LEARNING TO EXPLAIN: PROTOTYPE-BASED SURROGATE MODELS FOR LLM CLASSIFICATION

Bowen Wei
Department of Computer Science
George Mason University
Fairfax, VA 22030
`bwei2@gmu.edu`

Mehrdad Fazli
Department of Computer Science
George Mason University
Fairfax, VA 22030
`mfazli@gmu.edu`

Ziwei Zhu
Department of Computer Science
George Mason University
Fairfax, VA 22030
`zzhu20@gmu.edu`

ABSTRACT

Large language models (LLMs) have demonstrated impressive performance on natural language tasks, but their decision-making processes remain largely opaque. Existing explanation methods either suffer from limited faithfulness to the model's reasoning or produce explanations that humans find difficult to understand. To address these challenges, we propose **ProtoSurE**, a novel prototype-based surrogate framework that provides faithful and human-understandable explanations for LLMs. ProtoSurE trains an interpretable-by-design surrogate model that aligns with the target LLM while utilizing sentence-level prototypes as human-understandable concepts. Extensive experiments show that ProtoSurE consistently outperforms SOTA explanation methods across diverse LLMs and datasets. Importantly, ProtoSurE demonstrates strong data efficiency, requiring relatively few training examples to achieve good performance, making it practical for real-world applications.

1 Introduction

Large language models (LLMs) have achieved impressive performance across a broad range of natural language tasks. However, their decision-making processes remain largely opaque. This lack of transparency raises serious concerns in high-stakes domains such as healthcare Yu et al. [2018], law Zhong et al. [2020], and finance Arner et al. [2020], where accurate and understandable reasoning is essential. Attempts to directly analyze the internal computations of LLMs are often computationally intensive, methodologically fragile, and rarely yield explanations that generalize well or are accessible to human users.

Post-hoc explanation methods – such as SHAP Lundberg and Lee [2017], Integrated Gradients Sundararajan et al. [2017], Occlusion Zeiler and Fergus [2014], and DeepLIFT Shrikumar et al. [2017] – explain models by assigning attribution scores to individual tokens. However, this explanation paradigm often fails to capture the actual reasoning of models Jacovi and Goldberg [2020] and produces outputs that are difficult for humans to understand Spectra [2021]. An alternative strategy – prompting LLMs to generate self-explanations Madsen et al. [2024] or chain-of-thought reasoning Wei et al. [2022] – can yield fluent justifications, but these are often inadequate for revealing models' true inference processes Lanham et al. [2023]. These highlight two key limitations of existing methods: (1) a lack of faithfulness to the model's actual decision-making, and (2) limited human understandability.

To address these limitations, we propose **ProtoSurE** (**Proto**type-based **Sur**rogate **E**xplanations), a novel framework, as shown in Figure 1, that provides faithful and human-understandable explanations for LLM-based text classification. ProtoSurE trains an interpretable-by-design surrogate model to closely approximate the behavior of the target black-box LLM. The surrogate model's white-box interpretations are then used as faithful explanations of the LLM's predictions. To ensure alignment with the LLM and explanation faithfulness, ProtoSurE employs a knowledge distillation approach Hinton et al. [2015], training the surrogate to replicate the LLM's classification behavior.

Moreover, to enhance human comprehension, we design a prototype-based architecture for the surrogate model. Prototype-based methods have proven highly effective and interpretable, making decisions by comparing inputs to learned prototypes that represent meaningful concepts Chen et al. [2019]. These approaches have demonstrated strong performance across diverse tasks, including recognition Chen et al. [2019], classification Li et al. [2018], out-of-

distribution detection Ming et al. [2019], domain adaptation Tan et al. [2018a], and segmentation Donnelly et al. [2018]. Their intuitive "this looks like that" explanations Li et al. [2018] facilitate understanding of complex decisions by linking inputs to human-understandable patterns. In ProtoSurE, we design a sentence-level prototype-based architecture for the surrogate model, producing explanations that align more naturally with how humans understand and reason about language.

ProtoSurE's explanations excel in two critical dimensions: (1) **faithfulness**, through its rigorous knowledge distillation-based training that ensures the surrogate model accurately reflects the target LLM's behavior; and (2) **human understandability**, via clear and coherent sentence-level prototypes that align with how humans process language.

In summary, our contributions are: (1) We propose **ProtoSurE**, a prototype-based surrogate explanation framework for explaining black-box LLM predictions in text classification. (2) We introduce **sentence-level prototype explanations** that enhance human comprehension significantly beyond existing token-level approaches. (3) We validate ProtoSurE through extensive experiments across diverse text classification benchmarks, demonstrating its effectiveness in delivering faithful and comprehensible explanations.

2 Related Work

Post-hoc Explanation Methods. Post-hoc methods interpret black-box models by revealing input-output relationships. Feature attribution techniques like SHAP Lundberg and Lee [2017], Integrated Gradients Sundararajan et al. [2017], and DeepLIFT Shrikumar et al. [2017] assign attribution scores to individual tokens. However, these token-level methods struggle with faithfulness Jacovi and Goldberg [2020] and human interpretability Spectra [2021]. LLM self-explanations and chain-of-thought reasoning Wei et al. [2022], while promising, produce plausible but unfaithful explanations Madsen et al. [2024], Lanham et al. [2023], limiting their reliability for interpretability.

Prototype-based Neural Networks. Prototype-based methods improve interpretability by comparing inputs with representative examples rather than using abstract feature weights. Originally developed for computer vision Chen et al. [2019], these methods have enabled intuitive "this looks like that" explanations across various applications. Recent adaptations in NLP – such as PoetryNet Hong et al. [2023] and ProtoLens Wei and Zhu [2024] – have demonstrated effectiveness in delivering white-box text classification. However, these works primarily focus on building interpretable-by-design classifiers, typically based on traditional language models such as BERT Devlin et al. [2019]. In contrast, our work pursues a different goal – developing a prototype-based model to explain the prediction of a target LLM in a post-hoc way.

Surrogate Models as Explanations. Surrogate models explain complex models by approximating their behavior with simpler, interpretable ones Ribeiro et al. [2016], Molnar [2019]. LIME pioneered this approach with local linear approximations, while knowledge distillation techniques Hinton et al. [2015], Tan et al. [2018b] transfer complex model knowledge to simpler structures. However, current surrogate methods often face fidelity-interpretability trade-offs Rudin [2019], especially with LLMs. Our work addresses this challenge by introducing a prototype-based surrogate framework specifically designed to balance faithful approximation of LLM predictions with human-interpretable explanations at the sentence level.

3 Method

ProtoSurE learns an interpretable-by-design surrogate model that faithfully explains a target black-box LLM through sentence-level prototype-based explanations. An overview of the model architecture is shown in Figure 1.

3.1 Overall Structure

Problem Formulation. Given a target black-box LLM $\mathcal{M}_{\text{target}}$ and a set of text samples with corresponding predictions from $\mathcal{M}_{\text{target}}$, our goal is to construct an interpretable surrogate model that faithfully explains these predictions. Specifically, the set of text samples is denoted as $\mathcal{X} = \{X_1, X_2, \ldots, X_N\}$. The corresponding predictions by $\mathcal{M}_{\text{target}}$ are denoted as $\hat{\mathcal{Y}} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_N\}$.

To increase alignment between our surrogate model and $\mathcal{M}_{\text{target}}$, we leverage token-level attribution scores $\{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N\}$ obtained from any existing post-hoc explanation methods Chefer et al. [2021] applied to $\mathcal{M}_{\text{target}}$. These scores provide guidance about which tokens are most influential in the $\mathcal{M}_{\text{target}}$'s decision process, helping the surrogate model focus on the same textual elements that drive the LLM's predictions. Experiments in Section 4.4.2 confirm the effectiveness of incorporating these attribution scores.

ProtoSurE

Step 1

Input Text:

"The room was spotless and the bathroom is very clean. The front desk staff went out of their way to help with late check-in. The breakfast buffet offered very decent breads and the coffee was excellent. "

Sentence Splitting:

"The room was spotless and the bathroom is very clean. "

"The front desk staff went out of their way to help with late check-in. "

"The breakfast buffet offered only a few cold pastries, but the coffee was excellent."

