Beyond Fooling: Model Manipulation Under Explanation-aware Training

Anonymous ACL submission

Abstract

Feature-level explanations are commonly used to interpret transformer-based NLP models. Still, little is known about how explanation-aware objectives influence model behaviour during training. While prior work has demonstrated training-time manipulation of explanations in vision models, its implications for transformers and token-level explainability remain unexplored. We study training-time manipulation of token-level explanations in transformer-based NLP classifiers and introduce sequence-aware objectives suited to text input. We show that explanation-aware training systematically alters token relevance patterns while largely preserving task accuracy. Importantly, masking and cross-method evaluations reveal that these attribution changes can coincide with shifts in model reliance rather than isolated failures of specific explanation methods. Our results suggest that apparent vulnerabilities of feature-level explanations can reflect deeper model adaptations, underscoring the need to consider learning dynamics when interpreting explanation robustness. Our code is available at https://anonymous.4open.science/r/XAI-fool-388D

1 Introduction

The impact of transformer models in natural language processing has been paramount, and, as such, research on interpretability has also been steadily advancing (Fantozzi and Naldi, 2024). One family of model-specific explanations focuses on feature-level, i.e., token-level, explainability methods that yield saliency or relevance maps (Schneider, 2024). Such explanations are specifically helpful for smaller, task-specific (e.g., classification) models, as they demonstrate granular insight into a model's workings per sample (Liu et al., 2021). However, prior work has demonstrated the instability of such explanations in the image domain, particularly concerning Convolutional Neural Networks (CNNs). Specifically, Heo et al. (2019) demonstrates that explanations can be shaped by an adversary through model manipulation attacks as opposed to, e.g., input-focused attacks via so-called adversarial examples. As such, explainability methods are fooled by training that shapes model explanations through supervised explanation learning.

Explorations of transformer- and NLP-specific cases are scarce, making it unclear whether these vulnerabilities extend to the domain and architecture (Baniecki and Biecek, 2024). We extend Heo et al.'s (2019) framework to transformer models in the NLP context by proposing new attack scenarios and optimisation approaches. We ask the following question: Can transformer-based NLP classifiers be trained to manipulate token-level explanations produced by state-of-the-art transformer explainability methods without degrading task performance?

Related Work. Several transformer-specific explainability methods have been proposed. Earlier methods tend to focus on the attention mechanism as an approach to explainability (Fantozzi and Naldi, 2024). More advanced methods incorporate these transformer-specific elements but go beyond them. For instance, Ali et al. (2022) identified the failure modes of GradientxInput approaches (Denil et al., 2014) using the Layer-wise relevant propagation (LRP) framework (Bach et al., 2015) and demonstrated strong conservation properties through their adaptation. In contrast, Chefer et al. (2021a) generalises the Chefer et al.'s (2021b) method by explicitly building on attention-focused methods and extending them via gradient-based weighting, making the explanations class-specific and yielding the Generic Attention Explainability (GAE). Due to their strong empirical performance and widespread use, these methods form the primary targets of our analysis.

Our Contributions. This work, to our knowledge, is the first to investigate training-time manipulation of token-level explanations in transformer-

based NLP classifiers. We introduce novel sequence-aware attack objectives that adapt model manipulation approaches to text inputs, including ranking-based formulations, which enable stable manipulation of token relevance. Simultaneously, task accuracy is largely preserved with drops of less than 5% and typically 1%. We show that strong transformer-specific explainability methods exhibit systematic changes in attribution patterns under such training, particularly when suppressing highly influential tokens: previously highly relevant tokens rank among the top 10% least relevant tokens after the attack. Importantly, we highlight that these changes are often accompanied by shifts in the model's reliance on alternative input signals, rather than merely misleading a specific explanation method as Heo et al. (2019) claim. Finally, we find that successful manipulations can generalise across explainability methods, indicating shared structural weaknesses. Overall, our findings provide insight into the interaction between explanation methods and learned decision strategies, highlighting the need to consider model adaptation when interpreting explanation robustness.

2 Methodology

We consider an adversary with access to the model training across data and the model to manipulate model explanations at inference time. Moreover, we assume that the adversary is unaware of the samples and the specific explainability techniques at inference time, following related work (Heo et al., 2019). Instead, the learning objective during training is augmented with an explanation-aware token-level loss function to enforce explanations.

2.1 Dual Loss Framework and Optimisations

Model manipulation attacks are implemented using a dual-loss approach (Heo et al., 2019):

$$\mathcal{L}(D, D_{\text{fool}}, I; w, w_0) = \mathcal{L}_C(D; w) + \lambda \mathcal{L}_F^I(D_{\text{fool}}; w, w_0), \quad (1)$$

where $\mathcal{L}_C$ denotes the standard loss function, in this case, cross-entropy loss. $\mathcal{L}_F^I$ denotes the token-level penalty imposed based on the explanation method $\mathcal{I}$ yielding a saliency map. The attack is optimised based on the dataset D_{fool}. This dataset can be sub-sampled from D, the original training set used to finetune a pre-trained model with weights w_0 to obtain a model with weights w. The parameter λ controls the attack strength by regulating the balance between the standard loss $\mathcal{L}_C$ and the attack loss $\mathcal{L}_F^I$.

We instantiate the explanation-aware loss using three classes of objectives that differ in the structure they impose on token relevance. Value-based objectives penalise deviations from target relevance values (e.g., mean squared error or penalties on the magnitude of selected tokens such as top-k), following Heo et al. (2019). Distributional objectives, such as KL divergence and categorical losses, operate on normalised relevance distributions and encourage specific allocations of relevance mass across tokens. Finally, ranking-based objectives constrain the relative ordering of token relevance. Further details can be found in Appendix B

We distinguish between two manipulation goals at the level of token relevance. *Relevance elevation* aims to artificially emphasise a small set of tokens, such as a fixed position or selected token types, relative to the remainder of the input. This goal can be operationalised by enforcing target relevance values, shaping relevance distributions, or constraining relative token importance. It can therefore be instantiated using value-based, distributional, or ranking-based losses. *Relevance suppression*, in contrast, targets tokens that are already highly influential under a reference model and seeks to reduce their relative contribution to the prediction. Since this objective is inherently comparative—requiring influential tokens to be demoted relative to others, it can be expressed through ranking-based constraints or top-k penalties, as used in prior work (Heo et al., 2019). We design our experiments to reflect these distinctions.

