An Unsupervised Approach to Achieve Supervised-Level Explainability in Healthcare Records

Joakim Edin[1,3,*] **Maria Maistro**[1] **Lars Maaløe**[3]
Lasse Borgholt[3] **Jakob D. Havtorn**[3] **Tuukka Ruotsalo**[1,2]

University of Copenhagen[1] LUT University[2] Corti[3]

Abstract

Electronic healthcare records are vital for patient safety as they document conditions, plans, and procedures in both free text and medical codes. Language models have significantly enhanced the processing of such records, streamlining workflows and reducing manual data entry, thereby saving healthcare providers significant resources. However, the black-box nature of these models often leaves healthcare professionals hesitant to trust them. State-of-the-art explainability methods increase model transparency but rely on human-annotated evidence spans, which are costly. In this study, we propose an approach to produce plausible and faithful explanations without needing such annotations. We demonstrate on the automated medical coding task that adversarial robustness training improves explanation plausibility and introduce AttInGrad, a new explanation method superior to previous ones. By combining both contributions in a fully unsupervised setup, we produce explanations of comparable quality, or better, to that of a supervised approach. We release our code and model weights. [1]

1 Introduction

Explainability in natural language processing remains a largely unsolved problem, posing significant challenges for healthcare applications (Lyu et al., 2023). For every patient admission, a healthcare professional must read extensive documentation in the healthcare records to assign appropriate medical codes. A code is a machine-readable identifier for a diagnosis or procedure, pivotal for tasks such as statistics, documentation, and billing. This process can involve sifting through thousands of words to choose from over 140,000 possible codes (Johnson et al., 2016), making medical coding not only time-consuming but also error-prone (Burns et al., 2012; Tseng et al., 2018).

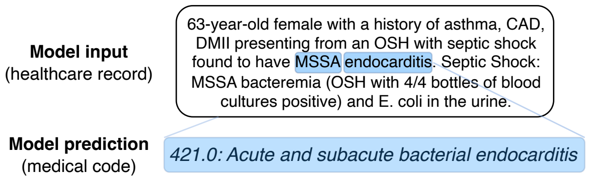

Model input
(healthcare record)
63-year-old female with a history of asthma, CAD, DMII presenting from an OSH with septic shock found to have MSSA endocarditis. Septic Shock: MSSA bacteremia (OSH with 4/4 bottles of blood cultures positive) and E. coli in the urine.
Model prediction
(medical code)
421.0: Acute and subacute bacterial endocarditis

Figure 1: Example of an input, prediction, and feature attribution explanation highlighted in the input.

Automated medical coding systems, powered by machine learning models, aim to alleviate these burdens by suggesting medical codes based on free-form written documentation. However, when reviewing suggested codes, healthcare professionals must still manually locate relevant evidence in the documentation. This is a slow and strenuous process, especially when dealing with extensive documentation and numerous medical codes. Explainability is essential for making this process tractable.

Feature attributions, a common form of explainability, can help healthcare professionals quickly find the evidence necessary to review a medical code suggestion (see Figure 1). Feature attribution methods score each input feature based on its influence on the model's output. These explanations are often evaluated through *plausibility* and *faithfulness* (Jacovi and Goldberg, 2020). Plausibility measures how convincing an explanation is to human users, while faithfulness measures the explanation's ability to reflect the model's logic.

While previous work has proposed feature attribution methods for automated medical coding, they only evaluated attention-based feature attribution methods. Furthermore, the state-of-the-art method uses a supervised approach relying on costly evidence-span annotations. This reliance on manual annotations significantly limits practical applicability, as each code system and its versions

[*] Correspondance to `je@corti.ai`

[1] `https://github.com/JoakimEdin/explainable-medical-coding`

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 4869–4890
November 12-16, 2024 ©2024 Association for Computational Linguistics

require separate manual annotations (Mullenbach et al., 2018; Teng et al., 2020; Dong et al., 2021; Kim et al., 2022; Cheng et al., 2023).

In this study, we present an approach for producing explanations of comparable quality to the supervised state-of-the-art method but without using evidence span annotations. We implement adversarial robustness training strategies to decrease the model's dependency on irrelevant features, thereby avoiding such features in the explanations (Tsipras et al., 2018). Moreover, we present more faithful feature attribution methods than the attention-based method used in previous studies. Our key contributions are:

1. We show that adversarially robust models produce more plausible medical coding explanations.
2. We propose a new feature attribution method, AttInGrad, which produces substantially more faithful and plausible medical coding explanations than previous methods.
3. We demonstrate that the combination of an adversarial robust model and AttInGrad produces medical coding explanations of similar, or better, plausibility and faithfulness compared to the supervised state-of-the-art approach.

2 Related work

Next, we present explainability approaches for automated medical coding and previous work on how adversarial robustness affects explainability.

2.1 Explainable automated medical coding

Automated medical coding is a multi-label classification task that aims to predict a set of medical codes from J classes based on a given medical document (Edin et al., 2023). In this context, the objective of explainable automated medical coding is to generate feature attribution scores for each of the J classes. These scores quantify how much each input token influences each class's prediction.

Most studies in explainable automated medical coding use attention weights as feature attribution scores without comparing to other methods (Mullenbach et al., 2018; Teng et al., 2020; Dong et al., 2021; Feucht et al., 2021; Cheng et al., 2023). However, two studies suggest alternative feature attribution methods. Xu et al. (2019) propose a feature attribution method tailored to their one-layer CNN architecture but do not compare performance with other methods. Kim et al. (2022) train a linear medical coding model using knowledge distillation and use its weights as the explanation. However, their method does not improve over the explanations of the popular attention approach. Cheng et al. (2023) improve the plausibility of the attention weights of the final layer by training them to align with evidence span annotations. However, obtaining such annotations is costly.

Previous work focused on plausibility using human ratings (Mullenbach et al., 2018; Teng et al., 2020; Kim et al., 2022), example inspection (Dong et al., 2021; Feucht et al., 2021; Liu et al., 2022), or evidence span overlap metrics (Xu et al., 2019; Cheng et al., 2023). Notably, no studies have assessed the faithfulness of the explanations nor compared the attention-based methods with other established methods.

2.2 Adversarial robustness and explainability

Adversarial robustness refers to the ability of a machine learning model to maintain performance under adversarial attacks, which involve making small changes to input data that do not significantly affect human perception or judgment (e.g., a small amount of image noise). Tsipras et al. (2018) and Ilyas et al. (2019) demonstrate that adversarial examples exploit the models' dependence on fragile, non-robust features. Adversarial robustness training embeds invariances to prevent models relying on such non-robust features, with regularization and data augmentation as main strategies (Tsipras et al., 2018; Ros and Doshi-Velez, 2018).

Previous work in image classification shows that adversarially robust models generate more plausible explanations (Ros and Doshi-Velez, 2018; Chen et al., 2019; Etmann et al., 2019). These studies demonstrate this phenomenon for three adversarial training strategies: 1) input gradient regularization, which improves the Lipschitzness of neural networks (Drucker and Le Cun, 1992; Ros and Doshi-Velez, 2018; Chen et al., 2019; Etmann et al., 2019; Rosca et al., 2020; Fel et al., 2022; Khan et al., 2023), 2) adversarial training, which trains models on adversarial examples, thereby embeds invariance to adversarial noise (Tsipras et al., 2018), and 3) feature masking, which masks unimportant features during training to embed invariance to such features (Bhalla et al., 2023).

The relationship between model robustness and explanation plausibility in Natural Language Processing (NLP) remains unclear, primarily due to the fundamental differences between textual tokens and image pixels. To date, only two studies, Yoo and Qi (2021) and Li et al. (2023), have explored this relationship in depth. These studies suggest that robust models produce more faithful explanations compared to their non-robust counterparts. However, their conclusions may be questionable due to the use of the Area Over the Perturbation Curve (AOPC) metric, which is unsuitable for cross-model comparisons (Edin et al., 2024). This inappropriate application of AOPC may have led to potentially misleading conclusions.

3 Methods

Here, we describe the adversarial robustness training strategies and feature attribution methods in the context of a prediction model for medical coding. The underlying automated medical coding model takes a sequence of tokens as input and outputs medical code probabilities (Section 4.2).

3.1 Adversarial robustness training strategies

We implemented three adversarial training strategies, which we hypothesized could decrease our medical coding model's reliance on irrelevant tokens: Input gradient regularization, projected gradient descent, and token masking. We chose these strategies because they have been shown to improve plausibility in image classification and faithfulness in text classification (Li et al., 2023)

Input gradient regularization (IGR) encourages the gradient of the output with respect to the input to be small. This aims to decrease the number of features on which the model relies, encouraging it to ignore irrelevant words (Drucker and Le Cun, 1992). We adapt IGR to text classification by adding to the task's binary cross-entropy loss, L_{BCE}, the ℓ^2 norm of the gradient of L_{BCE} wrt. the input token embedding sequence $\boldsymbol{X} \in \mathbb{R}^{N \times D}$. This yields the total loss,

$$L_{\text{BCE}}(f(\boldsymbol{X}), \boldsymbol{y}) + \lambda_1 \left\| \nabla_{\boldsymbol{X}} L_{\text{BCE}}(f(\boldsymbol{X}), \boldsymbol{y}) \right\|_2 \ ,$$

where $\boldsymbol{y} \in \mathbb{R}^J$ is a binary target vector representing the J medical codes, λ_1 is a hyperparameter, and $f : \mathbb{R}^{N \times D} \to \mathbb{R}^J$ is the classification model.

Projected gradient descent (PGD) increases model robustness by training with adversarial examples, thereby promoting invariance to such inputs (Madry, 2018). We hypothesized that PGD reduces the model's reliance on irrelevant tokens, as adversarial examples often arise from the model's use of such unrobust features (Tsipras et al., 2018). PGD aims to find the noise $\boldsymbol{\delta} \in \mathbb{R}^{N \times D}$ that maximizes the loss $L_{\text{BCE}}(f(\boldsymbol{X}+\boldsymbol{\delta}), \boldsymbol{y})$ while satisfying the constraint $\|\boldsymbol{\delta}\|_\infty \leq \epsilon$, where ϵ is a hyperparameter. PGD was originally designed for image classification; we adapted it to NLP by adding the noise to the token embeddings $\boldsymbol{X}$. We implemented PGD as follows,

$$\boldsymbol{Z}^* = \arg\max_{\boldsymbol{Z}} L_{\text{BCE}}(f(\boldsymbol{X} + \boldsymbol{\delta}(\boldsymbol{Z})), \boldsymbol{y}) \ ,$$

and enforced the constraint $\|\boldsymbol{\delta}\|_\infty \leq \epsilon$ by parameterizing $\boldsymbol{\delta}(\boldsymbol{Z}) = \epsilon \tanh(\boldsymbol{Z})$ and optimising $\boldsymbol{Z} \in \mathbb{R}^{N \times D}$ directly. We initialized $\boldsymbol{Z}$ with zeros. Finally, we tuned the model parameters using the following training objective:

$$L_{\text{BCE}}(f(\boldsymbol{X}), \boldsymbol{y}) + \lambda_2 L_{\text{BCE}}(f(\boldsymbol{X} + \boldsymbol{\delta}(\boldsymbol{Z}^*)), \boldsymbol{y}) \ ,$$

where λ_2 is a hyperparameter.

Token masking (TM) teaches the model to predict accurately while using as few features as possible, thereby encouraging the model to ignore irrelevant words. TM uses a binary mask to occlude unimportant tokens and train the model to rely only on the remaining tokens (Bhalla et al., 2023; Tomar et al., 2023). Inspired by Bhalla et al. (2023), we employed a two-step teacher-student approach. We used two copies of the same model already trained on the automated medical coding task: a teacher f_t with frozen model weights and a student f_s, which we fine-tuned. For each training batch, the first step was to learn a sparse mask $\hat{\boldsymbol{M}} \in [0,1]^{N \times D}$, that still provided enough information to predict the correct codes by minimizing:

$$\|\hat{\boldsymbol{M}}\|_1 + \beta \|f_s(\boldsymbol{X}) - f_s(x_m(\boldsymbol{X}, \hat{\boldsymbol{M}}))\|_1 \ ,$$

where β is a hyperparameter and $x_m : \mathbb{R}^{N \times D} \to \mathbb{R}^{N \times D}$ is the masking function:

$$x_m(\boldsymbol{X}, \boldsymbol{M}) = \boldsymbol{B} \odot (1 - \boldsymbol{M}) + \boldsymbol{X} \odot \boldsymbol{M} \ ,$$

where $\boldsymbol{B} \in \mathbb{R}^{N \times D}$ is the baseline input. We chose $\boldsymbol{B}$ as the token embedding representing the start token, followed by the mask token embedding repeated $N - 2$ times, followed by the end token embedding. After optimization, we binarized the mask $\boldsymbol{M} = \text{round}(\hat{\boldsymbol{M}})$, where around 90% of the features were masked. Finally, we tuned the model f_s using the following training objective:

$$\|f_s(\boldsymbol{X}) - f_t(\boldsymbol{X})\|_1 + \lambda_3 \|f_s(\boldsymbol{X}) - f_s(x_m(\boldsymbol{X}, \boldsymbol{M}))\|_1 \ ,$$

where λ_3 is a hyperparameter.

3.2 Feature attribution methods

We evaluated several feature attribution methods for automated medical coding, categorizing them into three types: attention-based, gradient-based, and perturbation-based (more details in Appendix B). **Attention-based** methods like Attention (Mullenbach et al., 2018), Attention Rollout (Abnar and Zuidema, 2020), and AttGrad (Serrano and Smith, 2019) rely on the model's attention weights. **Gradient-based** methods such as InputXGrad (Sundararajan et al., 2017), Integrated Gradients (IntGrad) (Sundararajan et al., 2017), and Deeplift (Shrikumar et al., 2017) use backpropagation to quantify the influence of input features on outputs. **Perturbation-based** methods, including LIME (Ribeiro et al., 2016), KernelSHAP (Lundberg and Lee, 2017), and Occlusion@1 (Ribeiro et al., 2016), measure the impact on output confidence by occluding input features.