Step 2

Encoder

Token Embeddings

Q K V

Matmul

r

SoftMax

Matmul

α c

Token Importance Context Embedding

e: Token Embeddings

Sentence Embedding:

$$\mathbf{h}_i = \sum_j^{l_i} \alpha_{i,j} \mathbf{c}_{i,j}$$

Step 3

Interpretable Classifier

class weight: [0.95, 0.05]

h "The room was spotless … comfortable." cos 0.92

Cleanliness is top

0.87 = 0.95 * 0.92

h "The front-desk staff went out … check-in." 0.88

Service is outstanding

0.7

Breakfast is decent

h "The breakfast buffet offered … was excellent." 0.75

0.45

Positive 2.02

Negative 0.61

Intuitive Explanation

This review comes out **Positive** because each sentence highlights something good about the stay. The first sentence raves about how clean the room and bathroom are, the second sentence praises how helpful the front-desk staff is, and the third sentence notes that the breakfast (and coffee) is really enjoyable. Since all three sentences point to positive aspects —cleanliness, service, and breakfast—the model predicts a positive overall experience.
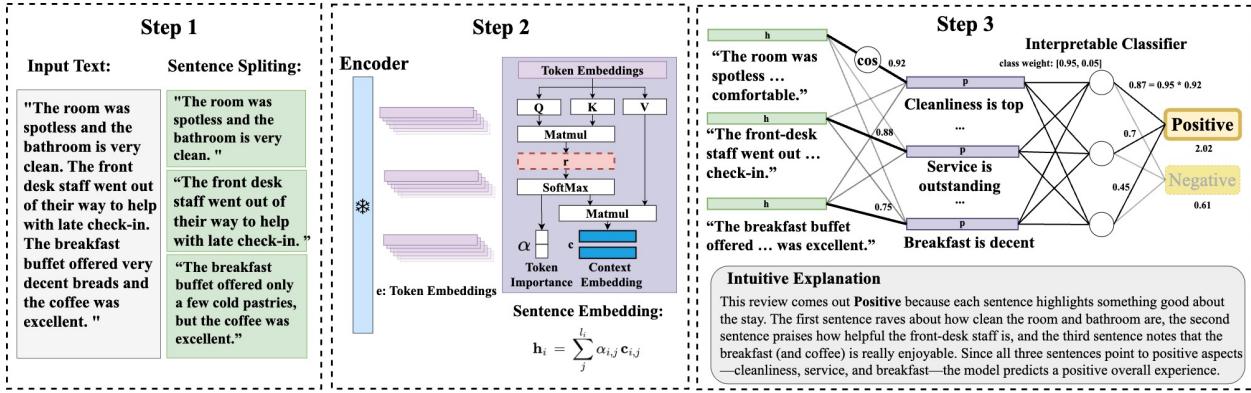
Figure 1: Overview of the ProtoSurE framework. The process consists of three main steps: (1) Sentence splitting that divides input text into semantically meaningful units; (2) An encoder that processes each sentence to generate token embeddings, applying self attention guided by token attribution scores to provide contextualized token embeddings and token attribution weights; and (3) An interpretable classifier that computes cosine similarities between sentence embeddings and learned prototypes, then applies class weights to determine the final prediction. As shown in the explanation box, for a hotel review example, ProtoSurE provides clear rationales by showing which sentences match specific prototypes (Cleanliness, Service, and Breakfast) and how their weighted contributions (0.87, 0.70, and 0.45) lead to the final Positive classification with a score of 2.02 versus 0.61 for Negative.

Model Overview. As illustrated in Figure 1, ProtoSurE processes input text through three main steps to generate explanations for LLM classifications.

In Step 1, ProtoSurE splits the input text into sentences using standard punctuation delimiters (., !, ?). Figure 1 shows a movie review segmented into three distinct sentences.

In Step 2, each sentence is first tokenized and passed through a text encoder to generate token embeddings. These token embeddings are then processed through a self-attention module to determine token importance α and create contextual embeddings $\mathbf{c}$ for tokens. The sentence embedding is computed as a weighted average of the contextual token embeddings, with weights determined by token attribution scores ($h_i = \sum_j \alpha_{i,j} \mathbf{c}_{i,j}$).

In Step 3, the sentence embeddings are compared against a set of trainable prototypes $\mathcal{P} = \{\mathbf{p}_k \in \mathbb{R}^d : k = 1, \ldots, K\}$, where each prototype is represented by an embedding vector, and the hyperparameter K is the number of prototypes specified. Each sentence activates different prototypes based on semantic similarity, with an interpretable classification head (such as a logistic regression model or decision tree) taking these activations as inputs to determine the final prediction.

Explanation Generation. Figure 1 illustrates this process on a hotel review classified as Positive. To simplify the example, only the most important prototype for each sentence is shown (each sentence still has nonzero similarities to other prototypes, but we show the largest contributor in this example). The first sentence ("The room was spotless and the bed was extremely comfortable.") activates the *Cleanliness* prototype (cosine similarity = 0.92) with a learned positive-class weight of 0.95, contributing $0.92 \times 0.95 = 0.87$ to the positive logit; the second sentence ("The front-desk staff went out of their way to help with late check-in.") activates the *Service* prototype (similarity = 0.88) with weight 0.80, contributing 0.70; and the third sentence ("The breakfast buffet offered only a few cold pastries, but the coffee was excellent.") activates the *Breakfast* prototype (similarity = 0.75) with weight 0.60, contributing 0.45. These three main contributions sum to a positive-class score of 2.02 versus a negative-class score of 0.45, yielding the **Positive** prediction. By grounding each sentence in the most influential prototypes, this framework delivers explanations that are both faithful to the model's internal reasoning and easy to understand.

3.2 Attribution-aware Sentence Embedding

A key innovation in ProtoSurE is its ability to create sentence embeddings that capture both semantic meaning and relevance to the classification task.

3

Given an input text X, we first segment it into sentences $\mathbf{s} = [s_1, s_2, \ldots, s_M]$. Each sentence s_i is tokenized into $\mathbf{t}^{(i)} = [t_{i,1}, \ldots, t_{i,\ell_i}]$ and encoded using a pre-trained text encoder $\mathcal{E}$ (e.g., MPNet Song et al. [2020], BGE Chen et al. [2024]), yielding token embeddings:

$$\mathbf{e}_{i,j} = \mathcal{E}(t_{i,j}) \in \mathbb{R}^d. \tag{1}$$

To ensure the learned surrogate model behaves similarly to the target LLM, we aim to incorporate the information about which tokens the LLM relies on for prediction. We can use established post-hoc explanation methods (e.g., attention relevancy maps Chefer et al. [2021], integrated gradients Sundararajan et al. [2017], or SHAP Lundberg and Lee [2017]) to obtain attribution scores $r_{i,j}$ for each token. These methods quantify the contribution of individual tokens to the model's predictions. We normalize the token attribution scores $r_{i,j}$ obtained from the target LLM:

$$\hat{r}_{i,j} = \frac{r_{i,j}}{\sum_{j'=1}^{\ell_i} r_{i,j'} + \epsilon}, \quad \epsilon = 10^{-9}. \tag{2}$$

We then use these normalized attribution scores to guide a self-attention mechanism. With query, key, and value matrices $\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i \in \mathbb{R}^{\ell_i \times d}$ derived from token embeddings, we compute attention as:

$$A_i = \mathrm{softmax}\left(\frac{\mathbf{Q}_i \mathbf{K}_i^{\top}}{\sqrt{d}} + \hat{\mathbf{r}}_i\right), \tag{3}$$

$$\mathbf{c}_i = A_i \mathbf{V}_i. \tag{4}$$

To quantify the importance of each token within a sentence, we compute token-level attribution scores $\alpha_{i,j}$. In self-attention, each token position attends to all token positions, creating an attention matrix A where $A_{i,k,j}$ represents the attention weight from token k to token j in sentence i. We average these attention weights across all source positions to obtain a measure of how much attention each token receives:

$$\alpha_{i,j} = \frac{\exp\left(\frac{1}{\ell_i} \sum_{k=1}^{\ell_i} A_{i,k,j}\right)}{\sum_{j'=1}^{\ell_i} \exp\left(\frac{1}{\ell_i} \sum_{k=1}^{\ell_i} A_{i,k,j'}\right)}. \tag{5}$$

These attribution scores are then used to create attribution-aware sentence embeddings through weighted pooling:

$$\mathbf{h}_i = \sum_{j=1}^{\ell_i} \alpha_{i,j} \cdot \mathbf{c}_{i,j} \in \mathbb{R}^d, \tag{6}$$

This approach ensures that our sentence embeddings not only represent semantic content but also reflect the relative importance of different parts of the text to the target LLM's classification decision.

3.3 Prototype Learning and Classification

To create meaningful and diverse prototypes that represent patterns in the data, we initialize prototype embeddings using a clustering-based approach. We first encode all sentences from the training data into embeddings. We then apply K-means clustering to these embeddings and use the resulting cluster centers as initial prototype embeddings. To provide interpretable explanations, we associate each prototype with its nearest sentence from the training data based on cosine similarity. The prototype embeddings can either be fixed after initialization or further refined through training. We conducted experiments to compare the performance of these two settings in Section 4.4.3.

For classification, each sentence embedding $\mathbf{h}_i$ is compared to all prototype embeddings using cosine similarity: $a_{i,k} = \cos(\mathbf{h}_i, \mathbf{p}_k)$. This produces a similarity vector $\mathbf{a}_i = [a_{i,1}, a_{i,2}, \ldots, a_{i,P}]$ that represents how strongly each sentence activates each prototype. We then apply a linear classifier to this similarity vector to generate prediction logits for each sentence: $\tilde{y}_i = f(\mathbf{a}_i)$. The final prediction for the entire text sample is computed by summing up these sentence-level predictions.

This approach ensures transparency in the classification process by enabling us to trace how each input sentence contributes to the final prediction based on its similarity to specific prototypes. The predictions can then be explained in human-understandable terms by showing how each sentence aligns with meaningful prototypical patterns.

3.4 Training Objective

ProtoSurE is trained with a multi-objective loss balancing fidelity, prototype coverage, and diversity:

$$\mathcal{L} = \text{CrossEntropy}(\hat{y}, \tilde{y}) + \lambda_1 \mathcal{L}_{\text{proto}} + \lambda_2 \mathcal{L}_{\text{diversity}}. \tag{7}$$

Hyperparameters λ_1 and λ_2 are set to 0.1 in our experiments. The prototype utilization loss encourages each prototype to match at least one training sentence:

$$\mathcal{L}_{\text{proto}} = -\frac{1}{P} \sum_{k=1}^{P} \max_{i} \text{sim}(\mathbf{h}_i, \mathbf{p}_k). \tag{8}$$

The diversity loss penalizes overlap between prototypes:

$$\mathcal{L}_{\text{diversity}} = \frac{1}{P(P-1)} \sum_{i=1}^{P} \sum_{\substack{j=1 \\ j \neq i}}^{P} |\mathbf{p}_i^\top \mathbf{p}_j|. \tag{9}$$

4 Experiments

In this section, we evaluate ProtoSurE across multiple dimensions to answer the following research questions: **RQ1:** How faithfully does ProtoSurE explain LLM predictions compared to existing explanation methods? **RQ2:** How does the training data size affect ProtoSurE's performance? **RQ3:** How do core components (encoder, token attribution score, prototype updating) contribute to model effectiveness? **RQ4:** How do key hyperparameters influence performance? **RQ5:** How do ProtoSurE explanations look like in a case study?