2.2 Experimental Evaluation

We structure our experiments around two manipulation goals at the level of token relevance: elevating and suppressing token importance. Tokens are either elevated based on their set position in the input sequence or based on their token ID. For the latter, tokens that are not highly relevant under a reference model are randomly sampled if they appear in a minimum of training samples. Token relevance is decreased for the top-k most relevant tokens per sequence, or for the n tokens that appear most frequently as the most relevant token across the training data of a given dataset. Further details about the implementation are in Appendix D. D_{fool} thus contains all samples in which the top tokens or randomly sampled tokens are present.

All experiments are performed on two datasets

Statistic	SST-2	IMDb
Train size	77,300	25,000
Validation size	872	12,500
Sequence Length:		
Median	10 (24)	233 (232)
Mean	13.3 (25.2)	313.9 (309.5)
Min.	3 (4)	13 (10)
Max.	66 (55)	3127 (3157)
Positives Rate	0.56	0.5

Table 1: Dataset statistics for SST-2 and IMDb. Test set labels are not available for SST-2. The IMDb validation set is created by randomly sampling 50% of the original test set. Length statistics are based on the bert-base-uncased tokeniser for the train set, with validation statistics in parentheses (pre-truncation).

and two models using first-order only optimisations. A simple binary sentiment classification task has been chosen to focus on the model's attribution behaviour. Accordingly, the datasets are the SST-2 and IMDb (Maas et al., 2011; Socher et al., 2013), for which statistics are presented in Table 1. We test all approaches with BERT (Devlin et al., 2019) and ALBERT (Lan et al., 2020) to explore the effects of different architectures via weight sharing. All experiments are implemented using PyTorch (Paszke et al., 2019) and Transformers (Wolf et al., 2020) and run on NVIDIA GeForce RTX 2080 Ti GPUs. Hyperparameters and further details are available in Appendix D. Given that second-order gradients are unstable and memory-intensive (Kashyap, 2023), we treat the gradients introduced by the explanation pass as constants in the optimisation procedure. Taken together, this combination allows us to examine robustness across datasets and architectures without overspecifying the attack setting.

The primary evaluation metric for the attack is the average ranking percentile of attacked tokens, as it reflects the common interpretation of token relevance as ranking (Liu et al., 2021). For each occurrence of a token of interest, the token's percentile rank within the relevance distribution of the given sequence is computed. These percentiles are then averaged across all occurrences in D_{fool} to indicate the average, normalised rank.[1]

Finally, we analyse how explanation-aware training affects model behaviour beyond attribution metrics. We masked all attacked tokens in the validation set to assess shifts in reliance on the core task and to evaluate whether attribution changes persist when explanations are generated using an alternative method not used during training.

[1]For details, see Appendix D.

3 Results & Discussion

Table 2 summarises the main findings for BERT on SST-2, with several consistent trends observed across additional settings reported in Appendix A. All reported results use $\lambda = 10$, which we found to provide a stable balance between task accuracy and attribution manipulation.[2] We evaluate effects on (i) core task accuracy, (ii) changes in token relevance rankings, and (iii) transfer across explainability methods.

Across all settings, explanation-aware training has a limited impact on task performance, with accuracy drops of at most 1–4%. This indicates that models can accommodate explanation-aware constraints without substantially degrading predictive performance. Thus, attribution changes are model adaptations rather than artefacts of task failure.

Manipulation effectiveness differs markedly by goal and model. Relevance suppression is more effective than relevance elevation, with highly ranked tokens consistently demoted to lower percentiles, especially for the LRP-trained models.[3] In contrast, elevation only achieves comparable performance in some settings when trained on GAE. We observe further differences among the explainability methods, especially regarding cross-method generalisation. Only the LRP-trained BERT model manipulations generalise well for token suppression, whereas other settings and models tend to generalise poorly. These systematic asymmetries suggest varying model adaptations across explainability methods.

Ranking-based objectives typically outperform magnitude- and distribution-based alternatives. Unlike value-based losses, ranking constraints directly operate on relative token importance, which aligns with how feature-level explanations are typically interpreted and compared (see, for instance, Ali et al. (2022); Liu et al. (2021). Additional comparisons are provided in Appendix C.

Masking experiments are consistent with the hypothesis that explanation-aware training often alters model reliance rather than merely misleading the explainability method. Under LRP-based training, masking attacked tokens improves and otherwise decreases accuracy, as shown in Table 3. This finding suggests that the model learns to treat them as noise. In contrast, accuracy decreases under GAE-based training, indicating that the model

[2]For a sensitivity analysis, see Appendix B.

[3]GAE-trained attacks on ALBERT are less effective.