Our investigation into feature attribution methods revealed an intriguing pattern: while individual methods often produced unreliable explanations, their shortcomings rarely overlapped. Attention-based methods and gradient-based approaches like InputXGrad frequently disagreed on which tokens were most important, yet both contributed valuable insights in different scenarios. This observation sparked a key question: could we leverage the complementary strengths of these methods to create a more robust attribution technique?

To address this question, we propose AttInGrad, a novel feature attribution method that combines Attention and InputXGrad. AttInGrad multiplies their respective attribution scores, aiming to amplify the importance of tokens deemed relevant by both methods while down-weighting those highlighted by only one or neither method.

We formalize the AttInGrad attribution scores for class j using the following equation:

$$\begin{bmatrix} \boldsymbol{A}_{j1} \cdot \left\| \boldsymbol{X}_1 \odot \frac{\partial f_j}{\partial \boldsymbol{X}_1}(\boldsymbol{X}) \right\|_2 \\ \vdots \\ \boldsymbol{A}_{jN} \cdot \left\| \boldsymbol{X}_N \odot \frac{\partial f_j}{\partial \boldsymbol{X}_N}(\boldsymbol{X}) \right\|_2 \end{bmatrix},$$

where $\boldsymbol{A} \in \mathbb{R}^{J \times N}$ is the attention matrix, $\odot$ is the element-wise matrix multiplication operation, N are the number of tokens in a document, and J is the number of classes.

In Section 5, we will provide an in-depth analysis of the mechanisms underlying AttInGrad's effectiveness, shedding light on why this combination of methods yields improved feature attributions.

Table 1: The two data splits used in this paper. MIMIC-III full comprises discharge summaries annotated with ICD-9 codes. MDACE comprises discharge summaries annotated with ICD-9 codes and evidence spans.

Split	Train	Val	Test
MIMIC-III full	47,719	1,631	3,372
MDACE	181	60	61

4 Experimental setup

In the following, we present our datasets, models, and evaluation metrics.

4.1 Data

We conducted our experiments using the open-access MIMIC-III and the newly released MDACE dataset (Johnson et al., 2016; Cheng et al., 2023). MIMIC-III[2] includes 52,722 discharge summaries from the Beth Israel Deaconess Medical Center's ICU, collected between 2008 and 2016 and annotated with ICD-9 codes. MDACE comprises 302 reannotated MIMIC-III cases, adding evidence spans to indicate the textual justification for each medical code. Not all possible evidence spans are annotated; for example, if hypertension is mentioned multiple times, only the first mention might be annotated, leaving subsequent mentions unannotated. We focused exclusively on discharge summaries, as most previous medical coding studies on MIMIC-III (Teng et al., 2022). Statistics are in Table 1.

For dataset splits, we used MIMIC-III full, a popular split by Mullenbach et al. (2018), and MDACE, introduced by Cheng et al. (2023) for training and evaluating explanation methods. All MDACE examples are from the MIMIC-III full test set, which we excluded from this test set when using MDACE in our training data.

4.2 Models

We used PLM-ICD, a state-of-the-art automated medical coding model architecture, for our experiments because its architecture is simple while outperforming other models according to Huang et al. (2022); Edin et al. (2023). To address stability

[2] We decided to use MIMIC-III instead of the newer MIMIC-IV because we wanted to use the same dataset as Cheng et al. (2023).

issues caused by numerical overflow in the decoder of the original model, we replaced the label-wise attention mechanism with standard cross-attention (Vaswani et al., 2017). This adjustment not only stabilized training but also slightly improved performance. We provide further details on the architecture modifications in Appendix A.

We compared five models: $\mathrm{B_U}$, $\mathrm{B_S}$, IGR, PGD, and TM. All models used our modified PLM-ICD architecture but were trained differently. $\mathrm{B_U}$ was trained unsupervised with binary cross-entropy, whereas $\mathrm{B_S}$ employed a supervised auxiliary training objective that minimized the KL divergence between the model's cross-attention weights and annotated evidence spans, as per Cheng et al. (2023). IGR, PGD, and TM training is as in Section 2.2. Best hyperparameters are in Appendix D.

4.3 Experiments

We trained all five models with ten seeds on the MIMIC-III full and MDACE training set. The supervised training strategy $\mathrm{B_S}$ used the evidence span annotations, while the others only used the medical code annotations. For each model, we evaluated the plausibility and faithfulness of the explanations generated by every explanation method.

We aimed to demonstrate a similar explanation quality as a supervised approach but without training on evidence spans. Therefore, after evaluating the models and explanation methods, we compared our best combination with the supervised strategy proposed by Cheng et al. (2023), who used the $\mathrm{B_S}$ model and the Attention explanation method. We also compared our best combination with the unsupervised strategy used by most previous works (see Section 2.1), comprising the $\mathrm{B_U}$ model and the Attention explanations method.

4.4 Evaluation metrics

We measured the explanation quality using metrics estimating plausibility and faithfulness. Plausibility measures how convincing an explanation is to human users, while faithfulness measures how accurate an explanation reflects a model's true reasoning process (Jacovi and Goldberg, 2020).

Plausibility metrics Our plausibility metrics measured the overlap between explanations and annotated evidence-spans. We assumed that a high overlap indicated plausible explanations for medical coders. We identified the most important tokens using feature attribution scores, applying a decision boundary for classification metrics, and selecting the top K scores for ranking metrics.

For classification metrics, we used Precision (P), Recall (R), and F1 scores, selecting the decision boundary that yielded the highest F1 score on the validation set (Cheng et al., 2023). Additionally, we included four more classification metrics: Empty explanation rate (Empty), Evidence span recall (SpanR), Evidence span cover (Cover), and Area Under the Precision-Recall Curve (AUPRC). Empty measures the rate of empty explanations when all attribution scores in an example are below the decision boundary. SpanR measures the percentage of annotated evidence spans where at least one token is classified correctly. Cover measures the percentage of tokens in an annotated evidence span that are classified correctly, given that at least one token is predicted correctly. AUPRC represents the area under the precision-recall curve generated by varying the decision boundary from zero to one.

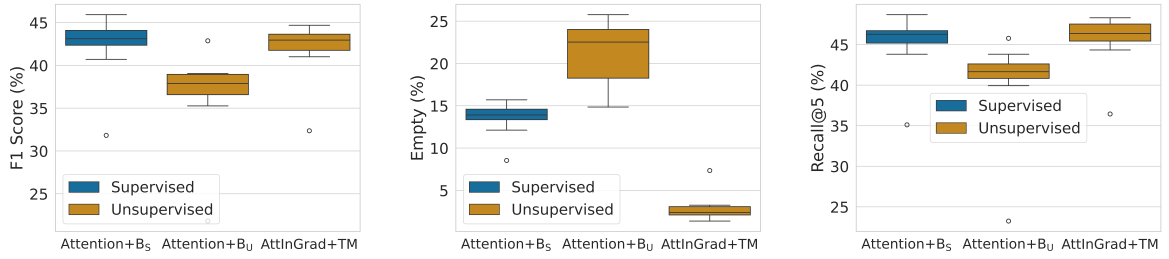
For ranking metrics, we selected the top K tokens with the highest attribution scores, using Recall@K, Precision@K, and Intersection-Over-Unions (IOU) (DeYoung et al., 2020).

Faithfulness metrics We use two metrics to approximate faithfulness: Sufficiency and Comprehensiveness (DeYoung et al., 2020); more details are in Appendix C. Faithful explanations yield high Comprehensiveness and low Sufficiency scores. A high Sufficiency score indicates that many important tokens are incorrectly assigned low attribution scores, while a low Comprehensiveness score suggests that many non-important tokens are incorrectly assigned high attribution scores.

5 Results

Next, we present experimental results for the different training strategies and explainability methods.

Rivaling supervised methods in explanation quality The objective of this paper was to produce high-quality explanations without relying on evidence span annotations. In Figure 2, we compare the plausibility of our approach (Token masking and AttnInGrad) with the unsupervised approach ($\mathrm{B_U}$ and Attention) and supervised state-of-the-art approach ($\mathrm{B_S}$ and Attention). Our approach was substantially more plausible than the unsupervised on all metrics. Compared with the supervised, our approach achieved similar F1 and Recall@5 and substantially better Empty scores.



(a) F1 score ↑

(b) Empty ↓

(c) Recall@5 ↑

Figure 2: Comparison of plausibility across various combinations of explanation methods and models from this study and previous work. Most previous studies used Attention and a standard medical coding model ($\mathrm{B_U}$). Cheng et al. (2023) instead used a supervised model trained on evidence-span annotations ($\mathrm{B_S}$). We proposed AttInGrad and an adversarial robust model (TM).

The supervised approach achieved similar plausibility to ours on most metrics (see Table 2). Our approach also achieved the highest comprehensiveness and lowest sufficiency scores (see Figure 3). The difference was larger in the sufficiency scores, where the supervised score was twice as high as ours.

Adversarial robustness improves plausibility We evaluated the explanation plausibility of every model and explanation method combination in Table 2. IGR and TM outperformed the baseline model $\mathrm{B_U}$ on most metrics and explanation methods. In Appendix E.5, we compare the unsupervised models on a bigger test set and see similar results. The supervised model $\mathrm{B_S}$ yielded better results for attention-based explanations but was weaker than the robust models when using the gradient-based explanation methods: InputXGrad, IG, and Deeplift.

AttInGrad is more plausible and faithful than Attention AttInGrad was more plausible than all other explanation methods across all training strategies and metrics. Notably, plausibility improvements were particularly significant, with relative gains exceeding ten percent in most metrics (see Table 2 and Appendix E.2). For instance, for $\mathrm{B_U}$, AttInGrad reduced the Empty metric from 21.1% to 3.0% and improved the Cover metric from 63.4% to 74.9%. However, these enhancements were less pronounced for $\mathrm{B_S}$, the supervised model.

AttInGrad was also more faithful than Attention (see Figure 3). However, while AttInGrad surpassed the gradient-based methods in comprehensiveness, its sufficiency scores were slightly worse.

Analysis of attention-based explanations While AttInGrad and Attention were more plausible than the gradient-based explanations, they had a three-fold higher inter-seed variance. We found that they often attributed high importance to tokens devoid of alphanumeric characters such as Ġ[, *, and Ċ, which we classify as *special tokens*. These special tokens, such as punctuation and byte-pair encoding artifacts, rarely carry semantic meaning. In the MDACE test set, they accounted for 32.2% of all tokens, compared to just 5.8% within the annotated evidence spans, suggesting they are unlikely to be relevant evidence.

In Figure 4, we analyze the relationship between explanation quality (y-axis) and the proportion of the top five most important tokens that are special tokens (x-axis). Each data point represents the average statistics across the MDACE test set for one seed/run of the $\mathrm{B_U}$ model. Figures 4a and 4b show F1 (plausibility) and comprehensiveness (faithfulness) respectively. For Attention and AttInGrad, we see strong negative correlations for both metrics with a large inter-seed variance. The regression lines fitted on Attention and AttInGrad overlap, with the data points from AttInGrad shifted slightly towards the upper left, indicating attribution of less importance to special tokens.

Conversely, for InputXGrad, we see a moderate negative correlation for the F1 score and no correlation for comprehensiveness. Furthermore, InputXGrad demonstrates a small inter-seed variance, where the proportion of special tokens more closely mirrors that observed in the evidence spans.

We hypothesized that AttInGrad's improvements over Attention stem from InputXGrad reducing special tokens' attribution scores. We tested this by zeroing out these tokens' scores. While it sub-

Table 2: Plausibility of Attention, InputXGrad, Integrated Gradients (IntGrad), Deeplift, and AttInGrad on the MDACE test set. Each experiment was run with ten different seeds. We show the mean of the seeds ± as the standard deviation. All the scores are presented as percentages. Bold numbers outperform the unsupervised baseline model, while underlined numbers outperform the supervised model. We included more feature attribution methods in Appendix E.2.

