Evaluating Explainability with LLM for Classification of Patient Safety Event Reports

Zhifei Dou

Supervisor: Eldan Cohen

Department of Mechanical and Industrial Engineering

University of Toronto

May 29, 2025

Contents

1 Abstract

Patient Safety Event (PSE) reports are essential for healthcare quality improvement; however, their classification presents challenges in consistency and correctness. While machine learning (ML) offers various solutions, traditional methods usually lack effective explainability, resulting in limited clinical utility and potentially biased insights. This project investigates the capabilities of Large Language Models (LLMs), specifically focusing on two primary methodologies: 1) Fine-tuning established transformer models including RoBERTa, DeBERTa, GPT2 nad evaluating post-hoc explainability techniques (Saliency, InputXGradient, LIME, Occlusion) using metrics such as faithfulness, confidence indication, and consistency. 2) Utilizing prompt-based generative LLMs to classify reports and generate Natural Language Explainations (NLEs), with explainability assessed through faithfulness tests (counterfactual editing) and factual correctness evaluations (BioBERT Entailment, FActScore).

Key findings indicate that the tested LLMs achieved classification performance comparable to baseline models. For post-hoc explainability techniques, no single explainability technique excelled across all evaluation metrics. To be specific, gradient-based techniques with L2-norm aggregation showed promise in faithfulness, LIME has demonstrated the best confidence indication across all models, and Occlusion has shown the best data consistency. In the generative approach, instruction-tuned and reinforcement-learning-trained models have demonstrated better faithfulness in their explanations compared to distilled models. On the other hand, evaluations also suggest potential issues with the factual correctness of the generated NLEs, emphasizing the demand for robust manual validation. This work has underscored the potential of LLMs in PSE report analysis while suggesting the critical importance of evaluating the reliability of explanations generated by these models.

2 Introduction

Patient Safety Report (PSE) reports are documents that record unexpected events, errors, accidents or situations that affect or potentially affect a patient's care in hospital [1]. Such reports are a way healthcare organizations can capture incidents that can be reviewed and improved in the future. The publication of the study 'To Err Is Human' has demonstrated to the public the essentialness of preventing patient injuries; thus, also identifying the essentialness of correct classification for these reports[2].

Currently, there have been several works on correctly classifying the PSE reports with machine learning models. For example, previous works done by Chen et al. have implemented a comprehensive evaluation for machine learning methods (SVM, K-nearest Neighbours) with static embeddings (Word2Vec, GloVe, etc.) and also context embeddings (BERT embedding, RoBERTa embedding)[3].

However, even though Chen et al. have adopted an explainability technique - LIME after classification to explain how the prediction is made by the machine learning model, there are many other explainability techniques, such as Saliency, InputXGradient, and Occlusion, not been implemented. Leaving the question: 'If different explainability techniques produce different saliency scores, which technique's explanation should we trust?' Also, the explainability technique evaluation method proposed by Chen et al. has only focused on the factual correctness of the explainability tools, leaving question mark about the faithfulness of the experimented models. In other words, we don't know if the produced model's explanation reflects its true reasoning path [3]. Therefore, in this project, the team has first adopted transformer encoder-only Large Language Models (LLMs), RoBERTa and its variants with three different types of explinability techniques (gradient-based, perturbation-based based and simplification-based) along with four evaluation metrics, including faithfulness, confidence indication, data consistency, and rationale consistency, proposed by Atanasova et al. to classify, explain, and evaluate the PSE reports with model's explaination [4] .

For all explainability techniques listed above, there is a critical limitation that their saliency map explanation can only highlight the 'critical' words in the input PSE report rather than Natural Language Explanation (NLE). The study from Yanagawa et al. has shown that showing traditional explanation results such as highlighting critical words or saliency maps has a limited benefit on clinical performance[5]. The clinical usefulness of such an explainability tool is thus doubtful. The study by Ghassemi et al. points out that traditional explainability tools sometimes even distract doctors from the true essential sections; thus, producing biased judgement[6].

As transformer-decoder auto-regressive LLMs, such as QwQ from Alibaba AI, have demonstrated a decent ability in the medical domain and perform general text classification tasks, adopting such models can potentially have better classification and explanations than the machine-learning models[7, 8]. Hence, in this project, the team has comprehensively adopted GPT-like generative LLMs to classify PSE reports. Considering the confidential issue of using API and the expensive cost of deploying full-size LLMs, small to middle-scale open-sourced LLMs are mainly deployed locally in this project.

Meanwhile, NLE is generated along with the predicted label from the LLM to reveal the rationales behind the models' decisions. By providing NLE as an explanation, a more human-like and complete rationale is provided and potentially augments the clinician's knowledge; thus solving the issue of traditional explainability tools[9]. On the other hand, studies have shown that NLE is also facing challenges regarding faithfulness and risk of generating hallucinations[10]. Thus, evaluation metrics for NLE, such as counterfactual tests and FActScore, are implemented to validate the explanation's quality in terms of faithfulness and factual correctness.

The structure of this project is visualized in following Figure 1.

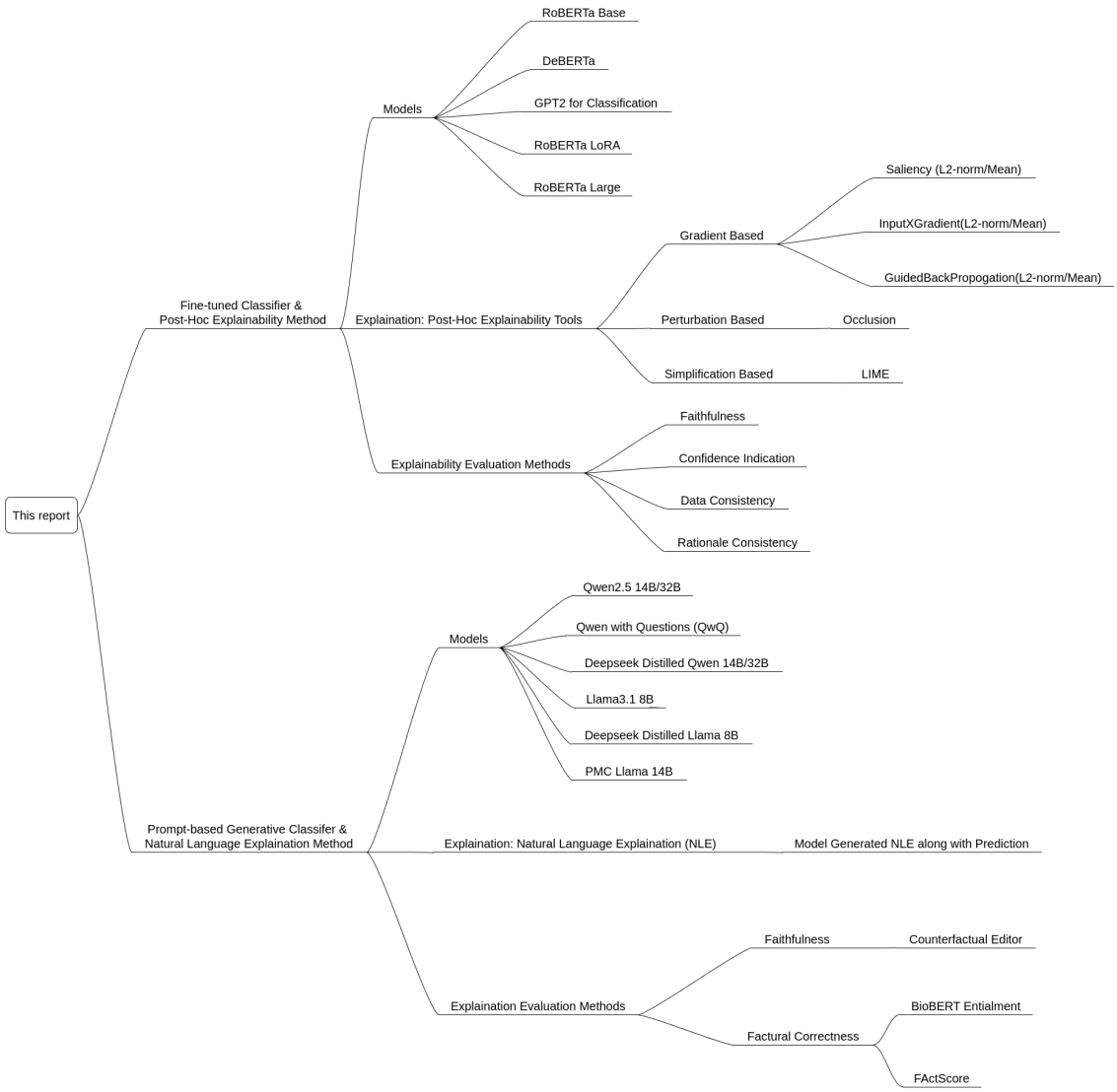

Figure 1: Project Structure Map

The code of this project can be found in:
`https://github.com/ZhifeiDou/Explanability-in-PSE-report-Classification.git`

3 Related Work

Prior to this work, the following list of works has built a solid foundation and required as a prerequisite reading for this paper.

As introduced, prior research by Chen et al. has explored the classification of PSE reports with various machine learning models, including Support Vector Machines (SVM) and K-nearest Neighbors, combined with different text embedding techniques, such as static Word2Vec, GloVe, and contextual BERT, RoBERTa embeddings. The classification performance achieved in their study serves as a baseline for this project. [3].

Furthermore, RoBERTa proposed by Liu et al. and its variants, BioMed RoBERTa proposed by Gururangan et al., DeBERTa proposed by He et al. serve as encoder-only classification models in this project[11, 12, 13]. GPT2 proposed by OpenAI is used with a classification layer as the decoder-only classification model in this

project [14]. Qwen2.5 series and QwQ proposed by Alibaba AI and Llama3.1 series proposed by Meta AI are used as generative classification models [15, 7, 16]. Deepseek has also proposed their distilled version of Qwen and Llama using their first reasoning model Deepseek-R1, these distilled models are also used as generative classification models. To investigate the effect of continuous pre-training, we have also adopted the medical variation of Llama - PMC Llama proposed by Wu et al[17].

Several fine-tuning and generation techniques have also been implemented in this project. Low-Rank Adaptation, proposed by Hu et al. is also used during model fine-tuning to research its impact on explainability tools[18]. Self-consistency and Retrieval Augmented Generation (RAG) are utilized to boost generative models' classification performance[19, 20].

Explanation for predictions is essential for the medical domain. Multiple explainability techniques based on gradient, perturbation, and simplification are utilized to explain fine-tuned models in this project: Saliency, InputXGradient, GuidedBackpropagation, LIME, and Occulsion[21, 22, 23, 24, 25]. Evaluation metrics, including faithfulness, confidence indication, data consistency, and rationale consistency, proposed by Atanasova et al. serve as our tool to assess explainability techniques[4]. For generative classification models, models are prompted to produce Natural Language Explanation (NLE) as their explanation for the prediction. Methodologies for evaluating the quality of NLE have been proposed in many other studies. Specifically, the work proposed by Atanasova et al. and Valentino et al. introduced methods to evaluate NLEs [9, 26]. These works inform the evaluation strategies employed in this project for assessing the explanations generated by the LLMs.

4 Data and Data Processing

4.1 PSE Report Dataset Collection

The PSE report dataset utilized in this project was obtained from 'Labor and Delivery' and 'Mother-baby' units of an academic hospitality organization located in the southeast of the United States. The dataset consists of 861 PSE reports which were issued from January 1st, 2019 to December 31st, 2020, along with the class labels. All PSE reports are anonymized according to the privacy regulation requirement.

4.2 Data Preprocessing

Then, the 'Report Description' and 'Event Type' are extracted as the free-text feature and target. To avoid sampling bias, only major classes are kept. The kept classes and the number of data pieces they contain are demonstrated in the following table:

Table 1: Processed PSE Dataset Classes and Frequencies

Class Name	**Frequency**
Care coordination / communication	186
Laboratory test	122
Medication related	89
Omission / errors in assessment, diagnosis, monitoring	67
Maternal	58
Equipment / devices	56
Supplies	49

The dataset has been split into training, validation, and testing subsets with each having portions of 60%, 20%, and 20%.