Attack Category	Loss	GAE	LRP	Train GAE → LRP	Train LRP → GAE	Accuracy (GAE / LRP)
Elevating token by position	MSE	91.31	59.66	**66.57**	56.21	0.83 / 0.81
	Rank Loss	**95.10**	**68.58**	52.27	**59.82**	0.80 / 0.83
	Baseline	53.53	52.29	52.29	53.53	0.83
Elevating specific tokens	MSE	83.05	53.98	51.04	27.74	0.82 / 0.82
	Rank Loss	**85.66**	**60.42**	**51.38**	**34.08**	0.82 / 0.82
	Baseline	42.30	50.30	50.22	29.85	0.83
Decreasing top token per sample	Top-k Loss	**33.69**	68.58	91.30	84.04	0.82 / 0.80
	Rank Loss	45.32	**37.82**	**91.17**	**77.53**	0.82 / 0.81
	Baseline	100.00	100.00	100.00	100.00	0.83
Decreasing overall top token	Top-k Loss	10.37	51.81	80.87	64.29	0.81 / 0.81
	Rank Loss	**9.68**	**13.21**	**79.90**	**45.95**	0.82 / 0.83
	Baseline	88.35	84.38	78.37	78.25	0.83

Table 2: Percentile rank comparison across explainability methods for the BERT model on SST. Columns report percentile ranks under GAE and LRP, cross-generalisation when training attacks on one method and evaluating with the other, and task accuracy (GAE / LRP). All values are single-run validation set evaluations. The best-performing loss function for each approach is highlighted.

Approach	Standard Validation Acc.	Subsampled Validation Acc.	Masked + Subsampled Acc.
Baseline	0.8257	0.8388	0.7479
LRP	0.8257	0.8375	0.8981
GAE	0.8165	0.8568	0.7768

Table 3: Validation accuracy for standard, subsampled (only samples with overall top tokens), and masked-subsampled evaluation settings for baseline and rank loss approaches. The subsampled dataset only contains samples that contain the top targets.

continues to rely on these tokens despite reduced attribution. Our optimisation procedure offers a plausible explanation for the observed differences in model adaptation and cross-method generalisation. In our setup, optimisation is restricted to first-order gradients. In LRP-based training, these gradients propagate only to the input embeddings, whereas in GAE-based training, they also influence the attention mechanism.

When suppressing token relevance, LRP-based training therefore tends to introduce noise at the embedding level, leading the model to downweight previously dominant tokens without directly modifying the attribution mechanism. This behaviour is consistent with the masking results: performance improves when such tokens are masked, suggesting that the model had over-relied on noisy representations and can fall back on alternative decision cues. In contrast, GAE-based training more directly constrains attribution scores via attention, resulting in reduced apparent relevance while preserving reliance on the same tokens, and consequently lower masked accuracy.

This distinction helps explain cross-method generalisation. Embedding-level adaptations induced by LRP-based training propagate throughout the model and affect attribution behaviour under alternative explainability methods.[4] In contrast, attention-focused adaptations are more tightly coupled to the specific explanation mechanism used during training. This perspective clarifies why relevance elevation is particularly difficult under LRP-based training: artificially amplifying token importance would require either increasing embedding dominance or suppressing competing tokens, conflicting with maintaining task performance.

Taken together, our results show that explanation-aware training in transformer-based NLP models does not uniformly "fool" explainability methods, but could instead induce systematic changes in how models distribute reliance across input features. We find that these changes depend on the interaction between the explanation method and the optimisation objective, with embedding-level adaptations exhibiting stronger cross-method generalisation than attention-focused constraints. Across settings, ranking-based objectives emerge as a particularly effective and stable mechanism for manipulating token-level explanations, as they directly constrain relative token importance in a manner aligned with how feature-level explanations are interpreted in language models. These findings extend Heo et al.'s (2019) prior work on model manipulation to the transformer domain and highlight that apparent explanation vulnerabilities often reflect deeper model adaptations.

[4] This generalisation does not manifest in ALBERT models and could be related to the weight sharing and thus repetitive strong influence of the same weights on the explanation.

Limitations and Ethical Concerns

Our study focuses on a narrow experimental setting: binary sentiment classification using transformer-based classifiers, English-language datasets, and a limited set of token-level explainability methods. While this choice allows for controlled analysis of explanation-aware training, it restricts the generalisability of our findings to other languages, tasks (e.g., sequence labeling or generation), model families, or explanation paradigms. Moreover, we restrict optimisation to first-order gradients and evaluate a small number of datasets and architectures without conducting an extensive hyperparameter search, prioritising interpretability and analytical clarity over exhaustive optimisation. Future work is needed to assess whether the observed adaptation dynamics extend across random seeds, to multilingual settings, more complex tasks, larger models, or alternative optimisation regimes.

The techniques studied in this work have a high potential for misuse. Explanation-aware training could be employed to deliberately obscure model reliance on sensitive or undesirable input features, thereby undermining the use of explainability methods for auditing, accountability, or regulatory compliance. While our goal is to diagnose and understand limitations of token-level explanations, we emphasise that such methods should not be used to evade oversight. We therefore present our results as a cautionary analysis, highlighting the need for robustness-aware interpretability and complementary evaluation strategies when explanations are used in high-stakes settings.

Acknowledgments

The authors used generative AI tools in accordance with the ACL Policy on AI Writing Assistance. Specifically, such tools were used for literature search support; for generating and refining low-novelty text describing pre-existing ideas (which were subsequently verified for accuracy and accompanied by appropriate citations); and for exploratory brainstorming, followed by independent literature review and validation by the authors. In addition, AI-based coding assistants (e.g., predictive code completion and conversational debugging support) were used during software development. All scientific decisions, analyses, interpretations, and final text remain the sole responsibility of the authors.

References

Ameen Ali, Thomas Schnake, Oliver Eberle, Grégoire Montavon, Klaus-Robert Müller, and Lior Wolf. 2022. XAI for transformers: Better explanations through conservative propagation. In *International conference on machine learning*, pages 435–451. PMLR.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140.

Hubert Baniecki and Przemyslaw Biecek. 2024. Adversarial attacks and defenses in explainable artificial intelligence: A survey. *Information Fusion*, 107:102303.

Hila Chefer, Shir Gur, and Lior Wolf. 2021a. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 397–406.

Hila Chefer, Shir Gur, and Lior Wolf. 2021b. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 782–791.

Misha Denil, Alban Demiraj, and Nando De Freitas. 2014. Extraction of salient sentences from labelled documents. *arXiv preprint arXiv:1412.6815*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Paolo Fantozzi and Maurizio Naldi. 2024. The explainability of transformers: Current status and directions. *Computers*, 13(4):92.

