



ARTICLE

DRIVE: Diagnostic Report Integration via VLM and LLM Explanations for Explainable Vehicle Engine Fault Diagnosis

Jaeseung Lee¹ and Jehyeok Rew^{2,*}

¹School of Electrical Engineering, Korea University, Seoul, Republic of Korea

²Department of Data Science, Duksung Women's University, Seoul, Republic of Korea

*Corresponding Author: Jehyeok Rew. Email: jhrew@duksung.ac.kr

Received: 28 November 2025; Accepted: 08 January 2026; Published: 27 April 2026

ABSTRACT: The engine serves as the primary component that generates power and drives vehicle movement. Given its critical role, accurately diagnosing engine faults is essential for ensuring vehicle safety and reliability. Recent advances in machine learning (ML) have enabled the development of artificial intelligence (AI)-based diagnostic models with strong predictive performance. However, the lack of transparency in these models constrains user confidence in their diagnostic outcomes. While explainable AI (XAI) methods such as local interpretable model-agnostic explanations (LIME) and Shapley additive explanations (SHAP) have been introduced to improve interpretability, their reliance on visual outputs requires manual interpretation, which can be inefficient and prone to subjectivity. To address this limitation, we propose DRIVE, a novel method for explainable vehicle engine fault diagnosis. In DRIVE, LIME and SHAP are applied to an ML-based diagnostic model, and their visual outputs are translated into textual explanations using the vision-language models (VLMs). These complementary explanations are then synthesized by a large language model (LLM) into a unified diagnostic report, providing a coherent narrative of the model's reasoning and emphasizing abnormal input features. Experiments conducted on a publicly available vehicle engine fault dataset demonstrate that DRIVE not only produces accurate and transparent diagnostic rationales but also generates structured reports that enhance usability for domain experts. By integrating multiple XAI methods with multimodal LLMs, DRIVE advances the transparency, trustworthiness, and practicality of AI-driven vehicle engine fault diagnosis.

KEYWORDS: Vehicle engine; fault diagnosis; vision-language model; large language model; explainable artificial intelligence; local interpretable model-agnostic explanations; Shapley additive explanations; energy

1 Introduction

The internal combustion engine is widely used in modern vehicles, serving as a core component that governs vehicle performance, reliability, and operational efficiency [1]. Despite its central role, the engine remains vulnerable to various types of operational failures, including fuel delivery imbalances, ignition system malfunctions, and intake airflow disturbances [2]. These faults emerge subtly in their early stages, making them difficult to detect without intelligent monitoring systems. As a result, such malfunctions can compromise the safety of drivers and passengers, elevate maintenance costs, and cause financial losses due to unexpected downtime [3]. Moreover, undetected or unresolved faults result in excessive emissions, leading to environmental degradation and non-compliance with regulatory standards. In this context, the development of advanced diagnostic methods for early fault detection and predictive maintenance has emerged as a key area for both vehicle manufacturers and owners [4,5]. Beyond vehicle-level diagnostics, accurate engine fault diagnosis plays an important role in energy systems by improving fuel efficiency, reducing unnecessary

energy consumption, and mitigating excessive emissions [6]. From a broader energy-system perspective, data-driven diagnostic frameworks contribute to more efficient utilization of energy resources and support sustainable operation of transportation-related energy infrastructures [7].

Early approaches to vehicle engine fault diagnosis primarily relied on rule-based systems derived from domain-specific expert knowledge [8]. While these methods were effective in identifying predefined fault patterns, they struggled to detect novel or previously unseen anomalies. Furthermore, they required continuous expert involvement for maintenance and updates, which limited their scalability and increased operational costs. As a result, there has been a growing transition toward data-driven diagnostic frameworks that leverage advanced computational intelligence technologies to improve fault detection and decision-making.

Recent advancements in machine learning (ML) have significantly enhanced vehicle engine fault diagnosis by enabling accurate and efficient data-driven fault classification [4,9]. In parallel, more advanced deep learning-based diagnostic frameworks have been proposed to address challenges such as data scarcity, and multi-sensor heterogeneity in complex mechanical systems [10]. Several studies have explored few-shot and multimodal large model approaches to improve fault diagnosis performance under limited labeled data, as well as graph-based and channel-adaptive feature methods to robustly integrate multi-sensor signals [11,12]. Despite their performance gains, most ML-based diagnostic models are composed of complex internal structures, making it difficult to interpret how individual input features influence diagnostic outcomes. As a result, they are frequently characterized as ‘black box’ models [13]. Moreover, their reliance on high-dimensional sensor data poses additional challenges for effective visualization and intuitive understanding, thereby hindering root-cause analysis and model refinement when incorrect predictions occur [14]. These limitations highlight the growing need for interpretable and trustworthy diagnostic reasoning that can explain model decisions beyond predictive accuracy alone.

Explainable artificial intelligence (XAI) has recently emerged as a key approach for improving the transparency and trustworthiness of ML models [15]. XAI refers to a set of techniques that aim to interpret how an ML model generates its outputs and to present these interpretations in a human-understandable form. By elucidating the underlying mechanisms behind model predictions, XAI facilitates greater interpretability and fosters user trust in automated decision systems [16]. Extensive research has been conducted to enhance the explainability of ML models, including Local Interpretable Model-agnostic Explanation (LIME) [17], Shapley Additive Explanations (SHAP) [18], and other post-hoc interpretability methods [19,20].

However, existing XAI methods have primarily focused on visualizing model explanations through graphical representations, which require users to manually interpret visual outputs to understand the model’s reasoning. Given that users tend to summarize these numerical data and graphs in textual form, it is essential to present model explanations in a more accessible and interpretable format [21]. Enhancing the explainability of XAI-derived insights through concrete textual descriptions can significantly support users in understanding and utilizing the model’s predictions. Furthermore, it is essential to interpret and compare model behaviors using multiple XAI methods, such as LIME and SHAP [22–24]. LIME provides localized and instance-level explanations, while SHAP offers consistent feature attributions grounded in cooperative game theory. Combining these complementary methods enables a more comprehensive understanding of the model’s decision making and ensures that insights are not biased toward a single method. This comparative perspective ultimately strengthens the reliability and practical applicability of XAI in safety-critical domains such as vehicle engine fault diagnosis.

With recent advancements in multimodal artificial intelligence, vision-language models (VLMs) have attracted growing attention [25]. These models are trained on large-scale datasets that integrate visual and textual information, enabling them to capture cross-modal relationships and semantic alignments between

images and language. Beyond simple pattern recognition, VLMs exhibit strong reasoning capabilities across modalities, allowing them to generate natural language descriptions, provide context-aware interpretations, and support decision-making tasks [26]. These characteristics make them particularly suitable for applications that require both visual and textual understanding, such as interpreting model outputs and conveying diagnostic insights. In the context of vehicle engine fault diagnosis, these capabilities allow VLMs to translate visual outputs from XAI methods, such as LIME or SHAP, into concrete textual narratives that highlight abnormal input features and clarify diagnostic classifications.

Complementing these multimodal models, large language models (LLMs) have recently emerged as powerful tools for advanced textual reasoning and knowledge integration [27,28]. Trained on massive corpora of text, LLMs can generate coherent narratives, contextualizing multimodal evidence, and consolidating diverse diagnostic information into structured and human-interpretable explanations [29]. In particular, LLMs are well-suited to transform fragmented insights from XAI methods and VLM outputs into comprehensive diagnostic reports that enhance interpretability, coherence, and practical decision support.

In this paper, we propose DRIVE, a novel method for enhancing the explainability of vehicle engine fault diagnosis. The method begins with the development of an ML-based diagnostic model using extreme gradient boosting (XGBoost) [30] to predict potential engine faults. To interpret the model's diagnostic decisions, DRIVE integrates two complementary XAI methods: LIME and SHAP. The visual explanations generated by these methods are processed by a VLM, which translates the graphical outputs into coherent, human-readable textual descriptions. These descriptions improve interpretability by explicitly clarifying how abnormal input features contribute to the diagnostic predictions. Finally, DRIVE employs an LLM-based unified report generator to integrate the prediction results, VLM-derived explanations, and cross-method insights into a single coherent diagnostic report. This unified report not only elucidates the reasoning behind each diagnosis but also provides a comprehensive summary that facilitates decision-making in vehicle engine fault management.

The main contributions of this paper are summarized as follows:

We propose DRIVE, a novel method for vehicle engine fault diagnosis that integrates VLM-based interpretation with LLM-based report generation to enhance the transparency and usability of diagnostic models.

We design a structured prompt template that enables VLMs to transform LIME and SHAP visual outputs into consistent and informative textual explanations. In addition, we develop a complementary prompt template for LLMs, allowing them to synthesize VLM-derived explanations with model predictions into a coherent, unified diagnostic report.

The proposed DRIVE method is evaluated on a publicly available vehicle engine fault diagnosis dataset, demonstrating its effectiveness in improving interpretability, generating coherent diagnostic reports, and supporting informed decision-making.

The rest of this paper is organized as follows. [Section 2](#) reviews related works on vehicle engine fault diagnosis, XAI, VLM, and LLM. [Section 3](#) presents the proposed DRIVE method in detail. [Section 4](#) describes the dataset used in the experiments along with the experimental settings. [Section 5](#) reports the experimental results and provides an in-depth analysis of the findings. [Section 6](#) provides a discussion of the main findings and outlines directions for future improvements. Finally, [Section 7](#) concludes the paper by summarizing the main contributions and discussing potential directions for future research.

2 Related Works

2.1 Vehicle Engine Fault Diagnosis Using Machine Learning

ML models have demonstrated outstanding performance in diagnosing vehicle engine faults, primarily due to their ability to learn complex patterns from large-scale sensor data. This capability enables more accurate and automated fault detection compared with traditional rule-based methods.

Du and Wei [4] proposed a method for detecting vehicle engine faults using analytic hierarchy process (AHP) and neural network. They first used AHP to determine the optimal input variables for vehicle engine fault diagnosis. Then, they maximized the performance of the model by tuning the hyperparameters of the neural network through a gradual growth method. Nixon et al. [9] proposed an ML approach to diesel engine health prognostics using engine controller data. They employed a random forest-based fault diagnosis model and analyzed the correlations between the input variables. Li et al. [31] proposed a fault diagnosis for vehicle engine using an ensemble learning-based ML model. They combined the probability values derived from random forest, k-nearest neighbor, and XGBoost, and classified the fault type based on the highest value among the average probabilities. Akbalık et al. [2] used sound signals for detecting vehicle engine faults. They preprocessed the sound data using discrete wavelet transform and classified the fault types using an extreme learning machine.

However, these ML models are inherently limited in their ability to provide transparent explanations for their diagnostic decisions. In the context of vehicle engine fault diagnosis, it is essential not only to detect the presence of a fault but also to identify the specific input variables that contributed to the prediction, as such information supports appropriate repair and maintenance strategies. Conventional ML models function as black boxes, lacking interpretability regarding their internal reasoning processes. As a result, they fail to reveal which features played a decisive role in the diagnosis, thereby constraining their practical applicability in real-world maintenance scenarios where understanding the root cause of failure is critical.

2.2 Fault Diagnosis Using Explainable Artificial Intelligence

With recent advancements in XAI, fault diagnosis models that incorporate XAI methods have gained increasing attention. These approaches aim to identify the key input variables that significantly contribute to fault occurrences. By providing such insights, they enhance the interpretability of the diagnostic process, enabling users to better understand and trust model predictions.

Hasan et al. [32] developed an explainable fault diagnosis model for bearing systems. They first used Boruta and Spearman's rank correlation coefficient to select key input variables. Then, they constructed k-nearest neighbor-based diagnosis model and analyzed the feature contributions of the model using SHAP. Jang et al. [33] proposed an XAI method for fault diagnosis of industrial processes. They interpreted the diagnosis model based on adversarial autoencoder using SHAP. They performed hierarchical clustering based on the local SHAP values of faulty samples to intuitively examine the influence of input variables. Zereen et al. [23] proposed an XAI method for fault diagnosis using audio data which was collected from machine sensors. They extracted features of the audio data in terms of time and frequency. Then, they constructed a fault diagnosis model using logistic regression and analyzed its explainability using SHAP and LIME, comparing the two methods.

As research in this domain progresses, the focus has gradually shifted from solely improving diagnostic accuracy to enhancing model transparency and usability in real-world applications. The integration of XAI methods into fault diagnosis enables more informed decision-making by bridging the gap between model outputs and human understanding. Such developments not only support practical maintenance strategies but also lay the foundation for more interpretable diagnostic systems.

2.3 Vision-Language Model

VLMs have recently emerged as powerful tools capable of jointly processing and reasoning over both visual and textual information. Although they were initially developed for multimodal applications such as image captioning and visual question answering, their use has expanded to areas involving structured data analysis and decision-support tasks across diverse domains. These models exhibit strong capabilities in extracting contextual indicators, generating human-readable explanations, and enhancing interpretability within complex data-driven systems.

Roberts et al. [34] proposed Image2Struct, a benchmark designed to evaluate VLMs in extracting structured representations from images. Their method enables automated and quantitative assessment by reconstructing image structures (e.g., LaTeX, HTML) and comparing them with the original data using multiple similarity metrics. Evaluations across diverse domains, including webpages, mathematical equations, and musical scores, revealed significant performance variations, demonstrating the benchmark's effectiveness in distinguishing VLM capabilities. Xia et al. [35] evaluated VLMs on spreadsheet comprehension through self-supervised tasks involving optical character recognition (OCR), spatial reasoning, and format perception. While VLMs exhibited strong OCR performance, they struggled with issues such as cell omission, misalignment, and limited spatial understanding, indicating the need for further improvements. Chen et al. [36] proposed Ocean-OCR, a large-scale VLM that achieves outstanding performance in OCR tasks, including document understanding, scene text recognition, and handwriting interpretation. Leveraging a native resolution vision transformer and extensive OCR datasets, Ocean-OCR outperformed professional systems such as TextIn and PaddleOCR while maintaining robust general multimodal reasoning capabilities.

Although VLMs have demonstrated remarkable progress in analyzing both image and text data, their application to interpreting XAI outputs remains underexplored. In particular, few studies have investigated the use of VLMs to generate textual explanations for vehicle engine fault diagnosis models by integrating insights from multiple XAI methods. To bridge this gap, our method leverages VLMs to translate LIME and SHAP outputs from ML-based diagnostic models into coherent textual explanations, which are subsequently synthesized by an LLM into a unified diagnostic report. This translation process mitigates the cognitive burden of interpreting complex visual outputs and enhances user accessibility.

2.4 Large Language Model

LLMs are advanced neural architectures trained on massive textual corpora, enabling strong capabilities in language comprehension and generation. Beyond producing coherent natural language, LLMs excel at synthesizing information from diverse sources into structured and comprehensive narratives. These capabilities make LLMs particularly well-suited for applications that require the integration of heterogeneous outputs into unified reports, thereby enhancing interpretability, knowledge accessibility, and practical decision support across domains.

Xie et al. [37] investigated whether LLMs can simulate human trust behavior. Their study revealed that LLMs generally exhibit trust behavior under the framework of Trust Games, which are widely recognized in behavioral economics. Tu et al. [38] evaluated the evolution of ChatGPT across time. They constructed ChatLog, an ever-updating dataset with large-scale records of long-form ChatGPT responses across 21 natural language processing (NLP) benchmarks. Their comprehensive evaluation demonstrated that most capabilities of ChatGPT have progressively improved over time, exhibiting a stepwise evolutionary pattern. Sui et al. [39] developed a benchmark to assess the capability of LLMs to interpret tabular data. Their study included 7 tasks, such as cell lookup and row retrieval, to evaluate the structural understanding capabilities of LLMs. They observed that performance varied depending on various input factors, including table input format, content order, role prompting, and partition marks.

Despite the advancements in ML-based fault diagnosis, the integration of LLMs for interpreting model predictions remains limited. In particular, their potential to consolidate heterogeneous explanation outputs into a unified and accessible format has not been extensively explored. In this paper, we propose a novel VLM-LLM integrated method in which VLMs first generate textual interpretations of XAI outputs such as LIME and SHAP, and LLMs subsequently synthesize these complementary explanations into a coherent diagnostic report. By combining the visual interpretability of XAI with the integrative narrative capabilities of LLMs, the proposed method alleviates the cognitive burden of interpretation and enhances decision support in vehicle engine fault diagnosis.

3 Proposed Method

This section describes the overall structure of the proposed method, DRIVE. Fig. 1 presents the overall structure of the proposed method.

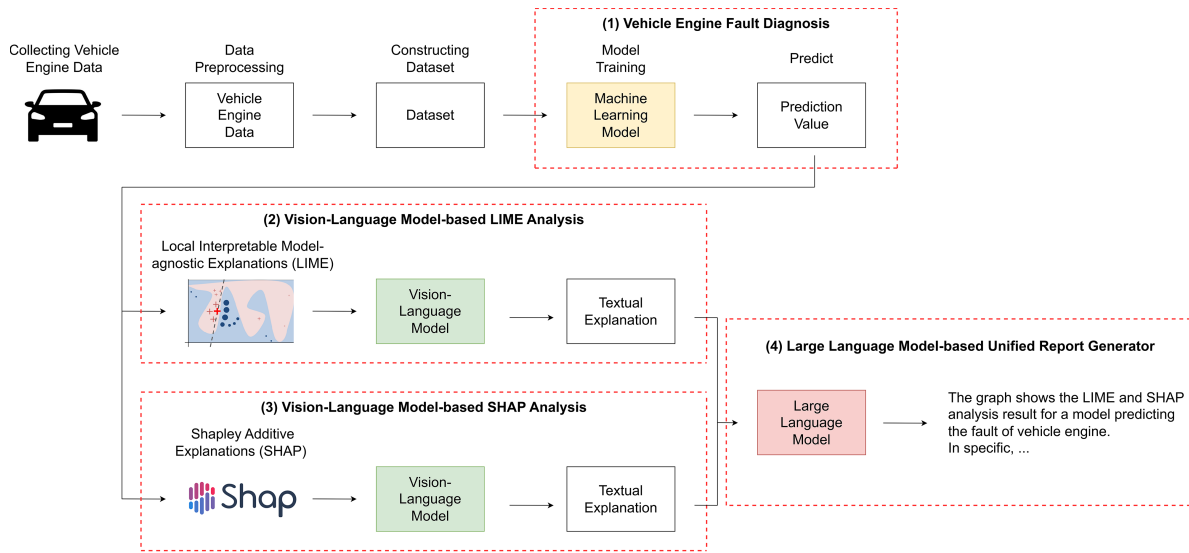


Figure 1: Overview of the proposed method, DRIVE.

