Can LLMs Recognize Their Own Analogical Hallucinations? Evaluating Uncertainty Estimation for Analogical Reasoning

Zheng Chen[1*] , **Zhaoxin Feng**[2] , **Jianfei Ma**[2] , **Jiexi Xu**[3] , **Bo Li**[1]
[1]Computer Science and Engineering, Hong Kong University of Science and Technology
[2]Chinese and Bilingual Studies, The Hong Kong Polytechnic University
[3]Faculty of Business and Economy, The University of Hong Kong
zchenin@connect.ust.hk, {zhaoxinbetty.feng, jian-fei.ma}@connect.polyu.edu.hk,
tomxuhi@connect.hku.hk, bli@cse.ust.hk

Abstract

Large language models (LLMs) often demonstrate strong performance by leveraging implicit knowledge acquired during pretraining. Analogical reasoning, which solves new problems by referencing similar known examples, offers a structured way to utilize this knowledge, but can also lead to subtle factual errors and hallucinations. In this work, we investigate whether LLMs can recognize the reliability of their own analogical outputs using *black-box uncertainty estimation (UE)*. We evaluate six UE metrics across two reasoning-intensive tasks: mathematical problem solving and code generation. Our results show that *Kernel Language Entropy (KLE)* and *Lexical Similarity (LexSim)* are the most robust indicators of correctness. Moreover, while analogical prompting lowers model uncertainty over direct prompting, most uncertainty arises during the analogy transfer step. These findings highlight the limitations of analogical knowledge transfer in LLMs and demonstrate the potential of UE methods for detecting hallucinated reasoning in black-box settings.

1 Introduction

Recent advances in large language models (LLMs) have highlighted their surprising ability to utilize internalized knowledge for solving complex tasks. This ability, often acquired through large-scale pretraining, enables models to answer factual questions, reason about concepts, and even perform domain-specific tasks without explicit retrieval (Yang et al., 2024; Zhang et al., 2025). However, such knowledge utilization remains opaque and error-prone. In particular, LLMs frequently produce responses that are fluent and confident but factually incorrect, which is a phenomenon known as *hallucination* (Qin et al., 2025).

To better understand how knowledge is used, represented, and sometimes misapplied by LLMs, we focus on a specific form of structured reasoning: *analogical reasoning*. This strategy encourages the model to solve a target problem by referencing a related, known problem. Analogical reasoning has roots in human cognition (Vosniadou and Ortony, 1989) and has been shown to enhance LLM performance across domains (Yasunaga et al., 2024; Yang et al., 2024; Zhang et al., 2025). Conceptually, it involves two stages: retrieving or constructing an analogy, and transferring it to the new context (Ramachandran, 2012).

Despite its potential, analogical reasoning is also prone to hallucination-like failure. Models may select an irrelevant analogy, or fail to adapt it correctly, leading to incorrect answers that nonetheless appear coherent and justified. These subtle errors are particularly dangerous in deployment settings, as they can undermine user trust in the model's reasoning ability. This raises a key research question: *can LLMs recognize when their analogical reasoning is unreliable?*

We address this question by investigating the utility of *black-box uncertainty estimation (UE)* metrics. These methods aim to quantify model uncertainty based solely on output patterns, without requiring access to internal activations or probabilities (Fadeeva et al., 2023). Prior work has applied UE to tasks such as translation and summarization (Fomicheva et al., 2020), but its effectiveness in analogical reasoning, where hallucinations arise from multi-step failures, remains underexplored.

In this paper, we evaluate six representative UE metrics in the context of analogical prompting. Our experiments span two reasoning-intensive benchmarks: GSM8K for mathematical problem solving, and Codeforces for code generation. We further dissect analogical responses into their subcomponents to understand where uncertainty arises: in the analogy itself or in its transfer. This work makes three main contributions:

[*] Corresponding author

Proceedings of the 3rd Workshop on Towards Knowledgeable Foundation Models (KnowFM), pages 84–93
August 1, 2025 ©2025 Association for Computational Linguistics

