.5B-pedagogical-rewardmodel on the huggingface hub. We use their input schema and exactly follow the scoring code.

All experiments are conducted on NVIDIA A100, RTX Pro 6000 and H200 GPUs with 80GB, 96GB, and 141GB memory. The experiments take around two to three hours to run for the full pipeline of generation, utility estimation, decision, and evaluation on one of the datasets. The Llama3.2-3B-Instruct model was trained in ca. 2 hours on one A100 GPU with 80GB memory. We average the results over three random seeds.

As Mathdial was collected with anonymous crowdworkers and LLM-simulated students, no personally-identifying data is contained or was used by us.

> Given a research paper and the review guidelines below, write a summary of its strengths and weaknesses. Be objective, thoughtful, critical and not too positive. Your points should be grounded in the paper. It is not necessary to balance out the number of strengths and weaknesses. Output a json dictionary.
> ## Review guidelines
> **Summary of Strengths** What are the major reasons to publish this paper at ICLR? These could include novel and useful methodology, insightful empirical results or theoretical analysis, clear organization of related literature, or any other reason why interested readers of ICLR papers may find the paper useful.
> **Summary of Weaknesses** What are the concerns that you have about the paper that would cause you to favor prioritizing other high-quality papers that are also under consideration for publication? These could include concerns about correctness of the results or argumentation, limited perceived impact of the methods or findings (note that impact can be significant both in broad or in narrow subfields), lack of clarity in exposition, or any other reason why interested readers of ICLR papers may gain less from this paper than they would from other papers under consideration. Where possible, please number your concerns as 1., 2., etc. so authors may respond to them individually.
> ## Output format Output only the json dictionary and follow the json schema exactly, with no extra keys, notes, comments, or explanations: "strengths": "...", "weaknesses": "..."

Figure 3: Prompt used to generate reviews for ICLR papers.

> Given a research paper and the review guidelines below, write a summary of its strengths and weaknesses. Be objective, thoughtful, critical and not too positive. Your points should be grounded in the paper. It is not necessary to balance out the number of strengths and weaknesses. Output a json dictionary.
> ## Review guidelines
> **Summary of Strengths** What are the major reasons to publish this paper at a selective *ACL venue? These could include novel and useful methodology, insightful empirical results or theoretical analysis, clear organization of related literature, or any other reason why interested readers of *ACL papers may find the paper useful.
> **Summary of Weaknesses** What are the concerns that you have about the paper that would cause you to favor prioritizing other high-quality papers that are also under con-

sideration for publication? These could include concerns about correctness of the results or argumentation, limited perceived impact of the methods or findings (note that impact can be significant both in broad or in narrow subfields), lack of clarity in exposition, or any other reason why interested readers of *ACL papers may gain less from this paper than they would from other papers under consideration. Where possible, please number your concerns as 1., 2., etc. so authors may respond to them individually.
Output format Output only the json dictionary and follow the json schema exactly, with no extra keys, notes, comments, or explanations: "strengths": "...", "weaknesses": "..."

Figure 4: Prompt used to generate reviews for ACL papers.

Given a review and scoring guidelines below, return a single number from the guidelines to indicate a score for a research paper that is consistent with the review. Be objective. A large number of strengths and few weaknesses indicate a good score. A large number of weaknesses and few strengths indicate a bad score. A similar number of both might be aborderline paper. ## Possible Scores
9: Top-quality paper: Top 17-8: Top-quality paper: Top 55-6: Accept: Strong paper with good contribution. 4: Weak Accept: Borderline paper, likely to be accepted. 3: Marginally below the acceptance threshold: Would not mind if the paper is accepted. 2: Weak Reject: Borderline paper, likely to be rejected. 0-1: Reject: Poor or deeply flawed paper
Output format Output only one number

Figure 5: Prompt used to score ICLR papers.

