

# Leveraging NTPs for Efficient Hallucination Detection in VLMs

Ofir Azachi<sup>\*1</sup>, Kfir Eliyahu<sup>\*1</sup>, Eyal El Ani<sup>\*1</sup>, Rom Himmelstein<sup>\*1</sup>,  
Roi Reichart<sup>1</sup>, Yuval Pinter<sup>2</sup>, Nitay Calderon<sup>1</sup>

<sup>1</sup>Department of Data and Decision Science, Technion - Israel Institute of Technology,

<sup>2</sup>Department of Computer Science, Ben-Gurion University of the Negev

Correspondence: romh@campus.technion.ac.il.

## Abstract

Hallucinations of vision-language models (VLMs), which are misalignments between visual content and generated text, undermine the reliability of VLMs. One common approach for detecting them employs the same VLM, or a different one, to assess generated outputs. This process is computationally intensive and increases model latency. In this paper, we explore an efficient on-the-fly method for hallucination detection by training traditional ML models over signals based on the VLM’s next-token probabilities (NTPs). NTPs provide a direct quantification of model uncertainty. We hypothesize that high uncertainty (i.e., a low NTP value) is strongly associated with hallucinations. To test this, we introduce a dataset of 1,400 human-annotated statements derived from VLM-generated content, each labeled as hallucinated or not, and use it to test our NTP-based lightweight method. Our results demonstrate that NTP-based features are valuable predictors of hallucinations, enabling fast and simple ML models to achieve performance comparable to that of strong VLMs. Furthermore, augmenting these NTPs with linguistic NTPs, computed by feeding only the generated text back into the VLM, enhances hallucination detection performance. Finally, integrating hallucination prediction scores from VLMs into the NTP-based models led to better performance than using either VLMs or NTPs alone. We hope this study paves the way for simple, lightweight solutions that enhance the reliability of VLMs. All data is publicly available at [?](#).

## 1 Introduction

*Vision-language models (VLMs)* have emerged as powerful tools capable of handling tasks involving visual and textual inputs. These models enable applications such as visual question answering (VQA; Li et al., 2019), and text-to-image generation (Radford et al., 2021; Zhao et al., 2024b). However,

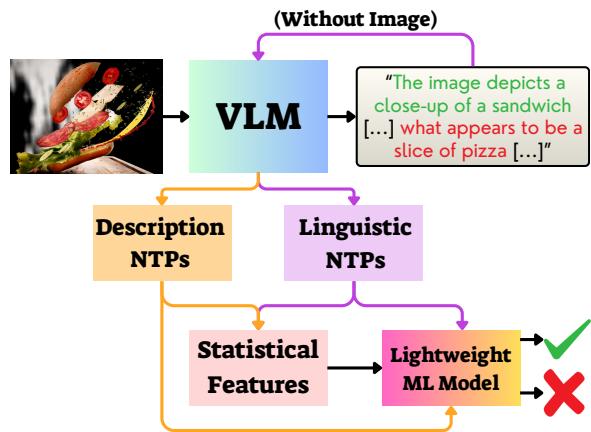


Figure 1: **Illustration of our method:** Linguistic NTPs are extracted during the VLM’s text generation process. Description NTPs require an additional forward pass using only the generated text. Statistical features are then computed from the NTPs, and a lightweight traditional ML model uses these features to detect hallucinations.

as these models become more widely used, concerns about *hallucinations*, errors or misleading outputs generated by the model, have become more prominent. Unlike humans, who are less likely to describe non-existent objects, misjudge colors, or miscount elements, these errors are more likely to appear in machine-generated content. Gunjal et al. (2024) found that even state-of-the-art VLMs frequently generate non-existent objects.

Currently, the primary method for detecting hallucinations involves using VLMs as hallucination predictors, either by asking a model to identify hallucinations in its own generated output or in others’ (Li et al., 2024). This approach has demonstrated success both in generative LLMs (Quevedo et al., 2024) and generative VLMs (Chen et al., 2024). However, these predictor VLMs exhibit two main weaknesses: First, they often require performing extensive computations, making them both computationally expensive and time-consuming, especially when multiple calls are needed to verify each sentence or clause in the generated content.

<sup>\*</sup>Equal contribution.

Second, they lack explainability and interpretability (Zhao et al., 2024a).

*Large language models (LLMs)* generate responses by sampling tokens from a learned probability distribution over the next token, conditioned on the input context. This auto-regressive generation process resembles human language production, where likely words are uttered based on contextual understanding and prior knowledge (Goldstein et al., 2022). Lu et al. (2021) found that, in humans, uncertainty plays a key role in the propagation of misinformation. Inspired by this, we hypothesize that *next-token probabilities (NTPs)* produced by VLMs may similarly encode uncertainty, and thus can serve as useful signals for hallucination detection. Indeed, prior work suggests that high uncertainty, reflected by low NTPs, is a strong indicator of hallucinations and related errors (Farquhar et al., 2024; Quevedo et al., 2024; Li et al., 2024).

We investigate the role of NTPs in detecting hallucinations in VLMs. Rather than relying on predictor VLMs, we propose leveraging the NTPs produced during generation to enable fast, real-time hallucination detection. Our goal is to design an effective approach for leveraging NTP-based features to predict hallucinations using fast, lightweight traditional machine learning (ML) models, such as Logistic Regression, Support Vector Machine, and XGBoost. As illustrated in Figure 1, we compare approaches that use raw NTPs directly from the VLMs (*Description NTPs*) with those that rely on statistical features derived from the NTPs. We explore integration of NTPs with VLM predictor outputs, and propose a method for neutralizing linguistic biases embedded within them using *Linguistic NTPs* resulting from reprocessing the generated text through the same VLM after omitting the visual input. Throughout, we assume that higher uncertainty, operationalized as lower next-token probabilities or higher entropy, correlates with hallucination risk (Farquhar et al., 2024), and we design our features to capture this signal.

A growing body of research shows that VLMs often rely heavily on linguistic priors (Zhu et al., 2024; Guan et al., 2024; eun Cho and Maeng, 2025; Wang et al., 2024), and may even prioritize them over conflicting visual evidence (Luo et al., 2024; Wu et al., 2024). These findings suggest that hallucinations in VLMs may stem, at least in part, from biases in their language modeling components, rather than solely from limitations in visual understanding. Based on these insights, we intro-

duce a novel dataset specifically curated to examine the relationship between NTPs and hallucinations in VLMs. We believe this dataset will serve as a valuable resource for future research in this area.<sup>1</sup> Using this dataset, we evaluate the effectiveness of NTPs generated by *LLaVA-1.5* and *LLaVA-1.6* (Liu et al., 2024) for hallucination detection. As baselines, we include predictions from both *LLaVA* and *PaliGemma* (Beyer et al., 2024), and also use these predictions as additional input features to traditional ML models.