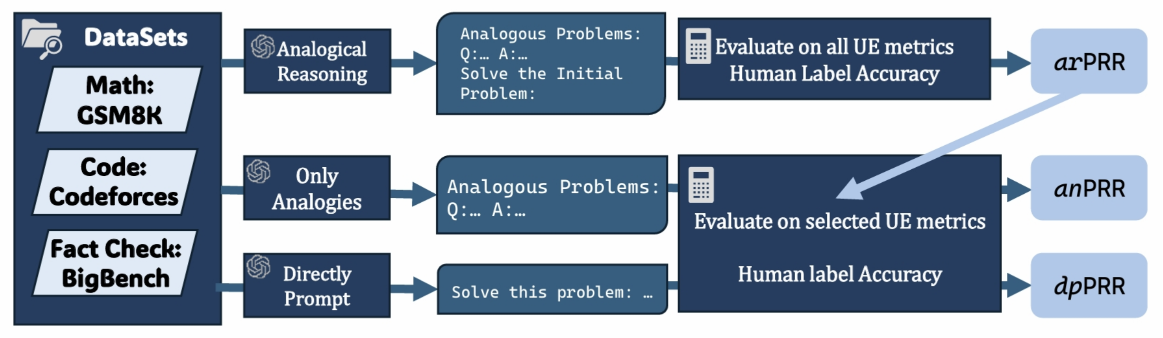


Figure 1: Overall procedure of our method. The first row illustrates the process for identifying robust uncertainty estimation metrics. The latter two rows demonstrate the steps of calculating the uncertainty in analogical reasoning.

- We present the first systematic evaluation of uncertainty estimation metrics for analogical reasoning in black-box LLMs.
- We identify two metrics, Kernel Language Entropy (KLE) and Lexical Similarity (LexSim), that best predict factual correctness.
- We show that analogical reasoning decreases model uncertainty, but most uncertainty arises from the transfer step.

Our findings provide insights into the mechanisms and limits of knowledge utilization in LLMs, and offer a practical pathway toward detecting hallucinated reasoning in analogical contexts. Our code can be found in `https://github.com/Bellafc/analogyUE/`.

2 Related Work

2.1 Analogical Reasoning

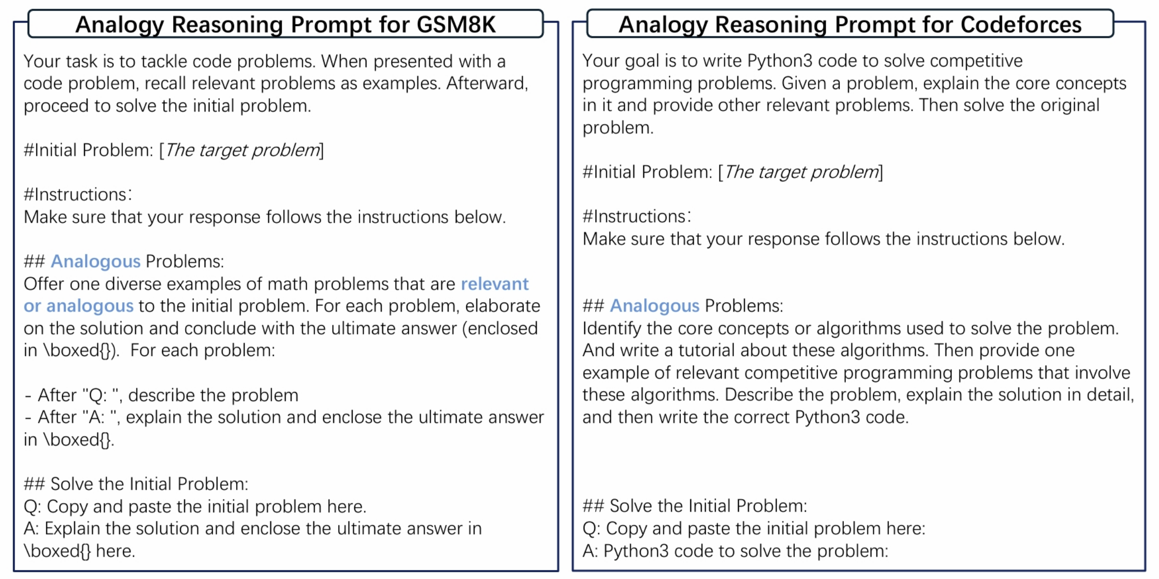
Analogical reasoning is a procedure of: 1) retrieving knowledge for obtaining similarities among questions, and 2) transferring the knowledge from the known source to the unknown target (Ramachandran, 2012). Analogical reasoning first identifies deep relational similarities (e.g., batteries and reservoirs both store and release energy, beyond surface differences). It then transfers these higher-order structures to the unknown problems (e.g., the "central force-orbital motion" in solar system-atom analogies) while ignoring superficial features (Gentner, 1983).

