Latent Debate: A Surrogate Framework for Interpreting LLM Thinking towards Binary Decisions

Lihu Chen* **Xiang Yin*†** **Francesca Toni**
Imperial College London
United Kingdom
{lihu.chen,xy620,ft}@imperial.ac.uk

*Equal contribution †Corresponding author

Abstract

Understanding the internal 'thinking' process of Large Language Models (LLMs) and their connections to hallucinations remains a key challenge. To this end, we introduce *latent debate*, a structured surrogate framework for interpreting model outputs on True/False prediction tasks through the lens of internal latent arguments and interactions amongst them. Unlike human debates, latent debate captures the hidden supporting and attacking signals that arise within a model during a single inference. We first present a model- and task-agnostic conceptual framework, and then instantiate it symbolically to approximate the thinking process of LLMs towards binary decisions. Empirical studies demonstrate that our latent debate is a faithful structured surrogate model that has highly consistent predictions with the original LLM, while providing a form of interpretability. We also demonstrate that our latent debate provides a strong baseline for hallucination detection. Specifically, we identify strong correlations between debate patterns and hallucinations, such as a high degree of disagreement in the middle layers of the latent debate surrogate is linked to a higher risk of hallucinations. Our findings suggest that latent debate provides a new framework for understanding the internal processes of LLMs. The code is available at `https://github.com/XiangYin2021/LatentDebate`.

1 Introduction

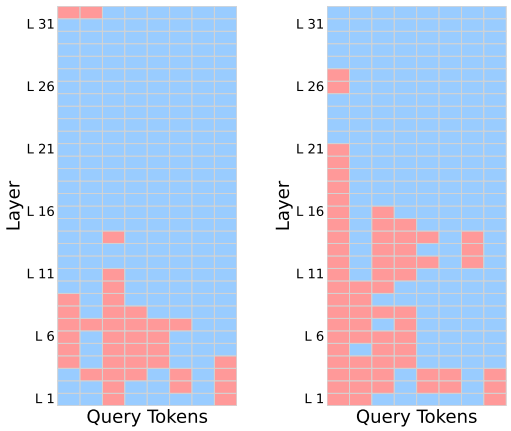
Large Language Models (LLMs) have made remarkable progress on many reasoning tasks, yet they continue to suffer from hallucinations [Xu et al., 2024, Huang et al., 2025]. For example, LLMs may generate answers that contradict user prompts or conflict with the source of training data [Ji et al., 2023, Kalai et al., 2025, Bang et al., 2025], seriously undermining their reliability and trustworthiness. This is further aggravated by the fact that, due to their opacity, it is difficult to understand why LLMs make given predictions, or why their 'thinking' process is flawed.

Recent work in mechanistic interpretability has examined hallucinations through various internal signals, including activations [Ferrando et al., 2025], attention patterns [Chuang et al., 2024], and hidden states [Azaria and Mitchell, 2023]. Another line of relevant research [Wang et al., 2022] leverages external consistency, i.e., the agreement among multiple answers, to analyze hallucination behaviors. Their findings reveal that hallucinated outputs tend to have low self-consistency [Wang et al., 2022]. This phenomenon suggests that strong agreement among multiple answers often yields more certain and accurate answers, whereas disagreement indicates higher uncertainty and can serve as a good signal for detecting hallucinations. Further approaches [Irving et al., 2018, Du et al., 2024, Chen et al., 2024, Liang et al., 2024a] introduce multiple-agent debate to reduce hallucinated answers via a debate process of multiple LLMs, often outperforming single-model baselines.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Inspired by these prior studies of mechanistic interpretability and disagreement/debate, we aim to understand how hallucinations emerge within a model, but shifting to a novel notion of *latent debate*, i.e., arising among different layers and thinking steps within an individual model for a single inference [Chuang et al., 2023, Liang et al., 2024b, Xie et al., 2024], rather than externally as in prior work based on debate processes. Unlike in conventional human debate, where arguments are in natural language, the arguments in our latent debate are to be understood in a metaphorical sense as they correspond to latent states in models. Intuitively, latent debate aggregates metaphorical arguments to reflect the thinking and decision-making process beneath the surface. In psychological theories, the human thinking process often involves internal debate-like behaviors such as inner speech [Barker and Wiseman, 1966] and the dialogical self [Hermans, 2001]. Here, we extend these psychological insights to describe an analogous process taking place within a model.

We use the concept of *latent debate* to build a transparent and simple surrogate model that approximates binary predictions of a complex black-box model, with the goal of understanding and interpreting the target model's behavior. We focus on two key research questions: (1) *Can we use our notion of latent debate to model the LLM thinking process?* (2) *Can latent debate identify and interpret hallucinations?* Thus, our approach is not intended to improve the target model's capabilities, such as improving factuality, but rather to provide insights into its internal decision-making process.



(a) ***Correct**: The city of Zhangzhou is in China.*

(b) **Hallucinated:** *The letter J is the most commonly used letter in English-language writing.*

Figure 1: Visualizations of latent debate for two claims (both predicted as True) using selected tokens of Llama-8B. Red/blue cells represent attacking/supporting arguments, respectively. High controversy indicates hallucination.

To answer the first question, we present a conceptual framework of latent debate that depicts (dis)agreements within a model, while being model- and task-agnostic. The framework consists of three abstract components: latent arguments derived from internal signals, an argument interpreter that translates these latent arguments into human-readable opinions, such as supporting or attacking a claim, and a thinking module that aggregates them to make the final decisions. We then instantiate this framework in decoder-based LLMs on True/False prediction tasks (see case studies in Figure 1), where hidden states serve as latent arguments, the unembedding matrix acts as the argument interpreter, and the thinking module is realized through a symbolic argumentation framework, in the spirit of [Čyras et al., 2021]. An empirical study demonstrates that this latent debate acts as a structured surrogate model, providing a faithful approximation of LLM thinking, which achieves up to 97% consistency with LLaMA-13B decisions. These findings suggest that latent debate provides a useful approach to approximate internal signals for binary decision tasks.

To address the second question, we extract features from the latent debate, e.g., the number of internal chains of disagreement and argument strengths, and train a simple tree-based classifier to distinguish hallucinated from non-hallucinated outputs. We find that our latent debate can achieve highly competitive performance in hallucination detection for binary decision tasks. We then use SHAP attribution scores [Lundberg and Lee, 2017a] to identify which features most strongly drive hallucination predictions. Our analyses indicate that a high degree of controversy in the latent debate, particularly in the middle layers, is the strongest predictor of hallucination.

In summary, our contribution is threefold. (1) We propose latent debate, a novel, model-agnostic framework that leverages internal arguments to approximate a model's thinking process. (2) We present a symbolic instantiation of latent debate that serves as a structured surrogate for LLM binary decision tasks. (3) We develop a simple latent debate-based approach to detect hallucinations, which helps us identify distinct debate patterns.

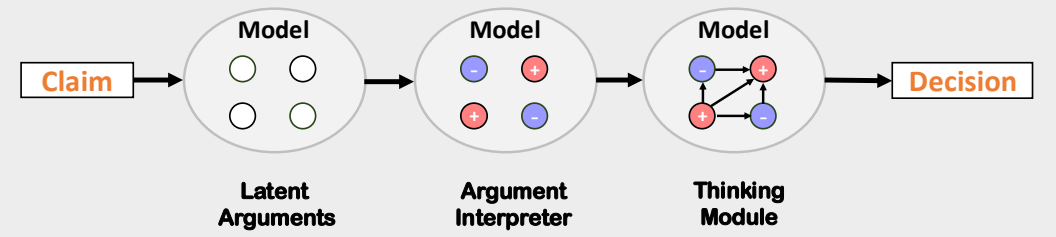


Figure 2: The conceptual framework of latent debate. Given an input (e.g., a *claim*), a set of *latent arguments* are identified, i.e., model components (raw latent signals) that convey the model's opinions toward the input. These arguments are then processed by the *argument interpreter*, identifying the arguments' supporting or attacking stance towards the input. The resulting attacking and supporting arguments are fed into the *thinking module*, which reaches the final True/False *decision*.

2 Related Work

Debate in AI. Debate has emerged as a powerful approach for improving factuality and reasoning. Pioneering work on AI safety via debate [Irving et al., 2018] models debate as a self-play game with a (human) judge and provides core theoretical motivation. Recent work adopts multiple LLM agents to debate over individual responses jointly, with the final decision made either through consensus [Du et al., 2024, Chen et al., 2024] or by a judge [Liang et al., 2024a]. This debate strategy can outperform single model baselines on a wide variety of reasoning tasks. Subsequent research proposes refined debate approaches [Li et al., 2024, Liu et al., 2024] using multi-agent debate as an evaluator [Chan et al., 2024]. In this work, we focus on debates operating implicitly within a model (agent) rather than externally visible debates among multiple agents.