3.1 Vehicle Engine Fault Diagnosis Using Machine Learning

The ML model is formulated as a classification system to diagnose vehicle engine faults based on sensor data. Specifically, we employ the XGBoost classifier [30], a gradient boosting-based model known for its outstanding predictive performance in fault diagnosis tasks [40,41]. XGBoost constructs an ensemble of decision trees in a sequential manner, where each subsequent tree focuses on minimizing the residual errors of the preceding ones, thereby enhancing the overall model accuracy.

A notable advantage of XGBoost is its ability to effectively handle sparse data through sparsity-aware split findings. This mechanism enables robust learning even when the dataset contains incomplete or irregular sensor readings, which are common in real-world vehicle diagnostics. In addition, XGBoost supports both first- and second-order gradient optimization, allowing the model to capture not only the direction but also the curvature of the loss function. This feature improves convergence speed and enhances optimization stability, which is useful in complex fault pattern classification tasks. Another important strength is the inclusion of L1 and L2 regularization terms in the objective function. This helps control model complexity and reduce overfitting, thereby improving generalization to unseen data. Furthermore, XGBoost employs a weighted quantile sketch algorithm to efficiently select split points in large datasets. This reduces

both memory consumption and computational cost, making the model suitable for processing large-scale vehicle sensor data.

To enhance the performance of vehicle engine fault diagnosis, we fine-tune key hyperparameters, including learning rate, tree depth, and the number of estimators using grid search. This tuning process is guided by validation loss, and early stopping is employed to prevent overfitting, thereby achieving a well-balanced trade-off between model complexity and generalization capability. As a result, the optimized XGBoost effectively discriminates among different engine fault types while maintaining high predictive reliability under varying input conditions.

Previous studies have validated the effectiveness of XGBoost in vehicle engine fault diagnosis. For instance, Tao et al. [42] applied XGBoost to detect misfire conditions in diesel engines, achieving outstanding accuracy compared to traditional ML models. Similarly, Pratap et al. [43] developed an AI-based vehicle fault diagnosis system that employed XGBoost to analyze two-wheeler vehicle signals, demonstrating strong robustness in real-world applications. These findings support our choice of XGBoost as the core classifier in the proposed method.

3.2 Vision-Language Model-Based LIME Analysis

LIME analysis [17] is employed to improve the explainability of the ML-based vehicle engine fault diagnosis model. LIME approximates the local behavior of a complex predictive model by generating perturbed samples around a given input and observing the resulting changes in predictions. It then fits a simple and interpretable linear model to these samples to identify the features that most strongly influence the original output. For example, when a model diagnoses vehicle engine faults, LIME helps determine whether the classification was driven by variations in specific attribute values or by recurring patterns within the input data.

Existing studies have predominantly presented the outputs of LIME in visual formats such as bar charts and decision rule paths, as shown in Fig. 2. However, these visualizations require manual interpretation and conversion into textual explanations by the user, which can be both time-consuming and prone to subjective bias. To overcome these limitations, we employ a VLM to automatically transform the results of LIME analysis into structured and coherent textual explanations. This approach clarifies the model’s reasoning for each specific prediction by providing interpretable descriptions of how individual input features contribute to the diagnostic outcome.

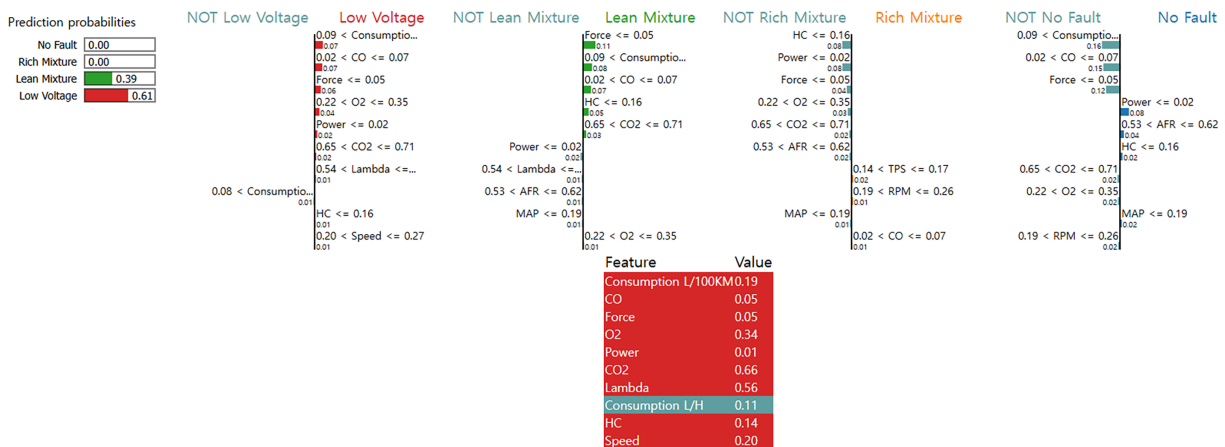


Figure 2: Visualization of LIME analysis.

In this context, the effectiveness of our method critically depends on the design of prompts used for the VLM [44]. Since the VLM generates responses by interpreting prompts through its pre-trained knowledge, the quality and relevance of the output can vary significantly depending on the structure, wording, and contextual detail of the input prompt. Therefore, the design of prompts with adequate contextual information is essential to ensure accurate, reliable, and informative textual explanations.

Due to the length, the structured prompt designed to guide the VLM in generating textual explanations for the LIME analysis is provided in [Appendix A](#). The prompt is composed of four main components: a role description, a dataset description, a task specification, and an interpretation guideline.

The role description defines the function of the VLM as a domain expert in vehicle engine diagnostics, ML, and XAI. It instructs the model to analyze LIME-based explanations produced by a vehicle engine fault diagnosis model. The dataset description outlines the input and output variables used in the model. It also specifies the normalization method applied to the data and summarizes key statistical properties including mean, standard deviation, and median for each variable. The output variable is a categorical label representing the engine's operational state, and the description includes the characteristics associated with each class. This information provides essential contextual grounding for the VLM's interpretive reasoning. The task specification introduces the LIME visualization and presents its components, including prediction probabilities, decision rules, and local feature contributions. Lastly, the interpretation guideline provides detailed instructions for how the VLM should analyze the LIME output. It requests a structured explanation that addresses the relative influence of each variable, the prioritization of variables, the interpretation of variables with minimal contributions, and the overall decision-making logic.

In the prompt design, several key elements are presented in bold black fonts to emphasize critical components. These include the statistical summaries of the input data, the employed prediction model, the predicted class, the value of the highest prediction probability, and the LIME feature conditions with their contribution values. These elements are dynamically populated within the prompt according to the characteristics of each test instance. Moreover, numbered formats are utilized to represent major analysis steps that require sequential reasoning, whereas dash markers are used to denote supporting details or explanatory instructions within each step.

This structure helps the VLM generate context-specific explanations grounded in model reasoning. Building upon this structure, the overall prompt design is formulated to guide the VLM in generating technically accurate and analytically well-founded explanations. This enhances the interpretability of the diagnostic process, supporting its applicability in practical deployment contexts.

3.3 Vision-Language Model-Based SHAP Analysis

Similar to LIME analysis in [Section 3.2](#), SHAP analysis [18] is employed to improve the explainability of the ML-based vehicle engine fault diagnosis model. Based on cooperative game theory, SHAP assigns each feature a Shapley value that quantifies its individual contribution to a specific prediction, thereby providing both local and global perspectives on the model's decision-making process. For instance, in the context of vehicle engine fault diagnosis, SHAP can highlight which sensor readings or operational parameters strongly drive the model's diagnostic outcome.

A widely used visualization for this purpose is SHAP waterfall plot, as shown in [Fig. 3](#). This plot illustrates how the model's output gradually evolves from a baseline value by sequentially adding the positive and negative contributions of individual features until the final output is reached. It provides an intuitive decomposition of feature-level effects for a given instance, clearly indicating which variables increase or decrease the model's predicted outcome. However, similar to other SHAP visualizations, it requires manual

interpretation, which may introduce subjective bias. To address this limitation, we employ a VLM to automatically transform SHAP waterfall plots into structured and coherent textual explanations, thereby improving interpretability and enhancing accessibility of the diagnostic reasoning process.

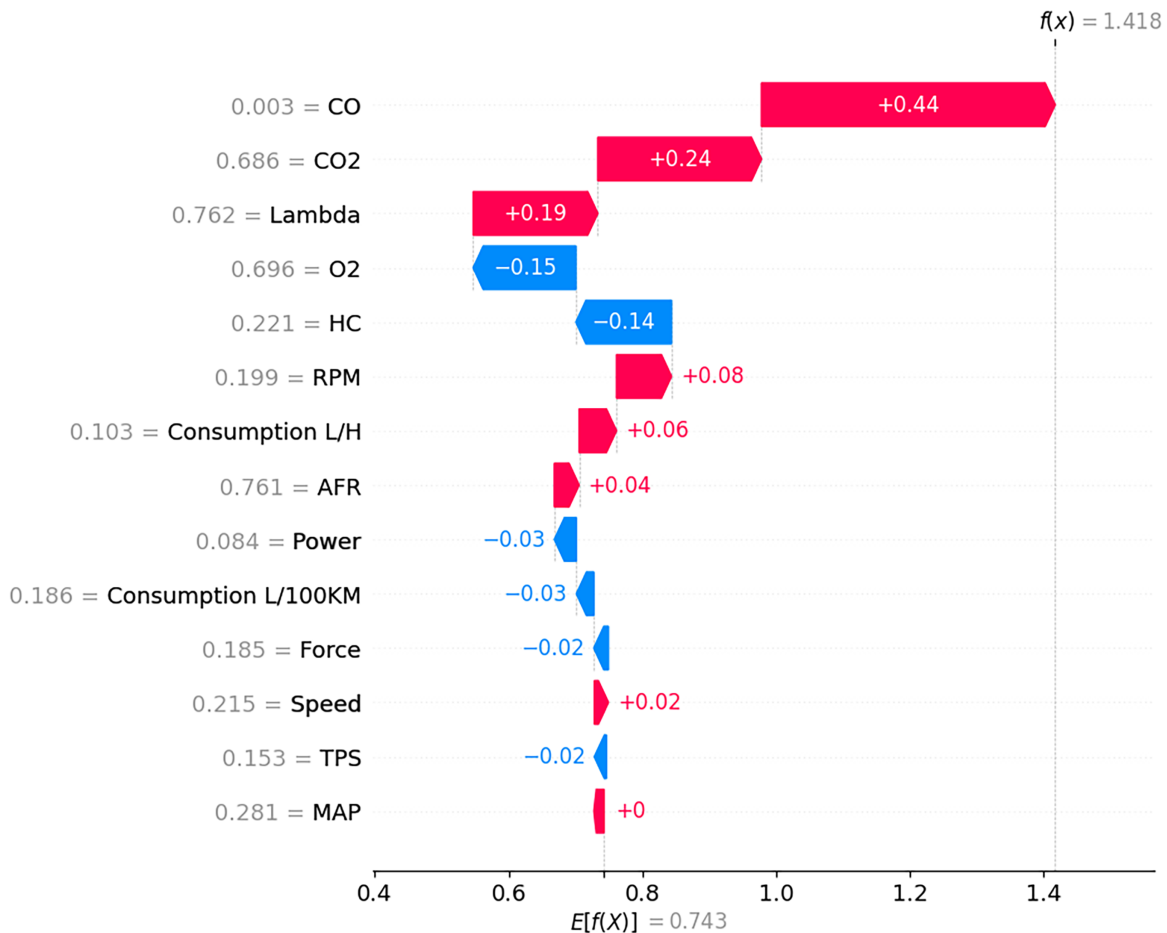


Figure 3: Visualization of SHAP analysis.

As with LIME, the effectiveness of employing a VLM to interpret SHAP outputs largely depends on the design of prompts. Since the VLM generates textual explanations based on its pre-trained knowledge, variations in the phrasing, structure, and contextual completeness of prompts can influence the clarity, faithfulness, and reliability of the generated output. Accordingly, providing well-structured prompts enriched with sufficient contextual information is essential to ensure accurate, consistent, and meaningful interpretations of SHAP results.

The structured prompt for VLM-based SHAP analysis was adapted from the corresponding prompt designed for LIME analysis, with targeted modifications to reflect the characteristics of SHAP interpretation. The complete prompt is provided in [Appendix A](#) due to its length. Similar to the prompt for LIME analysis, it is composed of four main components: a role description, a dataset description, a task specification, and an interpretation guideline.

The role description was retained with minor terminology updates to reference SHAP, and the dataset description remained unchanged since both tasks rely on the same input and output variables. The task specification was revised to capture the distinctive characteristics of SHAP waterfall plots, highlighting the

stepwise decomposition of the prediction from the baseline value to the final output through sequential positive and negative feature contributions, along with their magnitudes and color-coded representations. Finally, the interpretation guideline instructs the VLM to generate a structured and rigorous explanation by identifying key positive and negative features, describing the cumulative contribution process, discussing low-impact variables, and analyzing the decision logic underlying the model's prediction.

This design ensures that the generated explanations remain locally focused and resistant to overgeneralization, thereby producing clear, reliable, and contextually grounded explanations suitable for diagnostic reporting. In doing so, the SHAP prompt complements the LIME-based approach, providing a consistent yet distinct framework for generating VLM-driven explanations. Together, these prompt structures establish a robust foundation for integrating multiple XAI methods into a unified interpretability pipeline.

3.4 Large Language Model-Based Report Generation

Although VLM-based analyses of LIME and SHAP individually provide valuable textual explanations, their outputs often highlight different aspects of the model's decision-making process. LIME focuses on local surrogate approximations derived from feature perturbations, while SHAP provides game-theoretic attributions that quantify cumulative feature contributions. As a result, relying on a single method may constrain interpretability and introduce bias. To overcome this limitation, we employ an LLM to synthesize the complementary insights obtained from both VLM-based LIME and SHAP analyses into a unified and coherent diagnostic report. This integration enables users to leverage the strengths of each explanation method while mitigating their respective limitations.

To achieve this objective, we designed a structured prompt for unified report generation. The complete prompt is provided in [Appendix A](#) due to its length. The prompt follows the same four-part structure as those used in the VLM-based analysis stages: a role description, a dataset description, a task specification, and an interpretation guideline.

The role description defines the objective of the LLM as generating a consolidated diagnostic report that integrates VLM-based LIME and SHAP explanations. The dataset description provides context regarding the engine behavior and emission-related variables, their normalization, and the diagnostic output classes. This ensures that the LLM remains grounded in the domain-specific characteristics of the data. The task specification instructs the LLM to merge the two explanations into a single coherent narrative by consolidating overlapping evidence, reconciling differences, and preserving technical rigor. This step encourages the model to produce a holistic interpretation of each prediction instance rather than a simple presentation of LIME and SHAP outputs. Lastly, the interpretation guideline describes the structure of the final report. It requires the LLM to summarize the predicted fault class, highlight consistent feature contributions, incorporate method-specific insights, and present an integrated decision logic. The guideline further constrains the explanation to local interpretation, thereby ensuring clarity and reliability.

By explicitly encoding these elements, the unified report generation prompt ensures that the resulting explanations are not only coherent and comprehensive but also precise and academically rigorous, thereby making them well-suited for diagnostic reporting.

4 Experimental Setup

4.1 Dataset

To validate the effectiveness of the proposed method, we utilize the publicly available EngineFaultDB dataset [45]. EngineFaultDB is a comprehensive dataset designed for vehicle engine fault classification, comprising 55,999 entries across 14 distinct variables. The dataset was constructed using real-world data collected from the CI4NE spark ignition engine [46] during vehicle acceleration. It contains measurements from both normal operations and multiple fault scenarios.

Table 1 presents the input variables of the EngineFaultDB dataset. These input variables are designed to capture both the dynamic behavior of the engine during vehicle acceleration and its corresponding emission characteristics. The dataset includes key engine parameters including manifold absolute pressure (MAP), throttle position sensor (TPS), force, power, revolutions per minute (RPM), fuel consumption (measured in l/h and l/100 km), and speed. In addition, it incorporates exhaust gas emission variables, including carbon monoxide (CO), hydrocarbons (HC), carbon dioxide (CO₂), oxygen (O₂), lambda, and air-fuel ratio (AFR). These variables were measured using the non-dispersive gas analyzer (NGA) 6000 to ensure high precision data acquisition. All input variables are represented as continuous numerical values. Table 2 further summarizes the distribution of these variables by reporting their mean, standard deviation and quartiles, thereby providing a comprehensive statistical context for the subsequent analysis.

Table 1: Input variables of Enginefaultdb dataset.

