On the Instability of Local Posthoc Explanations

Anonymous ACL submission

Abstract

Explanations of model decisions are important for building trust in machine learning systems, especially in high-stakes areas like healthcare. However, existing post-hoc explanation methods often suffer from instability, producing inconsistent results for similar inputs and thereby undermining their reliability. In this paper, we conduct a systematic investigation into the factors contributing to this instability across different model architectures and explanation methods. Our analysis reveals that model type, rather than hyperparameters, is the primary driver of stability, with transformer models exhibiting greater instability compared to architectures like LSTMs, regardless of model size. We also explore the role of sparsity in transformer models, finding that while sparse pretrained transformers improve the stability of gradient-based explanations, similar benefits are not observed with perturbation-based methods. Furthermore, our findings suggest that a portion of the disagreement between different explanation methods can be traced back to this instability, highlighting the importance of stable model explanations for developing more reliable and interpretable AI systems.

1 Introduction

Explanations allow us to understand possible rationales behind complex model decisions, and decide when to rely on these predictions. These explanations can guide future choices; for instance, one might reject a model's recommendation after understanding its reasoning. As explanations are increasingly relied upon in critical sectors such as healthcare (Elshawi et al., 2019), law (Whitmore et al., 2016), and finance (Ibrahim et al., 2019), the explanations must be stable to draw reliable conclusions.

Several methods have been proposed to generate explanations *post hoc* or after a model has been trained. Post hoc explanations are practical, do not necessitate access to model internals, and are well-established methods that are straightforward to use. Yet, prior work indicates that local post-hoc explanations often exhibit instability (Ghorbani et al., 2018; Alvarez-Melis and Jaakkola, 2018a), susceptibility to perturbation attacks (Sinha et al., 2021), and vulnerability to deliberate adversarial manipulations (Slack et al., 2020). This fragility is evident in instances where similar inputs yield divergent explanations as seen in Figure 1, or a single input produces conflicting interpretations. Furthermore, there is a notable lack of consensus among different explanation techniques (Krishna et al., 2022) where the explanations between different methods offer conflicting results. Such inconsistency in explanations can erode trust in the model, amplify discord among methods, and potentially lead to erroneous

Figure 1: The same model and explanation method can yield different explanations for nearly identical inputs, with differences in the input highlighted in **bold**. Moreover, in example C), altering the internal SHAP hyperparameter (number of feature permutations) diverges the explanation further

decision-making.

This paper investigates the factors influencing explanation stability across different model architectures and explanation methods. Our findings indicate that model type is the primary determinant of stability, with hyperparameters playing a secondary role. While transformer models generally exhibit higher instability in explanations, this instability is not necessarily related to the number of parameters; for instance, DistilBERT, a larger transformer model, proved to be more stable than the smaller bert-tiny. Interestingly, even LSTMs with more parameters than transformers produce more stable explanations. We demonstrate that using sparse pretrained models can improve the stability of gradient-based explanations like Integrated Gradients, whereas fine-tuning for sparsity offers little to no benefit. Finally, we explore the downstream effects of instability, revealing that stable setups reduce disagreements between different explanation methods, suggesting that a portion of these disagreements stems from instability in the explanations themselves.

2 Background

Local post-hoc explanations provide insights into individual predictions, aiding in understanding specific decisions and debugging. Local post-hoc perturbation-based methods work by altering the input data (e.g., removing, masking, or substituting inputs) and observing the model's reaction to these changes, measuring the difference from the original output. These methods are model-agnostic as they do not require access to the model's internals. They compute feature attributions by training a simpler local model around a point of interest. Examples include LIME (Ribeiro et al., 2016a), SHAP (Lundberg and Lee, 2017), and BayesLIME (Zhao et al., 2020). Gradient-based methods, suitable for neural networks, rely on backpropagation to calculate the attribution of all input features in a single forward and backward pass. They compute the partial derivatives of the output concerning each input feature, resulting in a saliency map in applications like computer vision. Examples include Input Gradient (Hechtlinger), Integrated Gradients (Sundararajan et al., 2017a), Grad-CAM (Selvaraju et al., 2016), and SmoothGrad (Smilkov et al., 2017).

Previous work has shown that these local post hoc methods are unstable (Adebayo et al., 2020; Alvarez-Melis and Jaakkola, 2018a) even when the underlying model is stable. To address issues of instability, previous work has attempted averaging explanations (Lee et al., 2019), removing random perturbations from LIME (Zafar and Khan, 2019), creating credible intervals for feature attributions (Zhao et al., 2020), and introducing a regularization parameter during training (Lakkaraju et al., 2020; Chalasani et al., 2018). Agarwal et al. (2023a) evaluates the stability and faithfulness of different explanations across multiple datasets and finds that a model's stability and faithfulness vary depending on the dataset and task. Despite these challenges, working on local explanations is still worthwhile. Local explanations are pertinent for debugging individual predictions, understanding model behavior on a case-by-case basis, and ensuring fairness in specific instances. Unlike previous studies that focus on individual methods or specific aspects of model behavior, we conduct a large-scale analysis.