Our experiments reveal that statistical features derived from NTPs outperform raw NTP features across all models, making them a more effective and reliable signal for hallucination detection. These statistical features alone come close to matching the performance of VLM predictors while offering gains in efficiency, allowing for on-the-fly hallucination detection. While incorporating *Linguistic NTPs* offers only modest gains for statistical features, neutralization strategies such as element-wise subtraction of raw *Description* and *Linguistic* NTPs provide further evidence of the role of linguistic biases in hallucination generation. Finally, we find that augmenting VLM predictor outputs with NTP features yields consistent improvements, demonstrating that these signals are complementary and result in the strongest hallucination detection approach.

## 2 Related work

**Defining hallucinations.** The term *hallucinations* lacks a universal definition across different fields but, in general, describes instances where a model produces content that is disconnected from its input or from reality (Maleki et al., 2024). In NLP, this term typically refers to outputs that fail to accurately reflect real-world facts (Xu et al., 2024). The notion extends to other areas as well; for example, in medical imaging, deep learning techniques can create images that appear realistic, but contain fabricated structures, potentially misleading diagnostic efforts (Bhadra et al., 2021). Identifying hallucinations is critical because inaccuracies not only diminish user trust but also present significant risks across diverse domains (Benkirane et al., 2024; Tang et al., 2025), including low-resource language settings (Benkirane et al., 2024), legal contexts (Magesh et al., 2024), information retrieval (Faggioli et al., 2023), healthcare and au-

<sup>1</sup>Code and dataset will be released upon publication.

tonomous driving (Leng et al., 2024; Gunjal et al., 2024). Consequently, robust hallucination detection is essential to mitigate these challenges and safeguard the reliability of AI-generated content.

**Techniques for hallucination detection.** Various methods have been proposed to automatically detect hallucinated outputs. One common approach involves analyzing the model’s output probability distributions, where segments with low confidence, characterized by high entropy or significantly reduced token probabilities, are reliably flagged as hallucinations (Li et al., 2024; Ma et al., 2025; Guerreiro et al., 2022; Quevedo et al., 2024; Farquhar et al., 2024; Simhi et al., 2025). In contrast to these internal indicators, other methods deploy external models such as dedicated VLMs (Chen et al., 2024) or LLMs (Quevedo et al., 2024) to assess whether hallucinations are present in the generated content. Although this external verification yields promising results, it tends to be significantly more resource-intensive than relying solely on internal signals, and lacks explainability (Sarkar, 2024; Zhao et al., 2024a).

**Linguistic biases and their impact on VLMs.** A significant source of hallucinations in both VLMs and LLMs is their overdependence on linguistic priors and biases. Research indicates that large VLMs often generate plausible-sounding descriptions based on statistical patterns learned during training (e.g., "blue sky"), rather than by accurately anchoring every detail to the visual content (Zhu et al., 2024; Guan et al., 2024). This can result in errors such as attributing objects or attributes to a scene that, while contextually expected, are actually absent—a phenomenon commonly known as *object hallucination* in image captioning and VQA systems (Leng et al., 2024). In many cases, the language generation component can dominate the visual signal, with models relying solely on textual context even when it contradicts the visual evidence (Luo et al., 2024; Wu et al., 2024). Consequently, recent research focuses on minimizing these linguistic biases to reduce hallucinations originating from the multimodal interaction, for instance, by encouraging the model to more closely attend to the image during the decoding process (Zhu et al., 2024; Leng et al., 2024).

### 3 Method

**Problem definition.** A *probe* is a statement derived from a VLM-generated description of an im-

age. Each probe can either be truthful or contain a hallucination. For example, the probe ‘*There is a handbag.*’ from Figure 2 corresponds to the generated sentence ‘*There is also a handbag visible in the scene.*’ We define *hallucinations* as any textual information produced by the VLM that does not accurately reflect the visual content of the image. In particular, we consider the following as hallucinations: objects falsely perceived as present, incorrect object attributes (such as color or size), and misinterpretations of relationships within the scene. Our goal is to predict whether a probe contains a hallucination or not.

#### 3.1 Predicting Hallucinations

We employ two complementary approaches to predict whether a probe contains a hallucination. The first approach employs a predictor VLM (e.g., *LLaVA-1.5*, *LLaVA-1.6* or *PaliGemma*) which process the image using the prompt:

“According to the image, is the following sentence correct? {PROBE}. Answer only with Yes OR No.”

Here, {PROBE} represents a probe derived from the VLM-generated description of the image. We denote the probability that the probe is correct, as estimated using the NTP of the predictor VLM, by:

$$\frac{\mathbb{P}(\text{Yes})}{\mathbb{P}(\text{Yes}) + \mathbb{P}(\text{No})}$$

The main drawback of this approach is the reliance on a predictor VLM, which can be computationally expensive. In real-time applications, where we aim to verify that the content generated by the VLM is correct, this approach substantially increases latency, as each statement is verified separately. To address this, we propose an alternative approach that employs fast and lightweight traditional machine learning models (we use the term *traditional ML models* in the remaining of the text), such as Logistic Regression (LR), Support Vector Machine (SVM), and XGBoost. These models are trained to predict whether a probe is correct based on features derived from the NTPs of the VLM-generated description. Since these NTPs are by-products of the generation process, the models can assess the correctness of the generated content on the fly (i.e., during generation). In the following subsection, we describe these NTP-based features.

### 3.2 Next Token Probabilities (NTPs)

We present two types of NTPs that are used as features for the traditional ML models.

**Description NTPs.** When a VLM generates a response, it does so token by token, estimating a probability distribution over all possible tokens at each step. We hypothesize that these *Description NTPs* encode valuable information about the model’s certainty in its generated response and, therefore, may be beneficial for hallucination detection. Since *Description NTPs* can be obtained on the fly, they serve as our primary focus.

**Linguistic NTPs.** Our manual analysis of *Description NTPs* revealed recurring probability patterns that suggest linguistic influences beyond visual content. We hypothesize that these patterns arise from inherent linguistic biases in the model. Following the methodology of Liu et al. (2023); Srivastava et al. (2023), who demonstrated that linguistic effects in the generated text could be captured by feeding the text back into the same language model that produced it, we reinserted the VLM-generated text into its corresponding language model, this time without an instructional prompt or image. We term the extracted probabilities as *Linguistic NTPs*. Our motivation is to augment the *Description NTPs* with *Linguistic NTPs*, which help disentangle language-driven biases, such as syntactic or grammatical priors, from visually grounded signals, thereby improving the detection of hallucinated content.

