universität wien

MASTERARBEIT | MASTER'S THESIS

Titel | Title

Investigating LLM-as-a-Judge for Explanation Evaluation

verfasst von | submitted by

Vanessa Urban BSc

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2026

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 910
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Computational Science
Betreut von | Supervisor:	Univ.-Prof. Dr. -Ing. Benjamin Roth B.Sc. M.Sc.

Acknowledgements

I would like to thank Professor Benjamin Roth and Pedro Henrique Luz de Araujo for their continued guidance and advice throughout the development of this thesis. I am deeply grateful to my parents, whose strength and support over the years have made this achievement possible. My thanks go to my friends as well, especially Zsófi and Eva, who have shown me and continue to show me how strong and capable women are. Finally, I want to express my deepest gratitude to my partner Rahul, who has encouraged me from the start and has been the best support I could ever ask for.

Declaration

I acknowledge the use of AI tools in preparing this thesis, specifically Claude Opus and Gemini 3. They were used as writing assistance for rephrasing, grammar correction, and improving clarity. No AI tool was used to generate research content, results or interpretations.

Abstract

Explanations for large language models can only be trustworthy if they faithfully reflect the model's internal reasoning. Simulatability, which is the extent to which an observer can predict a model's behavior after seeing an explanation, offers a principled way to measure this faithfulness. Traditional simulatability studies rely on human subjects, which limits their scalability and reproducibility. This thesis investigates whether a human observer can be replaced by an LLM-as-a-Judge on a complex generative task, and whether input-level feature attributions improve the Judge's ability to simulate a target model's behavior. We formulate the problem as a five-way summary attribution task on the XSum dataset, in which the Judge must identify a target summary produced by DistilBART-xsum-12-6 from among four alternative candidates. Selene-1-Mini-Llama-3.1-8B serves as the Judge, and feature-based explanations are generated using the Multi-Level Explanations for Generative Language Models (MExGen) framework at both sentence and phrase level. A two-phase experimental design compares attribution accuracy without explanations (Phase 1) against accuracy with explanations (Phase 2), isolating the effect of the explanations directly.

Across all experimental variations, input feature attributions failed to meaningfully improve the Judge's attribution accuracy and in several configurations actively degraded it. The Judge consistently defaulted to the most fluent candidate rather than the Target Model. This behavior is driven by quality bias, sycophantic alignment, and post-hoc rationalization. Further analysis identifies lead bias at the sentence level and content overlap at the phrase level as two specific failure modes of input-level attributions in news summarization. These findings suggest that effective explanations for model attribution in generative tasks must characterize model style rather than input content.

Kurzfassung

Erklärungen für große Sprachmodelle können nur dann vertrauenswürdig sein, wenn sie die interne Logik des Modells getreu widerspiegeln. Simulierbarkeit, also das Ausmaß, in dem ein Beobachter das Verhalten eines Modells nach Erhalt einer Erklärung vorhersagen kann, bietet eine systematische Methode zur Messung dieser Genauigkeit (faithfulness). Herkömmliche Studien zur Simulierbarkeit basieren auf menschlichen Probanden, was ihre Skalierbarkeit und Reproduzierbarkeit einschränkt. Diese Arbeit untersucht, ob der menschliche Beobachter bei einer komplexen generativen Aufgabe durch ein LLM-as-a-Judge ersetzt werden kann und ob Input-Feature-Attributionen die Fähigkeit des Judges verbessern, das Verhalten eines Zielmodells zu simulieren. Wir formulieren das Problem als eine Summary-Attribution-Aufgabe auf dem XSum-Datensatz, bei der der Judge die vom Zielmodell DistilBART-xsum-12-6 erzeugte Zusammenfassung identifizieren muss, und zwar aus vier alternativen Kandidaten. Als Judge dient Selene-1-Mini-Llama-3.1-8B, und die Feature-Attributionen werden mit dem Multi-Level Explanations for Generative Language Models (MExGen) Framework sowohl auf Satz- als auch auf Phrasenebene erzeugt. Ein zweiphasiges experimentelles Design vergleicht die Attributionsgenauigkeit ohne Erklärungen (Phase 1) mit der Genauigkeit unter Einbeziehung von Erklärungen (Phase 2) und isoliert so den Effekt der Erklärungen.
Über alle experimentellen Variationen hinweg konnten Input-Feature-Attributionen die Attributionsgenauigkeit des Judges nicht signifikant verbessern und verschlechterten sie in einigen Konfigurationen sogar aktiv. Der Judge wählte durchgängig den flüssigsten Kandidaten anstelle des Zielmodells. Dieses Verhalten wird durch Quality Bias, sykophantisches Alignment und Post-Hoc-Rationalisierung bedingt. Weiterführende Analysen identifizieren Lead Bias auf Satzebene und Content Overlap auf Phrasenebene als zwei spezifische Fehlermodi von Input-Feature-Attributionen in der Nachrichtenzusammenfassung. Diese Ergebnisse legen nahe, dass effektive Erklärungen für die Modellzuordnung in generativen Aufgaben den Stil eines Modells charakterisieren müssen statt den Inhalt der Eingabe.

Contents

List of Tables

List of Figures

List of Algorithms

Listings

1. Introduction

Large Language Models (LLMs) are at the core of modern Natural Language Processing, but their internal decision-making remains largely opaque. These systems are increasingly deployed in high-stakes settings, so understanding *why* a model produces a particular output is one of the central objectives in the field of explainability. Explanations are intended to provide insight into the decision-making of a black box model, but an explanation is only useful if it accurately reflects the model's actual reasoning. Jacovi and Goldberg (2020) formalized the distinction between *plausibility* and *faithfulness*. When an explanation is plausible, it looks reasonable to a human, and when it is faithful, it accurately portrays the model's internal process. The authors warn that these two concepts have to be distinguished, since a convincing but unfaithful explanation can lead users to overtrust flawed predictions or overlook hidden biases. Faithfulness, therefore, must be evaluated on its own terms. One principled way to measure faithfulness is *simulatability* (Hase and Bansal, 2020). If an explanation truly conveys how a model works, then an observer who has seen that explanation should be better at predicting the model's behavior on new, unseen inputs. Traditional simulatability evaluations rely on human subjects, which makes them slow, expensive, and difficult to reproduce at scale. Pruthi et al. (2022) proposed replacing human observers with smaller student models, but this raises the question of whether the evaluation measures explanation quality or the student's learning capacity. A more recent alternative is to use a Large Language Model as the observer in a simulatability test (Chen et al., 2023). An LLM Judge can process natural language explanations directly, without task-specific training, and the LLM-as-a-Judge paradigm has shown strong alignment with human preferences on subjective evaluation tasks (Zheng et al., 2023). However, whether an LLM Judge can use input-level feature attributions for an objective evaluation of a complex generative task has not been directly tested. This thesis provides such a test. We frame simulatability as an attribution task, where the Judge identifies the Target Model's output among plausible alternatives rather than predicting it directly, since the output space in open-ended generation is too large for exact prediction. We investigate whether input-level feature attributions generated by the MExGen framework (Paes et al., 2025) improve an LLM Judge's ability to identify outputs produced by a specific generative Target Model. The task is framed as a five-way summary attribution problem on the XSum dataset. Given a source article and five candidate summaries, the Judge must select the one produced by the Target Model (DistilBART-xsum-12-6), with alternative summaries generated by Llama-3-8B-Instruct, Flan-T5-XL, a human-written gold summary, and a human-written summary randomly sampled from a different XSum article. We use Selene-1-Mini-Llama-3.1-8B as the Judge and compare attribution accuracy across two phases, a baseline without explanations (Phase 1) and the second phase in which MExGen attributions are added to the few-shot

prompt (Phase 2). Because the Judge retains no memory across inference calls, any change in accuracy between phases can be attributed directly to the explanations themselves. The thesis is organized around three research questions:

1. Baseline Attribution: Can an LLM Judge accurately identify a Target Model's output from few-shot examples alone, or do inherent evaluator biases prevent reliable performance?

2. The Impact of Feature-Based Explanations: Do input-level feature attributions improve the Judge's attribution accuracy, thereby demonstrating their utility as faithful representations of the Target Model's behavior? Does explanation granularity (sentence-level vs. phrase-level) affect this outcome?

3. Overriding Bias: Can advanced prompting strategies, such as contrastive examples, enable the Judge to use explanations effectively despite its inherent biases?

The empirical results are largely negative. MExGen feature attributions did not meaningfully improve attribution accuracy, and in several configurations actively degraded it. Rather than closing a methodological gap, these findings clarify *why* the approach fails in this setting. This thesis contributes a reproducible two-phase simulatability-style framework for evaluating explanations with an LLM Judge in place of human annotators. It provides systematic evidence of the biases that prevent small-scale LLM Judges from performing reliable model attribution in this setting, together with an analysis of the specific ways feature attributions fail in news summarization, namely lead bias and content overlap. Finally, it suggests a shift from content-level attribution to style-level characterization as a more promising basis for effective explanations in this task.

The remainder of the thesis is organized as follows: Chapter 2 reviews the relevant background on explainability, the LLM-as-a-Judge paradigm with its known biases, and feature-based explanation methods. Chapter 3 details the methodology, including the dataset, models, MExGen attribution pipeline, two-phase experimental design, and prompt engineering. Chapter 4 presents the empirical results across all experimental variations. Chapter 5 discusses why the approach failed, and Chapter 6 summarizes the findings, contributions and directions for future work.

2. Background and Related Work

2.1. Explainable AI: Faithfulness and Evaluation

Modern NLP systems increasingly rely on deep learning architectures and the need to understand their decision-making has grown correspondingly. This research field is called *explainability* and can be broadly defined as how well a model's internal reasoning can be explained in human terms (Lipton, 2018). Understanding why a model produces a particular output is essential for helping to expose underlying artifacts in the training data, like hidden biases or unintended patterns. Without this transparency, it is difficult to determine if a model is genuinely solving a given problem or if it is exploiting statistical cues and shortcuts present in the training data (Kaushik and Lipton, 2018). We want to diagnose model failures and gain trust in applications where incorrect predictions have real consequences (Kunkel et al., 2019). Explanations are intended to fill this gap, but an explanation is only useful if it aligns with the model's actual behavior. An explanation that sounds convincing, without corresponding to the model's internal reasoning, can cause users to overtrust these flawed predictions (Bansal et al., 2021), or hide harmful biases that would otherwise be detectable. Jacovi and Goldberg (2020) first formalized a critical distinction: the difference between an explanation's *plausibility* and its *faithfulness*. Plausibility captures whether an explanation appears reasonable to a human observer. Faithfulness captures whether it accurately portrays the model's internal reasoning process. These two properties are independent, so an explanation can be highly plausible yet unfaithful. Jacovi and Goldberg argue that treating these constructs interchangeably is dangerous and that faithfulness must be defined and evaluated on its own terms, independently of human judgments of explanation quality. This thesis adopts faithfulness as the central evaluation criterion. Existing approaches for evaluating faithfulness include perturbation-based tests, attention analysis, and predictive power evaluation (for a comprehensive survey, see Lyu et al. (2024)). This thesis builds on one specific approach which is called *simulatability*.

Simulatability as a Measure of Faithfulness The intuition behind simulatability is if an explanation faithfully conveys how a model works, then an observer who has seen that explanation should be able to predict what the model will do on new inputs. A model is *simulatable* when an observer can anticipate its behavior on unseen data, and the degree to which an explanation improves this ability is a direct measure of the explanation's faithfulness (Hase and Bansal, 2020). Hase and Bansal developed a rigorous experimental framework for measuring simulatability with human subjects. In their design, participants first establish a baseline accuracy by predicting model outputs from inputs alone. They

are then given algorithmic explanations of the model's behavior on separate training instances, and their prediction accuracy is again measured on the same test instances. The difference between these two measurements isolates the effect of the explanation. Their results did not yield clear improvements in simulatability for most of the explanation methods tested. In this thesis explanation methods that appear informative also do not necessarily help an observer predict how a system will behave.

Scaling Human Observers to LLM Judges Simulatability provides a principled evaluation framework, but in its original form it depends on human subjects, making it slow, expensive, and difficult to replicate at scale. Pruthi et al. (2022) proposed a model-based alternative by training smaller classifier models (the "students") to simulate a larger teacher model's predictions using explanations. When a student's accuracy improved after receiving explanations, the explanations were considered helpful. This approach eliminates human annotators, but it raises the question of whether the evaluation measures the explanation's quality or the student's learning capacity (influenced by the architecture and training of the student model). A third possibility lies in using Large Language Models (LLMs) as the observer in a simulatability test. An LLM Judge can process natural language instructions and explanations directly, without requiring task-specific training. Chen et al. (2023) demonstrated the feasibility of this approach in the context of counterfactual simulatability, a variant that evaluates whether explanations generalize to related inputs, not just the explained input. They found that GPT-4 approximates human simulators well on classification tasks. This thesis investigates whether the same idea extends further by replacing the human observer in the simulatability framework of Hase and Bansal (2020) with an LLM Judge and test whether it can be adapted and scaled to a complex generative NLP attribution task. However, using an LLM as the evaluator brings its own challenges, like the Judge having its own biases, preferences, and alignment constraints that may interfere with objective evaluation. These biases, and their consequences for explanation evaluation, are discussed in the following section.

2.2. LLM-as-a-Judge

The LLM-as-a-Judge paradigm uses Large Language Models to evaluate text outputs against predefined criteria. It provides much deeper and more context-aware assessment than traditional automatic metrics like BLEU or ROUGE (Gu et al., 2024). Zheng et al. (2023) validated this approach by demonstrating that GPT-4's evaluations align with human preferences over 80% of the time. Their work proposes using LLMs as automated judges for evaluating chatbots and introduces two foundational benchmarks: MT-Bench (80 specific multi-turn questions) and Chatbot Arena (a crowdsourced platform where models compete). More recently, Wang et al. (2025) investigated whether LLMs can evaluate the quality of machine learning explanations (including LIME and similarity-based explanations), comparing GPT-4o and Mistral-7B against 38 human judges in a iris classification scenario. Their findings reveal that LLMs performed well on subjective metrics such as understandability and usefulness, but they fell short on objective metrics.

GPT-4o showed no accuracy difference regardless of whether an explanation was provided. This suggests that the model does not rely on explanations the way humans do. In this thesis we test whether explanations help to improve an LLM Judge's performance on an objective attribution task.