Recent studies have applied analogical reasoning to mathematical problem-solving and code generation by prompting LLMs to generate relevant exemplars or knowledge, thereby enhancing reasoning performance (Yasunaga et al., 2024). However, while analogical reasoning effectively leverages implicit pretrained knowledge, it may introduce factual errors or hallucinations (Qin et al., 2025). This paper aims to investigate the reliability of LLMs in analogical reasoning, uncovering the sources of uncertainty.

2.2 Uncertainty Estimation

With the widespread adoption of LLMs, their generated outputs are prone to hallucination (Xiao and Wang, 2021; Dziri et al., 2022). Uncertainty estimation methods address this issue by quantifying the confidence of model predictions, enabling users to identify unreliable outputs and thereby enhancing the safety and reliability of LLM deployments (Fadeeva et al., 2023).

Uncertainty estimation mainly includes two methods: white-box and black-box. White-box methods, requiring access to internal model states, include information-theoretic approaches like maximum sequence probability and semantic entropy (Kuhn et al., 2023), ensemble-based techniques (Malinin and Gales, 2021) using cross-model prediction variances, and density estimation methods such as Mahalanobis distance (Lee et al., 2018) for out-of-distribution detection. Black-box methods, which operate solely on generated text outputs, encompassing semantic diversity analysis (Lin et al., 2024) that evaluates uncertainty by computing similarity matrices across multiple responses, as well as graph-theoretic approaches (Fadeeva et al., 2023). In contrast to white-box approaches, this paper focuses specifically on black-box uncertainty estimation for analogical reasoning, enabling reliable hallucination detection without access to internal model states.



Analogy Reasoning Prompt for GSM8K

Your task is to tackle code problems. When presented with a code problem, recall relevant problems as examples. Afterward, proceed to solve the initial problem.

#Initial Problem: [*The target problem*]

#Instructions:
Make sure that your response follows the instructions below.

Analogous Problems:
Offer one diverse examples of math problems that are relevant or analogous to the initial problem. For each problem, elaborate on the solution and conclude with the ultimate answer (enclosed in \boxed{}). For each problem:

- After "Q: ", describe the problem
- After "A: ", explain the solution and enclose the ultimate answer in \boxed{}.

Solve the Initial Problem:
Q: Copy and paste the initial problem here.
A: Explain the solution and enclose the ultimate answer in \boxed{} here.

Analogy Reasoning Prompt for Codeforces

Your goal is to write Python3 code to solve competitive programming problems. Given a problem, explain the core concepts in it and provide other relevant problems. Then solve the original problem.

#Initial Problem: [*The target problem*]

#Instructions:
Make sure that your response follows the instructions below.

Analogous Problems:
Identify the core concepts or algorithms used to solve the problem. And write a tutorial about these algorithms. Then provide one example of relevant competitive programming problems that involve these algorithms. Describe the problem, explain the solution in detail, and then write the correct Python3 code.

Solve the Initial Problem:
Q: Copy and paste the initial problem here:
A: Python3 code to solve the problem:

Figure 2: Analogical Reasoning Prompts for GSM8K and Codeforces.

3 Method

The overall experimental pipeline is illustrated in Figure 1. As part of this procedure, we apply analogical prompting to two reasoning-intensive datasets: *GSM8K* for mathematical problem solving and *Codeforces* for code generation. The specific analogical prompts used for these two datasets are provided in Figure 2.

3.1 Common UE metrics for black-box LLMs

To evaluate the uncertainty of model-generated responses in a black-box setting, we adopt six representative UE metrics, implemented via the library introduced in (Fadeeva et al., 2023). These metrics capture diverse statistical and structural properties of language model outputs. A brief overview is provided below.

- **Sum of Eigenvalues of the Graph Laplacian (EigV)** (Lin et al., 2024): This metric computes the sum of eigenvalues of the Laplacian matrix L constructed from a token-level similarity graph of the generated text. Intuitively, higher spectral mass reflects lower uncertainty.

$$\text{EigV}(x) = \sum_{i=1}^{n} \lambda_i \quad \text{where } Lx = \lambda x \tag{1}$$

- **Degree Matrix (Deg)** (Lin et al., 2024): Defined as the sum of degrees in the token similarity graph, this metric serves as a proxy for local cohesion in the response.

$$\text{Deg}(x) = \sum_{i} \deg(v_i) \tag{2}$$

- **Eccentricity (Ecc)** (Lin et al., 2024): This metric computes the maximum shortest-path distance from any node to all other nodes in the graph. Lower eccentricity indicates more centralized (and potentially more confident) responses.