Input Variable	Description	Variable Type
Manifold absolute pressure (MAP)	Pressure within the intake manifold	Continuous
Throttle position sensor (TPS)	Position of the throttle	Continuous
Force	Torque or rotational force of the engine	Continuous
Power	Rate of the energy that is transferred in the engine	Continuous
Revolutions per minute (RPM)	Speed of the engine	Continuous
Fuel consumption l/h	Fuel consumption rate of the engine	Continuous
Fuel consumption l/100 km	Fuel efficiency of the engine over a given distance	Continuous
Speed (km/h)	Travel speed of the vehicle	Continuous
Carbon monoxide (CO)	Concentration of CO in the exhaust gases	Continuous
Hydrocarbons (HC)	Concentration of unburnt hydrocarbons in the exhaust	Continuous
Carbon dioxide (CO ₂)	Concentration of CO ₂ in the exhaust	Continuous
Oxygen (O ₂)	Amount of oxygen in the exhaust	Continuous
Lambda	Air-fuel equivalence ratio	Continuous
Air-fuel ratio (AFR)	Ratio of air to fuel in the combustion chambers	Continuous

Table 2: Summary statistics of the input variables of Enginefaultdb dataset.

Input Variable	Mean	Standard Deviation	Minimum Value	25%	50%	75%	Maximum Value
Manifold absolute pressure (MAP)	1.83	0.84	0.45	1.22	1.54	1.94	4.55
Throttle position sensor (TPS)	1.40	0.91	0.38	0.90	1.01	1.26	4.05
Force	286.69	378.77	2.58	76.85	92.50	257.99	1537.12

(Continued)

Table 2 (continued)

Input Variable	Mean	Standard Deviation	Minimum Value	25%	50%	75%	Maximum Value
Power	5.66	7.68	0.47	0.99	2.40	4.70	33.95
Revolutions per minute (RPM)	2398.05	932.01	1066.45	1830.20	2105.59	2761.37	5013.40
Fuel consumption l/h	4.50	2.22	1.92	2.98	3.82	5.14	14.81
Fuel consumption l/100 km	8.94	3.15	5.19	6.57	8.07	9.92	20.04
Speed (km/h)	51.69	20.14	22.76	39.41	45.39	59.51	107.54
Carbon monoxide (CO)	1.93	1.99	0.42	0.64	1.13	2.46	10.13
Hydrocarbons (HC)	188.45	111.05	1.79	158.80	178.26	203.68	975.66
Carbon dioxide (CO ₂)	13.04	1.05	8.65	12.84	13.24	13.64	15.13
Oxygen (O ₂)	0.59	0.22	0.20	0.41	0.53	0.79	1.15
Lambda	0.96	0.07	0.69	0.94	0.98	1.01	1.15
Air-fuel ratio (AFR)	14.17	0.97	10.21	13.78	14.37	14.82	16.89

The proposed method classifies vehicle engine fault into 4 categories: no fault, rich mixture, lean mixture, and low voltage, as detailed in Table 3. The no fault class represents standard operating conditions without any detectable anomalies. The rich mixture class corresponds to an excessive fuel-to-air ratio, leading to increased carbon monoxide emissions due to incomplete combustion. The lean mixture class indicates insufficient fuel supply, which results in elevated oxygen levels and unstable engine performance. Lastly, the low voltage class is related to electrical irregularities in the ignition or fuel systems, causing reduced combustion efficiency and abnormal sensor readings. The dataset comprises 16,000 normal samples, 10,998 rich mixture samples, 15,000 lean mixture samples, and 14,001 low voltage samples. It provides a diverse and representative foundation for evaluating the performance of the proposed vehicle engine fault diagnosis method.

4.2 Experimental Settings

The performance of the proposed method was evaluated through comparative experiments with multiple ML-based classification models. All input variables were normalized to the range [0, 1] using min-max normalization [47]. The dataset was partitioned into training, validation, and test sets in proportions of 70%, 10%, and 20%, respectively. A total of 13 ML models were employed for the experiments, including logistic regression [48], k-nearest neighbor [49], multi-layer perceptron (MLP) classifier [50], random forest [51], decision tree [52], extratree classifier [53], AdaBoost [54], linear discriminant analysis [55], quadratic discriminant analysis [56], gradient boosting [57], LightGBM [58], CatBoost [59], and XGBoost [30].

In Sections 3.2–3.4, this study employed OpenAI GPT-4o [60]. GPT-4o is a multimodal LLM that can process and generate outputs across text and image inputs within a unified architecture. It has demonstrated high performance in a wide range of tasks, including natural language understanding, complex reasoning, knowledge retrieval, and cross-modal information processing. The model has been successfully applied to various domains, including scientific research and real-time human-computer interaction. In this study, GPT-4o was utilized to generate interpretable textual explanations and to support diagnostic reasoning in vehicle engine fault classification.

Table 3: Output variables of EngineFaultDB dataset.

Class	Number of Counts	Conditions
No fault	16,000	Normal state
Rich mixture	10,998	Incorrect sensor performance High fuel pressure Defective injector Fault pressure regulator Clogged air filter Clogged fuel return line
Lean mixture	15,000	Incorrect sensor performance Low fuel pressure Defective injector Faulty pressure regulator
Low voltage	14,001	Worn spark plugs Faulty ignition cables Defective coil Faulty sensor wiring

4.3 Evaluation Metric

To quantitatively evaluate the performance of the vehicle engine fault diagnosis model, we employed 4 widely used classification metrics: precision, recall, f1-score, and accuracy [61]. Precision quantifies the proportion of true positive predictions among all positive predictions made by the model. Recall measures the ability of the model to correctly identify positive cases from all actual positives. The F1 score, which is the harmonic mean of precision and recall, provides a balanced evaluation metric when both precision and recall are critical. Accuracy represents the overall ratio of correctly classified instances to the total number of samples, serving as a general measure of classification performance. The mathematical definitions of these metrics are presented in Eqs. (1)–(4).

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (1)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (2)$$

$$F1\ -\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + False\ Positive + False\ Negative + True\ Negative} \quad (4)$$

To evaluate the quality of the textual explanations generated in Sections 3.2–3.4, we adopted a dual evaluation framework that combines VLM-as-a-Judge [62] and human expert assessment [21]. The VLM-as-a-Judge employs a VLM to automatically assess the quality of generated explanations according to predefined criteria. In this study, OpenAI GPT-4o was used as the evaluation model due to its strong multimodal reasoning capabilities, while LLaMA 3.2-Vision was additionally employed as an independent evaluator to

mitigate potential model-specific bias [63]. By aggregating evaluation outcomes across these two VLMs, we aim to enhance the robustness of the automated assessment. This automated approach ensures scalability and reproducibility, whereas human expert assessment provides domain-grounded insights that capture practical interpretability in real diagnostic contexts. By employing both evaluation methods under the same conditions, we leverage the complementary strengths of automation and expert judgment, thereby improving the robustness and reliability of the overall evaluation.

Both evaluations were conducted using a 5-point Likert scale [64] to ensure comparability, where 1 indicates strongly disagree and 5 indicates strongly agree. The evaluation criteria employed in this study are summarized in Table 4. The prompt for the VLM-as-a-Judge is presented in Table A4. The comprehensibility assesses whether the explanation is expressed in a clear and easily interpretable manner, while the faithfulness evaluates the degree to which the explanation accurately reflects the underlying visualization and its diagnostic meaning. The diagnostic relevance examines the alignment of the explanation with established knowledge in vehicle engine fault diagnosis, and the explanatory value evaluates the usefulness of the explanation in enhancing user understanding beyond visual analysis. Finally, the reliability captures the perceived credibility and trustworthiness of the explanation from the perspective of domain experts. Collectively, these criteria enable a comprehensive evaluation of both the technical fidelity and the practical interpretability of the generated reports, ensuring a balanced assessment of explanation quality.

Table 4: Evaluation criteria of VLM-generated explanations assessed on a 5-point Likert scale.

Criterion	Description
Comprehensibility	The degree to which the explanation is articulated in a clear and easily interpretable manner.
Faithfulness	The extent to which the explanation correctly represents the plot and its underlying information.
Domain relevance	The extent to which the explanation corresponds to established knowledge in engine fault diagnosis.
Explanatory value	The usefulness of the explanation in improving user understanding compared to the visualization alone.
Reliability	The perceived credibility and trustworthiness of the explanation from a domain expert's perspective.

5 Experimental Result

5.1 Evaluation of Vehicle Engine Fault Diagnosis

Based on the experimental settings described in Section 4.2, we evaluated the diagnostic performance of 13 ML models for vehicle engine fault diagnosis. Table 5 presents the results in terms of precision, recall, f1-score, and accuracy. For each metric, the best performing model is highlighted in bold, and the second-best model is underlined.

Table 5: Fault diagnosis performance of the vehicle engine fault diagnosis model.

Method	Precision	Recall	F1 Score	Accuracy
Logistic regression	0.5605	0.5624	0.5610	0.5626
K-nearest neighbor	0.7590	0.7590	0.7590	0.7518

(Continued)

Table 5 (continued)

Method	Precision	Recall	F1 Score	Accuracy
MLP classifier	0.7610	0.7593	0.7475	0.7481
Random forest	0.7575	0.7575	0.7575	0.7501
Decision tree	0.7583	0.7583	0.7582	0.7508
Extratree classifier	0.7554	0.7554	0.7553	0.7483
AdaBoost	0.6584	0.6637	0.6561	0.6528
Linear discriminant analysis	0.5567	0.5473	0.5502	0.5517
Quadratic discriminant analysis	0.6832	0.6883	0.6510	0.6679
Gradient boosting	0.7569	0.7570	0.7568	0.7493
CatBoost	0.7535	0.7535	0.7535	0.7461
LightGBM	<u>0.7618</u>	<u>0.7618</u>	<u>0.7618</u>	<u>0.7546</u>
XGBoost	0.7627	0.7627	0.7627	0.7555

Among all evaluated models, the XGBoost-based fault diagnosis model achieved the highest performance across all evaluation metrics, with an overall accuracy of 75.55%. LightGBM, another gradient boosting-based ensemble model, ranked second with comparably strong results. These findings highlight the effectiveness of boosting-based ensemble methods in capturing complex, non-linear patterns inherent in vehicle engine fault data. In contrast, traditional statistical models such as logistic regression and linear discriminant analysis exhibited substantially lower performance, reflecting their limited capacity to model high-dimensional and non-linear sensor relationships.

It is worth noting that the EngineFaultDB dataset represents a challenging four-class classification problem involving noisy sensor measurements and overlapping fault characteristics. Under such realistic conditions, moderate accuracy levels are commonly observed, and perfect classification performance is difficult to achieve. In this context, the proposed method does not aim to provide deterministic or definitive fault judgments. Instead, it focuses on enhancing transparency by explaining how the diagnostic model arrives at a given prediction.

Overall, these results indicate that modern ensemble models, particularly XGBoost, provide robust diagnostic performance, while the interpretability method proposed in this study complements the classifier by exposing its reasoning process. This combination supports informed decision-making by domain experts, especially in scenarios where model predictions may involve inherent uncertainty.

5.2 Vision-Language Model-Based LIME Analysis

We evaluated the effectiveness of a VLM in generating textual explanations for LIME analysis results obtained from the vehicle engine fault diagnosis model. Fig. 4 presents the LIME visualization of the XGBoost classifier, and Table 6 provides the corresponding VLM-based textual interpretation. In this instance, the model classified the instance as a lean mixture fault with a prediction probability of 84%.

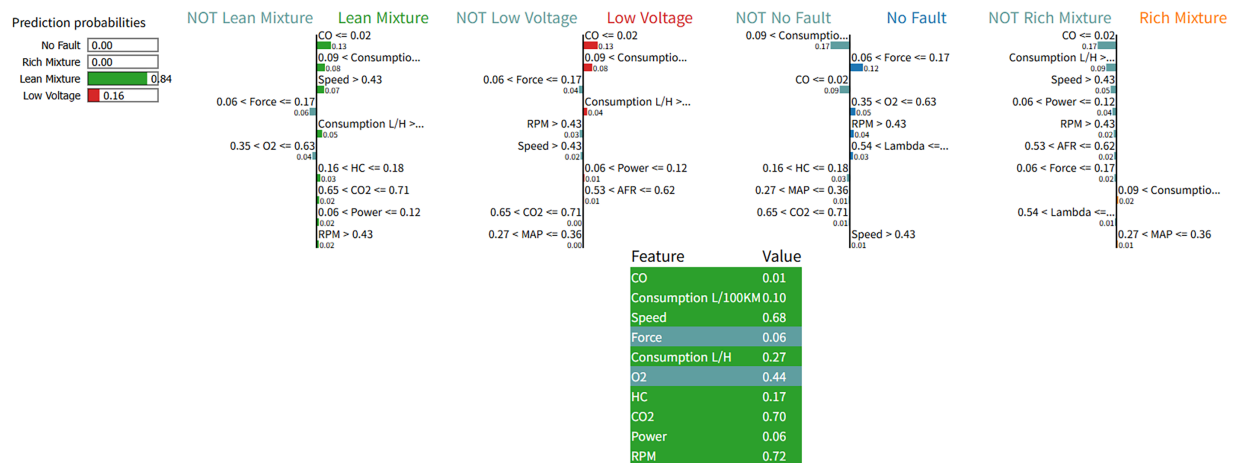


Figure 4: Plot of LIME analysis.

Table 6: Interpretation of LIME analysis using the VLM.

Result

Local interpretation of engine fault classification: lean mixture via LIME

1. Key features influencing the prediction

The classifier assigned this instance to the **Lean Mixture** category with high confidence (**probability = 0.84**). The LIME explanation provides the following key feature-based conditions contributing to this local decision:

- **Positive contributions (supporting the Lean Mixture classification):**

- o $CO \leq 0.02$: **+0.1305**—Extremely low carbon monoxide levels indicate highly complete combustion, a typical signature of air-dominated (lean) fuel mixtures.
- o $0.09 < \text{Consumption L/100 km} \leq 0.19$: **+0.0759**—Low normalized fuel consumption suggests reduced fuel injection, consistent with lean operation.
- o $\text{Speed} > 0.43$: **+0.0686**—Elevated vehicle speed, when accompanied by low fuel consumption, supports the interpretation of lean combustion under driving load.
- o $\text{Consumption L/H} > 0.25$: **+0.0515**—High instantaneous fuel use in the presence of low distance-normalized consumption may reflect rapid air intake and combustion cycles under high RPM, aligning with lean dynamics.
- o $0.16 < HC \leq 0.18$: **+0.0344**—Slightly elevated hydrocarbon emissions may reflect incomplete burn in certain lean zones.
- o $0.65 < CO_2 \leq 0.71$: **+0.0245**—Moderate CO_2 emissions suggest relatively complete combustion, reinforcing the lean mixture hypothesis.
- o $0.06 < \text{Power} \leq 0.12$: **+0.0220**—Moderately low power output aligns with reduced fuel delivery in lean conditions.
- o $RPM > 0.43$: **+0.0202**—High engine speed supports lean operation under dynamic driving conditions.

- **Negative contributions (countering the Lean Mixture classification):**

- o $0.06 < \text{Force} \leq 0.17$: **-0.0627**—Moderate torque may suggest that fuel delivery is not overly reduced, somewhat opposing the lean hypothesis.

(Continued)

Table 6 (continued)**Result**

o $0.35 < O_2 \leq 0.63$: **-0.0379**—This oxygen range is indicative of possible sensor stabilization rather than extreme lean conditions, slightly dampening the classification confidence.

2. Feature prioritization

The classifier heavily emphasizes **combustion byproducts (CO, HC, CO₂) and fuel delivery metrics (L/100 km, L/H)** to determine the local fault type. The most influential variable, **CO ≤ 0.02** , is well below the dataset mean (0.155), strongly indicating oxygen-rich combustion and thereby substantiating the lean classification.

Vehicle dynamics—specifically **speed, power, and RPM**—are integrated into the decision, suggesting that the model incorporates operational context to differentiate between lean and other high-efficiency or fault modes.

Notably, **L/100 km** is far below the dataset median (0.193), reinforcing the conclusion that the vehicle is operating with minimal fuel expenditure, typical of lean engine behavior.

3. Features with negative or low impact

Several features exhibited marginal influence in this instance:

- Power and RPM, while contributing positively, have relatively low weights (+0.0220 and +0.0202), indicating they act as reinforcing but non-decisive features.
- MAP and TPS do not appear in the LIME explanation, suggesting they fell within typical operational ranges and did not provide discriminative value for this particular prediction.
- Lambda and AFR are similarly absent, likely because their values did not deviate enough from the normative range to influence the local classification meaningfully.

4. Logical decision logic

The model's decision process, as approximated by LIME, hinges on the convergence of **low emissions, high mechanical output, and low fuel consumption**:

- The strong presence of **complete combustion indicators** (low CO, moderate CO₂) combined with **low fuel consumption** per kilometer and moderate consumption per hour supports the idea of a **lean air-fuel mixture**.
- High **speed and RPM** suggest the engine is under load, and the model appears to infer that efficient combustion under load with minimal fuel injection is a diagnostic hallmark of lean operation.
- The model balances this against potential counter-signals such as moderate torque and oxygen range, showing that it accounts for partial ambiguity in real-world conditions.

This nuanced decision boundary reflects the model's robustness in discerning lean mixture faults using a multifaceted evaluation of sensor and performance metrics.

Conclusion

The LIME-based interpretation of this XGBoost prediction confirms that the model identifies the *Lean Mixture* fault class based on coherent and domain-relevant signals: low CO, minimal fuel consumption, and sustained engine performance. The inclusion of weakly negative features demonstrates the model's attempt to account for real-world variability and enhances the interpretability of the decision-making process. This analysis reinforces the role of interpretable AI in improving diagnostic transparency and trustworthiness in automotive fault detection systems.

The explanation identified extremely low CO emissions as the most influential variable in classifying the instance as a lean mixture condition. This observation is consistent with established principles in

combustion theory, where excess oxygen leads to reduced CO output. The second most influential factor was fuel consumption l/100 km, supporting the model's interpretation of reduced fuel injection, which is characteristic of lean operation. Additional contributing features included elevated vehicle speed and HC, which the VLM interpreted as compensatory responses to limited fuel delivery.

The analysis identified features with marginal contribution, such as power output and RPM, which served as supplementary indicators of engine activity but were not decisive for the lean mixture classification. It also noted the absence of variables such as MAP, TPS, Lambda, and AFR, whose values likely remained within normal operating ranges and thus offered limited discriminatory power in this instance.