Aligning Language Model with Human Preferences For aligning a language model with human values and preferences, the dominant approach is Reinforcement Learning for Human Feedback (RLHF), introduced by Christiano et al. (2023) and first applied at large scale by Ouyang et al. (2022) in InstructGPT (a general-purpose instruction-following language model). The first stage of RLHF is training a reward model on a dataset of human preference pairs to score responses according to judgments of human annotators. In the second stage the language model is fine-tuned using a reinforcement learning algorithm, typically Proximal Policy Optimization (Schulman et al., 2017), to maximize the reward model's scores while staying close to the original model. This process produces models that generate text humans rate as more helpful, harmless, and honest, but it also introduces biases toward fluent and detailed text (Zheng et al., 2023). Rafailov et al. (2024) introduced Direct Preference Optimization (DPO) as an alternative to RLHF which is simpler to implement and optimizes the same objective. It eliminates the need for a separate reward model and the RL training phase by using a reparameterization that maps the reward function directly to the optimal policy. DPO applies a classification loss to preference pairs, increasing log-probabilities of chosen responses relative to rejected ones. Both RLHF and DPO share the same underlying objective which is aligning a language model's outputs with human preferences. Also, both produce models with similar behavioral characteristics, including the tendency to prefer verbosity and fluency. The Judge model used in this thesis, Selene-1-Mini (Alexandru et al., 2025), was aligned using DPO and therefore is found to exhibit the same behavioral characteristics.

Training Small Language Model Judges (SLMJs) The Judge model Selene-1-Mini is an example of a small, open-weight model trained specifically for the evaluation task. It has 8-billion parameters and is fine-tuned from Llama-3.1-8B-Instruct to serve as a general-purpose LLM evaluator. Selene Mini's training combines *Direct Preference Optimization (DPO)* with an additional *Supervised Fine-Tuning (SFT)*. A known limitation of standard DPO is that it can decrease the probability of rejected responses without necessarily increasing the probability of chosen ones. To address this, Selene Mini adds a negative log-likelihood (NLL) term on the chosen responses, following the approach of Pang et al. (2024):

$$\mathcal{L}_{\text{DPO+NLL}} = \mathcal{L}_{\text{DPO}}\left((q_i^c, j_i^c), (q_i^r, j_i^r) \mid x_i'\right) + \alpha \mathcal{L}_{\text{NLL}}(q_i^c, j_i^c \mid x_i') \tag{2.1}$$

Equation 2.1 shows the loss function where x_i' is the evaluation prompt (containing the original query, the response(s) being evaluated, and the evaluation criteria), (q_i^c, j_i^c) is the chosen critique-judgment pair, (q_i^r, j_i^r) is the rejected pair, and α controls the weight of the SFT objective. The DPO component creates relative separation between chosen and rejected evaluations, while the NLL term directly maximizes the likelihood of producing the chosen output. Critiques were included in 70% of the training pairs and the remaining

30% contain judgments only. This training approach is relevant to this thesis because Selene Mini was trained to produce a critique before a judgment, which motivates the Chain-of-Thought output format used in the experimental prompts (Section 3.6).

2.2.1. Prompting Strategies for LLM Evaluators

The effectiveness of LLM evaluators depends not only on their training but also on how they are prompted at inference time. This section reviews two prompting strategies relevant to this thesis.

Chain-of-Thought Prompting Chain-of-Thought (CoT) prompting can improve the quality of the judge's evaluation by eliciting intermediate reasoning, while also adding interpretability by letting humans trace the reasoning. Wei et al. (2023) demonstrated that including step-by-step reasoning demonstrations in few-shot prompts significantly improves performance on reasoning tasks. However, the benefits emerge primarily at large model scales. Recent findings by Gu et al. (2024) show that requiring LLM evaluators to provide self-explanations alongside their scores or selections had a negative impact on evaluation performance. The authors suggest that the self-explanation process can reinforce existing biases rather than correcting them. When a model is forced to articulate reasons for its judgment, it can talk itself into flawed reasoning. This finding is consistent with the post-hoc rationalization documented in Chapter 5 of this thesis.

Contrastive Prompting Contrastive Chain-of-Thought prompting was proposed by Chia et al. (2023) and augments standard CoT demonstrations with both correct and incorrect reasoning examples. The key insight is that providing incorrect reasoning examples alongside valid ones helps the model produce more accurate outputs. The contrastive prompting strategy evaluated in this thesis (Section 3.7.2) adapts this principle for the attribution task.

2.2.2. Known Biases of LLM Evaluators

Despite LLM-as-a-Judge as a promising scalable evaluation paradigm, LLM evaluators show systematic biases that can weaken evaluation reliability. Understanding these biases is important for interpreting the experimental results of this thesis.

Quality and Verbosity Bias Models aligned through RLHF or DPO develop a preference for text that is fluent, highly structured, and detailed, reflecting the characteristics that human annotators rewarded during preference data collection. In an evaluation setting, this manifests as a tendency to favor longer, more comprehensive-sounding responses regardless of their correctness or relevance to the task (Zheng et al., 2023; Ye et al., 2024). Zheng et al. (2023) tested this by creating artificially inflated versions of model answers without adding new information. Weaker judges such as GPT-3.5 and Claude-v1 favored the longer answer over 91% of the time, even when the prompt explicitly instructed them not to let length influence their evaluation. Ye et al. (2024) confirmed this pattern by

expanding lower-quality answers in a pair with semantically redundant sentences while keeping factual content identical. They found that judges shifted their preference toward the longer but still inferior response.

Position Bias LLM evaluators also exhibit a tendency to favor responses based on their position in the prompt rather than their actual quality. Wang et al. (2024) demonstrated that simply swapping the order of two responses can flip the evaluation outcome entirely, and that by manipulating response order, a weaker model can be made to appear superior to a stronger one. Shi et al. (2025) confirmed this at scale across 15 judges and over 150,000 evaluation instances, showing that position bias is a systematic effect that varies by both judge and task. Both studies find that the bias is most pronounced when candidate responses are close in quality. Critically for this thesis, Shi et al. (2025) also extended their analysis from pairwise to list-wise comparisons and found that weaker models degraded significantly when the number of candidates increased. Zheng et al. (2023) proposed mitigating this bias by presenting candidates in both orderings and only declaring a winner when the judgment is consistent. This thesis uses an alternative by randomly shuffling the order of all five candidates in the 5-way multiple-choice format for every inference call (Section 3.6).

Self-Preference and Family Bias LLM evaluators have been shown to systematically favor their own outputs or those produced by architecturally related models (Spiliopoulou et al., 2025). Wataoka et al. (2024) found that this preference is driven by the perplexity of the text. LLM evaluators give higher ratings to texts that are statistically familiar to their own distribution, meaning text with lower perplexity, regardless of which model generated them. Chen et al. (2025) draw an important distinction between legitimate self-preference, where the model favors its own objectively superior output, and harmful self-preference, where it favors its own output despite lower quality. They found that stronger models exhibit more pronounced harmful self-preference when they make a mistake, suggesting that greater capability does not eliminate this bias. They propose generating a long CoT before evaluation, which reduces the harmful self-preference. This thesis conducted a control experiment to isolate the effect of family-bias (Section 3.7.1).

Sycophancy and Post-Hoc Rationalization A related class of biases concerns the faithfulness of LLM-generated reasoning. *Sycophancy* refers to models adjusting their evaluation to align with stated user opinions, abandoning correct answers when challenged and agreeing with incorrect user claims. Sharma et al. (2023) demonstrated that sycophantic behavior, the agreement with user beliefs, is prevalent across models and tasks and is among the strongest predictors of which responses humans prefer. This suggests that sycophancy is a structural consequence of training on human preferences (e.g., RLHF) and is not a deficiency of any particular model. This tendency to prioritize agreeableness over accuracy extends to Chain-of-Thought reasoning. CoT explanations can function as plausible *post-hoc rationalizations* rather than faithful accounts of the model's decision process, which makes relying on CoT for transparency risky

(Turpin et al., 2023). Models sometimes largely ignore their own stated reasoning and larger, more capable models produce less faithful CoT than smaller ones, as they can already answer without relying on intermediate steps (Lanham et al., 2023). These findings raise a concern for the interpretation of Judge outputs in this thesis. The Judge's reasoning may reflect post-hoc justification of a predetermined selection rather than the actual basis for the decision. This possibility is examined in Chapter 5.

Compassion-Fade Bias Ye et al. (2024) identify a bias they term *compassion-fade bias*, where revealing model identities to the judge allows pre-training knowledge about model reputations to influence evaluation. They demonstrate that LLM judges sometimes rely on contextual signals like names, citations, and identity markers as shortcuts rather than carefully evaluating the actual content. In this thesis, candidates are labeled neutrally (Option A through E) and the identity of the dataset or the models is not revealed in the prompt.

2.3. Feature-Based Explanations

Feature-based explanation methods assign importance scores to individual input features, quantifying how much each feature contributed to a model's output. In NLP, these features are typically tokens, phrases, or sentences from the input text. Common approaches include gradient-based methods and perturbation-based methods. Gradient-based methods compute importance scores from the model's internal gradients and require access to model internals. Perturbation-based methods measure how the output changes when input features are removed or altered, treating the model as a black-box and querying it repeatedly on modified inputs. This thesis uses a perturbation-based approach, an extension of the LIME framework (Ribeiro et al., 2016) to generative models through MExGen (Paes et al., 2025).

2.3.1. Input-Level Attributions: LIME

Local Interpretable Model-Agnostic Explanations (LIME) was introduced by Ribeiro et al. (2016) and is a post-hoc explanation method that approximates the behavior of any black-box model locally around a single prediction. LIME fits a simple, interpretable model like linear regression to the model's output after perturbing the input. The coefficients of this simple model represent the importance scores for each input feature. Formally, for an input x, LIME generates n perturbed instances $z^i \in 0, 1^d$ by randomly removing input features, where each binary component indicates the presence or absence of a feature. Then it queries the black-box model f on each perturbation and solves the following objective (Ribeiro et al., 2016):

$$\xi = \arg\min_{w} \sum_{i=1}^{n} \pi(z^{(i)}) \left(w^T z^{(i)} - f(z^{(i)}) \right)^2 + \lambda R(w) \tag{2.2}$$

The objective finds a weight vector w that minimizes the weighted squared difference between the linear model's predictions $w^T z^{(i)}$ and the black-box model's outputs $f(z^{(i)})$. The proximity kernel $\pi(z^{(i)})$ weights perturbations by their similarity to the original input. $R(w)$ is a regularization term weighted by λ, that encourages sparsity of non-zero weights. The resulting coefficients w_s of the weight vector quantify how much input feature s contributed to the model's prediction.

2.3.2. MExGen: Multi-Level Explanations

The Multi-Level Explanations for Generative Language Models (MExGen) framework, proposed by Paes et al. (2025), extends standard input-level attribution methods like LIME and SHAP (Lundberg and Lee, 2017) to long-form text generation.

Why standard LIME does not scale to long-form generation Standard feature attribution methods were designed for classification tasks, where models produce a single scalar output per input. Generative language models produce a sequence of tokens instead of a scalar prediction, and input documents in tasks like summarization can contain hundreds of tokens. This makes performing exhaustive perturbations at a fine-grained level computationally unfeasible.

Scalarization To apply LIME to generative models, the model's sequential output must be reduced to a single scalar score. MExGen introduces *scalarizers*: functions S that map output text to real numbers. This thesis uses the *Log Probability* scalarizer, which measures how likely the model is to reproduce its original output y^o given a perturbed input x. Given an output sequence y^o and sequence length l, the Log Prob scalarizer is defined as (Paes et al., 2025):

$$S(x; y^o, f) = \frac{1}{\ell} \sum_{t=1}^{\ell} \log p(y_t^o \mid y_{<t}^o, x; f) \tag{2.3}$$

In this setting, perturbation means removing input units (e.g., sentences) from the original document. If removing a sentence causes a large drop in the scalarized output, that sentence was important to the model's generation. Replacing $f(z^i)$ in the standard LIME objective (Equation 2.2) with the scalarizer S yields the *Constrained LIME (C-LIME)* objective (Paes et al., 2025):

$$\xi = \arg\min_{w} \sum_{i=1}^{n} \pi(z^{(i)}) \left(w^T z^{(i)} - S(x^{(i)}; y^o, f) \right)^2 + \lambda R(w) \tag{2.4}$$

C-LIME also represents a more efficient, linear-complexity variant of LIME by limiting the number of perturbations n to a small multiple of the number of input units d. The number of simultaneously perturbed units is set to a small integer K. The resulting coefficients w assign an importance score to each input unit, quantifying its influence on the Target Model's generation.

Multi-Level Unit Refinement A single pass of C-LIME at sentence level identifies which sentences are most influential to model behavior but cannot reveal which specific phrases within those sentences drive the generation. MExGen addresses this by using a hierarchical refinement process. Attribution scores are first computed at sentence level and the most important sentences, selected based on a normalized threshold ϕ and a maximum count k, are then split into finer-grained units such as phrases. C-LIME is applied again at this lower level, producing multi-level explanations. The formal selection procedure and the specific parameters used in this thesis are described in Section 3.4.

2.3.3. Limitations of Feature-Based Attributions

While input-level attributions identify *which* parts of the input influenced a model's output, they do not explain *why* those parts were influential or *how* the model used them. Sun et al. (2025a) showed that token-level attributions can emphasize tokens with little semantic weight, such as prepositions, and that it remains unclear whether such patterns reflect model behavior or artifacts of the explanation method itself. This fundamental limitation has motivated a shift toward natural language explanations, where the explanation is a readable description of the model's reasoning rather than a set of numerical scores. Camburu et al. (2018) argued that numerical attributions alone are insufficient for understanding model behavior and proposed augmenting predictions with human-readable explanations. The experimental results of this thesis (Chapter 4) provide further evidence for this gap.

3. Methodology

3.1. Task Formulation: Simulatability as Model Attribution

A good explanation should enable an observer to understand why an AI system produces a particular output. If an explanation is effective, the observer should be able to anticipate how the model will behave on new, unseen data. In the field of explainable AI, this is known as *simulatability* (Hase and Bansal, 2020). The central objective of this thesis is to investigate whether input-level feature attributions, provided as explanations, improve the ability of an LLM-as-a-Judge to recognize the behavior of a target generative model. Prior work on simulatability has relied primarily on human studies, which are costly and difficult to scale. More recently, research has explored the use of Large Language Models as automated evaluators of explanations. This thesis adopts the LLM-as-a-Judge framework and applies it to a complex generative Natural Language Processing (NLP) task: news summarization. Because the output space of open-ended generation is too large for exact prediction to serve as a meaningful criterion, we adapt the classical simulatability paradigm into a *Summary Attribution Task*. In this task, the LLM Judge acts as the observer. It first reviews a small number of examples demonstrating the Target Model's behavior, its stylistic tendencies, content selection patterns, and characteristic errors such as hallucinations. The Judge is then presented with a new, unseen source document and has to identify the Target Model's summary from a pool of several distractor summaries. By introducing input-level explanations into this setup, we can measure whether they increase attribution accuracy. If the accuracy remains unchanged or declines, this would indicate limitations in the explanation method, the Judge's capacity to leverage explanations, or the Judge's inability to overcome intrinsic biases when evaluating generative text.