To quantify the relationship between *Description NTPs* and *Linguistic NTPs*, we computed Spearman’s correlation between the two probability series for each probe. The average correlation across all probes was 0.744, reinforcing our hypothesis that the two types of NTPs are inherently linked. Consequently, we examine the potential of *Description NTPs* both as standalone features and in combination with *Linguistic NTPs*.

### 3.3 Next Token Probabilities as Features

We next describe how the NTPs are used in practice as features for traditional ML models. *Description NTPs* are extracted on the fly during text generation. For each probe, we consider only the NTPs corresponding to the span of generated text associated with that probe, typically a sentence or clause, though not necessarily limited to that. *Linguistic NTPs*, on the other hand, are extracted separately, either after the full description has been generated

or after the span corresponding to each probe (e.g., after each sentence). The result is one (or two) matrices with a shape equal to the number of generated tokens in the span by the vocabulary size. In our main setup, we use only the probability values assigned by the VLM to the actually generated tokens, resulting in a dense vector of length equal to the number of tokens in the span.

Naturally, using these vectors as raw features presents several challenges. First, spans may vary in length, whereas traditional ML models require a fixed number of input features. Second, there are multiple ways to combine the *Description* and *Linguistic NTPs*. Third, the sequences can be long, which motivates aggregation and feature engineering. To address the challenge of varying sequence lengths, each sequence of NTPs (either *Description* or *Linguistic*) is zero-padded to match the length of the longest sequence in the dataset, which contains 42 tokens. To explore how to best combine the two types of NTPs, the following aggregation techniques were applied:

- **Only Description NTPs:** Use only the *Description NTPs* as input features.
- **Only Linguistic NTPs:** Use only the *Linguistic NTPs* as input features.
- **Concatenation:** Concatenate the *Description* and *Linguistic NTPs* sequences, resulting in a combined input of 84 features.
- **Element-wise subtraction:** Subtract the *Linguistic NTPs* from the *Description NTPs* token by token.
- **Element-wise division:** Divide the *Description NTPs* by the *Linguistic NTPs* token by token using:

$$t_i^{\text{div}} = \frac{t_i^{\text{Desc}}}{1 + t_i^{\text{Ling}}} \in [0, 1],$$

where  $t_i$  represents the corresponding NTP value of the  $i$ -th generated token.

While *raw NTP* values provide direct probabilistic information, they may not capture higher-level patterns or summarise statistics that might be useful for hallucination detection. To enrich the feature space, we also engineer *statistical features*:

- Mean of the generated-token NTPs.

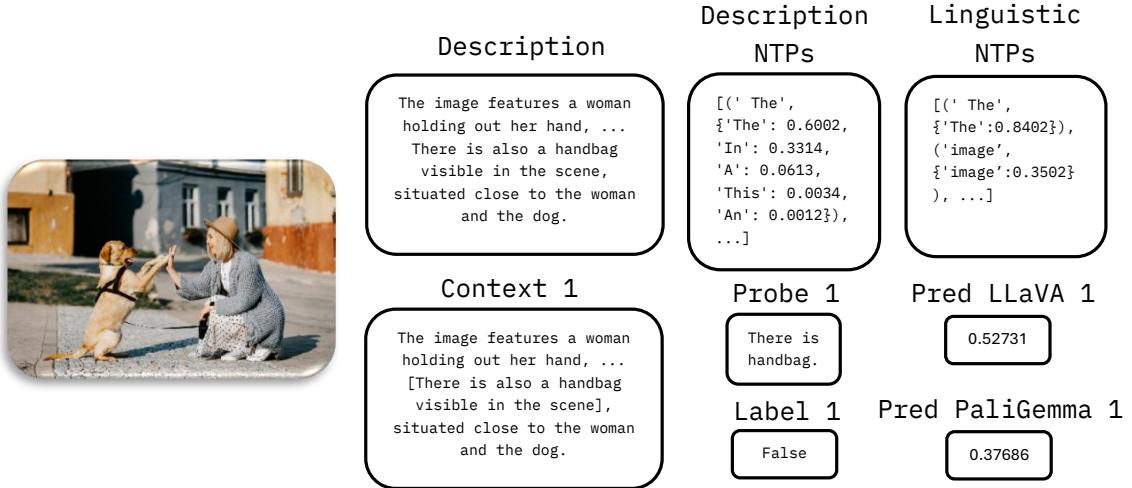


Figure 2: An example for the data features.

- Standard deviation of the NTPs.
- Mean of the logarithm and exponent of the NTPs ( $\log(\mathbb{P})$  and  $\exp(\mathbb{P})$ ).
- The top- $k$  dominant frequencies (excluding DC) from the Discrete Fourier Transform of real-valued NTPs, where  $k$  is a hyper-parameter ranging from 0 to 5 (0 serving as the control).

If both types of NTPs are available, we extract the following additional features:

- Mean of the element-wise product between the *Description* and *Linguistic NTPs*.
- Minimum between (i) the mean of the element-wise ratio of *Linguistic NTPs* to *Description NTPs*, and (ii) the mean of the element-wise ratio of *Description NTPs* to *Linguistic NTPs*.

## 4 Hallucination Detection Dataset

Our dataset consists of 350 images, sourced from Pixabay<sup>2</sup> and iStock.<sup>3</sup> For each image, a *LLaVA* model was prompted with the instruction: “Please provide a thorough description of this image”. The generated descriptions were manually reviewed, and only those containing at least one hallucination were retained. This procedure yielded 200 examples using *LLaVA-1.6* and 150 examples using *LLaVA-1.5*. From each VLM-generated description with at least one hallucination, four probes were

extracted, ensuring that at least one probe per description contained a hallucination. In total, the dataset comprises 1,400 probes, of which 42.9% are labeled as hallucinated. The annotation process was conducted by a group of seven undergraduate students (six males and one female), with ages ranging from 21 to 28 years. Each data sample includes the following features, with  $i \in [4]$ :

**Description:** The generated description by the *LLaVA* model. **Description NTPs:** The NTPs of the *LLaVA* generated tokens.<sup>4</sup> **Linguistic NTPs:** A sequence of probabilities, where each value represents the likelihood of a generated token when the description is processed without the image input. **Probe(i):** A statement written by the annotators that can be derived from the respective Description. At least one probe among the four contains a hallucination. **Label(i):** A binary label (True/False) that was manually assigned to decide the validity of *Probe(i)*. **Context(i):** A markup of the part of the generated description that *Probe(i)* refers to from the respective Description. **LLaVA Pred(i):** The *LLaVA* VLM estimation of *Probe(i)*’s correctness, as described in §3.1. **PaliGemma Pred(i):** The *PaliGemma* VLM estimation of *Probe(i)*’s correctness, see §3.1.