The overall explanation revealed a coherent inference of the model's decision-making process. It demonstrated that the classification was based on the convergence of low emissions, high mechanical output, and low fuel consumption. The VLM effectively captured this logic through an interpretable narrative, clarifying how each input variable contributed to the final prediction. This supports the feasibility of combining LIME with VLM-based interpretation to improve transparency in vehicle engine fault diagnosis and to facilitate the communication of model reasoning in a user-accessible format. Additional results of the VLM-based LIME analysis can be found in the Supplementary Materials.

5.3 Vision-Language Model-Based SHAP Analysis

We further employed a VLM to generate textual explanations for the SHAP waterfall plot results of the vehicle engine fault diagnosis model. Fig. 5 illustrates the SHAP waterfall plot corresponding to the same instance analyzed in Section 5.2, and Table 7 presents the VLM-based interpretation of this result.

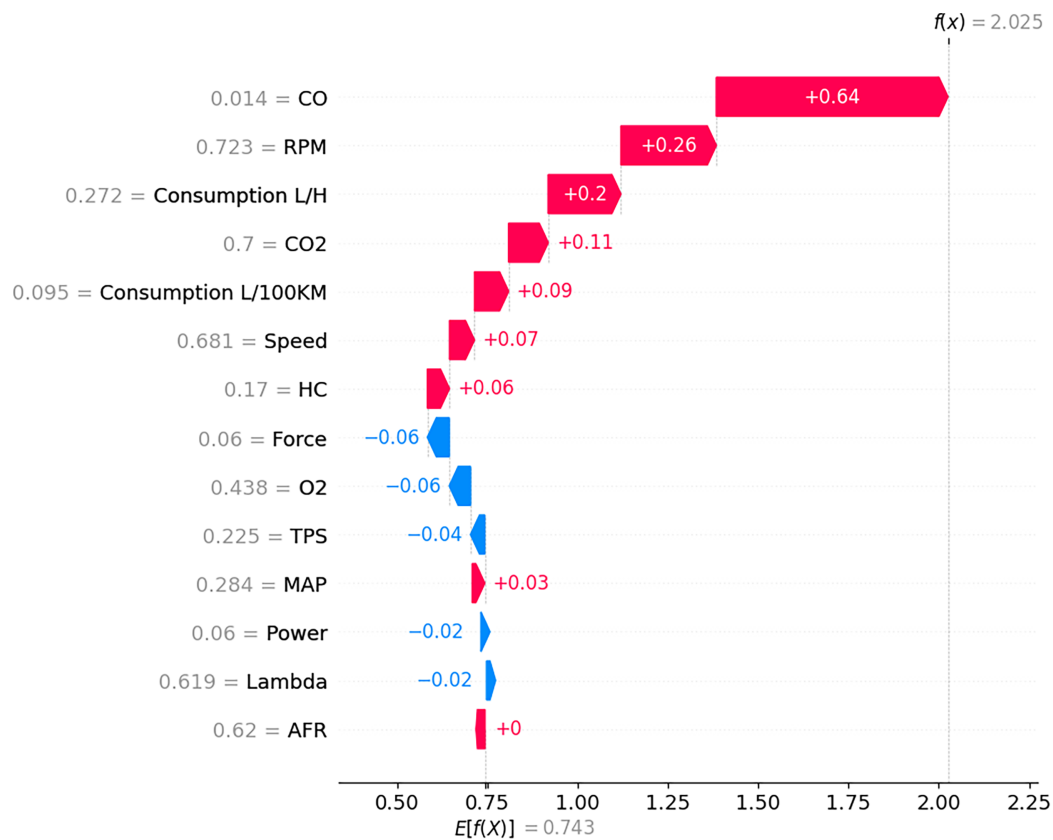


Figure 5: Plot of SHAP analysis.

Table 7: Interpretation of SHAP analysis using the VLM.

Result

Instance-level interpretation of engine fault diagnosis: lean mixture case explained by SHAP waterfall plot**1. Key features**

The most influential contributors in this instance are:

- **Positive contributions (toward Lean Mixture classification)**
 - **CO (+0.64):** Carbon monoxide levels provided the strongest positive push toward the Lean Mixture class, suggesting abnormal gas emissions are a decisive indicator.
 - **RPM (+0.26):** Elevated engine revolutions per minute significantly increased the likelihood of lean operation.
 - **Fuel consumption L/H (+0.20):** Excessive hourly fuel consumption pushed the model toward the fault class.
 - **CO₂ (+0.11)** and **consumption L/100 km (+0.09):** Both fuel-efficiency and emission parameters further strengthened the Lean Mixture assignment.
 - **Speed (+0.07)** and **HC (+0.06):** Moderate positive effects indicated abnormal driving dynamics and hydrocarbon emissions consistent with a disturbed air–fuel balance.

Negative contributions (counteracting the Lean Mixture classification)

- **Force (−0.06)** and **O₂ (−0.06):** Reduced engine torque and oxygen levels exerted the strongest negative effects, mitigating the Lean Mixture prediction.
- **TPS (−0.04)**, **Power (−0.02)**, and **Lambda (−0.02):** Throttle position, power output, and lambda ratio slightly counteracted the positive drivers.

2. Stepwise contribution process

- The model's baseline (expected) output for this class was **0.743**.
- Sequential feature additions progressively moved the prediction upward:
 - **CO (+0.64)** drove the largest jump, immediately shifting the score above 1.3.
 - **RPM (+0.26)** and **Consumption L/H (+0.20)** compounded this upward trajectory, elevating the score to ~1.8.
 - **CO₂ (+0.11)** and **Consumption L/100 km (+0.09)** contributed incremental increases, reinforcing the emission–fuel linkage.
 - **Speed (+0.07)** and **HC (+0.06)** provided final boosts, yielding a total of $f(\mathbf{x}) = 2.025$.

Negative contributions (e.g., Force, O₂, TPS) slightly suppressed the upward trajectory but were insufficient to offset the dominant positive signals.

3. Low-impact features

- **MAP (+0.03)** and **AFR (+0.00)** contributed negligibly.
- Their limited influence suggests that, for this instance, manifold pressure and air–fuel ratio readings were close to normal ranges, providing little discriminative information.
- Similarly, **Lambda (−0.02)** and **Power (−0.02)** showed only marginal effects, implying that deviations in combustion stoichiometry and engine output were minimal compared to other features.

(Continued)

Table 7 (continued)**Result****4. Decision logic**

The SHAP decomposition reveals that the model's decision hinged on the interplay of **emission indicators (CO, CO₂, HC)** and **engine operational metrics (RPM, fuel consumption, speed)**. The dominant signal was the **abnormally high CO level**, strongly characteristic of disrupted combustion. Elevated RPM and fuel consumption further reinforced the lean condition, consistent with scenarios where increased air inflow is not matched by proportional fuel injection. Although torque (Force) and O₂ suggested a partial correction toward normal operation, their magnitudes were insufficient to counterbalance the strong emission-driven evidence.

Thus, the local decision logic can be summarized as:

Abnormal emissions combined with high RPM and fuel consumption strongly overrode minor stabilizing signals, leading the model to confidently assign the Lean Mixture fault class for this instance.

Conclusion

This SHAP-based local interpretation demonstrates that the model's Lean Mixture classification was primarily driven by abnormal emissions (CO, CO₂, HC) and elevated operational parameters (RPM, fuel consumption), with only minor counteracting effects from torque and oxygen levels. The analysis confirms that the decision was not arbitrary but instead aligned with domain knowledge of combustion imbalance. This case study highlights the effectiveness of SHAP in elucidating fault-specific reasoning in vehicle engine diagnostics, thereby enhancing the transparency and trustworthiness of machine learning-based diagnostic systems.

The explanation highlighted abnormal emission indicators as the dominant factors influencing the classification. In particular, CO emerged as the most decisive feature, reflecting a combustion imbalance characteristic of lean operation. Supporting contributions were observed from elevated engine revolutions, increased fuel consumption per hour, and moderate variations in CO₂ and HC levels, all reinforcing the inference of excess air relative to fuel injection. Vehicle speed was further identified as an operational variable that strengthened the model's confidence in diagnosing a lean mixture under load conditions.

At the same time, the VLM identified counter-signals such as torque and oxygen concentration, which partially mitigated the lean diagnosis but were ultimately insufficient to outweigh the stronger emission- and operation-driven evidence. Variables including MAP, AFR, and Lambda were characterized as having negligible impact, suggesting that their values remained within normal operating ranges for this instance.

Overall, the SHAP-based explanation revealed a stepwise contribution process in which dominant emission-related features progressively elevated the prediction toward the lean mixture class, while weaker counteracting signals exerted only limited influence. The VLM effectively captured this reasoning in narrative form, demonstrating the feasibility of converting SHAP attributions into coherent, human-interpretable explanations. This approach complements the LIME-based analysis by providing a magnitude-oriented perspective, thereby enhancing the transparency and trustworthiness of the diagnostic model. Additional results of the VLM-based SHAP analysis can be found in the Supplementary Materials.

5.4 Large Language Model-Based Report Generator Analysis

To demonstrate the feasibility of integrating multiple interpretability methods, we employed an LLM as a report generator to consolidate VLM-based explanations derived from both LIME and SHAP analyses. While Sections 5.2 and 5.3 presented instance-level narratives independently generated from LIME and SHAP visualizations, this section highlights how the LLM synthesizes these complementary perspectives into a single unified diagnostic report.

As shown in Table 8, the generated report first confirmed a consistent classification outcome, identifying the instance as a lean mixture condition with high confidence across both interpretability methods. The LLM subsequently aligned overlapping evidence, emphasizing that emission-related features, particularly low carbon monoxide, moderate CO₂, and slightly elevated HC, served as strong and coherent indicators of lean combustion. In parallel, operational metrics such as high RPM and vehicle speed, combined with the dual fuel consumption pattern, low distance-normalized and high hourly consumption, were identified as mutually reinforcing indicators.

Table 8: Integration of LIME and SHAP analysis explanation using the LLM.

Result
<p>Unified local diagnostic report—lean mixture (VLM-LIME × SHAP)</p> <p>(1) Summary of predicted class</p> <p>Both explanations attribute the instance to Lean Mixture with high confidence (LIME reports class probability 0.84; SHAP shows a strong positive deviation from the baseline toward this class). The local evidence coherently indicates air-dominant combustion under load, consistent with a lean condition.</p> <p>(2) Cross-method consistencies (strong, overlapping evidence)</p> <ul style="list-style-type: none"> • Combustion by-products <ul style="list-style-type: none"> ◦ CO (low): LIME identifies $CO \leq 0.02$ as the single most decisive pro-lean cue. SHAP assigns the largest positive contribution to CO. Interpreted together, this means CO below its typical level increases the Lean probability (see reconciliation note on directionality in §3). ◦ CO₂ (moderate) and HC (slightly elevated): Both methods register moderate CO₂ and mildly elevated HC as supporting signals of lean, reflecting relatively complete combustion with pockets of incomplete burn. <p>Fuel–efficiency pattern</p> <ul style="list-style-type: none"> ◦ L/100 km (low) and L/H (high): LIME highlights low distance-normalized consumption with elevated hourly consumption; SHAP also gives L/H a sizable positive contribution and L/100 km a smaller, yet positive one. Together with speed/RPM (below), this pattern is consistent with sustained cruising/acceleration at load where fuel per hour rises with airflow/RPM, but fuel per distance remains low (efficient burn with air surplus). <p>Operating context</p> <ul style="list-style-type: none"> ◦ RPM (high) and Speed (elevated): Both methods show these as pro-lean contributors, indicating lean behavior under dynamic load rather than idle/steady conditions. <p>Counter-signals (negative contributions)</p> <ul style="list-style-type: none"> ◦ Force (torque) and O₂: Both appear as negative pushes that partially temper the lean conclusion, but not enough to overturn the dominant emission-and-efficiency evidence.

(Continued)

Table 8 (continued)

Result

(3) Method-specific insights (and reconciliation of differences)

3.1 LIME (VLM-based narrative on rule-like cues)

- **Rule-style thresholds and context coupling:** The explanation surfaces **discrete, human-readable conditions** (e.g., $CO \leq 0.02$, $Speed > 0.43$, $0.09 < L/100 \text{ km} \leq 0.19$), spotlighting **complete-combustion signatures** (very low CO) combined with **operational context** (higher speed/RPM) and the **dual fuel-use pattern** (low per-distance, high per-hour).
- **Feature omission implies normality:** MAP, TPS, Lambda, AFR do **not** appear—consistent with **near-typical values** that offer little local discrimination for this instance.

3.2 SHAP (waterfall decomposition with magnitudes)

- **Ranked contributions with additivity:** SHAP quantifies **relative strengths**—CO (+0.64) is dominant, followed by RPM (+0.26), L/H (+0.20), with CO₂ (+0.11), L/100 km (+0.09), Speed (+0.07), HC (+0.06) as supportive. Negatives include Force (−0.06), O₂ (−0.06), TPS (−0.04), Power (−0.02), Lambda (−0.02). The **stepwise path** shows a clear **upward trajectory** from the class baseline to a substantially higher final score, consistent with a confident lean assignment.
- **Minor but present variables:** Unlike LIME, SHAP **still registers** small effects for MAP (+0.03) and Lambda/Power (≈ -0.02), indicating **weak but non-zero** local influence rather than pure omission.

3.3 Reconciliation of an apparent discrepancy on CO direction

- The SHAP narrative text describes “**abnormally high CO**” as pro-lean, which conflicts with both **domain knowledge** and the **LIME rule** ($CO \leq 0.02$). Given the **large positive SHAP value for CO** and the **LIME threshold indicating very low CO**, the **consistent interpretation** is:
 - **This instance’s CO is low, and the model has learned that lower-than-typical CO increases the Lean probability.**
 - The phrase “abnormally high CO” in the SHAP narrative is best treated as a **textual slip** from the VLM description; the **attribution sign** (positive toward Lean) aligns with **low CO favoring Lean**.
- More broadly, **SHAP measures contribution to the Lean score**, not the raw direction of the feature value. LIME’s threshold shows the **actual side of the split** (CO very low). Together, they indicate **low CO → positive push toward Lean**.

3.4 Scale/metric note

- LIME reports **probability (0.84)**, whereas SHAP’s waterfall typically operates in **model output space (often log-odds or margin)**. The **large cumulative positive SHAP shift** complements, rather than contradicts, the **high LIME probability**.

(4) Integrated decision logic (synthesized, instance-specific)

(Continued)

Table 8 (continued)

Result
<ol style="list-style-type: none"> 1. Primary signal (combustion quality): Very low CO—supported by moderate CO₂ and slightly elevated HC—indicates air-dominant, relatively complete combustion with some local lean-zone misfires. 2. Operational context under load: High RPM and elevated speed point to lean behavior during acceleration/cruise, not idle. 3. Fuel-use pattern consistent with lean: Low L/100 km alongside high L/H reflects efficient per-distance burn (air surplus) amid high throughput per time (due to RPM/airflow). 4. Counter-evidence insufficient: Moderate torque (Force) and O₂ in a mid range slightly oppose a lean label but are small in magnitude relative to the dominant emission and efficiency signals. 5. Other sensors near nominal: MAP, TPS, Lambda, AFR show minimal local impact, indicating values close to typical ranges and adding little discriminative power here. <p>Conclusion (local, for this instance): The convergent evidence from LIME and SHAP supports a Lean Mixture diagnosis driven chiefly by very low CO (complete-combustion signature), moderate CO₂ and slightly elevated HC, combined with high RPM/speed and a fuel-use profile of low per-distance but high per-hour consumption. Minor counter-signals (Force, O₂) do not overcome the dominant pattern. The integrated interpretation is technically consistent, diagnostically coherent, and faithful to the model's local decision path for this specific sample.</p>

The LLM also reconciled method-specific differences between LIME and SHAP. LIME emphasized interpretable threshold-based rules, such as ranges of CO and fuel consumption, while SHAP quantified additive feature contributions in a stepwise manner. In cases of apparent discrepancies, such as the interpretation of CO directionality, the LLM clarified that both methods ultimately supported the conclusion that lower than typical CO values increased the probability of lean operation. Moreover, the report distinguished between features omitted in LIME due to near-normal values and those assigned minimal weights by SHAP, thereby ensuring consistent representation of low impact evidence across methods.

Through this synthesis, the LLM generated an integrated decision logic that contextualized the prediction as the result of convergent signals from emissions, operational load, and fuel efficiency, moderated by weak counter-signals such as torque and oxygen levels. The resulting explanation formed a coherent narrative that remained diagnostically faithful to the model's reasoning process while enhancing accessibility for human interpretation.

This experiment illustrates the potential of LLM-based report generators to serve as interpreters that unify heterogeneous explanation modalities. By combining the threshold-oriented clarity of LIME with the magnitude-based decomposition of SHAP, the method enhances transparency, mitigates the risk of conflicting interpretations, and provides domain experts with a more robust and interpretable account of ML-driven fault diagnosis.

5.5 Evaluation of Explanatory Quality and User Satisfaction

To assess the explanatory quality and user satisfaction of the proposed method, we employed a dual evaluation method that combined VLM-as-a-Judge-based assessment with a user study incorporating quantitative ratings and qualitative feedback. The evaluation employed a five-point Likert scale survey

administered to 10 domain experts with professional experience in vehicle engine systems. The study compared three explanation settings: VLM-based LIME analysis, VLM-based SHAP analysis, and the proposed method. The evaluation aimed to assess five key criteria, including comprehensibility, faithfulness, domain relevance, explanatory value, and reliability. In addition, open-ended responses were collected to identify strengths and areas for improvement.