3.2. Dataset: Extreme Summarization (XSum)

This thesis uses the Extreme Summarization (XSum) dataset (Narayan et al., 2018), which is originally constructed from British Broadcasting Corporation (BBC) news articles. It is designed for single-sentence summarization and requires models to generate highly abstractive and compressed summaries. Each data entry consists of a source document (the news article), a human-written introductory sentence for that article (which is serving as the *Gold summary*), and the unique article identifier. The full dataset contains 204,045 training examples, 11,332 validation examples and 11,334 test examples.

Subset Selection and Computational Constraints Due to the intensive computational requirements of generating multi-level explanations via the MExGen framework and run-

ning multiple inference passes for the Judge, experiments were conducted on representative subsets of the XSum data. Specifically, the first 100 documents of the validation split were used for prototyping and prompt optimization, including few-shot example selection and formatting refinement. The first 200 documents of the test split were reserved for final evaluation. This separation ensures that all prompt design decisions were finalized before evaluation on the test set, preventing prompt overfitting and keeping the test set results unbiased. All experiments were conducted using NVIDIA T4 (16GB VRAM), L4 (24GB VRAM), or A100 (40GB VRAM) GPU's within the Google Colab environment.

Language Filtering During initial data exploration, approximately 44 articles in the validation split and 37 in the test split were found to be written in regional dialects or non-English languages (such as Welsh and Scottish Gaelic). Since the LLM Judge is primarily trained for English-language evaluation, including these documents could compromise the reliability of the evaluation. A preprocessing step was implemented to filter out non-English documents from both splits prior to example selection and evaluation. This was done using the `langid` tool in Python.

Adversarial Characteristics Abstractive summarization represents a foundational generative NLP task, and XSum's properties create a stress test for the LLM Judge. Maynez et al. (2020) demonstrated that over 70% of the human-written gold summaries in XSum contain external knowledge or entities that do not appear anywhere in the source document. Models fine-tuned on this data learn to replicate this behavior and tend to generate hallucinated details as well. Another study by Pagnoni et al. (2021) reported that over 90% of model-generated summaries on XSum are factually incorrect, indicating that while models imitate the stylistic patterns of human summaries, they lack the world knowledge to include factually correct content. This characteristic presents a unique challenge for the Judge. The Target Model, trained on XSum, will generate hallucinated content that may be factually wrong. To correctly attribute summaries, the Judge has to recognize and favor this behavior, even when it conflicts with the Judge's own internal biases.

3.3. Models: The Judge and Candidate Models

This experimental setup involves three distinct roles: a Target Model whose behavior the Judge must learn to simulate, two alternative generative models that serve as distractors in the attribution task, and the Judge Model itself. In addition to the three model-generated summaries, two human-written summaries from the XSum dataset are included as candidate answers, bringing the total number of options to five per test instance.

3.3.1. Target Model: DistilBART-XSum-12-6

DistilBART was chosen as the Target Model, specifically the **DistilBART-xsum-12-6** variant (approximately 306 million parameters). DistilBART is a distilled version of the BART-large architecture, fine-tuned on the XSum and CNN/DailyMail datasets for

one-sentence summarization. Its training on XSum produces the abstractive, hallucination-prone summarization style characteristic of the dataset, making it a challenging target for the Judge to identify. Unlike closed-source API models, DistilBART's open-weight architecture provides full access to output logits, which is required for computing the multi-level explanations in the MExGen framework (Paes et al., 2025) (2.3.2).

3.3.2. Candidate Models: LLama-3-8B-Instruct and Flan-T5-XL

Two additional generative models provide alternative summaries, creating a complex attribution task. The first is **Llama-3-8B-Instruct**, a highly fluent instruction-tuned transformer. Following the protocol established by Paes et al. (2025), this model was prompted with the instruction: "Summarize the following article in one sentence. Do not preface the summary with anything." Due to hardware constraints, the quantized **Llama-3-8B-Instruct-bnb-4bit** version (AI@Meta, 2024) was used with greedy decoding and a generation limit of 100 tokens. The Llama-3 model characteristically produces more detailed and verbose summaries, providing a stylistically distinct candidate. The second candidate is the instruction-tuned **FLAN-T5-XL**, an encoder-decoder model. While Paes et al. originally used the 20-billion parameter FLAN-UL2 model in their setup, FLAN-T5-XL was substituted due to GPU memory limitations and restricted API access. It was loaded in 4-bit precision using the `BitsAndBytes` library (Dettmers et al., 2022) and evaluated with greedy decoding and a 100-token generation limit. This model did not require an additional system prompt.

3.3.3. Judge Model: Selene-1-Mini-Llama-3.1-8B

As the Judge Model, **Selene-1-Mini-Llama-3.1-8B** (Alexandru et al., 2025) was used, a state-of-the-art small language model-as-a-judge post-trained from LLama-3.1-8B-Instruct. Selene Mini was explicitly trained via Supervised Fine-Tuning (SFT) and Direct Preference Optimization (DPO) to evaluate generative outputs by producing a critique before assigning a score. It can be used as a general-purpose evaluation model which supports diverse input formats and scoring scales.

Deployment Due to memory limitations, running the full 8-billion parameter model in 16-bit precision was not feasible. The model was loaded using 4-bit quantization via `BitsAndBytes`. While quantization can lead to a slight degradation in reasoning power for more complex evaluation tasks (Liu et al., 2025), the same quantization level was applied consistently across both experimental phases. Since this thesis measures the *relative change* in attribution accuracy between Phase 1 (without explanations) and Phase 2 (with explanations), rather than absolute performance, the use of 4-bit quantization does not compromise the validity of the results.

3.4. Explanation Generation: Multi-Level Input Attribution

The explanations evaluated in this thesis take the form of input-level feature attributions generated by the Multi-Level Explanations for Generative Language Models (MExGen) framework (Paes et al., 2025). The original MExGen work focused on producing faithful explanations for human evaluation and automated perturbation metrics. This thesis extends the framework by integrating these explanations into an LLM-as-a-Judge pipeline to assess whether they improve automated model understanding.

3.4.1. Constrained LIME and Scalarization

We adopt a specific setting from Paes et al. that uses DistilBART-xsum-12-6 as the Target Model to be explained, the Constrained LIME (C-Lime) explainer, and the Log Prob scalarizer. C-LIME quantifies the importance of individual input text units (e.g., sentences) to the generated summary by repeatedly querying the summarization model on perturbed versions of the input. Then fitting a local linear model to the resulting output changes. Each unit receives an importance score reflecting its influence on the Target Model's generation. Following the parameter settings of Paes et al. for smaller models like DistilBART, the proportionality constant between the number of perturbations n and the number of units d was set to $n/d = 10$ (oversampling factor), with the maximum number of simultaneously perturbed units set to $K = 3$.

3.4.2. Multi-Level Unit Refinement

The public release of the MExGen codebase, the In-Context Explainability 360 (ICX360) toolkit (Wei et al., 2025), supports only single-level explanations (e.g., sentence-level, phrase-level or word-level). To achieve multi-level unit refinement, we follow the unit refinement procedure in Algorithm 1, defined by Paes et al. (2025). The corresponding Python implementation is provided in Appendix A.1. This algorithm selects the most important units for refinement based on two criteria: their scores must be larger than a predefined threshold ϕ and they must rank among the top-k highest-scoring units. Following the settings in Paes et al. for DistilBART summarization with sentence-level C-LIME scores, the maximum number of refined units was set to $k = 3$ (the top 3 sentences), and the significance threshold was set to $\phi = 1/3$.
The full multi-level attribution process follows three stages:

1. **Sentence-Level Attribution:** C-LIME is applied to the entire input document to obtain importance scores for each sentence.

2. **Phrase-Level Selection:** Algorithm 1 identifies the sentences whose scores exceed the threshold ϕ and rank among the top-k, selecting them for deeper refinement.

3. **Phrase-Level Attribution:** C-LIME is applied again to each selected sentence individually, producing fine-grained importance scores at phrase level

Input: $x_1, \ldots, x_d$ (Current units, input is split into d units)
$k \leq d$ (Maximum number of units to refined at most)
$\phi \in [-1, 1]$ (Significance threshold, minimum normalized important score)

Output: units_to_be_refined

Compute attribution scores ξ using the C-LIME objective (Equation 2.4)

$\psi \leftarrow 2\frac{\xi - \min_s \xi_s}{\max_s \xi_s - \min_s \xi_s} - 1$ (Normalize scores to range $[-1, 1]$)

Compute Top-$k(\psi)$ (sort normalized scores, identify top-k)

units_to_be_refined $\leftarrow \emptyset$

for $s \in [d]$ **do**

  if $\psi_s \geq \phi$ ***and*** $\psi_s \in$ Top-$k(\psi)$ **then**

    units_to_be_refined.add(x_s);

  end

end

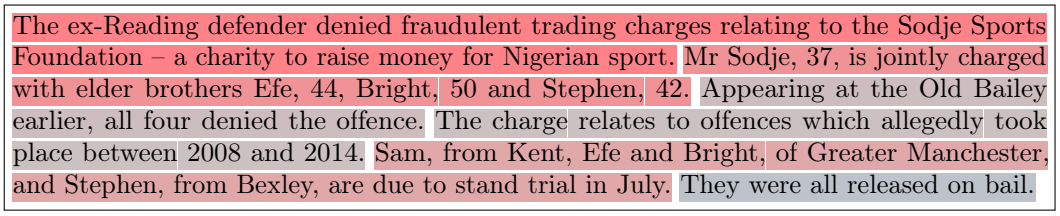
return units_to_be_refined;

Algorithm 1: Unit Refinement (Paes et al., 2025)

Visualizing Attributions The C-LIME explainer outputs a dictionary mapping input units to their importance scores. These scores can be formatted as highlighted spans of the input text, where brighter highlighting corresponds to higher importance for the Target Model's generation. Visualization was performed using the `color_units` function from the ICX360 toolkit. Figures 3.1 and 3.2 reproduce this output with the same color scheme for presentation in this thesis. Figure 3.1 illustrates sentence-level attributions for the first XSum document from the validation set, with the corresponding DistilBART summary shown for reference. Figure 3.2 shows the phrase-level refinement of the most influential sentence.

3.5. Two Phase Experimental Setup

We evaluate explanation faithfulness through an attribution-based simulatability framework, adapted from human simulatability studies (e.g.,Hase and Bansal (2020)). A faithful explanation, by definition, should accurately convey how the Target Model transforms its input into its output. If this information is successfully communicated to the Judge, the Judge's ability to recognize the Target Model's outputs on unseen inputs should improve. This experimental framework is divided into two independent phases: Phase 1 (baseline, without explanations) and Phase 2 (with explanations). A key property of this design is that, unlike human evaluators, the LLM Judge does not retain memory across separate inference calls. Any change in attribution accuracy between phases can therefore be attributed to the presence of explanations. An improvement in attribution accuracy from Phase 1 to Phase 2 would provide evidence that the explanations carry information that



Summary generated by DistilBART: *Former Premier League footballer Sam Sodje has appeared in court charged with fraud.*

Figure 3.1.: Sentence-level C-LIME attributions for a sample XSum document, produced by the MExGen framework with DistilBART as the Target Model. Color intensity corresponds to each sentence's importance score for the generated summary, where brighter highlighting indicates a stronger influence. The most influential sentence (brightest) describes the core fraud charges. The DistilBART summary is shown below the article for reference.

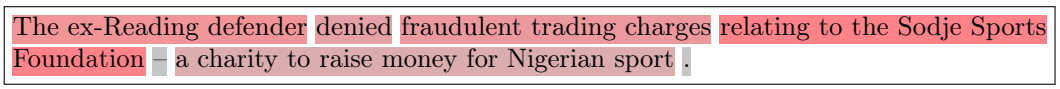


Figure 3.2.: Phrase-level C-LIME refinement of the most influential sentence identified in Figure 3.1. After sentence-level attribution, Algorithm 1 selects the top-ranked sentence for a second C-LIME pass at phrase granularity. The resulting phrase-level scores identify "relating to the Sodje Sports Foundation" as the span most responsible for DistilBART's summary.

helps the Judge recognize the Target Model's behavior.

3.5.1. Phase 1: Baseline Accuracy (Without Explanations)

In Phase 1, the Judge is provided with a few-shot learning prompt that contains example documents paired with the corresponding summary generated by the Target Model. For each test instance, the Judge is presented with a new source document and five candidate summaries, and must select the one produced by the Target Model. The five candidates are:

1. **Target Model:** The summary generated by DistilBART.
2. **Llama-3:** A characteristically detailed and more verbose summary.
3. **Flan-T5:** An alternative model-generated summary.
4. **Gold:** The human-written ground truth summary from the XSum dataset.
5. **Random:** A human-written summary randomly sampled from a different XSum document.

Since no exact duplicate answers exist among the candidates and there is only one correct answer per question, the random-guessing baseline is 20%.

3.5.2. Phase 2: Accuracy with Explanations

Phase 2 follows the same structure as Phase 1, with one modification. Now the few-shot examples include input-level attributions generated by the MExGen framework alongside the documents and Target Model outputs. For each test instance, the Judge again selects from the same five-candidate format. By comparing attribution accuracy between Phase 2 and Phase 1, we assess whether the explanations improve the Judge's understanding of the Target Model's behavior. A significant increase in accuracy would indicate that the explanations effectively guide the Judge toward the correct attribution.

3.6. Prompt Engineering

The performance of an LLM-as-a-Judge also depends on prompt design. To maximize the Judge's reasoning capability, all prompts were engineered to align with the format and conventions that Selene Mini was exposed to during training.

3.6.1. Prompting Architecture

All prompts were wrapped using the official Llama-3 chat template via `tokenizer.apply_chat_template`, following the reference implementation provided by the Selene Mini developers (Alexandru et al., 2025). The developers also provide prompt templates used during training and recommended for optimal performance. These templates enforce a structured output format consisting of a `**Reasoning:**` field followed by a `**Result:**` field. This activates the Judge's CoT processing and is consistent with the model's training setup. The original pairwise comparison template (Response A versus Response B) was adapted for the 5-way multiple-choice format used in this thesis. While Selene Mini is described as "robust to variations in prompt format" (Alexandru et al., 2025), a pairwise control experiment was conducted to isolate whether the adaptation from a pairwise to a 5-way format affects evaluation quality (Section 3.7.3). Greedy decoding was used to ensure deterministic, reproducible evaluations.