5 Method

The team proposes two methods to classify the PSE report which are: Fine-tuned Classification method and Prompt-based Generation method.

5.1 Fine-tuned Classifier and Post-Hoc Explainability Method

The first methodological component of this research centers on the classification of the PSE reports utilizing fine-tuned transformer LLMs. This method is conducted in conjunction with a systematic evaluation of post-hoc explainability techniques designed to explain model predictions. The method encompasses the adaptation of pre-trained language models followed by the application and assessment of implemented explainability methods.

5.1.1 Model Architectures and Fine-Tuning Protocol

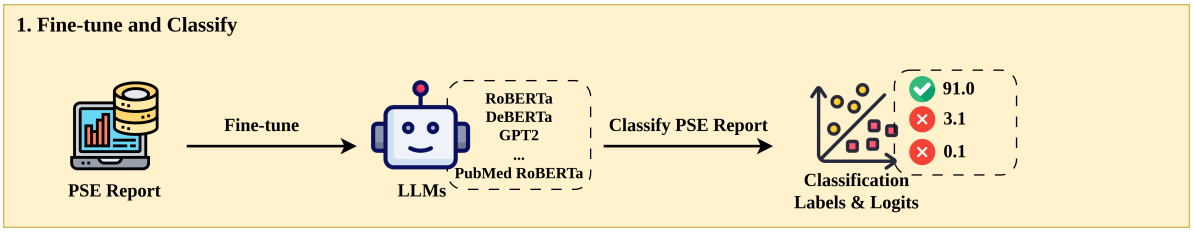

Figure 2: Fine-tuned Classifier and Post-Hoc Explainability Method Pipeline1

As demonstrated in Figure 2's process, a selection of transformer LLMs, which varying in architecture and pre-training data, are adapted in this method along with appending a linear classification layer as the classification head. The chosen models include both encoder-only and decoder-only based architectures:

RoBERTa (Base) RoBERTa is selected as a foundational model due to its improvements over the original BERT architecture, establishing state-of-the-art results on several NLP benchmarks. Its robustness stems from several pre-training modifications outlined following: optimized Masked Language Model(MLM) training, dynamic masking, expanded training data and duration, and byte-level BPE tokenization.

DeBERTa Characterized by its disentangled attention mechanism, which distinctly models content and positional embeddings, potentially yielding better performance gains over RoBERTa series architectures. [13].

RoBERTa-Large A scales-up version of RoBERTa with increased parameters, also potentially capturing more complex data patterns[11].

BioMed-RoBERTa BioMed-RoBERTa, proposed by Lewis et al., is a domain-specific RoBERTa which has further pre-trained on an extensive biomedical corpus from biomedical literature (e.g. PubMed), thereby augmenting its capacity for processing specialized terminology inherent in PSE reports [27] .

GPT2 Meanwhile, for the Decoder-based transformer model, we have implemented GPT2 for sequence classification, which is the standard GPT2 model substituting its generative head with a classification layer.

All the models implemented in this method and their model sizes are listed in the following table:
All models listed above are fine-tuned on the designated training dataset. Optimization was performed using Cross-Entropy Loss, with learning rates adjusted via a predefined schedule over 20 training epochs. To

Table 2: LLMs with Classification Head Chosen in this Project

Model Name	Number of Parameters (Million)
RoBERTa	125
DeBERTa	184
RoBERTa Large	355
GPT2	124
BioMed RoBERTa	125

investigate the potential influence of parameter-efficient fine-tuning's paradigms on the explainability tools, Low-Rank Adaptation (LoRA) has been implemented on the base RoBERTa model for comparison.

5.1.2 Explainability Tools for Classification Head Method

Following the fine-tuning, as visualized in Figure 3,numerous established post-hoc explainability techniques were applied to explain the prediction of the models. These techniques produce token-level attribution scores, reviewing the inferred importance of each input token to the result of classification. There are three categories for the implemented explainability techniques.

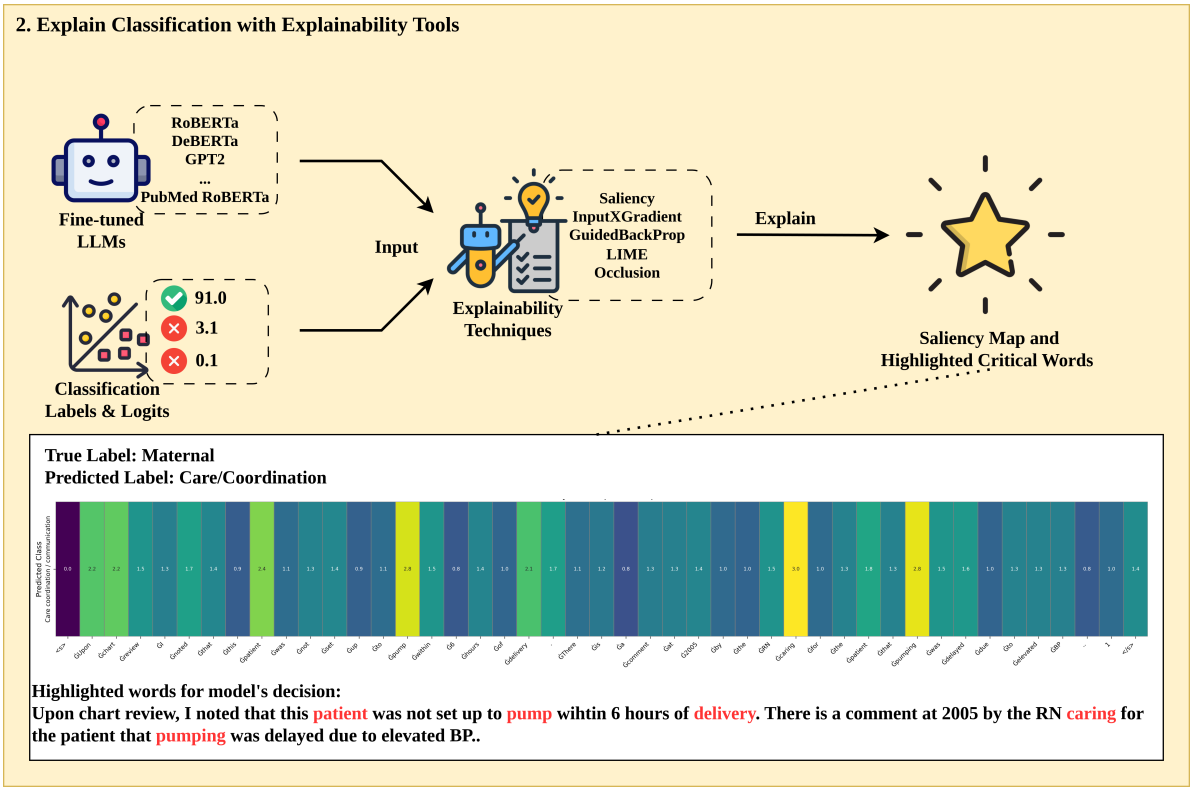

Figure 3: Fine-tuned Classifier and Post-Hoc Explainability Method Pipeline2

Beginning with the **gradient-based** explainability approaches, we have implemented Saliency along with its variants of InputXGradient and GuidedBackpropagation [21, 22, 23]. Given that token embeddings are multi-dimensional, scalar attribution scores per token were derived by applying either **L2-normalization (L2-norm)** or **averaging (mean)** aggregation across the embedding dimension of the resulting gradient vectors.

Saliency The Saliency computes the first-order partial derivatives of the predicted class score with respect to the input token embeddings[21]. Such derivatives represent the sensitivity of the output to infinitesimal changes in the input features[21].

InputXGradient InputXFradient calculates the element-wise product of the input embeddings and their corresponding gradients[22]. This technique effectively weights the sensitivity by the magnitude of the input feature itself [22].

GuidedBackpropogation GuidedBackpropogation modifies the standard backpropagation algorithm, particularly through the ReLU activation functions[23]. Such modification filters out negative gradients during the backward pass; therefore, it visualizes features that positively contribute to the neuron activations at higher layers[23].

Then, **perturbation-based** method determined feature importance by systematically altering part of the input and measuring the consequent impact on model output probability. Occlusion is the only technique that falls into this method[25].

Occlusion This technique assesses token importance by iteratively replacing each token within the input sequence with a predefined baseline value. The change in the model's output probability for the target class. Results from this replacement are used as the attribution score for the occluded token [25]. Following standard practice, a zero vector was employed as the baseline token representation [25].

Last comes the **simplification-based method**, this method involves approximating the local behavior of the non-linear complex model with a more interpretable surrogate linear model. LIME is the technique under the umbrella of this method[24].

LIME LIME, which stands for Local Interpretable Model-agnostic Explanations, explains individual predictions by learning a simpler, interpretable model, such as linear regression, within the vicinity of the instance being explained. It generates a neighborhood of perturbed samples around the instance, gets predictions of these samples from the original complex model, and then trains the interpretable model on this created local dataset with weighted by proximity to the original instance. The coefficients of the learned linear model is served as explanations for the feature importance[24].

5.1.3 Evaluation of Explainability Techniques for Classification Method

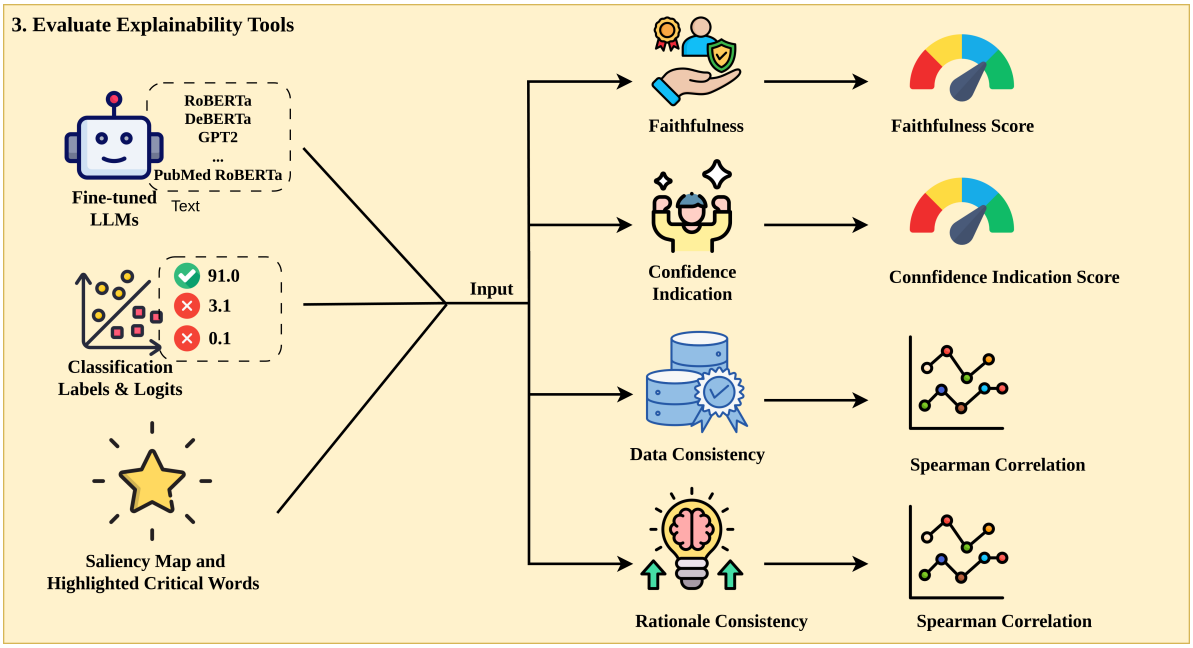

Figure 4: Fine-tuned Classifier and Post-Hoc Explainability Method Pipeline3

The trustworthiness of the applied post-hoc explainability techniques were assessed using a diagnostic framework derived from Atanasova et al.[4]. The evaluation process is demonstrated in Figure 4.This evaluation aimed to quantify the extent to which explainability techniques were introduced before providing reliable insights into the model's prediction process. The assessment consists of three aspects: Faithfulness, Confidence Indication, and Consistency.

Faithfulness Faithfulness is one of the most important attributes for evaluating the explainability tool. It demonstrates if the model's explanation reflects the model's internal decision path and is not based on random choices. To quantify faithfulness we follow the token–removal diagnostic proposed by Atanasova et al.,[4].