2.1 Defining an Explanation

Explainability is intrinsically tied to a problem, domain, and audience (Ehsan et al., 2023). The explanation for a machine learning practitioner is not the explanation for a healthcare professional. Depending on the context, explanations can vary from assigning feature importance scores to generating free-text rationalizations for a model's behavior (Slack et al., 2022; Shen et al., 2023; Lakkaraju et al., 2022). In this study, we use feature importance by highlighting tokens that drive a model's predictions, similar to saliency maps for images. This approach provides users with insights into the model's decision-making process and helps verify if the model focuses on relevant data features. These extractive explanations, while not fully transparent, offer plausible rationales for model predictions and are widely used in critical domains like healthcare (Elshawi et al., 2019) and finance (Ibrahim et al., 2019). They can be effective if they meaningfully correlate with the model's predictions (Wiegreffe and Pinter, 2019), despite the gap between these explanations and human understanding (Kaur et al., 2020; Shen and Huang, 2020). Such explanations also help practitioners calibrate their models (Ye and Durrett, 2022). When deployed, extractive explanations should adhere to the principle of stability (Sundararajan et al., 2017b). Our research examines the causes of instability and proposes mitigation strategies, extending applicability across explanatory frameworks. Additionally, more recent dialogue-based explanations (Slack et al.,

Method	Parameter	Values
LIME	# of samples of the original model used to train the surrogate interpretable model	[100, 1000, 5000]
SHAP	# of feature permutations tested	[25, 100, 1000]
Integrated Gradients	# of steps used by the approximation method	[50, 500]

Table 1: Comparison of hyperparameters for different explanation methods

2022; Shen et al., 2023) rely on feature-based explanations, and improving the stability of local post hoc explanations can enhance these interactive explanations.

3 Isolating Instability

There are known causes of instability in explanations. For instance, using too few perturbed samples for methods such as LIME can hinder fitting a local model (Zhao et al., 2020). Similarly, the inherent limitations of linear methods in capturing the complexities of non-linear relationships they aim to mirror (Ribeiro et al., 2016b). In the third example in Figure 1, the observed increase in variation among token attributions in the original explanations is attributed to the use of a smaller number of samples in the SHAP method. While factors contributing to instability have been identified, the extent of their impact on the stability of explanations remains unclear. Understanding the relative influence of each factor provides insight into creating stable explanations. Our investigation concentrates on the following hypotheses: instability is driven by hyperparameter selection of the explanation method, the complexity of the individual data point being explained, or the complexity of the model itself in terms of parameter size. Each hypothesis highlights a different aspect of the interaction between methods, models, and data in producing variable explanations. We acknowledge that these components do not operate in isolation and that each explanation method introduces unique considerations.

Explanation Method Hyperparameters Slack et al. (2021) propose modeling uncertainty in local post-hoc explanations as credible intervals, demonstrating that optimizing hyperparameters leads to decreased intervals of uncertainty. For instance, in the case of LIME, increasing the number of samples provides the local model with more data to fit, these models tend to converge on a more consistent explanation. Similarly, (Zhou et al., 2021) proposes S-LIME which uses a hypothesis-testing framework to determine the number of perturbation points needed to guarantee stability in LIME. This suggests that by carefully selecting and optimizing hyperparameters, we can reduce uncertainty in the explanation methods and enhance the overall stability of the explanations. This hypothesis underscores the importance of hyperparameter selection in improving the reliability and robustness of local post-hoc explanation methods.

Model Complexity The second hypothesis considers that instability emerges from the complexity of the model being explained. As the number of parameters increases, explanation methods may fail to capture the underlying relationships accurately, leading to increased variation in stability. Ribeiro et al. (2016b) acknowledge that local explanations provided by methods like LIME often fail to reflect the global behavior of complex models, resulting in discrepancies. This hypothesis posits that local surrogate models, which fit linear models to the local feature space, struggle to represent the intricate relationships in high-parameter models, leading to less reliable explanations. As the dimensionality and complexity of the model increase, the explanations provided by post-hoc methods become more variable.

Data Complexity Agarwal et al. (2023b) benchmark six explanation methods on two datasets, finding that the stability of each method varied depending on the dataset, even when other factors are held constant. This leads to the hypothesis that as the complexity of input data increases, so does the variation in token attribution. For example, longer texts or inputs with higher perplexity tend to produce more unstable explanations. Moreover, Alvarez-Melis and Jaakkola (2018b) notes that instability can be observed even when the underlying model is stable.

3.1 Quantifying Stability

Here stability refers to the consistency of explanations across slightly varied inputs. The intuition is that nearly identical inputs should receive *similar* explanations, a concept well-grounded in previous literature (Bhatt et al., 2020; Yeh et al., 2019; Dai

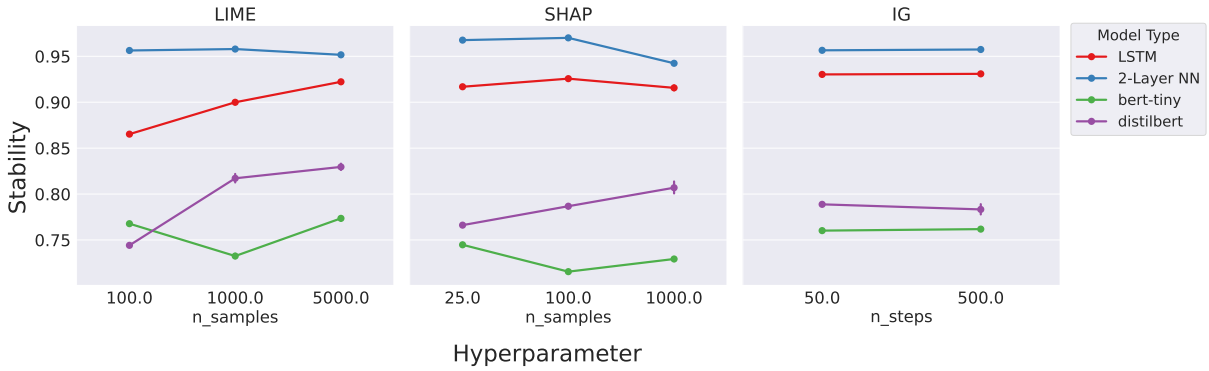

Figure 2: Stability of each explanation method as a function of a key hyperparameter. LIME's stability improves by up to 8% with more samples, while the number of approximation steps for Integrated Gradients has minimal impact. Across methods, model type significantly influences stabilit

et al., 2022). For extractive explanations, an effective stability metric should focus on the consistency of the top-ranked features, which are the weights assigned to each token measuring its importance in driving a prediction. These top-ranked features have an outsized impact on model decisions. Additionally, the stability metric should be designed to compare different explanation methods, even when they use different units and should be easy to interpret.

