Making Sense of LLM Decisions: A Prototype-based Framework for Explainable Classification

Bowen Wei, Mehrdad Fazli, Ziwei Zhu

George Mason University
{bwei2, mfazli, zzhu20}@gmu.edu

Abstract

Large language models have demonstrated impressive performance on natural language tasks, but their decision-making processes remain opaque. Existing explanation methods either suffer from limited faithfulness to the model's reasoning or produce explanations that are difficult for humans to understand. To address these challenges, we propose **ProtoSurE**, a novel prototype-based surrogate framework that provides faithful and understandable explanations for LLMs. ProtoSurE trains an interpretable-by-design surrogate model that aligns with the target LLM while utilizing sentence-level prototypes as understandable concepts. Extensive experiments show that ProtoSurE consistently outperforms state-of-the-art 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.

Code — https://github.com/weibowen555/ProtoSurE

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, Beam, and Kohane 2018), law (Zhong et al. 2020), and finance (Arner et al. 2020), where accurate and understandable reasoning is essential. Direct analysis of the internal computations of LLMs is typically computationally intensive, methodologically fragile, and rarely produces explanations that generalize well or are accessible to human users.

Post-hoc explanation methods – such as SHAP (Lundberg and Lee 2017), Integrated Gradients (Sundararajan, Taly, and Yan 2017), Occlusion (Zeiler and Fergus 2014), and DeepLIFT (Shrikumar, Greenside, and Kundaje 2017) – explain models by assigning attribution scores to individual tokens. However, this explanation paradigm often suffers from limited explanation faithfulness (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; Huang et al. 2023) or chain-of-thought reasoning (Wei et al. 2022) – can yield fluent justifications, but these often deviate from the models' actual inference processes (Lanham et al. 2023). These issues highlight two key limitations of existing methods: (1) limited faithfulness to the model's actual decision-making process, and (2) restrictive interpretability from a human perspective.

To address these limitations, we propose **ProtoSurE** (**Proto**type-based **Sur**rogate **E**xplanations), a novel framework (see 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 serve as faithful explanations of the LLM's predictions. To ensure alignment with the LLM and explanation faithfulness, ProtoSurE employs a knowledge distillation approach (Hinton, Vinyals, and Dean 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**, achieved via knowledge distillation-based training to ensure the surrogate accurately reflects the target LLM's behavior, and (2) **human understandability**, enabled by sentence-level prototypes that align with human reasoning. Notably, ProtoSurE attains these properties using only a small set of training samples and does not require access to ground-truth labels, thereby providing strong data

accessibility and efficiency.

In summary, our main contributions are as follows: (1) We propose **ProtoSurE**, a novel prototype-based surrogate explanation framework that delivers faithful, sentence-level explanations for black-box LLM text classification, moving beyond token-level attribution. (2) ProtoSurE consistently and substantially outperforms strong baselines – SHAP, Integrated Gradients, Occlusion, and DeepLIFT – across four LLMs (Llama-3.1-8B, Llama-3.2-3B, Qwen2.5-7B, Mistral-7B) and four diverse benchmarks (IMDB, Hotel, DBPedia, Consumer Complaint), achieving the best faithfulness on all seven evaluation metrics (e.g., $+19.6\%$ higher Comprehensiveness, -3.0% lower Decision Flip Fraction on average), with an overall fidelity of 89.6%. (3) ProtoSurE explanations are both human-comprehensible and highly data-efficient, requiring as few as 128 examples to achieve near-maximum performance, and are broadly applicable to both open- and closed-source LLMs such as GPT-4o-mini.

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, Taly, and Yan 2017), and DeepLIFT (Shrikumar, Greenside, and Kundaje 2017) assign attribution scores to individual tokens. However, these token-level methods struggle with faithfulness (Jacovi and Goldberg 2020) and human understandability (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.

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, Singh, and Guestrin 2016; Molnar 2019). LIME pioneered this approach with local linear approximations, while knowledge distillation techniques (Hinton, Vinyals, and Dean 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.

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 $\hat{\mathcal{Y}} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_N\}$. It is noteworthy that our method requires only a small training dataset with the LLM's predictions and does not rely on labeled data, offering promising practicality and feasibility.