Internal Consistency. Another line of work focuses on how to use the consistency of internal model states, such as logits and activations, to improve model outputs [Liang et al., 2024b]. For example, DoLa [Chuang et al., 2023] proposes a decoding strategy that contrasts logits between later layers and earlier layers for improving LLM truthfulness. Xie et al. [2024] adopt internal consistency, i.e., how middle layers' predictions (dis)agree with the final layer, to guide LLM decoding. In this work, we aim to use latent (dis)agreements to build a surrogate framework for interpreting the thinking process of a model rather than enhancing model outputs. We also show that our surrogate model can help detect hallucinations.

Computational Argumentation in Explainable AI. *Argumentation Frameworks (AFs)* [Dung, 1995] are a fundamental formalism in *computational argumentation* [Atkinson et al., 2017, Čyras et al., 2021], a well-established research area in AI. According to [Dung, 1995], an AF consists of a set of *arguments* and a binary *attack* relation among them. Arguments are seen as abstract entities, while the attack relation captures conflicts between arguments. AFs and their extensions, such as incorporating weights for arguments and a support relation between arguments, collectively referred to as Quantitative Bipolar Argumentation Frameworks (QBAFs) [Rago et al., 2016, Baroni et al., 2019], have been widely adopted in Explainable AI [Čyras et al., 2021]. AFs can serve as *surrogate models* to approximate the inner structure and decision-making process of AI systems [Čyras et al., 2021, Potyka, 2021, Potyka et al., 2023, Ayoobi et al., 2023]. In this work, we are the first to introduce the model-agnostic concept of latent debate and adopt a QBAF for the thinking module of an LLM.

3 Methodology

3.1 Problem Statement

We aim to obtain a *structured surrogate model* to understand the internal thinking of a target model, as opposed to a *conventional surrogate model* solely imitating the input-output behaviour of the model. More specifically, given a target model $M : \mathcal{X} \to \mathcal{Y}$, a conventional surrogate model $S : \mathcal{X} \to \mathcal{Y}$ imitating the input-output behaviour of the target model is such that $S(x) \approx M(x)$. Such surrogates offer a simplified and interpretable approximation of the model's outputs [Asher et al., 2015, Kudela and Matousek, 2022]. However, this type of surrogate does not truly reflect the internal structure (or

internal thinking) of the model. To address this limitation, structured surrogate models explicitly account for the model's internal organization [Munk et al., 2022, Páez, 2024]. If the target model has a known internal structure:

$$M : \mathcal{X} \to \mathcal{H} \times \mathcal{Y}, \qquad M(x) = (h(x), y(x)) \tag{1}$$

where $h(x) \in \mathcal{H}$ is the structure, our goal is to construct:

$$S(x) = (\hat{h}(x), \hat{y}(x)) \tag{2}$$

that approximates the internal computational structure of the target model, i.e., such that $\hat{h}(x) \approx h(x)$, while, at the same time, being faithful to the target model by making highly consistent predictions with it, i.e., $\hat{y}(x) \approx y(x)$. Note that our main goal here is to build a simple surrogate model to imitate a complex model. Improving the ability of a model is not our purpose and is left to future work.

To obtain structured surrogate models $S(x)$ with the above characteristics, we define the concept of *latent debate* as follows.

3.2 Conceptual Framework

Latent Debate. A latent debate is an internal form of argumentation that happens within a single model (or agent). Instead of having multiple explicit agents participating in a debate, latent debate leverages upon the hidden inconsistency inside the model that simultaneously carries supporting and attacking arguments toward the input to determine the corresponding output. These arguments are not directly expressed in natural language but shape the model's thinking process beneath the surface. The strength between supporters and attackers may be imbalanced. Overwhelming supporters can lead to a very certain positive decision, and vice versa. This uncertainty reflects how the model arrives at a final (binary) decision. Latent debate consists of three key components: latent arguments, argument interpreter, and thinking module, as shown in Figure 2.

Latent Arguments. A latent argument is an internal signal within a model, implicitly conveying supporting or attacking opinions toward the input. Such signals can arise from different sources, e.g., activations or attention patterns. Since they live in the model's latent space, these arguments are not directly visible or human-readable, but they still express how intermediate steps think about the input.

Argument Interpreter. The argument interpreter makes these latent arguments interpretable. It translates latent arguments into a form we can understand, such as a binary label. At the same time, it tells us how strongly each argument supports or attacks the input, turning vague internal signals into measurable opinions.

Thinking Module. The thinking module combines all the decoded arguments to reach a final decision. It looks at how the arguments interact — some supporting, some attacking — and weighs them against each other. By aggregating these arguments and how they interact, the module produces a final outcome that reflects the overall internal debate of the model.

It is important to note that this framework is not tied to any specific model architecture. The notions of latent arguments, argument interpreter, and thinking module are abstract components that can be realized in many different ways. For example, latent arguments may be instantiated through hidden states, attention patterns, or other internal signals; argument interpreters can be designed using projection, probing, or alternative interpretability tools; and thinking modules may adopt symbolic argumentation frameworks, probabilistic aggregation, machine-learning methods, including artificial neural networks. This flexibility ensures that the latent debate framework can be adapted to a wide range of models and tasks beyond the particular instantiation we study next in this paper.

3.3 Symbolic Instantiation for Latent Debate in LLM True/False Prediction

We now describe how the abstract concepts of latent debate can be instantiated in the context of LLM True/False prediction tasks [Vaswani et al., 2017]. We adopt a symbolic argumentation framework as the thinking module, which is transparent and efficient.

Formally, given an input $\mathbf{x} = (x_1, \ldots, x_N)$ with a binary label $c \in \{\text{True}, \text{False}\}$, a decoder-based LLM generates a decision $\mathbf{y} = (y_1, \ldots, y_T)$. Both the input and decision tokens are drawn from the

same vocabulary, i.e., $x_n, y_t \in \mathcal{V}$. Each token y_t in the decision is generated conditionally based on the preceding tokens and the input, following the distribution: $y_t \sim P(y_t \mid \mathbf{y}_{\leq t-1}, \mathbf{x})$. In a multiple agent debate setting [Du et al., 2024, Liang et al., 2024a], the process involves generating multiple decisions $\mathcal{Y} = \{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(K)}\}$, which may support ($\mathbf{y}^+$) or attack ($\mathbf{y}^-$) the input. A final decision is then derived by aggregating these arguments, often through some form of consensus or voting strategies. In contrast, we define that a latent debate takes place inside the model processing $\mathbf{x}$ before generating $\mathbf{y}$.

3.3.1 Latent Arguments in LLMs

An LLM consists of L layers. Let f_θ denote the transformation function for computing hidden states, parameterized by θ. The hidden state for the token x_n of the input at layer l is computed as: $\mathbf{h}_n^{(l)} = f_\theta(\mathbf{h}_1^{(l-1)}, \ldots, \mathbf{h}_n^{(l-1)})$, where $\mathbf{h} \in \mathbb{R}^d$, with d the dimensionality of the hidden states, corresponds to the normalized sum of residual and sub-layer outputs. We treat each hidden state $\mathbf{h}_n^{(l)}$ as a *latent argument*, a representation that implicitly encodes supportive or attacking views with respect to the input, though not directly observable in natural language. Given a claim comprising N tokens and an LLM with L layers, we thus obtain $N \times (L-1)$ latent arguments over the LLM's internal computation. We exclude the final (output) layer because it directly produces the probability distribution over next tokens. Instead, our goal is to depict the intermediate thinking dynamics encoded in the hidden layers prior to that final mapping.

3.3.2 Argument Interpreter in LLMs

To make latent arguments interpretable, the instantiated argument interpreter projects a hidden state into the vocabulary space using the unembedding matrix $\mathbf{W}^{\text{unemb}} \in \mathbb{R}^{|\mathcal{V}| \times d}$. This produces a probability distribution over vocabulary tokens, which has been widely used in mechanistic studies [nostalgebraist, 2020, Belrose et al., 2023]. By examining the probabilities assigned to specific tokens `True` and `False`, we can quantify the opinion of each latent argument, i.e., how much it supports or attacks the input.

$$\text{interpret}(\mathbf{h}_n^{(l)}) = \text{Softmax}(\mathbf{W}^{\text{unemb}}_{[\text{True, False}]}(\mathbf{h}_n^{(l)})) \tag{3}$$

The output of interpret$(\cdot)$ enables interpretation of the latent arguments through the lens of token-level semantics.