Alvarez-Melis and Jaakkola (2018a) measures stability by introducing a local Lipschitz metric, which evaluates the sensitivity of explanations to small changes in input by quantifying the maximum rate at which the explanation can change. Essentially, this metric captures how much the explanation can vary in response to minor perturbations in the input. A smaller Lipschitz constant indicates greater stability, as the explanation changes more slowly with respect to input variations. However, the Lipschitz metric has some limitations. It produces a unitless ratio, making it difficult to interpret practically. This ambiguity makes it challenging to determine what constitutes a "good" or "bad" stability score. Additionally, calculating the Lipschitz constant is not straightforward, as it requires evaluating the maximum change over all possible perturbations of the input, which can be computationally intensive.

Given these challenges, we choose to use the Normalized Discounted Cumulative Gain (NDCG) metric (Järvelin and Kekäläinen, 2000), a standard from the field of Information Retrieval, to quantify stability. NDCG assesses the ranking quality by considering both the position and relevance of items in a list, offering a measure of how well the ranking preserves the importance of all features, particularly the top features. This metric captures the essence of what we seek in a stable explanation: that the most influential factors remain consistently identified and ranked the same, even with minor variations in input. Our methodology involves comparing the attributions generated for the original input text to those generated for inputs that have been slightly modified. The stability is then calculated as the minimum NDCG value from the perturbed samples:

$$\text{Stability} = \min_{j=1}^{m} \text{NDCG}_j$$

Where m is the number of perturbed samples defaulting to 10. This minimum represents the "worst-case" deviation in explanations between the original and slightly modified examples. By adopting the minimum NDCG value from the perturbed samples, our stability metric captures the largest deviation in explanations among a set of highly similar inputs. Following the principles in Sundararajan et al. (2017b), we measure the absolute value of changes in explanations, highlighting shifts regardless of their direction. This approach focuses on preserving the order and relevance of top features, with the NDCG score ranging from 0 to 1, where 1 signifies perfect stability.

To create the slightly modified inputs, we perturb 10% of the input tokens by selecting synonyms from the embedding space using cosine similarity following (Garg and Ramakrishnan, 2020). On average, the perturbed inputs maintain a 95% cosine similarity evaluated with the all-MiniLM-L6-v2 sentence transformer to the original text, implying their explanations should be closely aligned. This approach ensures that the stability metric ef-

fectively measures the consistency of explanations, focusing on the most critical features.

4 Instability Experiments

In this section, we explore the factors contributing to instability in model explanations by systematically varying explanation method hyperparameters, model complexity, and data complexity. Our aim is to understand how these factors affect the consistency and reliability of generated explanations, with the ultimate goal of identifying why certain models or configurations are more prone to instability, thereby guiding the development of more trustworthy AI systems.

4.1 Experimental Setup

Similar to previous explanation research (Krishna et al., 2022; Wiegreffe and Pinter, 2019; Treviso and Martins, 2020), our experiments use the AG's News Corpus (Zhang et al., 2015), which contains 120,000 training and 7,600 test examples. The task involves classifying articles into four categories: world news, sports, business, and science and technology, based on their titles and descriptions. We evaluate four different model architectures: a 2-layer neural network, a vanilla long short-term memory (LSTM) network, and two pretrained transformer variants, bert-tiny (Bhargava et al., 2021) and DistilBERT (Sanh et al., 2019). For each model type, we explore a range of training configurations, including undertraining, overtraining, and variations in embedding sizes, to assess the impact of model complexity on explanation stability. To ensure robustness in our findings, we train 30 models for each configuration, resulting in a total of approximately 2,000 models. For explanation methods, we use LIME and SHAP for perturbation-based approaches and Integrated Gradients for a gradient-based approach, generating all explanations using the Captum library (Kokhlikyan et al., 2020). To measure stability, we randomly select 300 points from the test set and generate 10 perturbed versions of each, ensuring 95% semantic similarity to the original inputs, and compute the minimum NDCG score to obtain stability.

Explanation Methods Hyperparameters To examine the impact of hyperparameters on explanation stability, we select a key hyperparameter for each of the three explanation methods: the number of samples for LIME, the number of feature permutations for SHAP, and the number of approximation steps for Integrated Gradients. As expected, our results as seen in 2 show that perturbation methods generally benefit from a higher number of samples, with LIME showing up to an 8% improvement in stability as the number of samples increases to 5,000. However, the number of approximation steps for Integrated Gradients has less of an impact. Across all three explanation methods, model types appeared in the same order of stability, with the 2-layer neural network at the top, followed by the LSTM, and lastly bert-tiny and DistilBERT. This indicates that while hyperparameters are important, model type is the primary driver of stability.

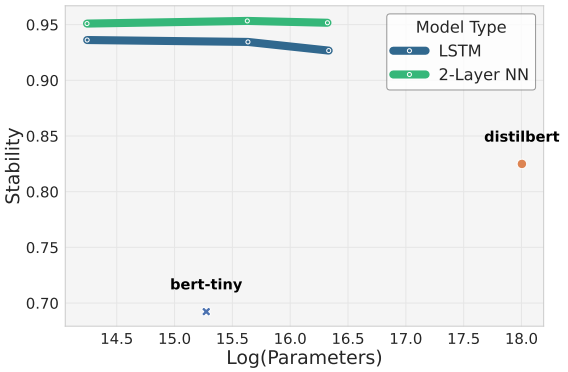

Figure 3: Stability of explanations across model complexities, showing that LSTM and 2-layer neural networks are more stable than transformers, regardless of parameter count. Notably, DistilBERT, which is larger than bert-tiny, also demonstrates greater stability. This underscores that model type, rather than size, is the primary driver of stability.