		Prediction							Ranking		
Explainer	Model	P ↑	R ↑	F1 ↑	AUPRC ↑	Empty ↓	SpanR ↑	Cover ↑	IOU ↑	P@5 ↑	R@5 ↑
Attention	B_S	41.7±5.0	42.9±3.2	42.2±3.7	37.3±3.9	13.5±1.9	60.5±3.7	72.3±3.5	46.1±4.5	32.3±2.6	45.3±3.7
	B_U	35.9±5.9	37.7±5.7	36.6±5.3	31.4±6.4	21.1±3.6	53.2±7.4	63.4±6.8	39.3±6.9	28.8±4.2	40.3±5.9
	IGR	35.5±6.2	**40.4±6.1**	**37.7±5.9**	**32.4±6.8**	**19.4±2.1**	**55.5±8.0**	**64.5±6.2**	**40.0±7.4**	**29.2±4.5**	**40.9±6.3**
	TM	**36.5±5.9**	**37.8±6.1**	**37.0±5.5**	**31.7±6.7**	21.7±3.3	**53.3±7.7**	63.3±6.9	**40.1±7.3**	**29.1±4.4**	**40.8±6.2**
	PGD	33.0±8.4	**38.4±8.0**	35.4±8.3	30.0±9.1	**19.1±1.6**	51.5±13.6	61.1±9.9	35.9±10.8	27.7±6.3	38.9±8.8
InputXGrad	B_S	30.7±2.0	33.7±2.6	32.0±1.4	24.9±2.0	7.3±1.3	48.5±2.8	64.0±2.8	32.2±2.1	26.6±1.2	37.3±1.6
	B_U	31.0±2.6	32.7±3.2	31.6±1.2	24.8±1.9	9.5±2.6	46.4±3.6	62.3±3.5	31.7±1.8	26.3±0.9	36.9±1.3
	IGR	**32.4±2.5**	**33.8±2.2**	**33.0±0.7**	**27.0±0.7**	9.7±2.0	**48.5±2.6**	**64.6±2.5**	**32.8±1.0**	**27.4±0.4**	**38.4±0.5**
	TM	**32.4±2.7**	**34.1±2.1**	**33.1±1.4**	**26.2±2.3**	**8.6±1.6**	**48.5±2.4**	**64.8±2.5**	**32.6±1.5**	**27.4±1.1**	**38.3±1.5**
	PGD	30.1±2.2	**32.9±1.7**	31.4±1.4	24.8±1.5	**9.3±2.2**	**46.6±2.1**	**62.6±2.6**	31.4±1.3	26.1±1.0	36.6±1.4
IntGrad	B_S	30.9±3.2	33.9±2.0	32.2±1.7	26.2±2.6	4.8±2.0	50.8±2.4	65.9±3.3	34.2±2.8	27.4±1.6	38.4±2.2
	B_U	31.3±4.3	32.8±2.7	31.9±3.2	26.0±4.3	5.2±1.0	49.4±3.0	64.4±3.1	33.9±4.2	26.5±2.5	37.1±3.6
	IGR	**32.6±2.2**	**33.8±2.3**	**33.1±1.3**	**26.6±2.0**	5.2±1.7	**49.7±2.5**	**64.7±3.3**	**34.9±2.3**	**27.6±0.9**	**38.7±1.2**
	TM	**33.1±3.6**	**34.0±3.7**	**33.4±2.8**	**27.5±3.7**	5.4±1.6	**51.0±3.9**	**65.5±4.1**	**35.3±4.2**	**27.6±2.1**	**38.7±3.0**
	PGD	30.0±5.4	**33.5±3.3**	31.4±4.2	25.4±4.8	**5.1±1.8**	**49.7±3.6**	**64.5±3.9**	33.6±4.8	26.4±3.0	36.9±4.2
Deeplift	B_S	29.2±2.3	34.2±3.0	31.3±1.2	24.1±1.8	6.4±1.8	48.9±3.4	65.2±3.3	31.1±1.9	26.2±1.1	36.7±1.5
	B_U	31.2±1.9	31.4±2.4	31.2±1.5	24.1±2.1	9.1±1.6	45.1±2.5	61.7±2.9	30.9±1.6	25.8±1.1	36.1±1.5
	IGR	**31.0±1.8**	**33.7±1.4**	**32.2±0.6**	**25.9±0.9**	**8.6±1.0**	**48.3±1.5**	**64.8±1.7**	**31.6±1.0**	**26.7±0.5**	**37.4±0.7**
	TM	**30.2±2.4**	**34.8±1.4**	**32.2±1.4**	**25.1±2.5**	**6.8±1.5**	**49.1±1.8**	**65.7±1.6**	**31.4±1.8**	**26.6±1.1**	**37.3±1.6**
	PGD	29.4±2.8	**32.8±1.5**	30.9±1.5	23.9±1.5	**8.1±1.8**	**46.7±1.8**	**63.3±2.2**	30.7±1.4	25.5±0.9	35.8±1.2
AttInGrad	B_S	40.6±2.2	45.9±3.1	43.0±1.9	38.8±2.1	1.2±0.5	63.9±2.9	79.0±1.9	45.7±2.6	33.3±1.4	46.6±1.9
	B_U	40.7±2.9	42.5±4.4	41.5±3.2	37.1±3.6	3.0±1.3	59.3±5.4	74.9±5.2	43.5±4.0	32.0±2.1	44.9±3.0
	IGR	38.8±4.5	**44.0±4.0**	41.2±4.0	**37.2±4.5**	**2.3±0.7**	**60.3±5.3**	74.5±4.6	43.3±4.9	31.9±2.7	44.7±3.8
	TM	40.2±3.0	**43.9±4.8**	**41.9±3.4**	**37.8±3.9**	**2.8±1.6**	**60.5±5.8**	**75.9±5.3**	**44.0±3.9**	**32.5±2.3**	**45.5±3.2**
	PGD	37.1±8.2	42.5±5.7	39.3±7.1	34.7±8.6	**2.3±0.9**	57.7±9.1	72.7±7.4	39.6±10.2	30.9±4.8	43.3±6.7

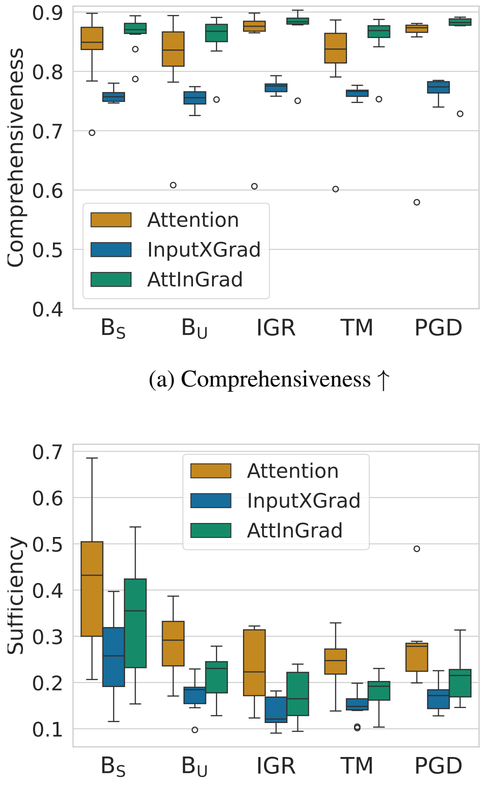
stantially enhanced Attention's F1 score, Attention remained lower than AttInGrad (see Table 3). If AttInGrad's sole contribution were filtering special tokens, we would expect similar F1 scores after zeroing their attributions. The fact that AttInGrad still outperforms Attention after controlling for special tokens suggests that there are additional factors beyond special token filtering contributing to AttInGrad's improved performance.

6 Discussion

Do we need evidence span annotations? We demonstrated that we could match the explanation quality of Cheng et al. (2023) but without supervised training with evidence-span annotations (see Section 5). This raises the question: are evidence-span annotations unnecessary?

Intuitively, training a model on evidence spans should encourage it to use features relevant to humans, thereby making its explanations more plausible. However, we hypothesize that the training strategy used by Cheng et al. (2023) primarily addresses the shortcomings of attention-based explanation methods rather than enhancing the model's underlying logic. The model B_S only produced more plausible explanations with attention-based feature attribution methods (see Figure 2). If the model truly leveraged more informative features, we would expect to see improvements across various feature attribution methods. Additionally, the differences between Attention and AttInGrad were negligible for B_S compared to the other models. This may suggest that the supervised training might have corrected some of the inherent issues in the Attention method, similar to what AttInGrad achieves.

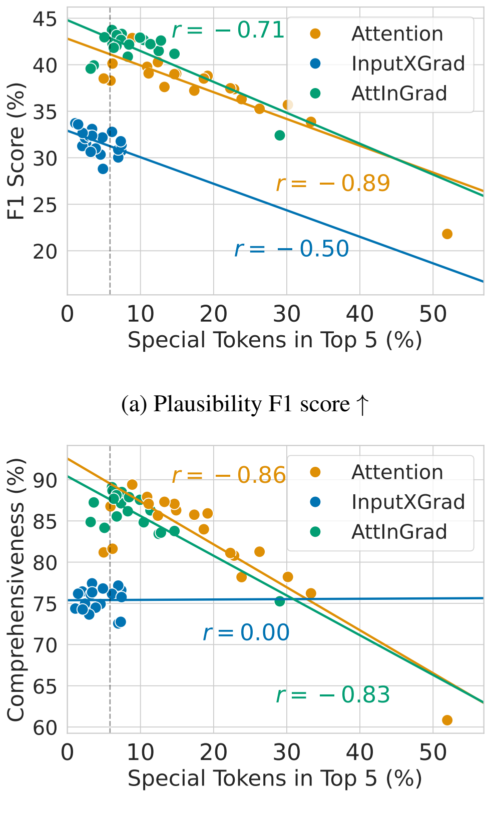
Adversarial robustness training strategies' impact on explanation plausibility While IGR and TM generated more plausible explanations than B_U, our evidence is insufficient to conclude whether the improvements were caused by our adversarially robust models relying on fewer irrelevant features. The adversarial robustness training strategies, especially PGD, had a larger impact on the plausibility of the explanations in previous image classification studies (Tsipras et al., 2018). We speculate that this discrepancy is caused by the inherent differences in the text and image modalities, causing techniques designed for image classifiers to be less effective



(a) Comprehensiveness ↑

(b) Sufficiency ↓

Figure 3: Faithfulness of Attention, InputXGrad, and AttInGrad across models.



(a) Plausibility F1 score ↑

(b) Comprehensiveness ↑

Figure 4: The relationship between explanation quality and the proportion of the top five most important tokens that are special tokens (tokens devoid of alphanumeric characters). Each data point is the average statistic on the MDACE test set for a seed of B_U. We fitted a linear regression for each explanation method and calculated the Pearson correlation (r). The dotted vertical lines represent the proportion of special tokens in the evidence-span annotations.

Table 3: Impact on F1 score of zeroing out feature attribution scores of special tokens for B_U.

	Before	After
Attention	36.5±5.4	40.2±3.9
InputXGrad	31.6±1.2	31.7±1.2
AttInGrad	41.5±4.4	42.4±2.4

for text classifiers (Etmann et al., 2019).

Limitations of attention-based explanations Despite Attention and AttInGrad outperforming other methods in plausibility and faithfulness, they exhibited significant shortcomings, including high sufficiency and inter-seed variation. These findings align with previous research questioning the faithfulness of solely relying on final layer attention weights (Jain and Wallace, 2019).

We hypothesize these limitations stem from misalignment between the positions of the original tokens and their encoded representations. Our analysis (Section 5) suggests the encoder may store contextual information in uninformative tokens, such as special tokens, which are then used by the final attention layer for classification. As the training loss does not penalize where contextualized information is placed, this location can vary across training iterations, leading to the observed high inter-seed variance in attention-based explanations.

Training strategies that enforce alignment between original tokens and their encoded representations could alleviate the limitations of Attention and AttInGrad. This alignment might explain the benefits of the supervised training strategy proposed by Cheng et al. (2023). However, rather than restricting the model, future research should explore feature attribution methods that incorporate information from all transformer layers, not just the final one (Kobayashi et al., 2021). Although attention rollout, a method incorporating all attention layers, proved unsuccessful in our experiments (see Appendix E.2), recent studies have highlighted its shortcomings and proposed alternative feature attribution methods that may be more suitable for our task (Modarressi et al., 2022, 2023).

Recommendations Similar to Lyu et al. (2023), we advocate that future research on feature attribution methods prioritize enhancing their faithfulness, as focusing solely on plausibility can yield misleading explanations. When models misclassify or rely on irrelevant features, explanations can only appear plausible if they ignore the model's actual reasoning process. Overemphasizing plausibility may inadvertently lead researchers to favor approaches that produce explanations disconnected from the model's true reasoning.

Instead, we propose that researchers prioritize improving the faithfulness of feature attribution methods while also working to align the model's reasoning process with that of humans. This approach not only enhances the plausibility and faithfulness of explanations but also contributes to the accuracy and robustness of model classifications.

7 Conclusion

Our goal was to enhance the plausibility and the faithfulness of explanations without evidence-span annotations. We found that training our model using input gradient regularization or token masking resulted in more plausible gradient-based explanations. We proposed a new explanation method, AttInGrad, which was substantially more plausible and faithful than the attention-based explanation method used in previous studies. By combining the best training strategy and explanation method, we showed results of similar quality to a supervised baseline (Cheng et al., 2023).

Limitations

Our study did not conclusively show why adversarial robustness training strategies improved the explanation plausibility. We hypothesized that these strategies force the model to rely on fewer features that weakly correlate with the labels, and such features are less plausible. However, validating this hypothesis proved challenging. Our analysis of feature attributions' entropy was inconclusive, as detailed in Appendix E.4. Moreover, we did not know which features the model relied on because this would require a perfect feature attribution method, which is what we aimed to develop. Despite these challenges, we demonstrated that the adversarial robust models produced more plausible explanations. We believe that our work has laid a solid foundation for future research into how model training strategies can impact explanation plausibility.

Our study's scope was constrained to a single data source (Beth Israel Deaconess Medical Center's MIMIC-III and MDACE) and one model architecture (PLM-ICD). The effectiveness of Attention and AttInGrad may vary with different architectures, particularly those employing multi-head attention and skip connections in the final layer. Further research is needed to investigate the generalizability of our findings across diverse model architectures, medical coding systems, languages, and healthcare institutions.

Furthermore, the limited size of the MDACE test set constrained our study, resulting in low statistical power for many experiments. Despite the desire to conduct more trials with various seeds, we limited ourselves to ten seeds per training strategy due to the high computational costs involved. Conducting more experiments or expanding the test set might have revealed nuances and differences that our initial setup failed to detect. Nevertheless, our results across runs, explanation methods, and analysis point in the same direction. Moreover, while the test set in the main paper only comprises 61 examples, each example contains 14 medical codes, each annotated with multiple evidence spans, providing greater statistical power. Finally, our comparison of the unsupervised approaches on the larger test set in Appendix E.5 demonstrated similar results as on the smaller test set in the main paper. We, therefore, believe that our claims in this paper are well substantiated with empirical evidence.

Ethics statement

Healthcare costs are continuously increasing worldwide, with administrative costs being a significant contributing factor (Tseng et al., 2018). In this paper, we propose methods that may help reduce these administrative costs by making the review of medical code suggestions easier and faster. The aim of this paper was to develop technology to assist medical coders in performing tasks faster instead of replacing them.

Plausible but unfaithful explanations may risk convincing medical coders to accept medical code suggestions that are incorrect, thereby risking the patient's safety (Jacovi and Goldberg, 2020). We, therefore, advocate faithfulness to be of higher priority than in previous studies.

Electronic healthcare records contain private information. The healthcare records in MIMIC-III, the dataset used in this paper, have been anonymized and stored in encrypted data storage accessible only to the main author, who has a license to the dataset and HIPAA training.

Acknowledgements

This research was partially funded by the Innovation Fund Denmark via the Industrial Ph.D. Program (grant no. 2050-00040B) and Research Council of Finland (grant no. 322653). We thank Jonas Lyngsø for insightful discussions and code for dataloading. Furthermore, we thank Simon Flachs, Lana Krumm, and Andreas Geert Motzfeldt for revisions.