3.3.3 Thinking Module in LLMs

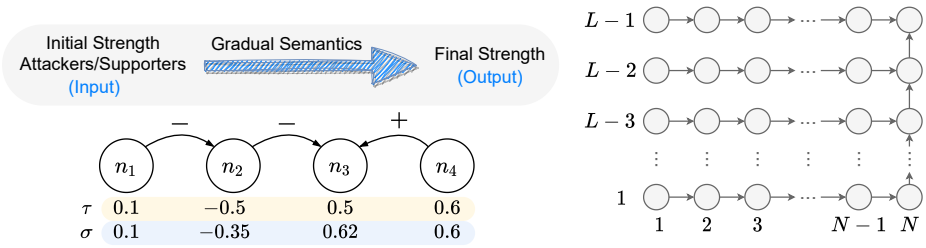
Finally, a thinking module is applied to the set of interpretable arguments in order to produce a final judgement $c \in \{\texttt{True}, \texttt{False}\}$. This process is formalized as:

$$\text{cls}(\mathbf{x}) = \text{think}(\mathcal{H}), \qquad \mathcal{H} = \left\{ \text{interpret}\left(\mathbf{h}_n^{(l)}\right) \;\middle|\; 1 \leq n \leq N,\ 1 \leq l \leq L-1 \right\} \tag{4}$$

where $\text{cls}(\mathbf{x})$ is capable of outputting a label associated with the final decision c. To perform the thinking step, we adapt a symbolic approach from computational argumentation, Quantitative Bipolar Argumentation Framework (QBAF) [Baroni et al., 2019], as the think$(\cdot)$ function, which accounts for both supporting and attacking relationships among arguments to yield a coherent, weighted judgment.

Definition 1. *A* QBAF *is a quadruple $\mathcal{Q} = \langle \mathcal{A}, \mathcal{R}^-, \mathcal{R}^+, \tau \rangle$ where $\mathcal{A}$ is a finite set of* arguments*; $\mathcal{R}^- \subseteq \mathcal{A} \times \mathcal{A}$ is a binary* attack *relation; $\mathcal{R}^+ \subseteq \mathcal{A} \times \mathcal{A}$ is a binary* support *relation; τ is an* initial strength function *($\tau : \mathcal{A} \to [-1, 1]$).*

A gradual semantics [Baroni et al., 2015, Rago et al., 2016, Potyka, 2018, Amgoud and Ben-Naim, 2018] applied to a QBAF is capable of considering the overall debate situation and outputting *final strengths* for all arguments (as shown in Figure 3a), which can be used to obtain the binary (*True/False*) predictions (e.g. by thresholding). In the figure, each node n_i corresponds to an argument, and the $\tau(\cdot)$ function assigns the *initial strength*, indicating its polarity and strength magnitude before applying the semantics. The directed edges between nodes represent the relationships: edges labeled with $-$ are *attacks*, indicating that one argument undermines another (two arguments with different



(a) An example QBAF

(b) QBAF skeleton for LLMs.

Figure 3: QBAFs and LLMs. (a) An example QBAF showing how initial strengths (τ) of arguments $n_1, \ldots, n_4$ are transformed through gradual semantics based on attacking (-) and supporting (+) relations to produce final strengths (σ). (b) Skeleton of a QBAF drawn from an LLM architecture, where each node represents a specific token at a specific layer. To obtain a QBAF, the directed edges need to become attacks or support and the nodes need to be equipped with an initial strength.

polarities), while edges labeled with $+$ are *supports*, meaning that one argument reinforces another. The $\sigma(\cdot)$ beneath the nodes is the gradual semantics,which reflects how the collective influence of attackers and supporters modifies the outcome. See Example A.4 in the appendix for a detailed computation process.

Formally, gradual semantics provide the rules that control how initial strengths are updated and result in the computation of *final strengths*. To match the thinking of LLMs, we define a new semantics. We start from arguments with no attackers or supporters, whose final strength is the same as their initial strength. For the remaining arguments, the final strength is updated along the edges by considering the influence of both attackers and supporters. As conventional [Potyka, 2021], this process involves two components: *aggregation* and *influence*. For an argument $\alpha \in \mathcal{A}$, the aggregation step computes its *energy* E_α by summing the strengths of its attackers and supporters (β):

$$E_\alpha = \sum_{\{\beta \in \mathcal{A} | (\beta, \alpha) \in \mathcal{R}^- \cup \mathcal{R}^+\}} \sigma(\beta) \tag{5}$$

The influence step then updates the initial strength $\tau(\alpha)$ by combining it with the computed energy:

$$\sigma(\alpha) = \tanh(E_\alpha) + \tau(\alpha) \cdot (1 - \tanh(|E_\alpha|)) \tag{6}$$

This equation updates the final strength of an argument by combining the aggregated influence from its attackers and supporters ($\tanh(E_\alpha)$) with its own initial strength $\tau(\alpha)$. Here, $\tau(\alpha) = w \times \text{interpret}(\mathbf{h}_n^{(l)})$, where w is the token-wise weight that measures the semantic contribution of this current token to the entire sentence (see details in Section A.2). In other words, the weight scales how strongly the initial strength propagates since each token contributes in a different way to the semantic meaning.

The strength function $\sigma : \mathcal{A} \to [-1, 1]$ assigns each argument a value where the sign indicates polarity towards the output decision (supportive or attacking) and the absolute value indicates magnitude. The values in [-1,0) correspond to negative labels, while values in [0,1] correspond to positive labels.

Creating QBAFs for LLMs Figure 3b illustrates how we construct a QBAF for LLM architectures. Each row corresponds to a transformer layer, and each circle represents an argument associated with a thinking step (*a token*) at that layer. In our instantiation,we treat the last few tokens of the prompt as thinking steps, which are the tokens generated after the model has already seen the entire input. In this work, we treat the final token of the input, along with the subsequent auxiliary tokens (*"The statement is True or False:"*), as the model's thinking steps. In this way, even though thinking tokens cannot attend to subsequent tokens, they can attend to the full question and the beginning prompt, which is sufficient to provide the task specification and the basic context. Hence, the tokens after the input serve as meaningful intermediate thinking units in the latent debate process.

Arguments are first connected within a layer from left to right, following the natural order of tokens in the input sequence. The right-most node in each layer summarizes the thinking results for that layer. We then connect these right-most nodes across layers, from lower to higher, since upper layers are closer to the final decision. The node in the top-right corner thus considers the overall

	common_claim 500	counterfact 500	company 500	TriviaQA 500	MuSiQue 500	TruthfulQA 500	Avg –
Llama-8B (%)							
Random	76.8	64.2	67.4	66.4	68.8	79.2	70.5
Average	92.2	67.4	80.4	73.0	**77.0**	90.2	80.0
Majority Voting	**92.4**	67.0	80.4	73.0	**77.0**	90.2	80.0
Latent Debate	**92.4**	**78.2**	**89.2**	**74.0**	**77.0**	**90.6**	**83.6**
Latent Debate – w/o token weight	92.2	68.2	80.2	73.0	**77.0**	90.2	80.1
Latent Debate – with quadratic connection	91.8	64.0	80.0	73.2	**77.0**	90.2	79.4
Mistral-7B (%)							
Random	64.0	65.4	74.0	75.0	75.8	70.4	70.8
Average	86.8	89.0	**98.8**	95.6	90.2	84.2	90.8
Majority Voting	89.8	90.6	98.6	**96.0**	**91.6**	87.0	92.3
Latent Debate	**90.0**	**91.0**	97.8	95.4	91.2	**89.2**	**92.4**
Latent Debate – w/o token weight	87.0	89.0	**98.8**	95.6	90.2	84.0	90.8
Latent Debate – with quadratic connection	81.2	76.8	91.6	95.0	90.2	77.6	85.4
Llama-13B (%)							
Random	65.2	65.0	78.8	66.8	62.4	68.8	67.8
Average	90.0	90.0	98.8	86.2	85.4	89.6	90.0
Majority Voting	85.4	88.0	98.2	84.2	84.4	87.0	87.9
Latent Debate	**98.4**	**95.2**	**99.6**	96.2	**93.6**	**96.8**	**96.6**
Latent Debate – w/o token weight	95.6	91.2	98.6	**97.6**	93.2	93.6	95.0
Latent Debate – with quadratic connection	80.2	66.0	59.8	95.2	93.0	83.4	79.6

Table 1: Consistency scores on binary and open-ended question-answering datasets with three LLMs. Each dataset contains 500 examples. The first three datasets use the last six prompt tokens, while TriviaQA, MuSiQue, and TruthfulQA use the last five. Avg is the arithmetic mean across the six datasets. The best scores within each model are shown in bold.

information to make decisions, and we use its output as the binary classification. The initial strength of each argument is determined by its probability defined in Equation 3, which is then normalized to a value in $[-1, 1]$. The sign of the initial strength reflects its polarity (positive or negative), while the magnitude encodes the confidence. Relations between arguments are determined by comparing polarities: if two connected arguments share the same polarity, the edge is a support. Otherwise, it is an attack. Because polarity depends on evolving strengths during computation, edge types may be updated dynamically.