Let $X = \{x^{(1)}, \ldots, x^{(N)}\}$ be the evaluation dataset, M is the fine-tuned classifier, and $\omega_{M,x} \in R^{|x|}$ an attribution vector that assigns an importance score to every token of x. For a masking percentage $i \in \{0, 10, 20, \ldots, 100\}$ we construct a perturbed set

$$X_i^{\omega} = \Big\{ \text{Mask}\big(x, \omega_{M,x}, i\big) \;\Big|\; x \in X \Big\}, \tag{1}$$

where $\text{Mask}\big(x, \omega_{M,x}, i\big)$ replaces the top-i% tokens (ranked by $\omega_{M,x}$) with a neutral `[MASK]` symbol (default mask tokens are used for BERT-like models, a new mask token created for GPT2). Denoting the task metric by $P(\cdot)$ (e.g. macro-F_1), the performance drop after masking i% of the most salient tokens is

$$\Delta_i = P\big(M(X_0^{\omega})\big) - P\big(M(X_i^{\omega})\big). \tag{2}$$

Faithfulness is summarised by the *area under the threshold–performance curve* (AUC-TP):

$$\text{AUC-TP}(\omega, M, X) = \int_0^1 \Big(P\big(M(X_0^{\omega})\big) - P\big(M(X_{\alpha}^{\omega})\big) \Big)\, d\alpha, \qquad \alpha = \frac{i}{100}. \tag{3}$$

In this project, the integral is approximated by a Riemann sum over the discrete percentages $i = 0, 10, \ldots, 100$. A **lower** AUC-TP indicates a more dramatic performance decline, which identifies more faithful attributions because the fine-tuned model quickly loses accuracy when its most important tokens, as identified by ω, are removed.

Confidence Indication When a model detects a very indicative pattern (strong output value after softmax a.k.a high confidence) of the particular class, the token forming that pattern should have strongly positive saliency scores for that class, and strongly negative scores for other classes. Thus, there is a linear regression model used to intake the saliency scores distance of all tokens with respect to this class and predict the output value after softmax. If the Mean Square Error of that prediction is high it means the tested saliency score is not reflecting the model's confidence thus also not faithful.

Also following Atanasova et al. [4], we tested if an explanation vector $\omega_{M,x}$ can predict how confident the model is in its own prediction. For each instance x_i with predicted class k and confidence $p_{i,k} = M_k(x_i)$, we first compute the *Saliency Distance* (SD) between the attribution scores for k and those for all *other* classes $K \backslash \{k\}$:

$$\mathrm{SD}(x_i) \;=\; \sum_{j=1}^{|x_i|} D\big(\omega_{M,x_i,j,k},\; \omega_{M,x_i,j,K\backslash\{k\}}\big), \tag{4}$$

where in original method, for $|K| = 2$, Atanasova et al. simply take the difference $D(a, b) = a - b$; however, for multiclass problem in this project we need to concatenate $\big[\max(b-a),\, \min(b-a),\, \mathrm{avg}(b-a)\big]$. Next, a logistic-regression surrogate LR is trained on SD vectors (derived from a training split) to predict the model's confidence. Performance is evaluated via the Mean Absolute Error (MAE):

$$\mathrm{MAE}(\omega, M, X) \;=\; \frac{1}{N} \sum_{i=1}^{N} \big| p_{i,k} \;-\; \mathrm{LR}\big(\mathrm{SD}(x_i)\big) \big|. \tag{5}$$

A **lower** MAE means the explanation scores carry richer information about the classifier's own certainty, hence provide a better confidence indication.

Rationale Consistency and Data Consistency The rationale consistency is simply built on the intuition that models with similar structures should have closer saliency scores for each explainability technique. Also, for similar data instances, they are supposed to demonstrate similar saliency scores compared with other unsimilar instances.

First coming the **Rationale Consistency**, we wish to measure whether two models with the same architecture generate similar explanations when their internal reasoning paths are similar. Let $\mathcal{M} = \{M_s, M_p\}$ be a pair of such models, and let $D(\cdot, \cdot)$ denote a distance function (we use the ℓ_1–norm over the flattened vectors). For an instance x_i with gold label y_i we first compare models' logits.

$$d_i^{\mathrm{act}} \;=\; D\big(M_s(x_i),\, M_p(x_i)\big). \tag{5}$$

We then compare the attribution scores produced by an explanation technique ω for the *same class* y_i:

$$d_i^{\mathrm{sal}} \;=\; D\big(\omega^{M_s}_{x_i,y_i},\, \omega^{M_p}_{x_i,y_i}\big). \tag{6}$$

Finally, Rationale Consistency is the *Spearman rank correlation* between these two sets of distances across all N instances:

$$\mathrm{RC}(\omega, \mathcal{M}, X) \;=\; \rho_{\mathrm{Spearman}}\Big(\{d_i^{\mathrm{act}}\}_{i=1}^{N}, \{d_i^{\mathrm{sal}}\}_{i=1}^{N}\Big). \tag{7}$$

A higher positive correlation means that whenever two models reason similarly (low d_i^{act}), they also assign similar saliency patterns (low d_i^{sal}), indicating a more consistent attribution method.

Meanwhile, **Data Consistency** tests whether similar instances receive similar explanations from the same model. Given a single trained model M and two instances x_i, x_j with identical gold label $y_i = y_j$, we compute

$$d_{i,j}^{\text{act}} \;=\; D\big(M(x_i),\, M(x_j)\big), \tag{8}$$

$$d_{i,j}^{\text{sal}} \;=\; D\big(\omega_{x_i,y_i}^{M},\, \omega_{x_j,y_i}^{M}\big). \tag{9}$$

Sampling P pairs (i, j) from the dataset (mixture of high and low lexical overlap, as proposed in the original study), we again take the Spearman correlation:

$$\text{DC}(\omega, M, X) \;=\; \rho_{\text{Spearman}}\Big(\{d_{i,j}^{\text{act}}\}_{(i,j)=1}^{P}, \{d_{i,j}^{\text{sal}}\}_{(i,j)=1}^{P}\Big). \tag{10}$$

A higher DC score implies that instances whose latent representations are close (small $d_{i,j}^{\text{act}}$) also share similar explanations (small $d_{i,j}^{\text{sal}}$), signalling a more dataset-consistent explanation method.

5.2 Prompt-based Generative Classifer & Natural Language Explaination Method

This method has prompted generative LLMs with an auto-regressive generation head as the last layer. In this section, we have discussed the rationales for the selection of generative LLMs, classification methods, and evaluation methods.

5.2.1 Model Selection for Generative Model Method

Currently, the latest full-size generative LLMs, such as ChatGPT-4o and Deepseek-V3, are usually deployed on cloud service. Meanwhile, PSE reports are usually confidential to protect the personal information of patients and healthcare staff [28]. Access to such models through cloud services has the risk of information leakage[29]. On the other hand, deploying such full-size generative LLMs locally is not always affordable for all healthcare organizations[30]. Also, recent studies from Qiu et al. have shown even small or middle reasoning models performed decently in the medical domain's tasks[8]. Hence, the team proposed experimenting with the performance of open-sourced small-scale models such as Llama-8B from Meta and Qwen-14B from Alibaba [16][15]. Also, the model distillation technique allows small-scale LLMs to learn reasoning ability from full-size LLMs. Deepseek has released distilled Qwen and Llama models from their first reasoning model Deepseek-R1, which potentially produces better explanations for the PSE report classification task[31]. Similarly, Alibaba has released its first reasoning Qwen: Qwen with Question (QwQ) highlighting outstanding performance in reasoning benchmarks[7]. To investigate the impact of continuous pre-training, we have also adopted the PMC Llama proposed by Wu et al. which is the Llama embedded with medical domain knowledge and instruction fine-tuned for medical question answering [17]. Hence, the team has chosen models demonstrated in the following table 3.

Table 3: LLMs with Generative Head Chosen in this Project

Model Name	**Number of Parameters**
Qwen2.5	14B, 32B
QwQ	32
Deepseek Distilled Qwen	14B, 32B
Llama3.1	8B
Deepseek Distilled Llama	8B
PMC Llama	14B

5.2.2 Classification Methods in Generative Model Method

todo: add rag as and sc in to the diagram All the following tasks are implemented on a single RTX4090 24G, 4-bit quantization, all LLMs and their tokenizers are downloaded from Huggingface. The seed is fixed to 42 for reproducibility. Due to the lack of a human-labelled golden explanation, instruction-based learning is the most feasible learning strategy for this project. The team has prompted the models listed in table 3 in zero-shot, few-shot, and Retrieval Augmented Generation(RAG) format along with techniques including explanation-based prompting, self-consistency, Chain-of-Thought(CoT) and persona embedding to increase the model's performance in a classification task. To make the result comparable with the other project, the testing dataset is used for the classification task which contains 124 PSE reports. All Qwen instruction series models are applied to chat templates for proper outputs.

Zero-shot Prompting Zero-shot prompting is the technique where generative LLMs are given an instruction prompt to perform the task without examples. The zero-shot prompt (base prompt) is demonstrated in the main body table 4. For zero-shot and RAG in the following, we adopted Top-P as 0.9 and temperature as 0.7 for all models due to the common recommendations from models using guidance[7, 15]. The process is visualized in Figure 5

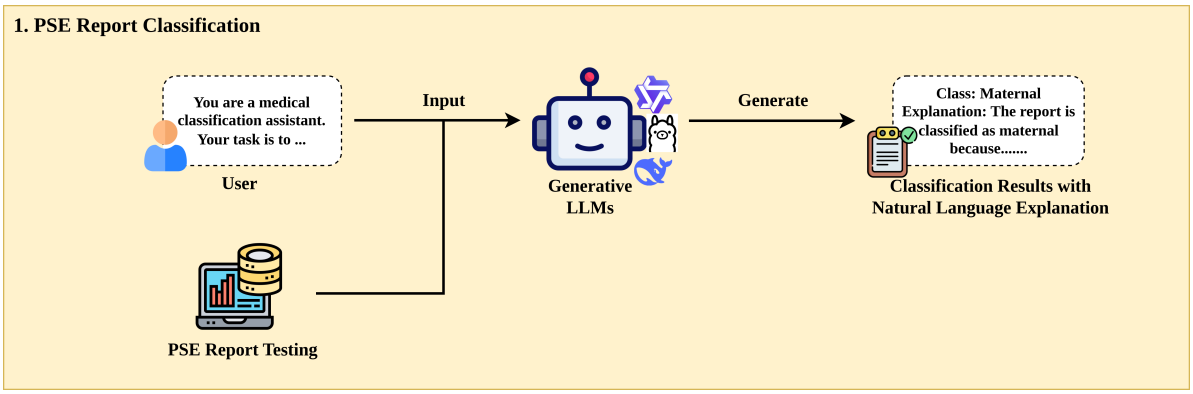

Figure 5: Generative Model I/O for Zero Shot Classification Task

Retrieval-Augmented Generation Retrieval-Augmented Generation proposed by Lewis et al. is a framework that combines the generative LLMs with a retrieval mechanism[20]. In this project, we use a separate BioBERT model's embedding as the representation due to its embedded biomedical knowledge within the embedding[32]. The PSE report contents of the training dataset are mapped into BioBERT embedding. We use Faiss from the Meta AI team as the similarity search tool to retrieve the closest content from the training dataset to the given one. Then, the following prompt demonstrated in table 4 is input into the generative LLM with the retrieved content. The process for RAG classification is demonstrated in Figure 6