Model Complexity Next, we explore the hypothesis that models with more parameters are harder to explain. To test this, we adjust the number of parameters in each model architecture. For the 2-layer neural network, we vary the embedding size; for the LSTM, we vary the embedding size and hidden dimension size; and for the transformers, we compared bert-tiny and DistilBERT, which have 4 million and 66 million parameters, respectively. We use the explanation methods with the optimal hyperparameter values identified in the prior analysis to minimize instability from the explanation method. Our results, shown in 3, plot the number of parameters against the average stability across LIME, SHAP, and Integrated Gradients. Surprisingly, even when the number of parameters in the 2-layer network and LSTM exceeded that of bert-tiny, their explanations are still more stable. Similarly, despite having significantly more parameters, Dis-

tilBERT is more stable on average than bert-tiny. This finding reinforces that model type, rather than the number of parameters, is the main driver of stability. A complete list of model configurations can be found in the supplementary materials B.

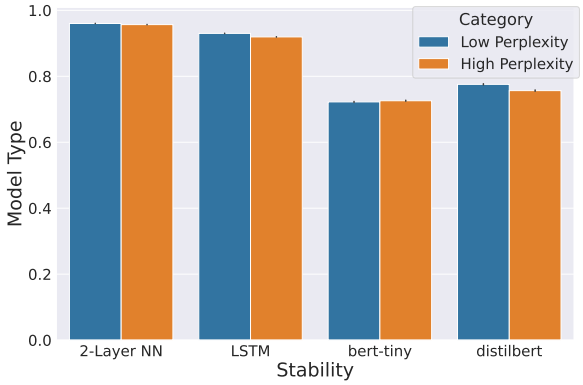

Figure 4: Stability comparison across model types for high and low perplexity texts, showing minimal differences and indicating that input complexity has little impact on explanation stability.

Data Complexity Lastly, to investigate whether the complexity of input data impacts the stability of explanations, we analyze samples categorized by their perplexity scores, using GPT-2 as a reference. We define "hard" texts as those in the top 25% of perplexity scores (above 100) and "easy" texts as those in the bottom 25% (below 50). We then compare the average stability of explanations generated for these two categories. As illustrated in Figure 4, the stability of explanations shows minimal variation between the "hard" and "easy" texts across different model types. This finding indicates that input complexity has a negligible effect on explanation stability. Instead, it highlights that the type of model, rather than the complexity of the input data, is the predominant factor influencing the stability of explanations.

5 Transformer Stability

In the previous section, we found that Transformer architectures exhibit greater instability in explanations, regardless of the explanation method or model size. This section delves into the possible reasons behind this instability. Previous research (Chen et al., 2020; Correia et al., 2019; Treviso and Martins, 2020) indicates that sparse models—those that focus on fewer input features—can enhance interpretability. We aim to determine whether sparse models also provide more stable explanations than their dense counterparts and whether they consistently identify the same key features. Our hypothesis is that the instability in Transformer models arises from over-parameterization or the effects of heterogeneous learning processes typical in ensemble methods. To test this, we experiment with Transformer models under different conditions, including applying sparsity during fine-tuning and using pretrained sparse models. By reducing the number of active parameters (addressing over-parameterization) and promoting more focused learning (potentially mitigating the effects of heterogeneity), we seek to determine whether these adjustments can lead to more stable and consistent model explanations.

5.1 Experimental Setup

To examine the impact of sparsity on Transformer models, we conduct experiments under two conditions: sparsity introduced during fine-tuning and sparsity inherent in a pretrained model. These experiments are performed on three datasets: the AG News Corpus (Zhang et al., 2015) for text classification, the SST2 dataset (Socher et al., 2013) for sentiment analysis, and the MNLI dataset (Williams et al., 2018) for recognizing textual entailment. We select the best-performing hyperparameter settings identified in earlier sections for each explanation method. From the pool of models used in the previous section, we sample 10 DistilBERT models and apply the two sparsity strategies. The first strategy involves introducing sparsity during fine-tuning using the *sparsemax* function (Martins and Astudillo, 2016), which generates sparse probability distributions by assigning zero probabilities to certain outputs, unlike the traditional softmax function. This method aims to create a more focused attention mechanism, potentially improving stability. The second strategy uses a pretrained sparse version of DistilBERT (Zafrir et al., 2021), which combines weight pruning and model distillation to achieve high sparsity while maintaining performance. To assess stability, we randomly select 300 samples from each test set, generate 10 perturbations for each, ensuring 95% semantic similarity, and measure the maximum deviation in the NDCG score as before.

5.2 Results

We assess the stability of explanations across three datasets: the AG News Corpus, the SST2 dataset, and the MNLI dataset. The results, summarized in