4.1 Experimental Setup

Datasets. We evaluate ProtoSurE on four diverse text classification datasets spanning single-label, multi-label, and domain-specific classification tasks: IMDB, Hotel, DBPedia, and Consumer Complaint. Details are provided in Appendix A.

Reproducibility. ProtoSurE was implemented using PyTorch. We train our model with the following hyperparameters: learning rate selected from {1e-2, 2e-2, 2e-3} with AdamW optimizer Loshchilov [2017], batch size of 16, and training for 10 epochs. The prototype number (P) is selected from {10, 20, 40, 100}, and we set $\lambda_1 = 0.1$, and $\lambda_2 = 0.1$ for the loss components. We employ the relevancy map approach proposed by Chefer et al. [2021] as the token attribution scores, which propagates classification-relevant gradients through the attention layers to identify important tokens in the LLM's decision process. The experiments were conducted on NVIDIA A100 80GB GPUs.

Baselines. We compare ProtoSurE against four widely-used post-hoc explanation methods: SHAP Lundberg and Lee [2017], Integrated Gradients (IG) Sundararajan et al. [2017], Occlusion Zeiler and Fergus [2014], and DeepLIFT Shrikumar et al. [2017]. Each method is applied to explain predictions from four target LLMs: Llama-3.1-8B-Instruct, Llama-3.2-3B, Qwen2.5-7B-Instruct-1M, and Mistral-7B-Instruct-v0.2. For fair comparison, we adapt all baseline methods to provide sentence-level attributions by aggregating token-level scores. Detailed descriptions of the baseline methods are provided in Appendix B.1.

Evaluation Metrics. We assess faithfulness using seven metrics: Accuracy (Acc), Comprehensiveness (Comp) DeYoung et al. [2020], Sufficiency (Suff) DeYoung et al. [2020], Decision Flip Fraction (DFF) Serrano and Smith [2019], Decision Flip with Most Important Sentence (DFS) Chrysostomou and Aletras [2021], Deletion Rank Correlation (Del) Alvarez-Melis and Jaakkola [2018], and Insertion Rank Correlation (Ins) Luss et al. [2021]. Details are provided in Appendix B.2.

4.2 Faithfulness Evaluation (RQ1)

In this section, we evaluate how faithfully ProtoSurE explains the predictions of target LLMs compared to SOTA post-hoc explanation methods. We assess faithfulness from two perspectives: (1) alignment with LLM behavior through accuracy on predicting classification results of the LLM, and (2) fidelity to the LLM's underlying reasoning process through established faithfulness metrics.

Our comprehensive faithfulness evaluation in Table 7 (Appendix D) examines performance across four target LLMs and four datasets. First, ProtoSurE demonstrates strong behavioral alignment with target LLMs, achieving an average

Table 1: Average ranking of explanation methods across six faithfulness metrics evaluated on all LLMs (Llama-3.1-8B, Llama-3.2-3B, Qwen2.5-7B, Mistral-7B) and all datasets. Lower rank indicates better performance.

Method	Comp	Suff	DFF	DFS	Del	Ins	Overall
SHAP	3.50	4.03	3.62	3.56	4.00	2.88	3.60
IG	2.19	2.09	2.28	2.81	3.09	2.28	2.46
Occl	3.44	2.78	3.69	2.88	2.03	3.22	3.01
DeepLIFT	4.88	4.41	4.34	4.06	4.12	4.50	4.39
ProtoSurE	**1.00**	**1.69**	**1.06**	**1.69**	**1.75**	**2.12**	**1.55**

Table 2: Impact of different encoders on ProtoSurE's Accuracy (%) for Llama-3.1-8B-Instruct across datasets. Full results across all target LLMs are in the Appendix E. Best results are in **bold**.

Encoder	Hotel	DBPedia	Consumer	Avg Rank
Llama-3.1-8B-Instruct				
SBERT	0.970	0.907	0.859	3.33
BGE	**0.991**	0.906	**0.863**	2.00
GTE	0.984	**0.910**	0.862	**1.67**
E5	0.989	0.908	0.860	2.67
T5	0.978	0.891	0.844	5.00

classification accuracy of 89.58% across all experiments. This high accuracy is essential for ensuring that our explanations reflect the actual decisions made by the target LLM. Table 1 summarizes the average rankings across all faithfulness metrics. ProtoSurE consistently outperforms baseline methods, achieving the best overall average ranking (1.55) compared to IG (2.46), Occlusion (3.01), SHAP (3.60), and DeepLIFT (4.39). This comprehensive advantage demonstrates our approach's superior ability to faithfully capture LLM reasoning processes.

ProtoSurE consistently outperforms baseline methods across all faithfulness metrics. It achieves the highest Comprehensiveness (Comp) score (0.235 vs. IG's 0.164), with a 55% gain on Mistral-7B for DBPedia (0.389 vs. 0.250); shows better Sufficiency (Suff) with a lower average (0.131 vs. IG's 0.141), notably on Hotel with Llama-3.1-8B (0.060 vs. 0.114); improves stability in Decision Flip Fraction (DFF) (0.685 vs. 0.706), especially on Mistral-7B for IMDB (0.512 vs. 0.541); outperforms in Decision Flip with Most Important Sentence (DFS) (0.187 vs. 0.158), with strong results on Mistral-7B for IMDB (0.325 vs. 0.285); and leads in both Deletion (Del: 0.105 vs. 0.095) and Insertion Rank Correlation (Ins: 0.336 vs. 0.334).

These gains support our central claim: sentence-level prototypes offer a more natural granularity for explaining LLM behavior, capturing semantic reasoning aligned with human understanding. The benefit is especially evident for larger instruction-tuned models like Llama-3.1-8B, where sentence-level explanations more faithfully reflect the model's decision process.

4.3 Impact of Training Data Size (RQ2)

In this section, we investigate how training data size affects ProtoSurE's ability to faithfully reproduce target LLM predictions. Figure 2 presents performance across varying training data sizes for all four target LLMs and three datasets. The key finding is that **ProtoSurE requires relatively little training data to effectively align with target LLMs**. With just 128 training examples, ProtoSurE achieves strong performance (80-85% accuracy) across most LLMs and datasets, demonstrating its data efficiency in capturing LLM decision patterns. For Llama-3.1-8B-Instruct on the Hotel dataset, accuracy reaches 95.0% with just 128 examples, approaching the 98.4% achieved with 1024 examples. Similarly, for DBPedia classification using Mistral-7B, performance with 128 examples (85.5%) closely approximates that with 1024 examples (87.0%). We observe that performance improvements generally plateau after 512 examples, with the Consumer dataset showing more pronounced benefits from additional data across all models. This data efficiency makes ProtoSurE particularly suitable for real-world applications where collecting large labeled datasets maybe impractical.