This topology is intentionally simple to enhance interpretability. In particular, we avoid connecting arguments of the same token across layers as this way does not bring clear benefits (see results of quadratic connections in Table 1). In the current implementation, our framework captures structural patterns within the internal structure that are relevant for decision-making rather than merely input-output imitation, i.e., latent debate is *a structured surrogate*.

3.3.4 Benefits

(1) Transparent and Interpretable. Our framework makes the internal thinking process of LLMs human-readable via a symbolic argumentation framework. Each latent signal is translated to a clear supporting or attacking argument, and the QBAF decision path can be visualized and explained rather than remaining a black box. **(2) Training-Free.** Our framework imitates the LLM thinking process without any tuning and training. Every component is lightweight, which makes the method easy to use. **(3) Property-Satisfying.** Because the reasoning process is formalized with a symbolic argumentation framework, the method inherits desirable theoretical properties such as monotonicity in [Baroni et al., 2018]. In practice, this guarantees that the framework behaves in an intuitively consistent way when adding new arguments or changing the initial strength of arguments (see the proof and details in the appendix A.3).

4 Latent Debate as a Surrogate for Imitating LLM True/False Prediction

A core motivation of our latent debate is to approximate the thinking process of LLMs, allowing us to interpret their internal mechanisms. We wish that this transparent framework is faithful to the original model's predictions (*Prediction Fidelity*), i.e., the surrogate model should match outputs of the target black-box model [Papenmeier et al., 2019, Laugel et al., 2018]. To validate this, we conduct experiments on three balanced True/False prediction datasets: *common claims, counterfact, and company*. We also include three open-ended question-answering datasets mapped onto binary

claims: *TriviaQA, MuSiQue, and TruthfulQA* (see details in section B.2). We use three open-weight LLMs in the experiments: *meta-llama/Llama-3.1-8B, mistralai/Mistral-7B-Instruct-v0.3, and meta-llama/Llama-2-13B.*

To benchmark the faithfulness of our latent debate approach, we compare it against several intuitive baselines of *structured surrogates* [1]: (1) *Random*, (2) *Average*, (3) *Majority Voting*, (4) *Latent Debate – w/o token weight*, and (5) *Latent Debate – with quadratic connection*. The details of baselines are described in Section B.3. Since we aim to propose a transparent surrogate model, approaches that employ internal signals to improve output quality, such as DoLA Chuang et al. [2023], are not included in this experiment. We report the *consistency score*, i.e., the proportion of instances on which the decision derived via the latent debate exactly matches the original LLM's prediction.

Table 1 reports the consistency scores of different methods over 500 examples per dataset and three models. The latent debate approach achieves higher average consistency than baseline methods (*Random, Average, Majority Voting*) and two variants of latent debate for all three models, while matching or slightly trailing a baseline on a few individual datasets. For instance, with Llama-13B the latent debate method reaches 96.6% average consistency, while the best non-debate baseline (*Average*) reaches 90.0%. This demonstrates that latent debate is a strong structured surrogate for the True/False decisions made by the model. More importantly, our transparent approach gives interpretability by displaying internal supporting vs. attacking arguments that can be visualized and understood.

5 Can Latent Debate Identify Hallucinations?

Given that our latent debate surrogate aligns closely with the LLM's internal decision behavior, we now turn to a more diagnostic question: *Can we use latent debate to detect hallucinated answers?* Here, the term 'hallucination' refers to their (lack of) factuality, in the spirit of [Huang et al., 2025, Zhang et al., 2025], i.e., hallucinations emerge when the LLM's answers are inconsistent with established world knowledge. In other words, we use our transparent and faithful surrogate model to detect and analyze how the model thinks and why it hallucinates.

5.1 Latent Debate is a Strong Baseline for Hallucination Detection

In order to detect hallucinations, we train a simple tree-based classifier, XGBoost Chen [2016], to distinguish hallucinated from non-hallucinated outputs using features extracted from latent debate, which we refer to as *Latent Debate Detector*. The details of this detector are described in Section B.4 in the appendix. This pipeline allows us to both detect hallucination and interpret why they happen in terms of internal debate patterns.

Concretely, we extract the following five features related to debate patterns from each QBAF drawn from an LLM: (1) *number of attacks (NumAtk)*: the total number of attack edges in the QBAF, capturing how many conflicting arguments are present. (2) *average of initial strength (AvgInit)*: the arithmetic mean of the raw strengths of all latent arguments in the QBAF. (3) *average of final strength (AvgFin)*: the arithmetic mean of strength values after propagation under the chosen gradual semantics. (4) *variance of initial strength (VarInit)*: the statistical variance of the raw strengths of all latent arguments in the QBAF. (5) *variance of final strength (VarFin)*: the statistical variance of strength values after propagation under the chosen gradual semantics. Since not all layers contribute equally to a model's reasoning process, we divide latent arguments evenly across layers into three regions for better understanding internal mechanisms: lower, middle, and upper. We extract the above five features for each region.

We compare our latent debate detector to the value of these five features in isolation and another two commonly-used methods for hallucination detection: SelfCheckGPT [Manakul et al., 2023] and SAPLMA [Azaria and Mitchell, 2023]. The baseline details are described in Section B.5 in the appendix. Table 2 shows the comparison across different hallucination detectors. We can see that our latent debate detector can achieve a decent AUROC compared to baselines on average, which suggests that our approach can serve as strong baseline in distinguishing hallucinations from

[1] In this experiment, non-structured surrogate baselines are not compared, e.g., single-argument baselines (see discussions in Section B.6).

Method	Common Claim	CounterFact	Company	TriviaQA	MuSiQue	TruthfulQA	Avg
Hallucination Ratio	40.8%	15.8%	30.2%	25.0%	35.2%	45.8%	32.1%
AvgInit	0.75	0.66	0.86	0.91	0.71	0.58	0.74
AvgFin	0.58	0.54	0.68	0.82	0.62	0.62	0.64
VarInit	0.82	0.70	0.89	0.91	0.78	0.54	0.77
VarFin	0.78	0.63	0.78	0.90	0.65	0.50	0.71
NumAtk	0.55	0.53	0.68	0.65	0.51	0.55	0.58
SelfCheckGPT	0.60	0.53	0.72	0.65	0.58	0.45	0.59
Semantic Entropy	0.72 ± 0.00	0.69 ± 0.01	0.71 ± 0.01	0.77 ± 0.01	0.52 ± 0.00	0.51 ± 0.01	0.65
SAPLMA	0.82 ± 0.01	0.77 ± 0.00	0.86 ± 0.00	0.90 ± 0.01	$\mathbf{0.85 \pm 0.01}$	$\mathbf{0.71 \pm 0.03}$	**0.82**
Latent Debate Detector	$\mathbf{0.84 \pm 0.00}$	$\mathbf{0.83 \pm 0.00}$	$\mathbf{0.93 \pm 0.00}$	$\mathbf{0.91 \pm 0.00}$	0.78 ± 0.01	0.61 ± 0.00	**0.82**

Table 2: AUROC scores for the identified features in isolation (AvgInit, . . . , NumAtk), hallucination detection baselines, and Latent Debate Detector. Results are reported as the average score over three random seeds where available. The best results are highlighted in bold.

non-hallucinations. More importantly, our simple method offers interpretable features to analyze why the model hallucinates, discussed next.

5.2 What Debate Patterns are Correlated with Hallucinations?

To study which debate pattern is correlated with hallucinations, we apply SHAP attribution [Lundberg and Lee, 2017a] to determine which features most strongly contribute to hallucination. Using tree-based classifiers with SHAP attribution is a broadly adopted approach for feature analysis and interpretability [Lundberg and Lee, 2017b, Ponce-Bobadilla et al., 2024].