References

Samira Abnar and Willem Zuidema. 2020. Quantifying Attention Flow in Transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4197, Online. Association for Computational Linguistics.

Usha Bhalla, Suraj Srinivas, and Himabindu Lakkaraju. 2023. Discriminative Feature Attributions: Bridging Post Hoc Explainability and Inherent Interpretability. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*.

E.M. Burns, E. Rigby, R. Mamidanna, A. Bottle, P. Aylin, P. Ziprin, and O.D. Faiz. 2012. Systematic review of discharge coding accuracy. *Journal of Public Health (Oxford, England)*, 34(1):138–148.

Jiefeng Chen, Xi Wu, Vaibhav Rastogi, Yingyu Liang, and Somesh Jha. 2019. Robust Attribution Regularization. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, Vancouver, Canada.

Hua Cheng, Rana Jafari, April Russell, Russell Klopfer, Edmond Lu, Benjamin Striner, and Matthew Gormley. 2023. MDACE: MIMIC Documents Annotated with Code Evidence. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7534–7550, Toronto, Canada. Association for Computational Linguistics.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A Benchmark to Evaluate Rationalized NLP Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online. Association for Computational Linguistics.

Hang Dong, Víctor Suárez-Paniagua, William Whiteley, and Honghan Wu. 2021. Explainable automated coding of clinical notes using hierarchical label-wise attention networks and label embedding initialisation. *Journal of Biomedical Informatics*, 116:103728.

H. Drucker and Y. Le Cun. 1992. Improving generalization performance using double backpropagation. *IEEE Transactions on Neural Networks*, 3(6):991–997.

Joakim Edin, Alexander Junge, Jakob D. Havtorn, Lasse Borgholt, Maria Maistro, Tuukka Ruotsalo, and Lars Maaløe. 2023. Automated Medical Coding on MIMIC-III and MIMIC-IV: A Critical Review and Replicability Study. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, pages 2572–2582, New York, NY, USA. Association for Computing Machinery.

Joakim Edin, Andreas Geert Motzfeldt, Casper L. Christensen, Tuukka Ruotsalo, Lars Maaløe, and Maria Maistro. 2024. Normalized AOPC: Fixing Misleading Faithfulness Metrics for Feature Attribution Explainability. *Preprint*, arXiv:2408.08137.

Christian Etmann, Sebastian Lunz, Peter Maass, and Carola Schoenlieb. 2019. On the Connection Between Adversarial Robustness and Saliency Map Interpretability. In *Proceedings of the 36th International Conference on Machine Learning*, pages 1823–1832. PMLR.

Thomas Fel, David Vigouroux, Remi Cadene, and Thomas Serre. 2022. How Good is your Explanation? Algorithmic Stability Measures to Assess the Quality of Explanations for Deep Neural Networks. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1565–1575, Waikoloa, HI, USA. IEEE.

Malte Feucht, Zhiliang Wu, Sophia Althammer, and Volker Tresp. 2021. Description-based Label Attention Classifier for Explainable ICD-9 Classification. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 62–66, Online. Association for Computational Linguistics.

Chao-Wei Huang, Shang-Chi Tsai, and Yun-Nung Chen. 2022. PLM-ICD: Automatic ICD Coding with Pretrained Language Models. In *Proceedings of the 4th Clinical Natural Language Processing Workshop*, pages 10–20, Seattle, WA. Association for Computational Linguistics.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. 2019. Adversarial Examples Are Not Bugs, They Are Features. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Alon Jacovi and Yoav Goldberg. 2020. Towards Faithfully Interpretable NLP Systems: How Should We Define and Evaluate Faithfulness? In *Proceedings* *of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, Online. Association for Computational Linguistics.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. *Preprint*, arXiv:1902.10186.

Alistair E. W. Johnson, Tom J. Pollard, Lu Shen, Liwei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3(1):160035.

Zulqarnain Khan, Davin Hill, Aria Masoomi, Joshua Bone, and Jennifer Dy. 2023. Analyzing Explainer Robustness via Lipschitzness of Prediction Functions. *Preprint*, arXiv:2206.12481.

Byung-Hak Kim, Zhongfen Deng, Philip S. Yu, and Varun Ganapathi. 2022. Can Current Explainability Help Provide References in Clinical Notes to Support Humans Annotate Medical Codes? *Preprint*, arXiv:2210.15882.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2021. Incorporating Residual and Normalization Layers into Analysis of Masked Language Models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4547–4568, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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. *Preprint*, arXiv:2009.07896.

Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. 2019. Quantifying the Carbon Emissions of Machine Learning. *Preprint*, arXiv:1910.09700.

Patrick Lewis, Myle Ott, Jingfei Du, and Veselin Stoyanov. 2020. Pretrained Language Models for Biomedical and Clinical Tasks: Understanding and Extending the State-of-the-Art. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pages 146–157, Online. Association for Computational Linguistics.

Dongfang Li, Baotian Hu, Qingcai Chen, and Shan He. 2023. Towards Faithful Explanations for Text Classification with Robustness Improvement and Explanation Guided Training. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 1–14, Toronto, Canada. Association for Computational Linguistics.

Leibo Liu, Oscar Perez-Concha, Anthony Nguyen, Vicki Bennett, and Louisa Jorm. 2022. Hierarchical label-wise attention transformer model for explainable ICD coding. *Journal of Biomedical Informatics*, 133:104161.

Ilya Loshchilov and Frank Hutter. 2022. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

Scott M Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Qing Lyu, Marianna Apidianaki, and Chris Callison-Burch. 2023. Towards Faithful Model Explanation in NLP: A Survey. *Preprint*, arXiv:2209.11326.

Zico Kolter and Aleksander Madry. 2018. Adversarial Robustness - Theory and Practice. http://adversarial-ml-tutorial.org/.

Ali Modarressi, Mohsen Fayyaz, Ehsan Aghazadeh, Yadollah Yaghoobzadeh, and Mohammad Taher Pilehvar. 2023. DecompX: Explaining Transformers Decisions by Propagating Token Decomposition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2649–2664, Toronto, Canada. Association for Computational Linguistics.

Ali Modarressi, Mohsen Fayyaz, Yadollah Yaghoobzadeh, and Mohammad Taher Pilehvar. 2022. GlobEnc: Quantifying Global Token Attribution by Incorporating the Whole Encoder Layer in Transformers. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 258–271, Seattle, United States. Association for Computational Linguistics.

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable Prediction of Medical Codes from Clinical Text. 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 1101–1111, New Orleans, Louisiana. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144, San Francisco California USA. ACM.

Andrew Slavin Ros and Finale Doshi-Velez. 2018. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18, pages 1660–1669, New Orleans, Louisiana, USA. AAAI Press.

Mihaela Rosca, Theophane Weber, Arthur Gretton, and Shakir Mohamed. 2020. A case for new neural network smoothness constraints. pages 21–32. PMLR.

Sofia Serrano and Noah A. Smith. 2019. Is Attention Interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy. Association for Computational Linguistics.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 3145–3153, Sydney, NSW, Australia. JMLR.org.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic Attribution for Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3319–3328. PMLR.

Fei Teng, Yiming Liu, Tianrui Li, Yi Zhang, Shuangqing Li, and Yue Zhao. 2022. A review on deep neural networks for ICD coding. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1.

Fei Teng, Wei Yang, L. Chen, Lufei Huang, and Qiang Xu. 2020. Explainable Prediction of Medical Codes With Knowledge Graphs. *Frontiers in Bioengineering and Biotechnology*.

Manan Tomar, Riashat Islam, Matthew E. Taylor, Sergey Levine, and Philip Bachman. 2023. Ignorance is Bliss: Robust Control via Information Gating. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*. arXiv.

Phillip Tseng, Robert S. Kaplan, Barak D. Richman, Mahek A. Shah, and Kevin A. Schulman. 2018. Administrative Costs Associated With Physician Billing and Insurance-Related Activities at an Academic Health Care System. *JAMA*, 319(7):691–697.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. 2018. Robustness May Be at Odds with Accuracy. In *International Conference on Learning Representations*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Keyang Xu, Mike Lam, Jingzhi Pang, Xin Gao, Charlotte Band, Piyush Mathur, Frank Papay, Ashish K. Khanna, Jacek B. Cywinski, Kamal Maheshwari, Pengtao Xie, and Eric P. Xing. 2019. Multimodal Machine Learning for Automated ICD Coding. In *Proceedings of the 4th Machine Learning for Healthcare Conference*, pages 197–215. PMLR.

Jin Yong Yoo and Yanjun Qi. 2021. Towards Improving Adversarial Training of NLP Models. *Preprint*, arXiv:2109.00544.

Table 4: Prediction performance on discharge summaries from the MIMIC-III full. The models are trained on the MIMIC-III full training set. The results are in percentages.

	MIMIC-III full test		
Model	F1 micro	F1 macro	mAP
PLM-ICD	59.5±0.2	23.3±0.6	64.3±0.2
PLM-CA	**60.0±0.1**	**24.7±0.5**	**64.7±0.1**

A Model architecture details

PLM-ICD is a state-of-the-art automated medical coding model (Huang et al., 2022; Edin et al., 2023). It comprises 131 million parameters. We experienced that PLM-ICD occasionally crashed during training. Therefore, we modified the architecture and called it pre-trained language model with class-wise cross attention (PLM-CA) (Huang et al., 2022). Our architecture comprises an encoder and a decoder (see Figure 5). The encoder transforms a sequence of tokens indices $\boldsymbol{t} \in \{0, 1, \ldots, V\}^N$ into a sequence of contextualized token representations $\boldsymbol{H} \in \mathbb{R}^{N \times D}$. Both PLM-ICD and PLM-CA use RoBERTa-PM, a transformer pre-trained on PubMed articles and clinical notes, as the encoder (Lewis et al., 2020).

Our decoder takes the token representations $\boldsymbol{H}$ as input and outputs a sequence of output probabilities $\hat{\boldsymbol{y}} \in [0, 1]^J$. It computes the output probabilities from the contextualized token representations using the following equation:

$$\boldsymbol{K} = \boldsymbol{H}\boldsymbol{W}_{\text{key}} \qquad \boldsymbol{V} = \boldsymbol{H}\boldsymbol{W}_{\text{value}} \tag{1}$$

$$\boldsymbol{A}_j = \text{softmax}(\boldsymbol{C}_j \boldsymbol{K}^T) \tag{2}$$

$$\hat{\boldsymbol{y}}_j = \text{sigmoid}(\text{layernorm}(\boldsymbol{A}_j \boldsymbol{V})\boldsymbol{W}_{\text{out}}) \tag{3}$$

Where $\boldsymbol{W}_{\text{key}} \in \mathbb{R}^{D \times D}$, $\boldsymbol{W}_{\text{value}} \in \mathbb{R}^{D \times D}$, and $\boldsymbol{W}_{\text{out}} \in \mathbb{R}^{D}$ are learnable weights, $\boldsymbol{C} \in \mathbb{R}^{J \times D}$ is a sequence of learnable class representations, $\boldsymbol{A} \in \mathbb{R}^{J \times N}$ is the attention matrix, and J is the number of classes. In addition to being more stable during training, we also found that PLM-CA outperforms PLM-ICD on most metrics (see Table 4).

B Feature attribution methods

Attention We use the raw attention weights A_j (see Equation (2)) in the cross-attention layer to explain class j. As mentioned in Section 2.1, this explanation method was used by most previous

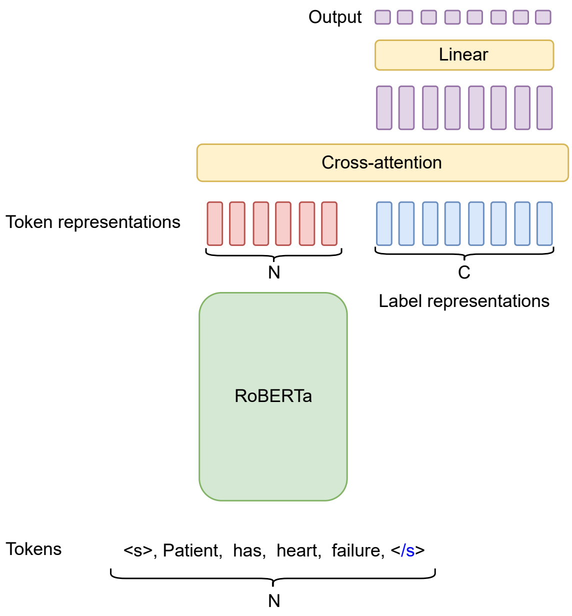


Figure 5: The PLM-CA architecture we used in our experiments.

studies in automated medical coding (Mullenbach et al., 2018; Kim et al., 2022; Dong et al., 2021; Teng et al., 2020; Cheng et al., 2023).

Attention Rollout (Rollout) The attention matrix in the cross-attention layer extracts information from the contextualized token representations encoded by RoBERTa (see Figure 5). The token representations are not guaranteed to be aligned with the input tokens. A token representation at position n could represent any and multiple tokens in the document. Attention rollout considers all the model's attention layers to calculate the feature attributions (Abnar and Zuidema, 2020). First, the attention matrices in each layer are averages across the heads. Then, the identity matrix is added to each layer's attention matrix to represent the skip connections. Finally, the attention rollout is calculated recursively using Equation (4).

$$\tilde{\boldsymbol{A}}^{(l)} = \begin{cases} \bar{\boldsymbol{A}}^{(l)} \cdot \tilde{\boldsymbol{A}}^{(l-1)} & \text{if } l > 0 \\ \bar{\boldsymbol{A}}^{(l)} & \text{if } l = 0 \end{cases} \tag{4}$$

where $\tilde{\boldsymbol{A}} \in \mathbb{R}^{N \times N}$ is the rollout attention, and $\bar{\boldsymbol{A}} \in \mathbb{R}^{N \times N}$ is the attention averaged across heads with the added identity matrix. We calculated the final feature attribution score by multiplying the rollout attention from the final layer with the attention matrix from the cross-attention layer: $\boldsymbol{A} \cdot \tilde{\boldsymbol{A}}^{(L)}$, where L is the number of attention layers.

Occlusion@1 Occlusion@1 calculates each feature's score by occluding it and measuring the change in output confidence. The change of output will be the feature's score (Ribeiro et al., 2016).