4.4 Ablation Study (RQ3)

We conduct a comprehensive ablation study to understand how different components contribute to ProtoSurE's overall performance, focusing on encoder selection, token importance design, and prototype updating strategies.
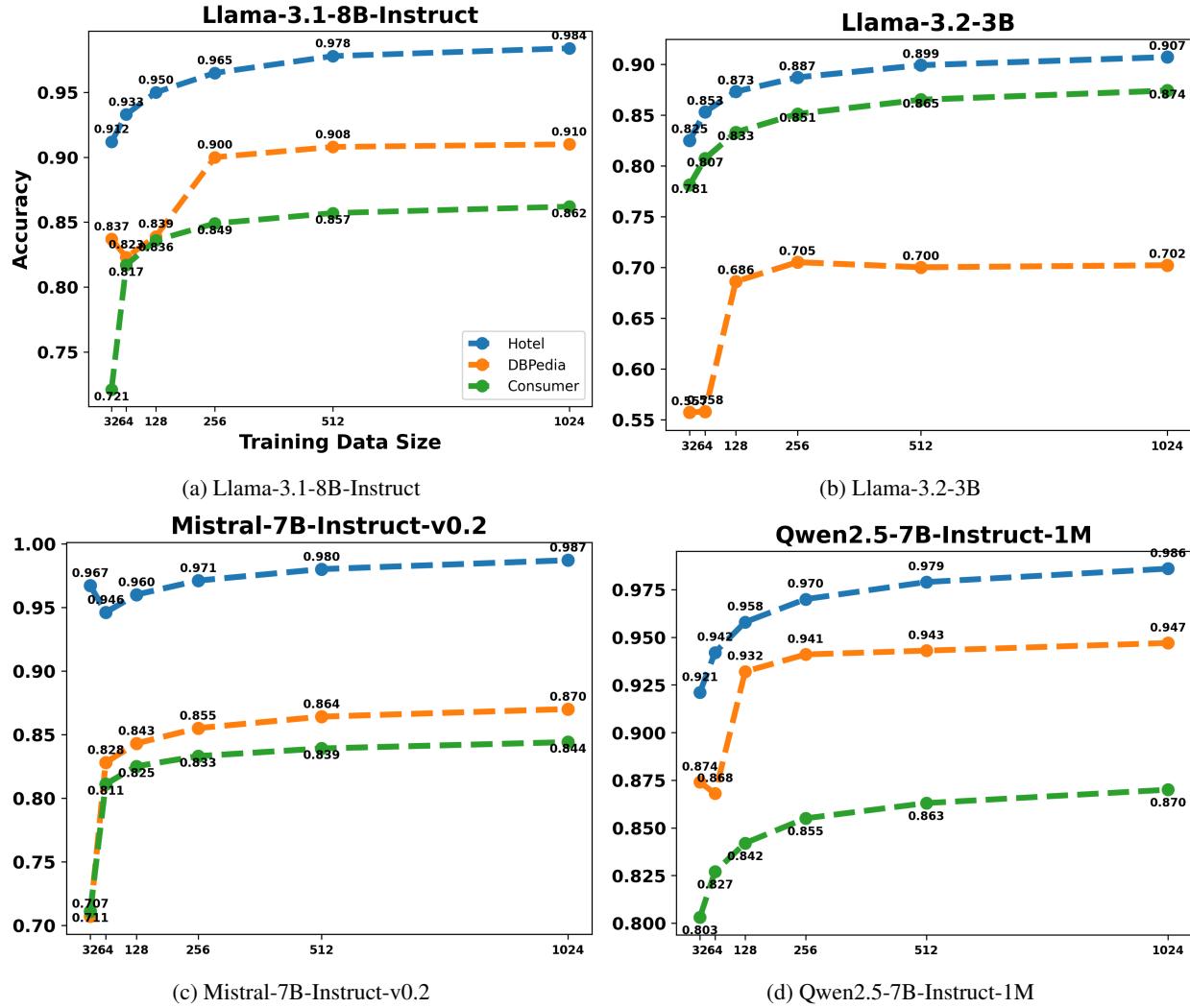
Figure 2: Impact of training data size on ProtoSurE's accuracy across different datasets and target LLMs. Results show consistent performance improvements as training data increases from 32 to 1024 examples. Notably, ProtoSurE achieves reasonable performance (80-85% accuracy) with just 128 training examples across most LLMs and datasets.

Table 3: Overall average accuracy and rank of encoders across all target LLMs and three datasets.

Encoder	Overall Avg	Avg Rank
GTE	**0.8952**	**2.00**
BGE	0.8939	2.50
E5	0.8933	3.00
SBERT	0.8881	3.75
T5	0.8891	3.75

4.4.1 Encoder Impact

The choice of encoder is an important component of ProtoSurE, as it determines the quality of sentence embeddings. We evaluate five state-of-the-art encoders: SBERT Reimers and Gurevych [2019], BGE Chen et al. [2024], GTE Li et al. [2023], E5 Wang et al. [2023], and T5 Ni et al. [2021].

Table 2 presents accuracy results for Llama-3.1-8B-Instruct across three datasets, with full results across all target LLMs available in Appendix E. Notably, all encoders achieve strong performance, with accuracies generally above 0.84

Table 4: Effect of token relevancy maps on accuracy (%) across all target LLMs and datasets. The relevancy map delivers consistent performance improvements across all models and datasets. Best results are in **bold**.

Model Variant	Hotel	DBPedia	Consumer	Avg
Llama-3.1-8B-Instruct				
w/o token attribution	0.975	0.901	0.856	0.911
w/ token attribution	**0.984**	**0.910**	**0.862**	**0.919**
Llama-3.2-3B				
w/o token attribution	0.898	0.695	0.866	0.820
w/ token attribution	**0.907**	**0.702**	**0.874**	**0.828**
Qwen2.5-7B-Instruct-1M				
w/o token attribution	0.979	0.939	0.863	0.927
w/ token attribution	**0.986**	**0.947**	**0.870**	**0.934**
Mistral-7B-Instruct-v0.2				
w/o token attribution	0.980	0.863	0.838	0.894
w/ token attribution	**0.987**	**0.870**	**0.844**	**0.900**

across all datasets and target LLMs. As shown in Table 3, when averaging across all target LLMs and datasets, GTE achieved the highest overall accuracy (0.8952) and average rank (2.00), followed closely by BGE (0.8939, 2.50) and E5 (0.8933, 3.00).

The relatively small performance differences between encoders (within 0.007 accuracy points) demonstrate that ProtoSurE is robust and not limited to any single embedding model. This flexibility is particularly valuable, as it allows practitioners to select encoders based on specific requirements such as efficiency, domain alignment, or resource constraints without performance degradation.

4.4.2 Token Attribution Score Impact

In this section, we examine whether incorporating token-level attributions from the target LLM enhances ProtoSurE's ability to mimic LLM behavior. Table 4 compares ProtoSurE variants with and without token attribution integration across all target LLMs and datasets. The results demonstrate consistent performance improvements when leveraging token attributions, with average accuracy gains ranging from 0.6 to 0.8 percentage points across different LLMs. For instance, incorporating token attribution scores with Llama-3.1-8B improves average accuracy from 91.1% to 91.9%, while similar gains are observed with Qwen2.5-7B (92.7% to 93.4%) and Llama-3.2-3B (82.0% to 82.8%).

These performance gains validate the effectiveness of token-level attributions. By prioritizing tokens deemed significant by the target LLM, ProtoSurE creates more faithful sentence representations that align with the original model's reasoning.

4.4.3 Prototype Update

We investigate whether prototype vectors should be updated during training or kept fixed after initialization. Table 5 demonstrates consistent improvements when prototypes are trainable across all target LLMs and datasets. While fixed prototypes captured by clustering provide a reasonable starting point, allowing them to adapt during training enables more refined decision boundaries that better approximate the target LLM's behavior. Importantly, the updated prototypes maintain interpretability while achieving better alignment with the target LLM's classification patterns, offering an optimal balance between accuracy and human-understandable explanations.

4.5 Hyperparameter Study (RQ4)

We investigate the effect of prototype count K on ProtoSurE's classification performance. Figure 3 present results for $K \in \{10, 20, 40, 100\}$ across three datasets and four LLMs. Our findings demonstrate that accuracy consistently improves as K increases from 10 to 40 before generally plateauing or showing diminishing returns. For Llama-3.1-8B, Hotel accuracy increases from 96.7% at $K = 10$ to 98.0% at $K = 40$, with only marginal improvement to 98.2% at $K = 100$. DBPedia performance with this model peaks earlier at $K = 20$ (91.0%), while Consumer accuracy continues improving to $K = 100$ (83.2%). Other LLMs exhibit similar patterns with optimal performance typically reached

8

Table 5: Impact of prototype updating strategies on ProtoSurE's Accuracy (%) across all target LLMs and datasets. Best results for each target LLM-dataset combination are in **bold**.

Update Strategy	Hotel	DBPedia	Consumer	Avg
Llama-3.1-8B-Instruct				
w/o update	97.9	89.5	81.8	89.7
w/ update (Ours)	**98.4**	**91.0**	**83.2**	**90.9**
Llama-3.2-3B				
w/o update	89.3	69.8	78.6	79.2
w/ update (Ours)	**90.7**	**71.7**	**80.1**	**80.8**
Qwen2.5-7B-Instruct-1M				
w/o update	98.5	93.8	85.6	92.6
w/ update (Ours)	**99.0**	**94.7**	**87.0**	**93.6**
Mistral-7B-Instruct-v0.2				
w/o update	98.1	86.2	83.5	89.3
w/ update (Ours)	**98.7**	**87.5**	**84.6**	**90.3**

between $K = 20$–40; notably, Qwen2.5-7B achieves its best results at $K = 40$ across datasets. These findings suggest that $K = 20$–40 offers an effective balance between model interpretability and representational capacity without excessive computational overhead.

4.6 Case Study (RQ5)

In this section, we present example visualizations of ProtoSurE . Figure 4 demonstrates ProtoSurE's explainability through a hotel review example. The review contains five sentences that our system analyzes against ten prototype categories. Two sentences show strong matches: the mention of "an amazing time" (similarity score: 0.81) with Prototype 1 (Overall positive experience) and staff going "above and beyond" (similarity score: 0.89) with Prototype 2 (Staff responsiveness). Additionally, we observe substantial similarity between the other sentences and Prototypes 6 (Building & style, 0.48), 5 (Comfort & coziness, 0.42), and 9 (Future visit intentions, 0.40). When these sentence-level similarities are aggregated, they yield a decisive positive sentiment prediction (73.8%), with negative prototype activations being negligible. This visualization demonstrates how ProtoSurE provides intuitive, interpretable reasoning for sentiment predictions by showing which specific textual elements activate meaningful prototype patterns. Additional examples are shown in Appendix F, Figure 5, and Figure 6.

5   Conclusion

In this work, we introduced **ProtoSurE**, a prototype-based surrogate framework that delivers faithful and human-understandable explanations for black-box LLMs. By distilling LLM behavior into an interpretable model that matches sentences to semantically meaningful prototypes, ProtoSurE overcomes the limitations of existing post-hoc explanation approaches. Extensive experiments on four state-of-the-art LLMs and four diverse datasets demonstrate that ProtoSurE faithfully reproduces LLM predictions (with an average fidelity over 88%) while providing intuitive explanations. We also show that ProtoSurE is data-efficient, requiring as few as 128 examples to approach full-data performance. In future work, we aim to extend ProtoSurE to span-level prototypes and explore its use in other modalities.