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, Gur, and Wolf 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 texts that drive the LLM's predictions. Experiments in Section 4 confirm the effectiveness of incorporating these attribution scores.

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 hotel 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, where the hyperparameter K specifies the number of prototypes. 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

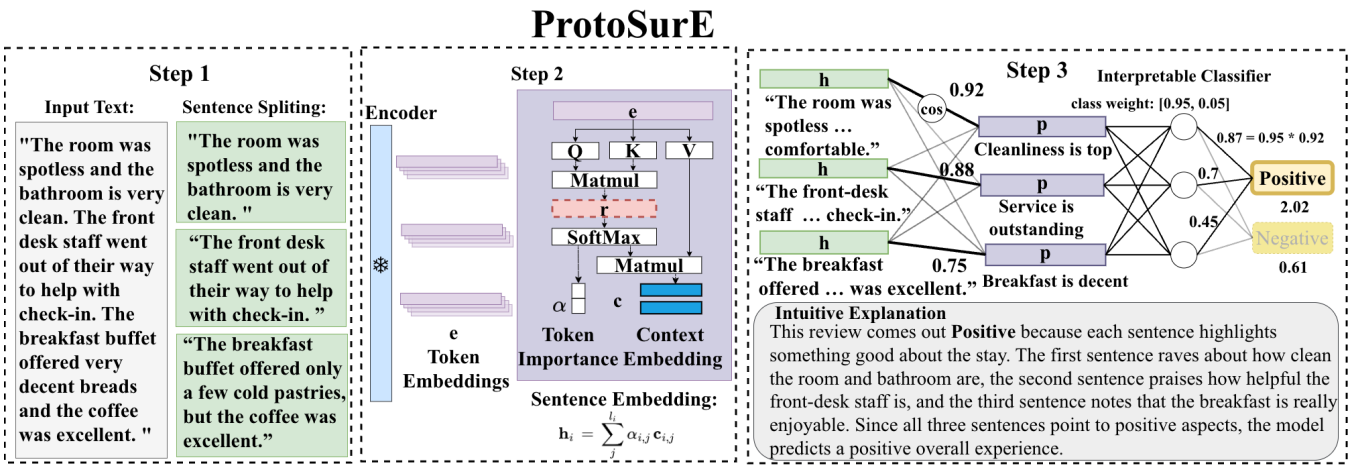

Figure 1: Overview of the ProtoSurE framework. The process consists of three main steps: (1) Sentence splitting; (2) An encoder that processes each sentence to generate token embeddings, applying self attention guided by token attribution scores to provide contextualized token embeddings and token attribution weights; and (3) An interpretable classifier that computes cosine similarities between sentence embeddings and prototypes, then applies class weights to determine the final prediction.

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), which is associated with a learned positive-class weight of 0.95. Thus, the first sentence contributes 0.92 × 0.95 = 0.87 to the positive logit. The second sentence activates the *Service* prototype (similarity = 0.88) with positive-class weight 0.80, contributing 0.70 to the positive logit. The third sentence activates the *Breakfast* prototype (similarity = 0.75) with weight 0.60, contributing 0.45. These three 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.

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.

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

To ensure that the learned surrogate model emulates the decision-making process of the target LLM, we incorporate information about which tokens the LLM relies on for its predictions. We can use established post-hoc explanation methods (e.g., attention relevancy maps (Chefer, Gur, and Wolf 2021), integrated gradients (Sundararajan, Taly, and Yan 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: $\hat{r}_{i,j} = \frac{r_{i,j}}{\sum_{j'=1}^{\ell_i} r_{i,j'} + \epsilon}, \quad \epsilon = 10^{-9}$.

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:

$$A_i = \text{softmax}\left(\frac{\mathbf{Q}_i \mathbf{K}_i^\top}{\sqrt{d}} + \hat{\mathbf{r}}_i\right), \tag{1}$$

$$\mathbf{c}_i = A_i \mathbf{V}_i. \tag{2}$$

To quantify the importance of each token within a sentence, we compute token-level attribution scores $\alpha_{i,j}$. In the self-attention mechanism, each token position attends to all other positions, resulting in an attention matrix A, where $A_{i,k,j}$ denotes the attention weight from source token k to target token j in sentence i. We obtain the overall attention received by each token by averaging these weights across all source positions, thus providing a measure of how much attention each token receives:

$$\alpha_{i,j} = \frac{\exp\left(\frac{1}{\ell_i}\sum_{k=1}^{\ell_i} A_{i,k,j}\right)}{\sum_{j'=1}^{\ell_i} \exp\left(\frac{1}{\ell_i}\sum_{k=1}^{\ell_i} A_{i,k,j'}\right)}. \tag{3}$$

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

$$\mathbf{h}_i = \sum_{j=1}^{\ell_i} \alpha_{i,j} \cdot \mathbf{c}_{i,j} \in \mathbb{R}^d, \tag{4}$$

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.