LIME Local Interpretable Model-agnostic Explanations (LIME) randomly occlude sets of tokens from a specific input and measure the change in output confidence. It uses these measurements to train a linear regression model that approximates the explained model's reasoning for that particular example. It then uses the linear regression weights to approximate each feature's influence (Ribeiro et al., 2016).

KernelSHAP Shapley Additive Explanations (SHAP) is based on Shapley values from cooperative game theory, which fairly distributes the payout among players by considering each player's contribution in all possible coalitions. Ideally, SHAP quantifies all possible feature combinations in an input by occluding them and measuring the impact. However, this would result in N forwards passes. KernelSHAP employs the LIME framework to approximate Shapley values using a weighted linear regression approach efficiently. We refer the reader to the seminal paper introducing SHAP and KernelSHAP for more details (Lundberg and Lee, 2017).

InputXGrad InputXGradient multiplies the input gradients with the input (Shrikumar et al., 2017). We used the L2 norm to get the final feature attribution scores. We calculated the feature attribution scores for class J as follows:

$$\begin{bmatrix} \left\| \boldsymbol{X}_1 \odot \frac{\partial f_j}{\partial \boldsymbol{X}_1}(X) \right\|_2 \\ \vdots \\ \left\| \boldsymbol{X}_N \odot \frac{\partial f_j}{\partial \boldsymbol{X}_N}(X) \right\|_2 \end{bmatrix} \tag{5}$$

where $\boldsymbol{X} \in \mathbb{R}^{N \times D}$ is the input token embeddings, $\odot$ is the element-wise matrix multiplication operation, D is the embedding dimension, N are the number of tokens in a document, and J is the number of classes.

Integrated Gradients (IntGrad) Integrated Gradients (IntGrad) assigns an attribution score to each input feature by computing the integral of the gradients along the straight line path from the baseline $\boldsymbol{B}$ to the input $\boldsymbol{X}$ (Sundararajan et al., 2017). Similar to InputXGradient, we used the L2-norm of the output to get the final attribution scores.

Deeplift DeepLIFT (Deep Learning Important FeaTures) backpropagates the contributions of all neurons in the model to every input feature (Shrikumar et al., 2017). It compares each neuron's activation to its baseline activation and assigns attribution scores according to the difference.

AttnGrad AttnGrad multiplies the attention Attention with the gradient of the model's output with respect to the attention weights (Serrano and Smith, 2019):

$$\begin{bmatrix} \boldsymbol{A}_{j1} \cdot |\frac{\partial f_j}{\partial \boldsymbol{A}_{j1}}(\boldsymbol{X})| \\ \vdots \\ \boldsymbol{A}_{jN} \cdot |\frac{\partial f_j}{\partial \boldsymbol{A}_{jN}}(\boldsymbol{X})| \end{bmatrix} \tag{6}$$

where $\boldsymbol{A} \in \mathbb{R}^{J \times N}$ is the attention matrix, N are the number of tokens in a document, and J is the number of classes.

AttInGrad We found Attention Rollout to perform poorly on our task. Therefore, we developed a simple alternative approach to incorporate the impact of neighboring tokens into the attention explanations. AttInGrad incorporates the context by multiplying the attention A_j with the InputXGrad feature attributions:

$$\begin{bmatrix} \boldsymbol{A}_{j1} \cdot \left\| \boldsymbol{X}_1 \odot \frac{\partial f_j}{\partial \boldsymbol{X}_1}(X) \right\|_2 \\ \vdots \\ \boldsymbol{A}_{jN} \cdot \left\| \boldsymbol{X}_N \odot \frac{\partial f_j}{\partial \boldsymbol{X}_N}(X) \right\|_2 \end{bmatrix} \tag{7}$$

where $\boldsymbol{A} \in \mathbb{R}^{J \times N}$ is the attention matrix, $\odot$ is the element-wise matrix multiplication operation, N are the number of tokens in a document, and J is the number of classes.

C Faithfulness evaluation metrics

Our faithfulness metrics, Sufficiency, and Comprehensiveness evaluate model output changes when important or unimportant features were masked (DeYoung et al., 2020).

Sufficiency measures how masking non-important features affects the output. A high sufficiency score indicates that many low-attribution features significantly impact the model's output, suggesting the presence of false negatives. We calculated sufficiency using the following equation:

$$\frac{1}{K} \sum_{i=N-K}^{N} \frac{\max(0, f(\boldsymbol{X}) - f(\boldsymbol{R}_i))}{f(\boldsymbol{X})} \tag{8}$$

Where $\boldsymbol{R}_i \in \mathbb{R}^{N \times D}$ represents the input with the ith least important feature replaced by mask tokens, N is the number of tokens in an example, and K is a hyperparameter.

Comprehensiveness measures how masking important features affects the output. A high comprehensiveness score indicates that features with high attribution scores strongly influence the model's output, while a low score suggests many false positives. We calculated comprehensiveness using the following equation:

$$\frac{1}{K} \sum_{i=0}^{K} \frac{\max(0, f(\boldsymbol{X}) - f(\bar{\boldsymbol{R}}_i))}{f(\boldsymbol{X})} \tag{9}$$

Where $\bar{\boldsymbol{R}}_i \in \mathbb{R}^{N \times D}$ denotes the input features with the ith highest attribution scores replaced by mask tokens.

We set $K = 100$ because including all features led to sufficiency scores close to zero and comprehensiveness scores close to one, making it difficult to distinguish differences. Considering fewer features also made evaluation faster.

D Training details

We used the same hyperparameter as Edin et al. (2023). We trained for 20 epochs with the ADAMW optimizer (Loshchilov and Hutter, 2022), learning rate at $5 \cdot 10^{-5}$, dropout at 0.2, no weight decay, and a linear decay learning rate scheduler with warmup. We found the optimal hyperparameters for the auxiliary adversarial robustness training objectives through random search. For each training strategy, we searched the following options: learning rate: $\{5 \cdot 10^{-5}, 1 \cdot 10^{-5}\}$, λ_1, λ_2, λ_3, β: $\{1.0, 0.5, 0.1, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$, and ϵ: $\{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$. We found these hyperparameters to be optimal: $\lambda_1 = 10^{-5}$, $\lambda_2 = 0.5$, $\lambda_3 = 0.5$, $\epsilon = 10^{-5}$, and $\beta = 0.01$. The learning rate was optimal at $5 \cdot 10^{-5}$ for all training strategies except for token masking, where $1 \cdot 10^{-5}$ was optimal. We optimized the token mask and adversarial noise using the ADAMW optimizer. In token-masking, we initialized the student and teacher model from a trained $\mathrm{B_U}$. We fine-tuned the student for one epoch. We used the same hyperparameters as Cheng et al. (2023) for the supervised training strategy.

We did not preprocess the text except to truncate the documents to a maximum length of 6000 tokens to reduce memory usage. Truncation is a common strategy in automated medical coding and has a negligible negative impact because few documents exceed the 6000 token limit (Edin et al., 2023).

E Additional results

Because of space constraints, we could not include all of our results in the main paper. In this section, we present the excluded results:

1. We show that the adversarial robustness training strategies do not affect the model's prediction performance.
2. We present the results for all feature attribution methods, including Rand, AttGrad, Rollout, Occlusion@1, LIME, and KernelSHAP.
3. We demonstrate that when the model struggles to predict the correct code, the explanations' plausibility drastically drops.
4. We analyze if the robust models use fewer features by comparing the entropy of the feature attribution scores.
5. We compare the unsupervised models on a bigger test set comprising 242 examples instead of 61.

E.1 Advesarial training does not affect code prediction performance

Previous papers have demonstrated that adversarial robustness often comes at the cost of accuracy (Li et al., 2023; Tsipras et al., 2018). Therefore, we evaluated whether the training strategies impacted the models' medical code prediction capabilities. As shown in Table 5, all models performed similarly on the MDACE test set. We also observed negligible performance differences on the MIMIC-III full test set.

E.2 Results from all feature attribution methods

In the main paper, we only presented the results of selected feature attribution methods because of space constraints. Here, we present the results for all the feature attribution methods: Attention, AttGrad, Attention Rollout (Rollout), InputXGrad, Integrated gradients (IntGrad), Deeplift, and AttInGrad. We compare these methods with a random baseline (Rand), which randomly generates attribution scores. We present the plausibility results in Table 7, and the faithfulness results in Table 8.

Table 5: Prediction performance on discharge summaries from the MDACE test sets. The results are in percentages. We use two classification metrics (F1 micro and macro) and mean Average Precision (mAP), a ranking metric.

	MDACE test		
Model	F1 micro ↑	F1 macro ↑	mAP ↑
B_S	67.5±0.4	51.5±1.1	72.0±0.7
B_U	67.7±0.3	51.6±0.7	72.0±0.4
IGR	68.0±0.6	52.2±1.2	72.4±1.0
TM	68.1±0.6	51.8±1.5	72.3±0.4
PGD	68.1± 0.5	52.1± 1.2	72.0± 0.7

We did not include Occlusion@1, LIME, and KernelSHAP in these tables because they were too slow to calculate. We used the Captum implementation of the algorithms (Kokhlikyan et al., 2020). It took around 45 minutes on an A100 GPU to calculate the explanations for a single example with LIME and KernelSHAP. Therefore, we only evaluated these methods on a single trained instance of B_U. We present the results in Table 9.

E.3 Relationship between confidence scores and explanation plausibility

In Table 10 and Table 11, we investigate the difference in explanation plausibility when the model correctly predicts an annotated code (true positive) and when it fails to predict an annotated code (false negative). The explanations are substantially better when the model correctly predicts the codes.

E.4 Entropy of explanation methods

We calculated the entropy of the feature attribution distributions to test our hypothesis that robust training strategies reduce the number of features the model uses (see Table 6). The training strategies did not reduce the entropy. While we would expect a reduced entropy if the model used fewer features, other feature attribution distribution differences may simultaneously increase the entropy. The analysis is, therefore, inconclusive.

Table 6: Entropy of the explanations.

Explanation	Model	Entropy ↓
InputXGrad	B_U	0.74±0.01
	IGR	0.74±0.00
	TM	0.73±0.00
Attention	B_U	0.50±0.01
	IGR	0.50±0.01
	TM	0.49±0.02
AttInGrad	B_U	0.28±0.02
	IGR	0.28±0.02
	TM	0.28±0.02

E.5 Unsupervised comparison on bigger test set

We included additional experiments on the unsupervised training strategies on a bigger test set. Since only the supervised training strategy required evidence-span annotations in the training set, we retrained our unsupervised methods on the MIMIC-III full training set and evaluated them on the MDACE training and test set (242 examples).

We present the plausibility results in Table 12. We observe that the results are similar to those of the main paper. However, the IGR produced substantially better attention-based explanations than in the main paper. In Figure 6, we inspect the inter-seed variance. We observe that IGR has no outliers. We, therefore, attribute the differences between this comparison and that in the main paper to none of the ten IGR runs happening to produce an outlier model. These results highlight the fragility of evaluating the attention-based feature attribution methods.

F CO_2 emissions

Experiments were conducted using a private infrastructure, which has a carbon efficiency of 0.185 $kgCO_2eq/kWh$. To train B_U or B_S, a cumulative of 8 hours of computation was performed on hardware of type A100 PCIe 40/80GB (TDP of 250W). Total emissions for one run are estimated to be 0.37 $kgCO_2eq$. The adversarial robustness training strategies required more hours of computation, therefore causing higher emissions. Input gradient regularization and projected gradient regularization required approximately 36 hours each (1.67 $kgCO_2eq$), while token masking required 2.5 hours of fine-tuning of B_U (0.09 $kgCO_2eq$). We ran each experiment 10 times, resulting in total emissions of 41.86 $kgCO_2eq$, which is equivalent to burning 20.9 Kg of coal. Estimations were conducted using the MachineLearning Impact calculator (Lacoste et al., 2019).

G Licenses

We used MIMIC-III version 1.4, which is distributed under a non-commercial license, as detailed here: `https://physionet.org/content/mimiciii/view-license/1.4/`. Consequently, all model weights released in this paper are also restricted to non-commercial use. However, the MDACE annotations and our code are available under the MIT License. We provide instructions to how to obtain the datasets in our GitHub repository.

Table 7: Plausibility comparison of Attention, AttGrad, Attention Rollout (Rollout), InputXGrad, Integrated gradients (IntGrad), Deeplift, and AttInGrad on the MDACE test set. Rand randomly generated attribution scores. Each experiment was run with ten different seeds. We show the mean of the seeds ± as the standard deviation. All the scores are presented as percentages. Bold numbers outperform the unsupervised baseline model, while underlined numbers outperform the supervised model.