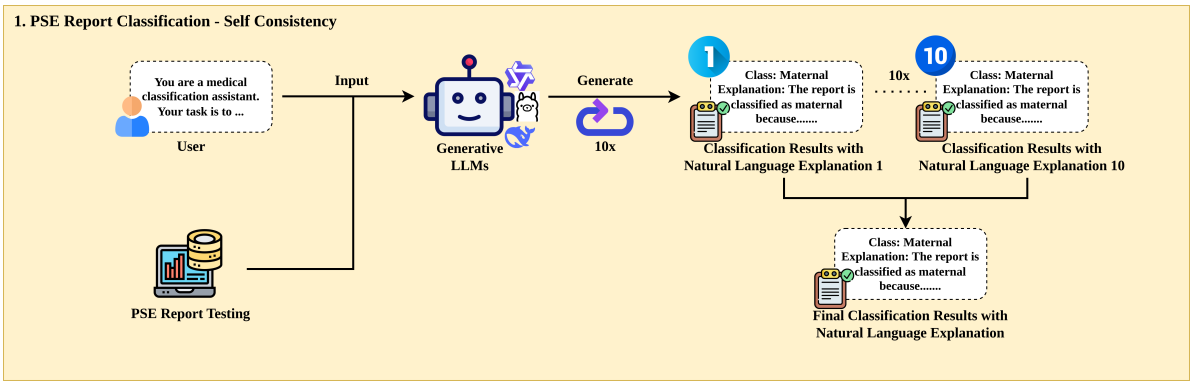

Figure 6: Generative Model I/O for Self Consistency Classification Task

Self-consistency Self-consistency is a decoding strategy designed to improve the reasoning abilities of generative LLMs. One of the most effective self-consistency methods proposed by Wang et al. is making generative LLMs generate multiple distinct reasoning paths for a given task or problem and perform majority voting to determine the final decision [19]. Following the original work, we have also performed hyperparameter tuning on key factors of temperature, Top K, and the number of reasoning paths sampled for each model and elect the best-performed hyperparameter settings. The results show that the best setting is temperature = 0.7, Top K = 40, and 10 as the sample number. The prompt in table 4 is also used. The visualized process is shown in Figure 7. The majority voting process can be demonstrated as the following functions:

$$\text{Given sampled\_classes} = \{c_1, c_2, \ldots, c_n\} \quad \text{and} \quad \text{sampled\_explanations} = \{e_1, e_2, \ldots, e_n\},$$

$$\text{let } \text{freq}(c) = \sum_{i=1}^{n} \mathbf{1}(c_i = c).$$

$$\text{Then, final\_class} = \arg\max_{c} \text{freq}(c),$$

$$\text{and we define final\_explanation} = e_j \quad \text{where } j = \min\{\, i \mid c_i = \text{final\_class} \}.$$

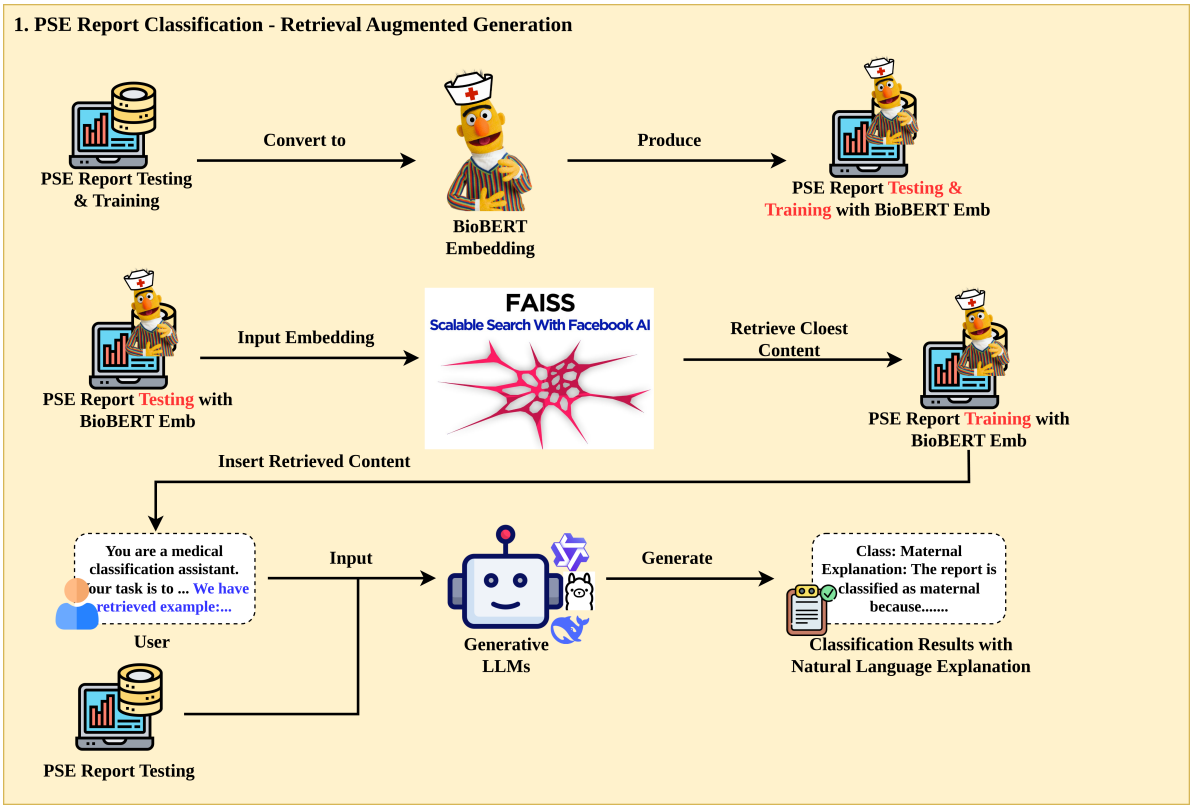

Figure 7: Generative Model I/O for RAG Classification Task

Table 4: LLMs Prompting for Classification and Explanation

Prmpt: Base prompt adopted (definition for each class omitted)
You are a medical classification assistant. Your task is to read the text below and decide which of the following categories best applies: if RAG: Below are some relevant records from our training dataset (content + label): {retrieved content} Categories (with brief definitions): Care coordination / communication: Incidents... Laboratory test: Definition: Laboratory test-related incidents ... Medication related: Medication-related incidents ... Omission / errors in assessment or diagnosis or monitoring: These incidents ... Maternal: Maternal safety incidents ... Equipment / devices: This category covers ... Supplies: Supplies-related incidents ... **Instructions**: 1. Read the text carefully and identify the key points or keywords that hint at one of the above categories. 2. Decide on the single most relevant category. 3. Provide a short explanation of how you arrived at this decision, referencing specific keywords or ideas from the text. 4. Think step by step like a professional doctor 5. Return the result in **exactly** the following Markdown format: **Class:** Your single best class here **Explanation:** Your explanation here Text: {PSE report content}

5.2.3 Models' Explanations in Generative Model Method

Through the prompting shown in table 4, the model is also introduced to articulate a step-by-step reasoning path in parallel with predicting the classification labels. This procedure yields an NLE alongside the prediction, producing a human-interpretable text rationale that demonstrates the model's decision-making process.

5.2.4 Evaluation of Explanations in Generative Model Method

Faithfulness For generative models, faithfulness measures if the explanation truly reflects the model's actual decision process [33]. To measure faithfulness, the team has set up a counterfactual editor and edited the input to see how many predictions changed the tested model, following the work proposed by Atanasova et al [9]. According to the proposed method, the team has adopted T5 as the base model for the counterfactual editor. The T5 model is fine-tuned with the following three losses: 1. Mask-filling loss. 2. Imitation loss. 3. Adversarial loss [9]. With this fine-tuned counterfactual editor, we produce and feed counterfactual PSE report content into the tested models. If the tested model's explanation is faithful, the model's prediction remains unchanged and vice versa [9].

Like the author, the counterfactual editor model we adopt is a T5-based generative model proposed by Google Brain and fine-tuned with the following loss[34].

First is the filling loss, where we randomly mask the PSE reports' word in content and let the T5 model predict the masked word. The cross-entropy loss of predicting this masked word is the filling loss demonstrated as the following equation.

$$\mathcal{L}_{\text{filling}} = -\sum_{t=1}^{|W|} \log\Big(p_\theta\big(w_t \mid w_{<t}, \text{Masked}(x_i)\big)\Big)$$

where $W = (w_1, w_2, \ldots, w_{|W|})$ is the sequence of ground-truth tokens that were masked, $p_\theta(\cdot)$ is the T5 probability distribution over vocabulary tokens, $\text{Masked}(x_i)$ denotes the original input x_i with a contiguous substring replaced by a placeholder, and we optionally condition on the current or target label y_i^C as an input prefix.

Then, we replace the mask with the predicted word by the T5 model and feed this processed PSE report content into the model we would like to test the faithfulness on, calling it the teacher model. We then take the explanation generated by the teacher model and let the T5 model mimic the teacher model's explanation. In other words, we perform a teacher forcing to the T5 model with the teacher model's explanation. The cross-entropy loss during this teacher forcing is called imitation loss, represented by the following function.

$$\mathcal{L}_{\text{imitation}} = -\sum_{t=1}^{|E^{(T)}|} \log\Big(p_\theta\big(E_t^{(T)} \mid E_{<t}^{(T)}, X\big)\Big)$$

where $E^{(T)} = \big(E_1^{(T)}, E_2^{(T)}, \ldots, E_{|E^{(T)}|}^{(T)}\big)$ is the teacher-provided explanation (token sequence), $p_\theta(\cdot)$ is the T5 probability distribution over vocabulary tokens, and X denotes the model's input which is processed PSE report content.

Finally, we take the logits of the mask-filling task and the explanation mimic task to calculate the difference between them, the negation of the difference is the adversarial loss which helps us to push these two logits as far away as possible. Hence, once we finish training, the text generated by the T5 model will be words that make the original content as far away from the explanation as possible. The adversarial loss $\mathcal{L}_{\text{adv}}$ can be demonstrated as the following function:

$$\text{fill\_mean}_b = \frac{1}{|\mathbf{F}_b|}\sum_{j=1}^{|\mathbf{F}_b|}\text{logits\_infill}_{b,j} \quad , \quad \text{expl\_mean}_b = \frac{1}{|\mathbf{E}_b|}\sum_{j=1}^{|\mathbf{E}_b|}\text{logits\_expl}_{b,j}$$

$$\text{diff}_b = \Big|\text{fill\_mean}_b - \text{expl\_mean}_b\Big|, \quad \overline{\text{diff}} = \frac{1}{B}\sum_{b=1}^{B}\text{diff}_b$$

$$\mathcal{L}_{\text{adv}} = \Big(-\overline{\text{diff}}\Big)$$

where $\text{logits\_infill}_{b,j}$ and $\text{logits\_expl}_{b,j}$ denote the respective logit values from (i) the editor's fill-generation pass and (ii) the editor's teacher-explanation pass, $|\mathbf{F}_b|$ and $|\mathbf{E}_b|$ are the total logit counts in each pass, B is the batch size, and λ_{adv} is the adversarial weight hyperparameter.

Then we put it all together:

$$\mathcal{L}_{\text{editor}} = \lambda_f\,\mathcal{L}_{\text{filling}} + \lambda_i\,\mathcal{L}_{\text{imitation}} + \lambda_a\,\mathcal{L}_{\text{adv}},$$

where $\mathcal{L}_{\text{filling}}$ is the cross-entropy filling loss, $\mathcal{L}_{\text{imitation}}$ is the teacher-forcing imitation loss, $\mathcal{L}_{\text{adv}} = -\overline{\text{diff}}$ is the adversarial term, and $\lambda_f, \lambda_i, \lambda_a$ are their respective weighting factors as hyperparameters.

Then the team has adopted this fine-tuned T5 model to edit the original PSE report content to produce counterfactual PSE report content for each model tested. The counterfactual content is used for each model to inference and produce predictions for counterfactual content. The number of differences between new predictions and original predictions directly correlated with the extent of the tested model's faithfulness.

The procedure of fine-tuning the counterfactual is visualized in Figure 8 and Figure 9.