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 understandable explanations, we associate each prototype with its nearest sentence from the training data based on cosine similarity to explain the prototype (e.g., the three prototypes in Figure 1 are about positive reviews of cleanliness, service, and breakfast). 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.

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.

Training Objective

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

$$\mathcal{L} = \text{CrossEntropy}(\hat{y}, \tilde{y}) + \lambda_1 \mathcal{L}_{\text{proto}} + \lambda_2 \mathcal{L}_{\text{diversity}}. \quad (5)$$

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:

$$\mathcal{L}_{\text{proto}} = -\frac{1}{P} \sum_{k=1}^{P} \max_i \text{sim}(\mathbf{h}_i, \mathbf{p}_k). \quad (6)$$

The diversity loss penalizes overlap between prototypes:

$$\mathcal{L}_{\text{diversity}} = \frac{1}{P(P-1)} \sum_{i=1}^{P} \sum_{\substack{j=1 \\ j \neq i}}^{P} |\mathbf{p}_i^\top \mathbf{p}_j|. \quad (7)$$

4 Experiments

In this section, we evaluate ProtoSurE along several dimensions to address the following research questions: **RQ1:** How faithfully does ProtoSurE explain LLM predictions in comparison to existing explanation methods? **RQ2:** How does the size of the training data impact ProtoSurE's performance? **RQ3:** How do key hyperparameters influence the model's effectiveness? **RQ4:** What is the contribution of each core component (encoder, token attribution, prototype updating) to the overall model performance? **RQ5:** What are the qualitative characteristics of ProtoSurE explanations as illustrated in a case study?

Experimental Setup

Datasets. We evaluate ProtoSurE on four 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, 50, 100, 1000, 4000}, and we set $\lambda_1 = 0.1$, and $\lambda_2 = 0.1$ for the loss components. We employ the relevancy map approach (Chefer, Gur, and Wolf 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 five widely used explanation methods: SHAP (Lundberg and Lee 2017), Integrated Gradients (IG) (Sundararajan, Taly, and Yan 2017), Occlusion (Zeiler and Fergus 2014), DeepLIFT (Shrikumar, Greenside, and Kundaje 2017), and SELF-EXP (Huang et al. 2023). Each method is applied to explain predictions from five target LLMs: Llama-3.1-8B-Instruct, Llama-3.2-3B, Qwen2.5-7B-Instruct-1M, Mistral-7B-Instruct-v0.2, and GPT-4o-mini. 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.

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).

Faithfulness Evaluation (RQ1)

In this section, we evaluate ProtoSurE's faithfulness in explaining predictions made by target LLMs compared to state-of-the-art post-hoc explanation methods. Faithfulness is assessed from two perspectives: (1) alignment with the target LLM, measured by accuracy in predicting LLM classification outcomes, and (2) fidelity to the LLM's underlying reasoning, measured by established faithfulness metrics.

Results across four open-source LLMs and four datasets are summarized in Table 1. We compare ProtoSurE against four baseline methods (SHAP, IG, Occlusion, DeepLIFT) in this comprehensive evaluation. Note that SELF-EXP, while included in our GPT-4o-mini evaluation, is excluded from Table 1 due to its unreliable performance with open-source models – it frequently fails to generate consistent explanations, ignores required output formats, or produces incomplete attributions for Llama, Qwen, and Mistral models.