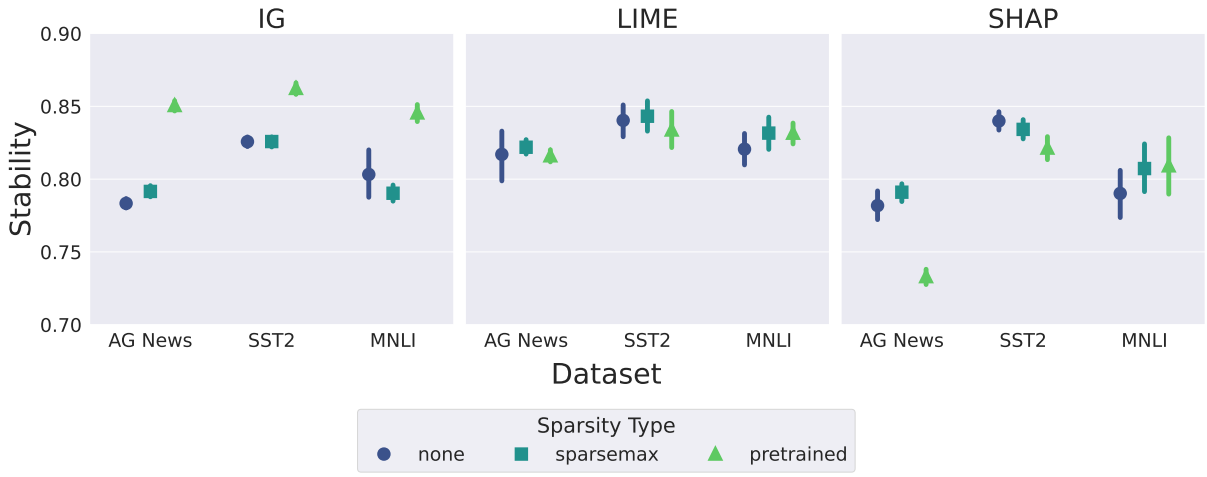

Figure 5: Comparison of explanation stability without sparsity, with sparsity introduced during fine-tuning, and using a pretrained sparse model. Pretrained sparsity consistently improves the stability of Integrated Gradients across three datasets, while perturbation-based methods like LIME and SHAP show no improvement.

Figure 5, show that using a sparse pretrained model consistently enhances the stability of explanations generated by Integrated Gradients, regardless of the dataset. For example, in the case of AG News, there is a 6% boost in average stability. This improvement is likely due to the reduction of noise in the gradient flow, enabling the model to focus more precisely on the most relevant features. As a result, the explanations become more consistent. These results support the hypothesis that sparsity enhances the stability of gradient-based methods by making the internal gradients more predictable and concentrated on critical inputs.

In contrast, perturbation-based methods like LIME and SHAP do not show benefits from sparsity. Likely because these methods are more influenced by the inherent randomness in their perturbation process and the model's sensitivity to input variations, which is not effectively mitigated through sparsity. Moreover, introducing sparsity during fine-tuning with the sparsemax function did not improve stability for any of the explanation methods tested. This suggests that applying sparsity during fine-tuning is insufficient to enhance the stability of explanations and suggests that the benefits of sparsity are more effectively realized when integrated during the pretraining phase.

5.3 Discussion

Our exploration into Transformer stability highlights that sparse models can enhance the stability of gradient-based explanations such as Integrated Gradients. Yet, sparsity does not help the stability of perturbation-based methods like LIME and SHAP. While versatile, perturbation-based methods exhibit greater variability due to their inherent reliance on random perturbations of input data. This randomness can lead to inconsistent explanations, a challenge that methods like D-LIME (Zafar and Khan, 2019) and S-LIME (Zhou et al., 2021) mitigate by controlling for variability in the perturbation process. Interestingly, adding sparsity during fine-tuning using sparsemax (Martins and Astudillo, 2016) dud not improve the stability of gradient-based explanations. Our results suggest that the advantages of sparsity are more pronounced when it is incorporated during the pretraining phase rather than applied later.

Furthermore, while sparsity can help focus attention and improve stability in certain methods, it is not universally beneficial across all explanation techniques. Meister et al. (2021) shows that sparse attention does not necessarily correspond to a sparse set of influential inputs. Instead, inducing sparsity can lead to increased contextualization within intermediate representations, making attention distributions less reflective of the actual importance of individual inputs. This issue is separate from the stability of an explanation and points to a more complex interaction between sparsity and interpretability. Despite these challenges, our study demonstrates that sparse pretrained transformers, when combined with gradient-based explanations, can strike a balance between explanation stability and model performance. While simpler models are often recommended for more reliable explanations, our findings suggest that sparse transformers offer

a viable alternative, particularly in scenarios where performance is critical.

6 Stability and Disagreement

Machine learning practitioners often rely on multiple explanation methods to interpret model decisions (Krishna et al., 2022). However, these methods can sometimes provide conflicting explanations, making it challenging to understand and trust the model's behavior. This section investigates whether some of these disagreements are due to instability in the explanations themselves.

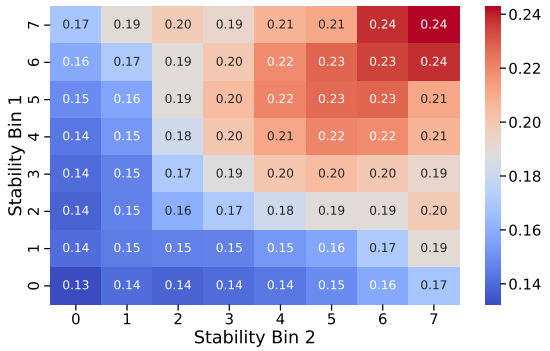

Figure 6: Higher stability setups show increased agreement between explanation methods, indicating that improving stability can reduce disagreement in feature identification

6.1 Experimental Setup

To explore this, we analyze how frequently different setups—combinations of model configurations, explanation methods, and hyperparameters—identify the same top three tokens in their explanations following Krishna et al. (2022). We use the 180 DistilBERT models previously trained on the AG's News Corpus. Each setup is assigned a stability score based on the average stability across all data points and grouped into one of eight stability bins using the Fischer-Jenks algorithm, where Bin 0 represents the least stable setup. We then calculate the average feature agreement within each stability bin, measuring how consistently the top three features are identified across different setups. The goal is to determine if more stable setups result in higher agreement between different explanation methods. The cutoffs for these bins are detailed in the Appendix 6.