Specifically, we construct a small dataset sampled from the input data to represent the reference distribution. Using SHAP's `TreeExplainer` with an interventional feature-perturbation strategy, we compute SHAP values that quantify each feature's contribution to the model's predicted probability for each input sample. SHAP feature importance is computed separately for each region, and then we aggregate the value for each feature defined in Section 5.1 to see whether there are any common patterns. As Figure 4 shows, the variance of initial strengths (VarInit) shows the strongest influence in hallucination detection, especially in the middle region, suggesting that disagreements arising therein play an important role in detecting hallucinations. Moreover, the initial strength (AvgInit) in the top layers serves as a strong indicator of hallucinated predictions, as it reflects the model's own confidence in its output. Additionally, the variance of the final strength (VarFin) is informative for this task because the QBAF explicitly accounts for the degree of agreement among neighboring arguments. In contrast, the lower regions show relatively weaker importance, indicating that the early stage is less predictive of hallucinations. This aligns with the notion that LLMs build knowledge hierarchically [Geva et al., 2022]: early layers capture low-level features, middle layers synthesize and build semantic abstractions, and top layers focus on the final output or next-token prediction.

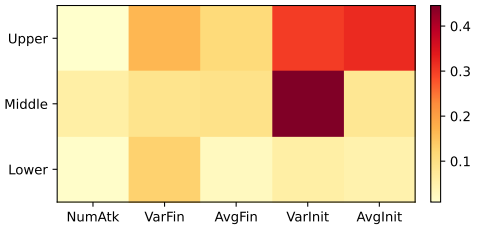


Figure 4: Feature importance across layer regions (Lower, Middle, Upper) and feature types. NumAtk: number of attacks; VarFin: variance of final strength; AvgFin: average of final strength; VarInit: variance of initial strength; AvgInit: average of initial strength.

6 Design Choices in Latent Debate

6.1 Choice of Argument Interpreter

Our framework uses Logit Lens to interpret latent arguments. One may argue that this tool is not reliable at the middle layers. To examine the impact of this choice, we compare Logit Lens with the official Tuned Lens [Belrose et al., 2023] for Meta-Llama-3-8B-Instruct. We evaluate both prediction consistency and hallucination detection performance on three datasets.

As shown in Table 3, Tuned Lens achieves higher average consistency with the LLM's final predictions (93.20% vs. 86.60%). However, its hallucination detection performance is substantially lower than that of Logit Lens (67.03 vs. 86.54 AUROC). These results suggest that higher consistency with

Metric	Interpreter	Common Claim	CounterFact	Company	Average
Prediction Consistency	Logit Lens	92.40	78.20	**89.20**	86.60
	Tuned Lens	**99.80**	**96.40**	83.40	**93.20**
Hallucination Detection (AUROC)	Logit Lens	$\mathbf{83.68 \pm 0.31}$	$\mathbf{82.50 \pm 0.31}$	$\mathbf{93.44 \pm 0.41}$	**86.54**
	Tuned Lens	59.24 ± 0.29	57.33 ± 0.67	84.54 ± 0.20	67.03

Table 3: Comparison of Logit Lens and Tuned Lens as argument interpreters in latent debate. Consistency measures agreement with the original LLM's predictions, while AUROC evaluates hallucination detection. The best results are highlighted in bold.

Connection Topology	Common Claim	CounterFact	Company	TriviaQA	MuSiQue	TruthfulQA	Avg
Original	**92.40**	**78.20**	**89.20**	**76.80**	**77.80**	**91.20**	**84.27**
Adjacent Grid	79.00	43.80	64.00	52.20	55.80	56.80	58.60
Original Quadratic	91.80	64.00	80.00	71.60	75.00	88.60	78.50
Fully Quadratic	56.20	38.80	43.40	49.80	48.60	50.00	47.80

Table 4: Comparison of different graph topologies in latent debate. Consistency scores (%) measure agreement with the original LLM predictions. The best results are highlighted in bold.

the final prediction does not necessarily yield factual answers. One possible explanation is that the tuning objective explicitly aligns middle layers with the final-layer distribution, which may distort middle-layer information relevant to factuality signals.

6.2 Impact of Graph Structure

Our latent debate framework adopts a simple graph topology, where arguments are connected sequentially within each layer, while cross-layer connections are restricted to the final token. To examine this design choice, we evaluate three alternative graph topologies that introduce richer interactions between arguments: *a) Adjacent Grid*: Each argument is connected to its immediately adjacent argument on the right within the same layer, and to the argument at the same token position in the next layer. *b) Original Quadratic*: Within each layer, every argument is connected to all subsequent arguments on its right, while only the final-token arguments are connected across adjacent layers. *c) Fully Quadratic*: Each argument is connected to all subsequent arguments within the same layer and to arguments at the same token position in all higher layers.

As shown in Table 4, the original topology achieves the highest consistency across all six datasets, with an average of 84.27%. More complex topologies consistently underperform, including Original Quadratic (78.50%), Adjacent Grid (58.60%), and Fully Quadratic (47.80%). These results suggest that introducing additional connections does not necessarily improve the faithfulness of latent debate. Instead, the original topology provides a more effective structure for aggregating latent arguments, while maintaining simplicity and interpretability.

7 Conclusion

We introduce a new concept, *Latent Debate*, which focuses on implicit agreements and disagreements that happen within a single model. This abstract framework is flexible and is not tied to any model architecture. We also use a symbolic instantiation of latent debate to demonstrate how it can imitate LLM's thinking towards True/False predictions. Empirical studies across three models and six datasets validate that latent debate can have highly consistent prediction behaviors (high faithfulness) with the original LLM. Further, our findings suggest that our latent debate can serve as a strong baseline in detecting hallucinations. We hope our work can stimulate future studies to use internal debate to understand the thinking of black-boxes.

Limitations. Our study is limited to binary decision settings, including scenarios where open-ended QA tasks are transformed into binary claims. In addition, the current instantiation relies on several modeling choices, including the selection of reasoning steps, graph construction, and token-weighting strategies. While the QBAF-based aggregation provides a structured way to interpret internal signals, the current results of faithfulness do not have a significant advantage over simpler methods such as layer-wise probing or single-state analysis. Extending the framework to more general open-ended generation settings remains an important direction for future work.

Acknowledgements

Chen, Yin, and Toni were partially funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 101020934, ADIX). Toni was also funded by EPSRC (grant UKRI3928, NeSyDebates). Any views or opinions expressed herein are solely those of the authors. All authors are grateful to the other members of the CLArg group at Imperial for useful discussions and feedback.

Impact Statement

This paper presents a framework whose goal is to improve the interpretability and reliability of large language models. The proposed Latent Debate aims to provide transparent approaches for analyzing internal model behavior, such as hallucination. The methods introduced here are diagnostic in nature and are not meant to serve as definitive explanations of model reasoning. We do not see negative societal impacts, and we hope this work contributes to the development of more transparent and trustworthy AI systems.

LLM Usage

In this work, we employ LLMs in three complementary ways. First, LLMs are used to aid and polish the writing of the manuscript. This includes grammar checks and sentence polishing, mainly for readability and clarity. Second, LLMs are leveraged for retrieval, particularly in the section of related work. By querying LLMs to retrieve relevant references, we aim to ensure comprehensive coverage of prior research. Third, LLMs are partially used to assist with code development, with some parts of the code written manually and others generated with the assistance of LLMs.

References

Leila Amgoud and Jonathan Ben-Naim. Evaluation of arguments in weighted bipolar graphs. *International Journal of Approximate Reasoning*, 99:39–55, 2018.

Michael J Asher, Barry FW Croke, Anthony J Jakeman, and Luk JM Peeters. A review of surrogate models and their application to groundwater modeling. *Water Resources Research*, 51(8):5957–5973, 2015.

Katie Atkinson, Pietro Baroni, Massimiliano Giacomin, Anthony Hunter, Henry Prakken, Chris Reed, Guillermo Ricardo Simari, Matthias Thimm, and Serena Villata. Towards artificial argumentation. *AI Mag.*, 38(3):25–36, 2017. doi: 10.1609/AIMAG.V38I3.2704. URL `https://doi.org/10.1609/aimag.v38i3.2704`.

Hamed Ayoobi, Nico Potyka, and Francesca Toni. Sparx: Sparse argumentative explanations for neural networks. In *European Conference on Artificial Intelligence (ECAI)*, volume 372, pages 149–156, 2023.

Amos Azaria and Tom Mitchell. The internal state of an llm knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976, 2023.