$$\text{Ecc}(x) = \max_{v \in V} \min_{u \in V} d(v, u) \tag{3}$$

- **Lexical Similarity (LexSim)** (Fomicheva et al., 2020): Based on pairwise cosine similarity among token embeddings, this metric reflects lexical cohesion in the output.

$$\text{LexSim}(x) = \frac{2}{n(n-1)} \sum_{i<j} \cos(\vec{e}_i, \vec{e}_j) \tag{4}$$

- **Kernel Language Entropy (KLE)** (Nikitin et al., 2024): This metric estimates the entropy of the response using a kernel-based density estimation over token embeddings. Lower entropy typically indicates lower uncertainty.

$$\text{KLE}(x) = -\sum_{i} \log \left(\sum_{j} K(x_i, x_j) \right) \tag{5}$$

- **LUQ (Local Uncertainty Quantification)** (Zhang et al., 2024): A recent metric that quantifies uncertainty by measuring the variance in local regions of the output embedding space.

$$\mathrm{LUQ}(x) = \frac{1}{n} \sum_i \mathrm{Var}(N_k(x_i)) \tag{6}$$

  where $N_k(x_i)$ denotes the k-nearest neighbors of token x_i.

3.2 Identify Robust UE Metrics

We begin by evaluating the reliability of six UE metrics in assessing the correctness of analogical reasoning outputs. Our study is conducted on two reasoning-intensive benchmarks introduced in (Yasunaga et al., 2024): *GSM8K* for mathematical problem solving (Cobbe et al., 2021), and *Codeforces* for code generation (Majd et al., 2019). From each dataset, we randomly sample 200 examples and apply the analogical reasoning prompting strategy proposed in prior work.

For each generated response, we compute six UE scores using the following black-box estimators: sum of graph Laplacian eigenvalues (EigV), degree matrix (Deg), eccentricity (Ecc), lexical similarity (LexSim), kernel language entropy (KLE), and LUQ (Lin et al., 2024; Fomicheva et al., 2020; Nikitin et al., 2024; Zhang et al., 2024). In parallel, we conduct human evaluation on all 400 analogical reasoning responses, where each response is assigned a score from 0 to 100 based on its factual correctness and reasoning quality. One of the author and a student research assistant jointly annotated the responses. These human scores serve as the ground-truth accuracy proxy.

To assess how well each UE metric correlates with human judgment, we compute the Predictive Rate Ratio (PRR) for each metric:

$$\mathrm{PRR} = \frac{\mathrm{AUCPRunc}}{\mathrm{AUCPRoracle}} \tag{7}$$

This ratio measures the area under the precision-recall curve (AUCPR) when ranking predictions by their uncertainty values, normalized by the oracle AUCPR (i.e., ideal ranking using ground-truth labels). A higher PRR indicates a stronger ability to distinguish between correct and incorrect responses based on uncertainty alone. We select the top-2 metrics with the highest PRR scores for use in subsequent stages.

3.3 Uncertainty Loss in Analogies

Building on the previous step, we further examine the interaction between analogical prompting and uncertainty estimation. Specifically, we aim to evaluate whether analogical reasoning lowers the uncertainty in LLM outputs and to what extent uncertainty varies across prompting strategies.

For each of the same 200 samples per dataset, we perform three types of evaluation:

Analogical Prompting (ar): Full analogical reasoning prompt used to generate response r_{ar}.

Direct Prompting (dp): A baseline prompt without analogical structure, producing r_{dp}.

Analogy-Only (an): The analogy section (e.g., retrieved or constructed examples) extracted from r_{ar}, yielding r_{an}.

For each of these three prompting modes, we compute the UE scores using only the top-2 metrics identified in the previous step. Human evaluators also score r_{dp} and r_{an} to provide corresponding correctness labels (a_{dp} and a_{an}).

This setup allows us to compute three sets of PRR scores:

***ar*PRR**: PRR from analogical reasoning outputs.

***dp*PRR**: PRR from direct prompting outputs.

***an*PRR**: PRR from analogy-only segments.

By comparing these three PRR scores, we can isolate the contribution of analogical structure to model uncertainty and quantify its influence on UE metric behavior.