6   Limitations

Assumption of Sentence Coherence.   ProtoSurE assumes that each sentence in the input text conveys a coherent and self-contained semantic unit. This assumption may not hold in settings with irregular sentence boundaries, fragmented user inputs, or highly technical documents, where meaningful phrases span multiple sentences or clauses. As a result, some explanations may lose granularity or coherence in such contexts.

Simple Attribution Aggregation.   Our method aggregates sentence-level prototype contributions using simple averaging and scaling. Although this strategy is transparent and effective, it may fail to capture interactions between
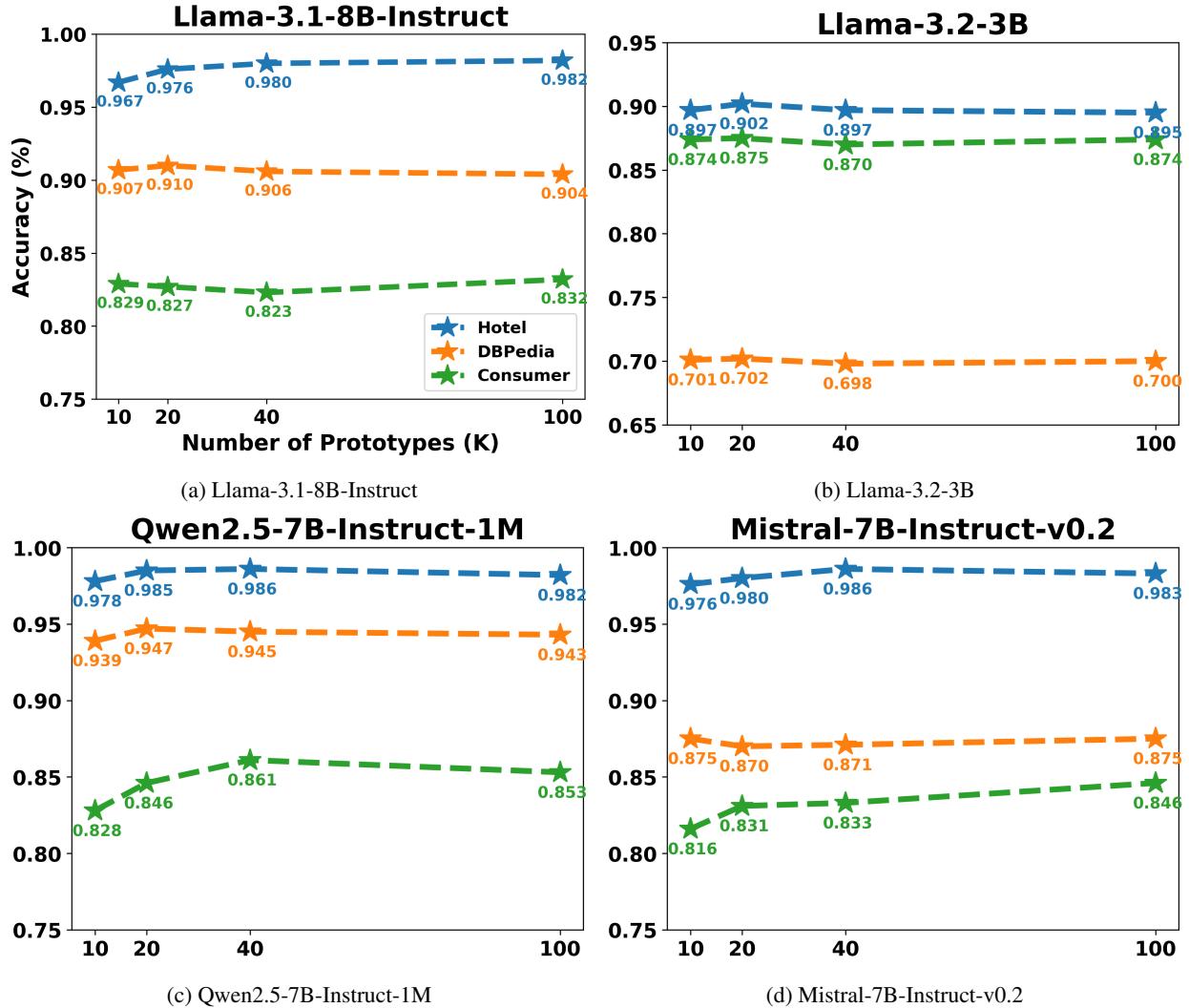
Figure 3: Impact of the number of prototypes (K) on accuracy across different datasets and LLMs. Performance generally improves as K increases until reaching a plateau; the optimal K varies by dataset and model.

sentences or emphasize subtle dependencies. More advanced mechanisms (e.g., attention-based or learnable aggregation) could improve alignment fidelity but would introduce additional complexity.

Lack of Multilingual Support. ProtoSurE is currently evaluated only on English-language data. While the framework itself is language-agnostic, sentence embeddings and attribution behavior can vary significantly across languages. Extending to multilingual or cross-lingual settings would require careful evaluation to ensure interpretability and alignment hold consistently.

No Modeling of Sentence Order. Our approach treats sentences as an unordered set when aggregating prototype influences. For tasks that require reasoning over narrative structure or discourse flow, such as dialogue analysis or story understanding, this may limit the quality of the generated explanations. Incorporating temporal or structural modeling could enhance performance in such scenarios.

7 Ethics

This work focuses on improving transparency and trustworthiness of large language models through post-hoc explanation techniques. ProtoSurE is designed as an interpretability tool and does not modify the underlying LLM, thereby avoiding
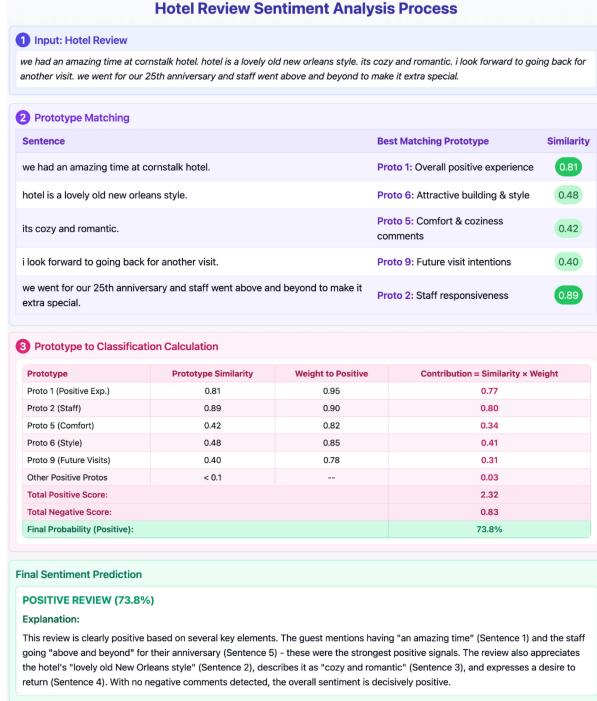
Figure 4: Visualization of the sentiment analysis process for a hotel review using ProtoSurE. The visualization shows: (1) the original review text, (2) prototype matching between sentences and sentiment patterns, (3) aggregated similarity scores across prototype categories, and (4) final sentiment prediction with explanation.

direct manipulation of model behavior. However, explanations produced by ProtoSurE are only as faithful as the surrogate model's approximation of the LLM, and misleading interpretations may arise if the surrogate fails to fully align. Care must be taken when deploying this method in high-stakes applications such as healthcare or legal decision-making. Additionally, while the datasets used in our experiments are publicly available and widely used for academic research, some may contain biased or sensitive content. We encourage practitioners to audit explanations for fairness and refrain from using ProtoSurE to justify harmful or discriminatory decisions.

References

Kun-Hsing Yu, Andrew L Beam, and Isaac S Kohane. Artificial intelligence in healthcare: past, present and future. *Nature biomedical engineering*, 2(10):719–731, 2018.

Lin Zhong, Charley Chen, Ziyu He, Sara Wang, and Ashley Deeks. Does the constitutional right to counsel apply to ai? *Penn State Law Review*, 125:1, 2020.

Douglas W. Arner et al. Fintech and regtech: Enabling innovation while preserving financial stability. *Georgetown Journal of International Law*, pages 367–400, 2020.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328, 2017.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153, 2017.

Alon Jacovi and Yoav Goldberg. 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*, 2020.

Spectra. Demystifying post-hoc explainability for ml models, 2021. URL https://spectra.mathpix.com/article/2021.09.00007/demystify-post-hoc-explainability.

Andreas Madsen et al. Are self-explanations from large language models faithful? *arXiv preprint arXiv:2401.07927*, 2024.

Jason Wei et al. Chain-of-thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.

Tamera Lanham et al. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*, 2023.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *NIPS Deep Learning and Representation Learning Workshop*, 2015.

Chaofan Chen, Oscar Li, Chaofan Tao, Alina J Barnett, Jonathan Su, and Cynthia Rudin. This looks like that: deep learning for interpretable image recognition. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

Yao Ming, Panpan Xu, Huamin Qu, and Liu Ren. Interpretable and steerable sequence learning via prototypes. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019.

Jun Tan, Changxin Wang, Bo Li, Qing Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Feature adaptation via learning a discriminative feature space for domain adaptation. In *Advances in Neural Information Processing Systems*, 2018a.

Patrick Donnelly, Jong Won Baek, Agustin Barla, and Annitta Sridhar. Deep interactive segmentation of medical volumes. In *Medical Image Deep Learning Workshop at MIDL*, 2018.

