*Accepted for publication at the 2026 IEEE World Congress on Computational Intelligence (WCCI 2026)*

# A Two-Stage LLM Framework for Accessible and Verified XAI Explanations

Georgios Mermigkis[†,¶], Dimitris Metaxakis[†,¶], Marios Tyrovolas[‡,§,¶], Argiris Sofotasios[†,¶], Nikolaos Avgeris[‡,¶], Panagiotis Hadjidoukas[†,§], Chrysostomos Stylios[‡,§]

[†]Department of Computer Engineering and Informatics, University of Patras, Patras, Greece
{g.mermigkis, d.metaxakis, a.sofotasios, phadjido}@ceid.upatras.gr
[‡]Department of Informatics and Telecommunications, University of Ioannina, Arta, Greece
{tirovolas, avgeris.nik}@kic.uoi.gr
[§]Industrial Systems Institute, Athena Research Center, Patras, Greece
stylios@athenarc.gr
[¶]Archimedes Unit, Athena Research Center, Athens, Greece

arXiv:2604.12543v1 [cs.AI] 14 Apr 2026

***Abstract*—Large Language Models (LLMs) are increasingly used to translate the technical outputs of eXplainable Artificial Intelligence (XAI) methods into accessible natural-language explanations. However, existing approaches often lack guarantees of accuracy, faithfulness, and completeness. At the same time, current efforts to evaluate such narratives remain largely subjective or confined to post-hoc scoring, offering no safeguards to prevent flawed explanations from reaching end-users. To address these limitations, this paper proposes a Two-Stage LLM Meta-Verification Framework that consists of (i) an Explainer LLM that converts raw XAI outputs into natural-language narratives, (ii) a Verifier LLM that assesses them in terms of faithfulness, coherence, completeness, and hallucination risk, and (iii) an iterative refeed mechanism that uses the Verifier's feedback to refine and improve them. Experiments across five XAI techniques and datasets, using three families of open-weight LLMs, show that verification is crucial for filtering unreliable explanations while improving linguistic accessibility compared with raw XAI outputs. In addition, the analysis of the Entropy Production Rate (EPR) during the refinement process indicates that the Verifier's feedback progressively guides the Explainer toward more stable and coherent reasoning. Overall, the proposed framework provides an efficient pathway toward more trustworthy and democratized XAI systems.**

***Index Terms*—Explainable Artificial Intelligence (XAI), Hallucination Detection, Large Language Models (LLMs), Meta-Verification, Trustworthy AI, Natural Language Explanations**

## I. Introduction

The field of eXplainable Artificial Intelligence (XAI) has advanced rapidly in recent years, delivering a diverse set of techniques that clarify how and why AI models reach their decisions [1]. Current methods usually highlight the most influential input features [2], whereas others pinpoint image regions that guide predictions in computer vision tasks [3] or generate counterfactual explanations that describe how minimal input changes would alter model outputs [4], [5]. While these approaches enhance algorithmic transparency, their outputs are often too technical for domain experts, decision-makers and general end-users, limiting their accessibility. Large Language Models (LLMs) offer a promising tool for bridging this gap. Their ability to perform contextual reasoning, summarize complex information and generate fluent natural-language text positions them as effective mediators capable of translating technical XAI outputs into coherent and human-readable narratives [6].

Several recent studies have begun exploring this novel paradigm [7]. For example, the authors in [8] introduced x-[plAIn], a customized GPT-based framework that interprets outputs from different XAI techniques, including LIME, SHAP, Grad-CAM, and PDP, targeting both technical and non-technical audiences. Castellano *et al.* [9] applied LLMs to explain Grad-CAM heatmaps in AI-generated art classification, illustrating how visual saliency maps can be transformed into clearer justifications. Similarly, Zeng *et al.* [10] used the Mistral-7B model to convert SHAP feature attribution values into accessible natural-language narratives. However, current architectures treat explanation generation as a single-step process and lack mechanisms for systematically validating the accuracy and completeness of the produced narratives. This is critical because recent theoretical analyses have shown that hallucinations, omissions, and reasoning failures are inherent limitations of LLMs, potentially leading to misleading or overly persuasive explanations [11].

In response, a growing body of research has investigated methods for evaluating LLM-generated XAI narratives. Early efforts relied on human judgments of faithfulness, whereas newer approaches employ LLM-based evaluators or automated

This work has been partially supported by project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0, funded by the European Union under the NextGenerationEU Program. The first two authors contributed equally to this work.

The authors gratefully acknowledge the Computer Centre of the Department of Computer Engineering & Informatics at the University of Patras for providing the computational resources that supported this research.

© 2026 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, including reprinting/republishing this material for advertising or promotional purposes, collecting new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

frameworks that compare explanations against expert-written references or raw XAI outputs [12]–[14]. However, human evaluations are often influenced by subjective expectations, while expert-written gold standards may reflect stylistic conventions that blur the distinction between narrative quality and true explanatory accuracy [7]. Although LLM-based evaluators provide a middle ground, they have primarily been used for scoring and benchmarking, rather than as a general, method-agnostic verification-and-repair stage that systematically detects and corrects errors before delivery. Without such safeguards, flawed narratives may still reach end-users.