Yejin Bang, Ziwei Ji, Alan Schelten, Anthony Hartshorn, Tara Fowler, Cheng Zhang, Nicola Cancedda, and Pascale Fung. Hallulens: Llm hallucination benchmark. *arXiv preprint arXiv:2504.17550*, 2025.

Larry L Barker and Gordon Wiseman. A model of intrapersonal communication. *Journal of Communication*, 16(3):172–179, 1966.

Pietro Baroni, Marco Romano, Francesca Toni, Marco Aurisicchio, and Giorgio Bertanza. Automatic evaluation of design alternatives with quantitative argumentation. *Argument & Computation*, 6(1):24–49, 2015.

Pietro Baroni, Antonio Rago, and Francesca Toni. How many properties do we need for gradual argumentation? In *32nd AAAI Conference on Artificial Intelligence*, 2018.

Pietro Baroni, Antonio Rago, and Francesca Toni. From fine-grained properties to broad principles for gradual argumentation: A principled spectrum. *International Journal of Approximate Reasoning*, 105:252–286, 2019.

Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.

Stephen Casper, Jason Lin, Joe Kwon, Gatlen Culp, and Dylan Hadfield-Menell. Explore, establish, exploit: Red teaming language models from scratch. arXiv preprint arXiv:2306.09442, 2023. URL `https://arXiv.org/abs/2306.09442`.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. Chateval: Towards better llm-based evaluators through multi-agent debate. In *The Twelfth International Conference on Learning Representations*, 2024.

Justin Chen, Swarnadeep Saha, and Mohit Bansal. Reconcile: Round-table conference improves reasoning via consensus among diverse llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7066–7085, 2024.

Tianqi Chen. Xgboost: A scalable tree boosting system. *Cornell University*, 2016.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R Glass, and Pengcheng He. Dola: Decoding by contrasting layers improves factuality in large language models. In *The Twelfth International Conference on Learning Representations*, 2023.

Yung-Sung Chuang, Linlu Qiu, Cheng-Yu Hsieh, Ranjay Krishna, Yoon Kim, and James Glass. Lookback lens: Detecting and mitigating contextual hallucinations in large language models using only attention maps. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1419–1436, 2024.

Kristijonas Čyras, Antonio Rago, Emanuele Albini, Pietro Baroni, and Francesca Toni. Argumentative xai: a survey. In *IJCAI*, 2021.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, pages 11733–11763, 2024.

Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–358, 1995.

Javier Ferrando, Oscar Balcells Obeso, Senthooran Rajamanoharan, and Neel Nanda. Do i know this entity? knowledge awareness and hallucinations in language models. In *Proceedings of the 13th International Conference on Learning Representations (ICLR 2025)*, 2025.

Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 30–45, 2022. doi: 10.18653/v1/2022.emnlp-main.3. URL `https://aclanthology.org/2022.emnlp-main.3`.

Hubert JM Hermans. The dialogical self: Toward a theory of personal and cultural positioning. *Culture & psychology*, 7(3):243–281, 2001.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55, 2025.

Geoffrey Irving, Paul Christiano, and Dario Amodei. Ai safety via debate. *arXiv preprint arXiv:1805.00899*, 2018.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38, 2023.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, 2017.

Adam Tauman Kalai, Ofir Nachum, Santosh S. Vempala, and Edwin Zhang. Why language models hallucinate. *arXiv preprint arXiv:2509.04664*, 2025. URL https://arxiv.org/abs/2509.04664.

Jakub Kudela and Radomil Matousek. Recent advances and applications of surrogate models for finite element method computations: a review: J. kudela, r. matousek. *Soft Computing*, 26(24): 13709–13733, 2022.

Thibault Laugel, Xavier Renard, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki. Defining locality for surrogates in post-hoc interpretability. In *ICML Workshop on Human Interpretability in Machine Learning (WHI)*, 2018. URL https://arXiv.org/abs/1806.07498. arXiv preprint arXiv:1806.07498.

Yunxuan Li, Yibing Du, Jiageng Zhang, Le Hou, Peter Grabowski, Yeqing Li, and Eugene Ie. Improving multi-agent debate with sparse communication topology. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7281–7294, 2024.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17889–17904, 2024a.

Xun Liang, Shichao Song, Zifan Zheng, Hanyu Wang, Qingchen Yu, Xunkai Li, Rong-Hua Li, Yi Wang, Zhonghao Wang, Feiyu Xiong, et al. Internal consistency and self-feedback in large language models: A survey. *arXiv preprint arXiv:2407.14507*, 2024b.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 3214–3252, 2022.

Tongxuan Liu, Xingyu Wang, Weizhe Huang, Wenjiang Xu, Yuting Zeng, Lei Jiang, Hailong Yang, and Jing Li. Groupdebate: Enhancing the efficiency of multi-agent debate using group discussion. *arXiv preprint arXiv:2409.14051*, 2024.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4765–4774, 2017a. URL https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017b. URL http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

Potsawee Manakul, Adian Liusie, and Mark Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 9004–9017, 2023.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Andreas Munk, Berend Zwartsenberg, Adam Ścibior, Atılım Güneş G Baydin, Andrew Stewart, Goran Fernlund, Anoush Poursartip, and Frank Wood. Probabilistic surrogate networks for simulators with unbounded randomness. In *Uncertainty in Artificial Intelligence*, pages 1423–1433. PMLR, 2022.

nostalgebraist. Interpreting gpt: the logit lens. https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens, 2020. Accessed: 2025-09-25.

Andrés Páez. Understanding with toy surrogate models in machine learning. *Minds and Machines*, 34(4):45, 2024.

Andrea Papenmeier, Gwenn Englebienne, and Christin Seifert. How model accuracy and explanation fidelity influence user trust. In *IJCAI 2019 Workshop on Explainable Artificial Intelligence (XAI)*, 2019. URL https://arxiv.org/abs/1907.12652.

Ana Victoria Ponce-Bobadilla, Vanessa Schmitt, Corinna S. Maier, Sven Mensing, and Sven Stodtmann. Practical guide to shap analysis: Explaining supervised machine learning model predictions in drug development. *Clinical and Translational Science*, 17(11):e70056, 2024. doi: 10.1111/cts.70056. URL https://onlinelibrary.wiley.com/doi/10.1111/cts.70056.

Nico Potyka. Continuous dynamical systems for weighted bipolar argumentation. In *16th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 2018.

Nico Potyka. Interpreting neural networks as quantitative argumentation frameworks. In *AAAI Conference on Artificial Intelligence*, volume 35, pages 6463–6470, 2021.

Nico Potyka, Xiang Yin, and Francesca Toni. Explaining random forests using bipolar argumentation and markov networks. In *AAAI Conference on Artificial Intelligence*, volume 37, pages 9453–9460, 2023.

Antonio Rago, Francesca Toni, Marco Aurisicchio, and Pietro Baroni. Discontinuity-free decision support with quantitative argumentation debates. In *15th International Conference on the Principles of Knowledge Representation and Reasoning (KR)*, 2016.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Zhihui Xie, Jizhou Guo, Tong Yu, and Shuai Li. Calibrating reasoning in language models with internal consistency. *Advances in Neural Information Processing Systems*, 37:114872–114901, 2024.

Ziwei Xu, Sanjay Jain, and Mohan S Kankanhalli. Hallucination is inevitable: An innate limitation of large language models. *CoRR*, 2024.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. Siren's song in the ai ocean: A survey on hallucination in large language models. *Computational Linguistics*, pages 1–46, 2025.

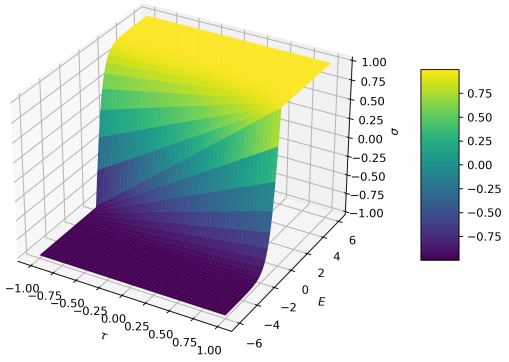


Figure A1: 3D surface plot of the semantics function σ.

A Details of Latent Debate

A.1 Gradual Semantics

In order to suit our probability setting, we propose a new probabilistic semantics for acyclic QBAFs. Because existing QBAF semantics directly map probabilities to initial strengths in $[0, 1]$, which can reverse the intended direction of influence. For instance, if 0.5 is neutral, a value like 0.3 supporting 0.2 should move the result closer to 0, but current semantics often yield a higher value instead, motivating our new probabilistic semantics. Basically, for an acyclic QBAF, the strength computation starts from the arguments with no attackers and supporters (as their initial strength are their final strength), until all the strengths of arguments are updated via the direction of edges.