3.6.2. Few-Shot Example Selection Strategies

Few-shot prompts were constructed dynamically by retrieving semantically relevant examples for each test instance. All documents were embedded using the **all-mpnet-base-v2 model** (768 dimensions, maximum sequence length 384) from the `SentenceTransformers` library. Two retrieval strategies were compared:

1. Dynamic Similarity Retrieval

For each test instance, the three most semantically similar documents from the corresponding set (validation or test) were retrieved based on embedding distance. Since the

embeddings are normalized, minimizing Euclidean distance is equivalent to maximizing cosine similarity. 2D Principal Component Analysis (PCA) was used to visualize the embedding space and confirm that this method provides localized and topic-specific examples, which may help the Judge infer DistilBART's summarization style for specific subject matters.

2. Dynamic Cluster-Based Retrieval

To test whether a more diverse set of examples benefits the Judge, a *K-Means Clustering* approach was implemented. The embedding space was partitioned into k = 3 clusters using K-Means++ initialization. For each test instance, the single closest document from each cluster was retrieved, ensuring a globally diverse representation of DistilBART's summarization behaviors while maintaining relevance to the test input. Additional cluster counts were explored during development and the corresponding visualizations are provided in Appendix A.5.

3.6.3. Presenting Explanations in the Prompt

A key challenge in using feature attributions within a language model prompt is the format in which they are presented. As Zytek et al. (2024) observed, both humans and LLMs struggle to interpret raw numerical score arrays produced by explainers such as SHAP or LIME. In human evaluation studies, including the original MExGen evaluation by Paes et al. (2025), attributions are typically presented as visually highlighted text rather than raw probabilities. This thesis adapts this principle for the LLM Judge by extracting the most influential sentences and phrases (based on MExGen importance scores) and stating them explicitly in natural language within the prompt. This approach allows a direct comparison of sentence-level attributions versus multi-level (sentence + phrase) attributions. Rather than presenting raw numerical scores, the prompt provides direct, readable text identifying why the Target Model generated a specific summary.

3.6.4. Prompt Construction and Bias Mitigation

The prompt incorporated several mitigation strategies for the LLM judge biases discussed in 2.2.2. The Selene Mini prompt template (Alexandru et al., 2025) explicitly instructed the Judge not to prefer responses based on length or position, addressing verbosity and position bias respectively. Additionally, the order of the five candidates was randomly shuffled for every inference call to mitigate position bias. To mitigate compassion-fade bias, the identity of the dataset and the candidate models was not revealed in the prompt. The same prompt structure was applied consistently across all test instances. The finalized prompt template is provided in Appendix A.2 for Phase 1 and the modification for Phase 2 in A.3. .

3.7. Experimental Variations

In addition to the main two-phase comparison, several controlled variations were introduced to isolate potential confounding factors and assess the robustness of the results. Each variation modifies one aspect of the experimental setup while holding all other variables constant.

3.7.1. Self-Preference Bias Control

LLM-as-a-Judge models frequently exhibit self-preference bias or family-bias (Section 2.2.2), where the evaluator systematically favors outputs generated by models from its own architectural family. Since the Selene Judge is fine-tuned from the Llama-3 architecture, it was likely to systematically select the Llama-3 candidate over the Target Model DistilBART. To test this hypothesis, we conducted a control experiment in which the Llama-3 candidate was removed from the candidate pool, and had the Judge evaluate the remaining candidates in both phases.

3.7.2. Contrastive Prompting

Initial experiments revealed that standard prompting often failed to overcome the Judge's inherent quality bias, its tendency to prefer fluent and verbose summaries. To address this, a contrastive prompting strategy was introduced in Phase 2, building on the principles discussed in Section 2.2.1. In each few-shot example, the Llama-3 output was explicitly presented as a negative example, accompanied by the instruction that this summary was *not* produced by the Target Model and does *not* match its summarization behavior. The Judge was further instructed to consider, during its reasoning, how the Target Model's output should *not* look like. An example of how the prompt template was modified to include the contrastive element is provided in Appendix A.4.

3.7.3. Pairwise Control Experiment Setup

In Section 3.6 we noted a potential limitation regarding the task complexity and the high cognitive load placed on the LLM-Judge. The primary experimental design uses a 5-way multiple-choice format, requiring the Judge to evaluate five different summaries simultaneously. To verify that this format was not suppressing the Judge's attribution capability, a pairwise control experiment was conducted. In this setting, the task was simplified to a tournament-style head-to-head comparison. Using the first 200 documents of the XSum test set with similarity-based few-shot prompts, each model was evaluated against every other model in direct pairwise matchups. The Random candidate was excluded from the tournament based on its negligible selection rates in prior experiments. This resulted in three matchups per document for each model and a total of 600 trials per model in each phase. Performance was measured as *win rate*, representing the percentage of trials in which a candidate was selected by the Judge out of its total matchups. The prompt template for pairwise evaluation, adapted from the Selene Mini template Alexandru et al. (2025) is provided in Appendix A.6.

4. Results

This chapter presents the empirical findings from the evaluation pipeline described in Chapter 3. The main objective is to determine whether providing an LLM-as-a-Judge with input-level feature attributions improves its ability to correctly identify text generated by a specific Target Model. The focus here is on presenting the experimental data and direct outcomes, while broader interpretations and implications are discussed in Chapter 5. This chapter is organized as follows. Section 4.1 establishes the Phase 1 baseline performance. Section 4.2 presents the Phase 2 results with MExGen attributions at varying granularities. Section 4.3 compares the two few-shot selection strategies. Sections 4.4, 4.5, and 4.6 present the control experiments, covering family bias, the contrastive prompting variation, and the pairwise format control.

4.1. Phase 1: Baseline Performance

Before introducing feature attributions, it was necessary to establish the Judge's baseline capability for identifying the Target Model from the few-shot examples alone.
The Judge was evaluated on a validation subset of 100 instances using three dynamically retrieved few-shot examples based on semantic similarity of document embeddings. The prompt contained no explanations. The examples displayed only the source article and the corresponding Target Model summary. The exact prompt format is provided in Appendix A.2. Table 4.1 shows the distribution of the Judge's selections across the five candidate summaries over five independent runs with randomized candidate ordering. Standard deviations are reported to quantify run-to-run variability. The baseline attribution accuracy on the validation set is $6.4 \pm 2.2\%$, indicating that the Judge fails to identify the Target Model (DistilBART). This accuracy falls below the random guessing baseline of 20%, which suggests that the Judge is not guessing randomly but instead systematically selecting other candidates. The Judge overwhelmingly favored the Llama-3-8B-Instruct model, selecting it $82.8 \pm 2.6\%$ of the time. The Gold (human-written) summaries were selected $7.8 \pm 1.9\%$ of the time, Flan-T5 summaries $3.0 \pm 1.6\%$, and the Random summary was never selected. The consistent rejection of the Random candidate indicates that the Judge can dismiss semantically incoherent summaries, but it lacks the ability to distinguish the Target Model's output from more fluent alternatives. Table 4.2 compares the Judge's average candidate selection between the validation set (100 instances, 5-run average) and the unseen test set (200 instances, 3-run average). The test set evaluation was limited to three runs due to the increased computational cost of the larger dataset. The test set results closely mirror the validation findings, with a baseline attribution accuracy of $6.2 \pm 0.6\%$. The reduced variance in the test set is consistent with the larger

sample size (200 vs. 100 instances). The persistence of the strong Llama-3 preference across both sets confirms that the Judge's inability to attribute DistilBART's output is a characteristic of the evaluation setup, not an artifact of the validation subset.

Candidate Model	**1**	**2**	**3**	**4**	**5**	**Mean ± SD (%)**
Llama-3	84	79	86	82	83	82.8 ± 2.6
Gold	6	11	7	7	8	7.8 ± 1.9
DistilBART	**7**	**5**	**5**	**10**	**5**	**6.4 ± 2.2**
Flan-T5	3	5	2	1	4	3.0 ± 1.6
Random	0	0	0	0	0	0.0 ± 0.0

Table 4.1.: Phase 1 baseline selection distribution on the XSum validation set, across 5 independent runs with randomized candidate ordering and similarity-based few-shot retrieval. The Judge (Selene-1-Mini) attributes the Target Model (DistilBART) correctly in only 6.4 ± 2.2% of cases, well below the 20% random guessing baseline.

Candidate Model	**Validation Set (100 inst.)** Mean ± SD of 5 runs (%)	**Test Set (200 inst.)** Mean ± SD of 3 runs (%)
Llama-3	82.8 ± 2.6	84.5 ± 0.9
Gold	7.8 ± 1.9	6.5 ± 1.0
DistilBART	**6.4 ± 2.2**	**6.2 ± 0.6**
Flan-T5-XL	3.0 ± 1.6	2.8 ± 1.3
Random	0.0 ± 0.0	0.0 ± 0.0

Table 4.2.: Phase 1 baseline comparison between the 100-instance validation set (5-run average) and the unseen 200-instance test set (3-run average). The Target Model (DistilBART) attribution accuracy of 6.2 ± 0.6% on the test is close to the 6.4 ± 2.2% on the validation set. The Llama-3 candidate has the highest selection rate observed in both sets and the Random summary was never selected.

4.2. Phase 2: MExGen Feature Attributions

This section reports the results of adding input feature attributions to the Judge's prompt. To evaluate the effect of attribution granularity, four distinct formats were tested:

1. **Top Sentence:** The single source sentence with the highest attribution score.

2. **Top 2 Sentences:** The two source sentences with the highest attribution scores.
3. **Top Phrase:** The phrase with the highest attribution score from each refined source sentence.
4. **Top 2 Phrases:** The two phrases with the highest attribution scores from each refined source sentence.

4.2.1. Validation Set Performance

The four attribution formats were evaluated on the 100-instance validation set, each tested across five independent runs with similarity-based few-shot retrieval. Table 4.3 presents the selection distributions with standard deviations alongside the Phase 1 baseline.

		Sentence-Level (%)		**Phrase-Level (%)**	
Candidate Model	**Baseline (%)**	Top 1	Top 2	Top 1	Top 2
Llama-3	82.8 ± 2.6	83.4 ± 2.3	82.4 ± 4.0	85.4 ± 3.0	82.6 ± 2.7
DistilBART	**6.4 ± 2.2**	**6.8 ± 1.9**	**7.8 ± 2.2**	**6.0 ± 2.5**	**8.6 ± 2.4**
Gold	7.8 ± 1.9	6.8 ± 2.7	7.4 ± 2.4	5.8 ± 0.8	5.4 ± 1.5
Flan-T5-XL	3.0 ± 1.6	3.0 ± 1.2	2.4 ± 1.5	2.8 ± 0.8	3.4 ± 1.8
Random	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0

Table 4.3.: Phase 2 selection distribution on the 100-instance validation set (5-run Mean ± SD) for the four MExGen attribution formats: top 1 and top 2 sentences, top 1 and top 2 phrases from refined sentences. These are compared against the Phase 1 baseline result. The highest DistilBART attribution accuracy is achieved with top 2 phrases (8.6 ± 2.4%), but all formats remain within run-to-run variance of the baseline (6.4 ± 2.2%). The Llama-3 preference never drops below 82.4 ± 4.0%.

The strong preference for Llama-3 summaries persists across all attribution formats, never dropping below 82.4 ± 4.0%. Providing two explanatory features (Top 2) for Target Model attribution consistently outperformed providing a single feature. The highest Target Model accuracy was achieved using the top 2 phrases per refined sentence (8.6 ± 2.4%). However, this is still well below the 20% random guessing baseline and does not represent a significant improvement over the Phase 1 baseline of 6.4 ± 2.2%. The small fluctuations across formats are within the range of run-to-run variance.

4.2.2. Test Set Performance

To verify the stability of these findings, the same experimental variations were evaluated on the 200-instance test set, averaged over three independent runs. Table 4.4 presents the detailed selection distribution. The test set results confirm the persistence of the Llama-3

Candidate Model	Baseline (%)	Sentence-Level (%)		Phrase-Level (%)	
		Top 1	Top 2	Top 1	Top 2
Llama-3	84.5 ± 0.9	83.3 ± 0.8	82.5 ± 2.6	82.0 ± 1.8	82.8 ± 2.4
DistilBART	**6.2 ± 0.6**	**5.8 ± 0.8**	**6.8 ± 0.3**	**4.2 ± 0.8**	**4.5 ± 2.3**
Gold	6.5 ± 1.0	7.3 ± 1.3	6.7 ± 1.6	8.8 ± 2.0	7.7 ± 0.3
Flan-T5-XL	2.8 ± 1.3	3.5 ± 0.9	4.0 ± 0.9	5.0 ± 0.5	5.0 ± 0.5
Random	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0

Table 4.4.: Phase 2 selection distribution on the 200-instance test set (3-run Mean ± SD). Sentence-level attributions remain comparable to the 6.2 ± 0.6% baseline, while phrase-level attributions decrease DistilBART attribution accuracy to 4.2 - 4.5 %, suggesting an introduction of noise. The Llama-3 selection rate never drops below 82.0 ± 1.8%.

bias across all attribution formats. The Target Model attribution accuracy for sentence-level formats remains comparable to the baseline, with the top 2 sentences achieving the highest accuracy (6.8 ± 0.3% vs. baseline 6.2 ± 0.6%). The phrase-level formats performed notably worse, with accuracy dropping to 4.2 ± 0.8% and 4.5 ± 2.3% for the top phrase and top 2 phrases respectively. This suggests that phrase-level attributions may introduce noise rather than present a useful signal for the Judge. However, given the small number of runs ($n = 3$) and the magnitude of the standard deviations, none of these differences can be considered statistically significant. The observed fluctuations are consistent with sampling variance rather than a meaningful effect of the explanations.

4.3. Few-Shot Selection: Similarity vs. Clustering

This section evaluates the impact of the few-shot example selection strategy on attribution accuracy. The experiments in Sections 4.1 and 4.2 use *semantic similarity* retrieval, where the few-shot examples are the nearest neighbors to the test instance in the document embedding space. Figure 4.1 visualizes this space using a 2D Principal Component Analysis (PCA) projection, showing the 100 validation test instances and their retrieved neighbors.