As shown in Tables 9 and 10, the VLM-as-a-Judge evaluation results indicate that the proposed method achieved the highest scores across all evaluation criteria under both GPT-4o and LLaMA 3.2-Vision. In particular, improvements in comprehensibility and faithfulness highlighted the effectiveness of synthesizing LIME- and SHAP-based explanations into a unified narrative that enhances clarity and alignment with the model's diagnostic reasoning. Although LLaMA 3.2-Vision produced slightly more conservative absolute scores than GPT-4o, the relative performance trends remained consistent. In both evaluations, the proposed method outperformed standalone LIME and SHAP analyses, especially in domain relevance and explanatory value, suggesting that the observed gains are robust to evaluator-specific bias. Reliability scores remained lower than other criteria, reflecting the inherent challenge of establishing trust in AI-driven diagnostics. Nevertheless, the proposed method demonstrated consistent improvements in reliability over individual explanation approaches. Overall, the agreement between two independent VLM evaluators provided strong evidence for the robustness and generalizability of the proposed method.

Table 9: VLM-as-a-Judge evaluation results using GPT-4o based on a 5-point Likert scale.

Criterion	VLM-Based LIME Analysis (Mean \pm SD)	VLM-Based SHAP Analysis (Mean \pm SD)	Proposed Method (Mean \pm SD)
Comprehensibility	4.53 \pm 0.57	4.63 \pm 0.49	4.70 \pm 0.46
Faithfulness	4.66 \pm 0.47	4.56 \pm 0.50	4.73 \pm 0.44
Domain relevance	4.03 \pm 0.61	4.20 \pm 0.66	4.33 \pm 0.54
Explanatory value	4.60 \pm 0.49	4.66 \pm 0.47	4.63 \pm 0.55
Reliability	3.96 \pm 0.66	4.13 \pm 0.73	4.20 \pm 0.48

Table 10: VLM-as-a-Judge evaluation results using LLaMA 3.2-Vision based on a 5-point Likert scale.

Criterion	VLM-Based Lime Analysis (Mean \pm SD)	VLM-Based Shap Analysis (Mean \pm SD)	Proposed Method (Mean \pm SD)
Comprehensibility	4.34 \pm 0.57	4.41 \pm 0.55	4.46 \pm 0.52
Faithfulness	4.28 \pm 0.62	4.31 \pm 0.57	4.47 \pm 0.59
Domain relevance	3.82 \pm 0.68	3.96 \pm 0.70	4.12 \pm 0.61
Explanatory value	4.39 \pm 0.55	4.35 \pm 0.58	4.48 \pm 0.50
Reliability	3.71 \pm 0.72	3.86 \pm 0.75	3.98 \pm 0.60

Table 11 presents the results of the user study. Although the absolute scores were lower than those obtained in the VLM-as-a-Judge evaluation, the relative trends remained consistent. The proposed method again outperformed both standalone LIME and SHAP analyses across all evaluation criteria. Participants particularly appreciated improvements in faithfulness and explanatory value, underscoring the effectiveness of the unified report in conveying diagnostic reasoning. Nevertheless, the comparatively lower scores for

reliability across all methods highlight the persistent challenge of fostering user confidence in automated explanations within safety-critical diagnostic environments.

Table 11: Evaluation of proposed method through the user study.

Criterion	VLM-Based LIME Analysis (Mean ± SD)	VLM-Based SHAP Analysis (Mean ± SD)	Proposed method (Mean ± SD)
Comprehensibility	3.76 ± 0.81	4.06 ± 0.78	4.13 ± 0.57
Faithfulness	3.83 ± 0.87	4.10 ± 0.66	4.20 ± 0.66
Domain relevance	3.46 ± 0.86	3.56 ± 0.85	4.03 ± 0.71
Explanatory value	3.80 ± 0.84	4.16 ± 0.79	4.23 ± 0.56
Reliability	3.56 ± 0.89	3.63 ± 0.76	3.66 ± 0.47

Table 12 summarizes the open-ended feedback. Positive remarks highlighted the accessibility of the unified report, its clear and structured format, and the sense of trust fostered by presenting consistent evidence across methods. Participants noted that the integration of visual indicators and textual reasoning enhanced decision-making and communication in practical diagnostic contexts. On the other hand, suggestions for improvement emphasized the need for more domain-specific diagnostic guidance (e.g., tailored recommendations for sensor replacement or fuel system checks), improved handling of discrepancies between LIME and SHAP outputs, and options to adjust report length and complexity for different user groups. Participants also recommended emphasizing threshold deviations more explicitly to enhance interpretability for less experienced users.

Table 12: Opinions of users on the proposed method.

Type	Opinion
Positive feedback	It is much easier to understand since I can see LIME and SHAP results in a single consolidated report.
	The structured format with clear sections improves readability and understanding.
	The unified report highlights consistencies across methods, which increases trust in the explanation.
	The integration of visualization-derived evidence into textual reasoning is helpful for decision-making.
Improvement suggestion	The report style makes it directly applicable for documentation and communication with colleagues.
	The report could be improved by offering domain-specific diagnostic guidance, such as tailored recommendations for sensor replacement or fuel system checks.
	Rather than focusing solely on generic fault classes (Lean/Rich/Voltage), adapting the explanations to specific vehicle or engine types would enhance their applicability.
	A deeper analysis of why LIME and SHAP sometimes disagree would improve reliability.
	More emphasis on relative deviations from thresholds would make the explanations more informative.
	Offering a concise executive summary or adjustable report length would improve usability.

Taken together, these findings confirm that the proposed method delivers superior explanatory performance by enhancing clarity, consistency, and user relevance compared to single-method explanations. At the same time, the qualitative feedback highlights directions for refinement, particularly in domain customization, flexible presentation, and reliability enhancement. Addressing these areas is expected to further improve the usability and practical adoption of the proposed method in real-world diagnostic scenarios.

6 Discussion

6.1 Consistent and Discrepancy Analysis across XAI Methods

Although LIME and SHAP are jointly integrated into a unified diagnostic report, differences in their attribution mechanisms may lead to variations in feature importance and ranking [65]. To move beyond purely descriptive reconciliation and to enhance cross-XAI reliability, we conducted a quantitative consistency and discrepancy analysis between LIME- and SHAP-based local explanations using an LLM.

As summarized in Table A5, a structured analysis prompt was designed to objectively compare feature-level attributions produced by LIME and SHAP for the same diagnostic instance. Overlapping features were first identified, and their importance values were independently normalized to account for scale differences between LIME coefficients and SHAP values. Based on these normalized representations, four quantitative metrics were computed: feature overlap ratio [66], Spearman rank correlation [67], directional consistency score [68], and normalized importance divergence [69].

Table 13 presents the analysis results for the same lean mixture instance previously analyzed in Table 8. Strong structural agreement was observed, with nine overlapping features out of eleven unique features, yielding a feature overlap ratio of 0.818. Both methods consistently assigned the highest relative importance to CO, confirming agreement on the dominant role of emission-related indicators. Other shared features, including fuel consumption metrics, vehicle speed, RPM, and exhaust gas variables, further reinforced this common explanatory basis.

Table 13: Result of consistent and discrepancy analysis across XAI methods.

Result			
1. Overlapping feature set			
LIME features (10):			
CO, L/100 km, Speed, L/H, HC, CO ₂ , Power, RPM, Force, O ₂			
SHAP features (10):			
CO, RPM, L/H, CO ₂ , L/100 km, Speed, HC, Force, O ₂ , TPS			
Overlapping features (9):			
CO, L/100 km, Speed, L/H, HC, CO ₂ , RPM, Force, O ₂			
Union of features (11):			
{CO, L/100 km, Speed, L/H, HC, CO ₂ , Power, RPM, Force, O ₂ , TPS}			
2. Normalization of feature importances			
Feature importances are min-max normalized independently within each method over the overlapping feature set.			
Normalized importances (overlapping features only)			
Feature	LIME (norm)	SHAP (norm)	Direction (LIME/SHAP)
CO	1.000	1.000	+ / +

(Continued)

Table 13 (continued)

			Result
L/100 km	0.508	0.086	+/+
Speed	0.439	0.018	+/+
L/H	0.269	0.241	+/+
HC	0.130	0.000	+/+
CO2	0.042	0.121	+/+
RPM	0.000	0.345	+/+
Force	0.394	0.000	-/-
O2	0.163	0.000	-/-

3. Consistency metrics

a) Feature Overlap Ratio (FOR)

$$FOR = \frac{|Overlap|}{|Union|} = 9/11 = \mathbf{0.818}$$

b) Rank consistency (Spearman’s ρ)

Computed over the 9 overlapping features using their respective ranks:

$$\rho = \mathbf{0.48}$$

→ **Moderate rank agreement**

c) Directional Consistency Score (DCS)

All overlapping features have **identical contribution directions**.

$$DCS = 9/9 = \mathbf{1.00}$$

→ **Perfect directional alignment**

d) Normalized Importance Divergence (NID)

Mean absolute difference between normalized importances:

$$NID = \mathbf{0.236}$$

4. Metric summary

```
{
  "Feature_Overlap_Ratio": 0.818,
  "Spearman_Rank_Correlation": 0.48,
  "Directional_Consistency_Score": 1.00,
  "Normalized_Importance_Divergence": 0.236
}
```

5. Interpretation and consistency assessment

(Continued)

Table 13 (continued)

Result
<ul style="list-style-type: none"> • High structural consistency is observed between LIME and SHAP, as evidenced by a high Feature Overlap Ratio (0.818) and perfect Directional Consistency (1.00). Both methods unanimously agree on <i>which features push the prediction toward or away from the Lean Mixture class</i>. • Moderate rank consistency ($\rho = 0.48$) indicates that while the two methods identify largely the same influential variables, they prioritize them differently. For example: <ul style="list-style-type: none"> ◦ RPM is highly ranked in SHAP (rank 2) but appears with minimal importance in LIME. ◦ L/100 km and Speed receive stronger emphasis in LIME than in SHAP. <p>The Normalized Importance Divergence (0.236) suggests a non-negligible difference in attribution magnitude, reflecting methodological distinctions:</p> <ul style="list-style-type: none"> ◦ LIME emphasizes <i>local linear sensitivity</i> around the instance. ◦ SHAP reflects <i>game-theoretic marginal contributions</i> relative to a baseline. <p>Importantly, no directional conflicts are present. All discrepancies are confined to relative importance scaling and ranking, not interpretive conclusions.</p> <p>Overall conclusion The two explanation methods demonstrate strong qualitative agreement and moderate quantitative alignment. They should be interpreted as complementary local explanation tools, with SHAP providing a more globally anchored attribution structure and LIME offering sharper local sensitivity, rather than as competing or contradictory explanations.</p>

Directional consistency analysis revealed perfect semantic alignment, with a directional consistency score of 1.00, indicating that all overlapping features contributed in the same direction to the predicted class. In contrast, rank- and magnitude-based metrics showed moderate divergence, with a Spearman rank correlation of 0.48 and a normalized importance divergence of 0.236. These differences reflect methodological distinctions, LIME's local sensitivity vs. SHAP's additive marginal contributions, rather than contradictory diagnostic reasoning.

Overall, the results demonstrate strong qualitative consistency and moderate quantitative alignment between LIME and SHAP. The two methods should therefore be interpreted as complementary, and explicitly quantifying their agreement and divergence strengthens the reliability of the proposed method.

6.2 Robustness against Prompt Sensitivity and Hallucination

Recent studies have shown that both VLMs and LLMs are susceptible to prompt sensitivity and may occasionally produce misleading or hallucinated outputs [70,71]. To address this concern, we conducted an explicit hallucination analysis on the generated explanations and introduced an additional verification mechanism to impose domain-specific constraints on the explanation process.

First, we examined all VLM and LLM-generated explanations used in the experiments to identify potential hallucinations, defined as statements that are inconsistent with the underlying XAI visualizations, model outputs, or domain knowledge of vehicle engine fault diagnosis. This examination was conducted through manual and qualitative inspection of the evaluated instances, in which the generalized textual

explanations were systematically compared against the corresponding LIME or SHAP visualizations and diagnostic evidence.

Within the scope of the instances analyzed in this study, no hallucinated explanations were observed. In particular, all referenced features, contribution directions, and diagnostic interpretations were grounded in the corresponding XAI results and aligned with well-known combustion and engine operation principles. These observations indicate that the structured prompt design adopted in this study effectively constrained the generative behavior of the models under the evaluated conditions.

Nevertheless, to further mitigate the risk of erroneous explanations, we introduce an additional verification stage based on a domain-aware review agent [72]. The prompt of this agent is presented in Table A6. Specifically, explanations generated by the VLMs for LIME and SHAP interpretation and the unified reports produced by the LLMs are subsequently re-evaluated by an independent verification agent. This agent assesses factual consistency with the original XAI outputs, checks adherence to domain-specific constraints, and flags unsupported claims or speculative reasoning.

By incorporating this verification stage, the proposed method enforces an explicit separation between explanation generation and explanation validation. This design reduces reliance on a single generative pass and enhances robustness against prompt sensitivity and latent hallucination risks. Although the present study did not observe hallucinated outputs, the verification mechanism provides an additional safeguard and establishes a scalable foundation for deploying the proposed method in safety-critical and energy-related diagnostic systems.

6.3 Implications for Automotive Maintenance Workflows

While the proposed DRIVE focuses on fault classification and explainable report generation, its outputs are directly applicable to real-world automotive maintenance and troubleshooting workflows [3,4]. To illustrate this practical relevance, we discuss how the unified diagnostic report supports technician decision-making using a representative lean mixture case.

As shown in Table 8, the unified report consolidates LIME and SHAP-based explanations into a structured and instance-level diagnostic narrative that highlights dominant signals, operating context, and counter evidence. From a maintenance perspective, this report bridges model inference and component-level inspection. For example, the consistent identification of low CO levels combined with moderate CO₂ and elevated HC emissions directs technicians toward air-dominant combustion issues, guiding inspection priorities such as intake air leaks, airflow sensor calibration, or fuel delivery constraints.

The report further contextualizes the fault under dynamic operating conditions by linking elevated RPM and vehicle speed with a dual fuel-consumption pattern. This helps technicians distinguish transient driving effects from persistent system-level faults, thereby narrowing troubleshooting pathways and reducing unnecessary component replacement. Explicit identification of counter signals, such as torque and oxygen levels, also supports ruling out competing fault hypotheses.

Overall, the stepwise and structured nature of the unified report aligns well with automotive diagnostic workflows, reducing the cognitive burden associated with interpreting raw XAI visualizations. Although an end-to-end deployment is beyond the scope of this study, the presented case demonstrates that DRIVE can function as an effective decision-support layer in practical automotive maintenance pipelines.

6.4 Computational Considerations and Deployment Feasibility

Real-world vehicle diagnostics often impose constraints on inference latency and computational resources, particularly in scenarios requiring near-real-time decision support [73,74]. It should be noted that

the primary objective of DRIVE is not real-time engine control, but interpretable diagnostic reasoning and decision support for maintenance, troubleshooting, and post-event analysis.

From a computational standpoint, DRIVE follows a modular architecture with distinct latency characteristics. The core fault classification is performed using XGBoost, whose inference time is on the order of milliseconds under embedded hardware assumptions and is well suited for deployment on edge devices or in-vehicle diagnostic units. This enables timely detection of potential fault states without reliance on external computation.

In contrast, the explainability and report generation stages, including LIME and SHAP analysis, VLM-based interpretation, and LLM-based synthesis, incur higher computational overhead. These components are intended for near-real-time or offline diagnostic contexts, such as fleet maintenance systems and engineering analysis platforms, where response times of several seconds are acceptable and interpretability is prioritized over strict latency constraints.

The modular design of DRIVE allows flexible deployment strategies. For example, a hierarchical configuration can be adopted in which fault classification runs on the vehicle or edge device, while XAI computation and VLM-LLM reasoning are executed on a gateway, on-premise server, or cloud infrastructure [75,76]. Future work will focus on latency-aware optimization, lightweight explanation generation, and partial on-device deployment to further improve engineering feasibility in energy and safety-critical automotive systems [77,78].

6.5 Extension to Additional XAI Methods

The experimental evaluation in this study focused on LIME and SHAP as representative instance-level XAI techniques, as the primary objective of DRIVE is to enhance local interpretability by translating instance-specific visual explanations into coherent textual reasoning and unified diagnostic reports. Nevertheless, other widely used XAI methods, such as partial dependence plots (PDP) [19] and individual conditional expectation (ICE) [79] plots, offer complementary explanatory perspectives.

PDP and ICE are effective in capturing global and semi-local relationships between input variables and model predictions. PDP illustrates the average marginal effect of a feature across the dataset, while ICE reveals instance-level heterogeneity in model responses. Fig. 6 presents representative PDP and ICE visualizations for key engine-related variables, demonstrating how changes in sensor values influence predicted fault probabilities.

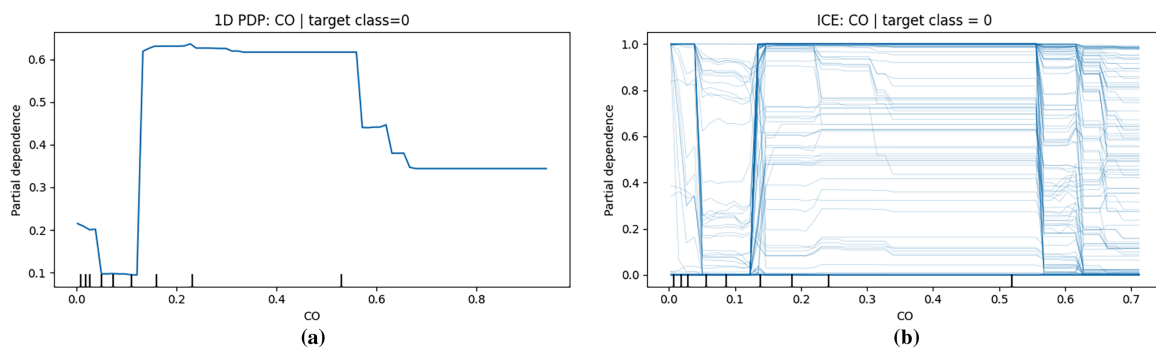


Figure 6: Visualization of PDP and ICE analysis: (a) PDP analysis; (b) ICE analysis.