		Prediction							Ranking		
Explainer	Model	P ↑	R ↑	F1 ↑	AUPRC ↑	Empty ↓	SpanR ↑	Cover ↑	IOU ↑	P@5 ↑	R@5 ↑
Rand	$\mathrm{B_S}$	0.3±0.1	1.3±0.8	0.4±0.0	0.0±0.0	49.2±37.6	1.6±0.9	3.2±1.8	0.1±0.0	0.1±0.0	0.1±0.1
	$\mathrm{B_U}$	0.4±0.1	1.0±0.7	0.4±0.1	0.0±0.0	64.7±37.9	1.1±0.7	2.3±1.6	0.1±0.1	0.1±0.1	0.2±0.1
	IGR	0.4±0.2	0.7±0.7	0.4±0.2	0.0±0.0	80.3±30.9	1.0±0.9	1.9±1.9	0.1±0.0	0.1±0.1	0.2±0.1
	TM	0.4±0.1	1.0±0.8	0.4±0.1	0.4±1.1	65.3±37.2	1.2±0.9	2.4±1.8	0.1±0.1	0.1±0.1	0.1±0.1
	PGD	0.4±0.2	0.8±0.6	0.4±0.1	0.0±0.0	76.1±33.2	1.1±0.7	2.0±1.7	0.1±0.1	0.1±0.1	0.2±0.1
Attention	$\mathrm{B_S}$	41.7±5.0	42.9±3.2	42.2±3.7	37.3±3.9	13.5±1.9	60.5±3.7	72.3±3.5	46.1±4.5	32.3±2.6	45.3±3.7
	$\mathrm{B_U}$	35.9±5.9	37.7±5.7	36.6±5.3	31.4±6.4	21.1±3.6	53.2±7.4	63.4±6.8	39.3±6.9	28.8±4.2	40.3±5.9
	IGR	35.5±6.2	**40.4±6.1**	**37.7±5.9**	**32.4±6.8**	**19.4±2.1**	**55.5±8.0**	**64.5±6.2**	**40.0±7.4**	**29.2±4.5**	**40.9±6.3**
	TM	**36.5±5.9**	**37.8±6.1**	**37.0±5.5**	**31.7±6.7**	21.7±3.3	**53.3±7.7**	63.3±6.9	**40.1±7.3**	**29.1±4.4**	**40.8±6.2**
	PGD	33.0±8.4	**38.4±8.0**	35.4±8.3	30.0±9.1	**19.1±1.6**	51.5±13.6	61.1±9.9	35.9±10.8	27.7±6.3	38.9±8.8
Rollout	$\mathrm{B_S}$	1.6±0.1	23.4±5.4	2.9±0.1	0.3±0.0	0.0±0.0	25.0±5.1	32.2±5.0	0.4±0.1	0.5±0.1	0.7±0.1
	$\mathrm{B_U}$	1.7±0.1	24.1±2.4	3.1±0.2	0.3±0.0	0.0±0.0	25.1±2.8	31.4±2.9	0.3±0.1	0.5±0.1	0.7±0.1
	IGR	1.7±0.1	24.3±2.5	3.1±0.2	0.3±0.0	0.0±0.0	25.4±3.1	31.6±2.9	0.3±0.1	0.5±0.1	0.7±0.1
	TM	1.7±0.1	25.3±2.3	3.2±0.2	0.3±0.0	0.0±0.0	26.3±2.7	32.4±2.7	0.3±0.1	0.5±0.1	0.7±0.1
	PGD	1.7±0.0	24.8±0.6	3.2±0.1	0.3±0.0	0.0±0.0	26.0±0.7	32.6±0.8	0.3±0.0	0.5±0.0	0.6±0.0
$a\nabla a$	$\mathrm{B_S}$	37.4±5.8	37.8±3.0	37.4±4.1	31.5±5.1	11.9±2.4	55.0±3.8	69.9±3.8	39.7±6.1	29.4±3.1	41.2±4.4
	$\mathrm{B_U}$	34.4±7.5	36.2±3.9	34.8±5.0	29.0±5.7	16.0±2.8	51.8±4.8	64.7±4.3	36.2±7.3	27.4±4.0	38.4±5.6
	IGR	**35.1±7.0**	**38.2±4.1**	**36.3±5.2**	**30.8±6.4**	**15.0±2.2**	**54.2±5.5**	**65.7±4.0**	**38.3±9.0**	**28.9±4.6**	**40.5±6.5**
	TM	**34.6±6.9**	**36.5±4.2**	**35.2±5.2**	**29.1±5.9**	16.1±2.7	**52.2±5.0**	**65.0±4.4**	**36.3±7.2**	**27.8±4.3**	**39.0±6.1**
	PGD	32.7±8.0	**37.9±4.4**	**34.9±6.9**	29.1±7.9	**14.3±3.0**	**52.8±7.6**	64.7±5.4	35.7±10.8	**28.4±5.6**	**39.8±7.8**
InputXGrad	$\mathrm{B_S}$	30.7±2.0	33.7±2.6	32.0±1.4	24.9±2.0	7.3±1.3	48.5±2.8	64.0±2.8	32.2±2.1	26.6±1.2	37.3±1.6
	$\mathrm{B_U}$	31.0±2.6	32.7±3.2	31.6±1.2	24.8±1.9	9.5±2.6	46.4±3.6	62.3±3.5	31.7±1.8	26.3±0.9	36.9±1.3
	IGR	**32.4±2.5**	**33.8±2.2**	**33.0±0.7**	**27.0±0.7**	9.7±2.0	**48.5±2.6**	**64.6±2.5**	**32.8±1.0**	**27.4±0.4**	**38.4±0.5**
	TM	**32.4±2.7**	**34.1±2.1**	**33.1±1.4**	**26.2±2.3**	**8.6±1.6**	**48.5±2.4**	**64.8±2.5**	**32.6±1.5**	**27.4±1.1**	**38.3±1.5**
	PGD	30.1±2.2	**32.9±1.7**	31.4±1.4	24.8±1.5	**9.3±2.2**	**46.6±2.1**	**62.6±2.6**	31.4±1.3	26.1±1.0	36.6±1.4
IG	$\mathrm{B_S}$	30.9±3.2	33.9±2.0	32.2±1.7	26.2±2.6	4.8±2.0	50.8±2.4	65.9±3.3	34.2±2.8	27.4±1.6	38.4±2.2
	$\mathrm{B_U}$	31.3±4.3	32.8±2.7	31.9±3.2	26.0±4.3	5.2±1.0	49.4±3.0	64.4±3.1	33.9±4.2	26.5±2.5	37.1±3.6
	IGR	**32.6±2.2**	**33.8±2.3**	**33.1±1.3**	**26.6±2.0**	5.2±1.7	**49.7±2.5**	**64.7±3.3**	**34.9±2.3**	**27.6±0.9**	**38.7±1.2**
	TM	**33.1±3.6**	**34.0±3.7**	**33.4±2.8**	**27.5±3.7**	5.4±1.6	**51.0±3.9**	**65.5±4.1**	**35.3±4.2**	**27.6±2.1**	**38.7±3.0**
	PGD	30.0±5.4	**33.5±3.3**	31.4±4.2	25.4±4.8	**5.1±1.8**	**49.7±3.6**	**64.5±3.9**	33.6±4.8	26.4±3.0	36.9±4.2
Deeplift	$\mathrm{B_S}$	29.2±2.3	34.2±3.0	31.3±1.2	24.1±1.8	6.4±1.8	48.9±3.4	65.2±3.3	31.1±1.9	26.2±1.1	36.7±1.5
	$\mathrm{B_U}$	31.2±1.9	31.4±2.4	31.2±1.5	24.1±2.1	9.1±1.6	45.1±2.5	61.7±2.9	30.9±1.6	25.8±1.1	36.1±1.5
	IGR	**31.0±1.8**	**33.7±1.4**	**32.2±0.6**	**25.9±0.9**	**8.6±1.0**	**48.3±1.5**	**64.8±1.7**	**31.6±1.0**	**26.7±0.5**	**37.4±0.7**
	TM	**30.2±2.4**	**34.8±1.4**	**32.2±1.4**	**25.1±2.5**	**6.8±1.5**	**49.1±1.8**	**65.7±1.6**	**31.4±1.8**	**26.6±1.1**	**37.3±1.6**
	PGD	29.4±2.8	**32.8±1.5**	30.9±1.5	23.9±1.5	**8.1±1.8**	**46.7±1**.8	**63.3±2.2**	30.7±1.4	25.5±0.9	35.8±1.2
AttInGrad	$\mathrm{B_S}$	40.6±2.2	45.9±3.1	43.0±1.9	38.8±2.1	1.2±0.5	63.9±2.9	79.0±1.9	45.7±2.6	33.3±1.4	46.6±1.9
	$\mathrm{B_U}$	40.7±2.9	42.5±4.4	41.5±3.2	37.1±3.6	3.0±1.3	59.3±5.4	74.9±5.2	43.5±4.0	32.0±2.1	44.9±3.0
	IGR	38.8±4.5	**44.0±4.0**	41.2±4.0	**37.2±4.5**	**2.3±0.7**	**60.3±5.3**	74.5±4.6	43.3±4.9	31.9±2.7	44.7±3.8
	TM	40.2±3.0	**43.9±4.8**	**41.9±3.4**	**37.8±3.9**	**2.8±1.6**	**60.5±5.8**	**75.9±5.3**	**44.0±3.9**	**32.5±2.3**	**45.5±3.2**
	PGD	37.1±8.2	42.5±5.7	39.3±7.1	34.7±8.6	**2.3±0.9**	57.7±9.1	72.7±7.4	39.6±10.2	30.9±4.8	43.3±6.7

Table 8: Faithfulness comparison on the MDACE test set. Each experiment was run with ten different seeds. We show the mean of the seeds ± the standard deviation. The bold numbers represent the best score for each model and metric, while the underscore represents better than the Attention explanation method.

Model	Explainer	Comp ↑	Suff ↓
B_U	Rand	0.03±0.02	0.92±0.08
	Attention	0.82±0.08	0.29±0.07
	InputXGrad	0.75±0.01	$\underline{\mathbf{0.17}\pm\mathbf{0.03}}$
	IG	0.74±0.02	$\underline{0.19\pm0.04}$
	Deeplift	0.76±0.02	$\underline{0.18\pm0.03}$
	Rollout	0.39±0.02	0.47±0.08
	$a\nabla a$	0.77±0.07	$\underline{0.28\pm0.06}$
	AttInGrad	$\underline{\mathbf{0.86}\pm\mathbf{0.04}}$	$\underline{0.22\pm0.05}$
B_S	Rand	0.02±0.02	0.93±0.08
	Attention	0.84±0.06	0.42±0.15
	InputXGrad	0.76±0.01	$\underline{\mathbf{0.26}\pm\mathbf{0.09}}$
	IG	0.74±0.01	$\underline{0.28\pm0.10}$
	Deeplift	0.76±0.01	$\underline{\mathbf{0.26}\pm\mathbf{0.09}}$
	Rollout	0.39±0.02	0.47±0.08
	$a\nabla a$	0.77±0.06	$\underline{0.41\pm0.14}$
	AttInGrad	$\underline{\mathbf{0.86}\pm\mathbf{0.03}}$	$\underline{0.34\pm0.12}$
IGR	Rand	0.04±0.01	0.95±0.02
	Attention	0.85±0.08	0.23±0.07
	InputXGrad	0.77±0.01	$\underline{\mathbf{0.14}\pm\mathbf{0.03}}$
	IG	0.75±0.02	$\underline{\mathbf{0.14}\pm\mathbf{0.03}}$
	Deeplift	0.77±0.01	$\underline{\mathbf{0.14}\pm\mathbf{0.03}}$
	Rollout	0.39±0.02	0.41±0.03
	$a\nabla a$	0.82±0.06	0.23±0.07
	AttInGrad	$\underline{\mathbf{0.87}\pm\mathbf{0.04}}$	$\underline{0.17\pm0.05}$
TM	Rand	0.02±0.01	0.92±0.06
	Attention	0.82±0.08	0.24±0.06
	InputXGrad	0.76±0.01	$\underline{\mathbf{0.15}\pm\mathbf{0.03}}$
	IG	0.74±0.01	$\underline{\mathbf{0.15}\pm\mathbf{0.03}}$
	Deeplift	0.77±0.01	$\underline{\mathbf{0.15}\pm\mathbf{0.03}}$
	Rollout	0.37±0.01	0.40±0.04
	$a\nabla a$	0.77±0.06	0.24±0.05
	AttInGrad	$\underline{\mathbf{0.86}\pm\mathbf{0.04}}$	**0.18±0.04**
PGD	Rand	0.04±0.01	0.93±0.06
	Attention	0.84±0.10	0.28±0.08
	InputXGrad	0.77±0.01	$\underline{\mathbf{0.17}\pm\mathbf{0.03}}$
	IG	0.75±0.02	**0.18±0.03**
	Deeplift	0.77±0.01	$\underline{\mathbf{0.17}\pm\mathbf{0.03}}$
	Rollout	0.40±0.01	0.41±0.04
	$a\nabla a$	0.81±0.07	0.28±0.07
	AttInGrad	$\underline{\mathbf{0.86}\pm\mathbf{0.05}}$	**0.21±0.05**

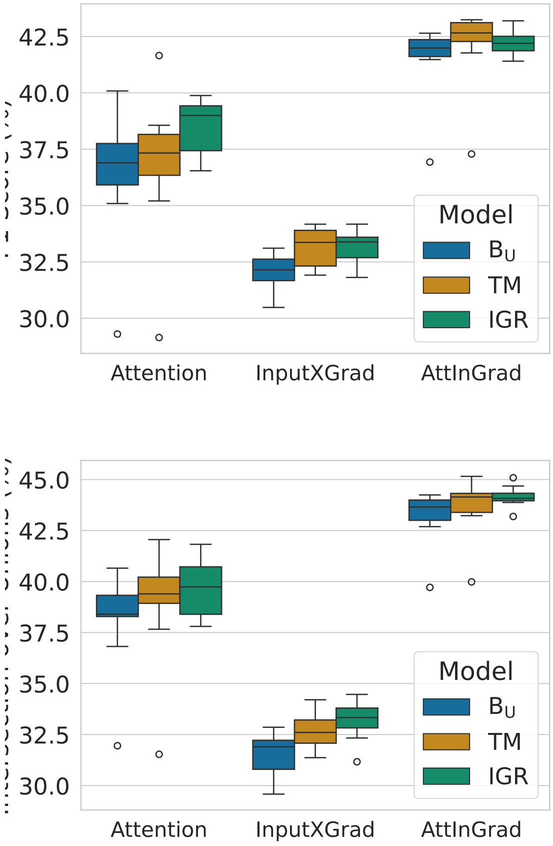


Figure 6: F1 scores and IOU of the unsupervised approaches on the bigger test set. Notice that there are no outliers for the attention-based explanations produced by IGR, which explains its higher mean scores in Table 12.

Table 9: Evaluation of perturbation-based feature attribution methods. We chose a random seed of $\mathrm{B_U}$ and compared all feature attribution methods for that model. We have divided the feature attribution into attention-based, gradient-based, and perturbation-based. The highest values are bold, and the second highest are underlined.