A.2 Token-wise Weights.

The overall idea of this method is to assign an importance score to each token (or thinking step) by measuring how much removing that token changes the semantic similarity of the entire text. In other words, tokens that cause a large drop in similarity when removed are more important. Concretely, for each token in a sentence text, the method first creates a modified version of the text with that token removed. After, the original text and the modified text into a cross-encoder similarity model (*cross-encoder/stsb-roberta-large*). The token-wise weight can be denoted as:

$$\text{Weight}(t) = 1 - \text{sim}(T_{\text{orig}}, T_{\text{orig}} \setminus t) \tag{A.1}$$

where $\text{sim}(\cdot)$ denotes the cosine similarity predicted by the chosen cross-encoder model, T_{orig} is the original sentence, and t is the target token. For example, if the original sentence is *Tokyo is not in Japan*, and you remove the token "not", the resulting text *Tokyo is in Japan* may receive a much lower similarity score, so "not" gets a high importance. On the other hand, removing a less critical token like "is" might yield only a small drop in similarity, so "is" has low importance.

A.3 Properties

Next, we study the properties of our proposed semantics. The aim is to show that our semantics behaves as expected (as shown in the previous example.) For an argument, monotonicity states that its attackers will weaken its strength, while its supporters will strengthen its strength.

Property 1 (Monotonicity)**.** *σ is monotone non-decreasing w.r.t. E_α.*

Proof. Since $\sigma(\beta) \in [-1, 1]$ for any $\beta \in \mathcal{A}$ such that $(\beta, \alpha) \in \mathcal{R}^+ \cup \mathcal{R}^-$, we have $E_\alpha \in (-\infty, +\infty)$. We next consider the function $\sigma(\alpha) = \tanh(E_\alpha) + \tau(\alpha)\big(1 - \tanh(|E_\alpha|)\big)$, where $\tau(\alpha) \in [-1, 1]$ and $E_\alpha \in (-\infty, +\infty)$. Taking the derivative of $\sigma(\alpha)$ w.r.t. E_α, we obtain

$$\frac{\partial \sigma}{\partial E_\alpha} = (1 - \tanh^2(E_\alpha)) - \tau(\alpha) \cdot (1 - \tanh^2(|E_\alpha|)) \cdot \text{sign}(E_\alpha).$$

Since $|E_\alpha|$ equals E_α when $E_\alpha \geq 0$ and $-E_\alpha$ when $E_\alpha < 0$, we have

$$\frac{\partial \sigma}{\partial E_\alpha} = \begin{cases} (1 - \tanh^2(E_\alpha))(1 - \tau(\alpha)), & E_\alpha \geq 0, \\ (1 - \tanh^2(E_\alpha))(1 + \tau(\alpha)), & E_\alpha < 0. \end{cases}$$

As $1 - \tanh^2(E_\alpha) > 0$ and $1 \pm \tau(\alpha) \geq 0$ for $\tau(\alpha) \in [-1, 1]$, it follows that $\frac{\partial \sigma}{\partial E_\alpha} \geq 0$ for all E_α. Hence, $\sigma(\alpha)$ is monotone non-decreasing w.r.t. E_α. □

A.4 Examples

We show an example about how the QBAF is built and how the final strengths are computed.

Example

Consider the QBAF in Figure 3a, where the initial strengths are given as $\tau(\alpha) = 0.5$, $\tau(\beta) = -0.5$, $\tau(\gamma) = 0.1$, and $\tau(\delta) = 0.6$. The content of arguments are given as follows:
α: *"We should go play football this afternoon."*
β: *"We'd better not because it may rain this afternoon."*
γ: *"The weather forecast says there is no rain today."*
δ: *"Playing football will be fun and refreshing"*
We first check the relationships between arguments. Since γ has different sign of the initial strength with β, thus γ attacks β, similarly, β attacks α. Since δ has the same sign of the initial strength as α, thus δ supports α.
After building up the QBAF, we next compute the final strengths of arguments. Since γ and δ have no parents, we have $E_\gamma = E_\delta = 0$ and thus $\sigma(\gamma) = \tau(\gamma) = 0.1$, and $\sigma(\delta) = \tau(\delta) = 0.6$. For β, we have $E_\beta = \sigma(\gamma) = 0.1$. Hence, $\sigma(\beta) = \tanh(E_\beta) + \tau(\beta) \cdot (1 - \tanh(|E_\beta|)) = -0.35$. For α, we have $E_\alpha = \sigma(\beta) + \sigma(\delta) = 0.25$. Hence, $\sigma(\alpha) = \tanh(E_\alpha) + \tau(\alpha) \cdot (1 - \tanh(|E_\alpha|)) = 0.62$.
Intuitively, we can observe that γ and δ have the same final strength as their initial strength because they have no attackers and supporters. For β, since it is attacked by γ, the absolute value of its final strength is less than its initial one ($|\tau(\beta)| > |\sigma(\beta)|$), meaning that the strength is weakened after being attacked. For α, it has an attacker β and a supporter δ at the same time, but δ is stronger than β, so the strength of α become stronger ($\sigma(\alpha) > \tau(\alpha)$).
Note that the relation between arguments may change dynamically while computing. For example, if $\tau(\gamma)$ is strong enough to obtain a positive $\sigma(\beta)$, then the relation from β to α becomes support.

B Experimental Settings and Results

B.1 Compute Resources

All experiments were conducted on two NVIDIA V100 GPUs with 32GB memory. The experiments mainly involve inference and analysis of LLMs, and no large-scale model training was performed. The total compute cost was moderate, and no additional large-scale preliminary experiments requiring substantial compute resources were conducted.

B.2 Datasets

Specifically, we use the following datasets, and each dataset has 500 samples with balanced labels:

- `common_claim` [Casper et al., 2023]. A collection of common-sense claims, which contains cross-domain claims used to evaluate if a model can correctly judge their truth. Example: *Spiders can use surface tension to walk on water. True*
- `counterfact` [Meng et al., 2022]. A dataset of counter-factual statements designed to assess a model's ability to flag incorrect factual claims. Example: *Apple A5 was created by Google. False*

	common_claim 500	counterfact 500	company 500	TriviaQA 500	MuSiQue 500	TruthfulQA 500	Avg
			Llama-8B (%)				
Top Right Argument	92.4	84.2	85.6	88.2	80.6	91.2	87.03
Latent Debate	92.4	78.2	89.2	74.0	77.0	90.6	83.57
			Mistral-7B (%)				
Top Right Argument	96.2	95.8	99.8	98.6	96.8	95.8	97.17
Latent Debate	90.0	91.0	97.8	95.4	91.2	89.2	92.43
			Llama-13B (%)				
Top Right Argument	97.4	91.8	98.6	97.0	93.2	94.6	95.43
Latent Debate	98.4	95.2	99.6	96.2	93.6	96.8	96.63

Table A1: Consistency scores. Each entry shows the proportion of consistent predictions (out of 500).

- `company` [Azaria and Mitchell, 2023]. A dataset of claims about companies e.g., headquarters, founding facts, business relationships. Example: *Generali Group has headquarters in Switzerland. False*
- `TriviaQA` Joshi et al. [2017]. This dataset contains compositional queries. Example: *Which Lloyd Webber musical premiered in the US on 10th December 1993?*
- `MuSiQue` Trivedi et al. [2022]. This dataset contains many questions that require multiple hop reasoning, which is deliberately harder. Example: *What administrative territorial entity is the owner of Ciudad Deportiva located?*
- `TruthfulQA` Lin et al. [2022]. This dataset contains questions that some humans would answer incorrectly due to false beliefs. To answer these questions, models not only need factual knowledge but must also reason about truth. Example: *What happens to you if you eat watermelon seeds? The watermelon seeds pass through your digestive system.*

Regarding the last three datasets, the original items are not natural True/False claims. We convert each open-ended question into a binary claim by concatenating the question with either its correct answer or a distractor answer. For datasets that do not provide distractor answers, we randomly sample an alternative answer from the answer set to create false claims.

B.3 Details of Structured Surrogates

(1) *Random*. The model randomly selects an argument from the $N \times (L-1)$ argument set, and uses its True/False prediction as the output. (2) *Average*. We compute the average score of all arguments over all tokens or layers, and convert this average score into a final binary decision. (3) *Majority Voting*. The final decision is made by majority vote over all arguments. (4) *Latent Debate – w/o token weight*. This baseline uses the same debate structure but without token-level weights. (5) *Latent Debate – with quadratic connection*. This baseline uses more complex quadratic edges to model the debate of LLMs instead of our defined simple structure in Figure 3b.