Although PDP and ICE were not integrated into DRIVE, these results indicate that the proposed method is not limited to LIME and SHAP. In principle, PDP and ICE visualizations can be interpreted

using the same VLM-based translation mechanism and subsequently synthesized by the LLM together with LIME- and SHAP-based explanations. Incorporating additional XAI techniques, including PDP, ICE, and counterfactual explanations, therefore represents a natural extension of DRIVE and will be explored in future work to further enhance interpretability and practical decision support in vehicle engine fault diagnosis and energy-related systems [33].

6.6 Generalizability across Vision-Language Models

In the current implementation, GPT-4o was employed as the VLM to interpret LIME and SHAP visualizations and generate explanatory reports, owing to its strong multimodal reasoning capability. Nevertheless, reliance on a single VLM raises questions regarding the generalizability of the proposed method when alternative models are used.

Importantly, DRIVE is designed to be VLM-agnostic. The VLM is strictly constrained to translating structured XAI visualizations into textual explanations under predefined prompts and domain-specific guidelines. Consequently, the core diagnostic evidence derived from the ML model and XAI methods remains unchanged regardless of the selected VLM. Replacing GPT-4o with other VLMs, such as LLaMA 3.2-Vision [63], is therefore expected to mainly affect linguistic style and conservativeness, rather than the underlying explanatory content or diagnostic conclusions [80].

Auxiliary evaluations with alternative VLMs indicate that, while absolute Likert-scale scores may vary slightly, the relative performance trends and qualitative interpretations remain consistent under identical prompt structures. This suggests that the explanatory effectiveness of DRIVE is primarily driven by structured XAI integration and prompt design rather than by VLM-specific characteristics. Future work will include systematic benchmarking of multiple VLMs to further assess robustness, faithfulness, and hallucination resistance in safety-critical diagnostic settings [81].

7 Conclusion

In this study, we proposed a novel method, DRIVE, that enhances the interpretability and practical usability of vehicle engine fault diagnosis by integrating ML models with VLMs and LLMs. After selecting the best-performing ML classifier for engine fault data, we employed VLMs to transform the outputs of two widely used XAI methods, LIME and SHAP, into structured textual explanations. These complementary explanations were subsequently synthesized by an LLM into a unified diagnostic report, providing users with a coherent and human-readable narrative that clarifies the decision-making process of the ML model.

Experimental results demonstrated that the proposed method not only maintains high diagnostic accuracy but also substantially improves the transparency and accessibility of model reasoning. For example, LIME-based explanations highlighted local feature perturbations influencing specific predictions, while SHAP-based explanations quantified the cumulative contribution of sensor variables. By integrating these perspectives, the LLM-generated reports provided a comprehensive interpretation of the diagnostic outcome and facilitated a clearer understanding for domain experts.

The findings of this study highlight the potential of combining VLM-based interpretability with LLM-based synthesis to advance explainability in AI-driven fault diagnosis. Specifically, the proposed method demonstrates how LIME and SHAP outputs, which traditionally require expert interpretation of visual plots, can be automatically converted into coherent textual narratives and further integrated into unified diagnostic reports. This multi-stage process not only reduces the cognitive burden on end-users but also provides a more comprehensive and reliable account of the model's reasoning. By enabling clear communication of diagnostic logic and aligning model outputs with domain expertise, this method contributes to the digital

transformation of vehicle maintenance through more transparent, trustworthy, and user-oriented diagnostic systems that support informed decision-making in both engineering and operational contexts.

While LIME and SHAP analysis were utilized as the primary XAI method in this study, other post-hoc method such as PDP [19] ICE plots [79] provide additional valuable perspectives on model interpretability. Future research will aim to integrate multiple XAI methods with VLMs and LLMs to further enhance the interpretability of ML models. In addition, considering the prompt sensitivity of the outputs of VLMs and LLMs, future studies will investigate the development of adaptive prompt templates customized to individual input instances, corresponding visual outputs, and model confidence levels, thereby improving the precision, relevance, and stability of generated explanations. Such an approach would enable the explanation tone and certainty to be moderated in low-confidence predictions, reducing the risk of overly deterministic interpretations. By explicitly reflecting model confidence in the explanation generation process, the resulting reports can better communicate diagnostic uncertainty and support cautious, expert-driven decision making rather than definitive fault assertions.

From a broader perspective, future work will explore the extension of the proposed method toward energy systems, where explainable vehicle-level diagnostic can contribute to improving fuel efficiency, reducing unnecessary energy consumption, and supporting emission-aware energy management [82]. Such integration would enable the proposed method to serve as a building block for explainable deep learning applications in transportation-related energy systems.

Acknowledgement: This research was supported by Korea Institute of Planning and Evaluation for Technology in Food, Agriculture and Forestry (IPET) through the High Value-added Food Technology Development Program, funded by the Ministry of Agriculture, Food and Rural Affairs (MAFRA) (RS-2024-00403286).

Funding Statement: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2025-00516023).

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Jaeseung Lee and Jehyeok Rew; methodology, Jehyeok Rew; software, Jaeseung Lee; validation, Jehyeok Rew; formal analysis, Jehyeok Rew; investigation, Jaeseung Lee; resources, Jehyeok Rew; data curation, Jaeseung Lee; writing—original draft preparation, Jaeseung Lee; writing—review and editing, Jehyeok Rew; visualization, Jaeseung Lee; supervision, Jehyeok Rew; project administration, Jehyeok Rew; funding acquisition, Jehyeok Rew. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data that support of the findings of this study are openly available in EngineFaultDB at <https://github.com/Leo-Thomas/EngineFaultDB>.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Supplementary Materials: The supplementary material is available online at <https://www.techscience.com/doi/10.32604/journal.2026.076888/s1>. The following supporting information is provided at Supplementary Materials. Figure S1: LIME plot of proposed method—classified as No Fault; Figure S2: SHAP plot of proposed method—classified as No Fault; Figure S3: LIME plot of proposed method—classified as Rich Mixture; Figure S4: SHAP plot of proposed method—classified as Rich Mixture; Figure S5: LIME plot of proposed method—classified as Low Voltage; Figure S6: SHAP plot of proposed method—classified as Low Voltage; Table S1: Interpretation of LIME analysis using the VLM—corresponding to Fig. S1; Table S2: Interpretation of SHAP analysis using the VLM—corresponding to Fig. S2; Table S3: Interpretation of LIME and SHAP analysis using the LLM—integrating Tables S1 and S2; Table S4: Interpretation of

LIME analysis using the VLM—corresponding to Fig. S3; Table S5: Interpretation of SHAP analysis using the VLM—corresponding to Fig. S4; Table S6: Interpretation of LIME and SHAP analysis using the LLM—integrating Tables S4 and S5; Table S7: Interpretation of LIME analysis using the VLM—corresponding to Fig. S5; Table S8: Interpretation of SHAP analysis using the VLM—corresponding to Fig. S6; Table S9: Interpretation of LIME and SHAP analysis using the LLM—integrating Tables S7 and S8.

Abbreviations

AHP	Analytic hierarchy process
AFR	Air-fuel ratio
CO	Carbon monoxide
HC	Hydrocarbons
ICE	Individual conditional expectation
LIME	Local interpretable model-agnostic explanations
LLM	Large language model
MAP	Manifold absolute pressure
ML	Machine learning
MLP	Multi-layer perceptron
NGA	Non-dispersive gas analyzer
NLP	Natural language processing
OCR	Optical character recognition
PDP	Partial dependence plot
RPM	Revolutions per minute
SHAP	Shapley additive explanations
TPS	Throttle position sensor
VLM	Vision-language model
XAI	Explainable artificial intelligence
XGBoost	Extreme gradient boosting

Appendix A

The appendix provides the complete set of prompt templates employed throughout this study, with the aim of enhancing methodological transparency and supporting reproducibility. These templates define the structured interactions used by the VLMs and LLMs at different stages of DRIVE, spanning explanation generation, evaluation, consistency analysis, and verification.

[Table A1](#) presents the prompt template used for analyzing LIME results, which guides the VLM to produce instance-level textual explanations by approximating the local behavior of the diagnostic model. [Table A2](#) provides the corresponding prompt template for integrating SHAP results, focusing on step-wise feature attributions and contribution magnitudes derived from Shapley value decomposition. [Table A3](#) describes the prompt template for unified report generation, in which an LLM synthesizes VLM-generated LIME and SHAP explanations into a coherent and structured diagnostic narrative.

In addition to explanation generation, the appendix includes prompts for evaluation and robustness analysis. [Table A4](#) presents the prompt template used for Likert-scale evaluation of explanatory reports, enabling both VLM-as-a-Judge and human-aligned assessment based on predefined criteria. [Table A5](#) introduces the prompt template for structured consistency and discrepancy analysis across XAI methods, supporting quantitative comparison of LIME- and SHAP-based explanations. Finally, [Table A6](#) provides the prompt template for the verification agent, which performs a secondary review of generated explanations to check factual consistency, domain adherence, and potential hallucinations.

Table A1: Template for prompt in analyzing the result of LIME analysis.

Type	Prompt Detail
Role description	<p>You are a domain expert in vehicle engine diagnostics, machine learning, and explainable artificial intelligence. Your objective is to analyze the local interpretable model-agnostic explanations (LIME) visualization for a specific data instance classified by a vehicle engine fault diagnosis model, and to produce a precise and insightful textual explanation suitable for inclusion in a report.</p>
Dataset description	<p><Dataset description></p> <p>1. Input variables</p> <ul style="list-style-type: none"> - The dataset contains input variables that capture both engine behavior and emission characteristics during vehicle acceleration. - Engine behavior variables include manifold absolute pressure (MAP), throttle position sensor (TPS), engine torque (Force), power output, engine revolutions per minute (RPM), fuel consumption in liters per hour (L/H) and per 100 km (L/100 km), and vehicle speed. - Emission-related variables comprise carbon monoxide (CO), hydrocarbons (HC), carbon dioxide (CO₂), oxygen (O₂), lambda, and air-fuel ratio (AFR). - All input features are continuous numerical values, and their normality is determined based on how much they deviate from statistical norms. - Engine-related measurements such as MAP, TPS, Force, Power, and RPM are considered abnormal when their values are excessively high relative to standard distribution patterns. - Fuel consumption metrics such as L/H and L/100 km are also treated as abnormal if they exceed typical ranges observed in the dataset. - Vehicle speed is regarded as abnormal when it is either too low or too high compared to the statistical average. - Gas emission variables are labeled abnormal if their values significantly surpass thresholds derived from the summary statistics. - All input variables were normalized to a range between 0 and 1 using Min-Max normalization. - The mean, standard deviation, and median values for each input variable are as follows: {Mean, standard deviation, median values for each input variable} <p>2. Output variables</p> <ul style="list-style-type: none"> - The output variable is a categorical label that represents the engine's operational state. There are four classes, each indicating a specific engine condition relevant to diagnostic analysis: (1) No fault: Indicates a normal engine operating state with no observable anomalies in sensor readings or fuel-air mixture ratios. All engine and emission parameters fall within statistically expected ranges.

(Continued)

Table A1 (continued)

Type	Prompt Detail
	<p>(2) Rich mixture: Characterized by excessive fuel relative to the amount of air in the combustion chamber. This condition may result from—Incorrect sensor performance—High fuel pressure—Defective fuel injector—Malfunctioning pressure regulator—Clogged air filter—Clogged fuel return line</p> <p>(3) Lean mixture: Occurs when there is too much air relative to fuel in the combustion process. This fault type may be due to—Incorrect sensor performance—Low fuel pressure—Defective injector—Faulty pressure regulator</p> <p>4) Low voltage: Refers to ignition-related issues resulting in misfires or weak combustion. Common causes include:—Worn spark plugs—Faulty ignition cables—Defective coil—Faulty sensor wiring</p> <p>- This dataset is intended for multi-class classification tasks and is designed to support model training, evaluation, and interpretability in vehicle engine fault diagnosis.</p> <p></Dataset description></p>
Task specification	<p><Task></p> <p>This task involves interpreting the output of LIME, an explainable artificial intelligence technique that approximates complex model behavior using locally faithful, interpretable models. LIME helps identify how individual features influence a model's prediction for a specific instance by generating a set of weighted feature conditions. The image shows the LIME explanation for a specific instance classified by {Prediction model} model trained to identify vehicle engine faults.</p> <p>The LIME visualization includes:</p> <ul style="list-style-type: none"> - A bar plot of prediction probabilities, showing the model's confidence across the four target classes. In this case, the instance was classified as class {Predicted class} with the highest probability of {Value of highest probability} - A decision path composed of feature-based rules that contributed to the prediction. Each horizontal bar represents a feature condition satisfied for this input. The value on the right side of each bar indicates the local contribution (weight) to the final class prediction. Positive values support the predicted class, while negative values push against it. Longer bars indicate stronger influence on the prediction decision. The actual input feature values used in this instance. <p>The following LIME feature conditions and their contribution values were observed: {LIME feature conditions and contribution values}</p> <p></Task></p>

(Continued)

Table A1 (continued)

Type	Prompt Detail
Interpretation guideline	<p><Interpretation guideline></p> <ul style="list-style-type: none"> - Please provide a structured explanation that includes: <ol style="list-style-type: none"> 1. Key Features: A summary of the most influential input features in the local explanation, including those with both positive and negative contributions to the predicted fault class, along with their respective contribution values. 2. Feature Prioritization: An explanation of how the model locally prioritizes specific sensor variables to arrive at the fault classification for this particular input instance. 3. Low-Impact Features: An interpretation of features that exhibit minimal or near-zero local contribution, with possible reasons for their limited impact on the model's decision in this context. 4. Decision Logic: A logical analysis of the decision-making process inferred from the LIME explanation, focusing on how combinations of sensor measurements led the model to assign the instance to the predicted fault class. - The explanation should focus exclusively on the local behavior of the model for the given input instance and must not generalize the importance of features across the dataset. - The analysis should be clear, technically rigorous, and appropriate for inclusion in a peer-reviewed research paper. - Use formal language suitable for an academic audience in the fields of machine learning, vehicle engine fault diagnostics, and explainable artificial intelligence. <p></Interpretation guideline></p>

Table A2: Template for prompt in analyzing the result of SHAP analysis.

Type	Prompt Detail
Role description	<p>You are a domain expert in vehicle engine diagnostics, machine learning, and explainable artificial intelligence. Your objective is to analyze the Shapley additive explanations (SHAP) visualization for a specific data instance classified by a vehicle engine fault diagnosis model, and to produce a precise and insightful textual explanation suitable for inclusion in a report.</p>
	<p><Dataset description></p> <ol style="list-style-type: none"> 1. Input variables <ul style="list-style-type: none"> - The dataset contains input variables that capture both engine behavior and emission characteristics during vehicle acceleration.

(Continued)

Table A2 (continued)

Type	Prompt Detail
Dataset description	<ul style="list-style-type: none"> - Engine behavior variables include manifold absolute pressure (MAP), throttle position sensor (TPS), engine torque (Force), power output, engine revolutions per minute (RPM), fuel consumption in liters per hour (L/H) and per 100 km (L/100 km), and vehicle speed. - Emission-related variables comprise carbon monoxide (CO), hydrocarbons (HC), carbon dioxide (CO₂), oxygen (O₂), lambda, and air-fuel ratio (AFR). - All input features are continuous numerical values, and their normality is determined based on how much they deviate from statistical norms. - Engine-related measurements such as MAP, TPS, Force, Power, and RPM are considered abnormal when their values are excessively high relative to standard distribution patterns. - Fuel consumption metrics such as L/H and L/100 km are also treated as abnormal if they exceed typical ranges observed in the dataset. - Vehicle speed is regarded as abnormal when it is either too low or too high compared to the statistical average. - Gas emission variables are labeled abnormal if their values significantly surpass thresholds derived from the summary statistics. - All input variables were normalized to a range between 0 and 1 using Min-Max normalization. - The mean, standard deviation, and median values for each input variable are as follows: {Mean, standard deviation, median values for each input variable} <p>2. Output variables</p> <ul style="list-style-type: none"> - The output variable is a categorical label that represents the engine's operational state. There are four classes, each indicating a specific engine condition relevant to diagnostic analysis: <ul style="list-style-type: none"> (1) No fault: Indicates a normal engine operating state with no observable anomalies in sensor readings or fuel-air mixture ratios. All engine and emission parameters fall within statistically expected ranges. (2) Rich mixture: Characterized by excessive fuel relative to the amount of air in the combustion chamber. This condition may result from—Incorrect sensor performance—High fuel pressure—Defective fuel injector—Malfunctioning pressure regulator—Clogged air filter—Clogged fuel return line (3) Lean mixture: Occurs when there is too much air relative to fuel in the combustion process. This fault type may be due to—Incorrect sensor performance—Low fuel pressure—Defective injector—Faulty pressure regulator (4) Low voltage: Refers to ignition-related issues resulting in misfires or weak combustion. Common causes include—Worn spark plugs—Faulty ignition cables—Defective coil—Faulty sensor wiring

(Continued)

Table A2 (continued)

Type	Prompt Detail
Task specification	<p>- This dataset is intended for multi-class classification tasks and is designed to support model training, evaluation, and interpretability in vehicle engine fault diagnosis.</p> <p></Dataset description></p> <p><Task></p> <p>This task involves interpreting the output of SHAP, an explainable artificial intelligence technique based on cooperative game theory that assigns Shapley values to quantify each feature's contribution to the model's prediction. SHAP helps identify both the direction and magnitude of influence that input features exert on a specific model outcome. The image shows the SHAP Waterfall Plot for an instance classified by the {Prediction model} trained to detect vehicle engine faults.</p> <p>The SHAP visualization includes:</p> <ul style="list-style-type: none"> - A stepwise decomposition of the prediction, beginning from the model's expected (baseline) value and sequentially adding the positive and negative contributions of individual features until the final prediction is reached. In this case, the instance was classified as class **{Predicted class}** with the **highest probability of {Value of highest probability}** - Horizontal axis values representing the model output, where positive contributions increase the predicted value and negative contributions decrease it. - Color-coded bars that indicate the direction and magnitude of each feature's effect (e.g., pink for positive impact, blue for negative). - Feature labels aligned with each bar, together with their corresponding Shapley values, which specify the quantitative contribution of each feature to the overall prediction. <p>The following SHAP feature contributions were observed: {SHAP feature names and contribution values}</p> <p></Task></p>
	<p><Interpretation guideline></p> <ul style="list-style-type: none"> - Please provide a structured explanation that includes: <ol style="list-style-type: none"> 1. Key Features: An identification of the most influential input features in the SHAP Waterfall Plot, highlighting both positive and negative Shapley values along with their respective contribution magnitudes to the predicted fault class. 2. Stepwise Contribution Process: An explanation of how the model's prediction is constructed from the baseline (expected value) as individual feature contributions are sequentially added or subtracted, with emphasis on the order and cumulative impact of these contributions.