This approach is compared against *K-Means clustering* (k = 3), which retrieves examples from distinct semantic clusters to maximize diversity (Section 3.6.2). Figure 4.2 visualizes the clustered embedding space with the corresponding test inputs and retrieved documents. Table 4.5 presents the comparison of attribution accuracy across both retrieval strategies on the 200-instance test set (3-run Mean ± SD). The clustering strategy achieves a baseline accuracy of 7.0 ± 1.3%, which dropped to 4.8 ± 0.6% when sentence-level attributions were introduced. In contrast, semantic similarity retrieval maintained a more stable performance, shifting from 6.2 ± 0.6% at baseline to 6.8 ± 0.3% with

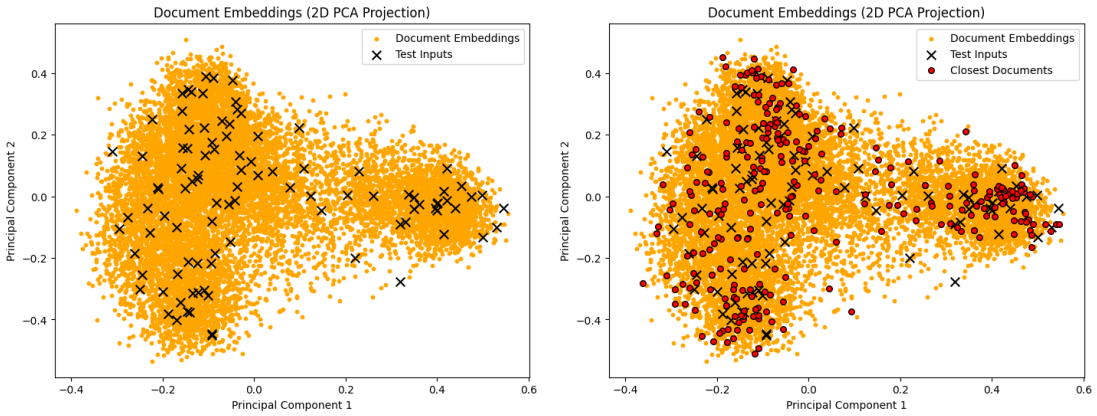


Figure 4.1.: 2D PCA projection of the embedding space for the XSum validation set, illustrating the semantic similarity few-shot retrieval strategy. Black crosses mark the 100 validation test inputs and red dots mark the three nearest-neighbor documents retrieved per test instance. The 2D projection is for visualization only and does not perfectly preserve local pairwise distances. The nearest-neighbor distances are computed in the full 768-dimensional embedding space.

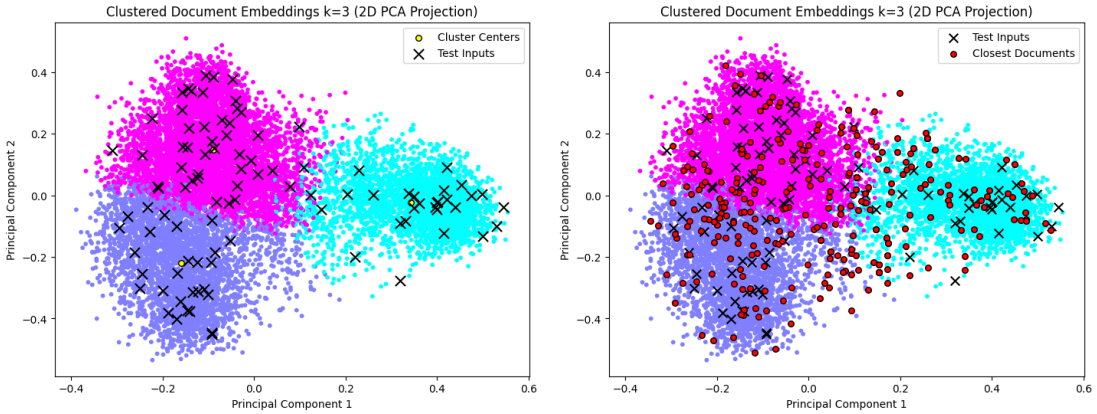


Figure 4.2.: 2D PCA projection of the same embedding space visualizing the K-Means clustering retrieval strategy (k=3). The embedding space is partitioned into three clusters, and for each test input (black crosses) the single closest document from each cluster is retrieved (red dots). Unlike similarity retrieval, the selected documents are spread across the full semantic space.

attributions. Neither strategy yields a meaningful improvement over the Phase 1 baseline. The clustering strategy exhibited the largest regression observed in any experimental variation, and its attribution accuracy (4.8 ± 0.6%) was also notably lower than that of similarity retrieval (6.8 ± 0.3%) for the same explanation format. While the small sample size ($n = 3$) limits definitive conclusions, this pattern suggests that diverse few-shot examples may be less effective than topically similar ones for guiding the Judge's attribution. Based on these results, semantic similarity retrieval was used exclusively for all subsequent experiments.

	Semantic Similarity (%)		**K-Means Clustering ($k = 3$) (%)**	
Candidate Model	Baseline	Top 2 Sent.	Baseline	Top 2 Sent.
Llama-3	84.5 ± 0.9	82.5 ± 2.6	86.0 ± 3.0	84.2 ± 2.0
DistilBART	**6.2 ± 0.6**	**6.8 ± 0.3**	**7.0 ± 1.3**	**4.8 ± 0.6**
Gold	6.5 ± 1.0	6.7 ± 1.6	5.0 ± 1.8	7.0 ± 1.0
Flan-T5-XL	2.8 ± 1.3	4.0 ± 0.9	2.0 ± 1.3	4.0 ± 1.3
Random	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0

Table 4.5.: Comparison of the two dynamic few-shot retrieval strategies on the 200-instance test set (3-run Mean ± SD), presenting Phase 1 baseline and Phase 2 with top 2 sentence attributions. Semantic similarity retrieval is stable across phases, while K-Means clustering regresses from 7.0 ± 1.3% to 4.8 ± 0.6% for DistilBART.

4.4. The Impact of Self-Preference Bias

The evaluation pipeline uses Selene-1-Mini-LLama-3.1-8B, a Llama family model fine-tuned as a judge, to evaluate candidate summaries that include outputs from Llama-3-8B-Instruct. As outlined in Section 3.7.1, the Judge's frequent selection of Llama-3 summaries may be attributable to self-preference or family bias, given that Selene shares the underlying Llama-3 architecture. To isolate this variable, the Llama-3 candidate was completely excluded from the multiple-choice pool, reducing the task to a four-way selection with a random guessing baseline of 25%. Table 4.6 presents the attribution accuracies for Phase 1 (baseline) and Phase 2 (top 2 sentences and top 2 phrases) on the 100-instance validation set, averaged over five independent runs with randomized candidate ordering. The results demonstrate that architectural self-preference is not the sole factor driving the Judge's selections. With the Llama-3 candidate removed, the Judge shifts its preference to the next most fluent option, which are the human-written Gold summaries, selecting them at a rate of up to 41.8 ± 2.4%. The Target Model achieves a baseline accuracy of 32.2 ± 1.6%, which exceeds the 25% random guessing baseline but remains a weak signal. The addition of feature attributions provides only

Candidate Model	Baseline (%)	Sentence-Level (%) Top 2	Phrase-Level (%) Top 2
Gold	41.6 ± 1.9	38.8 ± 2.6	41.8 ± 2.4
DistilBART	**32.2 ± 1.6**	**34.6 ± 3.8**	**33.4 ± 3.1**
Flan-T5-XL	26.0 ± 1.2	26.2 ± 2.3	24.8 ± 2.8
Random	0.2 ± 0.4	0.4 ± 0.5	0.0 ± 0.0

Table 4.6.: Self-preference bias control experiment on the 100-instance validation set (5-run Mean ± SD) with the Llama-3 candidate removed from the pool. The task is reduced to a four-way selection (random baseline 25%). The Judge redirects its preference to the human-written Gold summaries (up to 41.8 ± 2.4% selection rate), rather than to DistilBART. The highest achieved attribution accuracy is 34.6 ± 3.8% for sentence-level explanations.

marginal improvements, with the highest accuracy reaching 34.6 ± 3.8% for sentence-level explanations. The Judge's shift toward Gold rather than toward DistilBART suggests that the Judge's behavior is driven primarily by quality bias rather than architectural familiarity. The implications of this observation, alongside a qualitative analysis of the Judge's generated reasoning, are discussed further in Chapter 5.

4.5. Contrastive Prompting

As established in the previous sections, the Judge overwhelmingly favors the Llama-3 candidate regardless of the provided feature attributions, exhibiting a persistent quality bias. To directly counteract this behavior, a contrastive prompting strategy was evaluated. As detailed in Section 3.7.2, the Phase 2 prompt was adapted to present the Llama-3 summary as a negative example, with the Judge explicitly instructed that this summary was not produced by the Target Model and does not match its behavior. This contrastive element was included in every few-shot example following the Target Model summary. The strategy was tested on the 100-instance validation set for Phase 1 and Phase 2 (top 2 sentences and top 2 phrases), averaged over five runs.

Table 4.7 shows that the contrastive strategy produces a measurable shift in the Judge's behavior. The Llama-3 selection rate drops from the standard baseline of 82.8 ± 2.6 (Table 4.1) to 73.8 ± 3.9%, while the Target Model accuracy increases from 6.4 ± 2.2% to 11.4 ± 2.1%. This represents the largest improvement observed across all experimental variations, but it remains well below the 20% random guessing baseline. Adding feature attributions to the contrastive prompt did not yield further improvements. When the top 2 sentences or top 2 phrases were included, the Target Model's accuracy decreased to 9.2 ± 1.1% and 8.0 ± 2.4%, respectively. The Llama-3 selection rate increased by nearly 5%. Notably, the contrastive baseline (11.4 ± 2.1%) and the sentence-level result (9.2 ±

Candidate Model	Baseline (%)	Sentence-Level (%) Top 2	Phrase-Level (%) Top 2
Llama-3	73.8 ± 3.9	77.4 ± 2.7	78.2 ± 2.7
DistilBART	**11.4 ± 2.1**	**9.2 ± 1.1**	**8.0 ± 2.4**
Gold	10.2 ± 4.1	8.6 ± 1.5	7.6 ± 2.8
Flan-T5-XL	4.6 ± 1.8	4.8 ± 1.3	6.2 ± 1.3
Random	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0

Table 4.7.: Contrastive prompting resultes on the 100-instance validation set (5-run Mean ± SD), in which the Llama-3 output is presented as an explicit negative example in every few-shot instance. The contrastive baseline increases DistilBART attribution accuracy from 6.4 ± 2.2% (Table 4.1) to 11.4 ± 2.1%, the largest improvement observed so far. Adding MExGen attributions reduces the accuracy down to 9.2 ± 1.1% at sentence-level and 8.0 ± 2.4% at phrase-level.

1.1%) show minimal overlap in their standard deviations, suggesting that the addition of attributions may have actively hindered the Judge's performance. The source of this regression, possibly competing signals or cognitive overload from combining negative examples with attribution features, requires further investigation.

4.6. Pairwise Control Experiment

The goal of this control experiment was to observe whether simplifying the task and adapting the prompt format to fit the provided Selene Mini prompt templates would have a positive impact on the Judge's attribution capability. As detailed in Section 3.7.3, the task was simplified to a tournament-style head-to-head comparison using the 200-document test set with similarity-based few-shot prompts. Each model was evaluated against every other model in direct pairwise matchups, excluding the Random candidate. This resulted in three matchups per document for each model, totaling 600 trials per model in each phase. Table 4.8 presents the per-matchup breakdown, and Figure 4.3 summarizes the aggregate win rates across all opponents. Llama-3 dominated the tournament, winning approximately 90% of its matchups in both Phase 1 (91.8%) and Phase 2 (89.5%). In a perfectly unbiased evaluation where all candidates were equally capable of matching the target style, win rates would be closer to 50%. The Target Model's win rate increased slightly from 37.3% in Phase 1 to 40.2% in Phase 2, but failed to cross the 50% threshold. The per-matchup breakdown in Table 4.8 reveals that DistilBART is competitive against Gold and outperforms Flan. However, it is overwhelmingly rejected in favor of Llama-3, confirming that the Judge's bias is specifically triggered by the high fluency of the Llama-3 outputs rather than a general inability to recognize the Target Model. This pattern is consistent across all candidates. Gold and Flan similarly lose to Llama-3 by wide margins

while remaining competitive against each other. These results confirm that the 5-way multiple-choice format is not the root cause of the Judge's poor attribution accuracy.

Matchup	**Phase 1**	**Phase 2**
DistilBART vs. Llama	17 / 183	24 / 176
DistilBART vs. Gold	99 / 101	100 / 100
DistilBART vs. Flan	108 / 92	117 / 83
Gold vs. Llama	18 / 182	22 / 178
Gold vs. Flan	120 / 80	122 / 78
Flan vs. Llama	14 / 186	17 / 183

Table 4.8.: Head-to-head results from the pairwise tournament control experiment on the 200-instance test set, reported as wins / losses per matchup. Phase 2 includes top sentence attributions. DistilBART is competitive against Gold and beats Flan, but is overwhelmingly rejected against Llama-3 (17/183 in Phase 1, 24/176 in Phase 2).

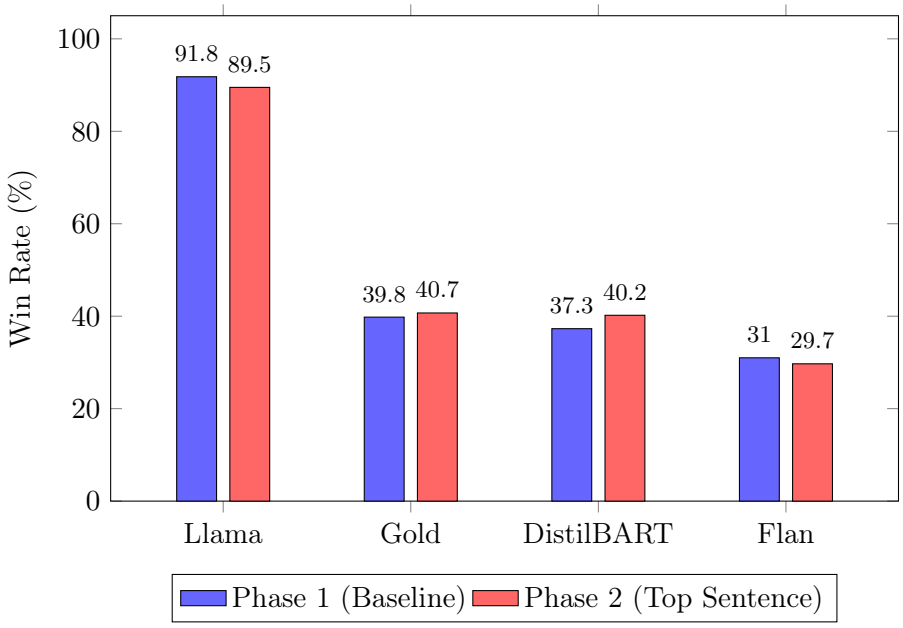


Figure 4.3.: Aggregate win rates from the pairwise tournament (600 trials per model, 200 documents x 3 matchups), comparing Phase 1 (baseline) and Phase 2 (top sentence attributions). Llama-3 wins approximately 90% of its matchups in both phases, while DistilBART improves only marginally from 37.3% to 40.2%.