Given a review and scoring guidelines below, return a single number from the guidelines to indicate a score for a research paper that is consistent with the review. Be objective. A large number of strengths and few weaknesses indicate a good score. A large number of weaknesses and few strengths indicate a bad score. A similar number of both might be aborderline paper. ## Possible Scores
9 = Top-Notch: This is one of the best papers I read recently, of great interest for the (broad or narrow) sub-communities that might build on it 8 7 = This paper represents solid work, and is of significant interest for the (broad or narrow) sub-communities that might build on it 6 5 = Good: This paper makes a reasonable contribution, and might be of interest for some (broad or narrow) sub-communities, possibly with minor revisions 4 3 = Revisions Needed: This paper has some merit, but also significant flaws, and needs work before it would be of interest to the community 2 1 = Major Revisions Needed: This paper has significant flaws, and needs substantial work before it would be of interest to the community 0 = This paper is not relevant to the *ACL community (for example, is in no way related to natural language processing)
Output format Output only one number

Figure 6: Prompt used to score ACL papers. Note that we have transformed the scores to map to single number tokens for simplicity.

Given are a review and two guidelines. The first guideline explains what score should be given to a paper. The second guideline explains how to score the review for the paper based on review and score.
Your task is to score the review.
Paper Score Guideline
9: Top-quality paper: Top 17-8: Top-quality paper: Top 55-6: Accept: Strong paper with good contribution. Typically many strengths and some weaknesses. 4: Weak Accept: Borderline paper, likely to be accepted. Typically similar amount of strengths and weaknesses but strengths outweigh weaknesses. 3: Marginally below the acceptance threshold: Would not mind if the paper is accepted. Typically similar amount of strengths and weaknesses but strengths outweigh weaknesses slightly. 2: Weak Reject: Borderline paper, likely to be rejected. Typically more weaknesses than strengths. 0-1: Reject: Poor or deeply flawed paper. Typically a paper with many weaknesses that can not easily be resolved.
Review Score Guideline

0: The review is not consistent with the paper score or of low quality. For example, the review could be very positive and the score low, or the review could be negative but the score high. A low quality review is indicated by unsubstantiated claims, feedback that is not actionable, helpful, or verifiable. 1: The review is only somewhat consistent with the review score or of rather low quality. For example, the review contains many unsubstantiated claims and is unfair, contains only little actionable feedback, is not very helpful, and many points are hard to verify. 2: The review fits the score to some extent and is of average quality. For example, it is not in-depth or contains some unsubstantiated claims, some claims that are not actionable, is somewhat helpful, and has a few points that are hard to verify. Some points are grounded in the paper. 3: The review and score fit. The review has at most very few unsubstantiated claims, fairly addresses the papers weaknesses, and provides actionable guidelines to the authors. The review is helpful and most points are grounded in the paper. 4: The review and score fit. The review has no unsubstantiated claims, fairly addresses the papers weaknesses, and provides actionable guidelines to the authors. The review is of exceptional quality. All points are grounded well in the paper.
Output Format Return only one score in the range 0-4 from the review score guidelines and nothing else.

Figure 7: Prompt for estimating the utility of ICLR reviews used in $u(\omega, \mathbf{y})$.

Given are a review and two guidelines. The first guideline explains what score should be given to a paper. The second guideline explains how to score the review based on review and score.
Your task is to score the review objectively.
Paper Score Guideline
9 = Top-Notch: This is one of the best papers I read recently, of great interest for the (broad or narrow) sub-communities that might build on it 8 7 = This paper represents solid work, and is of significant interest for the (broad or narrow) sub-communities that might build on it 6 5 = Good: This paper makes a reasonable contribution, and might be of interest for some (broad or narrow) sub-communities, possibly with minor revisions 4 3 = Revisions Needed: This paper has some merit, but also significant flaws, and needs work before it would be of interest to the community 2 1 = Major Revisions Needed: This paper has significant flaws, and needs substantial work before it would be of interest to the community 0 = This paper is not relevant to the *ACL community (for example, is in no way related to natural language processing)
Review Score Guideline
0: The review is not consistent with the paper score or of low quality. For example, the review could be very positive and the score low, or the review could be negative but the score high. A low quality review is indicated by unsubstantiated claims, feedback that is not actionable, helpful, or verifiable. 1: The review is only somewhat consistent with the review score or of rather low quality. For example, the review contains many unsubstantiated claims and is unfair, contains only little actionable feedback, is not very helpful, and many points are hard to verify. 2: The review fits the score to some extent and is of average quality. For example, it is not in-depth or contains some unsubstantiated claims, some claims that are not actionable, is somewhat helpful, and has a few points that are hard to verify. Some points are grounded in the paper. 3: The review and score fit. The review has at most very few unsubstantiated claims, fairly addresses the papers weaknesses, and provides actionable guidelines to the authors. The review is helpful and most points are grounded in the paper. 4: The review and score fit. The review has no unsubstantiated claims, fairly addresses the papers weaknesses, and provides actionable guidelines to the authors. The review is of exceptional quality. All points are grounded well in the paper.
Output Format Return only one score in the range 0-4 from the review score guidelines and nothing else.