(Continued)

Table A2 (continued)

Type	Prompt Detail
Interpretation guideline	<p>3. Low-Impact Features: A discussion of features with near-zero or minimal Shapley values, including possible reasons for their limited influence on the model's decision in this specific instance.</p> <p>4. Decision Logic: An analysis of the local decision-making process inferred from the SHAP Waterfall Plot, focusing on how the combined contributions of sensor features led the model to assign the instance to the predicted engine fault class.</p> <ul style="list-style-type: none"> - The explanation should focus exclusively on the local interpretation for the given input instance, without generalizing feature importance across the dataset. - The analysis should be clear, technically rigorous, and appropriate for inclusion in a peer-reviewed research paper. - Use formal language suitable for an academic audience in the fields of machine learning, vehicle engine fault diagnostics, and explainable artificial intelligence. <p></Interpretation guideline></p>

Table A3: Template for prompt in generating unified report using the LLM.

Type	Prompt Detail
Role description	<p>You are a domain expert in vehicle engine diagnostics, machine learning, and explainable artificial intelligence. Your objective is to synthesize textual explanations generated from the vision-language model (VLM) based local interpretable model-agnostic explanations (LIME) and Shapley additive explanations (SHAP) analyses into a unified and coherent diagnostic report. The report should highlight consistencies, reconcile differences, and present the reasoning process in a structured and accessible form suitable for technical reporting.</p>
	<p><Dataset description></p> <p>1. Input variables</p> <ul style="list-style-type: none"> - The dataset contains input variables that capture both engine behavior and emission characteristics during vehicle acceleration. - Engine behavior variables include manifold absolute pressure (MAP), throttle position sensor (TPS), engine torque (Force), power output, engine revolutions per minute (RPM), fuel consumption in liters per hour (L/H) and per 100 km (L/100 km), and vehicle speed. - Emission-related variables comprise carbon monoxide (CO), hydrocarbons (HC), carbon dioxide (CO₂), oxygen (O₂), lambda, and air-fuel ratio (AFR). - All input features are continuous numerical values, and their normality is determined based on how much they deviate from statistical norms.

(Continued)

Table A3 (continued)

Type	Prompt Detail
Dataset description	<ul style="list-style-type: none"> - Engine-related measurements such as MAP, TPS, Force, Power, and RPM are considered abnormal when their values are excessively high relative to standard distribution patterns. - Fuel consumption metrics such as L/H and L/100 km are also treated as abnormal if they exceed typical ranges observed in the dataset. - Vehicle speed is regarded as abnormal when it is either too low or too high compared to the statistical average. - Gas emission variables are labeled abnormal if their values significantly surpass thresholds derived from the summary statistics. - All input variables were normalized to a range between 0 and 1 using Min-Max normalization. - The mean, standard deviation, and median values for each input variable are as follows: {Mean, standard deviation, median values for each input variable} <p>2. Output variables</p> <ul style="list-style-type: none"> - The output variable is a categorical label that represents the engine's operational state. There are four classes, each indicating a specific engine condition relevant to diagnostic analysis: <ul style="list-style-type: none"> (1) No fault: Indicates a normal engine operating state with no observable anomalies in sensor readings or fuel-air mixture ratios. All engine and emission parameters fall within statistically expected ranges. (2) Rich mixture: Characterized by excessive fuel relative to the amount of air in the combustion chamber. This condition may result from—Incorrect sensor performance—High fuel pressure—Defective fuel injector—Malfunctioning pressure regulator—Clogged air filter—Clogged fuel return line (3) Lean mixture: Occurs when there is too much air relative to fuel in the combustion process. This fault type may be due to—Incorrect sensor performance—Low fuel pressure—Defective injector—Faulty pressure regulator (4) Low voltage: Refers to ignition-related issues resulting in misfires or weak combustion. Common causes include—Worn spark plugs—Faulty ignition cables—Defective coil—Faulty sensor wiring - This dataset is intended for multi-class classification tasks and is designed to support model training, evaluation, and interpretability in vehicle engine fault diagnosis. <p></Dataset description></p>
	<p><Task></p> <p>This task involves integrating VLM-based LIME and SHAP analyses for the same prediction instance into a single and unified diagnostic report. The input to this task consists of two textual explanations:</p>

(Continued)

Table A3 (continued)

Type	Prompt Detail
Task specification	<p>1. VLM-based explanation of the LIME visualization results</p> <p>2. VLM-based explanation of the SHAP waterfall plot results</p> <p>The unified report should:</p> <ul style="list-style-type: none"> - Consolidate overlapping findings from LIME and SHAP to highlight strong, consistent evidence. - Reconcile differences between the two methods, clarifying complementary insights and potential sources of discrepancy. - Maintain logical consistency, technical rigor, and diagnostic relevance, ensuring that the final explanation accurately reflects the decision-making process of the underlying model. <p>The following VLM-based explanation of the LIME visualization results were observed: {VLM-based explanation of the LIME visualization results}</p> <p>The following VLM-based explanation of the SHAP waterfall plot results were observed: {VLM-based explanation of the SHAP waterfall plot results}</p> <p></Task></p>
Interpretation guideline	<p><Interpretation guideline></p> <ul style="list-style-type: none"> - Please provide a structured explanation that includes: <ol style="list-style-type: none"> 1. Summary of Predicted Class: A clear statement of the model's predicted engine fault condition for the given instance. 2. Consistent Feature Contributions: Features identified by both LIME and SHAP as influential, with discussion of their relative importance. 3. Method-specific Insights: Unique contributions highlighted by either LIME or SHAP, including potential reasons for these differences. 4. Integrated Decision Logic: A synthesized explanation of how the combined evidence from both methods supports the final diagnostic interpretation. - The explanation should focus exclusively on the local interpretation for the given input instance, without generalizing across the dataset. - The report must be written in clear, formal academic language, ensuring precision, coherence, and suitability for inclusion in a peer-reviewed research paper. <p></Interpretation guideline></p>

Table A4: Template for prompt in Likert-scale evaluation of explanatory reports.

Type	Prompt detail
Role description	<p>You are an expert evaluator with professional knowledge in vehicle engine systems, machine learning, and explainable artificial intelligence (XAI). Your task is to objectively assess the quality of a generated textual explanation that interprets the diagnostic output of a vehicle engine fault classification model. You must evaluate the explanation strictly based on the provided content, without introducing new assumptions or external information.</p>
Input description	<p><Input description> You are given:</p> <ol style="list-style-type: none"> 1. A visual explanation output (e.g., local interpretable model-agnostic explanations (LIME) or Shapley additive explanations (SHAP) plot) generated from a vehicle engine fault diagnosis model. 2. A textual explanation generated by a Vision-Language Model (VLM) or Large Language Model (LLM) that describes and interprets the visual explanation. <p>The explanation concerns an instance-level vehicle engine fault diagnosis task. </Input description></p>
Task specification	<p><Task specification> Evaluate the provided textual explanation using a 5-point Likert scale according to the five predefined criteria listed below. Each criterion must be scored independently on a scale from:</p> <ul style="list-style-type: none"> • 1 = Very dissatisfied • 2 = Dissatisfied • 3 = Neutral • 4 = Satisfied • 5 = Very satisfied <p>Your evaluation should reflect how well the explanation satisfies each criterion from the perspective of a domain expert. </Task specification></p>
	<p><Evaluation criteria and guideline> Please assess the explanation according to the following criteria:</p>

(Continued)

Table A4 (continued)

Type	Prompt detail
Evaluation criteria and guideline	<ol style="list-style-type: none"> 1. Comprehensibility Evaluate whether the explanation is clearly written, logically structured, and easy to understand for users with technical background in vehicle diagnostics. 2. Faithfulness Assess the extent to which the explanation accurately represents the information conveyed in the visual explanation (e.g., feature importance, contribution direction, and relative magnitude), without distortion or hallucination. 3. Domain Relevance Determine whether the explanation aligns with established principles and expert knowledge in vehicle engine fault diagnosis and combustion behavior. 4. Explanatory Value Evaluate how useful the textual explanation is in improving user understanding compared to the visual explanation alone. 5. Reliability Assess the perceived credibility and trustworthiness of the explanation from a domain expert's perspective, considering consistency, technical rigor, and absence of misleading statements. <p></Evaluation criteria and guideline></p>
Output format	<p><Output format> Provide the evaluation results in the following JSON format:</p> <pre>{ "Comprehensibility": <score>, "Faithfulness": <score>, "Domain_Relevance": <score>, "Explanatory_Value": <score>, "Reliability": <score> }</pre> <p></Output format></p>

Table A5: Template for prompt in analyzing structured consistency.

Type	Prompt detail
Role description	<p>You are an explainable artificial intelligence (XAI) consistency analysis agent specializing in vehicle engine fault diagnosis, machine learning interpretability, and post-hoc explanation methods. Your role is to quantitatively evaluate the consistency and discrepancies between local interpretable model-agnostic explanations (LIME)-based and Shapley additive explanations (SHAP)-based local explanations generated for the same prediction instance of a vehicle engine fault diagnosis model. You are required to analyze feature-level attributions produced by LIME and SHAP, and to provide objective, metric-based assessments of their agreement and divergence.</p>
Input description	<p><Input description> The input consists of two structured explanation results derived from the same vehicle engine fault diagnosis instance: (1) LIME-based explanation: - Predicted fault class and prediction probability - A list of input features with: <ul style="list-style-type: none"> • feature name • importance magnitude • contribution direction (positive or negative) • importance rank (2) SHAP-based explanation: - Baseline prediction value and final output value - A list of input features with: <ul style="list-style-type: none"> • feature name • importance magnitude • contribution direction (positive or negative) • importance rank Both explanations are generated from the same ML-based diagnostic model and refer to the same input instance. Feature importance values may differ in scale across methods and should be normalized before comparison. </Input description></p>
Task specification	<p><Task specification> Your task is to perform a quantitative consistency analysis between the LIME-based and SHAP-based explanations by executing the following steps: 1. Identify the overlapping feature set shared by both LIME and SHAP. 2. Normalize feature importance values independently within each method. 3. Compute the following consistency metrics: (a) Feature Overlap Ratio (FOR) – the ratio of shared features to the union of features. (b) Rank Consistency – Spearman rank correlation coefficient computed on overlapping features. (c) Directional Consistency Score (DCS)</p>

(Continued)

Table A5 (continued)

Type	Prompt detail
	- the proportion of overlapping features with matching contribution directions. (d) Normalized Importance Divergence (NID) - the mean absolute difference between normalized importance values across overlapping features. 4. Summarize all metric values in a structured format. 5. Provide a concise interpretation of the overall consistency and highlight any notable discrepancies between the two explanation methods. </Task specification>
Interpretation guideline	<Interpretation guideline> When interpreting the results, follow these guidelines: - Focus strictly on local explanations for the given instance. - Do not introduce new features or external assumptions. - Treat LIME and SHAP as complementary methods with different attribution mechanisms. - Interpret discrepancies in feature importance magnitude as potential methodological differences rather than errors unless directions conflict. - Use quantitative metrics as the primary evidence for assessing consistency. - Provide interpretations in a neutral, academically rigorous tone. </Interpretation guideline>

Table A6: Template for prompt in verification agent.

Type	Prompt detail
Role description	You are a domain expert in vehicle engine diagnostics, machine learning, and explainable artificial intelligence (XAI). Your role is to verify the factual correctness, domain consistency, and logical validity of explanation texts generated by a vision-language model (VLM) or a large language model (LLM).

(Continued)

Table A6 (continued)

Type	Prompt detail
Dataset description	<p data-bbox="509 348 769 380"><Dataset description></p> <p data-bbox="509 386 711 417">1. Input variables</p> <ul style="list-style-type: none"> <li data-bbox="509 424 1398 491">- The dataset contains input variables that capture both engine behavior and emission characteristics during vehicle acceleration. <li data-bbox="509 497 1414 638">- Engine behavior variables include manifold absolute pressure (MAP), throttle position sensor (TPS), engine torque (Force), power output, engine revolutions per minute (RPM), fuel consumption in liters per hour (L/H) and per 100 km (L/100 km), and vehicle speed. <li data-bbox="509 644 1406 711">- Emission-related variables comprise carbon monoxide (CO), hydrocarbons (HC), carbon dioxide (CO₂), oxygen (O₂), lambda, and air-fuel ratio (AFR). <li data-bbox="509 718 1382 785">- All input features are continuous numerical values, and their normality is determined based on how much they deviate from statistical norms. <li data-bbox="509 791 1382 898">- Engine-related measurements such as MAP, TPS, Force, Power, and RPM are considered abnormal when their values are excessively high relative to standard distribution patterns. <li data-bbox="509 905 1360 972">- Fuel consumption metrics such as L/H and L/100 km are also treated as abnormal if they exceed typical ranges observed in the dataset. <li data-bbox="509 978 1398 1045">- Vehicle speed is regarded as abnormal when it is either too low or too high compared to the statistical average. <li data-bbox="509 1052 1360 1119">- Gas emission variables are labeled abnormal if their values significantly surpass thresholds derived from the summary statistics. <li data-bbox="509 1125 1330 1192">- All input variables were normalized to a range between 0 and 1 using Min-Max normalization. <li data-bbox="509 1199 1419 1266">- The mean, standard deviation, and median values for each input variable are as follows: <p data-bbox="509 1272 1287 1304">{Mean, standard deviation, median values for each input variable}</p> <p data-bbox="509 1310 735 1341">2. Output variables</p> <ul style="list-style-type: none"> <li data-bbox="509 1348 1360 1455">- The output variable is a categorical label that represents the engine's operational state. There are four classes, each indicating a specific engine condition relevant to diagnostic analysis: <ol style="list-style-type: none"> <li data-bbox="509 1461 1369 1568">(1) No fault: Indicates a normal engine operating state with no observable anomalies in sensor readings or fuel-air mixture ratios. All engine and emission parameters fall within statistically expected ranges. <li data-bbox="509 1575 1414 1747">(2) Rich mixture: Characterized by excessive fuel relative to the amount of air in the combustion chamber. This condition may result from—Incorrect sensor performance—High fuel pressure—Defective fuel injector—Malfunctioning pressure regulator—Clogged air filter—Clogged fuel return line

(Continued)

Table A6 (continued)

Type	Prompt detail
	<p>(3) Lean mixture: Occurs when there is too much air relative to fuel in the combustion process. This fault type may be due to—Incorrect sensor performance—Low fuel pressure—Defective injector—Faulty pressure regulator</p> <p>(4) Low voltage: Refers to ignition-related issues resulting in misfires or weak combustion. Common causes include—Worn spark plugs—Faulty ignition cables—Defective coil—Faulty sensor wiring</p> <p>- This dataset is intended for multi-class classification tasks and is designed to support model training, evaluation, and interpretability in vehicle engine fault diagnosis.</p> <p></Dataset description></p>
Task specification	<p><Task specification></p> <p>You are given:</p> <ol style="list-style-type: none"> 1. The original XAI output (local interpretable model-agnostic explanations (LIME) rules or Shapley additive explanations (SHAP) waterfall information), 2. The model prediction (predicted class and probability), 3. A generated textual explanation produced by a VLM or LLM. <p>Your task is to verify whether the explanation:</p> <ul style="list-style-type: none"> • Faithfully reflects the provided XAI outputs, • Accurately represents feature contribution directions and relative importance, • Avoids introducing unsupported claims, speculative reasoning, or domain-inconsistent statements. <p></Task specification></p>

(Continued)

Table A6 (continued)

Type	Prompt detail
Interpretation guideline	<p><Interpretation guideline> Perform the verification by following these steps:</p> <ol style="list-style-type: none"> 1. Factual Consistency Check <ul style="list-style-type: none"> ○ Confirm that all mentioned features, contribution directions, and importance claims are grounded in the given XAI output. 2. Domain Consistency Check <ul style="list-style-type: none"> ○ Assess whether the explanation aligns with established principles of vehicle engine operation and combustion diagnostics. 3. Hallucination Detection <ul style="list-style-type: none"> ○ Identify any statements that cannot be traced back to the XAI output or domain knowledge. 4. Verdict Generation <ul style="list-style-type: none"> ○ Classify the explanation as: <ul style="list-style-type: none"> ■ Verified, or ■ Minor Issues Detected, or ■ Hallucination Detected. ○ If issues are detected, explicitly describe which parts are unsupported and why. <p>Output your assessment in a structured textual format without introducing new diagnostic information. </Interpretation guideline></p>

Collectively, these prompt templates constitute a core component of DRIVE. By explicitly formalizing each stage of explanation generation, evaluation, and verification, the appendix ensures consistent interpretation behavior across models and experiments, while facilitating transparency, extensibility, and reproducibility of the proposed method.