**Figure 2** illustrates the features described above. An example of the data collection pipeline is provided in [Appendix A](#). A detailed analysis of the *Description NTPs* and *Linguistic NTPs* is presented in [Appendix B](#), along with supporting evidence for their potential usefulness as input features to the

<sup>2</sup><https://pixabay.com/>

<sup>3</sup><https://www.istockphoto.com/>

<sup>4</sup>We also saved non-generated tokens with probabilities above a set minimum threshold of 1e-3.

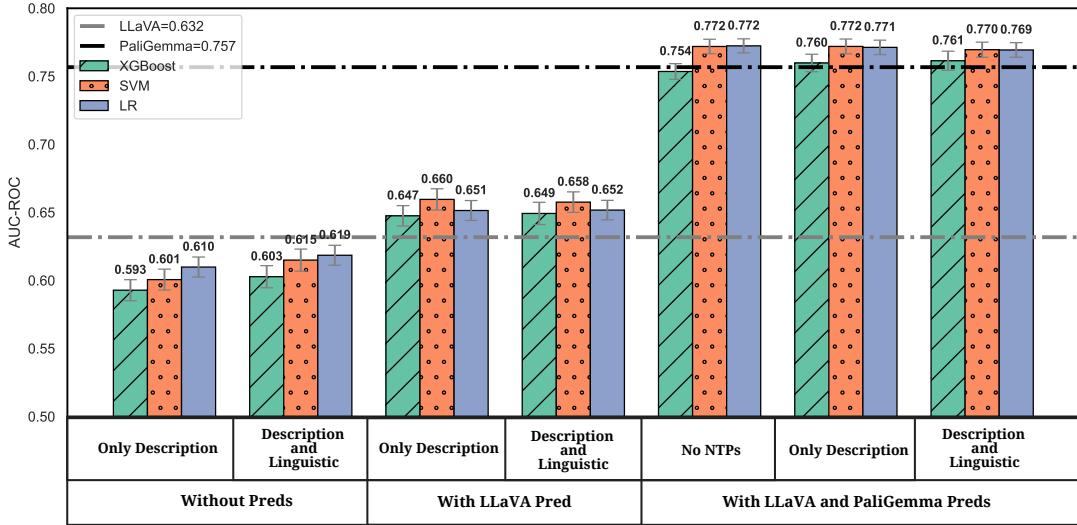


Figure 3: AUC-ROC performance of traditional ML models using statistical features of NTPs and various **Pred** features. Each bar group corresponds to a specific feature combination, while the dashed lines denote the *LLaVA* and *PaliGemma* baselines. Error bars indicate 95% confidence intervals.

models introduced in the following section.

## 5 Experimental Setup

**VLM Predictors** To evaluate the effectiveness of probe-based hallucination detection, we employ two VLM predictors. The first is a *LLaVA*-based predictor,<sup>5</sup> corresponding to the same VLM that generated the image description. The rationale is to compare the performance of traditional ML models that rely on the VLM’s NTPs with that of using the same VLM for self-verification of its own generated content. The second VLM predictor is an external model, *PaliGemma*.<sup>6</sup> Naturally, using an external VLM also imposes additional computational and memory overhead.

**Traditional ML models** We experiment with three traditional ML models: Logistic Regression (LR), Support Vector Machine (SVM), and XG-Boost. We employ two sets of features, as described in §3.3: (i) raw NTPs, using either *Description NTPs*, *Linguistic NTPs*, or a combination of both; and (ii) statistical features extracted from the NTPs. Each model is trained on 1000 examples (71.4% of the full dataset), with an additional 200 examples (14.3%) used for validation (for hyperparameter tuning), and evaluated on a test set of 200 examples (14.3%). To ensure the robustness of our results, the reported results reflect the average performance over 100 random splits.

<sup>5</sup>[huggingface.co/llava-hf/llava-1.5-7b-hf](https://huggingface.co/llava-hf/llava-1.5-7b-hf); [huggingface.co/llava-hf/llava-v1.6-mistral-7b-hf](https://huggingface.co/llava-hf/llava-v1.6-mistral-7b-hf)

<sup>6</sup>[huggingface.co/google/PaliGemma-3b-pt-224](https://huggingface.co/google/PaliGemma-3b-pt-224)

## Combining NTP-based features with VLM predictors

We investigate whether combining the **Pred** feature obtained from a predictor VLM (*LLaVA* or *PaliGemma*) with the NTP-based features improves detection. Accordingly, the input of traditional ML models is augmented with one or both predictor outputs. While this approach introduces additional computational cost due to extra VLM inference, it allows us to assess whether combining fast NTP-based features with direct VLM predictions offers complementary benefits.

**Hyperparameter tuning.** We perform hyperparameter tuning for each train-validation split to ensure optimal model performance. The tuning process aims to maximize the Area Under the ROC Curve (AUC-ROC) on the validation set. Given the variability in input representations and model configurations, the specific hyperparameter ranges for each setting are provided in Appendix C.

## 6 Results

We present the key results for the statistical NTP-based features in Figure 3 and the complete results in Table 1 in Appendix D. Results for the raw NTP-based features are shown in Figure 4. Below, we discuss our main findings.

**Statistical features of NTPs can be competitive to VLM predictions** We begin by comparing the performance of statistical features derived from *Description NTPs* with that of the **Pred** feature of *LLaVA*. This comparison is natural, as the NTPs are extracted from the same model used for predic-

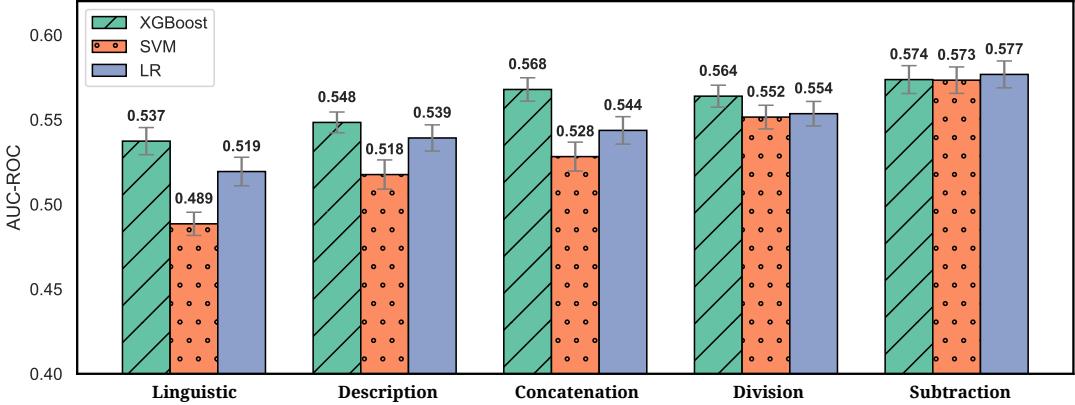


Figure 4: AUC-ROC performance of ML models using different aggregation techniques of **raw** NTP features.