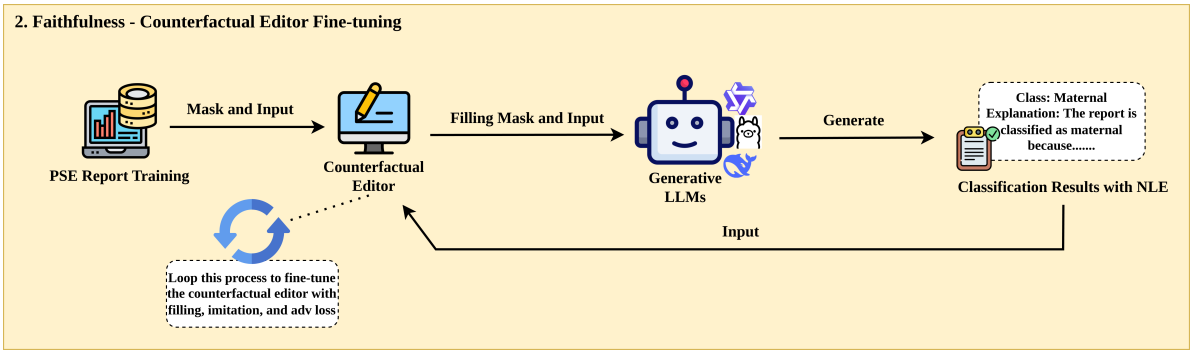

Figure 8: Fine-tuning Procedure Map for Counterfactual Editor - Fine-tune

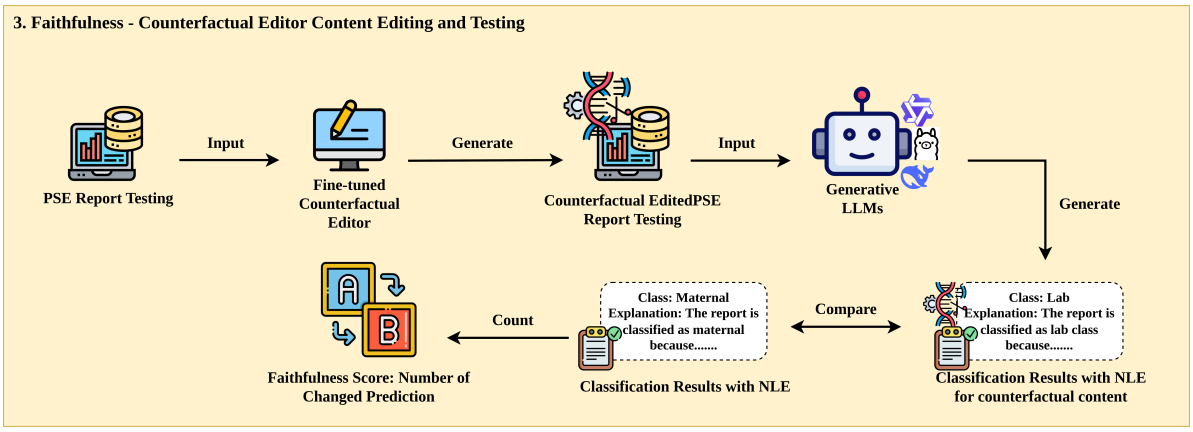

Figure 9: Fine-tuning Procedure Map for Counterfactual Editor - Edit

Factual Correctness Then we also need to evaluate the factual correctness of the explanation, which we propose to define as whether the explanation is true to the facts of the PSE report as well as the general medical knowledge. In this project, we have used three methods to evaluate this metric which are: the **BioBERT entailment method**, the **FActScore method**[35, 36].

The BioBERT entailment method, proposed by Jeong et al., is under the suggestion that: If an explanation is factually correct, it should have an entailment relationship with the PSE report content[35]. Work proposed by Valentino et al. also validated if an explanation is factually correct, it should be entailed by the input content[26]. Hence, following the method of Jeong et al., the team has fine-tuned a BioBERT model with the well-known e-SNLI entailment dataset proposed by Camburu et al[37] to test the factual correctness.

The BioBERT model is fine-tuned with the following cross-entropy loss function.

$$\mathcal{L}_{\text{CE}}(\theta) = -\sum_{i=1}^{N}\sum_{c=1}^{C} \mathbf{1}[y_i = c]\,\log\Big(p_\theta\big(c \mid x_i\big)\Big),$$

where x_i is the input text, which is premise and hypothesis from e-SNLI dataset, y_i is the true label which is one of the entailment, neutral, and contradiction, in e-SNLI. $p_\theta(c \mid x_i)$ is the predicted probability of class c under the model parameters θ.

After fine-tuning, the original PSE report content and the explanation generated by the tested model are formed as the premise and hypothesis input to the BioBERT model. If the BioBERT model's prediction is settled in neutral or entailment, it implies the explanation is factually correct, if the prediction is a contradiction, it implies the explanation potentially contains hallucination information. The procedure for using the fine-tuned BioBERT is represented in Figure 10.

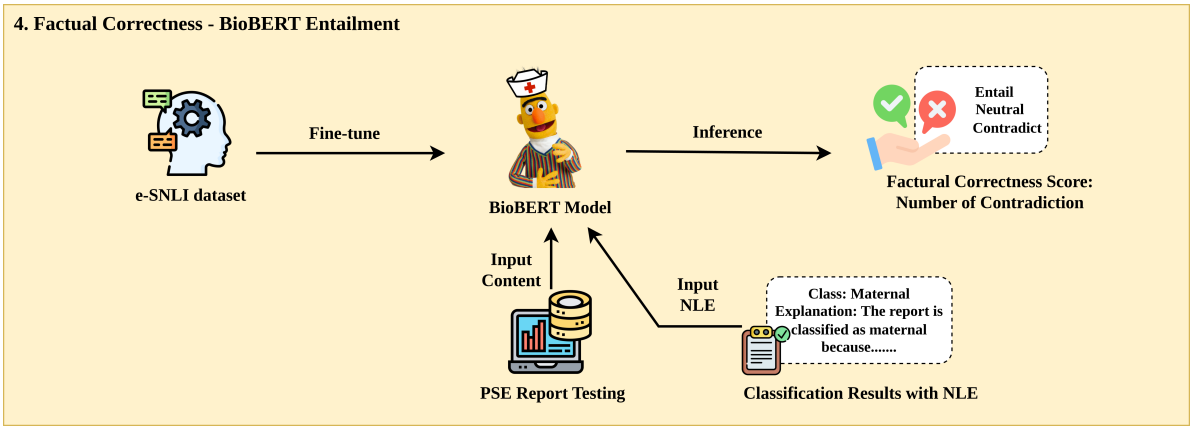

Figure 10: Fine-tuned BioBERT Entailment Procedure

Meanwhile, the team also adopted FActScore proposed by Min et al., which breaks the explanation into atomic facts and fact checks, to further validate the factual correctness of the explanation [36]. Due to the hardware constraint, we adopted Qwen14B as the assist model which is in charge of forming atomic facts and validating factual correctness. Following the original method, we have locally stored and indexed Wikipedia via pyserini. For each atomic fact, we retrieve the closest wiki content by BM25 as suggested in the original paper. Then the atomic fact and wiki content are input into the validator model, the model makes its judgement if the atomic fact is true according to the content. After all atomic facts are judged, the percentage of true predictions within all predictions would be the representation of model explanations' factual correctness. The FActScore process can be visualized with the following figure 11.

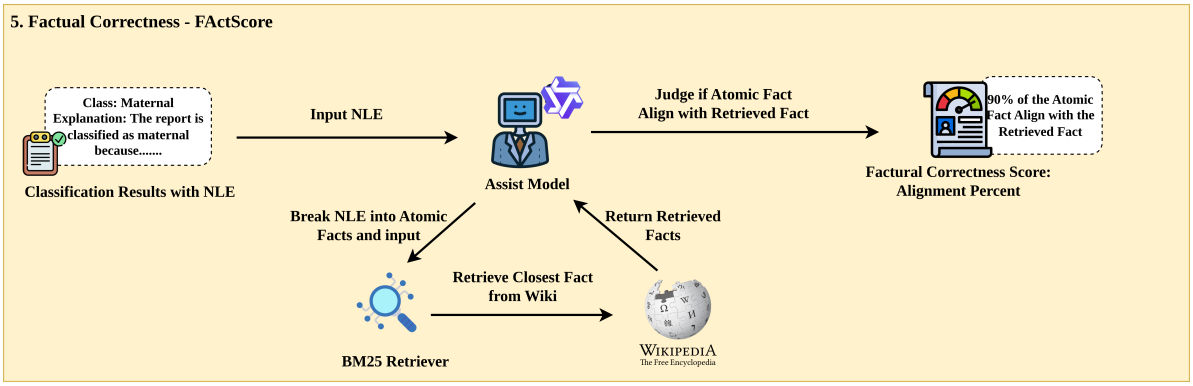

Figure 11: FActScore Evaluation Procedure

6 Experiment Result and Discussion

6.1 Result for Fine-tuned Classifier and Post-hoc Explainability Method

6.1.1 Classification Result

For each type of model introduced before, we have fine-tuned 5 of them with different seed initialized. The final accuracy is then calculated by the average of these accuracy which is shown in the following table 5:

Table 5: Fine-tuning Metrics for Each Model - Classification Head

Model Name	Accuracy	Precision	Recall	F1 Score	Loss
RoBERTa	0.7048	**0.6932**	0.6454	0.6490	1.1920
DeBERTa	0.6968	0.6735	**0.6528**	**0.6566**	1.3512
RoBERTa Large	0.6935	0.6478	0.6478	0.6349	1.4910
GPT2	0.6306	0.6262	0.5740	0.5792	1.2741
BioRoBERTa	0.7048	0.6855	0.6398	0.6505	1.2329
RoBERTa LoRA	**0.7096**	0.6782	0.6262	0.6279	**0.8646**

As demonstrated in table 5 above, the parameter-efficient RoBERTa LoRA variant outperforms every other model on accuracy, which is around 71%. DeBERTa, on the other hand, is slightly ahead of other models in terms of the F1 score, which is around 0.657.

Meanwhile, all fine-tuned transformer encoder models are more tend to be conservative, where their precision is higher than recall. It potentially identifies all models to avoid false positives at the cost of missing some true cases. Specifically, DeBERTa narrows the gap between precision and recall the most, which aligns with its leading F1 score observation stated before. In contrast, GPT2 has demonstrated the most dramatic drop in recall, which is down to 0.574. It suggests that the autoregressive decoder models struggle to capture minority classes in patient safety report texts.

In term of model size, scaling RoBERTa from base to large hurts performance in every metric and inflates loss. Considering the small dataset sample size, the RoBERTa Large, which contains 355 million parameters, is prone to overfit within the same epoch budget.

BioMed RoBERTa, which is the domain-specific version of RoBERTa pre-trained on PubMed and MIMIC, has shown similar performance with RoBERTa base on accuracy but slightly better on F1 score. The exposure to the biomedical dataset has improved RoBERTa's minority-class recall without sacrificing the precision.

LoRA has provided the most significant boost for RoBERTa, the RoBERTa LoRA with fewer parameters updated delivers the top accuracy and the lowest loss; thus, aligns with the expectation on LoRA's performance on small sample sizes. The low-rank adapter regularizes the model and prevents catastrophic forgetting, which is potentially caused by fine-tuning.

For transformer decoder-only model GPT2 with a classification head, it has the worst performance in the board. The lack of bidirectional context and the demand for an artificial classification token have made it the least suitable for this classification task.

6.1.2 Evaluation of Explainability Techniques Result

All explainability tools are evaluated by the Confidence, Faithfulness, Data Consistency, and Rationale Consistency introduced before. The following plots in Figure 12 demonstrate the evaluation scores separated by evaluation methods. Detailed original result data can be found in Appendix table 910 11 12. Overall, there is no single tool has dominated every metric. For example, the InputXGradient with L2 norm aggregation

as consistently demonstrated decent performance on the Faithfulness metric, LIME has overall the best performance in terms of Confidence Indication, and there is no top explainability tool for Data and Rationale Consistency.

Confidence Indication For Confidence Indication, the simplification-based technique LIME has generally performed the best among all methods, demonstrating top performance on all models' scores. Most methods have obtained fluctuating scores among various models with relatively poorer performance. The potential cause behind LIME's good performance is LIME's design goal closely aligns with how the confidence indication metric is calculated. The confidence indication metric asks if an explanation can predict the evaluated classification model's own softmax probability. It collapses each saliency map into a single scalar, then it fits a logistic regression and measures the MAE. LIME samples thousands of binary token masks and queries the classification model for the same softmax probability with confidence indication. LIME then fits a weighted linear regression and reports its coefficients as attributions. These coefficients have the same numerical scale as the probability that the confidence indication tries to recover. In contrast, the gradient-based techniques are derivatives of the unnormalized logit, whose numeric range has no direct correspondence to the probability domains. The blank-token masking perturbations that LIME utilizes mirror the masking used in the confidence indication's distance function. Hence, LIME would obtain the best performance among all techniques.