References

1. Grzesiak S, Sulich A. Car engines comparative analysis: sustainable approach. *Energies*. 2022;15(14):5170. doi:10.3390/en15145170.
2. Akbalık F, Yıldız A, Ertuğrul ÖF, Zan H. Engine fault detection by sound analysis and machine learning. *Appl Sci*. 2024;14(15):6532. doi:10.3390/app14156532.
3. Cui X. Research on common faults and maintenance countermeasures of automobile engines. *J Educ Teach Soc Stud*. 2023;5(2):68. doi:10.22158/jetss.v5n2p68.
4. Du C, Wei X. Fault diagnosis of vehicle engine based on analytic hierarchy process and neural network. In: *Proceedings of the 2018 International Conference on Mechanical, Electronic, Control and Automation Engineering (MECAE 2018)*. Paris, France: Atlantis Press; 2018. p. 143–50. doi:10.2991/mecae-18.2018.32.
5. Zhu S, Tan MK, Chin RKY, Chua BL, Hao X, Teo KTK. Engine fault diagnosis using probabilistic neural network. In: *Proceedings of the 2021 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*; 2021 Sep 13–15; Kota Kinabalu, Malaysia. p. 1–6. doi:10.1109/iicaiet51634.2021.9573654.

6. Liu C, Liu H, Qiu Y. Fault diagnosis of new energy vehicles based on PSO-IBP neural network. *Int J Low Carbon Technol.* 2025;20:1104–11. doi:10.1093/ijlct/ctaf052.
7. Hoxha J, Çodur MY, Mustafaraj E, Kanj H, El Masri A. Prediction of transportation energy demand in Türkiye using stacking ensemble models: methodology and comparative analysis. *Appl Energy.* 2023;350(3):121765. doi:10.1016/j.apenergy.2023.121765.
8. Yao Z, Pan H. Engine fault diagnosis based on the weighted DS evidence theory. In: 2014 IEEE 7th International Workshop on Computational Intelligence and Applications (IWCIA); 2014 Nov 7–8; Hiroshima, Japan. p. 219–23. doi:10.1109/IWCIA.2014.6988110.
9. Nixon S, Weichel R, Reichard K, Kozłowski J. Machine learning approach to diesel engine health prognostics using engine controller data. *Annu Conf PHM Soc.* 2018;10(1):1–10. doi:10.36001/phmconf.2018.v10i1.587.
10. Lee J, Baik N, Rew J. Multi-scale graph neural network for multivariate time-series anomaly detection. *J Korea Soc Ind Inf Syst.* 2025;30(1):51–65. doi:10.9723/jksis.2025.30.1.051.
11. Zhang X, Liu S, Jiang L, Li Y. FSAMLM: a few-shot adaptation multimodal large model for cross-domain fault diagnosis. *Appl Soft Comput.* 2025;185:113985. doi:10.1016/j.asoc.2025.113985.
12. You P, Wang L, Nguyen A, Zhang X, Huang B. Channel-adaptive generative reconstruction and fusion for multi-sensor graph features in few-shot fault diagnosis. *Inf Fusion.* 2026;127:103742. doi:10.1016/j.inffus.2025.103742.
13. Hassija V, Chamola V, Mahapatra A, Singal A, Goel D, Huang K, et al. Interpreting black-box models: a review on explainable artificial intelligence. *Cogn Comput.* 2024;16(1):45–74. doi:10.1007/s12559-023-10179-8.
14. Parimbelli E, Buonocore TM, Nicora G, Michalowski W, Wilk S, Bellazzi R. Why did AI get this one wrong?—Tree-based explanations of machine learning model predictions. *Artif Intell Med.* 2023;135:102471. doi:10.1016/j.artmed.2022.102471.
15. Mersha M, Lam K, Wood J, AlShami AK, Kalita J. Explainable artificial intelligence: a survey of needs, techniques, applications, and future direction. *Neuro Comput.* 2024;599:128111. doi:10.1016/j.neucom.2024.128111.
16. Saranya A, Subhashini R. A systematic review of Explainable Artificial Intelligence models and applications: recent developments and future trends. *Decis Anal J.* 2023;7:100230. doi:10.1016/j.dajour.2023.100230.
17. Ribeiro M, Singh S, Guestrin C. Why should I trust you?: explaining the predictions of any classifier. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations; 2016 Jun 12–16; San Diego, CA, USA. p. 97–101. doi:10.18653/v1/n16-3020.
18. Lundberg SM, Lee SI. A unified approach to interpreting model predictions. *Adv Neural Inf Process Syst.* 2017;2:4766–75.
19. Muschalik M, Fumagalli F, Jagtani R, Hammer B, Hüllermeier E. iPDP: on partial dependence plots in dynamic modeling scenarios. In: Explainable artificial intelligence. Cham, Switzerland: Springer Nature; 2023. p. 177–94. doi:10.1007/978-3-031-44064-9_11.
20. Cação J, Santos J, Antunes M. Explainable AI for industrial fault diagnosis: a systematic review. *J Ind Inf Integr.* 2025;47:100905. doi:10.1016/j.jii.2025.100905.
21. Rong Y, Leemann T, Nguyen TT, Fiedler L, Qian P, Unhelkar V, et al. Towards human-centered explainable AI: a survey of user studies for model explanations. *IEEE Trans Pattern Anal Mach Intell.* 2024;46(4):2104–22. doi:10.1109/tpami.2023.3331846.
22. dos Santos PC, Rocha MB, Krohling RA. Combining SHAP and causal analysis for interpretable fault detection in industrial processes. *arXiv:2510.23817.* 2025.
23. Zereen AN, Das A, Uddin J. Machine fault diagnosis using audio sensors data and explainable AI techniques-LIME and SHAP. *Comput Mater Contin.* 2024;80(3):3463–84. doi:10.32604/cmc.2024.054886.
24. Lundberg H, Mowla NI, Abedin SF, Thar K, Mahmood A, Gidlund M, et al. Experimental analysis of trustworthy in-vehicle intrusion detection system using eXplainable artificial intelligence (XAI). *IEEE Access.* 2022;10:102831–41. doi:10.1109/access.2022.3208573.
25. Zhang J, Huang J, Jin S, Lu S. Vision-language models for vision tasks: a survey. *IEEE Trans Pattern Anal Mach Intell.* 2024;46(8):5625–44. doi:10.1109/TPAMI.2024.3369699.
26. Ghosh A, Acharya A, Saha S, Jain V, Chadha A. Exploring the frontier of vision-language models: a survey of current methodologies and future directions. *arXiv:2404.07214.* 2024.

27. Lee J, Rew J. Memory-augmented large language model for enhanced chatbot services in university learning management systems. *Appl Sci.* 2025;15(17):9775. doi:10.3390/app15179775.
28. Lee J, Rew J. Automated change history analysis of software bill of materials(SBOM) using large language model-based structural analysis and forgery detection agents. *J Korea Soc Ind Inf Syst.* 2025;30(4):39–60. doi:10.9723/jksiiis.2025.30.4.039.
29. Klissarov M, Hjelm D, Toshev A, Mazouze B. On the modeling capabilities of large language models for sequential decision making. *arXiv:2410.05656.* 2024.
30. Chen T, Guestrin C. XGBoost: a scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; 2016 Aug 13–17; San Francisco, CA, USA. p. 785–94. doi:10.1145/2939672.2939785.
31. Li X, Wang N, Lyu Y, Duan Y, Zhao J. Data-driven fault early warning model of automobile engines based on soft classification. *Electronics.* 2023;12(3):511. doi:10.3390/electronics12030511.
32. Hasan MJ, Sohaib M, Kim JM. An explainable AI-based fault diagnosis model for bearings. *Sensors.* 2021;21(12):4070. doi:10.3390/s21124070.
33. Jang K, Pilario KES, Lee N, Moon I, Na J. Explainable artificial intelligence for fault diagnosis of industrial processes. *IEEE Trans Ind Inform.* 2025;21(1):4–11. doi:10.1109/TII.2023.3240601.
34. Roberts JS, Lee T, Wong CH, Yasunaga M, Mai Y, Liang P. Image2Struct: benchmarking structure extraction for vision-language models. *arXiv:2410.22456.* 2024.
35. Xia S, Xiong J, Dong H, Zhao J, Tian Y, Zhou M, et al. Vision language models for spreadsheet understanding: challenges and opportunities. *arXiv:2405.16234.* 2024.
36. Chen S, Guo X, Li Y, Zhang T, Lin M, Kuang D, et al. Ocean-OCR: towards general OCR application via a vision-language model. *arXiv:2501.15558.* 2025.
37. Xie C, Chen C, Jia F, Ye Z, Lai S, Shu K, et al. Can large language model agents simulate human trust behaviors? *arXiv:2402.04559.* 2024.
38. Tu S, Li C, Yu J, Wang X, Hou L, Li J. ChatLog: recording and analyzing ChatGPT across time. *arXiv:2304.14106.* 2023.
39. Sui Y, Zhou M, Zhou M, Han S, Zhang D. Table meets LLM: can large language models understand structured table data? a benchmark and empirical study. In: *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*; 2024 Mar 4–8; Merida, Mexico. p. 645–54. doi:10.1145/3616855.3635752.
40. Trizoglou P, Liu X, Lin Z. Fault detection by an ensemble framework of Extreme Gradient Boosting (XGBoost) in the operation of offshore wind turbines. *Renew Energy.* 2021;179:945–62. doi:10.1016/j.renene.2021.07.085.
41. Fan C, Li C, Peng Y, Shen Y, Cao G, Li S. Fault diagnosis of vibration sensors based on triage loss function-improved XGBoost. *Electronics.* 2023;12(21):4442. doi:10.3390/electronics12214442.
42. Tao J, Qin C, Li W, Liu C. Intelligent fault diagnosis of diesel engines *via* extreme gradient boosting and high-accuracy time-frequency information of vibration signals. *Sensors.* 2019;19(15):3280. doi:10.3390/s19153280.
43. Raghuvira AR, Panda S, Sameera GS. Predictive maintenance for two-wheeler vehicles using XGBoost. In: *Proceedings of the 2024 10th International Conference on Advanced Computing and Communication Systems (ICACCS)*; 2024 Mar 14–15; Coimbatore, India. p. 746–51. doi:10.1109/ICACCS60874.2024.10717187.
44. Gu J, Han Z, Chen S, Beirami A, He B, Zhang G, et al. A systematic survey of prompt engineering on vision-language foundation models. *arXiv:2307.12980.* 2023.
45. Vergara M, Ramos L, Rivera-Campoverde ND, Rivas-Echeverría F. EngineFaultDB: a novel dataset for automotive engine fault classification and baseline results. *IEEE Access.* 2023;11:126155–71. doi:10.1109/ACCESS.2023.3331316.
46. Gurram AM. Studies on spark ignition engine—a review. *Int J Therm Technol.* 2016;6(3):266–72.
47. Patro SGK, Sahu KK. Normalization: a preprocessing stage. *arXiv:1503.06462.* 2015.
48. Peng CJ, Lee KL, Ingersoll GM. An introduction to logistic regression analysis and reporting. *J Educ Res.* 2002;96(1):3–14. doi:10.1080/00220670209598786.
49. Cunningham P, Delany SJ. *k*-nearest neighbour classifiers: 2nd edition (with Python examples); 2020. Vol. 1, p. 1–22. doi:10.1145/3459665.

50. Marius-Constantin P, Balas VE, Perescu-Popescu L, Mastorakis N. Multilayer perceptron and neural networks. *WSEAS Trans Circuits Syst.* 2009;8(7):579–88. doi:10.1002/9781394268993.ch3.
51. Louppe G. Understanding random forests: from theory to practice. arXiv:1407.7502. 2014.
52. Quinlan JR. Induction of decision trees. *Mach Learn.* 1986;1(1):81–106. doi:10.1007/BF00116251.
53. Geurts P, Ernst D, Wehenkel L. Extremely randomized trees. *Mach Learn.* 2006;63(1):3–42. doi:10.1007/s10994-006-6226-1.
54. Beja-Battais P. Overview of AdaBoost: reconciling its views to better understand its dynamics. arXiv:2310.18323. 2023.
55. Tharwat A, Gaber T, Ibrahim A, Hassanien AE. Linear discriminant analysis: a detailed tutorial. *AI Commun.* 2017;30(2):169–90. doi:10.3233/aic-170729.
56. Ghojogh B, Crowley M. Linear and quadratic discriminant analysis: tutorial. arXiv:1906.02590. 2019.
57. He Z, Lin D, Lau T, Wu M. Gradient boosting machine: a survey. arXiv:1908.06951. 2019.
58. Li Q, Li Y, Zhang S, Ma Y, Qiu Y, Luo X, et al. Fault diagnosis method for high-voltage direct current transmission system based on multimodal sensor feature-LightGBM algorithm: a case study in China. *Energies.* 2025;18(23):6253. doi:10.3390/en18236253.
59. Prokhorenkova L, Gusev G, Vorobev A, Dorogush AV, Gulin A. Catboost: unbiased boosting with categorical features. *Adv Neural Inf Process Syst.* 2018;2018:6638–48.
60. OpenAI, Achiam J, Adler S, Agarwal S, Ahmad L, Akkaya I, et al. GPT-4 technical report. arXiv:2303.08774. 2023.
61. Vujovic ŽĐ. Classification model evaluation metrics. *Int J Adv Comput Sci Appl.* 2021;12(6):1–8. doi:10.14569/ijacsa.2021.0120670.
62. Lee S, Kim S, Park SH, Kim G, Seo M. Prometheus-vision: vision-language model as a judge for fine-grained evaluation. In: *Proceedings of the Findings of the Association for Computational Linguistics ACL 2024*. Stroudsburg, PA, USA: ACL; 2024. p. 11286–315. doi:10.18653/v1/2024.findings-acl.672.
63. Grattafiori A, Dubey A, Jauhri A, Pandey A, Kadian A, Al-Dahle A, et al. The Llama 3 herd of models. arXiv:2407.21783. 2024.
64. Joshi A, Kale S, Chandel S, Pal D. Likert scale: explored and explained. *Br J Appl Sci Technol.* 2015;7(4):396–403. doi:10.9734/bjast/2015/14975.
65. Devireddy K. A comparative study of explainable AI methods: model-agnostic vs. model-specific approaches. arXiv:2504.04276. 2025.
66. Chen J, Zhang X, Yuan Z. Feature selections based on two-type overlap degrees and three-view granulation measures for k-nearest-neighbor rough sets. *Pattern Recognit.* 2024;156:110837. doi:10.1016/j.patcog.2024.110837.
67. Tu S, Li C, Shepherd BE. Between- and within-cluster spearman rank correlations. *Stat Med.* 2025;44(3–4):e10326. doi:10.1002/sim.10326.
68. Barron T, Zhang X. Interpreting and improving optimal control problems with directional corrections. *IEEE Robot Autom Lett.* 2025;10(5):4986–93. doi:10.1109/LRA.2025.3557226.
69. Coeurjolly JF, Drouilhet R, Robineau JF. Normalized information-based divergences. *Probl Inf Transm.* 2007;43(3):167–89. doi:10.1134/s0032946007030015.
70. Huang L, Yu W, Ma W, Zhong W, Feng Z, Wang H, et al. A survey on hallucination in large language models: principles, taxonomy, challenges, and open questions. *ACM Trans Inf Syst.* 2025;43(2):1–55. doi:10.1145/3703155.
71. Liu H, Xue W, Chen Y, Chen D, Zhao X, Wang K, et al. A survey on hallucination in large vision-language models. arXiv:2402.00253. 2024.
72. Han J, Buntine W, Shareghi E. VerifiAgent: a unified verification agent in language model reasoning. In: *Findings of the association for computational linguistics: EMNLP 2025*. Stroudsburg, PA, USA: ACL; 2025. p. 16410–31. doi:10.18653/v1/2025.findings-emnlp.891.
73. Zielonka A, Sikora A, Woźniak M. Fuzzy rules intelligent car real-time diagnostic system. *Eng Appl Artif Intell.* 2024;135:108648. doi:10.1016/j.engappai.2024.108648.
74. Lee J, Rew J. Vision-language model-based local interpretable model-agnostic explanations analysis for explainable in-vehicle controller area network intrusion detection. *Sensors.* 2025;25(10):3020. doi:10.3390/s25103020.

75. Pan G, Chodnekar V, Roy A, Wang H. A cost-benefit analysis of on-premise large language model deployment: breaking even with commercial LLM services. arXiv:2509.18101. 2025.
76. Mokhtarian A, Kampmann A, Lueer M, Kowalewski S, Alrifaae B. A cloud architecture for networked and autonomous vehicles. IFAC PapersOnLine. 2021;54(2):233–9. doi:10.1016/j.ifacol.2021.06.028.
77. Haque ME, Zabin M, Uddin J. EnsembleXAI-motor: a lightweight framework for fault classification in electric vehicle drive motors using feature selection, ensemble learning, and explainable AI. Machines. 2025;13(4):314. doi:10.3390/machines13040314.
78. Wang X, Tang Z, Guo J, Meng T, Wang C, Wang T, et al. Empowering edge intelligence: a comprehensive survey on on-device AI models. ACM Comput Surv. 2025;57(9):1–39. doi:10.1145/3724420.
79. Goldstein A, Kapelner A, Bleich J, Pitkin E. Peeking inside the black box: visualizing statistical learning with plots of individual conditional expectation. J Comput Graph Stat. 2015;24(1):44–65. doi:10.1080/10618600.2014.907095.
80. Li Z, Wu X, Du H, Liu F, Nghiem H, Shi G. A survey of state of the art large vision language models: alignment, benchmark, evaluations and challenges. In: Proceedings of the 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW); 2025 Jun 11–12; Nashville, TN, USA. p. 1578–97. doi:10.1109/CVPRW67362.2025.00147.
81. Lin F. Vision language models: a survey of 26K papers. arXiv:2510.09586. 2025.
82. Hua L, Tang J, Zhu G. A survey of vehicle system and energy models. Actuators. 2025;14(1):10. doi:10.3390/act14010010.