Juyeon Heo, Sunghwan Joo, and Taesup Moon. 2019. Fooling neural network interpretations via adversarial model manipulation. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Rohan Kashyap. 2023. A survey of deep learning optimizers – first and second order methods. *Preprint*, arXiv:2211.15596.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Shengzhong Liu, Franck Le, Supriyo Chakraborty, and Tarek Abdelzaher. 2021. On exploring attention-based explanation for transformer models in text classification. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 1193–1203.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and 2 others. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Johannes Schneider. 2024. Explainable generative AI (GenXAI): a survey, conceptualization, and research agenda. *Artificial Intelligence Review*, 57(11):289.

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. HuggingFace's Transformers: State-of-the-art natural language processing. *Preprint*, arXiv:1910.03771.

A Additional Results

This appendix presents results for the additional dataset and model combinations. Table 4 highlights the continued strong performance of LRP-based token suppression for the BERT model. However, the performance of GAE-based training is weaker than on SST-2, indicating that more extended sequences pose challenges for clear optimisation.

Tables 5 and 6 show the results for the ALBERT model. Again, the tables underline the strong performance of LRP-based token suppression, albeit with poorer evidence of generalisation, likely due to the attention mechanism's consistent impact through weight-sharing. They also highlight further difficulties of attacks based on GAE. While the attacks do elevate or suppress target tokens, the magnitude of change, i.e., attack effectiveness, is lower than for BERT-based experiments. However, generalisation appears stronger, indicating LRP's high sensitivity to attention weights.

Attack Category	Loss	GAE	LRP	Train GAE → LRP	Train LRP → GAE	Accuracy (GAE / LRP)
Elevating token by position						
	MSE	41.44	49.31	48.07	40.76	0.87 / 0.87
	Rank Loss	**99.14**	**53.30**	**56.97**	**45.61**	0.87 / 0.87
	Baseline	40.65	48.58	48.58	40.65	0.88
Elevating specific tokens						
	MSE	47.36	52.15	51.63	44.97	0.87 / 0.87
	Rank Loss	**71.32**	**52.66**	**59.68**	**45.99**	0.87 / 0.87
	Baseline	42.03	48.79	52.26	42.51	0.88
Decreasing top token per sample						
	Top-k Loss	**76.89**	66.78	**93.60**	84.46	0.87 / 0.87
	Rank Loss	81.21	**42.97**	93.67	**79.70**	0.87 / 0.87
	Baseline	100.00	100.00	100.00	100.00	0.88
Decreasing overall top tokens						
	Top-k Loss	**32.98**	48.62	74.96	55.05	0.87 / 0.87
	Rank Loss	36.02	**32.03**	**74.86**	**49.47**	0.87 / 0.87
	Baseline	91.58	88.35	94.45	84.04	0.88

Table 4: Percentile rank comparison across explainability methods for the BERT model on IMDb. Columns report percentile ranks under GAE and LRP, cross-generalisation when training attacks on one method and evaluating with the other, and task accuracy (GAE / LRP). All values are single-run validation set evaluations.

Attack Category	Loss	GAE	LRP	Train GAE → LRP	Train LRP → GAE	Accuracy (GAE / LRP)
Elevating token by position						
	MSE	77.16	67.68	**52.66**	66.30	0.86 / 0.84
	Rank Loss	**93.45**	**69.04**	51.16	**83.73**	0.86 / 0.82
	Baseline	50.90	53.01	53.01	50.90	0.86
Elevating specific tokens						
	MSE	44.81	57.14	50.51	45.42	0.86 / 0.83
	Rank Loss	**48.59**	**63.73**	**51.33**	**50.38**	0.86 / 0.83
	Baseline	41.43	53.45	50.20	41.96	0.86
Decreasing top token per sample						
	Top-k Loss	**78.82**	53.64	**49.50**	90.10	0.87 / 0.83
	Rank Loss	84.41	**16.59**	52.39	**87.32**	0.86 / 0.84
	Baseline	100.00	100.00	100.00	100.00	0.86
Decreasing overall top tokens						
	Top-k Loss	**75.47**	47.02	**49.69**	55.42	0.86 / 0.85
	Rank Loss	76.66	**20.77**	52.08	**53.03**	0.85 / 0.85
	Baseline	90.89	62.32	58.52	59.80	0.86

Table 5: Percentile rank comparison across explainability methods for the ALBERT model on SST. Columns report percentile ranks under GAE and LRP, cross-generalisation when training attacks on one method and evaluating with the other, and task accuracy (GAE / LRP). All values are single-run validation set evaluations.

Attack Category	Loss	GAE	LRP	Train GAE → LRP	Train LRP → GAE	Accuracy (GAE / LRP)
Elevating token by position						
	MSE	76.08	52.87	52.66	78.58	0.90 / 0.90
	Rank Loss	**99.17**	**64.07**	**54.78**	**96.60**	0.90 / 0.89
	Baseline	73.71	52.47	52.47	73.71	0.90
Elevating specific tokens						
	MSE	57.86	**52.87**	**54.84**	**38.82**	0.90 / 0.90
	Rank Loss	**58.84**	51.28	49.63	35.29	0.91 / 0.89
	Baseline	50.67	48.36	51.81	37.25	0.90
Decreasing top token per sample						
	Top-k Loss	94.99	47.86	**25.11**	92.95	0.90 / 0.90
	Rank Loss	**82.63**	**13.00**	53.10	**88.66**	0.90 / 0.89
	Baseline	100.00	100.00	100.00	100.00	0.90
Decreasing overall top tokens						
	Top-k Loss	59.11	53.37	62.26	**56.59**	0.90 / 0.90
	Rank Loss	**53.63**	**11.72**	**61.78**	86.13	0.90 / 0.88
	Baseline	62.66	68.35	54.74	64.34	0.90

Table 6: Percentile rank comparison across explainability methods for the ALBERT model on IMDb. Columns report percentile ranks under GAE and LRP, cross-generalisation when training attacks on one method and evaluating with the other, and task accuracy (GAE / LRP). All values are single-run validation set evaluations.