5. Discussion

This chapter provides a detailed analysis and interpretation of the empirical findings presented in Chapter 4. The main objective is to contextualize the persistent failure of feature attributions and advanced prompting strategies to overcome the Judge's inherent biases. A secondary goal is to identify the implications for future work on explanation evaluation using LLM-as-a-Judge frameworks. We begin by analyzing the dominance of quality bias and the Judge's tendency toward post-hoc rationalization in Section 5.1. This is followed by addressing why input-level feature attributions themselves failed to provide a discriminative signal in Section 5.2. Section 5.3 explores how the Target Model's inconsistent behavior and the properties of the XSum dataset negatively influence the attribution task. This chapter concludes with Section 5.4, which outlines the limitations of our experimental design and methodology.

5.1. Quality Bias and Post-Hoc Rationalization

Why did the Judge overwhelmingly default to the most fluent summaries, and how does it rationalize this preference when explicitly instructed otherwise? The empirical results demonstrate that the Judge model is fundamentally not optimized for model attribution. Instead of acting as an objective analyzer, Selene Mini shows the characteristics of a standard reward model. It consistently selects the most detailed and verbose summary in the candidate pool, regardless of whether that summary was produced by the Target Model. The behavior persisted across every experimental variation tested in Chapter 4. Self-preference or family bias cannot be entirely excluded, but the results of the Llama-3 exclusion experiment (Section 4.4) demonstrate that shared architectural family is not the main driver of the Judge's behavior. When the Llama-3 candidate was removed, attribution accuracy did not meaningfully improve. Instead, the Judge redirected its preference to the next most fluent option, the human-written Gold summaries. This behavior is characteristic of quality bias, where the model conflates the "best" output with the correct one and ignores the constraints provided in the prompt. This quality bias is reinforced by the model's inherent sycophancy (see Section 2.2.2). Having been fine-tuned via human feedback to prefer helpful, accurate, and detailed responses, the Judge resists selecting a lower-quality summary even when the few-shot examples explicitly demonstrate the Target Model's behavior. The model's alignment training interferes with its ability to follow the task instructions. The pairwise control experiment (Section 4.6) provides further evidence for this interpretation.

Post-Hoc Rationalization To resolve the conflict between its quality bias and the task instructions, the Judge resorts to post-hoc rationalization. It arrives at a decision based on its preference for the high-quality Llama-3 summary and then generates unfaithful reasoning to justify that predetermined choice.

Qualitative Error Analysis This phenomenon is illustrated in Figure 5.1, which presents the Judge's generated reasoning for an evaluation instance from the experiment with included MExGen attributions (top 2 most influential sentences). The article from the XSum validation set (article no. 2) addresses a photo shoot with British royalty for a magazine. The example exposes two specific failures in the Judge's reasoning process. First, the Judge fails to identify the unique stylistic patterns of the Target Model despite being provided with few-shot examples and input sentence attributions. Instead, it generates a generic description of the target behavior (e.g., summaries that are "concise", "straightforward", and "focus on the main points"). This description is so broad that it could accurately describe any baseline summarization model in the candidate pool, including Flan, Gold, and DistilBART. Second, the Judge actively contradicts its own stated criteria. In its reasoning, it claims to be looking for a candidate that "lacks embellishments" and states that its chosen candidate "avoids unnecessary details". However, it ultimately selects the Llama-3 summary, which is objectively the longest and most detailed option in the pool. We could argue that this contradiction stems from different definitions of "unnecessary details" between LLMs and humans, but the few-shot examples explicitly showed what a concise Target Model summary looks like. This example demonstrates that the Judge generates plausible-sounding reasoning to justify selecting the most verbose summary. This results in the most embellished candidate being praised for lacking embellishments.

5.2. Failure of Input Feature Attributions

Why did the inclusion of input-level feature attributions fail to isolate the Target Model for the LLM Judge?
Section 5.1 explains the Judge's inherent bias toward fluent and detailed text, but it does not fully account for why the Phase 2 MExGen attributions failed to correct the bias, and in some cases actively degraded performance. The empirical data suggests that input-level attributions function as distracting noise rather than a clarifying signal in this experimental setup. Two properties of the summarization task contribute to this failure:

5.2.1. The Lead Bias Problem: Sentence-Level Attributions

In standard journalistic writing, articles follow an inverted pyramid structure in which the most important information appears at the beginning (Pöttker, 2003). In the news domain, "sentence position bias dominates the learning signal for summarization" (Kedzie et al., 2018), a phenomenon known as *lead bias*. Because all competent summarization models attend to these same introductory sentences, input-based attributions provide

Candidate Summaries (Input):

Response A (Flan): The Duchess of Cambridge has appeared on the cover of British Vogue in a series of portraits.

Response B (DistilBART): The Duchess of Cambridge is to feature on the cover of British Vogue for the first time.

Response C (Gold): The Duchess of Cambridge will feature on the cover of British Vogue to mark the magazine's centenary.

Response D (Llama): The Duchess of Cambridge, patron of the National Portrait Gallery since 2012, has sat for a photo shoot with photographer Josh Olins, resulting in seven photographs appearing in the June edition of British Vogue, which also marks the magazine's 100th anniversary.

Response E (Random): The UK has a "moral" duty to guarantee the status of EU nationals living in the country ahead of negotiations over its exit, a group of peers has said.

Judge's Generation (Output):
Reasoning: Step 1: Infer the common style or pattern from the three examples. The summaries are *concise* and *focus on the main points* of the articles. They often highlight key phrases or sentences that capture the essence of the original content. The summaries also tend to be *straightforward and lack embellishments.*
Step 2: Identify which candidate matches that inferred pattern. Candidate D aligns with the observed pattern by providing a clear and concise summary that captures the main points of the article, including the Duchess's role as patron and the significance of the photo shoot. It avoids unnecessary details and maintains a straightforward tone.
Result: D (Llama)

Figure 5.1.: Qualitative error analysis of XSum validation article no.2 (British Vogue photo shoot), with top 2 sentence attributions included in the prompt. This evaluation instance demonstrates the Judge's reasoning and post-hoc rationalization. The Judge correctly identifies how the Target Model (DistilBART) writes summaries (*"concise", "straightforward", "lack embellishments"*) in Step 1, but falsely attributes these characteristics to the Llama-3 candidate (Response D) in Step 2.

no discriminative power for distinguishing between models. Figure 5.2 illustrates this problem using an instance from the XSum test set (article no. 34 with similarity-based prompt examples). The MExGen attributions identify the first two sentences as the most influential, capturing the core facts of the story. However, every candidate summary incorporates information from at least one of these sentences. The DistilBART, Gold, and Flan summaries abstract the specific details (e.g., generalizing "Lara Clarke" to "a woman"), while the Llama-3 summary includes nearly every specific detail mentioned in the attributions. Because all candidates use information from the same lead sentences, the attributions provide no stylistic differentiation. The Judge defaults to its quality bias and selects the Llama-3 output.

5.2.2. The Content Overlap Problem: Phrase-Level Attributions

The phrase-level attributions introduce an additional complication. Figure 5.3 presents an example from the XSum validation set (article no. 39 with similarity-based prompt examples) concerning a police raid. The Judge is provided with the top two most influential phrases extracted by MExGen, which correctly highlight important entities from the article. However, this phrase extraction unintentionally penalizes the Target Model. The DistilBART summary abstracts the input, paraphrasing "the boys, aged 13,15, and 16" into "three teenage boys" and generalizing the specific drug charge to a "drugs raid". In contrast, the Llama-3 summary reproduces the extracted phrases nearly verbatim. Since the prompt explicitly states that the attributions represent the phrases the Target Model prioritized, the Judge is actively directed toward the wrong answer. It learns that the target behavior involves copying important input phrases rather than paraphrasing them. Despite being provided with few-shot examples that demonstrate DistilBART's highly abstractive style, the Judge fails to internalize this pattern. Instead, by directing attention to specific source content, the attributions incentivize the Judge to select the model that incorporated that content most thoroughly rather than the model that exhibited the target's characteristic abstractive style.

The Contrastive Prompting Conflict This content overlap phenomenon also explains the unexpected decline in attribution accuracy when feature attributions were combined with contrastive prompting (Section 4.5). The contrastive baseline achieved the highest Target Model accuracy observed in any experimental variation ($11.4 \pm 2.1\%$), showing that the negative example strategy was partially effective at suppressing the Llama-3 preference. However, when sentence-level or phrase-level attributions were added, accuracy dropped to $9.2 \pm 1.1\%$ and $8.0 \pm 2.4\%$ respectively. This regression is consistent with the content overlap mechanism described above. The contrastive prompt instructed the Judge to avoid the Llama-3 style, but the MExGen attributions pointed directly to it content-wise. These two signals created a contradiction within the prompt, reducing some of the improvements obtained from contrastive prompting.

Truncated Source Article: Lara Clarke, from Walsall, won the award at the Cake International contest with her 5ft 10in (1.7m) creation, inspired by the Hunger Games star. Ms Clarke previously triumphed at the contest with a life-sized Johnny Depp cake, which made headlines around the world. She said she found her second consecutive success "hugely exciting". Ms Clarke also won silver in the large decorative exhibit section at the competition, with a cake based on her favourite Game of Thrones character Tyrion Lannister. She spent two-and-a-half months working on the cakes ahead of the competition at Birmingham's National Exhibition Centre, The Jennifer Lawrence sponge cake required 150 eggs, 10kg (22lb) flour and 10kg butter. [...]

Most Influential Sentences (MExGen Attributions):

- Lara Clarke, from Walsall, won the award at the Cake International contest with her 5ft 10in (1.7m) creation, inspired by the Hunger Games star.
- Ms Clarke previously triumphed at the contest with a life-sized Johnny Depp cake, which made headlines around the world.

Candidate Summaries:

Target Summary (DistilBART): A woman who baked a life-sized Jennifer Lawrence cake has won the top prize at a global baking competition.

Non-Target Summary (Llama-3): Lara Clarke won the Cake International contest with a 5ft 10in cake inspired by Jennifer Lawrence's character from the Hunger Games, after previously winning with a life-sized Johnny Depp cake.

Non-Target Summary (Gold): A woman has won a gold award at national baking competition by entering a life-sized Jennifer Lawrence cake.

Non-Target Summary (Flan): A woman has won a gold award at a national baking competition for a life-sized Jennifer Lawrence cake.

Figure 5.2.: Analysis of XSum test article no.34 (Cake International contest), with top 2 sentence MExGen attributions, illustrating lead bias. Every candidate summary takes information from the same first two sentences of the article. DistilBART, Gold, and Flan abstract the details (e.g., using "a woman"), while Llama-3 reproduces them almost verbatim.

Full Source Article: **The boys, aged 13, 15, and 16**, are among those charged with **possession of a Class A drug with intent to supply**. Fifteen people have been remanded in custody and are due to appear in court on Thursday. Gwent Police raided 35 addresses in the city and made 25 arrests. Three people have been released with no charge and two are awaiting questioning.

Most Influential Phrases (MExGen Attributions):

- **possession of a Class A drug with intent to supply**
- **The boys, aged 13, 15, and 16**

Candidate Summaries:

Target Summary (DistilBART): **Three teenage boys** have been **charged in connection with a drugs raid** in Newport.

Non-Target Summary (Llama): Fifteen people, including **three boys aged 13, 15, and 16**, have been charged with **possession of a Class A drug with intent to supply** after a police raid in Gwent, with 15 of them remanded in custody and due to appear in court on Thursday.

Figure 5.3.: Analysis of XSum validation article no.39 (Police drug raid), with top 2 phrase MExGen attributions, illustrating content overlap. The extracted phrases appear almost verbatim in the Llama-3 summary but are abstracted by DistilBART.

Shifting from Content to Style The failure of both feature highlighting and contrastive prompting reveals a fundamental limitation of using input-based feature attributions for model attribution in summarization tasks. When all candidate models summarize the same source text, particularly news articles, identifying which input content the model attended to provides insufficient discriminative power. Any competent summarization model will attend to the same core facts, so content-based explanations offer no basis for distinguishing between models. Lacking stylistic cues, the Judge defaults to selecting the candidate that reproduces the highlighted content most thoroughly, which is always the most fluent and verbose model. This analysis suggests that effective explanations for model attribution in generative tasks must shift from content-level attribution ("what the model looked at") to style-level characterization ("how the model transforms its input"). This could include high-level, human-readable descriptions of the Target Model's generative patterns rather than input-level feature scores. Whether such stylistic explanations can overcome the Judge's quality bias remains an open question for future research.

5.3. Target Model Inconsistency

How do the unpredictable behaviors of the Target Model and the properties of the underlying dataset impact the feasibility of model attribution?
Beyond the Judge's biases and the limitations of the explanations, the Target Model itself contributed to the difficulty of the attribution task. Successful model attribution from a small number of examples requires the target to exhibit a stable, predictable style. However, DistilBART is fundamentally inconsistent. While it typically produces heavily compressed and simplified summaries, it frequently hallucinates, injecting facts and entities that never appeared in the source article. The few-shot prompt examples cannot adequately capture this erratic alternation between abstraction and hallucination. Because the Judge retains no memory across inference calls, the few-shot examples in each prompt represent its entire understanding of the Target Model. If those examples happen to show correct and concise summaries, the Judge has no basis for predicting that DistilBART may hallucinate on the test instance. Furthermore, even if the Judge did recognize a hallucinated summary, its alignment training would likely prevent it from selecting it. Modern LLM evaluators are trained to penalize unfaithful or factually incorrect text (Section 2.2). In this attribution task, hallucination is the behavioral feature the Judge should identify, which is a requirement that directly conflicts with its training objectives. When the Judge is faced with this conflict, it is likely to discard the hallucinated summary. Figure 5.4 illustrates this failure with an evaluation instance from the XSum test set (article no. 3), where the prompt included the top two most influential sentences as attributions. The DistilBART summary contains hallucinated information not present in the source text, so the Judge identifies the Llama-3 output as the most factually accurate representation of the input and selects it.

The XSum Ground Truth Problem DistilBART's hallucination tendency is not accidental, it is a learned behavior inherited from the training data. As discussed in Section 3.2, the XSum dataset's human-written Gold summaries frequently contain external background knowledge not present in the source documents. DistilBART was fine-tuned on this exact dataset, so it learned to mimic this behavior. While human journalists include factually correct background knowledge, the model lacks the world knowledge to do so accurately, and its additions frequently manifest as hallucinations. This creates a problem for the evaluation pipeline. The Gold standard includes external information, the Target Model copies this behavior and hallucinates, and the Judge is aligned to penalize precisely this kind of unfaithful text.