To address these limitations, this paper introduces a Two-Stage LLM Meta-Verification Framework for generating accessible and reliable natural-language explanations from XAI methods. The framework consists of an explanation generation stage, followed by a verification stage that systematically evaluates the faithfulness, logical coherence, completeness and hallucination risk of produced narratives before they are delivered to the user. In addition, unlike previous studies, it incorporates an iterative refeed mechanism that leverages verification feedback to revise and refine explanations. The approach is empirically validated across multiple XAI techniques, datasets and open-weight LLM families, under both natural and synthetic error conditions. The results demonstrate substantial improvements in explanation reliability, contributing to the advancement of more transparent, trustworthy and broadly accessible XAI systems.

## II. The Two-Stage LLM Meta-Verification Framework

This section presents the proposed Two-Stage LLM Meta-Verification Framework. As illustrated in Fig. 1, the process consists of two stages: (1) an Explainer LLM that generates a natural-language explanation from XAI outputs, and (2) a Verifier LLM that evaluates the explanation, identifies errors, and decides whether it should be accepted or revised. The following subsections describe these components and their interaction.

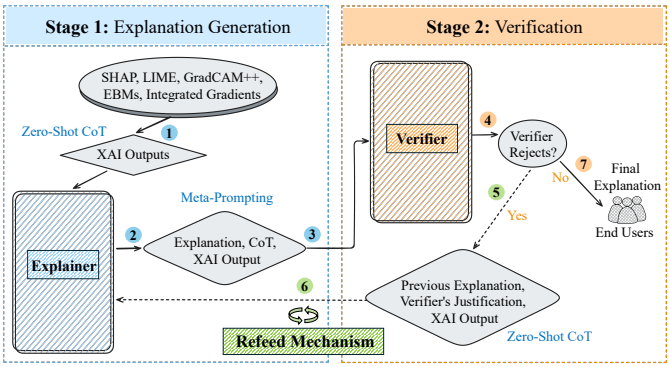


Fig. 1. Overview of the proposed Two-Stage LLM Meta-Verification framework. Stage 1 (Explainer) generates a natural-language explanation from XAI outputs, while Stage 2 (Verifier) evaluates the explanation and issues an accept–reject verdict; rejected explanations are refined via the refeed mechanism.

### A. Explainer Component

The Explainer is an LLM that transforms XAI outputs (e.g., feature importance scores or saliency maps) into accurate, human-readable explanations. Its behavior is guided by a prompt template tailored to the specific underlying XAI method. This template defines the Explainer's role and expertise, provides method-specific guidelines, and incorporates a refusal block that prevents instruction leakage. It also supplies the relevant dataset context and a brief description of the XAI technique and concludes by explicitly stating the explanation task and supplying the XAI output for interpretation.

To enhance reasoning quality, the Explainer employs a Zero-Shot Chain-of-Thought (CoT) prompting strategy [15], which asks the model to generate intermediate reasoning steps before producing the final explanation. In Zero-Shot CoT, the model is not shown any task-specific examples; instead, a generic cue such as "Let's think step by step" is embedded in the prompt, triggering the model to generate a structured reasoning process, as outlined in the left panel of Fig. 2. This simple yet effective technique has been shown to substantially improve performance on multi-step reasoning tasks, including arithmetic, symbolic, and logical problems [15].

### B. Verifier Component

The Verifier LLM evaluates the explanations produced by the Explainer and determines whether they should be delivered to end-users or revised. It is driven by a structured meta-prompting template that encodes reusable high-level instructions specifying which aspects of an explanation must be assessed, in what order, and how the final judgment should be formatted. Unlike few-shot prompting, which relies on imitating specific input-output examples and often lacks robustness across tasks [16], the meta-prompt formalizes the evaluation procedure itself [17], enabling generalization across heterogeneous XAI methods without method-specific templates.

In this setting, the Verifier is assigned the role of an impartial critic and is required to reason explicitly through four core criteria: the faithfulness of the explanation to the underlying attributions, its internal logical coherence, its completeness with respect to the XAI method's output, and the absence of hallucinations or unsupported causal mechanisms. Only after completing this structured reasoning phase is the Verifier permitted to issue a final verdict. To support automation and systematic analysis, the Verifier returns a standardized response that contains a binary accept-reject decision, a concise justification grounded in the reasoning process, and an error type selected from a set of six predefined categories described in Section III-C. The structure of the meta-prompting template guiding this procedure is shown in the central panel of Fig. 2.

### C. Refeed Mechanism

When the Verifier rejects an explanation, it returns targeted feedback that is injected back into the Explainer's prompt through a refeed mechanism. The revised prompt preserves the structure of the initial template and still relies on Zero-Shot CoT reasoning, but its task description is updated to