Faithfulness For faithfulness, all L2-norm aggregated InputXGradient explainability methods demonstrate consistently decent performance across all models. Multiplying the gradient with the token embedding unsaturates small probabilities by amplifying small logits; hence, truly critical tokens consistently receive large saliency scores. Taking the L2-norm preserves total magnitude while discarding sign. It prevents the cancellations between positive and negative elements. Since the faithfulness diagnostic only cares how much accuracy drops when the top-ranked tokens are moved, tokens with the largest L2-norm are exactly the tokens which reduce the performance the most. Thus, InputXGradient with L2-norm yields consistently better (lower) faithfulness scores. On the other hand, all average aggregated gradient-based explainability methods are much worse. Such observation reflects that taking the average of the embedding's saliency score would cause losing capture of magnitude (positive-negative cancellation); thus, resulting in non-faithful saliency scores. LIME has highly variant performance across models. It has low to medium performance among shallow models, with GPT2's score as its worst performance.

Data Consistency Among all explainability techniques, Occlusion performs relatively the best with respect to data consistency. It achieves excellent on RoBERTa base, RoBERTa base with LoRA, and BioMed RoBERTa. But it has also obtained the worst performance on GPT2 and RoBERTa-Large. This observation potentially reflects that Occlusion's single-token deletion works when the model relies on a repeatable set of lexical anchor words, which are distinctive words strongly identifying the label such as 'specimen' or 'wrong-dose'. RoBERTa base and RoBERTa base with LoRA, and BioMed RoBERTa tend to hinge on anchor words due to their relatively small scale and fewer attention parameters; hence, any two similar data piece would produce very close occlusion maps. On the other hand, for larger-scale models like DeBERTa and RoBERTa-large, they spread evidence over many heads, which reduces their dependence on anchor words; thus, resulting in less data consistency for Occlusion in such models. For the other techniques, LIME has achieved the best on DeBERTa but catastrophically negative on GPT2. Gradient-based L2-norm techniques demonstrate small positives on DeBERTa and GPT2, mild negatives for other models. As discussed before, L2-norm aggregation keeps only the size of each token's gradient vector and ignores its direction. Size is reasonably stable for similar input tokens; therefore, their dataset consistency correlation is usually positive but stays modest. But for gradient-based techniques with mean aggregation, we can observe a general underperformance among all models, potentially because the mean aggregation collapses all dimensions with signs, even the smallest directional change would dramatically change the saliency score map, and even near-duplicate input tokens would produce high-volatility saliency maps; thus, leaning to poor Spearman's correlation.

Rationale Consistency All explainability techniques have demonstrated inconsistent performance in terms of rationale consistency. First, LIME has demonstrated wildly inconsistent performance across all models. It is slightly positive on RoBERTa base but strongly negative on DeBERTa, GPT2, and RoBERTa LoRA. Gradient-based techniques with L2-norm aggregation show slightly positive results on DeBERTa and GPT2 but bad results on BioMed RoBERTa and RoBERTa Large. Among all techniques, the mean aggregated gradient-based techniques still obtain the worst performance.

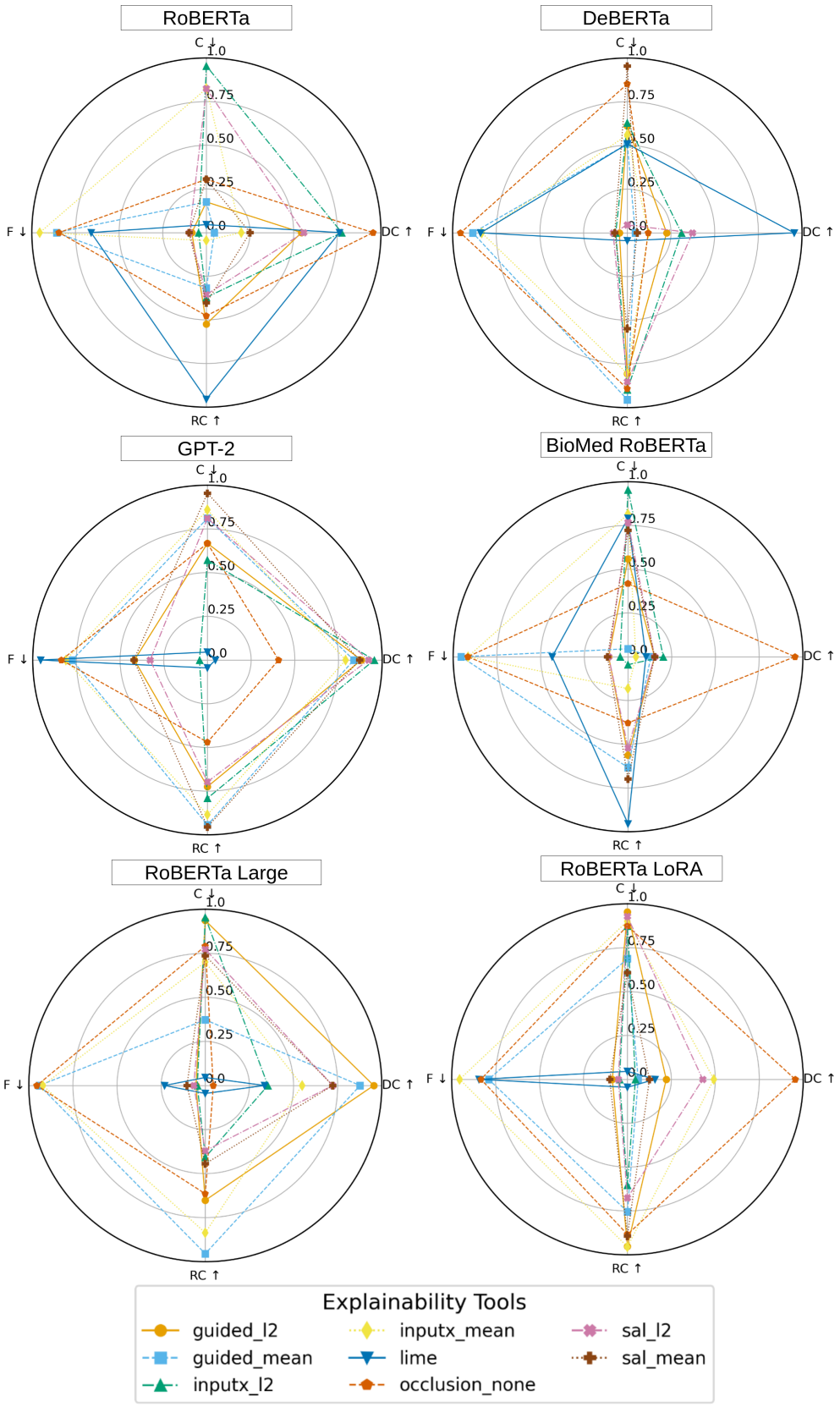

Figure 12: Normalized Evaluation Results of Explainability Tools (For the labels on axis, C - Confidence Indication, F - Faithfulness, DC - Data Consistency, RC - Rationale Consistency. Arrow points up - The higher the better. Arrow points down - The lower the better)

6.2 Experiment Result for Generative Model Method

6.2.1 Classification Result

Table 6: Classification performance of generative models (**EP** = empty predictions; lower is better for EP and generation time). **ZS** = zero-shot; **SC** = self-consistency; **RAG** = retrieval-augmented generation; **DS** = DeepSeek-distilled.

Model	Prompt	Acc.	Prec.	Recall	F1	EP	Gen. Time
Qwen-14B	ZS	0.6935	0.5884	0.5685	0.5719	0	5'22"
	SC	0.6774	**0.6879**	0.6350	0.6401	0	54'10"
	RAG	0.6774	0.6339	0.5942	0.6040	0	5'23"
Qwen-32B	ZS	0.6452	0.6144	0.6040	0.5867	0	9'45"
	SC	0.6452	0.6016	0.6017	0.5842	0	108'43"
	RAG	0.6935	0.6226	0.6173	0.6141	0	10'28"
QwQ-32B	ZS	0.6935	0.6895	0.6667	0.6635	0	47'30"
	SC	0.7016	0.6717	**0.6704**	**0.6610**	0	501'06"
	RAG	**0.7097**	0.5824	0.5931	0.5856	0	47'37"
DS-Qwen-14B	ZS	0.5081	0.5519	0.4823	0.4795	1	5'58"
	SC	0.4274	0.6171	0.3575	0.3523	9	288'28"
	RAG	0.6935	0.5781	0.5532	0.5527	0	6'30"
DS-Qwen-32B	ZS	**0.7097**	0.6040	0.5886	0.5844	0	10'35"
	SC	**0.7097**	0.6664	0.6665	0.6587	0	260'10"
	RAG	0.7016	0.5590	0.5624	0.5564	0	32'45"
Llama-8B	ZS	0.6129	0.6306	0.6769	0.6224	0	3'29"
	SC	0.5887	0.6009	0.6454	0.5987	0	37'06"
	RAG	0.6290	0.5906	0.6093	0.5848	0	3'55"
DS-Llama-8B	ZS	0.5242	0.4992	0.5845	0.4468	0	32'04"
	SC	0.6048	0.5682	0.5957	0.5764	0	145'02"
	RAG	0.2742	0.2583	0.4229	0.1933	0	26'32"
PMC-Llama	N/A	—	—	—	—	all	60'34"

The classification result is demonstrated in the table 6. All tested models have demonstrated similar performance except for distilled Qwen14B, Deepseek distilled Llama8B, and PMC Llama, which generally have the worst performance with empty predictions. Especially for PMC Llama, even though it is instruction fine-tuned with medical question answering tasks and the provided chat template is applied during generation, it still failed to finish the classification task and was not able to generate the required content; thus, reflecting its lack of comprehensiveness during instruction fine-tuning (Thus, omit PMC Llama in all following evlautions).

Reasoning models distilled Qwen32B and QwQ, have generally demonstrated better performance than non-reasoning Qwen32B, suggesting the effectiveness of reasoning ability given by distillation and reinforcement learning (RL) processes. On the other hand, QwQ is suffering from a long thinking time, compared with the other 32B scaled models due to its unique math and coding fine-tuning RL process[7].

Comparing the instruction-learning techniques, larger scale models with self-consistency have achieved top classification performance in all accuracy, precision, recall, and f1-score metrics. It identifies the effectiveness of self-consistency in boosting the generative LLMs' classifying ability. However, it also takes longer generation time which is approximately N (sample number we take vote from) times longer than zero-shot generation. Also, self-consistency decreases the performance of smaller models which fail to generate the correct output format. We can observe this from Deepseek distilled Qwen14B which generates a single unformatted output by itself, but generated 9 unformatted outputs when implementing self-consistency which aligns with the finding from Wang et al [19]. Meanwhile, RAG generally improved performance over the

zero-shot baseline for most models while maintaining the similar generation times, demonstrating its effectiveness in leveraging relevant examples from the training data.

Comparing to the baseline results reported by Chen et al., the LLMs used in this study have shown competitive performance. The baseline's top performance classifier using static text representation (MLR with TF-IDF) has achieved an accuracy of 66.7% and the F1-score of 0.631 [3]. Several models in this study, including Qwen14B, QwQ (with zero-shot and self-consistency), Deepseek Distilled Qwen32B (also with zero-shot and self-consistency), and various RAG configuration models, outperformed this static baseline. However, the baseline study also reported a higher benchmark using contextual embeddings (SVM with RoBERTa-base model), which has achieved an accuracy of 75.4% and an F1-score of 0.753[3]. The LLMs tested in this project, while also demonstrating relatively strong performance and surpassing the static baseline, did not reach this higher contextual baseline. This suggests that even though generative LLMs offer advantages in terms of generating NLEs, further optimization or domain-specific model choices might be needed to match the peak classification accuracy achieved by specialized classifier models.