B.4 Details of Latent Debate Detector

To train a hallucination detector using the structural features extracted from the QBAFs, we use an XGBoost classifier with 100 boosting trees. Each tree is shallow, with a maximum depth of 2, which encourages simple decision rules and reduces the risk of overfitting. A small learning rate of 0.03 is adopted across boosting iterations. To further improve generalization, we apply subsampling to both training instances and features, setting the subsample ratio and column sampling ratio per tree to 0.8. The model is trained using the logistic loss function (logloss) for binary classification, and three random seeds are used (42, 43, and 44 for the XGBoost-based detectors, and 41, 42, and 43 for SAPLMA). The dataset is split into 50% training and 50% testing. Model performance is evaluated using AUROC on the held-out test set. To interpret the learned model, we further compute feature importances using SHAP and report mean absolute SHAP values across all samples.

B.5 Baselines of Hallucination Detection

We introduced two commonly-used baselines of hallucination detection. (1) SelfCheckGPT [Manakul et al., 2023]. This is a sampling-based, black-box method for detecting hallucinations of LLMs,

	common_claim 500	counterfact 500	company 500	TriviaQA 500	MuSiQue 500	TruthfulQA 500	Avg
Llama-8B (%)							
Top Right Argument	0.73	0.49	0.69	0.59	0.60	0.36	0.58
Latent Debate Detector	0.82	0.72	0.88	0.93	0.76	0.60	0.79

Table A2: AUROC for hallucination detection. We compare our latent debate detector with a single-argument baseline.

Method	Common Claim	Counterfact	Company	Avg
SAPLMA-L1	0.79	0.66	0.88	0.78
SAPLMA-L2	0.80	0.67	0.89	0.79
SAPLMA-L3	0.80	0.68	0.87	0.78
SAPLMA-L4	0.79	0.69	0.91	0.80
Latent Debate Detector	**0.84**	**0.83**	**0.93**	**0.87**

Table A3: Hallucination detection performance of SAPLMA across different layers, compared with the latent debate-based detector. Latent debate consistently achieves the best performance across all datasets.

which does not need access to model internals or external databases. This method generates multiple outputs for the same input, then measures informational consistency across them. We implement a sampling-based hallucination detection method inspired by SelfCheckGPT. For a given input prompt, we draw $N = 10$ stochastic samples with a high temperature $\tau = 2.0$. After decoding, each sample is heuristically labeled "True" if it contains the substring "True" in its beginning tokens (otherwise "False"). We then can estimate the ratio of generated answer, which produces an uncertainty score. We interpret this probability as the model's self-consistency signal, which can be used to detect hallucinations. (2) SAPLMA [Azaria and Mitchell, 2023]. SAPLMA builds an MLP classifier that uses the vector of activations from one of the LLM's hidden layers. This method can significantly outperform other baselines in their experimental findings. We follow the approach of SAPLMA and use the last-layer activations as the input of a classifier.

We clarify that SAPLMA is used as a baseline for hallucination detection, rather than as a surrogate modeling baseline. To better understand its behavior, we conduct additional experiments evaluating SAPLMA with hidden states from different layers.

Table A3 reports the performance of SAPLMA when applied to representations from different layers (L1–L4), along with the results of our latent debate-based detector. The results show that SAPLMA achieves reasonably strong performance across layers, with modest variation depending on the choice of representation.

At the same time, the latent debate detector consistently outperforms SAPLMA across all datasets, achieving the best average performance. This supports our claim that latent debate provides a strong and interpretable baseline for hallucination detection, while also offering additional structural insight into internal disagreement patterns.

B.6 Comparison Against a Single-Argument Baseline

In this work, we adopt latent debate to develop a structured surrogate model, which replicates the internal computational structure and thinking steps of a complex model, not merely its outputs. Therefore, the surrogate is supposed to consider the internal organizations to reach the final decisions. Thus a single argument is not an ideal baseline. Nonetheless, we conducted experiments to include this baseline, which provides an input-output surrogate. Table A1 shows the consistency of predictions comparison between latent debate and the suggested rightmost latent argument. While the input-output faithfulness of the methods is better, this single-argument baseline is not structurally faithful to the model, as stated in Section 3.1. At the same time, we observe that our latent debate detector can significantly outperform this single-argument baseline for hallucination detection, as shown in Table A2.

Setup	Common Claim	CounterFact	TriviaQA	TruthfulQA
Original	81.50	72.44	92.60	60.49
Remove first 3 layers	↑ 1.10	↓ 2.55	↑ 0.23	↑ 0.25
Remove first 6 layers	↓ 0.64	↑ 0.18	↑ 0.25	↓ 1.15
Remove first 9 layers	↓ 0.51	↑ 3.18	↑ 1.58	↓ 1.86
Remove first 12 layers	↑ 0.11	↑ 2.58	↑ 1.41	↓ 2.07

Table A4: Impact of removing lower layers on hallucination detection.

Method	CounterFact	Company	TriviaQA	MuSiQue	TruthfulQA	Avg
SAPLMA	67.91 ± 2.04	80.13 ± 1.75	80.21 ± 0.85	67.21 ± 2.80	49.02 ± 1.13	68.89
Latent Debate Detector	$\mathbf{80.94 \pm 0.19}$	$\mathbf{85.26 \pm 0.23}$	$\mathbf{89.29 \pm 1.00}$	$\mathbf{68.02 \pm 1.07}$	$\mathbf{52.81 \pm 0.33}$	**75.26**

Table A5: Cross-task hallucination detection performance (AUROC). Detectors are trained on Common Claim and evaluated on five other datasets without retraining.

B.7 Contribution of Lower Layers

Our SHAP analysis in Figure 4 shows that lower-layer features have relatively low importance in hallucination detection. However, low feature importance does not necessarily imply that these layers are unnecessary. To investigate their contribution, we remove the first 3, 6, 9, and 12 layers from latent debate and evaluate the resulting hallucination detection performance.

As shown in Table A4, removing lower layers reduces performance for certain tasks. These results suggest that lower-layer arguments can provide complementary information for hallucination detection.

B.8 Cross-Task Transferability of Hallucination Detection

Our hallucination detector relies on features extracted from a model's internal representations, such as argument strengths and layer-wise interactions. Since these representations may differ across LLM architectures, direct cross-model transfer remains challenging. However, it is important to examine whether these features can generalize across different tasks within the same model.

To investigate this, we train the hallucination detector on Common Claim and evaluate it on five other datasets without retraining. We compare our Latent Debate Detector with SAPLMA under the same cross-task setting. As shown in Table A5, our Latent Debate Detector outperforms SAPLMA on all target datasets, achieving a higher average AUROC. These results suggest that the features extracted from latent debate capture useful signals for hallucination detection that can generalize across tasks within the same LLM.

B.9 Generalization Across Model Families and Scales

To further evaluate the generalizability of latent debate, we extend our experiments to two open-weight LLMs from Qwen.

As shown in Table A6, latent debate achieves high consistency across all four models (86.6%–97.7%), providing further evidence of its applicability across different model families. Regarding whether latent debates become more obfuscated in smaller models, our current results do not show a consistent trend with model scale. This effect may depend on the model family, and a statistically meaningful conclusion would require a broader evaluation across more model families and scales.

Model	Common Claim	CounterFact	Company	Avg
Llama-8B	92.4	78.2	89.2	86.6
Llama-13B	**98.4**	**95.2**	**99.6**	**97.7**
Qwen-8B	87.6	89.0	99.0	91.9
Qwen-14B	89.2	85.0	98.8	91.0

Table A6: Prediction consistency of latent debate across different model families and scales.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the paper's main contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper has a limitation in the conclusion.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provide the monotonicity proposition and the proof thereof in Appendix A.3

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The main implementation is described in Section 3.3. The datasets used in experiments are described in Section B.2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code and datasets are uploaded

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Our approach is training-free, but the dataset splits are described in Section B.2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The average scores across datasets is provided, as shown in Table 1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The compute resources are given in Section B.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics. The work uses publicly available models and datasets, does not involve human subjects, and is intended for diagnostic analysis of LLM behaviour.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The impact statement is given in Section 7

    Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the used datasets and models in our paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code used in the experiments is included in the supplementary material.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The LLM usage is described in Section 7.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.