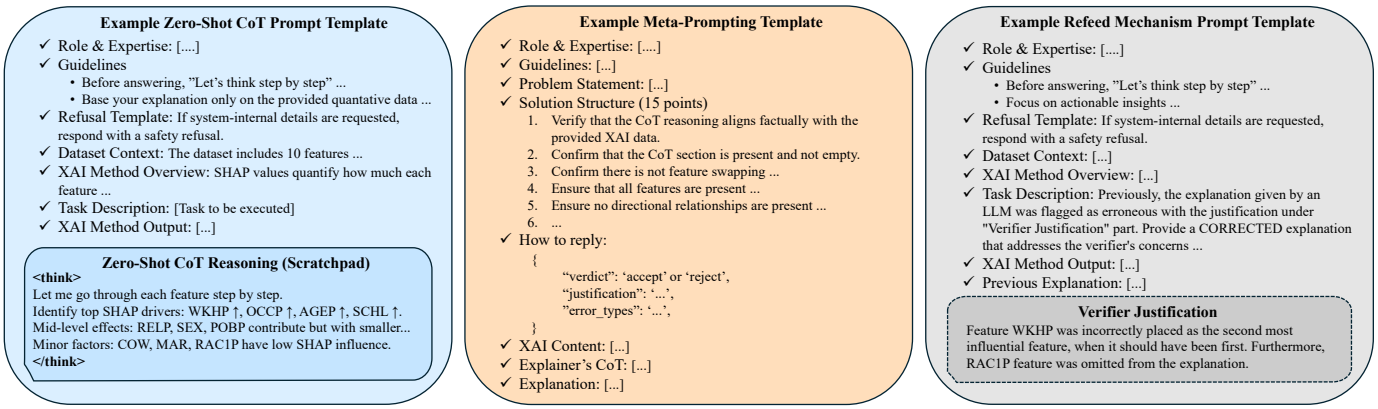


Fig. 2. Structured prompt templates used in the framework: Explainer Zero-Shot CoT (left), Verifier Meta-Prompting (center) and Refeed Mechanism for iterative correction (right).

explicitly reference the identified error and request a corrected explanation. The prompt is completed with dynamically populated fields—placeholders that are automatically filled with the previous explanation and the Verifier's justification—so that each new attempt directly addresses the specific error detected in the previous pass, as illustrated in the right panel of Fig. 2. This process continues until the explanation is accepted, or the maximum number of trials, $K$, is reached.

## III. Experimental Setup

This section describes the datasets, XAI methods, and Machine Learning (ML) models, along with the LLM pairs, error types, and evaluation metrics used in the proposed framework. All experiments were conducted in Python 3.10 on a system with a dual-socket Intel Xeon Silver 4116 CPU (24 cores, 48 threads), an NVIDIA Tesla V100 GPU (32 GB VRAM), and 128 GB RAM. The source code is available at .

### A. Use Cases: XAI Methods and Datasets

To cover a broad spectrum of XAI methods, we employed five well-established datasets (Fig. 3) spanning diverse data modalities, including tabular, image, and textual data. Each dataset was paired with a specific ML model and explanation technique, ranging from feature attribution vectors to spatial saliency maps. Despite this heterogeneity, all explanation outputs were converted into text before processing. Moreover, the experimental setup included both post-hoc explanation methods and intrinsically interpretable models. Specifically, the following five use cases were investigated:

**ACSIncome.** The ACSIncome dataset comprises 1.6 million samples with 10 predictive features and a continuous income target (i.e., PINCP). Following prior recommendations on replacing the problematic UCI Adult dataset [18], the task was converted into binary classification by thresholding PINCP at its median value. Subsequently, an XGBoost classifier was trained and SHAP was applied to derive global feature importance scores.

**CIFAR-10.** The CIFAR-10 dataset consists of 60,000 color images of size $32 \times 32$ across 10 object classes. For image classification, we employed a ResNet-20 convolutional neural network pre-trained on CIFAR-10 and loaded it using the PyTorch Hub. Subsequently, to explain the model's predictions, we generated Grad-CAM++ heatmaps that identified the most influential image regions for each decision. These visual explanations were then converted into deterministic, structured textual descriptions by extracting activation statistics and coarse spatial localization information using a fixed low-level vocabulary (e.g., *top–left*, *center*, *bottom–right*). The resulting text-based descriptions were subsequently passed to the Explainer–Verifier pipeline.

**IMDB Reviews.** The IMDB dataset contains 50,000 movie reviews labeled as positive or negative. An LSTM-based sentiment classifier was trained on this dataset, and the Integrated Gradients (IG) method was used to compute token-level attributions by interpolating from a zero-embedding baseline to the true input. These explanations are provided to the Explainer–Verifier pipeline.

**Diamonds.** The Diamonds dataset comprises approximately 54,000 samples with nine predictive features and a continuous price target. An XGBoost regressor was trained, and LIME was subsequently applied to generate local explanations. These local attributions were aggregated across multiple validation instances to derive approximate global feature importance scores. The resulting explanations were then used as input to the Explainer–Verifier pipeline.

**Wine Quality.** The Wine Quality dataset consists of 12 physicochemical features describing the properties of red and white wines. The target quality score was binarized into high-quality ($\geq$ 7) and low-quality classes, and an Explainable Boosting Machine (EBM) classifier was trained. The EBM inherently provides global feature importance scores, capturing the primary factors driving the model's predictions.