5.4. Limitations

While the empirical results of this study demonstrate the failure of input feature attributions and expose the inherent biases of the LLM Judge, it is important to acknowledge the boundaries of the experimental setup.

Full Source Article: The 48-year-old former Arsenal goalkeeper played for the Royals for four years.He was appointed youth academy director in 2000 and has been director of football since 2003.A West Brom statement said: "He played a key role in the Championship club twice winning promotion to the Premier League in 2006 and 2012."

Candidate Summaries:

Response A (DistilBART): West Brom manager Tony Mowbray has signed a new two-year contract with the club.

Response B (Flan): West Brom have sacked director of football Steve Bruce after he was found guilty of misconduct in a football match.

Response C (Gold): West Brom have appointed Nicky Hammond as technical director, ending his 20-year association with Reading.

Response D (Random): Sam Allardyce has resigned as Crystal Palace manager five months after he joined the Premier League club.

Response E (Llama): The 48-year-old former Arsenal goalkeeper, who played for Reading for four years, has been the club's director of football since 2003 and played a key role in the team's two promotions to the Premier League in 2006 and 2012.

Judge's Generation (Output):
Reasoning: Step 1: Analyze the Target Model's behavior in the provided examples. The summaries typically focus on the key information about the player or manager, such as their name, position, and significant achievements. They often omit less relevant details and maintain a concise structure.
Step 2: Identify which candidate matches this inferred pattern. Candidate E directly addresses the key points about the former Arsenal goalkeeper, including his role at Reading and his contributions to West Brom's promotions. It maintains a concise structure, similar to the Target Model's summaries.
Result: E (Llama)

Figure 5.4.: Analysis of XSum test article no.3 (West Brom director appointment), with top 2 sentence attributions, demonstrating hallucination-related attribution failure. The DistilBART summary (Response A) hallucinates information absent from the source, which is a defining characteristic of the Target Model summaries. The Judge selects the faithful and detailed Llama-3 summary instead.

Scale and Computational Constraints The evaluation was conducted on relatively small subsets (100 validation and 200 test instances), and results were averaged over 3–5 independent runs. While these sample sizes were sufficient to identify consistent behavioral patterns, they limit statistical power, as reflected in the standard deviations reported throughout Chapter 4. Since the initial experiments showed no positive signal from the addition of attributions, investing the computational resources required for large-scale runs (e.g., 500+ instances) was not justified without a preliminary indication of effectiveness.

Judge Model Scale and Capabilities The experiments were limited to a single Judge model with 8 billion parameters, loaded in 4-bit quantization. 8B-parameter models represent a capable open-source baseline and the use of consistent quantization across both phases preserves the validity of relative comparisons. However, larger models may possess stronger reasoning capabilities that enable better instruction following in complex evaluation prompts. It is possible that a significantly larger Judge would be more responsive to the feature attributions and advanced prompting strategies tested here.

Dataset and Target Model The choice of XSum as the evaluation dataset and DistilBART as the Target Model acted like a stress test for the evaluation pipeline. XSum's Gold summaries contain external knowledge, and DistilBART exhibits inconsistent behavior alternating between heavy abstraction and hallucination. While this design revealed important failure modes, it also means that the negative results may be partially due to the difficulty of the specific experimental setup rather than a fundamental limitation of the approach. A more consistent Target Model paired with a more reliable dataset might yield different results.

Task Scope This thesis focuses exclusively on a single generative NLP task which is news summarization. News summarization inherently suffers from lead bias and content overlap between candidate models, which diminishes the discriminative power of input-level attributions. This limitation may be task-specific. Applying the same attribution pipeline to a generative task with a less rigid input structure, like open-ended question answering or creative text generation, might bring different results. In this case, input attributions could carry more distinctive information about model-specific behavior.

6. Conclusion and Future Work

6.1. Summary of Findings

This thesis investigated whether input-level feature attributions can improve an LLM Judge's ability to identify text generated by a specific Target Model, a task we framed as an attribution-based adaptation of simulatability. The goal was to determine whether LLM-as-a-Judge frameworks offer a scalable alternative to human evaluation for assessing explanation faithfulness. Across extensive experiments using multiple explanation formats, retrieval strategies, prompting variations, and control conditions, the results were consistently negative. Notably, the experimental design does not require the Judge to achieve high absolute accuracy in Phase 1. What matters is the *relative change* between Phase 1 (without explanations) and Phase 2 (with explanations). A substantial improvement in Phase 2 would provide evidence that the explanations carry faithful information about the Target Model's reasoning. The absence of such improvement indicates that the current combination of feature-based explanations and LLM Judge is insufficient for this evaluation setup. The three central research questions posed at the beginning of this thesis can now be answered:

1. Baseline Attribution The LLM Judge failed the baseline attribution task. Rather than learning the Target Model's behavioral patterns from the few-shot examples, the Judge overwhelmingly selected the most fluent candidate, the Llama-3 model (84.5 $\pm$ 0.9% on the test set). The DistilBART baseline attribution accuracy of 6.2 $\pm$ 0.6% on the test set fell below the 20% random guessing threshold, which indicates that the Judge was not guessing but systematically selecting other candidates. This behavior is driven by an inherent quality bias, reinforced by the model's sycophantic alignment. The Judge engaged in post-hoc rationalization, generating plausible-sounding reasoning to justify a predetermined choice.

2. The Impact of Feature-Based Explanations Input-level feature attributions failed to improve the Judge's attribution accuracy across both tested granularities. Sentence-level attributions failed because of lead bias. Since all candidate models attend to the same introductory sentences in news articles, highlighting these sentences provided no discriminative power. Phrase-level attributions suffered from a content overlap problem. The extracted phrases matched the Llama-3 summary's verbatim reproduction more closely than DistilBART's abstractive paraphrases. This actively directed the Judge toward the wrong answer. The best-performing explanation format (top 2 sentences) achieved only 6.8 $\pm$ 0.3% on the test set, with overlapping standard deviations relative to

the baseline (6.2 ± 0.6%). This confirms that the improvement was not significant.

3. Overriding Bias Advanced prompting strategies achieved partial but insufficient results. Contrastive prompting produced the largest improvement observed in any experimental variation, raising the Target Model validation baseline accuracy from 6.4 ± 2.2% to 11.4 ± 2.1%. However, this remains well below the 20% random guessing baseline. Combining contrastive prompting with feature attributions caused a regression, revealing that the negative constraint directed the Judge away from the Llama-3 style, while the content-based attributions pointed back toward it. Removing the Llama-3 candidate confirmed that the bias is not exclusively architectural but reflects a deeper quality preference. The pairwise control experiment further demonstrated that task format was not the root cause. DistilBART was competitive against Gold and Flan in head-to-head matchups but was overwhelmingly rejected in favor of Llama-3 (17 to 183), regardless of whether attributions were provided.

6.2. Contributions

Despite negative empirical results, this thesis makes several contributions to the fields of explainable AI and LLM-based evaluation.

Reproducible Evaluation Framework The two-phase evaluation design provides a rigorous, scalable method for evaluating explanation faithfulness without human annotators. It can be applied to other tasks, models, and explanation types.

Systematic Evidence of LLM Judge Limitations The experiments document that current small-scale LLM Judges cannot reliably perform model attribution due to quality bias, sycophantic alignment, and post-hoc rationalization. These findings add to the evidence on the limitations of LLM-as-a-Judge frameworks (e.g., Ye et al. (2024), Sharma et al. (2025)).

Analysis of Feature Attribution Failure in Summarization Input-level feature attributions, specifically C-LIME via MExGen, fail to provide discriminative power in a news summarization task. This is mainly due to two specific mechanisms: lead bias and content overlap.

Content-to-Style Insight The central conclusion, that effective explanations for model attribution in generative tasks must characterize *how* a model transforms its input rather than *what* input it attended to, provides a concrete direction for future explanation design.

6.3. Implications and Future Work

Input-level feature attributions generated by methods such as SHAP and LIME (Lundberg and Lee (2017), Ribeiro et al. (2016)) represent a local, mathematically grounded perspective on model behavior. This thesis demonstrates that mainly identifying which input features influenced a specific output is an incomplete perspective. Especially in a summarization task where multiple models attend to the same core content, knowing what the model looked at does not reveal how it processed that information. A human expert asked to distinguish between model outputs would most likely describe behavioral patterns instead of pointing to sentences or phrases. Qualitative, high-level characterizations such as "this model copies phrases verbatim" or "this model tends to hallucinate" represent a type of explanation that current algorithmic methods in explainability do not produce. Existing methods focus on local feature attribution for individual predictions, not on global behavioral characterization across instances. The question of *"How does this model systematically differ from other models?"* requires explanations that capture recurring patterns across multiple instances. A framework that accommodates both feature attributions and behavioral characterizations, as well as methods for comparing their utility, is an important direction for the field. A separate challenge is presented in the evaluation model itself. Even if explanations are faithful and informative, the LLM Judge may be unable to use them effectively due to internal biases. This thesis documented post-hoc rationalization, where the Judge generates unfaithful reasoning to justify a predetermined choice. This is consistent with recent findings (Turpin et al. (2023), Lanham et al. (2023)). The Judge's alignment training, designed to reward helpful and faithful text, directly conflicts with the task of recognizing a model whose defining characteristic is producing unfaithful text. This raises the question of how well RLHF-trained models can be used as evaluation tools for tasks that require rewarding behaviors the training explicitly penalized.

Style-Based Features for Model Attribution The most direct extension of this work is to replace input-level feature attributions with style-based explanations that describe the Target Model's behavior in terms of writing style rather than pointing to specific input tokens. This thesis suggests that the Judge needs high-level, human-readable characterizations of how models differ. A concrete experimental design would involve annotating each candidate summary with a short behavioral description of its model (e.g., "tends to hallucinate details", "copies input phrases verbatim", "uses external background knowledge"), requiring the Judge to match these descriptions against the patterns observed in the few-shot examples. Recent work on concept-based explanations (Sun et al., 2025b) suggests a potential direction for generating such characterizations algorithmically.

Task Generalization The failure of feature attributions in this thesis may be partially task-specific. Applying the same two-phase evaluation framework to generative tasks with greater stylistic diversity, such as creative writing or open-ended question answering

could yield different results. In such tasks, input-level feature attributions would likely highlight more diverse input regions across different models.

Judge Model Scaling and Architecture This thesis used a single 8-billion parameter 4-bit quantized Judge model. Larger models with stronger reasoning capabilities may be more responsive to complex prompt instructions and better equipped to override quality biases. Future work should evaluate whether scaling the Judge model changes the relationship between explanations and attribution accuracy.

Bibliography

AI@Meta (2024). Llama 3 model card.

Alexandru, A., Calvi, A., Broomfield, H., Golden, J., Dai, K., Leys, M., Burger, M., Bartolo, M., Engeler, R., Pisupati, S., Drane, T., and Park, Y. S. (2025). Atla selene mini: A general purpose evaluation model.

Bansal, G., Wu, T., Zhou, J., Fok, R., Nushi, B., Kamar, E., Ribeiro, M. T., and Weld, D. (2021). Does the whole exceed its parts? the effect of ai explanations on complementary team performance. In *Proceedings of the 2021 CHI conference on human factors in computing systems*, pages 1–16.

Camburu, O.-M., Rocktäschel, T., Lukasiewicz, T., and Blunsom, P. (2018). e-snli: Natural language inference with natural language explanations. *Advances in Neural Information Processing Systems*, 31.

Chen, W.-L., Wei, Z., Zhu, X., Feng, S., and Meng, Y. (2025). Do llm evaluators prefer themselves for a reason? *arXiv preprint arXiv:2504.03846*.

Chen, Y., Zhong, R., Ri, N., Zhao, C., He, H., Steinhardt, J., Yu, Z., and McKeown, K. (2023). Do models explain themselves? counterfactual simulatability of natural language explanations.

Chia, Y. K., Chen, G., Tuan, L. A., Poria, S., and Bing, L. (2023). Contrastive chain-of-thought prompting. *arXiv preprint arXiv:2311.09277*.

Christiano, P., Leike, J., Brown, T. B., Martic, M., Legg, S., and Amodei, D. (2023). Deep reinforcement learning from human preferences.

Dettmers, T., Lewis, M., Belkada, Y., and Zettlemoyer, L. (2022). Llm.int8(): 8-bit matrix multiplication for transformers at scale. *arXiv preprint arXiv:2208.07339*.

Gu, J., Jiang, X., Shi, Z., Tan, H., Zhai, X., Xu, C., Li, W., Shen, Y., Ma, S., Liu, H., et al. (2024). A survey on llm-as-a-judge. *The Innovation*.

Hase, P. and Bansal, M. (2020). Evaluating explainable ai: Which algorithmic explanations help users predict model behavior?

Jacovi, A. and Goldberg, Y. (2020). Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 4198–4205.

Kaushik, D. and Lipton, Z. C. (2018). How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 5010–5015.

Kedzie, C., McKeown, K., and Daumé III, H. (2018). Content selection in deep learning models of summarization. In Riloff, E., Chiang, D., Hockenmaier, J., and Tsujii, J., editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1818–1828, Brussels, Belgium. Association for Computational Linguistics.

Kunkel, J., Donkers, T., Michael, L., Barbu, C.-M., and Ziegler, J. (2019). Let me explain: Impact of personal and impersonal explanations on trust in recommender systems. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–12.

Lanham, T., Chen, A., Radhakrishnan, A., Steiner, B., Denison, C., Hernandez, D., Li, D., Durmus, E., Hubinger, E., Kernion, J., et al. (2023). Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*.

Lipton, Z. C. (2018). The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57.

Liu, R., Sun, Y., Zhang, M., Bai, H., Yu, X., Yu, T., Yuan, C., and Hou, L. (2025). Quantization hurts reasoning? an empirical study on quantized reasoning models.

Lundberg, S. and Lee, S.-I. (2017). A unified approach to interpreting model predictions.

Lyu, Q., Apidianaki, M., and Callison-Burch, C. (2024). Towards faithful model explanation in nlp: A survey. *Computational Linguistics*, 50(2):657–723.

Maynez, J., Narayan, S., Bohnet, B., and McDonald, R. (2020). On faithfulness and factuality in abstractive summarization.

Narayan, S., Cohen, S. B., and Lapata, M. (2018). Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. (2022). Training language models to follow instructions with human feedback.