	Prediction							Ranking		
Explainer	P ↑	R ↑	F1 ↑	AUPRC ↑	Empty ↓	SpanR ↑	Cover ↑	IOU ↑	P@5 ↑	R@5 ↑
Attention	37.7	38.9	38.3	33.1	14.8	55.1	72.4	40.9	29.7	41.6
$a\nabla a$	**43.5**	33.6	37.9	31.6	14.0	50.2	68.7	40.0	29.3	41.1
AttInGrad	41.5	38.5	**39.9**	35.7	2.6	55.3	75.7	**41.0**	**30.5**	**42.8**
InputXGrad	32.8	29.8	31.3	24.2	12.5	42.7	59.4	30.1	26.0	36.4
IG	33.5	35.4	34.4	29.3	5.3	51.6	66.1	38.2	27.7	38.9
Deeplift	30.0	32.1	31.0	23.6	9.6	44.8	61.6	28.7	25.1	35.1
Occl	40.2	27.8	32.9	27.3	11.3	41.9	61.7	33.0	25.8	36.1
KernelSHAP	24.7	19.0	21.5	**38.9**	29.9	34.6	52.5	33.0	24.9	35.0
LIME	33.3	**39.5**	36.1	27.9	**0.3**	**58.4**	**81.6**	38.4	29.2	41.0

Table 10: Plausibility scores. True positives

		Prediction							Ranking		
Explainer	Model	P ↑	R ↑	F1 ↑	AUPRC ↑	Empty ↓	SpanR ↑	Cover ↑	IOU ↑	P@5 ↑	R@5 ↑
Rand	$\mathrm{B_S}$	0.2±0.2	1.0±0.7	0.3±0.1	0.0±0.0	48.3±38.5	1.3±0.9	2.7±1.7	0.1±0.0	0.1±0.1	0.1±0.1
	$\mathrm{B_U}$	0.3±0.2	0.7±0.6	0.3±0.1	0.0±0.0	64.1±38.6	0.9±0.7	1.8±1.5	0.1±0.1	0.2±0.1	0.2±0.1
	IGR	0.3±0.2	0.6±0.7	0.3±0.1	0.0±0.0	79.9±31.7	0.8±0.9	1.5±1.9	0.1±0.0	0.1±0.1	0.2±0.1
	TM	0.3±0.1	0.8±0.8	0.3±0.1	0.4±1.2	64.8±38.1	1.1±1.0	2.1±1.9	0.1±0.1	0.1±0.1	0.1±0.1
	PGD	0.3±0.2	0.6±0.6	0.3±0.1	0.0±0.0	75.3±34.0	0.9±0.7	1.7±1.5	0.1±0.1	0.1±0.1	0.2±0.1
Attention	$\mathrm{B_S}$	40.7±5.0	51.1±3.7	45.2±4.3	41.2±4.5	4.2±1.4	69.5±4.9	81.8±4.3	48.4±5.2	34.3±3.2	49.0±4.3
	$\mathrm{B_U}$	35.1±6.0	45.8±7.0	39.6±6.0	35.5±7.5	9.1±3.3	62.7±9.1	73.5±7.7	43.3±8.2	31.0±4.9	44.5±7.1
	IGR	**35.3±6.5**	**49.5±7.7**	**41.1±6.8**	**36.9±8.0**	**7.5±1.6**	**65.8±10.0**	**75.1±7.5**	**44.2±8.3**	**31.7±5.2**	**45.4±7.3**
	TM	**35.7±6.0**	**45.9±7.4**	**40.0±6.2**	**35.8±7.9**	9.7±3.4	**62.9±9.6**	**73.6±8.2**	**44.0±8.5**	**31.3±5.0**	**44.8±7.2**
	PGD	32.8±8.8	**47.3±10.8**	38.7±9.8	34.5±10.9	**7.3±1.2**	61.6±16.9	71.8±12.9	40.4±12.3	30.1±7.1	43.1±10.1
Rollout	$\mathrm{B_S}$	1.7±0.1	28.6±5.8	3.1±0.1	0.3±0.0	0.0±0.0	30.7±6.0	37.8±5.6	0.4±0.1	0.5±0.1	0.8±0.1
	$\mathrm{B_U}$	1.8±0.1	28.6±2.6	3.3±0.1	0.3±0.0	0.0±0.0	29.8±3.1	35.7±3.5	0.4±0.0	0.5±0.1	0.8±0.1
	IGR	1.8±0.1	29.3±2.9	3.4±0.2	0.3±0.0	0.0±0.0	30.7±3.6	36.9±3.4	0.4±0.0	0.5±0.1	0.8±0.1
	TM	1.8±0.1	29.6±2.3	3.3±0.1	0.3±0.0	0.0±0.0	31.0±3.0	36.7±3.3	0.4±0.0	0.5±0.1	0.7±0.1
	PGD	1.8±0.0	30.4±0.7	3.4±0.1	0.3±0.0	0.0±0.0	31.7±0.9	38.4±1.1	0.4±0.0	0.5±0.1	0.8±0.1
$a\nabla a$	$\mathrm{B_S}$	35.9±5.7	43.1±3.2	39.0±4.4	33.7±5.3	4.4±1.8	60.8±4.0	76.5±4.3	40.7±6.5	30.8±3.5	43.9±4.8
	$\mathrm{B_U}$	33.1±7.0	42.2±4.1	36.7±5.2	31.5±6.1	6.0±1.8	59.2±4.9	72.6±3.7	38.3±7.7	28.8±4.2	41.3±6.0
	IGR	**34.3±6.9**	**45.0±4.5**	**38.6±5.7**	**33.7±7.0**	**5.0±1.7**	**62.7±5.9**	**74.4±3.9**	**41.1±9.7**	**30.6±4.7**	**43.9±6.7**
	TM	**33.4±6.4**	**42.6±4.8**	**37.1±5.3**	**31.6±6.3**	**5.8±2.2**	**59.7±5.5**	**73.1±4.3**	**38.4±7.4**	**29.1±4.5**	**41.7±6.4**
	PGD	32.2±8.0	**44.9±5.7**	**37.2±7.8**	**32.1±8.9**	**4.8±1.8**	**60.7±9.2**	**73.3±7.0**	**38.4±11.7**	**30.0±5.8**	**42.9±8.2**
InputXGrad	$\mathrm{B_S}$	30.4±2.0	37.7±2.6	33.6±1.4	26.7±2.3	3.4±1.1	53.5±2.8	69.2±2.7	34.2±2.3	28.3±1.3	40.4±1.6
	$\mathrm{B_U}$	30.9±2.6	37.0±3.8	33.4±1.2	26.8±2.1	4.6±1.9	51.9±4.2	67.6±3.6	34.8±1.9	28.1±1.0	40.3±1.6
	IGR	**32.0±2.7**	**38.3±2.5**	**34.7±1.0**	**29.3±1.0**	4.9±1.5	**54.4±2.8**	**70.3±2.7**	**35.8±1.5**	**29.1±0.5**	**41.7±0.6**
	TM	**32.5±2.3**	**39.2±2.7**	**35.4±1.3**	**28.8±2.2**	**3.7±1.3**	**55.0±3.0**	**71.3±2.7**	**35.5±1.5**	**29.4±1.0**	**42.2±1.4**
	PGD	29.8±2.3	**37.2±1.7**	33.0±1.7	**27.1±1.7**	4.7±1.5	**52.3±2.3**	**68.0±2.9**	34.1±1.4	28.0±1.3	40.0±1.7
IG	$\mathrm{B_S}$	31.6±3.6	37.3±2.6	34.1±2.4	27.8±3.2	2.1±1.2	55.3±2.6	70.6±3.8	36.7±3.3	29.0±1.8	41.5±2.8
	$\mathrm{B_U}$	31.5±4.5	36.5±3.2	33.7±3.7	27.7±4.9	1.3±0.5	54.9±3.2	69.7±3.6	37.0±5.0	28.2±2.9	40.4±3.9
	IGR	**32.6±2.2**	**37.4±2.4**	**34.7±1.4**	27.7±2.5	**1.2±0.6**	54.9±2.8	69.5±3.6	**37.5±2.6**	**29.1±0.9**	**41.7±1.2**
	TM	**33.7±3.6**	**38.5±4.4**	**35.8±3.2**	**29.8±4.6**	1.3±0.9	**57.0±4.9**	**71.5±4.7**	**38.8±5.2**	**29.6±2.6**	**42.5±3.5**
	PGD	30.4±5.5	**37.2±3.9**	33.2±4.6	27.0±5.4	1.8±1.0	**55.2±4.4**	**69.8±4.6**	36.6±5.5	28.1±3.3	40.2±4.8
Deeplift	$\mathrm{B_S}$	29.1±2.3	38.4±3.0	33.0±1.3	26.1±1.9	3.2±1.3	54.2±3.6	70.5±3.4	33.4±2.1	27.7±1.2	39.6±1.5
	$\mathrm{B_U}$	31.2±1.9	35.7±2.7	33.2±1.5	26.2±2.3	4.0±1.4	50.7±2.9	67.3±3.0	33.9±1.8	27.6±1.1	39.6±1.6
	IGR	30.6±2.0	**38.2±1.5**	**33.9±0.9**	**28.1±1.2**	**3.8±0.8**	**54.3±1.7**	**70.5±1.7**	**34.2±1.4**	**28.5±0.6**	**40.8±0.8**
	TM	30.3±2.1	**39.9±1.9**	**34.4±1.3**	**27.7±2.3**	**2.3±0.5**	**55.8±2.2**	**72.5±1.7**	**34.3±2.0**	**28.6±1.0**	**41.0±1.5**
	PGD	29.3±3.0	**37.2±1.8**	32.6±1.8	26.1±1.8	**3.8±1.3**	**52.5±2.3**	**69.0±2.7**	33.5±1.5	27.4±1.1	39.2±1.6
AttInGrad	$\mathrm{B_S}$	41.8±2.4	50.2±3.2	45.6±2.1	41.8±2.3	0.0±0.1	68.7±3.2	83.4±2.2	48.1±2.7	35.0±1.5	50.0±1.9
	$\mathrm{B_U}$	41.3±3.3	47.2±5.3	43.9±3.9	40.3±4.4	0.2±0.4	65.3±6.4	80.4±5.8	46.4±4.9	33.7±2.6	48.4±3.8
	IGR	39.7±4.9	**49.1±4.7**	43.9±4.7	**41.0±5.4**	**0.1±0.1**	**66.9±6.1**	80.4±4.8	**46.6±5.5**	**33.8±3.2**	48.4±4.5
	TM	40.9±3.5	**48.5±6.1**	**44.3±4.3**	**41.3±4.9**	0.3±0.6	**66.3±7.3**	**81.3±6.3**	**47.1±5.2**	**34.2±2.8**	**49.0±4.3**
	PGD	37.9±9.3	47.1±8.0	41.7±8.8	38.4±10.2	0.2±0.2	63.2±12.0	77.8±9.7	42.8±11.9	32.7±5.7	46.8±8.1