3.4 Overall Procedure

Algorithm 1 outlines the complete evaluation pipeline. For each sample, we first generate a response using analogical prompting. We then evaluate this response using all six UE metrics, resulting in six corresponding uncertainty scores u_{ar}^m. Human annotators assess the correctness of each analogical response to yield the score a_{ar}. Using these uncertainty-accuracy pairs, we compute the analogical reasoning PRR scores arPRR$_m$ for all metrics and identify the top two performing metrics. Subsequently, we evaluate the same sample with direct prompting and analogy-section-only extraction. For each of the direct prompting results and the analogy-section only extraction, we apply only the top-2 UE metrics selected based on arPRR. The resulting responses are scored for correctness (a_{dp} and a_{an}), and corresponding uncertainty estimates (u_{dp}^m and u_{an}^m) are computed for each selected metric m. Finally, we compute the corresponding PRRs

Algorithm 1 Evaluation Pipeline for Analogical Reasoning uncertainty Analysis

```
for each sample in dataset do
    // Analogical Reasoning Prompt
    $r_{\text{ar}} \leftarrow$ LLM(AnalogicalPrompt(sample))
    for each UE metric $m$ in {EigV, Deg, Ecc, LexSim, KLE, LUQ} do
        $u^m_{\text{ar}} \leftarrow \text{UE}_m(r_{\text{ar}})$
        $a_{\text{ar}} \leftarrow \text{HumanScore}(r_{\text{ar}})$
    end for
end for
// Compute arPRR for all metrics
for each metric $m$ do
    $\text{arPRR}_m \leftarrow \text{ComputePRR}(u^m_{\text{ar}}, a_{\text{ar}})$
end for
// Select top-2 metrics based on arPRR
Top2Metrics $\leftarrow \text{SelectTopK}(\{\text{arPRR}_m\}, k = 2)$
for each sample in dataset do
    // Direct Prompting
    $r_{\text{dp}} \leftarrow$ LLM(DirectPrompt(sample))
    // Extracted Analogy Section
    $r_{\text{an}} \leftarrow \text{ExtractAnalogySection}(r_{\text{ar}})$
    for each metric $m$ in Top2Metrics do
        $u^m_{\text{dp}} \leftarrow \text{UE}_m(r_{\text{dp}})$
        $u^m_{\text{an}} \leftarrow \text{UE}_m(r_{\text{an}})$
        $a_{\text{dp}} \leftarrow \text{HumanScore}(r_{\text{dp}})$
        $a_{\text{an}} \leftarrow \text{HumanScore}(r_{\text{an}})$
    end for
end for
// Compute PRRs for top-2 metrics
for each metric $m$ in Top2Metrics do
    $\text{dpPRR}_m \leftarrow \text{ComputePRR}(u^m_{\text{dp}}, a_{\text{dp}})$
    $\text{anPRR}_m \leftarrow \text{ComputePRR}(u^m_{\text{an}}, a_{\text{an}})$
end for
```

for both direct prompting (*dp*PRR) and analogy-section-only (*an*PRR), allowing us to compare the predictive utility of uncertainty estimates across prompting strategies.

4 Results

4.1 KLE and LexSim are Robust UE metrics

Table 1 reveals that **KLE** and **LexSim** outperform other UE metrics across benchmarks. This divergence stems from the distinct demands of analogical reasoning:

1. KLE's Robustness to Semantic Diversity Analogical reasoning often involves *structurally valid but lexically diverse solutions* (e.g., different algorithmic implementations for the same programming problem). KLE's semantic kernel captures this structural coherence by encoding logical relationships beyond surface features. For instance, in Codeforces, valid code analogies may share no lexical overlap (e.g., recursive vs. iterative solutions) but exhibit high semantic similarity in control flow or data structures. KLE's entropy quantifies this implicit consistency, making it task-agnostic.

2. LexSim's Domain-Specific Utility LexSim excels in *mathematical reasoning (GSM8K)*, where answers often follow rigid templates (e.g., arithmetic expressions like $3x + 5 = 20$). Here, correct analogies inherently share high lexical overlap (e.g., repeated operators or variables), aligning LexSim with human judgment. However, its reliance on surface patterns fails in tasks requiring flexible logical expression, leading to poor performance (PRR=0.092).

3. Failure of Graph-Based and NLI Metrics

- **EigV/Deg/Ecc:** These graph-based metrics assume that semantic similarity correlates with logical validity. However, analogical reasoning allows structurally distinct but logically equivalent answers (e.g., different proof paths in math), violating this assumption.
- **LUQ:** NLI models struggle to assess bidirectional entailment in complex analogies (e.g., code logic), often misclassifying valid variations as contradictions.