tion. As shown in [Figure 3](#), *LLaVA Pred* (dashed line) achieves slightly better performance than the statistical features extracted from the *Description NTPs* (three leftmost bars), with the ROC AUC difference for LR being 0.013. Notice, however, that using *LLaVA Pred* requires an additional forward pass of the VLM for every probe (and a single generated text can contain several probes). In contrast, *Description NTP* features are obtained on-the-fly during generation and only require inference from a lightweight traditional ML model. Our results suggest that using *Description NTPs* offers a compelling trade-off between performance and efficiency, making it a practical option for real-time applications where latency is paramount.

**Linguistic NTPs provide a modest improvement** We next examine whether incorporating statistical features from *Linguistic NTPs* improves the performance of traditional ML models. Although using *Linguistic NTPs* introduces additional computational costs compared to using only *Description NTPs*, this cost remains relatively low. *Linguistic NTPs* can be computed with a single forward pass of the language model after the text is generated, in contrast to the multiple VLM calls required for predictor VLMs (one for every probe). As shown in [Figure 3](#), comparing the second group of bars (bars 4–6: *Description + Linguistic NTPs*) to the first group (bars 1–3: *Description NTPs* only) reveals a consistent, albeit modest, performance gain across all ML models. The improvement in ROC AUC is approximately 0.01 and is not statistically significant, as indicated by overlapping confidence intervals. While these results suggest a positive effect from including *Linguistic NTPs*, the benefit is limited, and further investigation is needed to understand their full potential.

### Statistical features of NTPs enhance VLM predictor performance.

So far, we have shown that NTP-based features offer a fast and lightweight solution for hallucination detection, although they moderately underperform compared to using the same VLM as a predictor. We now investigate whether combining both approaches can yield further improvements. As shown in [Figure 3](#) (bars 7–9), augmenting the *Pred* feature with statistical features from *Description NTPs* consistently improves performance across all traditional ML models. This indicates that NTPs alone can enhance hallucination detection when used alongside a predictor VLM. Specifically, the ROC AUC improvements over using *LLaVA Pred* alone are 0.015, 0.028, 0.019 for XGBoost, SVM, and LR, respectively. We do not observe any further improvement regarding combining *Linguistic NTP*-based features (see bars 10–12).

In addition to *LLaVA*, we evaluate *PaliGemma* as an alternative VLM predictor. While using an external predictor that differs from the generator introduces additional memory overhead, *PaliGemma Pred* achieves substantially better performance than *LLaVA Pred* (ROC AUC of 0.757 vs. 0.632). We further assess whether combining both predictors improves performance. As shown in [Figure 3](#) (bars 13–15), using both *Pred* features as input to SVM and LR yields an improvement over using *PaliGemma Pred* alone, with an ROC AUC gain of 0.015. Finally, we examine whether adding statistical NTP-based features provides additional benefit in this combined predictor setup. While no improvement is observed for SVM and LR, XGBoost does show a performance gain when NTP features are included.

### Subtraction is the best aggregation of raw

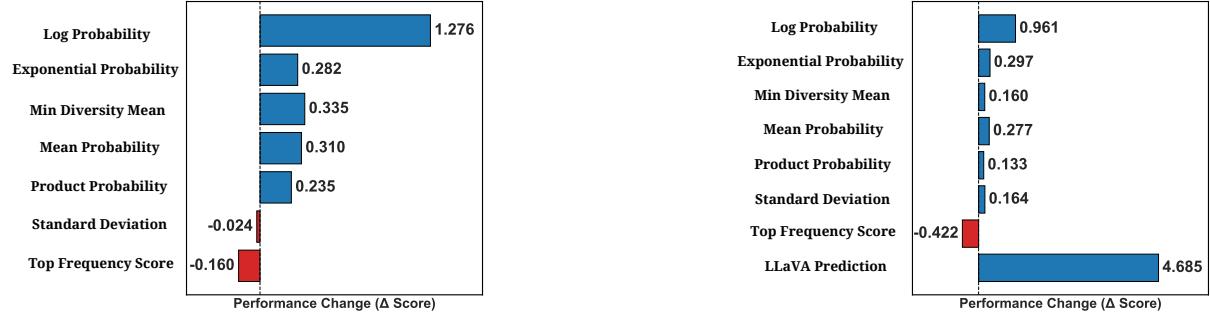


Figure 5: Leave-one-out ablation study on our features. Excluding (left) and including (right) *LLaVA* predictions.

**NTPs** Although our primary analysis emphasizes statistical features due to their superior performance compared to raw NTPs (compare bars 1–6 in Figure 3 to the bars in Figure 4), we also explore raw NTP-based features, as they may offer additional insights for future work. In particular, we investigate how combining raw *Description* and *Linguistic NTPs* affects model performance. As shown in Figure 4, aggregation methods that aim to neutralize the influence of linguistic biases, such as element-wise subtraction or division of *Description NTPs* by *Linguistic NTPs*, consistently outperform simple concatenation across most ML models. Among these, subtraction yields the highest performance. This suggests that underlying linguistic patterns in the model shape the generated descriptions, and that these influences can be partially corrected through neutralization-based aggregation.

### 6.1 Feature Importance Analysis

We now assess the contribution of individual statistical features extracted from both *Description* and *Linguistic NTPs*. We consider multiple configurations, including models with and without the **LLaVA Pred** feature. To evaluate feature importance, we conduct a leave-one-feature-out analysis: for each feature, we measure the change in performance ( $\Delta$ ) as the difference in AUC-ROC between the full model (with all features) and the model with it removed. Results are presented in Figure 5.

Unsurprisingly, the **LLaVA Pred** feature is the most influential, providing a significantly larger performance gain than any of the NTP-based features. This aligns with its higher computational cost and the richer information it encapsulates from a full VLM inference pass. Among the NTP-based statistical features, we find that transformations of the probabilities, specifically, log-probabilities and exponentiated probabilities, are more informative than raw probabilities. This likely stems from the

nature of the softmax distribution over generated tokens. These raw values offer limited variance and may obscure fine-grained differences in uncertainty. In contrast, applying logarithmic or exponential transformations expands the range, making subtle distinctions more detectable to the model. Finally, time series features derived from the Discrete Fourier Transform (e.g., dominant frequencies) perform the worst. In some cases, including them even degrades model performance relative to the baseline, suggesting they may introduce noise or redundancy rather than useful signal.