Table 11: Plausibility scores. False negatives

		Prediction							Ranking		
Explainer	Model	P ↑	R ↑	F1 ↑	AUPRC ↑	Empty ↓	SpanR ↑	Cover ↑	IOU ↑	P@5 ↑	R@5 ↑
Rand	$\mathrm{B_S}$	0.4±0.1	1.9±0.9	0.6±0.1	0.0±0.0	51.2±35.8	2.2±1.0	4.3±2.3	0.1±0.1	0.1±0.1	0.1±0.1
	$\mathrm{B_U}$	0.4±0.2	1.4±0.9	0.6±0.2	0.0±0.0	65.8±36.6	1.6±1.0	3.4±1.9	0.0±0.1	0.1±0.1	0.1±0.1
	IGR	0.5±0.2	1.1±0.9	0.6±0.3	0.0±0.0	81.2±29.0	1.4±1.1	2.6±2.1	0.0±0.0	0.1±0.1	0.1±0.1
	TM	0.4±0.2	1.2±0.8	0.5±0.2	0.0±0.0	66.5±35.4	1.4±0.8	2.9±1.6	0.1±0.1	0.1±0.2	0.2±0.2
	PGD	0.5±0.2	1.2±0.8	0.6±0.1	0.0±0.0	77.9±31.4	1.4±0.8	2.8±1.9	0.1±0.1	0.1±0.1	0.1±0.1
Attention	$\mathrm{B_S}$	46.3±4.8	26.4±2.8	33.5±2.9	30.1±3.1	33.4±3.6	41.3±2.8	52.1±3.5	41.2±3.3	28.1±1.6	37.9±2.6
	$\mathrm{B_U}$	40.1±6.0	21.4±3.1	27.7±3.6	23.8±4.0	47.2±5.4	32.5±4.0	41.6±5.3	30.6±4.3	23.8±2.8	31.9±3.7
	IGR	36.6±5.1	21.1±2.7	26.8±3.3	23.3±4.2	**46.0±3.1**	32.0±3.6	40.4±3.5	30.4±5.4	23.6±3.1	31.5±4.2
	TM	**40.5±5.5**	**21.7±4.0**	**28.1±4.3**	**24.0±4.7**	47.7±4.8	**32.6±4.3**	41.2±5.3	**31.9±5.1**	**24.6±3.4**	**32.9±4.6**
	PGD	33.8±7.0	19.8±2.5	24.8±3.9	20.8±5.2	**45.4±3.6**	29.0±6.3	37.3±3.7	26.1±7.5	22.4±4.5	30.1±6.2
Rollout	$\mathrm{B_S}$	1.2±0.2	12.9±4.7	2.2±0.3	0.2±0.0	0.0±0.0	13.0±3.5	20.1±4.2	0.3±0.3	0.4±0.1	0.5±0.2
	$\mathrm{B_U}$	1.4±0.2	15.1±2.4	2.6±0.4	0.2±0.1	0.0±0.0	15.1±2.1	22.1±1.9	0.2±0.2	0.4±0.1	0.6±0.2
	IGR	1.3±0.2	13.6±3.4	2.4±0.5	0.2±0.0	0.0±0.0	13.4±3.2	19.7±4.1	0.1±0.2	0.3±0.1	0.4±0.1
	TM	1.5±0.2	16.6±2.6	2.7±0.4	0.2±0.1	0.0±0.0	16.0±2.3	23.0±2.3	0.2±0.2	0.4±0.1	0.5±0.1
	PGD	1.3±0.1	13.1±0.9	2.3±0.2	0.2±0.0	0.0±0.0	13.4±1.4	19.7±2.4	0.1±0.0	0.3±0.0	0.3±0.1
$a\nabla a$	$\mathrm{B_S}$	43.0±6.5	27.2±3.1	33.2±3.8	28.2±4.8	28.0±3.8	42.7±4.0	55.9±3.3	37.5±5.5	26.5±2.5	35.7±3.7
	$\mathrm{B_U}$	39.7±9.5	24.1±4.0	29.6±5.1	24.8±5.4	37.5±5.4	35.8±5.3	47.5±6.1	31.5±6.7	24.2±4.0	32.4±5.4
	IGR	39.0±7.8	23.8±3.5	29.2±4.3	**25.8±5.4**	37.5±3.8	35.2±4.6	46.1±4.5	**32.1±7.9**	**25.0±4.5**	**33.4±6.1**
	TM	**39.8±9.2**	**24.3±4.2**	**29.9±5.5**	**25.2±5.8**	38.2±5.0	**36.1±5.1**	47.5±5.8	**31.8±7.3**	**25.1±4.3**	**33.6±5.7**
	PGD	35.0±8.2	23.4±2.3	27.6±4.5	23.6±5.9	**35.5±6.3**	35.2±4.5	45.6±3.0	29.7±8.9	**24.8±5.1**	**33.3±7.0**
InputXGrad	$\mathrm{B_S}$	31.5±2.9	25.7±2.8	28.2±1.9	21.4±2.0	15.8±3.4	37.6±3.2	52.9±3.4	27.7±1.7	23.1±1.5	31.1±2.1
	$\mathrm{B_U}$	31.5±2.6	24.1±2.3	27.2±1.2	21.0±1.8	20.0±4.2	34.4±2.6	50.9±3.9	25.1±2.1	22.4±1.1	30.0±1.6
	IGR	**33.7±2.1**	**24.5±2.0**	**28.3±1.1**	**22.5±1.2**	20.6±3.6	**35.0±2.8**	**51.7±3.1**	**26.2±1.0**	**23.6±1.2**	**31.5±1.3**
	TM	**32.2±4.2**	24.1±1.6	**27.4±2.0**	**21.2±3.0**	**19.2±3.3**	**34.5±2.2**	50.9±3.1	**26.3±2.1**	**22.9±1.6**	**30.7±2.3**
	PGD	31.2±2.0	24.0±1.7	27.0±1.2	20.3±1.2	**19.7±4.2**	34.1±2.0	50.6±2.4	**25.4±1.4**	22.0±0.7	29.5±0.9
IG	$\mathrm{B_S}$	29.0±3.0	26.9±1.4	27.8±0.9	23.1±1.6	10.7±4.0	41.1±2.5	56.0±3.1	28.7±2.4	23.8±1.4	32.0±1.9
	$\mathrm{B_U}$	31.0±4.4	25.3±1.9	27.8±2.6	22.6±3.6	13.6±3.2	37.7±2.7	53.0±2.9	27.4±3.2	22.8±2.4	30.5±3.3
	IGR	**32.7±3.0**	**26.2±2.5**	**28.9±1.7**	**24.4±1.7**	14.3±4.3	**37.9±2.7**	**53.9±3.5**	**29.1±2.7**	**24.2±1.4**	**32.3±2.0**
	TM	**31.7±4.3**	24.9±2.8	27.7±2.4	**22.8±2.7**	14.2±4.1	**38.1±2.6**	52.7±3.9	**28.0±2.9**	**23.3±1.8**	**31.1±2.6**
	PGD	29.0±5.4	**25.8±2.5**	27.0±3.3	21.9±3.9	**12.3±3.9**	37.5±2.6	52.8±2.6	27.0±3.4	22.5±2.3	30.1±3.1
Deeplift	$\mathrm{B_S}$	29.4±2.9	25.7±3.1	27.2±1.7	20.4±1.9	13.3±4.0	37.6±3.3	54.0±3.9	26.3±1.8	22.9±1.6	30.8±2.1
	$\mathrm{B_U}$	31.1±2.1	22.7±2.0	26.2±1.8	20.0±2.0	20.0±2.4	33.0±2.1	49.7±2.9	24.4±1.7	21.8±1.2	29.2±1.8
	IGR	**32.2±1.9**	**24.3±1.8**	**27.7±1.4**	**21.6±1.5**	**19.4±1.7**	**34.8±2.6**	**52.0±3.2**	**25.6±1.4**	**22.8±1.3**	**30.4±1.5**
	TM	29.9±3.5	**24.5±1.5**	**26.8±1.9**	**20.1±2.9**	**16.3±4.2**	**34.7±2.5**	**51.2±2.5**	**25.0±2.0**	**22.3±1.5**	**29.9±2.1**
	PGD	29.7±2.6	**23.6±1.2**	26.2±0.9	19.4±1.0	**17.8±3.7**	**33.7±1.1**	**50.7±2.1**	**24.5±1.5**	21.4±0.6	28.6±0.9
AttInGrad	$\mathrm{B_S}$	37.7±2.1	37.2±3.5	37.3±1.9	32.4±2.1	3.9±1.6	53.6±3.5	69.7±2.9	40.7±3.2	29.6±1.4	39.8±2.3
	$\mathrm{B_U}$	39.1±2.7	33.1±3.0	35.7±1.9	30.2±2.1	8.9±3.5	46.3±3.7	62.8±4.5	37.2±2.8	28.3±1.3	37.9±1.5
	IGR	36.0±3.6	**33.3±2.5**	34.6±2.7	28.6±2.6	**7.2±2.0**	45.4±3.8	61.3±4.4	36.0±3.9	27.6±1.8	36.9±2.6
	TM	38.7±3.3	**34.7±2.9**	**36.4±1.8**	**30.7±2.6**	**8.4±4.1**	**48.1±3.3**	**64.3±3.8**	37.2±1.9	**28.9±1.5**	**38.6±1.8**
	PGD	34.8±5.8	33.0±2.3	33.5±3.1	26.6±4.6	**7.2±2.6**	45.5±3.1	61.3±2.9	32.4±6.7	26.9±2.9	36.1±4.0

Table 12: Plausibility comparison on unsupervised models on the bigger test set. Each experiment was run with ten different seeds. We show the mean of the seeds ± as the standard deviation. All the scores are presented as percentages. Bold numbers outperform the unsupervised baseline model ($\mathrm{B_U}$)

		Prediction							Ranking		
Explainer	Model	P ↑	R ↑	F1 ↑	AUPRC ↑	Empty ↓	SpanR ↑	Cover ↑	IOU ↑	P@5 ↑	R@5 ↑
Rand	$\mathrm{B_U}$	0.4±0.2	1.0±0.8	0.4±0.1	0.2±0.3	66.7±32.6	1.2±0.8	2.4±1.9	0.1±0.0	0.1±0.0	0.1±0.0
	PGD	0.4±0.2	1.0±0.8	0.4±0.1	0.5±0.8	66.9±32.3	1.2±0.8	2.5±2.0	0.1±0.0	0.1±0.0	0.1±0.0
	TM	0.4±0.1	1.3±0.9	0.4±0.0	0.6±0.9	60.2±32.9	1.5±1.0	3.0±2.3	0.1±0.0	0.1±0.0	0.1±0.0
	IGR	0.4±0.2	0.8±0.7	0.4±0.1	0.4±0.6	74.0±30.6	0.9±0.8	1.9±1.9	0.1±0.0	0.1±0.0	0.1±0.0
Attention	$\mathrm{B_U}$	36.2±2.5	36.8±4.1	36.4±2.7	31.4±2.9	22.2±3.1	52.4±4.2	61.1±3.4	38.1±2.3	29.0±2.1	39.0±2.8
	PGD	**38.5±2.8**	**37.3±2.0**	**37.7±1.1**	**33.1±1.4**	22.6±3.7	**52.8±1.8**	61.0±3.4	**39.1±1.4**	**30.0±0.8**	**40.4±1.1**
	TM	**37.3±2.9**	36.6±4.2	**36.8±3.0**	**31.9±3.2**	23.5±3.3	52.1±4.4	60.6±3.7	**38.9±2.7**	**29.5±2.2**	**39.7±3.0**
	IGR	**38.0±1.6**	**39.1±1.8**	**38.5±1.1**	**33.5±1.4**	**21.2±1.6**	**54.5±1.8**	**62.1±1.7**	**39.6±1.4**	**30.4±0.7**	**41.0±0.9**
Rollout	$\mathrm{B_U}$	1.7±0.0	23.8±1.1	3.2±0.1	0.3±0.0	0.0±0.0	24.5±1.3	31.0±1.5	0.4±0.0	0.5±0.0	0.7±0.0
	PGD	1.7±0.1	24.8±5.9	3.1±0.2	0.4±0.0	0.0±0.0	25.4±6.4	31.3±6.1	0.4±0.0	0.5±0.0	0.7±0.0
	TM	1.8±0.0	25.0±1.1	3.3±0.1	0.4±0.0	0.0±0.0	25.8±1.3	32.0±1.4	0.4±0.0	0.5±0.0	0.7±0.0
	IGR	1.7±0.1	25.2±4.4	3.2±0.2	0.4±0.0	0.0±0.0	25.5±4.7	31.7±4.5	0.4±0.0	0.6±0.0	0.8±0.0
$a\nabla a$	$\mathrm{B_U}$	35.7±4.9	34.4±3.7	34.9±3.6	29.5±3.7	18.0±2.2	50.2±4.1	61.6±3.3	37.0±4.4	28.5±3.1	38.3±4.2
	PGD	**38.9±3.3**	**36.9±2.4**	**37.7±1.4**	**32.7±1.9**	**16.9±1.9**	**52.9±2.5**	**63.8±1.4**	**40.3±1.8**	**30.9±1.1**	**41.6±1.5**
	TM	34.6±5.3	**36.3±3.5**	**35.3±3.9**	**30.0±3.9**	**16.5±2.5**	**52.2±3.8**	**63.5±3.1**	**37.4±4.5**	**28.9±3.2**	**38.9±4.3**
	IGR	**38.1±3.0**	**39.1±2.6**	**38.4±1.1**	**33.2±1.1**	**15.7±1.9**	**55.5±2.5**	**65.5±2.2**	**41.2±1.5**	**31.3±0.8**	**42.2±1.1**
InputXGrad	$\mathrm{B_U}$	32.4±2.6	31.9±1.8	32.0±0.8	25.7±1.3	10.1±2.1	45.6±1.9	60.8±1.8	31.6±1.0	26.8±0.6	36.0±0.8
	PGD	31.1±1.7	31.7±2.2	31.3±1.0	24.5±1.6	9.7±1.7	45.2±2.5	60.6±2.8	30.7±1.0	26.2±0.7	35.3±1.0
	TM	32.3±1.9	**34.3±2.1**	**33.1±0.9**	**26.9±1.1**	8.8±1.4	**48.3±2.2**	**63.5±2.1**	**32.7±0.9**	**27.5±0.6**	**36.9±0.8**
	IGR	**33.9±2.9**	**33.0±2.8**	**33.2±0.7**	**27.6±0.9**	10.5±2.7	**47.1±2.9**	**62.4±2.9**	**33.2±0.9**	**27.6±0.5**	**37.1±0.7**
IG	$\mathrm{B_U}$	33.0±2.4	32.9±2.2	32.9±1.7	26.6±3.5	5.9±1.4	49.3±2.6	64.3±2.6	34.5±3.3	27.9±1.1	37.5±1.5
	PGD	32.2±4.5	31.8±1.9	31.8±2.0	25.7±2.6	**5.7±2.1**	48.0±2.3	63.1±2.4	34.1±1.8	26.9±1.3	36.2±1.7
	TM	**33.8±2.7**	**33.6±2.6**	**33.6±1.7**	**27.7±3.0**	5.9±1.3	**50.6±3.0**	**65.2±2.9**	**35.9±3.1**	**28.4±1.0**	**38.2±1.3**
	IGR	**34.3±2.5**	**33.5±2.2**	**33.8±1.5**	**27.9±2.4**	5.9±1.2	**49.6±2.5**	64.3±2.5	**35.6±1.7**	**28.4±0.7**	**38.2±0.9**
Deeplift	$\mathrm{B_U}$	30.6±1.5	32.5±1.6	31.5±0.8	24.9±1.4	8.3±1.2	46.4±1.7	62.1±1.4	30.7±0.9	26.2±0.6	35.3±0.8
	PGD	30.2±2.2	31.7±2.8	30.8±0.9	24.0±1.3	8.9±2.1	45.3±3.3	61.1±3.0	30.0±1.0	25.8±0.7	34.7±0.9
	TM	**33.2±1.2**	32.0±2.4	**32.5±0.9**	**26.3±1.0**	9.4±1.5	46.1±2.7	62.0±2.7	**31.9±0.7**	**27.1±0.5**	**36.5±0.6**
	IGR	**33.7±2.3**	32.0±2.2	**32.7±0.7**	**26.8±1.1**	**10.3±2.0**	46.0±2.2	61.8±2.6	**32.2±1.1**	**27.2±0.6**	**36.5±0.8**
AttInGrad	$\mathrm{B_U}$	41.2±1.1	42.1±3.2	41.5±1.6	37.0±1.6	2.8±0.6	59.1±3.0	73.5±2.0	43.2±1.3	32.7±1.1	44.0±1.5
	PGD	39.7±1.8	**44.0±2.5**	**41.7±0.8**	**37.2±0.9**	**2.3±0.5**	**60.8±2.3**	**74.4±1.8**	**43.3±0.7**	**32.9±0.4**	**44.2±0.6**
	TM	**41.4±1.7**	**43.2±3.1**	**42.2±1.7**	**37.7±1.8**	**2.7±0.6**	**60.3±2.8**	**74.8±2.1**	**43.7±1.4**	**33.0±1.1**	**44.4±1.5**
	IGR	40.9±2.3	**43.9±2.5**	**42.2±0.5**	**38.1±0.6**	**2.5±0.8**	**60.8±2.2**	**74.2±2.1**	**44.1±0.5**	**33.2±0.4**	**44.7±0.5**