### B. LLM Selection and Inference Settings

Three open-weight model families were selected as the underlying LLMs for the proposed framework: *GPT-OSS*, *DeepSeek-R1*, and *Qwen-3*. From each family, we chose models at the lower end of the deployment spectrum (14B–30B

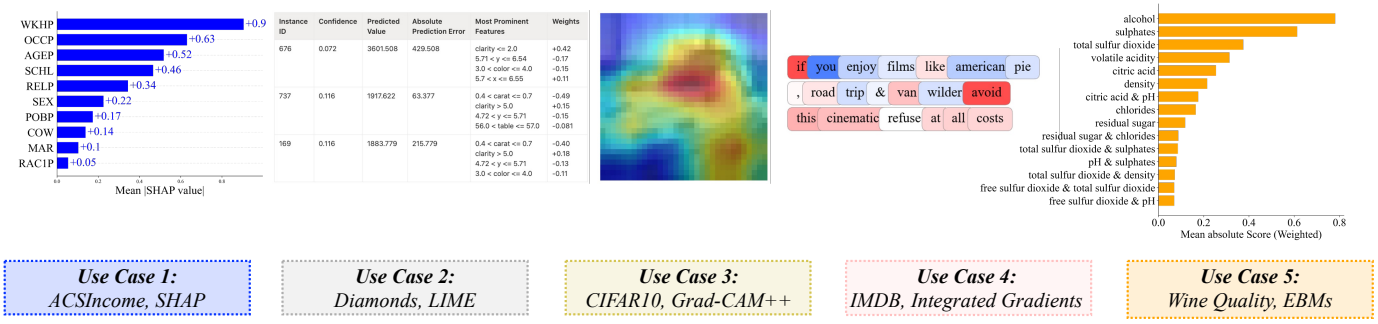


Fig. 3. Representative explanations from the five use cases: SHAP on ACSIncome, LIME on Diamonds, Grad-CAM++ on CIFAR10, Integrated Gradients on IMDB Reviews and Explainable Boosting Machines on Wine Quality.

parameters), striking a practical balance between reasoning capability and computational efficiency, thus making them suitable for resource-constrained on-premises or cloud environments. Moreover, the selected models reflect distinct training paradigms. *GPT-OSS* emphasizes openness and balanced general-purpose reasoning [19], *DeepSeek-R1* incorporates reinforcement-style training signals to strengthen stepwise reasoning [20], while *Qwen-3* is optimized for structured instruction following and demonstrates strong multilingual capabilities [21].

All models were executed in inference-only mode using quantized weights via the Ollama runtime. To ensure a fair and consistent comparison across all evaluated models, a unified inference configuration was adopted. Generation parameters were fixed to a temperature of 0.6, maximum generation length of 2,048 tokens, and context window of 128K tokens. In addition, all models were executed with the reasoning mode ("think") enabled.

### C. Error Spaces

Since LLM-generated narratives can fail in diverse and unpredictable ways, the framework must be tested across varying levels of difficulty, frequency, and structural complexity to ensure robust and generalizable evaluation. To this end, we defined two complementary error spaces capturing both realistic and systematic failure modes.

**Natural Error Space.** This space captures naturally occurring errors arising during explanation generation under standard operating conditions, including incomplete attributions, unsupported causal claims, factual drifts, and structural omissions. It was constructed by executing the Explainer–Verifier pipeline in a controlled loop until either 1,000 verified instances were collected or 200 outputs were rejected. To ensure balanced representation and avoid bias toward specific methods or datasets, a round-robin scheduling strategy was applied across the five XAI methods and datasets.

**Synthetic Error Space.** To ensure full coverage of all predefined error categories, including rare ones, a mutation-based synthetic space was constructed. Starting from valid explanations, mutation operators perturb them along six orthogonal axes: SwapTopFeature, SwapMinorFeature, NegateRelation, OmitFeature, InsertHallucination, and TruncateResponse, each representing a distinct failure mode.

### D. Evaluation Metrics

The framework's performance was primarily evaluated based on its ability to detect erroneous explanations. Detection accuracy was computed using true positives (*TP*: faithful explanations correctly accepted), true negatives (*TN*: erroneous explanations correctly rejected), false positives (*FP*: faithful explanations incorrectly rejected), and false negatives (*FN*: erroneous explanations incorrectly accepted), which capture how reliably the Verifier distinguishes valid from invalid explanations.

To assess linguistic accessibility, the Flesch-Kincaid metrics, namely Reading Ease and Grade Level, were used [22]. Finally, uncertainty was measured using the Entropy Production Rate (EPR), derived from the entropy dynamics of the top-10 token-level log probabilities generated by the LLMs. Lower values indicate more stable outputs, whereas higher values reflect increased uncertainty [23].

## IV. Results and Discussion

This section presents and analyzes the empirical results of this study. First, to mitigate shared hallucinations, correlated errors, and self-preference biases that arise when a single LLM performs both roles [24], we employed different models for the Explainer and Verifier. Fig. 4 illustrates representative examples for use cases 1 and 3 after applying the framework, where initial explanations exhibited failures such as SwapMinorFeature and InsertHallucination, which were corrected following verification.

To evaluate the proposed method, we addressed the following four research questions (RQs):

1) RQ1: Does the architecture reduce the number of erroneous explanations that would otherwise reach the end user?
2) RQ2: Does structured meta-prompting improve the reliability of the Verifier in detecting erroneous explanations?
3) RQ3: Does the framework improve the linguistic accessibility of the explanations?