Metric	Method	Llama-3.1-8B				Llama-3.2-3B				Qwen2.5-7B				Avg	Rank
		IM	Hot	DB	Con	IM	Hot	DB	Con	IM	Hot	DB	Con		
Acc ↑	**Ours**	**93.5**	**98.4**	**91.0**	**83.2**	**85.5**	**90.7**	**70.2**	**80.1**	**95.6**	**98.6**	**94.7**	**87.0**	**87.4**	**–**
Comp ↑	SH	.105	.088	.176	.231	-.006	.038	.152	.042	.112	.095	.231	.268	.128	3.42
	IG	.118	.086	.188	.241	-.006	.083	.161	.048	.121	.108	.237	.273	.138	2.25
	Oc	.106	.086	.179	.231	-.005	.064	.159	.041	.108	.090	.231	.271	.130	3.33
	DL	.092	.067	.167	.219	-.008	.070	.144	.039	.094	.082	.198	.208	.114	4.92
	Ours	**.156**	**.141**	**.191**	**.268**	**.002**	**.109**	**.212**	**.104**	**.172**	**.146**	**.283**	**.317**	**.192**	**1.08**
Suff ↓	SH	**.205**	.192	.234	.257	.063	.050	.173	.027	.196	.185	.171	.190	.162	3.75
	IG	.236	.114	.218	.244	.104	.094	.160	**.022**	.138	.126	**.166**	.186	.151	2.25
	Oc	.208	.119	.229	.253	.097	.088	.164	.026	.145	.132	.171	**.184**	.151	2.67
	DL	.234	.126	.240	.261	.096	.084	.181	.026	.157	.140	.203	.244	.166	4.42
	Ours	.214	**.060**	**.207**	**.231**	**.051**	**.039**	**.156**	.024	**.116**	**.102**	.170	.189	**.130**	**1.92**
DFF ↓	SH	.710	.694	.859	.699	.695	.681	.723	.759	.712	.703	.860	.799	.741	3.58
	IG	.684	.694	.841	.678	.662	.653	**.698**	.730	.695	.689	.854	.791	.722	2.25
	Oc	.709	.702	.858	.707	.671	.658	.721	.803	.701	.694	.867	.798	.741	3.67
	DL	.716	.704	.877	.720	.667	.652	.723	.800	.723	.710	.918	.881	.758	4.42
	Ours	**.645**	**.634**	**.837**	**.633**	**.641**	**.637**	.719	**.721**	**.662**	**.651**	**.850**	**.771**	**.700**	**1.08**
DFS ↑	SH	.170	.180	**.111**	.212	.045	.041	.250	.134	.023	.020	.121	.081	.116	3.58
	IG	.196	.205	.091	.202	.043	.039	.294	.175	.035	.031	.162	.110	.132	2.92
	Oc	.197	.205	.091	.202	.037	.031	.266	.130	.037	.033	**.162**	**.111**	.125	2.75
	DL	.183	.190	.101	.182	.037	.033	.255	.128	.038	.033	.081	.101	.113	4.00
	Ours	**.225**	**.216**	.091	**.221**	**.187**	**.180**	**.310**	**.192**	**.051**	**.045**	.155	.103	**.165**	**1.75**
Del ↑	SH	**-.005**	.003	.107	.106	.015	.030	.145	.027	.012	.015	.163	.053	.056	3.83
	IG	-.028	-.020	.092	.166	.003	.039	.173	.080	.021	.018	.194	.076	.068	3.25
	Oc	-.031	-.023	.116	.183	**.056**	.056	.196	.098	**.034**	**.028**	**.216**	**.169**	.092	1.83
	DL	-.019	-.011	.065	.091	.003	.012	.135	.062	-.011	-.008	-.051	-.068	.017	4.17
	Ours	-.035	**.084**	**.137**	**.187**	.020	**.073**	**.204**	**.112**	.031	.027	.210	.153	**.100**	**1.92**
Ins ↑	SH	.161	.153	.516	.304	-.058	.203	.399	.448	.201	.192	**.645**	.583	.312	3.25
	IG	.219	**.210**	.513	.329	-.043	.230	.392	.439	**.212**	**.204**	.638	**.586**	.327	2.33
	Oc	**.220**	.210	.559	.316	-.043	**.233**	.385	.405	.207	.198	.636	.567	.324	3.17
	DL	.164	.157	.580	.322	-.049	.219	.397	.420	.186	.180	.590	.523	.307	4.25
	Ours	.215	.205	**.591**	**.342**	**-.035**	.223	**.405**	**.450**	.202	.196	.640	.580	**.335**	**2.00**

Table 1: Faithfulness across three target LLMs and four datasets on 7 metrics introduced in Section 4. Best results are in bold. Full Results are detailed in the Appendix.