4.2 Analogical Reasoning Lowers the Uncertainty, but Transfer Reduces It

The results presented in Table 2 show the relationship of *an*PRR, *ar*PRR, and *dp*PRR, with the measurement of the selected two UE metrics. As mentioned in Section 3, *an*PRR measures the uncertainty of the whole uncertainty reasoning process, while *an*PRR focuses on the uncertainty of the analogous questions and answers. *dp*PRR evaluates the uncertainty estimate for responses generated through direct prompting, without any analogical reasoning component.

The results show that the *an*PRR values are consistently higher than the *ar*PRR values across all datasets. This suggests that the LLM is more confident in the analogous questions and answers. The model is likely confident in identifying relevant analogies and applying them to the problem at hand.

	GPT-3.5-Turbo			GPT-4		
UE Method	**GSM8K**	**Codeforces**	**Avg**	**GSM8K**	**Codeforces**	**Avg**
KLE	0.187±0.013	0.200±0.015	0.194	0.201±0.008	0.215±0.019	0.208
LexSim	0.285±0.014	0.101±0.013	0.193	0.296±0.021	0.113±0.018	0.205
EigV	0.032±0.015	0.023±0.013	0.028	0.039±0.011	0.027±0.019	0.033
Ecc	-0.014±0.013	0.014±0.013	0.000	-0.005±0.012	0.021±0.012	0.008
Deg	-0.135±0.010	-0.018±0.012	-0.077	-0.127±0.008	-0.012±0.014	-0.070
LUQ	-0.106±0.012	-0.136±0.010	-0.121	-0.101±0.021	-0.130±0.019	-0.116

Table 1: Performance of UE methods on two datasets (***ar*PRR** and its variance), comparing `gpt-3.5-turbo` and `gpt-4`. Values are color-coded from light blue (lowest) to dark blue (highest) within each column group.

Model	**Dataset**	**Metric**	**KLE**	**LexSim**
GPT-3.5-Turbo	GSM8K	*ar*PRR	0.187	0.285
		*an*PRR	0.354	0.372
		*dp*PRR	0.103	-0.002
GPT-3.5-Turbo	Codeforces	*ar*PRR	0.200	0.028
		*an*PRR	0.289	0.163
		*dp*PRR	0.098	0.009
GPT-4	GSM8K	*ar*PRR	0.201	0.310
		*an*PRR	0.389	0.402
		*dp*PRR	0.115	0.011
GPT-4	Codeforces	*ar*PRR	0.215	0.075
		*an*PRR	0.317	0.190
		*dp*PRR	0.121	0.023

Table 2: UE metric values (*ar*PRR, *an*PRR, *dp*PRR) across datasets and models for KLE and LexSim.

However, the lower *ar*PRR values indicate that the model's uncertainty increases when it comes to transferring the solution from the analogy to the original problem. This could be because the process of adapting and applying the analogy to the new context introduces additional uncertainty. The higher *an*PRR values suggest that analogical reasoning is an effective strategy for lowering the uncertainty, whereas the low *dp*PRR values emphasize the limitations of direct prompting without such reasoning.

5 Discussion

Our findings highlight two key insights into UE in analogical reasoning tasks. First, KLE (Nikitin et al., 2024) and LexSim (Fomicheva et al., 2020) emerge as robust and complementary UE metrics, each excelling in different domains due to their underlying assumptions about semantic and lexical similarity. Second, analogical reasoning lowers model uncertainty, but this uncertainty increases during the transfer phase, underscoring a critical bottleneck in applying analogies to novel problems. Graph-based metrics (e.g., EigV, Deg, Ecc) and NLI-based LUQ underperform, suggesting a misalignment with the nature of analogical reasoning. These methods assume that surface-level similarity or binary entailment captures uncertainty effectively. However, analogical tasks often require recognizing logically valid yet structurally diverse answers. Their poor average scores and high variances confirm their inadequacy in capturing nuanced analogical consistency.

The second set of results reveals that analogical reasoning lowers the model's self-assessed uncertainty (as reflected by higher *an*PRR), yet this uncertainty loss does not fully translate into successful application (lower *ar*PRR). This divergence points to a key challenge: while models can identify useful analogies, the process of adapting them to new contexts introduces epistemic uncertainty. The lowest scores observed in the *dp*PRR condition further reinforce the value of analogy-based prompting over direct prompting. However, the drop from *an*PRR to *ar*PRR indicates that the analogy transfer step is a critical weakness in current LLM capabilities.

These findings suggest that future uncertainty metrics should better account for the two-step nature of analogical reasoning: analogy retrieval and transfer. While KLE and LexSim provide partial solutions, hybrid models or adaptive metrics that dynamically weigh lexical and semantic coherence may further improve reliability.