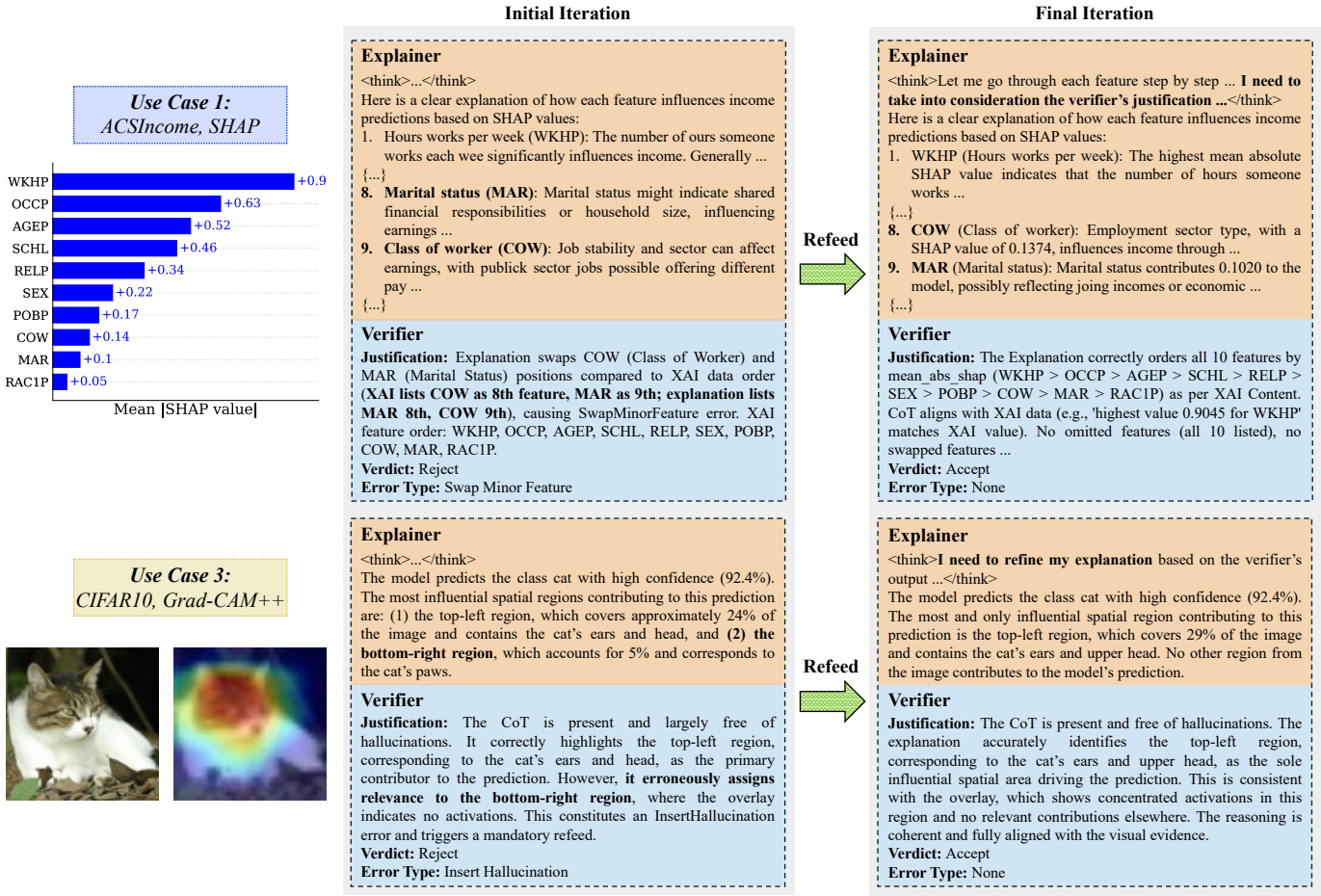


Fig. 4. Illustrative examples of the proposed framework correcting erroneous explanations after verifier-guided refinement.

4) RQ4: Do uncertainty signals, captured through EPR, reflect convergence and explanation stability?

Concerning RQ1, Table I reports the performance of all examined Explainer–Verifier pairs in the Natural Error Space experiment, as this setting is more representative of real-world use than synthetic perturbations. In this experiment, each Explainer generated outputs until either 1,000 verified explanations or 200 were rejected by an initial Verifier. In practice, all the Explainers reached the rejection threshold before attaining the acceptance target, making the 200-error limit the actual stopping criterion. The collected explanations were manually annotated by the authors according to a predefined protocol, with any disagreements resolved through discussion until a consensus was achieved. Finally, the annotated explanations were reassessed using two alternative Verifiers to evaluate the classification accuracy and robustness across the verifier configurations.

The results revealed clear performance differences among the three model families. As Explainers, the models varied substantially: `gpt-oss:20b` produced the lowest error rate (22.23%; 209 failures out of 940 outputs), followed by `deepseek-r1:14b` (32.5%), while `qwen3:30b` performed the worst at 40.95%. However, when acting as Verifiers, the ranking is reversed. Specifically, `qwen3:30b` was the most reliable critic, achieving 95.21% accuracy and 96.94% F1-score when evaluating explanations from `gpt-oss:20b`. The second-best Verifier is `gpt-oss:20b`, which achieves 90% accuracy and a 92.86% F1-score when verifying `deepseek-r1:14b`. In contrast, `deepseek-r1:14b` performs poorly as a Verifier, tending to over-accept incorrect explanations and offering limited safety guarantees; therefore, it is excluded from subsequent Verifier analyses. Overall, these findings identify `gpt-oss:20b` as the most reliable Explainer and `qwen3:30b` as the strongest Verifier and their combination yields the highest end-to-end verification accuracy (95.21%).