ProtoSurE achieves strong alignment with target LLM behavior (89.6% average accuracy) and consistently outperforms all baselines on faithfulness metrics. It attains the highest Comprehensiveness (0.189), lowest Sufficiency (0.139), best Decision Flip Fraction (0.685), highest Decision Flip with Most Important Sentence (0.186), and superior ranking correlations (Deletion: 0.099, Insertion: 0.330), demonstrating its effectiveness in identifying sentences that reflect the LLM's reasoning.

Extension to GPT-4o-mini. To demonstrate ProtoSurE's generalizability beyond open-source LLMs, we evaluate its performance on GPT-4o-mini using the Hotel dataset, comparing against SHAP, Occlusion, and SELF-EXP (Table 2). Unlike with open-source models, SELF-EXP performs reliably with GPT-4o-mini, making it a viable baseline for this evaluation. ProtoSurE consistently outperforms all baselines across every faithfulness metric, achieving the highest Comprehensiveness (0.005 vs. negative values for SHAP/Occlusion), lowest Sufficiency (0.001 vs. 0.026 for SHAP), best Decision Flip Fraction (74.9% vs. 75.8-78.4%), highest Decision Flip with Most Important Sentence (19.7% vs. 14.2-18.8%), and superior ranking correlations (Deletion: -0.239, Insertion: 0.116). These results demonstrate that ProtoSurE maintains its superior faithfulness and explanation quality when applied to closed-source models, confirming its broad applicability across diverse LLM architectures.

Metric	SHAP	Occl	SELF-EXP	ProtoSurE
Comp ↑	-0.020	-0.018	0.004	**0.005**
Suff ↓	0.026	0.002	0.017	**0.001**
DFF (%) ↓	0.758	0.784	0.769	**0.749**
DFS (%) ↑	0.188	0.163	0.142	**0.197**
Del ↑	-0.307	-0.368	-0.289	**-0.239**
Ins ↑	0.102	0.107	0.104	**0.116**

Table 2: Faithfulness on GPT-4o-mini (Hotel).

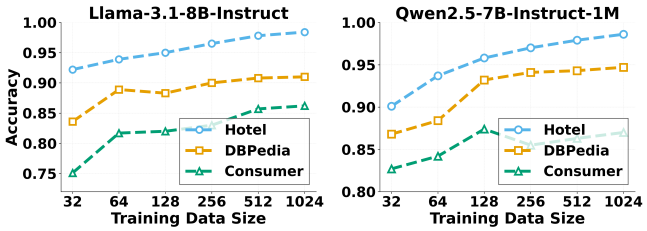

Figure 2: Impact of training data size on ProtoSurE's accuracy across different datasets and target LLMs.

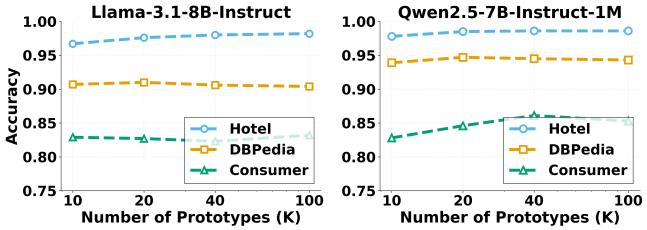

Figure 3: Impact of the number of prototypes (K) on accuracy across different datasets and LLMs.

Impact of Training Data Size (RQ2)

Figures 2 and 4 demonstrate that ProtoSurE achieves high accuracy and faithfulness with as few as 128–256 training examples for most datasets. For example, on Llama-3.1-8B-Instruct, the Hotel dataset reaches 96.5% accuracy with just 128 examples, approaching the maximum performance of 98.4% achieved with 1024 examples. Both accuracy and faithfulness metrics plateau quickly as training size increases, demonstrating that only a small dataset is needed for strong performance. Importantly, ProtoSurE requires only a small number of input texts and corresponding LLM predictions for training – no ground-truth labels are needed, making it highly practical for real-world applications where labeled data is scarce or expensive to obtain.