Limitations

While our study presents a systematic evaluation of black-box uncertainty estimation in analogical reasoning, several limitations remain.

First, our analysis is restricted to two datasets, which, although representative of mathematical and algorithmic reasoning, may not fully capture the diversity of analogical tasks across domains such as law, science, or creative writing. Extending our evaluation to other datasets like BigBench or domain-specific benchmarks would strengthen the generalizability of our findings.

Second, our evaluation focuses exclusively on black-box LLMs, namely GPT-3.5-Turbo and GPT-4, due to API access and usage constraints. While this reflects realistic deployment conditions, it excludes signals from white-box techniques such as self-consistency voting or intermediate activation inspection. Hybrid approaches that combine surface-level uncertainty metrics with internal model signals may further improve performance, especially during the analogy-transfer stage where uncertainty loss is limited.

Third, all human annotations were conducted by one author, supplemented by DeepSeek-V3-0324 model suggestions. To ensure label reliability, we verified a randomly sampled subset and observed high agreement ($\kappa > 0.8$). Nonetheless, future studies could benefit from a full multi-annotator protocol with inter-annotator agreement reporting.

Lastly, while we adopt the term "uncertainty loss" to describe reductions in estimated uncertainty, this does not directly equate to calibrated model confidence. Our measurements are inherently proxy-based and reflect surface-level output coherence rather than epistemic access to the model's belief state. Future work may explore adaptive uncertainty metrics or calibration techniques that better align with the two-stage nature of analogical reasoning.

Acknowledgments

We thank anonymous reviewers for their constructive feedback and suggestions, which have significantly helped us to refine this study.

References

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Nouha Dziri, Sivan Milton, Mo Yu, Osmar Zaiane, and Siva Reddy. 2022. On the origin of hallucinations in conversational models: Is it the datasets or the models? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5271–5285, Seattle, United States. Association for Computational Linguistics.

Ekaterina Fadeeva, Roman Vashurin, Akim Tsvigun, Artem Vazhentsev, Sergey Petrakov, Kirill Fedyanin, Daniil Vasilev, Elizaveta Goncharova, Alexander Panchenko, Maxim Panov, Timothy Baldwin, and Artem Shelmanov. 2023. LM-polygraph: Uncertainty estimation for language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 446–461, Singapore. Association for Computational Linguistics.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Dedre Gentner. 1983. Structure-mapping: A theoretical framework for analogy. *Cognitive science*, 7(2):155–170.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2024. Generating with confidence: Uncertainty quantification for black-box large language models. *Transactions on Machine Learning Research*.

Amirabbas Majd, Mojtaba Vahidi-Asl, Alireza Khalilian, Ahmad Baraani-Dastjerdi, and Bahman Zamani. 2019. Code4bench: A multidimensional benchmark of codeforces data for different program analysis techniques. *Journal of Computer Languages*, 53:38–52.

Andrey Malinin and Mark Gales. 2021. Uncertainty estimation in autoregressive structured prediction. In *International Conference on Learning Representations*.

Alexander Nikitin, Jannik Kossen, Yarin Gal, and Pekka Marttinen. 2024. Kernel language entropy: Fine-grained uncertainty quantification for llms from semantic similarities. *Advances in Neural Information Processing Systems*, 37:8901–8929.

Chengwei Qin, Wenhan Xia, Tan Wang, Fangkai Jiao, Yuchen Hu, Bosheng Ding, Ruirui Chen, and Shafiq Joty. 2025. Relevant or random: Can llms truly perform analogical reasoning? *Preprint*, arXiv:2404.12728.

Vilayanur S Ramachandran. 2012. *Encyclopedia of human behavior*. Academic Press.

Stella Vosniadou and Andrew Ortony. 1989. *Similarity and analogical reasoning*. Cambridge University Press.

Yijun Xiao and William Yang Wang. 2021. On hallucination and predictive uncertainty in conditional language generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2734–2744, Online. Association for Computational Linguistics.

Ling Yang, Zhaochen Yu, Tianjun Zhang, Shiyi Cao, Minkai Xu, Wentao Zhang, Joseph E Gonzalez, and Bin Cui. 2024. Buffer of thoughts: Thought-augmented reasoning with large language models. *Advances in Neural Information Processing Systems*, 37:113519–113544.