## 7 Conclusion

In this paper, we explore the potential of leveraging uncertainty-related features to improve hallucination detection in text generated by VLMs. Specifically, we use NTPs extracted from VLMs in combination with traditional, efficient ML models to enhance detection performance while remaining computationally lightweight. Our results show that statistical features derived from *Description NTPs* provide a lightweight and effective alternative to using VLM predictors. While *Linguistic NTPs* offer performance gains when **Pred** features are unavailable, they contribute little when such features are present, often making their additional computational cost unjustified. Finally, we find that combining NTP-based features with **Pred** scores leads to consistently improved detection performance, demonstrating their complementary nature.

We hope this work serves as a valuable resource for advancing the understanding and practical use of NTPs in hallucination detection. Our findings point to two promising directions for future research: (1) developing efficient models of hallucination detection to support response refinement or the expression of uncertainty, and (2) further investigating the relationship between *Description*

and *Linguistic NTPs*, whose integration may prove valuable beyond hallucination detection.

## References

- Kenza Benkirane, Laura Gongas, Shahar Pelles, Naomi Fuchs, Joshua Darmon, Pontus Stenetorp, David Ifeoluwa Adelani, and Eduardo Sánchez. 2024. Machine translation hallucination detection for low and high resource languages using large language models. *arXiv preprint arXiv:2407.16470*.
- Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey A. Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, and 16 others. 2024. **Paligemma: A versatile 3b vlm for transfer**. *CoRR*, abs/2407.07726.
- Sayantan Bhadra, Varun A Kelkar, Frank J Brooks, and Mark A Anastasio. 2021. On hallucinations in tomographic image reconstruction. *IEEE transactions on medical imaging*, 40(11):3249–3260.
- X. Chen, C. Wang, Y. Xue, N. Zhang, X. Yang, Q. Li, and H. Chen. 2024. Unified hallucination detection for multimodal large language models. *arXiv preprint*, arXiv:2402.03190.
- Ye eun Cho and Yunho Maeng. 2025. **The influence of visual and linguistic cues on ignorance inference in vision-language models**. *arXiv e-prints*.
- Guglielmo Faggioli, Laura Dietz, Charles LA Clarke, Gianluca Demartini, Matthias Hagen, Claudia Hauff, Noriko Kando, Evangelos Kanoulas, Martin Potthast, Benno Stein, and 1 others. 2023. Perspectives on large language models for relevance judgment. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 39–50.
- Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.
- Ariel Goldstein, Zaid Zada, Eliav Buchnik, Mariano Schain, Amy Price, Bobbi Aubrey, Samuel A Nasstase, Amir Feder, Dotan Emanuel, Alon Cohen, and 1 others. 2022. Shared computational principles for language processing in humans and deep language models. *Nature neuroscience*, 25(3):369–380.
- Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, and 1 others. 2024. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14375–14385.
- Nuno M Guerreiro, Elena Voita, and André FT Martins. 2022. Looking for a needle in a haystack: A comprehensive study of hallucinations in neural machine translation. *arXiv preprint arXiv:2208.05309*.
- Anisha Gunjal, Jihan Yin, and Erhan Bas. 2024. Detecting and preventing hallucinations in large vision language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18135–18143.
- Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. 2024. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13872–13882.
- Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. **Visualbert: A simple and performant baseline for vision and language**. *Preprint*, arXiv:1908.03557.
- Qing Li, Chenyang Lyu, Jiahui Geng, Derui Zhu, Maxim Panov, and Fakhri Karray. 2024. Reference-free hallucination detection for large vision-language models. *arXiv preprint*, arXiv:2408.05767.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. Visual instruction tuning. *Advances in neural information processing systems*, 36.
- Tong Liu, Iza Škrjanec, and Vera Demberg. 2023. **Temperature-scaling surprisal estimates improve fit to human reading times - but does it do so for the "right reasons"?** In *Annual Meeting of the Association for Computational Linguistics*.
- Jiahui Lu, Meishan Zhang, Yan Zheng, and Qiyu Li. 2021. Communication of uncertainty about preliminary evidence and the spread of its inferred misinformation during the covid-19 pandemic—a weibo case study. *International Journal of Environmental Research and Public Health*, 18(22):11933.
- Tiange Luo, Ang Cao, Gunhee Lee, Justin Johnson, and Honglak Lee. 2024. **vvlm: Exploring visual reasoning in vlms against language priors**. *OpenReview ICqNxBGPp5*.
- Huan Ma, Jingdong Chen, Guangyu Wang, and Changqing Zhang. 2025. Estimating llm uncertainty with logits. *arXiv preprint arXiv:2502.00290*.
- Varun Magesh, Faiz Surani, Matthew Dahl, Mirac Suzgun, Christopher D Manning, and Daniel E Ho. 2024. Hallucination-free? assessing the reliability of leading ai legal research tools. *arXiv preprint arXiv:2405.20362*.
- Negar Maleki, Balaji Padmanabhan, and Kaushik Dutta. 2024. Ai hallucinations: a misnomer worth clarifying. In *2024 IEEE conference on artificial intelligence (CAI)*, pages 133–138. IEEE.

E. Quevedo, J. Yero, R. Koerner, P. Rivas, and T. Cerny. 2024. Detecting hallucinations in large language model generation: A token probability approach. *arXiv preprint*, arXiv:2405.19648.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. [Learning transferable visual models from natural language supervision](#). *Preprint*, arXiv:2103.00020.

Advait Sarkar. 2024. Large language models cannot explain themselves. *arXiv preprint arXiv:2405.04382*.

Vaishnavi Shrivastava, Percy Liang, and Ananya Kumar. 2023. [Llamas know what gpts don't show: Surrogate models for confidence estimation](#). *ArXiv*, abs/2311.08877.

Adi Simhi, Itay Itzhak, Fazl Barez, Gabriel Stanovsky, and Yonatan Belinkov. 2025. Trust me, i'm wrong: High-certainty hallucinations in llms. *arXiv preprint arXiv:2502.12964*.

Zilu Tang, Rajen Chatterjee, and Sarthak Garg. 2025. [Mitigating hallucinated translations in large language models with hallucination-focused preference optimization](#). *Preprint*, arXiv:2501.17295.