Seunghoon Hong et al. Protorynet: Prototype trajectory network for text classification with dynamic prototype representations. *arXiv preprint*, 2023.

Bowen Wei and Ziwei Zhu. Advancing interpretability in text classification through prototype learning, 2024. URL https://arxiv.org/abs/2410.17546.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL https://arxiv.org/abs/1810.04805.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "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, 2016.

Christoph Molnar. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. Online book, 2019.

Sarah Tan, Rich Caruana, Giles Hooker, and Yin Lou. Distill-and-compare: Auditing black-box models using transparent model distillation. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 2018b.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization, 2021. URL https://arxiv.org/abs/2012.09838.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33:16857–16867, 2020.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*, 2024.

I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Khanna, Bhargavi Khod, Nazneen Fatema Rajani, Caiming Xiong, Richard Socher, and Dragomir Radev. Eraser: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458. Association for Computational Linguistics, 2020. doi:10.18653/v1/2020.acl-main.408.

Sofia Serrano and Noah A. Smith. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951. Association for Computational Linguistics, 2019. doi:10.18653/v1/P19-1282.

George Chrysostomou and Nikolaos Aletras. Improving the faithfulness of attention-based explanations with task-specific information for text classification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 477–488. Association for Computational Linguistics, 2021. doi:10.18653/v1/2021.acl-long.40.

David Alvarez-Melis and Tommi S. Jaakkola. Towards robust interpretability with self-explaining neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*, pages 7775–7784. Curran Associates, Inc., 2018.

Ronny Luss, Pin-Yu Chen, Amit Dhurandhar, Prasanna Sattigeri, Karthikeyan Shanmugam, and Chun-Chen Tu. Leveraging latent features for local explanations. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1134–1143, 2021. doi:10.1145/3447548.3467229.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992. Association for Computational Linguistics, 2019. doi:10.18653/v1/D19-1410. URL `https://aclanthology.org/D19-1410`.

Shitao Li, Chi Zhang, Jiazhao Ma, Jiawen Ma, Yidong Lv, Yidong Lu, Yuhao Wu, Zihan Wei, Tao Liu, Shuaiqiang Zhao, Ji Zhang, Dawei Zhu, Bin Zhao, and Yelong Liu. Towards generative text embeddings. *arXiv preprint arXiv:2309.15972*, 2023.

Liang Wang, Nan Liu, Xiaoqing Guo, Po-Sen Huang, Xia Liu, Michael Johnson, and Siqi Tang. Text embeddings by weakly-supervised contrastive pre-training. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9157–9171. Association for Computational Linguistics, 2023.

Jianmo Ni, Gustavo Hernández Niu, Daniel Cer, Yun Yang, Noah Constant, Jax Pillias, Benjamin Schlesinger, and Sean Larson. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2861–2873. Association for Computational Linguistics, 2021.

A Datasets

The IMDB dataset contains 25,000 balanced training and test samples and follows a binary sentiment classification format. The Hotel dataset includes 20,000 reviews evaluating 1,000 hotels. Reviews with fewer than 10 characters or containing less than two sentences were excluded.

The DBPedia dataset is a multiclass dataset extracted from Wikipedia. For the experiments in this paper, we use only 4 labels: "Person," "Animal," "Building," and "Natural Place." Similarly, the Consumer Complaints dataset is a multiclass dataset. For the experiments, we use only 4 classes: "Checking or Savings Account," "Credit Card or Prepaid Card," "Debt Collection," and "Mortgage."

B Experimental Details

B.1 Baseline Methods

We compare ProtoSurE against four widely-used post-hoc explanation methods:

- **SHAP** Lundberg and Lee [2017]: A unified approach for interpreting model predictions based on Shapley values, which attributes importance to individual tokens. SHAP assigns attribution scores by calculating the average contribution of each token to the prediction across all possible subsets of tokens. The key insight of SHAP is its ability to satisfy desirable properties like local accuracy, missingness, and consistency. For our implementation, we use KernelSHAP with 1000 samples per instance to approximate the Shapley values.

- **Integrated Gradients (IG)** Sundararajan et al. [2017]: A gradient-based feature attribution method that computes the integral of gradients with respect to inputs along a straight path from a baseline to the input. IG addresses the gradient saturation problem by integrating the gradients along this path. We use a zero embedding as the baseline and approximate the integral using 50 steps of the Riemann sum.

- **Occlusion (Occl)** Zeiler and Fergus [2014]: A perturbation-based approach that measures feature importance by systematically masking individual tokens and observing the impact on the model's output. For each token, we replace it with a padding token and record the change in the output probability. The attribution score is proportional to the magnitude of this change. We experimented with different perturbation strategies (zero embedding, [MASK] token, random token) and found padding tokens to yield the most stable results.

- **DeepLIFT** Shrikumar et al. [2017]: A backpropagation-based attribution technique that compares activation of each neuron to its reference activation and assigns contribution scores. DeepLIFT addresses the gradient saturation problem by considering the difference in activation from a reference state. We use a zero embedding as the reference input, following common practice in the literature.

Each explanation method is applied to explain predictions from four state-of-the-art LLMs:

- **Llama-3.1-8B-Instruct**: A larger instruction-tuned model from the Llama family with 8 billion parameters, designed to follow complex instructions.

- **Llama-3.2-3B**: A more compact model from the Llama family with 3 billion parameters, offering a balance between performance and computational efficiency.

- **Qwen2.5-7B-Instruct-1M**: An instruction-tuned multilingual model with 7 billion parameters, trained on diverse multilingual data.

- **Mistral-7B-Instruct-v0.2**: An instruction-tuned model from Mistral AI with 7 billion parameters, known for its strong performance on various NLP tasks.

For fair comparison, we adapt all baseline methods to provide sentence-level attributions by aggregating token-level scores within each sentence. Specifically, for each sentence, we compute the mean attribution score of all tokens in that sentence. We also experimented with alternative aggregation strategies (max, sum) but found mean aggregation to yield the most reasonable results.

B.2 Evaluation Metrics

We employ seven complementary metrics to assess the faithfulness of explanations:

- **Accuracy (Acc)**: Measures the percentage agreement between the surrogate model's predictions and the target LLM's predictions. Higher values indicate better fidelity of the explainer to the target model. Formally:

$$\text{Acc} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{K}[y_{\text{surr}}(x_i) = y_{\text{tgt}}(x_i)] \tag{10}$$

where $y_{\text{surr}}(x_i) = \text{argmax}(f_{\text{surrogate}}(x_i))$ and $y_{\text{tgt}}(x_i) = \text{argmax}(f_{\text{target}}(x_i))$ are the predicted labels from the surrogate and target models, respectively.

- **Comprehensiveness (Comp)** DeYoung et al. [2020]: Quantifies how much the model's confidence drops when important sentences identified by the explanation method are removed. Calculated as:

$$\text{Comp}(x, e) = 1 - \frac{1}{L+1} \sum_{l=0}^{L} [f(x) - f(\tilde{x}_e^{(l)})] \tag{11}$$

where $f(x)$ is the model's confidence in the predicted class, and $\tilde{x}_e^{(l)}$ is the input with l most important features (sentences in our case) removed. Higher values indicate that the removed content was truly important to the model's decision.

- **Sufficiency (Suff)** DeYoung et al. [2020]: Assesses whether the identified important sentences alone are sufficient to maintain the model's prediction. Calculated as:

$$\text{Suff}(x, e) = 1 - \frac{1}{L+1} \sum_{l=0}^{L} [f(x) - f(\hat{x}_e^{(l)})] \tag{12}$$

where $\hat{x}_e^{(l)}$ is the input with only the l most important features present. Lower values indicate that the identified important sentences capture the essential information needed for the model's decision.

- **Decision Flip Fraction (DFF)** Serrano and Smith [2019]: Measures the fraction of feature removals needed to flip the decision:

$$\text{DFF}(x, e) = \frac{\arg\min_l \ g(\tilde{x}_e^{(l)}) \neq g(x)}{L} \tag{13}$$

where $g(x)$ is the function that outputs the most likely class. We define DFF = 1 if no number of removals leads to a decision flip. Lower values are desirable, indicating that fewer important features need to be removed to change the model's decision.

- **Decision Flip with Most Important Sentence (DFS)** Chrysostomou and Aletras [2021]: Measures whether removing just the single most important sentence changes the model's decision:

$$\text{DFS}(x, e) = \mathbb{K}_{g(\tilde{x}_e^{(1)}) \neq g(x)} \tag{14}$$

where $\tilde{x}_e^{(1)}$ represents the input with the most important sentence removed, and $\mathbb{K}$ is the indicator function that equals 1 when the condition is true and 0 otherwise. Across a dataset, its average value gives the overall decision flip rate, and a higher value is desirable, indicating that the explanation correctly identifies sentences critical to the model's decision.

- **Deletion Rank Correlation (Del)** Alvarez-Melis and Jaakkola [2018]: Evaluates the correlation between feature importance rankings and the impact of removing individual features:

$$\text{Del}(x, e) = \rho(\delta_f, e) \tag{15}$$

where $\delta_f = [f(x) - f(x_{-,e}^{(1)}), \ldots, f(x) - f(x_{-,e}^{(L)})]$, with $x_{-,e}^{(l)}$ being the original input with only the l-th important feature removed, and ρ is the Spearman rank correlation. Higher correlation suggests that suppressing more important features has a larger impact on the model prediction.