B Loss Function Formulations

Generally, the adversary's goal can be pursued by optimising the objective function below, adopted from (Heo et al., 2019):

$$L(D, D_{\text{fool}}, I; w, w_0) = L_C(D; w) + \lambda \mathcal{L}_F^I(D_{\text{fool}}; w, w_0), \quad (2)$$

where

- $\mathcal{L}_C$ denotes the standard loss function, in this case, cross-entropy loss.
- $\mathcal{L}_F^I$ denotes the penalty imposed based on the explanation, specific to
- D_{fool}, the dataset used to optimise the attack, and potentially sub-sampled from
- D, the original training set used to finetune, some pre-trained model with weights
- w_0 to reach a model with weights w.
- λ controls the attack strength by regulating the balance between the standard loss $\mathcal{L}_C$ and the attack loss $\mathcal{L}_F^I$.

$\mathcal{L}_F^I$ can be adapted to meet a number of requirements. In the following, we detail multiple approaches to increase the relevance of specific tokens, decrease the relevance of specific tokens, or generally learn specific, uninformative distributions as explanations.

B.1 Increasing Token Relevance

Token relevance can be increased by targeting either specific tokens or specific positions of the input. Both approaches can be optimised via a number of different losses, which are shown below.

Mean Squared Error

Mean Squared Error (MSE) is shown in Eq. (3) and is a common loss function for regression tasks. However, it can also be used to reduce the distance between an explanation h_c^I and a target mask m for a given sample of length T as shown in (Heo et al., 2019). Unlike images, samples in NLP tasks vary in length. As such, the aggregation method across N samples can be crucial. Hence, we propose experimenting with both micro- and macro-averaging.

$$\text{MSE}_i = \frac{1}{T_i} \sum_{t=1}^{T_i} \left(h_{c_i,t}^I(w, x_i) - m_{i,t}\right)^2. \quad (3)$$

MSE Loss - Macro

Macro-averaging as shown in Eq. (4) implies that each sample bears equal weight in the size of the penalty and has the advantage that longer samples do not exert overly strong influence over the overall learning process.

$$\begin{aligned} \mathcal{L}_{\text{MSE}_{\text{macro}}}^I &= \frac{1}{N} \sum_{i=1}^{N} \text{MSE}_i \\ &= \frac{1}{N} \\ &\quad \sum_{i=1}^{N} \left(\frac{1}{T_i} \sum_{t=1}^{T_i} \left(h_{c_i,t}^I(w, x_i) - m_{i,t}\right)^2 \right). \end{aligned} \quad (4)$$

MSE Loss - Micro

Contrastingly, micro-averaging as shown in Eq. (5) allows for fine-grained feedback per token, allowing the relevance of tokens in long sequences to not be suppressed.

$$\mathcal{L}^{I}_{\text{MSE}_{\text{micro}}} = \frac{1}{\sum_{i=1}^{N} T_i} \sum_{i=1}^{N} \sum_{t=1}^{T_i} \left(h^{I}_{c_i,t}(w, x_i) - m_{i,t} \right)^2. \quad (5)$$

Categorical Approaches

Given that only a single or very few tokens are emphasised, loss functions originating from more categorical approaches are also suitable candidates. However, unlike purely categorical losses, setting the remaining tokens to zero will likely yield unstable results due to strong gradient feedback. As such, we propose to adapt cross-entropy loss (CE loss) to a harder and softer version of the KL-divergence. Below, we define these two loss functions for this case.

For a sequence x_i of length T_i with class label c_i, let $h^{I}_{c_i}(w, x_i) \in \mathbb{R}^{T_i}$ denote the relevance scores produced by interpretation method I for model parameters w. First, we define a temperature-scaled softmax distribution of relevance scores over tokens:

$$p_{i,t} = \frac{\exp\left(h^{I}_{c_i,t}(w, x_i)/\tau_p\right)}{\sum_{s=1}^{T_i} \exp\left(h^{I}_{c_i,s}(w, x_i)/\tau_p\right)},$$

$$t = 1, \ldots, T_i.$$

In the hard-target case, we normalise the target mask into a categorical distribution:

$$q_{i,t} = \frac{\text{target}_{i,t}}{\sum_{s=1}^{T_i} \text{target}_{i,s}}, \qquad t = 1, \ldots, T_i.$$

Alternatively, in the soft-target case, the target scores are treated similarly to logits and converted into a probability distribution via a soft-max function using a temperature τ_q:

$$q_{i,t} = \frac{\exp\left(\text{target}_{i,t}/\tau_q\right)}{\sum_{s=1}^{T_i} \exp\left(\text{target}_{i,s}/\tau_q\right)}, \qquad t = 1, \ldots, T_i.$$

Finally, the loss is computed as the KL divergence between both distributions, with the difference being whether the target distributions have been softened or not:

$$\begin{aligned} \mathcal{L}^{I}_{\text{KL}} &= \frac{1}{N} \sum_{i=1}^{N} \text{KL}(q_i \,\|\, p_i) \\ &= \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T_i} q_{i,t} \, \log \frac{q_{i,t}}{p_{i,t}}. \end{aligned} \quad (6)$$

Generally, a setup of $\tau_p > \tau_q$ leads to a spikier target distribution q, enforcing a strong signal yet a smoother input distribution p to avoid exploding feedback, especially given that relevance scores will likely not all go to exactly 0, potentially leading the model to react in a stable manner. Accordingly, temperature parameters are set at $\tau_p = 2$ and $\tau_q = 1$, unless otherwise specified. This argument can be extended to the hard case where both the temperature smoothing and softmaxing are entirely absent.