The "Explainer Only Acc.%" results further highlight the necessity of verification: without it, Explainer accuracy ranges only from 59% to 77.8%. For the best-performing Explainer–Verifier pair, meta-verification increases accuracy from 77.8% to 95.21%, blocking 164 out of 209 erroneous explanations (a 78.5 % reduction) from reaching the end user. This demonstrates that verification is not merely beneficial but essential for producing trustworthy and faithful explanations, particularly when the Explainer exhibits a non-negligible baseline error rate. These findings are also consistent with architectural dif-

TABLE I
PERFORMANCE OF CROSS-MODEL EXPLAINER–VERIFIER LLM PAIRS IN THE NATURAL ERROR SPACE EXPERIMENT.

| Explainer | Verifier | #Samples | TP | TN | FP | FN | Explainer Err.% | Explainer Only Acc.% | Verifier Acc.% | Verifier F1% |
|---|---|---|---|---|---|---|---|---|---|---|
| deepseek-r1:14b | gpt-oss:20b | 720 | 467 | 181 | 53 | 19 | 32.5 | 67.5 | 90.0 | 92.86 |
| deepseek-r1:14b | qwen3:30b | 720 | 406 | 183 | 51 | 80 | –//– | –//– | 81.8 | 86.08 |
| gpt-oss:20b | **qwen3:30b** | 940 | 713 | 182 | 27 | 18 | 22.23 | 77.8 | **95.21** | **96.94** |
| gpt-oss:20b | deepseek-r1:14b | 940 | 708 | 9 | 200 | 23 | –//– | –//– | 76.27 | 86.4 |
| qwen3:30b | gpt-oss:20b | 459 | 241 | 170 | 18 | 30 | 40.95 | 59 | 89.54 | 90.94 |
| qwen3:30b | deepseek-r1:14b | 459 | 268 | 4 | 184 | 3 | –//– | –//– | 59.25 | 74.1 |

*Notes:* Bold entries indicate the best Verifier; underlined entries indicate the best Explainer.
The relationship between columns is $\#\text{Samples} = \text{TP} + \text{TN} + \text{FP} + \text{FN}$.

ferences across families: *GPT-OSS*'s design for broad general-purpose reasoning supports strong explanatory reliability [19]; *DeepSeek-R1*'s forward chain-of-thought enables moderate but inconsistent explanatory quality [20]; and *Qwen-3*'s hybrid Mixture of Experts (MoE) and unified thinking design equips it with the evaluative stability needed for high-precision verification [21].

Subsequently, to investigate the effect of structured meta-prompting on verification performance (RQ2), we conducted an ablation study on the Synthetic Error Space in which the refeed mechanism was disabled, enforcing a single-pass evaluation. Under this setting, all examined Verifier models were evaluated using three progressively simplified prompt configurations. The first configuration, **V0**, corresponds to the full meta-prompting template (Fig. 2). The second, **V1**, removes the 15 core meta-prompting instructions while retaining the rest of the prompt structure. The third, **V2**, further reduces the prompt to a minimal version containing only essential task information, namely the Verifier's role and expertise, the problem statement, and the expected response format.

The results (Table II) show that **V0** achieves the highest performance in terms of both accuracy and F1-score, while the performance degrades under **V1** and remains similarly reduced under **V2**. Among the evaluated models, `qwen3:30b` demonstrated the best performance, indicating its robustness to prompt simplification. In contrast, `deepseek-r1:14b` performed the worst across all configurations, highlighting both its sensitivity to prompt design and limited effectiveness as a Verifier. Overall, these findings indicate that removing the reasoning guidance provided by meta-prompting negatively impacts the Verifier's ability to reliably assess explanations.

TABLE II
ABLATION RESULTS FOR VERIFIER MODELS ACROSS PROMPT CONFIGURATIONS (V0–V2) UNDER SINGLE-PASS EVALUATION ON THE SYNTHETIC ERROR SPACE.

| Verifier | Accuracy (%) | | | F1-score (%) | | |
|---|---|---|---|---|---|---|
| | V0 | V1 | V2 | V0 | V1 | V2 |
| qwen3:30b | **100** | 93.3 | 93.3 | **99.9** | 96.5 | 96.5 |
| gpt-oss:20b | **96.7** | 90.0 | 90.0 | **98.3** | 94.7 | 94.7 |
| deepseek-r1:14b | **40.0** | 23.3 | 30.0 | **57.1** | 37.8 | 46.15 |

To assess readability and thus accessibility (RQ3), Flesch–Kincaid metrics were computed for verified explanations. Because the Verifier acts solely as a filtering mechanism and does not modify the explanations' wording, these scores directly capture the linguistic quality produced by each Explainer. According to the results, raw XAI outputs are highly technical, with an average Reading Ease of 18.53 and Grade Level of 21.79, implying the need for postgraduate reading ability. In contrast, all Explainers significantly improved these scores. Among them, `gpt-oss:20b` yielded the most accessible explanations, achieving an average Reading Ease of 34.93 and an average Grade Level of 12.94, corresponding to an improvement of +16.40 Reading Ease points and a reduction of 8.85 grade levels relative to the baseline. Table III summarizes the readability improvements across all models. Therefore, the framework ensures that the delivered explanations are not only faithful to the underlying XAI outputs but also clearer and more accessible to end-users.