Paes, L. M., Wei, D., Do, H. J., Strobelt, H., Luss, R., Dhurandhar, A., Nagireddy, M., Ramamurthy, K. N., Sattigeri, P., Geyer, W., and Ghosh, S. (2025). Multi-level explanations for generative language models.

Pagnoni, A., Balachandran, V., and Tsvetkov, Y. (2021). Understanding factuality in abstractive summarization with frank: A benchmark for factuality metrics.

Pang, R. Y., Yuan, W., Cho, K., He, H., Sukhbaatar, S., and Weston, J. (2024). Iterative reasoning preference optimization. *Advances in Neural Information Processing Systems*, 37:116617–116637.

Pöttker, H. (2003). News and its communicative quality: the inverted pyramid—when and why did it appear? *Journalism Studies*, 4(4):501–511.

Pruthi, D., Bansal, R., Dhingra, B., Soares, L. B., Collins, M., Lipton, Z. C., Neubig, G., and Cohen, W. W. (2022). Evaluating explanations: How much do explanations from the teacher aid students? *Transactions of the Association for Computational Linguistics*, 10:359–375.

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. (2024). Direct preference optimization: Your language model is secretly a reward model.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "why should i trust you?": Explaining the predictions of any classifier.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Sharma, M., Tong, M., Korbak, T., Duvenaud, D., Askell, A., Bowman, S. R., Cheng, N., Durmus, E., Hatfield-Dodds, Z., Johnston, S. R., et al. (2023). Towards understanding sycophancy in language models. *arXiv preprint arXiv:2310.13548*.

Sharma, M., Tong, M., Korbak, T., Duvenaud, D., Askell, A., Bowman, S. R., Cheng, N., Durmus, E., Hatfield-Dodds, Z., Johnston, S. R., Kravec, S., Maxwell, T., McCandlish, S., Ndousse, K., Rausch, O., Schiefer, N., Yan, D., Zhang, M., and Perez, E. (2025). Towards understanding sycophancy in language models.

Shi, L., Ma, C., Liang, W., Diao, X., Ma, W., and Vosoughi, S. (2025). Judging the judges: A systematic study of position bias in llm-as-a-judge. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 292–314.

Spiliopoulou, E., Fogliato, R., Burnsky, H., Soliman, T., Ma, J., Horwood, G., and Ballesteros, M. (2025). Play favorites: A statistical method to measure self-bias in llm-as-a-judge. *arXiv preprint arXiv:2508.06709*.

Sun, J., Atanasova, P., and Augenstein, I. (2025a). Evaluating input feature explanations through a unified diagnostic evaluation framework. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10559–10577.

Sun, Y., Wang, D., Sheng, Q., Cao, J., and Li, J. (2025b). Enhancing the comprehensibility of text explanations via unsupervised concept discovery.

Turpin, M., Michael, J., Perez, E., and Bowman, S. R. (2023). Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting.

Wang, B., Li, Y., Zhou, J., and Chen, F. (2025). Can llm assist in the evaluation of the quality of machine learning explanations?

Wang, P., Li, L., Chen, L., Cai, Z., Zhu, D., Lin, B., Cao, Y., Kong, L., Liu, Q., Liu, T., et al. (2024). Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450.

Wataoka, K., Takahashi, T., and Ri, R. (2024). Self-preference bias in llm-as-a-judge. *arXiv preprint arXiv:2410.21819*.

Wei, D., Luss, R., Hu, X., Paes, L. M., Chen, P.-Y., Ramamurthy, K. N., Miehling, E., Vejsbjerg, I., and Strobelt, H. (2025). ICX360: In-Context eXplainability 360 toolkit.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., and Zhou, D. (2023). Chain-of-thought prompting elicits reasoning in large language models.

Ye, J., Wang, Y., Huang, Y., Chen, D., Zhang, Q., Moniz, N., Gao, T., Geyer, W., Huang, C., Chen, P.-Y., Chawla, N. V., and Zhang, X. (2024). Justice or prejudice? quantifying biases in llm-as-a-judge.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al. (2023). Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Zytek, A., Pidò, S., and Veeramachaneni, K. (2024). Llms for xai: Future directions for explaining explanations.

A. Appendix

A.1. MExGen Unit Refinement Implementation

```
def mixed_level_attributions(documents, explainer, n_over_d=10, K=3,
    segment_type="s", k=3, phi=1/3):

  all_results = {}

  for doc_idx, document in enumerate(documents):
    # Sentence-level attributions
    # (1) - compute attribution scores
    output_dict_sent = explainer.explain_instance(
        document,
        oversampling_factor=n_over_d,
        max_units_replace=K,
        segment_type=segment_type
        )
    # sentences (units)
    sent_units = output_dict_sent["attributions"]["units"]
    # raw sentence attribution scores
    xi = np.array(output_dict_sent["attributions"]["prob"], dtype=float)
    # (2) - normalize scores in [-1,1]
    xi_min, xi_max = np.min(xi), np.max(xi)
    if xi_max == xi_min:
      psi = np.zeros_like(xi)
    else:
      psi = 2 * ( (xi - xi_min) / (xi_max - xi_min) ) - 1
    # (3) - compute Top-k psi - k largest psi scores
    top_k_idx = np.argsort(psi)[::-1][:k]
    top_k_mask = np.zeros_like(psi, dtype=bool)
    top_k_mask[top_k_idx] = True
    # (4) - score is higher or equal to threshold phi and in top-k
    refine_mask = (psi >= phi) & top_k_mask
    units_to_be_refined = np.where(refine_mask)[0].tolist()

    # Phrase-level attributions for refined sentences
    phrase_explanations = {}
    for idx in units_to_be_refined:
      sentence = sent_units[idx]
      output_dict_phrase = explainer.explain_instance(
          sentence,
          oversampling_factor=n_over_d,
          max_units_replace=K,
          segment_type="ph"
          )
```

```
42        attributions = output_dict_phrase["attributions"]
43        # save attribution dictionary for every top sentence
44        phrase_explanations[idx] = attributions
45
46      all_results[doc_idx] = {
47          "doc": document,
48          "distilbart_sum": output_dict_sent["output_orig"].output_text[0],
49          "sentence_attributions": output_dict_sent["attributions"],
50          "refined_indices": units_to_be_refined,
51          "phrase_attributions": phrase_explanations
52          }
53
54    return all_results
```

Listing A.1: Python implementation of the Multi-Level Unit Refinement procedure (Algorithm 1 Paes et al. (2025)) and the following calculation of MExGen phrase-level attributions (using the ICX360 toolkit Wei et al. (2025)).

A.2. System Prompt: Phase 1 (Baseline)

```
You are an Expert LLM Evaluator specializing in Model Identification.
Your goal is to select the summary from candidates A-E that was produced
by the Target Model.

  You will be provided with 3 examples of the Target Model's behavior.
For a new article, your task is to select the candidate that best matches
the Target Model's summarization behavior shown in the provided examples.

  Here are the rules of the evaluation:

  (1) You should prioritize evaluating whether the candidate summary matches
the target model's style and behavior shown in the examples.

  (2) You should NOT select a response because it is more fluent, detailed,
or human-like.  The Target Model often has specific constraints
(e.g., simplistic verbs, strict length) that must be preserved.

  (3) You should avoid any potential bias and your judgment should be as
objective as possible.  Here are some potential sources of bias:

   - The order in which the summaries are presented should NOT affect
     your judgment, as all candidates A-E are equally likely to be
     the Target Model's summary.

   - Summary length should only be a factor if it matches the Target Model's
     length constraints observable in the examples.

  Your reply should strictly follow this format:

  **Reasoning:** <Step 1:  Infer the common style or pattern from the three
                  examples.  Step 2:  Identify which candidate matches
                  that inferred pattern.>

  **Result:** <A, B, C, D or E>

  Here is the data.

  Target Model Demonstrations:
  {demonstrations}

  New Input Article:
  {NewInput}

  Response A: {Candidate1}
  Response B: {Candidate2}
  Response C: {Candidate3}
  Response D: {Candidate4}
  Response E: {Candidate5}
```

Figure A.1.: Phase 1 system prompt template for the 5-way multiple-choice attribution task, adapted from the Selene Mini pairwise evaluation template (Alexandru et al., 2025). Placeholders in curly braces are filled dynamically for each test instance.

A.3. System Prompt: Phase 2 (Top Sentence) Modification

The Phase 2 prompt follows the same structure as the Phase 1 prompt template (Figure A.1) with the following addition:

```
For every example you will receive:

  1. The Input Article.

  2. The Most Influential Sentence:  The specific sentence from the article
     that the Target Model prioritized to create its summary.

  3. The Target Model's Summary.
```

A.4. Contrastive Prompt Modifications

The contrastive prompt extends the Phase 1 prompt template (Figure A.1) with an additional element:

```
You will be provided with 3 examples of the Target Model's behavior.
For every example you will also be provided with a summary that was
NOT produced by the Target Model and does NOT match the Target Model's
summarization style.  Your task is to select the candidate that best
matches the Target Model's summarization behavior shown in the provided
examples.  Avoid picking the candidate that was NOT produced by the Target
Model.
```

For Phase 2, the demonstration block from Section A.3 is extended to:

```
For every example you will receive:

  1. The Input Article.

  2. The Most Influential Sentence:  The specific sentence from the article
     that the Target Model prioritized to create its summary.

  3. The Target Model's Summary.

  4. A Non-Target Summary that was NOT produced by the Target Model and
     does NOT match the Target Model's summarization style.
```

The reasoning instruction in both phases was also modified to:

```
**Reasoning:** <Step 1:  Infer the common style or pattern from the
three examples.  Step 2:  Identify which candidate matches that inferred
pattern.  Do not forget how the summary should NOT look like.>
```

A.5. K-Means ++ Cluster Variations XSum Validation Set

2D PCA projections of the XSum validation embedding space under K-Means++ clustering with k = 2,3,4,5, explored during development to select the cluster count for the diverse few-shot retrieval strategy (Section 3.6.2). Black dots mark cluster centers. For the final experiments k = 3 was chosen, balancing few-shot prompt size against cluster coherence.

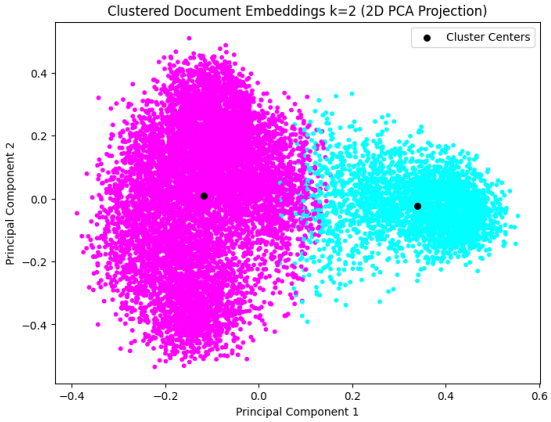


Figure A.2.: k = 2

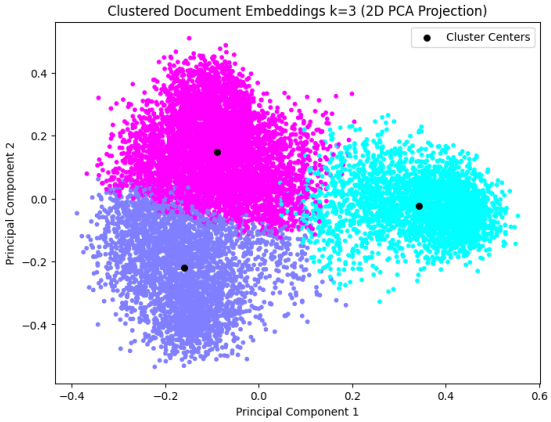


Figure A.3.: k = 3

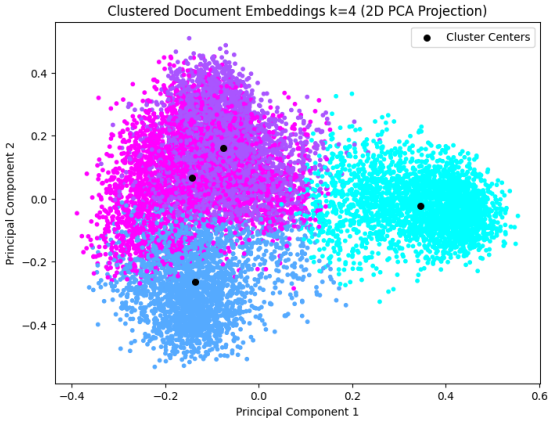


Figure A.4.: k = 4

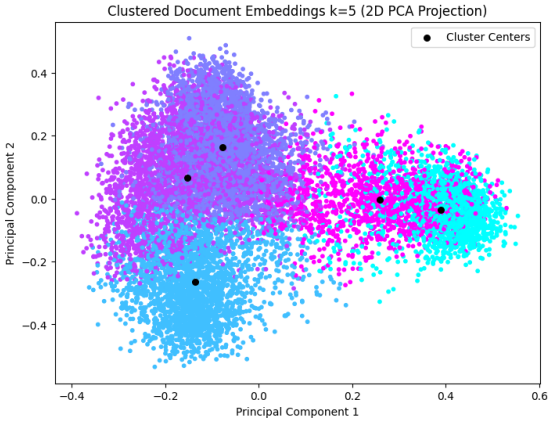


Figure A.5.: k = 5

A.6. System Prompt Example Pairwise Control Experiment

```
You are an Expert LLM Evaluator specializing in Model Identification.
Your goal is to select the response that matches the Target Model
described in the instruction.
Select Response A or Response B, whichever is more consistent with the
Target Model demonstrations provided in the instruction.
```

Here are the rules of the evaluation:

```
(1) You should prioritize evaluating whether the response is consistent
with the examples provided in the instruction.
(2) You should NOT select a response because it is more fluent, detailed,
or human-like.
(3) You should avoid any potential bias and your judgment should be as
objective as possible.  Here are some potential sources of bias:
   - The order in which the responses were presented should NOT affect
     your judgment, as Response A and Response B are **equally likely** to
     be created by the Target Model.
   - The length of the responses should only be a factor if it matches the
     Target Model behavior observable in the given examples.
```

Your reply should strictly follow this format:

```
**Reasoning:** <feedback evaluating the responses>
**Result:** <A or B>
```

Here is the data.

```
 Instruction:
‘‘‘
{instruction}
‘‘‘

 Response A:
‘‘‘
{Candidate1}
‘‘‘

 Response B:
‘‘‘
{Candidate2}
‘‘‘
```

Figure A.6.: Phase 1 system prompt template used for the tournament-style control experiment (Section 3.7.3), adapted directly from the Selene Mini pairwise evaluation template (Alexandru et al., 2025). This template preserves the original Response A vs. Response B format Selene was trained on. The Phase 2 version adds the most influential sentence to each demonstration.