Ranking Approach

Finally, given that only a few tokens are present in some samples, and the main goal is relative ranking rather than overall magnitude, ranking approaches appear suitable. As such, we propose using margin ranking loss as an additional approach to improve token-level relevance.

In this approach, all possible pairs of tokens of a sample i that can be made from the two sets P_i (set of target tokens either to suppress or elevate, depending on y) and N_i (set of remaining tokens) are compared, and a loss is formed using a hinge function as shown in Eq. (7). The parameter m enforces a margin between emphasised and de-emphasised tokens, set at 0.5 for all experiments. Beyond that, the meaning of P_i and N_i can be flipped through y. If $y = 1$, then $p \in P_i$ should be ranked above $n \in N_i$. This setting applies to token elevation experiments. If $y = -1$, then the opposite applies: $n \in N_i$ should be ranked above $p \in P_i$, i.e., in the case of token suppression. The resulting loss is normalised by the number of samples N and the number of pairs. Overall, such a loss does not enforce the overall magnitude but rather a token ranking with an additional gap between positive token p and negative tokens n, allowing the model more flexibility yet enforcing the desired explainability characteristics.

$$\mathcal{L}_{\text{rank}} = \frac{1}{\sum_{i=1}^{N} |P_i|\,|N_i|} \sum_{i=1}^{N} \sum_{p \in P_i} \sum_{n \in N_i} \max\Big(0,\ -y\Big(\begin{matrix} h^I_{c_i,p}(w, x_i) \\ - h^I_{c_i,n}(w, x_i) \end{matrix}\Big) + m\Big). \tag{7}$$

All of the approaches above can be "plugged" into the overall objective shown in Eq. (2). Thereby, the tokens whose relevance should be increased are emphasised relative to the relevance of the remaining tokens. By doing so, the approaches cover a wide range of loss functions, such as regression and categorical approaches to ranking loss, allowing for wide exploration. However, they share the commonality that they attack the relevance scores through an overall approach across a given sample by increasing and decreasing a token's desired relevance, depending on the token's status. If successful, such an approach can be used to emphasise any arbitrary token.

B.2 Decreasing Token Relevance

Clearly, the opposite of increasing token relevance is decreasing token relevance. However, in contrast to the above approach, decreasing token relevance should be more targeted, since this attack is explicitly aimed at high-relevance tokens. Hence, the loss functions can accordingly be more targeted than learning entire distributions, as is the case with the categorical and regression approaches from above. Two approaches that aim to fulfil this promise are introduced below.

Top-k Loss

First, simply taking the absolute magnitude of the top k tokens can serve as an adequate penalty to decrease that magnitude. This approach is inspired by the top-k loss of (Heo et al., 2019). However, given the limited number of tokens as compared to pixels in an image, k indicates the number of tokens, rather than the percentage.

$$\mathcal{L}_{\text{Top-}k} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{k} \sum_{t \in \text{Top-}k\left(h^I_{c_i}(w_0, x_i)\right)} \left| h^I_{c_i,t}(w, x_i) \right|. \tag{8}$$

Ranking Loss

Similar to the approach above, top tokens can be decreased explicitly in terms of ranking. As such, they can be emphasised by setting $y = -1$ and targeting them as part of P_i. Therefore, the optimisation follows the same principles and directly uses Eq. (7).

Both approaches can be applied to different definitions of top tokens. On the one hand, a top token can be understood as the highest-ranking token per sample. On the other hand, it can also be a number of selected tokens, such as the most important tokens across a reference corpus and model. In that case, D_{fool} is subsampled to those samples that contain the relevant tokens.

C Additional Loss Functions & Sensitivity Analysis Results

Sensitivity Analysis LRP

Tables 7 to 10 show that accuracy tends to decrease as λ increases. This effect becomes noticeable at $\lambda = 10$. Thus, further increases in the attack strength are not advisable to preserve task accuracy. The opposite trend holds for the attack evaluations, with noticeable jumps in losses with higher attack emphasis.

Furthermore, Table MSE macro (see Tables 8 and 7) and top-k (see Tables 9 and 10) losses tend to optimise magnitude-based losses best, whereas ranking loss is typically most efficient for ranking-based objectives.

Sensitivity Analysis GAE

The results of LRP are mirrored for GAE, as can be seen in Tables 11 and 12 for token elevation, and in Tables 13 and 14 for token suppression.

λ	Attack $\mathcal{L}$	Accuracy	MSE Macro	MSE Micro	Rank Loss
0.1	KL Hard	0.830	0.115	0.086	0.508
0.1	KL Soft	0.828	0.115	0.086	0.516
0.1	MSE Macro	0.830	0.112	0.084	0.513
0.1	MSE Micro	0.827	0.115	0.086	0.517
0.1	Rank	0.830	0.115	0.086	0.506
1	KL Hard	0.825	0.130	0.097	0.470
1	KL Soft	0.828	0.110	0.083	0.503
1	MSE Macro	0.820	0.093	0.073	0.492
1	MSE Micro	0.828	0.109	0.082	0.511
1	Rank	0.826	0.115	0.088	0.462
10	KL Hard	0.823	0.204	0.161	0.435
10	KL Soft	0.818	0.092	0.073	0.467
10	MSE Macro	0.805	0.076	0.062	0.465
10	MSE Micro	0.817	0.091	0.070	0.489
10	Rank	0.826	0.096	0.081	0.416

Table 7: *Attack evaluation metrics of the attack on the first token of the sequence by λ and attack loss using LRP.* The table shows the results aiming to increase the relevance of the first token with the Bert model on SST-2 with standard hyperparameters as described above. Results are shown for the validation set.