Impact of Prototype Number (RQ3)

We explore how the prototype number K affects ProtoSurE's faithfulness metrics, as depicted in Figure 3 and Figure 5. We evaluate values of $K \in \{10, 20, 50, 100, 1000, 4000\}$ on DBPedia and Consumer datasets using Llama-3.1-8B-Instruct. Results indicate that increasing K generally improves all faithfulness metrics. Specifically, completeness improves consistently with higher prototype numbers on both datasets, whereas sufficiency remains largely stable or declines slightly. DFS and Del also exhibit steady improvements with increasing K, particularly noticeable up to $K = 1000$. Conversely, DFF notably decreases as K rises. Overall, the results indicate that a higher number of prototypes enables finer-grained prototype concepts, thus enhancing the model's faithfulness.

Ablation Study (RQ4)

We conduct a comprehensive ablation study to understand how different components contribute to ProtoSurE's overall

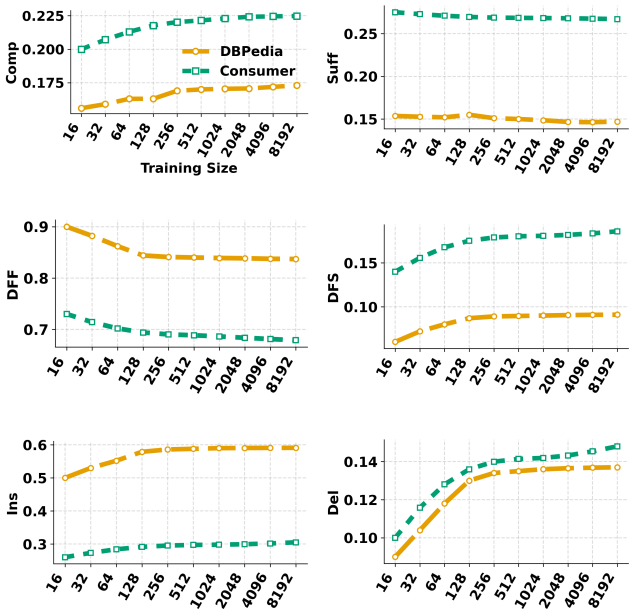

Figure 4: Effect of training size N on faithfulness.

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

Table 3: Overall average accuracy and rank of encoders.

performance, focusing on encoder selection, token importance design, and prototype updating strategies.

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). As shown in Table 3, when averaging across all target LLMs and datasets, GTE achieves the highest overall accuracy (0.8952) and average rank (2.00), followed closely by BGE and E5, with the full results across all target LLMs available in Appendix E. The relatively small performance differences between encoders (within approximately 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 significant performance degradation.

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 and Table 10 in Appendix F compares ProtoSurE variants with and without token attribution integration across all target LLMs and datasets. The results demonstrate consistent performance improvements when leverag-

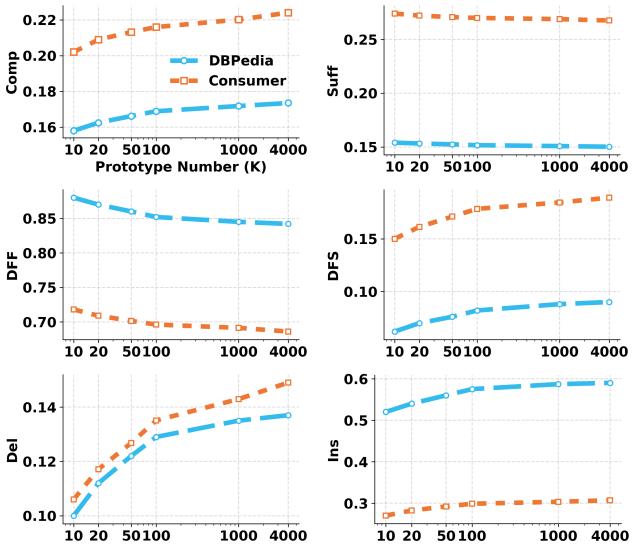

Figure 5: Effect of prototype number K on faithfulness.

Model Variant	Hotel	DBPedia	Consumer	Avg
Llama-3.1-8B-Instruct				
w/o attribution	0.975	0.901	0.856	0.911
w/ attribution	**0.984**	**0.910**	**0.862**	**0.919**

Table 4: Effect of token attribution on accuracy (%).

ing 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 target LLM's reasoning.