Fei Wang and 1 others. 2024. Can linguistic knowledge improve multimodal alignment in vision-language pretraining? *ACM Transactions on Multimedia Computing, Communications and Applications*, 20(12):1–22.

Xiyang Wu, Tianrui Guan, Dianqi Li, Shuaiyi Huang, Xiaoyu Liu, Xijun Wang, Ruiqi Xian, Abhinav Shrivastava, Furong Huang, Jordan Lee Boyd-Graber, and 1 others. 2024. Autohallusion: Automatic generation of hallucination benchmarks for vision-language models. *arXiv preprint arXiv:2406.10900*.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*.

Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2024a. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*, 15(2):1–38.

Rui Zhao, Hangjie Yuan, Yujie Wei, Shiwei Zhang, Yuchao Gu, Lingmin Ran, Xiang Wang, Jay Zhangjie Wu, David Junhao Zhang, Yingya Zhang, and Mike Zheng Shou. 2024b. [Evolvedirector: Approaching advanced text-to-image generation with large vision-language models](#). In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Lanyun Zhu, Deyi Ji, Tianrun Chen, Peng Xu, Jieping Ye, and Jun Liu. 2024. Ibd: Alleviating hallucinations in large vision-language models via image-biased decoding. *arXiv preprint arXiv:2402.18476*.

## Appendix

### A Data Collection Pipeline

In this section, we will demonstrate the data collection pipeline and the calls for the LLM for a single example of an image. In Figure 6 the pipeline starts by instructing the LLM to return a description of the image, which it does. The description it generates in the figure contains a hallucination which is marked in - marked in purple. In blue there is a correct statement though. Four probes are manually derived from this generated description, and the model is asked whether each probe is correct or not. This is judged by human feedback (represented by the person's icon), which represents the “true labels”, and by the *LLaVA* model (represented by the computer's icon). In the first probe both the model and the human judgments are the same, and they both agree on the correctness of the probe. This is not the case with the fourth probe which is a false statement the model generated, but the model predicts it is correct. From this two calls for the *LLaVA* model, we can collect all features mentioned in §4, and the other features which were not mentioned in this paper.

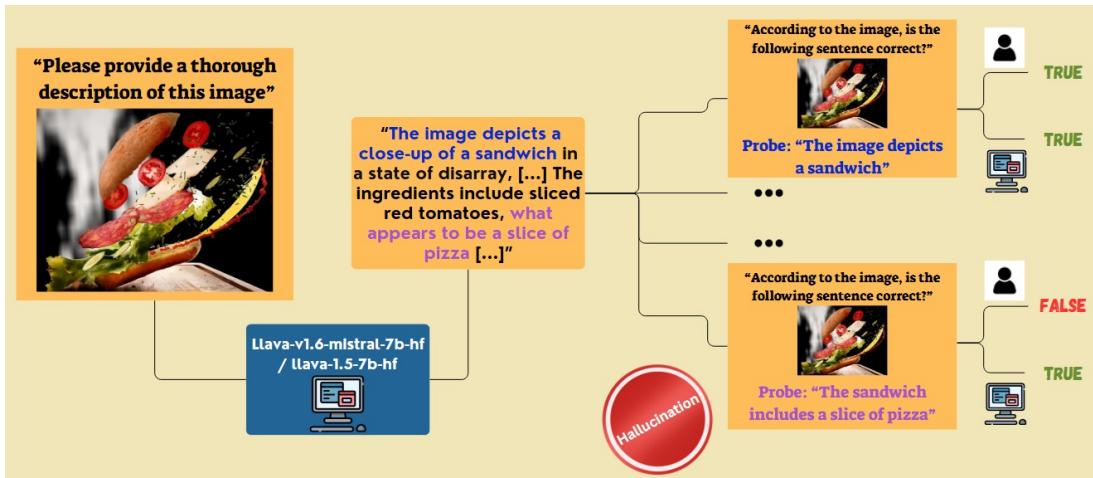


Figure 6: An illustration of the data collection pipeline.

### B NTPs analysis

In order to justify the use of both the *Description NTPs* and *Linguistic NTPs*, some statistics were examined of both types.

#### B.1 Description NTPs

Figure 7 demonstrates that the *Description NTPs* are a viable feature that can differentiate in some manner between texts that do not contain hallucinations and texts which do. Though the distributions share a great amount of probability mass, the difference between these two distributions is still notable, and the difference between the two can also be observed in the box plot. Hence, we believe in the potential of these NTPs as a useful feature that can assist in detecting hallucinations.

#### B.2 Linguistic NTPs

We witnessed the merits of using the *Description NTPs* for detecting hallucinations, and their analysis revealed some repetitive peaks and patterns, which were hypothesized to be connected to the linguistic component of the NTPs. To examine the influence of using the collected *Linguistic NTPs*, as a proxy for the linguistic part of the text, we first checked the correlation between both types of NTPs. It was hypothesized that a high correlation between them can indicate the merits of using *Linguistic NTPs* as a tool to decrease the noise and anomalies coming from the linguistic part of the generation. Considering the Spearman Correlation, the result was that the average correlation is 0.755, and the median correlation was

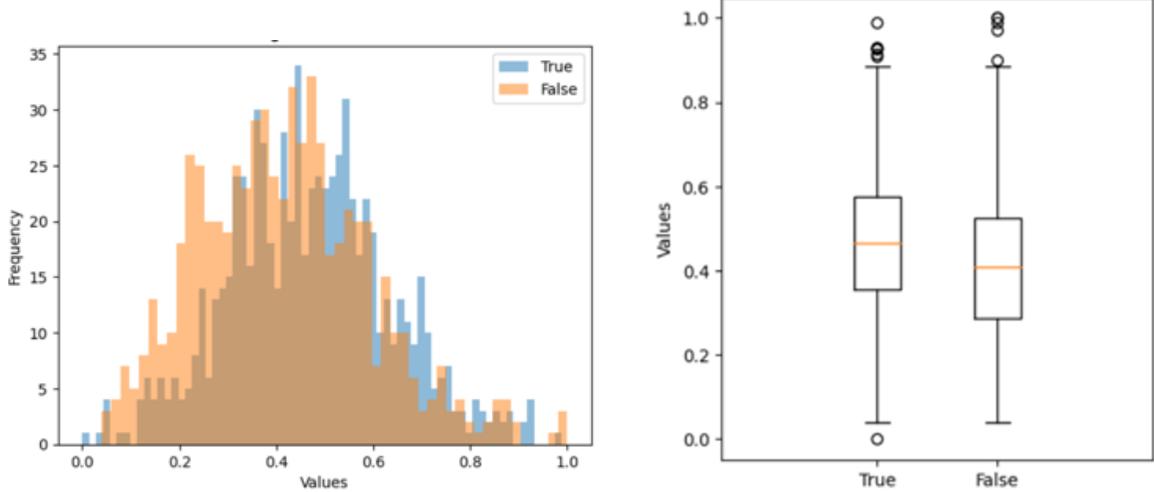


Figure 7: Distributions (left) and box-plot (right) of *Description NTPs* in contexts that do not contain hallucinations and in contexts that do. In the box plot, the left box corresponds to the *Description NTPs* in contexts that do not contain hallucinations, and the right one corresponds *Description NTPs* in contexts that contain hallucination(s). In both plots, NTPs are aggregated using a geometric mean to produce a single number for each context.