- **Insertion Rank Correlation (Ins)** Luss et al. [2021]: Measures the correlation between feature importance rankings and the impact of sequentially adding features:

$$\text{Ins}(x, e) = \rho(v, [0, \ldots, L]) \tag{16}$$

where $v = [f(\tilde{x}_e^{(L)}), \ldots, f(\tilde{x}_e^{(0)})]$, with $\tilde{x}_e^{(l)}$ being the sequence of inputs with increasingly more important features inserted. Higher correlation indicates better alignment between the explanation's importance rankings and the model's behavior when features are incrementally added.

Table 6: Prompts used for each dataset and target LLM. The placeholder {review} is replaced with the actual text to classify.

Dataset	Prompt Template
IMDB / Hotel	Classify the sentiment of the following review as either A (positive) or B (negative). Provide only the letter (A or B) as your response, with no additional explanation. Review: {review} Output:
DBPedia	Classify the following Review into one of the categories: 1 (Person), 2 (Animal), 3 (Building), or 4 (Natural Place). Respond with only the corresponding integer (1, 2, 3, or 4) and no explanation. Your answer must be exactly one of: 1, 2, 3, or 4. Review: {review} Output:
Consumer	Classify the following Review into one of the categories: 1 (Checking or Savings Account), 2 (Credit Card or Prepaid Card), 3 (Debt Collection), or 4 (Mortgage). Respond with only the corresponding integer (1, 2, 3, or 4) and no explanation. Your answer must be exactly one of: 1, 2, 3, or 4. Review: {review} Output:

C   Target LLM Prompting

To generate classifier predictions from the target LLMs, we use task-specific prompts designed to elicit consistent outputs. For binary classification tasks (IMDB and Hotel), we instruct the model to output either "A" (positive) or "B" (negative). For multi-class tasks (DBPedia and Consumer), we ask for an integer from 1 to 4. Table 6 shows the prompts used for each dataset and target LLM.

These prompt templates were consistently applied across all target LLMs (Llama-3.1-8B-Instruct, Llama-3.2-3B, Qwen2.5-7B-Instruct-1M, and Mistral-7B-Instruct-v0.2). The instructions emphasize producing only the classification label without additional explanation, which ensures consistent outputs for our evaluation. For our surrogate model training, we collected 2,000 labeled examples for each dataset-model combination using these prompts.

D   Faithfulness Evaluation

Table 7 presents a detailed comparison of ProtoSurE and four token-level attribution baselines—SHAP, Integrated Gradients (IG), Occlusion, and DeepLIFT—across four datasets and four target LLMs. We evaluate using multiple faithfulness metrics: Accuracy (Acc), Comprehensiveness (Comp), Sufficiency (Suff), Decision Flip Fraction (DFF), Decision Flip with Most Important Sentence (DFS), Deletion Rank Correlation (Del), and Insertion Rank Correlation (Ins).

ProtoSurE consistently outperforms all baselines across most metrics. It achieves the highest Comprehensiveness score (0.235 vs. IG's 0.164), indicating superior ability to identify truly influential input components. It also yields the lowest Sufficiency score (0.131 vs. IG's 0.141), demonstrating that the remaining tokens after removing important ones are less sufficient for prediction—suggesting better focus on salient content. ProtoSurE also shows lower DFF (0.685 vs. IG's 0.706), suggesting higher stability, and higher DFS (0.187 vs. IG's 0.158), reflecting better localization of critical evidence. Additionally, it leads in Del (0.105 vs. Occlusion's 0.095) and Ins (0.336 vs. IG's 0.334), reinforcing that the ranked importance aligns more faithfully with model behavior under perturbations.

Overall, ProtoSurE ranks first across nearly all metrics, validating its effectiveness as a faithful and interpretable explanation framework, particularly when explaining black-box LLM predictions using sentence-level semantic abstractions.

E   Encoder Impact

We investigate the effect of different sentence encoders on ProtoSurE's classification performance across three datasets (Hotel, DBPedia, Consumer) and four target LLMs. As shown in Table 8, we compare five commonly used sentence encoders: SBERT, BGE, GTE, E5, and T5.

GTE consistently achieves strong performance, ranking the best on average across most settings. For instance, it achieves the highest overall accuracy on the DBPedia and Consumer datasets under Llama-3.1-8B and Qwen2.5-7B. BGE and E5 also perform competitively across all LLMs. T5, by contrast, generally lags behind in both accuracy and

Table 7: Faithfulness evaluation : Faithfulness metrics across four target LLMs and four datasets. We report Accuracy (Acc, % ↑), Comprehensiveness (Comp ↑), Sufficiency (Suff ↓), Decision Flip Fraction (DFF, % ↓), Decision Flip with Most Important Sentence (DFS, % ↑), Deletion Rank Correlation (Del ↑), and Insertion Rank Correlation (Ins ↑). Arrows indicate whether higher (↑) or lower (↓) values are better. Avg represents the average value across all datasets and LLMs, while Avg Rank shows the average ranking among all methods (lower is better). Best results for each metric are in **bold**.

Metric	Method	Llama-3.1-8B				Llama-3.2-3B				Qwen2.5-7B				Mistral-7B				Avg	Avg Rank
		IMDB	Hotel	DBPedia	Consumer	IMDB	Hotel	DBPedia	Consumer	IMDB	Hotel	DBPedia	Consumer	IMDB	Hotel	DBPedia	Consumer		
Acc (%) ↑	**ProtoSurE**	**93.5**	**98.4**	**91.0**	**83.2**	**85.5**	**90.7**	**70.2**	**80.1**	**95.6**	**98.6**	**94.7**	**87.0**	**94.7**	**98.7**	**87.0**	**84.4**	**89.58**	–
Comp ↑	SHAP	0.105	0.088	0.176	0.231	-0.006	0.038	0.152	0.042	0.112	0.095	0.231	0.268	0.159	0.148	0.234	0.291	0.153	3.50
	IG	0.118	0.086	0.188	0.241	-0.006	0.083	0.161	0.048	0.121	0.108	0.237	0.273	0.162	0.146	0.250	0.308	0.164	2.19
	Occl	0.106	0.086	0.179	0.231	-0.005	0.064	0.159	0.041	0.108	0.090	0.231	0.271	0.154	0.146	0.243	0.302	0.155	3.44
	DeepLIFT	0.092	0.067	0.167	0.219	-0.008	0.070	0.144	0.039	0.094	0.082	0.198	0.208	0.103	0.098	0.204	0.216	0.129	4.88
	ProtoSurE	**0.156**	**0.141**	**0.271**	**0.268**	**0.002**	**0.109**	**0.212**	**0.104**	**0.172**	**0.146**	**0.283**	**0.317**	**0.365**	**0.343**	**0.389**	**0.351**	**0.235**	**1.00**
Suff ↓	SHAP	**0.205**	0.192	0.234	0.257	0.063	0.050	0.173	0.027	0.196	0.185	0.171	0.190	0.159	0.137	0.184	0.215	0.165	4.03
	IG	0.236	0.114	0.218	0.244	0.104	0.094	0.160	**0.022**	0.138	0.126	**0.166**	0.186	0.098	0.086	**0.164**	**0.201**	0.141	2.09
	Occl	0.208	0.119	0.229	0.253	0.097	0.088	0.164	0.026	0.145	0.132	0.171	**0.184**	**0.095**	0.086	0.174	0.207	0.144	2.78
	DeepLIFT	0.234	0.126	0.240	0.261	0.096	0.084	0.181	0.026	0.157	0.140	0.203	0.244	0.175	0.157	0.216	0.287	0.172	4.41
	ProtoSurE	0.214	**0.060**	**0.207**	**0.231**	**0.051**	**0.039**	**0.156**	0.024	**0.116**	**0.102**	0.170	0.189	0.152	**0.136**	0.173	0.210	**0.131**	**1.69**
DFF (%) ↓	SHAP	0.710	0.694	0.859	0.699	0.695	0.681	0.723	0.759	0.712	0.703	0.860	0.799	0.534	0.523	0.849	0.752	0.722	3.62
	IG	0.684	0.694	0.841	0.678	0.662	0.653	**0.698**	0.730	0.695	0.689	0.854	0.791	0.541	0.530	0.819	0.730	0.706	2.28
	Occl	0.709	0.702	0.858	0.707	0.671	0.658	0.721	0.803	0.701	0.694	0.867	0.798	0.545	0.538	0.836	0.748	0.722	3.69
	DeepLIFT	0.716	0.704	0.877	0.720	0.667	0.652	0.723	0.800	0.723	0.710	0.918	0.881	0.538	0.526	0.900	0.865	0.745	4.34
	ProtoSurE	**0.645**	**0.634**	**0.837**	**0.633**	**0.641**	**0.637**	0.719	**0.721**	**0.662**	**0.651**	**0.850**	**0.771**	**0.512**	**0.504**	**0.815**	**0.728**	**0.685**	**1.06**
DFS (%) ↑	SHAP	0.170	0.180	**0.111**	0.212	0.045	0.041	0.250	0.134	0.023	0.020	0.121	0.081	0.290	0.300	0.090	0.081	0.134	3.56
	IG	0.196	0.205	0.091	0.202	0.043	0.039	0.294	0.175	0.035	0.031	0.162	0.110	0.285	0.280	**0.162**	**0.222**	0.158	2.81
	Occl	0.197	0.205	0.091	0.202	0.037	0.031	0.266	0.130	0.037	0.033	**0.162**	**0.111**	0.287	0.281	0.159	0.222	0.153	2.88
	DeepLIFT	0.183	0.190	0.101	0.182	0.037	0.033	0.255	0.128	0.038	0.033	0.081	0.101	0.268	0.274	0.091	0.051	0.128	4.06
	ProtoSurE	**0.225**	**0.216**	0.091	**0.221**	**0.187**	**0.180**	**0.310**	**0.192**	**0.051**	**0.045**	0.155	0.103	**0.325**	**0.315**	0.152	0.213	**0.187**	**1.69**
Del ↑	SHAP	**-0.005**	0.003	0.137	0.106	0.015	0.030	0.145	0.027	0.012	0.015	0.163	0.053	-0.021	-0.017	0.138	0.039	0.053	4.00
	IG	-0.028	-0.020	0.192	0.166	0.003	0.039	0.173	0.080	0.021	0.018	0.194	0.076	-0.017	-0.011	0.168	0.112	0.075	3.09
	Occl	-0.031	-0.023	0.216	0.183	**0.056**	0.056	0.196	0.098	**0.034**	**0.028**	0.216	**0.169**	-0.016	-0.011	**0.201**	**0.162**	0.095	2.03
	DeepLIFT	-0.019	-0.011	0.165	0.091	0.003	0.012	0.135	0.062	-0.011	-0.008	-0.051	-0.068	0.009	0.013	0.146	-0.099	0.023	4.12
	ProtoSurE	-0.035	**0.084**	**0.237**	**0.187**	0.020	**0.073**	**0.204**	**0.112**	0.031	0.027	**0.210**	0.153	**0.022**	**0.017**	0.192	0.143	**0.105**	**1.75**
Ins ↑	SHAP	0.161	0.153	0.416	0.354	-0.058	0.203	0.399	0.448	0.201	0.192	**0.645**	0.583	0.184	0.179	0.447	0.430	0.325	2.88
	IG	0.219	**0.210**	0.413	0.329	-0.043	0.230	0.392	0.439	**0.212**	**0.204**	0.638	**0.586**	0.183	0.178	0.436	**0.432**	0.334	2.28
	Occl	**0.220**	0.210	0.389	0.316	-0.043	**0.233**	0.385	0.405	0.207	0.198	0.636	0.567	0.182	0.176	0.408	0.400	0.324	3.22
	DeepLIFT	0.164	0.157	0.380	0.322	-0.049	0.219	0.397	0.420	0.186	0.180	0.590	0.523	0.162	0.156	0.400	0.263	0.297	4.50
	ProtoSurE	0.215	0.205	**0.422**	**0.342**	**-0.035**	0.223	**0.405**	**0.450**	0.202	0.196	0.640	0.580	**0.202**	**0.195**	**0.450**	0.425	**0.336**	**2.12**