Figure 8: Prompt for estimating the utility of ACL reviews used in $u(\omega, \mathbf{y})$.

You are given a conversation between a student and a teacher, where the teacher attempts to guide the student towards correcting their incorrect solution attempt. Your task is to indicate the tutoring strategy that the teacher should use in their next utterance from the following choices. DO NOT CLASSIFY THE CURRENT TEACHER UTTERANCE. Follow the guidelines below, where there is an explanation of intents behind each strategy with examples.
0. Focus: - Seek Strategy: So what should you do next? - Guiding Student: Can you calculate . . . ? - Recall Relevant Information: Can you reread the question and tell me what is . . . ?
1. Probing: - Asking for Explanation: Why do you think you need to add these numbers? - Seeking Self Correction: Are you sure you need to add here? - Perturbing the Question: How would things change if they had . . . items instead? - Seeking World Knowledge: How do you calculate the perimeter of a square?
2. Telling: - Revealing Strategy: You need to add . . . to . . . to get your answer. - Revealing Answer No, he had . . . items.
3. Generic: - Greeting/Farewell: Hi . . . , how are you doing with the word problem? - General inquiry: Can you go walk me through your solution?
Telling should only be used rarely, for example, if the student is stuck.
Output Format Return only one numer in the range 0-3 to indicate the tutoring strategy from the list and nothing else.

Figure 9: Prompt for classifying tutor strategies.

You are given a conversation between a student and a teacher, where the teacher attempts to guide the student towards correcting their incorrect solution attempt, a guideline for tutoring strategies and a guideline for scoring. Your task is to predict the quality of the last reponse generated by the teacher model.
Tutoring Strategies
0. focus: - Seek Strategy: So what should you do next? - Guiding Student: Can you calculate . . . ? - Recall Relevant Information: Can you reread the question and tell me what is . . . ?
1. probing: - Asking for Explanation: Why do you think you need to add these numbers? - Seeking Self Correction: Are you sure you need to add here? - Perturbing the Question: How would things change if they had . . . items instead? - Seeking World Knowledge: How do you calculate the perimeter of a square?
2. telling: - Revealing Strategy: You need to add . . . to . . . to get your answer. - Revealing Answer No, he had . . . items.
3. generic: - Greeting/Farewell: Hi . . . , how are you doing with the word problem? - General inquiry: Can you go walk me through your solution?
Scoring Guidelines
Use these criteria to score a response according to the following rubrics: (a) correctness: the teacher should guide the student towards the correct answer and not state incorrect facts (b) scaffolding instead of giving away the answer: the teacher should help the student to cognitively engage with the problem and discover the answer on their own (c) encourage self-correction: by correctly identifying the student mistake and first giving the student the opportunity to self-correct and learn from a mistake (d) not overloading the student: manage cognitive load by not giving too much information at once.
0: The response is either inappropriate, does not fit to the given tutoring strategy or does not satisfy any of the above criteria.
1: The response tells the solution too early, or does not provide the student with any actionable feedback or room to improve their understanding. Overall, it only satisfies few of the criteria.
2: The response has good pedagogical quality and provides the student with some room to improve their understanding. It satisfies most of the criteria but not all of them. It fits the tutoring strategy but not perfectly.
3: The response is of high pedagogical quality. It satisfies almost all criteria above and fits the tutoring strategy but is not completely tailored to it. It does not tell the solution directly, unless at the end of the conversation.
4: The response is of exceptional pedagogical quality. It does not tell the solution directly unless absolutely necessary. It provides the student with explicit ways to improve their un-

derstanding. It satisfies all of the above criteria. The response is tailored to the tutoring strategy. ## Output Format Return only one numer in the range 0-4 to indicate the quality of the teacher response according to the Scoring Guidelines and nothing else.

Figure 10: Prompt for estimating the utility of tutor responses used in $u(\omega, \mathbf{y})$. The examples are taken verbatim from Macina et al. (2023a).