λ	Attack $\mathcal{L}$	Accuracy	MSE Macro	MSE Micro	Rank Loss
0	-	0.826	0.292	0.272	0.534
0.1	KL Soft	0.827	0.291	0.271	0.533
0.1	MSE Macro	0.829	0.286	0.268	0.531
0.1	MSE Micro	0.828	0.291	0.271	0.533
0.1	Rank	0.826	0.291	0.271	0.530
1	KL Soft	0.829	0.283	0.267	0.530
1	MSE Macro	0.822	0.263	0.255	0.517
1	MSE Micro	0.828	0.282	0.266	0.530
1	Rank	0.817	0.288	0.270	0.504
10	KL Soft	0.818	0.258	0.251	0.508
10	MSE Macro	0.817	0.246	0.244	0.495
10	MSE Micro	0.820	0.262	0.252	0.516
10	Rank	0.819	0.266	0.260	0.462

Table 8: *Attack evaluation metrics of the relevance elevation on specific tokens by λ and attack loss using LRP.* The table shows the results aiming to increase the relevance of the chosen tokens with the Bert model on SST-2 with standard hyperparameters as described above. Results are shown for the entire validation set.

λ	Attack $\mathcal{L}$	Accuracy	Top k Loss	Rank Loss
0	-	0.815	0.579	0.898
0.1	Rank Loss	0.830	0.519	0.741
0.1	Top k Loss	0.827	0.445	0.765
1	Rank Loss	0.830	0.346	0.478
1	Top k Loss	0.826	0.210	0.629
10	Rank Loss	0.811	0.344	0.395
10	Top k Loss	0.799	0.143	0.580

Table 9: *Attack evaluation metrics of the attack on the highest ranking token of a sequence by λ and attack loss using LRP.* The table shows the results aiming to decrease the relevance of the highest ranking token with the Bert model on SST-2 with standard hyperparameters as described above. Results are shown for the validation set.

λ	Attack $\mathcal{L}$	Accuracy	Top k Loss	Rank Loss
0	-	0.826	0.235	0.608
0.1	Rank Loss	0.823	0.348	0.433
0.1	Top k Loss	0.826	0.171	0.598
1	Rank Loss	0.826	0.504	0.243
1	Top k Loss	0.819	0.098	0.513
10	Rank Loss	0.826	0.621	0.156
10	Top k Loss	0.811	0.071	0.505

Table 10: *Attack evaluation metrics of the attack on the top ten highest ranking tokens of the training set by λ and attack loss using LRP.* The table shows the results aiming to decrease the relevance of these tokens with the Bert model on SST-2 with standard hyperparameters as described above. Results are shown for the full validation set.

λ	Attack $\mathcal{L}$	Accuracy	MSE Macro	MSE Micro	Rank Loss
0	-	0.826	0.079	0.064	0.530
0.1	KL Hard	0.827	0.083	0.068	0.533
0.1	KL Soft	0.827	0.083	0.068	0.534
0.1	MSE Macro	0.827	0.083	0.068	0.534
0.1	MSE Micro	0.828	0.083	0.068	0.534
0.1	Rank	0.826	0.083	0.068	0.533
1	KL Hard	0.822	0.082	0.067	0.524
1	KL Soft	0.827	0.083	0.067	0.533
1	MSE Macro	0.826	0.081	0.066	0.530
1	MSE Micro	0.827	0.083	0.067	0.533
1	Rank	0.826	0.081	0.066	0.504
10	KL Hard	0.797	0.075	0.062	0.424
10	KL Soft	0.820	0.079	0.065	0.517
10	MSE Macro	0.833	0.078	0.064	0.501
10	MSE Micro	0.827	0.079	0.065	0.526
10	Rank	0.804	0.077	0.063	0.430

Table 11: *Attack evaluation metrics of the attack on the first token of the sequence by λ and attack loss using GAE.* The table shows the results aiming to increase the relevance of the first token with the Bert model on SST-2 with standard hyperparameters as described above. Results are shown for the validation set.

λ	Attack $\mathcal{L}$	Accuracy	MSE Macro	MSE Micro	Rank Loss
0	-	0.826	0.079	0.064	0.530
0.1	KL Hard	0.825	0.263	0.255	0.444
0.1	KL Soft	0.827	0.263	0.255	0.444
0.1	MSE Macro	0.825	0.263	0.255	0.444
0.1	MSE Micro	0.827	0.263	0.255	0.444
0.1	Rank	0.827	0.263	0.255	0.444
1	KL Hard	0.821	0.262	0.254	0.443
1	KL Soft	0.821	0.263	0.255	0.444
1	MSE Macro	0.817	0.261	0.253	0.440
1	MSE Micro	0.822	0.263	0.254	0.444
1	Rank	0.819	0.260	0.252	0.439
10	KL Hard	0.808	0.253	0.247	0.427
10	KL Soft	0.822	0.258	0.251	0.436
10	MSE Macro	0.818	0.252	0.246	0.425
10	MSE Micro	0.827	0.257	0.250	0.433
10	Rank	0.819	0.251	0.245	0.424

Table 12: *Attack evaluation metrics of the relevance elevation on specific tokens by λ and attack loss using GAE.* The table shows the results aiming to increase the relevance of the chosen tokens with the Bert model on SST-2 with standard hyperparameters as described above. Results are shown for the entire validation set.

λ	Attack $\mathcal{L}$	Accuracy	Top k Loss	Rank Loss
0	-	0.826	0.046	0.515
0.1	Rank Loss	0.822	0.044	0.513
0.1	Top k Loss	0.826	0.044	0.513
1	Rank Loss	0.815	0.027	0.497
1	Top k Loss	0.813	0.026	0.497
10	Rank Loss	0.819	0.000	0.473
10	Top k Loss	0.820	0.000	0.475

Table 13: *Attack evaluation metrics of the attack on the highest ranking token of a sequence by λ and attack loss using GAE.* The table shows the results aiming to decrease the relevance of the highest ranking token with the Bert model on SST-2 with standard hyperparameters as described above. Results are shown for the validation set.