6.2.2 Evaluation Result

Table 7: Faithfulness Metric: Number of Prediction Flips after Rationale Removal (lower is better). **ZS** = zero-shot; **SC** = self-consistency; **RAG** = retrieval-augmented generation ; **DS** = Deepseek Distilled

Model	Classification Mode	# Changed Predictions
Qwen14B	ZS	11
	SC	11
	RAG	10
Qwen32B	ZS	12
	SC	15
	RAG	14
QwQ	ZS	12
	SC	**5**
	RAG	18
DS-Qwen14B	ZS	35
	SC	53
	RAG	22
DS-Qwen32B	ZS	18
	SC	23
	RAG	25
Llama8B	ZS	23
	SC	19
	RAG	25
DS-Llama8B	ZS	66
	SC	18
	RAG	58

Faithfulness From the result table 7, we can observe that overall, the base Qwen models are the most faithful. Qwen14B had 10 - 11 prediction changed across all classification modes. Qwen32B is closely as good but the self-consistency slightly hurts versus zero-shot. QwQ has achieved the lowest number of changed predictions with self-consistency, highlighting its robust reasoning path and decent faithfulness. This result aligns with the expectation as Qwen series because most Qwen checkpoints already carry more CoT data, such as math proofs, step-by-step analyses, Q&A forums, and large curated reasoning sets, than Llama3.1 series. Especially for QwQ, which utilized reinforcement learning rather than pure Supervised Fine-tuning (SFT), was directly punished for answers that contradict an external verifier by its policy during fine-tuning; hence, it learns to keep the latent decision logic tightly coupled to its prediction label and resulting in decent

faithfulness.

Also from the table 7, Llama8B has demonstrated less faithful performance, which has more prediction changed, than Qwen family. This is potentially caused by Llama8B smaller parameter scales, where models with higher parameter scales have more hidden states to keep intermediate reasoning in consideration. Also, Qwen2.5 series has adopted Group-Query Attention and QKV bias, giving them more expressive power per parameter anda longer usable context window. Fewer heads and shorter context of Llama8B potentially increase the need to truncate thoughts; thus, deleting the visible rationale disturbs the hidden computation more.

Deepseek distilled models generally have more prediction changes than the original version of models, which implies the distilled models are less faithful than the original models. This is potentially caused by the distillation process, where the student model has been forced to think in the teacher model's reasoning path and then make a prediction. Still, the taught reasoning path is likely memorized by the student models rather than producing the student model's own reasoning path. This finding aligns with the finding from Wang et al. where the simple supervised fine-tuning distillation, which is the distillation technique adopted for distilled Qwens, would produce an unfaithful student model [38, 31].

Classification technique wise, self-consistency can generally both positively and negatively impact distilled models' faithfulness. As shown by the result of distilled Qwen14B, self-consistency can highlight less faithful models by amplifying the testing result. On the other hand, distilled Llama8B has reduced its changed prediction from 66 for zero-shot to 18 for self-consistency. This observation has further shown the unstable internal reasoning path for distilled models when taking the majority voting. In contrast, both original Llama and Qwen models tend to be more faithful when majority voting is applied; reflecting their more robust internal reasoning path. RAG, meanwhile, has overall slight impact on models faithfulness.

Table 8: Factual Correctness—BioBERT entailment results (**Contradiction Count** ↓, **FActScore** ↑). **ZS** = zero-shot; **SC** = self-consistency; **RAG** = retrieval-augmented generation; **DS** = DeepSeek Distilled.

Model	**Prompting Strategy**	**Contradictions**	**FActScore**
Qwen14B	ZS	85	33.88%
	SC	**81**	33.96%
	RAG	84	32.26%
Qwen32B	ZS	89	33.53%
	SC	88	**34.90**%
	RAG	82	31.13%
QwQ	ZS	96	31.97%
	SC	101	33.66%
	RAG	91	31.60%
DS-Qwen14B	ZS	97	32.35%
	SC	93	31.66%
	RAG	91	32.40%
DS-Qwen32B	ZS	97	33.53%
	SC	96	33.69%
	RAG	87	32.33%
Llama-8B	ZS	**81**	32.65%
	SC	85	33.69%
	RAG	89	33.96%
DS-Llama-8B	ZS	87	31.66%
	SC	90	31.40%
	RAG	94	32.40%

Factual Correctness - BioBERT Entailment As the table 8 demonstrates, the contradiction count is the number of contradiction predictions from fine-tuned BioBERT to the generative LLM's explanation. The total prediction number is 124. We can observe that all models tested have demonstrated a high contradiction count, indicating the significant existence of false information within the explanation.

This observation can be further verified by the FActScore's result. Besides the contradiction count is the FActScore for each model, none of the scores have exceeded 34% identifying one-third of the atomic facts are not supported by the facts stored in Wikipedia. As a baseline, ChatGPT has obtained 58% FActScore in the work proposed by Min et al[36]. It further suggests all models tested potentially failed to provide true information resulting in poor factual correctness.

These results are potentially caused by the lack of domain-specific knowledge for tested models. As Qwen series and its variants are trained on board, general text corpus, these datasets may lack deep and nuanced understanding of specific medical terminology, hospital procedures, or the particular context regarding PSE reports. Especially our PSE reports often contain a significant number of jargons, abbreviations, and descriptions of situations which require specialized knowledge to interpret and explain accurately.

But the team would also like to argue that this observation needs to be further validated with an ablation study on BioBERT Entailment and FActScore. The BioBERT entailment methods adopted the e-SNLI dataset in the original method rather than the medical domain inference dataset; thus, the fine-tuned BioBERT model potentially lacks medical domain-specific inference knowledge. For the FActScore, due to the hardware and confidential constraint, Qwen14B is used as the assistant generative LLM. Compared with the full-size ChatGPT adopted in the original FActScore paper, Qwen14B potentially lacks of power to perform proper judgement on the provided atomic facts and Wikipedia contents[36]. Thus, potentially leading to incorrect judgment.

7 Conclusion and Future Work

This project comprehensively investigated the application of both fine-tuned transformer models and generative Large Language Models (LLMs) for the classification of Patient Safety Event (PSE) reports and the generation and evaluation of accompanying explanations.

The study first demonstrated that fine-tuned classifiers, such as RoBERTa and DeBERTa variants, can achieve competent classification performance on PSE reports, with techniques like LoRA showing benefits for classification accuracy. The evaluation of post-hoc explainability methods (Saliency, InputXGradient, LIME, Occlusion) applied to these classifiers revealed that no single technique proved universally superior across all evaluation criteria, including faithfulness, confidence indication, data and rationale consistency. However, we have discovered some patterns such as gradient-based methods with L2-norm aggregation showing strength in faithfulness, simplification-based methods like LIME performed well in confidence indication, and Occlusion has good performance in data consistency.

Subsequently, the research found that tested open-source generative LLMs, using instruction-learning settings (zero-shot, RAG, self-consistency), showed classification capabilities comparable to established ML baselines. A critical finding related to natural language explanation (NLEs): non-distilled models, especially for those fine-tuned with reinforcement learning (QwQ), have produced explanations that more faithfully reflected their internal decision processes compared with models created via simple supervised fine-tuning distillation. Also, assessments for factual correctness with BioBERT entailment and FActScore revealed major concerns, with high contradiction rates and low scores demonstrating potential inaccuracies or hallucinations with the explanation generated. On the other hand, we also observed methodological limitations, such as domain mismatch in the BioBERT entailment model's dataset and shallow evidence retrieval in FActScore, that likely introduced noise and reduced the accuracy of these measurements

For both methods, future works should focus on manual evaluation of both the classification labels and

the generated explanations (Post-Hoc and NLE) by clinical domain experts, which is necessary to ensure the trustworthiness of these models in real-world PSE contexts to further validate the evaluation results. Also, employing larger and better fine-tuned domain-specific generative models for the entailment model or FActScore assistant can potentially. Furthermore, if possible, the team would work on collecting a dataset with the PSE reports, labels, and a golden explanation in natural language so that we can investigate the generative model's classification and explanation performance with supervised instruction fine-tuning.

References

[1] S. Alder et al. What is incident reporting in healthcare? https://www.hipaajournal.com/what-is-incident-reporting-in-healthcare/, 2024. Accessed: April 14, 2025.

[2] L. Kohn et al. To err is human: Building a safer health system. https://pubmed.ncbi.nlm.nih.gov/25077248/, 2000. Accessed: April 14, 2025.

[3] H. Chen et al. A machine learning approach with human-ai collaboration for automated classification of patient safety event reports: Algorithm development and validation study. https://humanfactors.jmir.org/2024/1/e53378/, 2024. Accessed: April 14, 2025.

[4] P. Atanasova et al. A diagnostic study of explainability techniques for text classification. https://arxiv.org/abs/2009.13295, 2020. Accessed: April 14, 2025.

[5] M. Yanagawa et al. Seeing is not always believing: Discrepancies in saliency maps. https://pmc.ncbi.nlm.nih.gov/articles/PMC10831517/, 2023. Accessed: April 14, 2025.

[6] M. Ghassemi et al. The false hope of current approaches to explainable artificial intelligence in health care. https://www.thelancet.com/journals/landig/article/PIIS2589-7500(21)00208-9/fulltext, 2021. Accessed: April 14, 2025.

[7] Qwen Team. Qwq-32b: Embracing the power of reinforcement learning. https://qwenlm.github.io/blog/qwq-32b/, 2025. Accessed: April 14, 2025.

[8] P. Qiu et al. Exploring the limits of transfer learning with a unified text-to-text transformer. https://arxiv.org/abs/2503.04691, 2025. Accessed: April 14, 2025.

[9] P. Atanasova et al. Faithfulness tests for natural language explanations. https://arxiv.org/abs/2305.18029, 2023. Accessed: April 14, 2025.

[10] A. Madsen et al. Are self-explanations from large language models faithful? https://arxiv.org/abs/2401.07927, 2024. Accessed: April 14, 2025.

[11] Y. Liu et al. Roberta: A robustly optimized bert pretraining approach. https://arxiv.org/abs/1907.11692, 2019. Accessed: April 14, 2025.

[12] S. Gururangan et al. Don't stop pretraining: Adapt language models to domains and tasks. https://arxiv.org/abs/2004.10964, 2020. Accessed: April 14, 2025.

[13] P. He et al. Deberta: Decoding-enhanced bert with disentangled attention. https://arxiv.org/abs/2006.03654, 2020. Accessed: April 14, 2025.

[14] A. Radford et al. Language models are unsupervised multitask learners. https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf, 2019. Accessed: April 14, 2025.

[15] A. Yang et al. Qwen2.5 technical report. https://arxiv.org/abs/2412.15115, 2025. Accessed: April 14, 2025.

[16] A. Grattafiori et al. The llama 3 herd of models. https://arxiv.org/abs/2407.21783, 2024. Accessed: April 14, 2025.

[17] C. Wu et al. Pmc-llama: Towards building open-source language models for medicine. https://arxiv.org/abs/2304.14454, 2023. Accessed: April 14, 2025.

[18] E. Hu et al. Lora: Low-rank adaptation of large language models. https://arxiv.org/abs/2106.09685, 2021. Accessed: April 14, 2025.

[19] X. Wang et al. Self-consistency improves chain of thought reasoning in language models. https://arxiv.org/abs/2203.11171, 2023. Accessed: April 14, 2025.

[20] P. Lewis et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. https://arxiv.org/abs/2005.11401, 2020. Accessed: April 14, 2025.

[21] K. Simonyan et al. Deep inside convolutional networks: Visualising image classification models and saliency maps. https://arxiv.org/abs/1312.6034, 2014. Accessed: April 14, 2025.

[22] A. Shrikumar et al. Not just a black box: Learning important features through propagating activation differences. https://arxiv.org/abs/1605.01713, 2017. Accessed: April 14, 2025.

[23] J. Springenberg et al. Striving for simplicity: The all convolutional ne. https://arxiv.org/abs/1412.6806, 2014. Accessed: April 14, 2025.

[24] M. T. Ribeiro et al. "why should i trust you?": Explaining the predictions of any classifier. https://arxiv.org/abs/1602.04938, 2016. Accessed: April 14, 2025.

[25] M. D. Zeiler et al. Visualizing and understanding convolutional networks. https://arxiv.org/abs/1311.2901, 2013. Accessed: April 14, 2025.

[26] M. Valentino et al. Do natural language explanations represent valid logical arguments? verifying entailment in explainable nli gold standards. https://arxiv.org/abs/2105.01974, 2021. Accessed: April 14, 2025.

[27] P. null et al. Scott: Self-consistent chain-of-thought distillation. https://arxiv.org/abs/2305.01879, 2023. Accessed: April 14, 2025.