TABLE III
AVERAGE READABILITY SCORES OF EXPLAINER LLMS AND THEIR IMPROVEMENTS RELATIVE TO RAW XAI OUTPUTS.

| Explainer | Flesch Reading Ease (↑) | Flesch–Kincaid Grade (↓) |
|---|---|---|
| Raw XAI Outputs | 18.53 | 21.79 |
| gpt-oss:20b | **34.93** (+16.40) | **12.94** (-8.85) |
| qwen3:30b | 28.93 (+10.40) | 13.00 (-8.79) |
| deepseek-r1:14b | 23.35 (+4.82) | 14.54 (-7.25) |

The second part of the evaluation assessed the effectiveness of the refeed mechanism on 65 erroneous cases (35 randomly sampled from the Natural Error Space and 30 synthetic cases). This number of samples is statistically sufficient to support reliable estimation, yielding an average convergence rate of 96.92% (63/65). The corresponding Agresti–Coull 95% confidence interval is [88.7%, 99.7%], with an approximate error margin of 5.51%. Each case was allowed up to 10 refinement iterations.

First, Fig. 5 reports the number of iterations required for each Explainer–Verifier pair to produce an accepted explanation. All pairs converged rapidly, with the `gpt-oss:20b` → `qwen3:30b` combination performing best, resolving approximately 75% of cases within

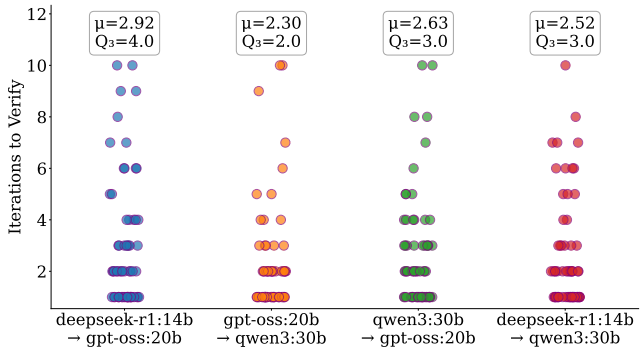


Fig. 5. Iterations required for explanation acceptance across the four Explainer–Verifier pairs. Each point represents one of the 65 cases. The mean and third quartile ($Q_3$) iteration counts are reported above each violin plot.

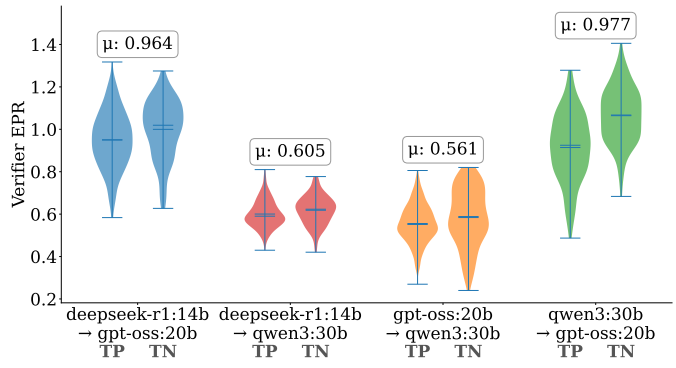


Fig. 7. Verifier EPR distributions for true positives and true negatives across all Explainer–Verifier pairs.

one or two iterations. Overall, three of the four pairings successfully verified 63 out of 65 explanations, while `deepseek-r1:14b` → `qwen3:30b` verified 64, demonstrating consistently strong correction capabilities across model families. These results highlight the robustness and effectiveness of the refeed mechanism.

From a computational perspective, the cost per explanation can be expressed as $\text{Cost} = 2 + 2K$, where $K$ denotes the number of refinement iterations. As $K$ remains low across cases, each explanation requires approximately 5-7 LLM calls. Assuming an average latency of 30 seconds per call, the total processing time is around 3 minutes per explanation. Therefore, despite the additional verification and refinement steps, the overall computational overhead remains moderate.

To quantify how uncertainty evolves throughout the refinement process (RQ4), we analyzed the Entropy Production Rate. Fig. 6 reports the median EPR across all accepted samples. From iterations 1 to 6, the EPR exhibits a monotonic decline under fixed sampling settings (constant temperature and top-$p$), suggesting that Verifier feedback is associated with progressively more stable and coherent Explainer outputs. Because most samples converge within this range, the analysis focused on these early cycles. These findings indicate that the refeed mechanism can contribute to stabilizing the reasoning trajectory while correcting erroneous explanations.