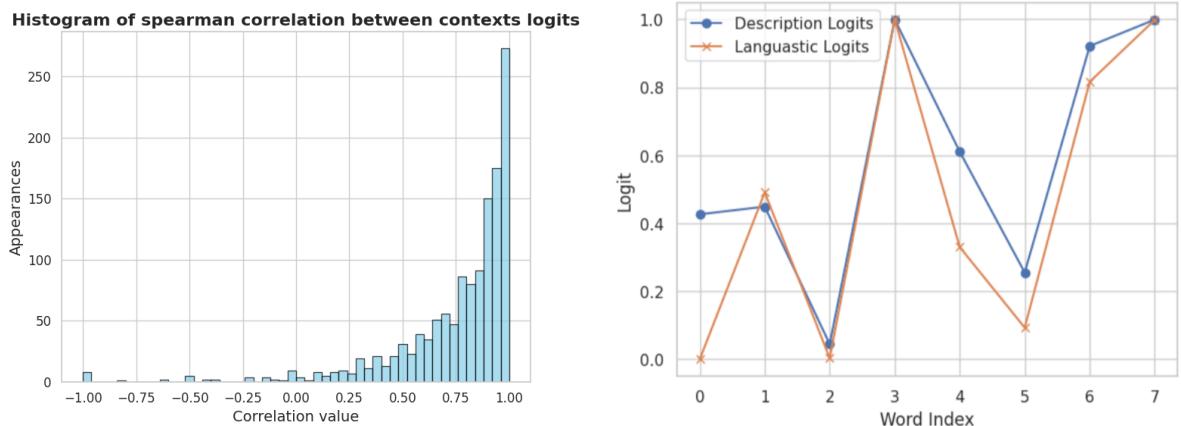


Figure 8: Histogram of the Spearman Correlation values between the *Description NTPs* and the *Linguistic NTPs* (left). A single sampled example of the similar trends both NTPs exhibit in one of the contexts (right)

0.857. Figure 8 illustrates the distribution of correlations among the different contexts, and demonstrates the strong correlation between both NTPs types.

## C Hyperparameter Tuning

For our three ML models, we performed hyperparameter tuning using grid search to identify the optimal parameters that maximize the AUC-ROC score on the validation set. The best-performing parameters were then used to train the final model, which was evaluated on the test set.

LR and SVM were implemented using the LogisticRegression and SVC classes from the scikit-learn library. The XGBoost model was implemented using the train function from the xgboost library. The specific search grids for each model are detailed below.

**LR:** We optimized the regularization strength and penalty type while considering different solvers. The search grid included:

- $C \in \{0.1, 1, 10, 100\}$  (Regularization strength)
- Penalty type:  $\{L_1, L_2\}$

- Solvers: {lbfgs, liblinear, newton-cg, newton-cholesky, sag, saga}

**SVM:** We explored different values for the regularization parameter ( $C$ ), kernel type, and kernel coefficient ( $\gamma$ ) for the rbf kernel:

- $C \in \{0.1, 1, 10, 100\}$
- Kernel type: {linear, rbf}
- $\gamma \in \{\text{scale}, \text{auto}, 1, 0.1, 0.01, 0.001\}$

**XGBoost:** We tuned multiple hyperparameters including tree depth, learning rate, regularization terms, and subsampling ratios:

- Maximum tree depth: {3, 5}
- Learning rate: {0.1, 0.2}
- Minimum child weight: {3, 5, 7}
- Gamma (regularization parameter): {0.01, 0.1}
- Subsample ratio: {0.6, 0.7}
- Column sampling ratio: {0.6, 0.7}
- L1 regularization ( $\alpha$ ): {0.1, 1, 10}
- L2 regularization ( $\lambda$ ): {1, 10, 100}

Grid search with cross-validation was employed to systematically evaluate all parameter combinations. The best-performing hyperparameter set for each model was then used for final training and evaluation on the test dataset.

## D Tabular Results for Figure 3

ML Models Performance					
Preds	Linguistic	XGBoost	SVM	LR	
No Preds	No	$0.589 \pm 0.008$	$0.597 \pm 0.008$	$0.606 \pm 0.007$	
	Yes	$0.599 \pm 0.008$	$0.611 \pm 0.008$	$0.615 \pm 0.007$	
<i>LLaVA</i>	No	$0.647 \pm 0.007$	$0.660 \pm 0.008$	$0.651 \pm 0.007$	
	Yes	$0.649 \pm 0.008$	$0.658 \pm 0.008$	$0.652 \pm 0.007$	
<i>PaliGemma</i>	No	$0.739 \pm 0.006$	$0.758 \pm 0.005$	$0.761 \pm 0.006$	
	Yes	$0.735 \pm 0.007$	$0.759 \pm 0.006$	$0.761 \pm 0.006$	
<i>LLaVA</i> and <i>PaliGemma</i>	No	$0.760 \pm 0.006$	$0.772 \pm 0.005$	$0.771 \pm 0.005$	
	Yes	$0.761 \pm 0.007$	$0.770 \pm 0.006$	$0.769 \pm 0.005$	
VLM Performance					
VLM Type	Raw Score	XGBoost	SVM	LR	
<i>LLaVA</i>	$0.632 \pm 0.007$	–	–	–	
<i>PaliGemma</i>	$0.757 \pm 0.005$	–	–	–	
<i>LLaVA</i> and <i>PaliGemma</i>	–	$0.754 \pm 0.006$	$0.772 \pm 0.005$	$0.772 \pm 0.005$	

Table 1: Detailed AUC-ROC performance (with 95% confidence intervals) of traditional ML models and VLMs across different configurations. The upper section evaluates ML models using only NTP-based features as distinct inputs or in combination with VLM predictions. The lower section reports standalone VLM performance. Where a single VLM prediction is directly adopted as the final prediction and when both VLM predictions are combined, ML models utilize both prediction features to make the final prediction.