6.2 Results

Our results, depicted in the heatmap of Figure 6, indicate that more stable setups lead to higher agreement between explanation methods. The most stable setups (in the higher bins) showed up to an 11% increase in agreement compared to the least stable ones. This suggests that a portion of the disagreement between explanation methods is due to instability in the explanations themselves. Although overall feature agreement is relatively low due to averaging across all three methods, improving stability appears to be a promising strategy for reducing disagreement, leading to more reliable and interpretable AI systems.

7 Conclusion

This work addresses the question of what factors contribute to instability in model explanations and how to mitigate them. Our investigation reveals that while hyperparameter tuning can enhance stability in some explanation methods like LIME, the model type plays a more significant role, particularly in complex architectures like transformers. We demonstrate that transformer models exhibit greater instability compared to simpler models like LSTMs, which are not driven by parameter count, and that sparsity, especially when introduced during pretraining, can improve the stability of gradient-based explanations. However, this benefit does not extend to perturbation-based methods. Furthermore, our analysis shows that instability contributes to the disagreement between different explanation methods, suggesting that efforts to enhance stability can reduce this discord and lead to more reliable and interpretable AI systems. Future work may focus on extending the reliability of explanation stability to natural text explanations, which present unique challenges, such as maintaining stability when multiple valid free-text rationales are possible. Addressing these challenges could lead to more robust and trustworthy AI systems in domains where textual explanations are critical.

8 Limitations

A limitation of our study is its focus on a single dataset, the AG News Corpus, for text classification. This choice was made in alignment with prior studies (Krishna et al., 2022; Agarwal et al., 2023a) that have documented issues of disagreement and variable stability within this text classification task. To conduct this large-scale analysis, we trained and evaluated 2,000 models across three explanation methods. For each stability calculation, we required a minimum of 10 similar examples, resulting in 11 explanations per calculation. With 300 test points, this totals 3,300 explanations per model, leading to approximately 20 million explanations overall. This scale of computation demanded significant GPU resources, raising concerns about the ecological impact and sustainability of such research practices. The intensive use of computational resources highlights the need for future studies to consider more sustainable approaches, particularly as the environmental costs of large-scale AI research become increasingly important. Moreover, It is important to recognize that explanations do not solve the underlying issues of bias or unbalanced representation. Heavier emphasis should be placed on data collection and curation, label choice for model optimization, and the use of inherently transparent models. As previously stated, we choose to focus on feature rankings as an explanation when there is a known gap between feature weight and interpretability, Kaur et al. (2020); Shen and Huang (2020) show attributions are often misused or misunderstood. Yet, extractive explanations provide a targeted, albeit not fully transparent, insight into AI decision-making processes, critical in domains such as healthcare and finance, while also highlighting the challenges and necessary improvements in explanation stability and quality across various explanatory frameworks.

References

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. 2020. Sanity Checks for Saliency Maps. ArXiv:1810.03292 [cs, stat].

Chirag Agarwal, Satyapriya Krishna, Eshika Saxena, Martin Pawelczyk, Nari Johnson, Isha Puri, Marinka Zitnik, and Himabindu Lakkaraju. 2023a. Openxai: Towards a transparent evaluation of model explanations.

Chirag Agarwal, Satyapriya Krishna, Eshika Saxena, Martin Pawelczyk, Nari Johnson, Isha Puri, Marinka Zitnik, and Himabindu Lakkaraju. 2023b. OpenXAI: Towards a Transparent Evaluation of Model Explanations. ArXiv:2206.11104 [cs].

David Alvarez-Melis and Tommi S. Jaakkola. 2018a. On the Robustness of Interpretability Methods. ArXiv:1806.08049 [cs, stat].

David Alvarez-Melis and Tommi S. Jaakkola. 2018b. Towards Robust Interpretability with Self-Explaining Neural Networks. ArXiv:1806.07538 [cs, stat].

Prajjwal Bhargava, Aleksandr Drozd, and Anna Rogers. 2021. Generalization in nli: Ways (not) to go beyond simple heuristics.

Umang Bhatt, Adrian Weller, and José M. F. Moura. 2020. Evaluating and Aggregating Feature-based Model Explanations. ArXiv:2005.00631 [cs, stat].

Prasad Chalasani, Jiefeng Chen, Amrita Roy Chowdhury, Somesh Jha, and Xi Wu. 2018. Concise explanations of neural networks using adversarial training. *CoRR*, abs/1810.06583.

Tianlong Chen, Sijia Liu, Shiyu Chang, Yu Cheng, Lisa Amini, and Zhangyang Wang. 2020. Adversarial Robustness: From Self-Supervised Pre-Training to Fine-Tuning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 696–705, Seattle, WA, USA. IEEE.

Gonçalo M. Correia, Vlad Niculae, and André F. T. Martins. 2019. Adaptively sparse transformers.

Jessica Dai, Sohini Upadhyay, Ulrich Aivodji, Stephen H. Bach, and Himabindu Lakkaraju. 2022. Fairness via Explanation Quality: Evaluating Disparities in the Quality of Post hoc Explanations. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, pages 203–214. ArXiv:2205.07277 [cs].

Upol Ehsan, Philipp Wintersberger, Elizabeth A Watkins, Carina Manger, Gonzalo Ramos, Justin D. Weisz, Hal Daumé Iii, Andreas Riener, and Mark O Riedl. 2023. Human-centered explainable ai (hcxai): Coming of age. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI EA '23, New York, NY, USA. Association for Computing Machinery.

Radwa Elshawi, Mouaz H. Al-Mallah, and Sherif Sakr. 2019. On the interpretability of machine learning-based model for predicting hypertension. *BMC Medical Informatics and Decision Making*, 19(1):146.