Prototype Update We examine whether prototype vectors should remain fixed after initialization or be updated during training. As shown in Table 5 and Table 11 in Appendix G, making prototypes trainable consistently improves performance across all target LLMs and datasets. While fixed prototypes provide reasonable initialization, updating prototypes during training yields more precise decision boundaries that better match the target LLM's behavior. Notably, the learned prototypes retain their interpretability while providing better alignment with the target model's predictions, achieving balance between accuracy and understandability.

Qualitative Case Study (RQ5)

In this section, we present example visualizations of ProtoSurE. Table 6 demonstrates ProtoSurE's explainability through a hotel review example with three sentences analyzed against three prototype categories. The review shows strong matches with Prototype 1 (Cleanliness, similarity: 0.92), Prototype 2 (Service, similarity: 0.88), and Prototype 3 (Food, similarity: 0.75). When these sentence-level similarities are aggregated with their respective class weights, they yield a decisive positive sentiment prediction, with negligible negative activations. This visualization demonstrates how ProtoSurE provides intuitive, interpretable reasoning by showing which specific textual elements activate meaningful prototype patterns. Additional examples are shown in Appendix H, Figure 9, and Figure 10.

Update Strategy	Hotel	DBPedia	Consumer	Avg
Llama-3.1-8B-Instruct				
w/o update	97.9	89.5	81.8	89.7
w/ update	**98.4**	**91.0**	**83.2**	**90.9**
Qwen2.5-7B-Instruct-1M				
w/o update	98.5	93.8	85.6	92.6
w/ update	**99.0**	**94.7**	**87.0**	**93.6**

Table 5: Impact of prototype update on Accuracy (%).

Input: Hotel Review		
"The room was spotless and the bed was extremely comfortable. The staff were incredibly helpful. The complimentary breakfast exceeded expectations."		
Prototype Matching		
Sentence	**Proto**	**Sim**
The room was spotless and the bed was extremely comfortable.	P1: Clean	0.92
The staff were incredibly helpful.	P2: Service	0.88
The complimentary breakfast exceeded expectations.	P3: Food	0.75
POSITIVE REVIEW		
Explanation: This review is positive based on three key aspects. The guest praises cleanliness and comfort (Sentence 1, similarity 0.92), which strongly contributes to the positive classification (0.87). Staff service (Sentence 2, similarity 0.88) and breakfast quality (Sentence 3, similarity 0.75) further support the positive sentiment.		

Table 6: Visualization of the sentiment analysis process.

5 Conclusion

We introduced ProtoSurE, which provides faithful and human-understandable explanations for black-box LLMs. By distilling LLM behavior into an interpretable model that aligns sentences with semantically meaningful prototypes, ProtoSurE addresses key limitations of existing post-hoc explanation approaches. Extensive experiments on five LLMs and four diverse datasets show that ProtoSurE faithfully reproduces LLM predictions and delivers intuitive explanations. ProtoSurE is highly data-efficient, requiring as few as 128 examples to approach optimal performance. Overall, ProtoSurE is a practical and feasible solution for explainable LLM deployment in real-world applications.

Acknowledgments

This work is in part supported by NSF grant IIS-2452129. Computational resources for experiments were provided by the Office of Research Computing at George Mason University (URL: https://orc.gmu.edu) and funded in part by grants from the National Science Foundation (Awards Number 1625039 and 2018631).

References

Alvarez-Melis, D.; and Jaakkola, T. S. 2018. Towards Robust Interpretability with Self-explaining Neural Networks. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*, 7775–7784. Curran Associates, Inc.

Arner, D. W.; et al. 2020. FinTech and RegTech: Enabling Innovation While Preserving Financial Stability. *Georgetown Journal of International Law*, 367–400.

Chefer, H.; Gur, S.; and Wolf, L. 2021. Transformer Interpretability Beyond Attention Visualization. arXiv:2012.09838.

Chen, C.; Li, O.; Tao, C.; Barnett, A. J.; Su, J.; and Rudin, C. 2019. This looks like that: deep learning for interpretable image recognition. In *Advances in Neural Information Processing Systems*, volume 32.