average rank. These results suggest that selecting a high-quality encoder, particularly one tuned for semantic similarity like GTE or BGE, can substantially improve ProtoSurE's alignment with the underlying LLM's decision boundary.

F Case Study

We provide qualitative examples to illustrate how PROTOSURE decomposes predictions into interpretable prototype-level contributions. Figures 5 and 6 present two hotel reviews—one positive and one negative—and show how each sentence is matched to a semantically aligned prototype, contributing to the final classification.

In the positive case (Figure 5), the model identifies four aspects with high similarity to positive prototypes: room quality (Proto 6), staff friendliness (Proto 4), location (Proto 0), and breakfast quality (Proto 9). All sentences match prototypes with high confidence scores (0.65–0.85), leading to a strong positive classification with 68.4% probability.

Conversely, the negative review (Figure 6) expresses dissatisfaction across multiple dimensions. Sentences match closely with prototypes representing general negative sentiment (Proto 8), room complaints (Proto 3), and staff issues (Proto 4). Each prototype receives a high similarity score (up to 0.91), with negative prototype activations outweighing the positive ones and yielding a 76.5% confidence for the negative class.

These examples demonstrate PROTOSURE's ability to provide faithful, aspect-aware explanations by aligning input sentences with relevant prototypes and transparently combining their contributions into the final decision.

Table 8: Impact of different encoders on ProtoSurE's Accuracy (%) across all target LLMs and datasets. Best results are in **bold**.

Encoder	Hotel	DBPedia	Consumer	Avg Rank
Llama-3.1-8B-Instruct				
SBERT	0.970	0.907	0.859	3.33
BGE	**0.991**	0.906	**0.863**	2.00
GTE	0.984	**0.910**	0.862	**1.67**
E5	0.989	0.908	0.860	2.67
T5	0.978	0.891	0.844	5.00
Llama-3.2-3B				
SBERT	0.867	0.712	0.859	4.33
BGE	0.901	0.708	0.873	3.00
GTE	**0.907**	0.702	0.874	2.33
E5	0.903	**0.717**	0.876	**2.00**
T5	0.892	0.702	**0.881**	3.33
Qwen2.5-7B-Instruct-1M				
SBERT	0.976	0.946	0.864	3.33
BGE	**0.990**	0.943	0.859	2.67
GTE	0.986	**0.947**	**0.870**	**1.33**
E5	**0.990**	0.945	0.846	2.67
T5	0.982	0.936	0.861	4.00
Mistral-7B-Instruct-v0.2				
SBERT	0.981	**0.875**	0.841	3.00
BGE	0.986	0.873	0.834	3.33
GTE	0.987	0.870	0.844	2.67
E5	**0.990**	0.861	0.835	**2.33**
T5	0.986	0.870	**0.846**	3.67

Hotel Review Sentiment Analysis Flow

1 Input Review

rooms were clean and fairly spacious. staff were very friendly and helpful we felt very welcomed. great location only minutes to the beach and to shopping areas. breakfast selection service and quality was top notch.

2 Sentence-to-Prototype Matching

Sentence	Matched Prototype	Similarity
rooms were clean and fairly spacious.	**Proto 5:** Comfort & coziness comments	0.65
staff were very friendly and helpful we felt very welcomed.	**Proto 2:** Friendly Helpful Staff	0.85
great location only minutes to the beach and to shopping areas.	**Proto 11:** Great Convenient Location	0.78
breakfast selection service and quality was top notch.	**Proto 10:** Excellent Breakfast Quality	0.81

3 Prototype to Classification Calculation

Prototype	Prototype Similarity	Weight to Positive	Contribution = Similarity × Weight
Proto 11 (Great Convenient Location)	0.78	0.89	0.69
Proto 2 (Friendly Helpful Staff)	0.85	0.92	0.78
Proto 5 (Comfort & coziness comments)	0.65	0.86	0.56
Proto 10 (Excellent Breakfast Quality)	0.81	0.88	0.71
Other Positive Protos	< 0.1	--	0.04
Total Positive Score:			2.78
Total Negative Score:			1.28
Final Probability (Positive):			68.4%

4 Final Sentiment Prediction

POSITIVE REVIEW (68.4%)

Explanation:

This review is clearly positive because the guest specifically praises four key aspects of their stay: **staff** being "very friendly and helpful" (strongest signal at 85% match), **breakfast** quality as "top notch" (81% match), convenient **location** "minutes to the beach" (78% match), and **rooms** being "clean and spacious" (65% match). The consistent use of positive language and complete absence of any complaints or negative comments reinforces the positive sentiment. The 68% confidence (rather than higher) reflects that while thoroughly positive, the review uses straightforward praise rather than extremely enthusiastic language.
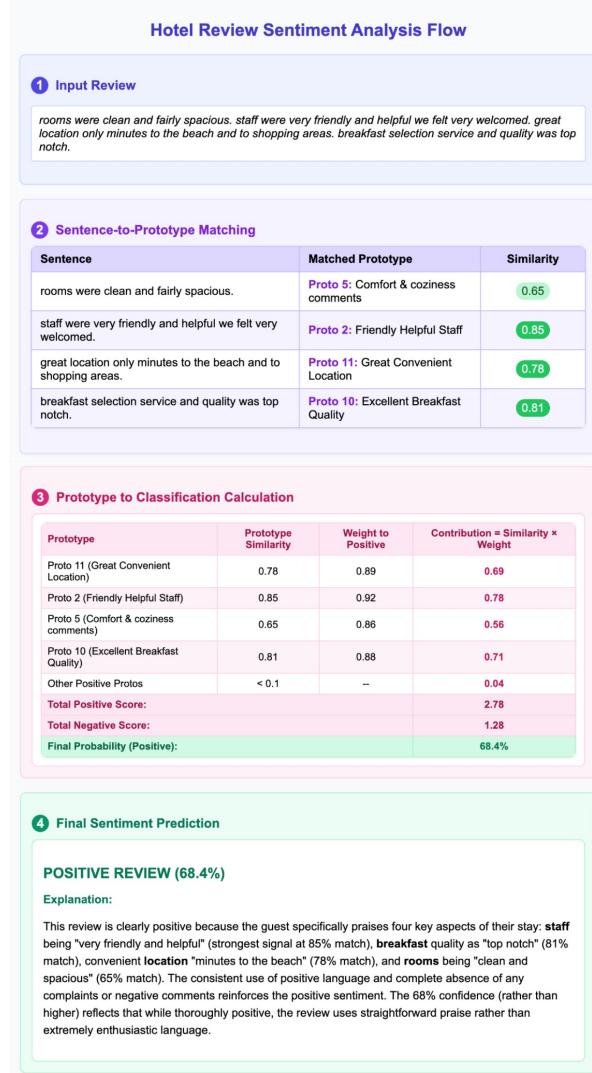
Figure 5: A positive review correctly classified by PROTOSURE, with aligned prototypes highlighting praise for staff, room, location, and breakfast.

Figure 6: A negative review where PROTOSURE identifies strong alignment with negative prototypes such as room complaints and general dissatisfaction.