Siddhant Garg and Goutham Ramakrishnan. 2020. BAE: BERT-based Adversarial Examples for Text Classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181. ArXiv:2004.01970 [cs].

Amirata Ghorbani, Abubakar Abid, and James Zou. 2018. Interpretation of Neural Networks is Fragile. ArXiv:1710.10547 [cs, stat].

Yotam Hechtlinger. Interpretation of prediction models using the input gradient.

Mark Ibrahim, Melissa Louie, Ceena Modarres, and John Paisley. 2019. Global explanations of neural networks: Mapping the landscape of predictions.

Kalervo Järvelin and Jaana Kekäläinen. 2000. Ir evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, page 41–48, New York, NY, USA. Association for Computing Machinery.

Harmanpreet Kaur, Harsha Nori, Samuel Jenkins, Rich Caruana, Hanna M. Wallach, and Jennifer Wortman Vaughan. 2020. Interpreting interpretability: Understanding data scientists' use of interpretability tools for machine learning. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. 2020. Captum: A unified and generic model interpretability library for pytorch.

Satyapriya Krishna, Tessa Han, Alex Gu, Javin Pombra, Shahin Jabbari, Steven Wu, and Himabindu Lakkaraju. 2022. The Disagreement Problem in Explainable Machine Learning: A Practitioner's Perspective. ArXiv:2202.01602 [cs].

Himabindu Lakkaraju, Nino Arsov, and Osbert Bastani. 2020. Robust and Stable Black Box Explanations. ArXiv:2011.06169 [cs].

Himabindu Lakkaraju, Dylan Slack, Yuxin Chen, Chenhao Tan, and Sameer Singh. 2022. Rethinking Explainability as a Dialogue: A Practitioner's Perspective. ArXiv:2202.01875 [cs].

Eunjin Lee, David Braines, Mitchell Stiffler, Adam Hudler, and Daniel Harborne. 2019. Developing the sensitivity of LIME for better machine learning explanation. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*,

volume 11006, page 1100610. International Society for Optics and Photonics, SPIE.

Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *CoRR*, abs/1705.07874.

André F. T. Martins and Ramón Fernandez Astudillo. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. *CoRR*, abs/1602.02068.

Clara Meister, Stefan Lazov, Isabelle Augenstein, and Ryan Cotterell. 2021. Is sparse attention more interpretable?

Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016a. "why should I trust you?": Explaining the predictions of any classifier. *CoRR*, abs/1602.04938.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016b. "why should i trust you?": Explaining the predictions of any classifier.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. 2016. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *CoRR*, abs/1610.02391.

Hua Shen, Chieh-Yang Huang, and Tongshuang Wu. 2023. ConvXAI : Delivering Heterogeneous AI Explanations via Conversations to Support Human-AI Scientific Writing.

Hua Shen and Ting-Hao Kenneth Huang. 2020. How useful are the machine-generated interpretations to general users? a human evaluation on guessing the incorrectly predicted labels.

Sanchit Sinha, Hanjie Chen, Arshdeep Sekhon, Yangfeng Ji, and Yanjun Qi. 2021. Perturbing Inputs for Fragile Interpretations in Deep Natural Language Processing. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 420–434, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. 2020. Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods. ArXiv:1911.02508 [cs, stat].

Dylan Slack, Sophie Hilgard, Sameer Singh, and Himabindu Lakkaraju. 2021. Reliable Post hoc Explanations: Modeling Uncertainty in Explainability. ArXiv:2008.05030 [cs, stat].

Dylan Slack, Satyapriya Krishna, Himabindu Lakkaraju, and Sameer Singh. 2022. TalkToModel: Explaining Machine Learning Models with Interactive Natural Language Conversations. ArXiv:2207.04154 [cs].

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viégas, and Martin Wattenberg. 2017. Smoothgrad: removing noise by adding noise. *CoRR*, abs/1706.03825.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017a. Axiomatic attribution for deep networks. *CoRR*, abs/1703.01365.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017b. Axiomatic Attribution for Deep Networks. ArXiv:1703.01365 [cs].

Marcos V. Treviso and André F. T. Martins. 2020. The Explanation Game: Towards Prediction Explainability through Sparse Communication. ArXiv:2004.13876 [cs].

Leanne S. Whitmore, Anthe George, and Corey M. Hudson. 2016. Mapping chemical performance on molecular structures using locally interpretable explanations.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not Explanation. ArXiv:1908.04626 [cs].

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Xi Ye and Greg Durrett. 2022. Can explanations be useful for calibrating black box models? In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6199–6212, Dublin, Ireland. Association for Computational Linguistics.

Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Sai Suggala, David I. Inouye, and Pradeep Ravikumar. 2019. On the (In)fidelity and Sensitivity for Explanations. ArXiv:1901.09392 [cs, stat].

Muhammad Rehman Zafar and Naimul Mefraz Khan. 2019. DLIME: A Deterministic Local Interpretable Model-Agnostic Explanations Approach for Computer-Aided Diagnosis Systems. ArXiv:1906.10263 [cs, stat].

Ofir Zafrir, Ariel Larey, Guy Boudoukh, Haihao Shen, and Moshe Wasserblat. 2021. Prune once for all: Sparse pre-trained language models.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification.

Xingyu Zhao, Xiaowei Huang, Valentin Robu, and David Flynn. 2020. Baylime: Bayesian local interpretable model-agnostic explanations. *CoRR*, abs/2012.03058.

Zhengze Zhou, Giles Hooker, and Fei Wang. 2021. S-lime: Stabilized-lime for model explanation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery amp; Data Mining*, KDD '21. ACM.