Chen, J.; Xiao, S.; Zhang, P.; Luo, K.; Lian, D.; and Liu, Z. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Chrysostomou, G.; and Aletras, N. 2021. 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*, 477–488. Association for Computational Linguistics.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805.

DeYoung, J.; Jain, S.; Khanna, N. F.; Khod, B.; Rajani, N. F.; Xiong, C.; Socher, R.; and Radev, D. 2020. ERASER: A Benchmark to Evaluate Rationalized NLP Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4443–4458. Association for Computational Linguistics.

Donnelly, P.; Baek, J. W.; Barla, A.; and Sridhar, A. 2018. Deep interactive segmentation of medical volumes. In *Medical Image Deep Learning Workshop at MIDL*.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *NIPS Deep Learning and Representation Learning Workshop*.

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

Huang, S.; Mamidanna, S.; Jangam, S.; Zhou, Y.; and Gilpin, L. H. 2023. Can Large Language Models Explain Themselves? A Study of LLM-Generated Self-Explanations. arXiv:2310.11207.

Jacovi, A.; and Goldberg, Y. 2020. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

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

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

Li, S.; Zhang, C.; Ma, J.; Ma, J.; Lv, Y.; Lu, Y.; Wu, Y.; Wei, Z.; Liu, T.; Zhao, S.; Zhang, J.; Zhu, D.; Zhao, B.; and Liu, Y. 2023. Towards generative text embeddings. *arXiv preprint arXiv:2309.15972*.

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

Lundberg, S. M.; and Lee, S.-I. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

Luss, R.; Chen, P.-Y.; Dhurandhar, A.; Sattigeri, P.; Shanmugam, K.; and Tu, C.-C. 2021. Leveraging Latent Features for Local Explanations. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 1134–1143.

Madsen, A.; et al. 2024. Are self-explanations from Large Language Models faithful? *arXiv preprint arXiv:2401.07927*.

Ming, Y.; Xu, P.; Qu, H.; and Ren, L. 2019. Interpretable and steerable sequence learning via prototypes. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

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

Ni, J.; Niu, G. H.; Cer, D.; Yang, Y.; Constant, N.; Pillias, J.; Schlesinger, B.; and Larson, S. 2021. Sentence-T5: Scalable Sentence Encoders from Pre-trained Text-to-Text Models. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2861–2873. Association for Computational Linguistics.

Reimers, N.; and Gurevych, I. 2019. 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)*, 3982–3992. Association for Computational Linguistics.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1135–1144.

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

Serrano, S.; and Smith, N. A. 2019. Is Attention Interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2931–2951. Association for Computational Linguistics.

Shrikumar, A.; Greenside, P.; and Kundaje, A. 2017. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, 3145–3153.

Song, K.; Tan, X.; Qin, T.; Lu, J.; and Liu, T.-Y. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33: 16857–16867.

Spectra. 2021. Demystifying Post-hoc Explainability for ML models.

Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, 3319–3328.

Tan, J.; Wang, C.; Li, B.; Li, Q.; Ouyang, W.; Yin, C.; and Yan, J. 2018a. Feature adaptation via learning a discriminative feature space for domain adaptation. In *Advances in Neural Information Processing Systems*.

Tan, S.; Caruana, R.; Hooker, G.; and Lou, Y. 2018b. Distill-and-compare: Auditing black-box models using transparent model distillation. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*.

Wang, L.; Liu, N.; Guo, X.; Huang, P.-S.; Liu, X.; Johnson, M.; and Tang, S. 2023. Text Embeddings by Weakly-Supervised Contrastive Pre-training. In *Findings of the Association for Computational Linguistics: ACL 2023*, 9157–9171. Association for Computational Linguistics.

Wei, B.; and Zhu, Z. 2024. Advancing Interpretability in Text Classification through Prototype Learning. arXiv:2410.17546.

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

Yu, K.-H.; Beam, A. L.; and Kohane, I. S. 2018. Artificial intelligence in healthcare: past, present and future. *Nature biomedical engineering*, 2(10): 719–731.

Zeiler, M. D.; and Fergus, R. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, 818–833. Springer.

Zhong, L.; Chen, C.; He, Z.; Wang, S.; and Deeks, A. 2020. Does the constitutional right to counsel apply to AI? *Penn State Law Review*, 125: 1.