λ	Attack $\mathcal{L}$	Accuracy	Top k Loss	Rank Loss
0.1	Rank Loss	0.828	0.019	0.490
0.1	Top k Loss	0.827	0.018	0.490
1	Rank Loss	0.818	0.008	0.480
1	Top k Loss	0.823	0.007	0.480
10	Rank Loss	0.817	0.000	0.474
10	Top k Loss	0.808	0.000	0.475

Table 14: *Attack evaluation metrics of the attack on the top ten highest ranking tokens of the training set by λ and attack loss using GAE.* The table shows the results aiming to decrease the relevance of these tokens with the Bert model on SST-2 with standard hyperparameters as described above. Results are shown for the full validation set.

D Experimental Setup Details

Hyperparameters

We experiment with a set of standard hyperparameters for comparability. These include a learning rate of 1e-5 with a linear scheduler and 10% warmup steps. The model is optimised for four epochs using AdamW and a batch size of 32, or 8 for IMDb and Albert, due to memory constraints. These choices are based on previously found good performance for these models and tasks.[5] We use the standard tokenisers, with truncation to the models' maximum sequence length (512) from the beginning of each sequence.[6] All experiments are run with first-order only optimisation, i.e. treating gradients introduced by the explanation pass as constants. In practice, we detach gradients from the explanation's backward pass from the computation graph. All reported experimental results are single runs based on truncated sequences with the given hyperparameters.

Details Elevating Token Relevance

Increasing token relevance is split into two experiments: increasing tokens by their position and increasing specific tokens. In terms of position, the token at the first position is chosen, i.e., the first token following the CLS token, resulting in D_{fool} being the full dataset. This token is typically uninformative for sentiment analysis and, therefore, suitable for showing the effectiveness of the attack.

Choosing specific tokens to increase their relevance artificially is primarily guided by the size of D_{fool}. Based on initial tuning, ten tokens are sampled from all tokens that appear in at least 25% of the training samples. Additionally, tokens should not be in the top 10 most relevant tokens per class (see below for details). Suppose only an insufficient number of tokens can be sampled. In that case, the threshold for minimum frequency is lowered, leading to a threshold of 10% or 15% for the SST dataset, depending on the explainability method.[7]The sizes of the resulting attack sets D_{fool} can be found below in Table 15. Once a token's position is found, a mask highlighting it can be created. This mask is a binary relevance sequence that can then be used as a target explanation for the optimisation functions shown above.

Details Decreasing Token Relevance

High-relevance tokens are either identified per sample or across a corpus based on a reference model. In the former case, simply the highest positive token is identified per sample and targeted with the losses detailed above. Thus, D_{fool} is of the same size as the overall dataset. In the latter case, the top token per sample is recorded based

[5] See here for a number of different successfully finetuned models https://huggingface.co/textattack/models.

[6] Head truncation may underrepresent later evidence.

[7] Tokens are chosen based on the model-specific tokenisers.

on the training set and an unattacked model. The 10 most frequently appearing tokens are saved per class, dataset, model, and tokeniser.[8] The sizes of the resulting attack sets D_{fool} can be found below in Table 15. Once the tokens and their positions are identified, a binary mask can be made. In this binary mask, the highly relevant tokens are marked, and their relevance magnitudes are minimised using the optimisation functions above.

Evaluation Metric Details

We compute ranking percentiles at the level of token occurrences. For each sequence, we identify all tokens belonging to a predefined set of tokens of interest. For each such token occurrence (i, j), we compute its percentile rank within the same sequence as

$$p_{i,j} = \frac{100}{|\mathcal{T}_i|} \sum_{k \in \mathcal{T}_i} \mathbb{I}[r_{i,k} \leq r_{i,j}]$$

where $r_{i,k}$ denotes the relevance score of token j in sequence i, and T_i is the set of tokens in that sequence. Percentiles are computed over signed relevance scores for the target class, including negative scores (evidence for the opposite class). This yields a normalised score in $[0, 100]$ indicating how highly the token ranks relative to all other tokens in the sequence. The final metric is obtained by averaging these percentile scores across all token occurrences, allowing multiple tokens of interest to contribute within the same sequence.

Sizes of Attack Dataset.

[8] Explanations are with regard to a specific class. We use the target class. As such, the highest-ranking tokens are understood to have the highest positive relevance. Negative relevance implies, in a binary problem, that this token has positive relevance for the opposite class, which is clearly separated here; therefore, the signed magnitude, not the absolute magnitude, is used.

Approach	Model	Dataset	Expl. Method	Train.	Val.	Train Frac.	Val Frac.
Overall Top Tokens	BERT	SST-2	LRP	44,595	726	0.66	0.83
	BERT	SST-2	GAE	22,473	475	0.33	0.54
	ALBERT	SST-2	LRP	67,349	872	1.00	1.00
	ALBERT	SST-2	GAE	67,349	872	1.00	1.00
	BERT	IMDb	LRP	24,966	12,486	1.00	1.00
	BERT	IMDb	GAE	22,831	11,345	0.91	0.91
	ALBERT	IMDb	LRP	25,000	12,500	1.00	1.00
	ALBERT	IMDb	GAE	25,000	12,500	1.00	1.00
Token Elevation	BERT	SST-2	LRP	47,243	866	0.70	0.99
	BERT	SST-2	GAE	67,349	872	1.00	1.00
	ALBERT	SST-2	LRP	30,666	661	0.46	0.76
	ALBERT	SST-2	GAE	67,349	872	1.00	1.00
	BERT	IMDb	LRP	24,974	12,488	1.00	1.00
	BERT	IMDb	GAE	24,464	12,204	0.98	0.98
	ALBERT	IMDb	LRP	16,674	8,256	0.67	0.66
	ALBERT	IMDb	GAE	18,015	9,062	0.72	0.72

Table 15: D_{fool} training and validation set coverage by approach, model, dataset, and explanation method (target tokens differ). The table shows wide coverage for almost all experiments.