A Explanation Methods

A.1 LIME

LIME explanations were generated using the `captum.attr.LimeBase` class from the Captum library. The `LimeBase` class was configured with a Lasso linear model (`SkLearnLasso(alpha=1e-5, random_state=1)`) to serve as the interpretable surrogate model. To generate explanations, a cosine similarity function was used to measure the similarity between the original text embeddings and those of the perturbed versions, based on the model's embedding layers. An exponential function was then applied to this similarity to create weights, enabling the interpretable model to make meaningful comparisons between the original and perturbed inputs per the Captum documentation. The perturbations were generated by randomly selecting each word in the input text with a 50% chance, using a Bernoulli distribution. Finally, these perturbed texts were adjusted to match the original input structure by adding necessary padding. Full code available upon release.

A.2 SHAP

The SHAP explanations were generated using the `captum.attr.ShapleyValueSampling` class from the Captum library. To generate explanations, the `ShapleyValueSampling` method from Captum was employed, which calculates SHAP values by sampling various subsets of input features and estimating their contributions to the model's predictions. In this setup, the baselines were set to `None`, meaning that zero baselines were used by default, and the `feature_mask` was also set to `None`, treating each scalar as an independent feature.

A.3 Integrated Gradients

The explanations were generated using the `captum.attr.LayerIntegratedGradients` class from the Captum library. To generate explanations, we applied the IG method to the model embedding layer, using `<unk>` as the reference token for baseline comparisons. This method calculates attributions by integrating the gradients of the model's output with respect to its input tokens, summed across the embedding dimensions, and normalized for consistency..

B Models

EMBEDDING_SIZE	MAX_EPOCHS	LR
16	1	0.001
16	1	0.01
16	1	0.1
16	5	0.001
16	5	0.01
16	5	0.1
16	10	0.001
16	10	0.01
16	10	0.1
64	1	0.001
64	1	0.01
64	1	0.1
64	5	0.001
64	5	0.01
64	5	0.1
64	10	0.001
64	10	0.01
64	10	0.1
128	1	0.001
128	1	0.01
128	1	0.1
128	5	0.001
128	5	0.01
128	5	0.1
128	10	0.001
128	10	0.01
128	10	0.1

Table 2: 2-Layer NN model variations

EMBEDDING_AND_HIDDEN_DIM	MAX_EPOCHS	LR
16	1	0.001
16	1	0.01
16	1	0.1
16	5	0.001
16	5	0.01
16	5	0.1
16	10	0.001
16	10	0.01
16	10	0.1
64	1	0.001
64	1	0.01
64	1	0.1
64	5	0.001
64	5	0.01
64	5	0.1
64	10	0.001
64	10	0.01
64	10	0.1
128	1	0.001
128	1	0.01
128	1	0.1
128	5	0.001
128	5	0.01
128	5	0.1
128	10	0.001
128	10	0.01
128	10	0.1

Table 3: LSTM model variations

MAX_EPOCHS	LR	Pretrained Model
1	0.001	prajjwal1/bert-tiny
1	0.001	distilbert-base-uncased
1	0.01	prajjwal1/bert-tiny
1	0.01	distilbert-base-uncased
1	0.1	prajjwal1/bert-tiny
1	0.1	distilbert-base-uncased
5	0.001	prajjwal1/bert-tiny
5	0.001	distilbert-base-uncased
5	0.01	prajjwal1/bert-tiny
5	0.01	distilbert-base-uncased
5	0.1	prajjwal1/bert-tiny
5	0.1	distilbert-base-uncased
10	0.001	prajjwal1/bert-tiny
10	0.001	distilbert-base-uncased
10	0.01	prajjwal1/bert-tiny
10	0.01	distilbert-base-uncased
10	0.1	prajjwal1/bert-tiny
10	0.1	distilbert-base-uncased

Table 4: Transformer model variations

C Experiments

C.1 Sparsity

Epochs	Learning Rate
5	0.1
5	0.01
5	0.001
5	0.01
5	0.01
5	0.01
1	0.01
1	0.001
5	0.01
1	0.1

Table 5: DistilBERT configurations sampled for sparsity experiments

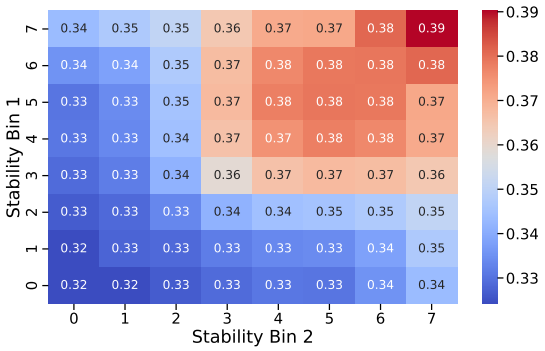

Figure 8: Average feature agreement across all 3 explanation methods for the top 10 tokens

C.2 Disagreement

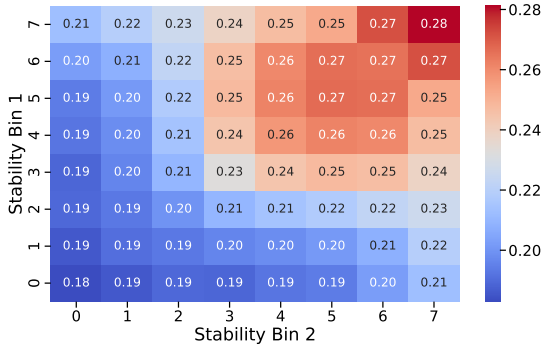

Figure 7: Average feature agreement across all 3 explanation methods for the top 5 tokens

Index	Fisher-Jenks Bin
0	0.70
1	0.73
2	0.75
3	0.77
4	0.78
5	0.80
6	0.82
7	0.84
8	0.89

Table 6: Fischer Jenks Stability bins for the DistilBERT models