Michihiro Yasunaga, Xinyun Chen, Yujia Li, Panupong Pasupat, Jure Leskovec, Percy Liang, Ed H. Chi, and Denny Zhou. 2024. Large language models as analogical reasoners. In *The Twelfth International Conference on Learning Representations*.

Caiqi Zhang, Fangyu Liu, Marco Basaldella, and Nigel Collier. 2024. Luq: Long-text uncertainty quantification for llms. *arXiv preprint arXiv:2403.20279*.

Yufeng Zhang, Xuepeng Wang, Lingxiang Wu, and Jinqiao Wang. 2025. Enhancing chain of thought prompting in large language models via reasoning patterns. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25985–25993.

A Data Examples

Dataset	GSM8K
Initial Problem	Mary does her grocery shopping on Saturday. She does her shopping only at a specific store where she is allowed a credit of \$100, which must be paid in full before her next shopping trip. That week she spent the full credit limit and paid \$15 of it on Tuesday and \$23 of it on Thursday. How much credit will Mary need to pay before her next shopping trip?
Analogical Reasoning Answer (r_ar)	Analogous Problems: Q1: Sarah has a \$200 store credit limit. She spends the full amount in one trip. She pays back \$50 on Monday and \$70 on Friday. How much does she still owe? A1: Sarah starts with \$200 of debt. She pays back \$50 + \$70 = \$120. So, the remaining debt is \$200 - \$120 = $\boxed{80}$. Solve the Initial Problem: Q: Initial Problem A: Mary spent the full credit: \$100. She paid back \$15 on Tuesday and \$23 on Thursday. Total paid: \$15 + \$23 = \$38. Remaining to pay: \$100 - \$38 = $\boxed{62}$
Direct Prompt Answer (r_dp)	Total credit used: \$100. Payment on Tuesday: \$15. Payment on Thursday: \$23. Total paid so far: \$15 + \$23 = \$38. Remaining credit to be paid: \$100 - \$38 = \$62. Mary will need to pay \$62 before her next shopping trip.
Extracted Analogical Section (r_an)	Q1: Sarah has a \$200 store credit limit. She spends the full amount in one trip. She pays back \$50 on Monday and \$70 on Friday. How much does she still owe?A1: Sarah starts with \$200 of debt. She pays back \$50 + \$70 = \$120. So, the remaining debt is \$200 - \$120 = $\boxed{80}$.

Table 3: Example 1

Field	**codeforces200 Example**
Dataset	`codeforces200`
Initial Problem	Misha and Vasya participated in a Codeforces contest. Each of them solved one problem, submitted on first attempt. Misha solved a problem worth $a$ points at $c$ minutes, and Vasya solved one worth $b$ points at $d$ minutes. The final score is computed as: `score = max(3p/10, p - (p*t)/250)` Determine who scored higher.

Analogical Reasoning Answer (r_ar)	Analogous Problem: Alice solves a 200-pt question in 30min. Bob solves a 150-pt one in 20min.
	`Alice: max(3*200/10, 200 - 200*30/250) = max(60, 176) = 176`
	`Bob:   max(3*150/10, 150 - 150*20/250) = max(45, 138) = 138`
	Result: **Alice wins**.
	Mapping to Original Problem: This is analogous to comparing scores after time-adjusted penalties.
	Use the same scoring rule:
	`score = max(3p/10, p - (p*t)/250)`
	This is analogous to a disjoint set union (DSU) problem. We track original to current handle via mapping. E.g., given a sequence of changes `[("A", "B"), ("B", "C")]` we map original → final.
	Python3 Code:
	`def track_handle_changes(n, changes):`
	`original_to_current = {}`
	`current_to_original = {}`
	`for old, new in changes:`
	`original = current_to_original.get(old, old)`
	`original_to_current[original] = new`
	`current_to_original[new] = original`
	`current_to_original.pop(old, None)`
	`print(len(original_to_current))`
	`for orig, final in original_to_current.items():`
	`print(orig, final)`

Direct Prompt Answer (r_dp)	To determine who scored more, compute both scores using: `score = max(3p/10, p - (p*t)/250)` for both Misha and Vasya. Compare the two results.

Extracted Analogical Section (r_an)	**Analogous Problem:** Alice solves a 200-pt question in 30min. Bob solves a 150-pt one in 20min.
	`Alice: max(3*200/10, 200 - 200*30/250) = max(60, 176) = 176`
	`Bob:   max(3*150/10, 150 - 150*20/250) = max(45, 138) = 138`
	Result: **Alice wins**.

Table 4: Transposed example from codeforces200 showing reasoning and analogical mapping.