[28] L. Ginsburg et al. Development of a measure of patient safety event learning responses. https://pmc.ncbi.nlm.nih.gov/articles/PMC2796318, 2009. Accessed: April 14, 2025.

[29] S. Kim et al. Propile: Probing privacy leakage in large language models. https://arxiv.org/abs/2307.01881, 2023. Accessed: April 14, 2025.

[30] N. Alkhaldi et al. Assessing the cost of implementing ai in healthcare. https://itrexgroup.com/blog/assessing-the-costs-of-implementing-ai-in-healthcare/, 2024. Accessed: April 14, 2025.

[31] Deepseek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. https://arxiv.org/abs/2402.12749, 2025. Accessed: April 14, 2025.

[32] J. Lee et al. Biobert: a pre-trained biomedical language representation model for biomedical text mining. https://arxiv.org/abs/1901.08746, 2019. Accessed: April 14, 2025.

[33] C. Agarwal et al. Faithfulness vs. plausibility: On the (un)reliability of explanations from large language models. https://arxiv.org/abs/2402.04614, 2024. Accessed: April 14, 2025.

[34] C. Raffel et al. Exploring the limits of transfer learning with a unified text-to-text transformer. https://arxiv.org/abs/1910.10683, 2019. Accessed: April 14, 2025.

[35] M. Jeong et al. Olaph: Improving factuality in biomedical long-form question answering. https://arxiv.org/abs/2305.18029, 2025. Accessed: April 14, 2025.

[36] S. Min et al. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. https://arxiv.org/abs/2305.14251, 2023. Accessed: April 14, 2025.

[37] O. Camburu et al. e-snli: Natural language inference with natural language explanations. https://arxiv.org/abs/1812.01193, 2018. Accessed: April 14, 2025.

[38] P. Wang et al. Scott: Self-consistent chain-of-thought distillation. https://arxiv.org/abs/2305.01879, 2023. Accessed: April 14, 2025.

A Appendix: Post-Hoc Explainability Techniques Evaluation Data

Table 9: Confidence Indication Metrics for Every Model and Saliency Method (lower is better)

Model	Saliency Method	MAE	MaxErr
RoBERTa	guided_l2	0.225 ± 0.019	0.341 ± 0.031
	guided_mean	0.225 ± 0.022	0.354 ± 0.016
	inputx_l2	0.231 ± 0.021	0.355 ± 0.047
	inputx_mean	0.230 ± 0.020	0.366 ± 0.035
	lime	0.224 ± 0.018	0.376 ± 0.058
	occlusion_none	0.226 ± 0.024	0.390 ± 0.092
	sal_l2	0.230 ± 0.022	0.341 ± 0.037
	sal_mean	0.226 ± 0.021	0.346 ± 0.021
DeBERTa	guided_l2	0.214 ± 0.017	0.358 ± 0.076
	guided_mean	0.208 ± 0.030	0.314 ± 0.045
	inputx_l2	0.224 ± 0.024	0.323 ± 0.042
	inputx_mean	0.218 ± 0.035	0.345 ± 0.070
	lime	0.230 ± 0.018	0.374 ± 0.037
	occlusion_none	0.220 ± 0.022	0.335 ± 0.049
	sal_l2	0.223 ± 0.026	0.337 ± 0.062
	sal_mean	0.219 ± 0.026	0.332 ± 0.067
RoBERTa Large	guided_l2	0.247 ± 0.026	0.401 ± 0.110
	guided_mean	0.236 ± 0.021	0.374 ± 0.073
	inputx_l2	0.254 ± 0.018	0.366 ± 0.073
	inputx_mean	0.250 ± 0.026	0.371 ± 0.040
	lime	0.245 ± 0.026	0.398 ± 0.087
	occlusion_none	0.261 ± 0.017	0.388 ± 0.029
	sal_l2	0.252 ± 0.022	0.396 ± 0.074
	sal_mean	0.250 ± 0.024	0.397 ± 0.060
GPT2	guided_l2	0.213 ± 0.024	0.317 ± 0.028
	guided_mean	0.216 ± 0.022	0.315 ± 0.027
	inputx_l2	0.224 ± 0.018	0.323 ± 0.031
	inputx_mean	0.221 ± 0.025	0.319 ± 0.030
	lime	0.210 ± 0.023	0.317 ± 0.030
	occlusion_none	0.213 ± 0.024	0.330 ± 0.056
	sal_l2	0.222 ± 0.026	0.331 ± 0.053
	sal_mean	0.219 ± 0.023	0.322 ± 0.067
BioRoBERTa	guided_l2	0.230 ± 0.012	0.361 ± 0.041
	guided_mean	0.208 ± 0.021	0.401 ± 0.027
	inputx_l2	0.247 ± 0.019	0.363 ± 0.034
	inputx_mean	0.241 ± 0.030	0.367 ± 0.036
	lime	0.240 ± 0.024	0.373 ± 0.031
	occlusion_none	0.224 ± 0.028	0.341 ± 0.023
	sal_l2	0.239 ± 0.016	0.378 ± 0.030
	sal_mean	0.237 ± 0.035	0.382 ± 0.045
RoBERTa LoRA	guided_l2	0.177 ± 0.015	0.381 ± 0.079
	guided_mean	0.167 ± 0.010	0.359 ± 0.021
	inputx_l2	0.174 ± 0.017	0.381 ± 0.054
	inputx_mean	0.175 ± 0.010	0.358 ± 0.012
	lime	0.143 ± 0.016	0.344 ± 0.053
	occlusion_none	0.174 ± 0.009	0.358 ± 0.022
	sal_l2	0.176 ± 0.015	0.399 ± 0.045
	sal_mean	0.164 ± 0.007	0.340 ± 0.016

Table 10: Faithfulness Metrics for Every Model and Saliency Method (lower is better)

Model	**Saliency Method**	**Mean Value**
RoBERTa	guided_l2	26.420 ± 3.480
	guided_mean	34.230 ± 3.750
	inputx_l2	26.040 ± 4.040
	inputx_mean	35.200 ± 3.610
	lime	32.240 ± 5.500
	occlusion_none	34.090 ± 2.630
	sal_l2	26.550 ± 2.720
	sal_mean	26.590 ± 2.890
DeBERTa	guided_l2	23.040 ± 2.450
	guided_mean	28.840 ± 3.210
	inputx_l2	23.070 ± 3.770
	inputx_mean	31.050 ± 4.810
	lime	28.160 ± 2.950
	occlusion_none	30.760 ± 2.320
	sal_l2	23.430 ± 2.770
	sal_mean	23.480 ± 2.890
RoBERTa Large	guided_l2	30.340 ± 3.520
	guided_mean	38.100 ± 3.580
	inputx_l2	30.410 ± 3.210
	inputx_mean	40.100 ± 4.400
	lime	32.920 ± 4.920
	occlusion_none	35.420 ± 2.430
	sal_l2	30.680 ± 2.760
	sal_mean	30.830 ± 3.430
GPT-2	guided_l2	21.990 ± 2.600
	guided_mean	24.070 ± 2.720
	inputx_l2	22.340 ± 2.450
	inputx_mean	25.420 ± 2.740
	lime	24.500 ± 3.050
	occlusion_none	24.070 ± 3.090
	sal_l2	22.560 ± 2.950
	sal_mean	22.610 ± 3.120
BioRoBERTa	guided_l2	28.200 ± 2.210
	guided_mean	36.920 ± 2.630
	inputx_l2	28.150 ± 2.660
	inputx_mean	37.900 ± 2.670
	lime	31.330 ± 2.420
	occlusion_none	34.020 ± 2.610
	sal_l2	28.390 ± 1.440
	sal_mean	28.510 ± 2.010
RoBERTa LoRA	guided_l2	18.360 ± 0.980
	guided_mean	24.810 ± 0.670
	inputx_l2	17.990 ± 1.010
	inputx_mean	26.320 ± 1.300
	lime	25.310 ± 1.870
	occlusion_none	25.210 ± 0.720
	sal_l2	18.050 ± 1.430
	sal_mean	18.500 ± 1.690

Table 11: Data-Consistency Metrics for Every Model and Saliency Method (higher is better)

Model	Saliency Method	Spearman	p-Value
RoBERTa	guided_l2	–0.012	0.200
	guided_mean	–0.051	0.200
	inputx_l2	0.006	0.200
	inputx_mean	–0.039	0.300
	lime	0.005	0.000
	occlusion_none	0.020	0.200
	sal_l2	–0.011	0.200
	sal_mean	–0.035	0.200
DeBERTa	guided_l2	–0.041	0.100
	guided_mean	–0.044	0.100
	inputx_l2	–0.036	0.100
	inputx_mean	–0.028	0.200
	lime	–0.032	0.100
	occlusion_none	0.009	0.200
	sal_l2	–0.040	0.100
	sal_mean	–0.040	0.100
RoBERTa Large	guided_l2	–0.139	0.100
	guided_mean	–0.122	0.100
	inputx_l2	–0.129	0.100
	inputx_mean	–0.101	0.100
	lime	–0.122	0.100
	occlusion_none	–0.071	0.200
	sal_l2	–0.135	0.100
	sal_mean	–0.138	0.100
GPT-2	guided_l2	–0.045	0.100
	guided_mean	–0.054	0.100
	inputx_l2	–0.048	0.100
	inputx_mean	–0.036	0.200
	lime	–0.050	0.100
	occlusion_none	–0.024	0.200
	sal_l2	–0.050	0.100
	sal_mean	–0.048	0.100
BioRoBERTa	guided_l2	–0.146	0.100
	guided_mean	–0.155	0.100
	inputx_l2	–0.149	0.100
	inputx_mean	–0.145	0.100
	lime	–0.145	0.000
	occlusion_none	–0.070	0.100
	sal_l2	–0.145	0.100
	sal_mean	–0.145	0.000
RoBERTa LoRA	guided_l2	–0.072	0.100
	guided_mean	–0.082	0.000
	inputx_l2	–0.083	0.200
	inputx_mean	–0.055	0.300
	lime	–0.076	0.100
	occlusion_none	–0.026	0.400
	sal_l2	–0.059	0.000
	sal_mean	–0.078	0.000

Table 12: Rationale-Consistency Metrics for Every Model and Saliency Method (higher is better)

Model	Saliency Method	Spearman	p-Value
RoBERTa	guided_l2	–0.010	0.916
	guided_mean	–0.049	0.589
	inputx_l2	–0.039	0.671
	inputx_mean	–0.101	0.263
	lime	0.072	0.430
	occlusion_none	–0.019	0.830
	sal_l2	–0.042	0.646
	sal_mean	–0.033	0.714
DeBERTa	guided_l2	–0.041	0.100
	guided_mean	–0.044	0.100
	inputx_l2	–0.036	0.100
	inputx_mean	–0.028	0.200
	lime	–0.032	0.100
	occlusion_none	0.009	0.200
	sal_l2	–0.040	0.100
	sal_mean	–0.040	0.100
RoBERTa Large	guided_l2	–0.139	0.100
	guided_mean	–0.122	0.100
	inputx_l2	–0.129	0.100
	inputx_mean	–0.101	0.100
	lime	–0.122	0.100
	occlusion_none	–0.071	0.200
	sal_l2	–0.135	0.100
	sal_mean	–0.138	0.100
GPT-2	guided_l2	–0.045	0.100
	guided_mean	–0.054	0.100
	inputx_l2	–0.048	0.100
	inputx_mean	–0.036	0.200
	lime	–0.050	0.100
	occlusion_none	–0.024	0.200
	sal_l2	–0.050	0.100
	sal_mean	–0.048	0.100
BioRoBERTa	guided_l2	–0.146	0.100
	guided_mean	–0.155	0.100
	inputx_l2	–0.149	0.100
	inputx_mean	–0.145	0.100
	lime	–0.145	0.000
	occlusion_none	–0.070	0.100
	sal_l2	–0.074	0.414
	sal_mean	–0.044	0.624
RoBERTa LoRA	guided_l2	0.120	0.183
	guided_mean	0.039	0.667
	inputx_l2	–0.023	0.799
	inputx_mean	0.121	0.182
	lime	–0.251	0.005
	occlusion_none	0.092	0.310
	sal_l2	0.007	0.937
	sal_mean	0.096	0.291