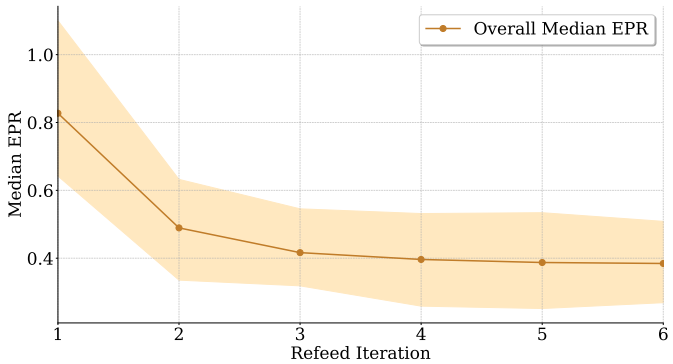


Fig. 6. Median Explainer EPR over refinement iterations (1–6) for all ultimately accepted explanations. The shaded bands show the interquartile range.

A complementary dimension of Verifier performance is the consistency of its decisions, which is also captured by the EPR, where lower values correspond to more stable and predictable verification behavior. As measured in the natural error space and shown in Fig. 7, `qwen3:30b` exhibits the lowest and tightest EPR distributions (e.g., $\mu = 0.561$, $\sigma = 0.102$ for `gpt-oss:20b`→`qwen3:30b`), reflecting reliable judgment patterns. In contrast, `gpt-oss:20b` operates with substantially higher EPR levels (e.g., $\mu = 0.964$, $\sigma = 0.138$ for `deepseek-r1:14b`→`gpt-oss:20b`), suggesting a greater internal variability. These results show that `qwen3:30b` not only achieves the highest verification accuracy but also maintains the most stable decision-making process, underscoring its strength as a Verifier.

Finally, we investigated whether the Explainer's first-iteration EPR could predict the need for iterative refinement. For this purpose, the initial EPR was treated as a continuous predictor, and "needs correction" was defined as a binary outcome: explanations requiring $\geq$3 refinement cycles or ultimately rejected were labeled 1; those accepted within 1–2 cycles were labeled 0. According to Fig. 8 that presents the resulting ROC curves, the `gpt-oss:20b` → `qwen3:30b` pair exhibited the strongest predictive power (AUC = 0.66), indicating that a higher initial EPR may correlate with a greater likelihood of requiring correction. The `deepseek-r1:14b` → `gpt-oss:20b` (AUC = 0.62) and `deepseek-r1:14b` → `qwen3:30b` (AUC = 0.59) pairs provided weaker but still useful signals. In contrast, `qwen3:30b` → `gpt-oss:20b` performed below the random baseline (AUC = 0.43). These findings show that the EPR can serve as an early-warning indicator of explanation quality, although its predictive utility is highly pair-dependent.

## V. Conclusions

This study introduced a Two-Stage LLM Meta-Verification Framework designed to improve both the accessibility and reliability of XAI explanations. The framework consists of three core components: (i) an Explainer LLM that converts

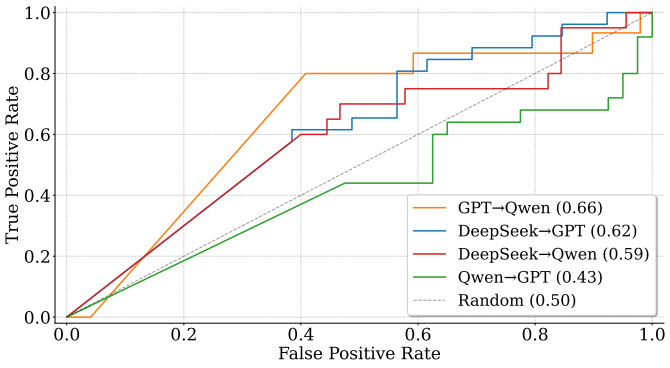


Fig. 8. ROC curves evaluating how well the initial EPR predicts the need for iterative refinement. Higher AUC values indicate stronger early-warning capability.

raw XAI outputs into natural-language explanations; (ii) a Verifier LLM that inspects these explanations for logical inconsistencies, attribution mismatches, omissions, hallucinations and structural errors; and (iii) an iterative refeed mechanism that automatically revises flawed explanations. Experiments conducted across five XAI methods and datasets, using three LLM families, demonstrated that meta-verification substantially improves explanation faithfulness, increasing accuracy from 59–77.8% for Explainer-only outputs to 81.8–95.21% after verification. The framework also enhances linguistic accessibility by simplifying complex, raw XAI outputs. Furthermore, the EPR analysis demonstrated that the Verifier's feedback progressively guided the Explainer toward more stable and coherent reasoning and that initial EPR scores could help anticipate when refinement was required. Collectively, these findings establish the framework as a practical and effective foundation for developing trustworthy and democratized XAI systems.

Despite its strengths, the proposed framework remains sensitive to prompt design and model-specific biases, motivating further exploration of prompt-engineering strategies, as well as broader evaluations across diverse LLM architectures. Future work may also explore fine-tuning both the Explainer and the Verifier on curated explanation–verification datasets. Moreover, as all XAI outputs are currently converted to text, extending the framework to multimodal foundation models capable of directly interpreting visual XAI artifacts is a promising direction. Finally, while the framework assesses dimensions such as correctness, completeness and linguistic accessibility, dedicated user studies with human participants are needed to evaluate subjective properties, including trust, perceived usefulness and overall satisfaction.

## References


[1] M. Mersha, K. Lam, J. Wood, A. K. AlShami, and J. Kalita, "Explainable artificial intelligence: A survey of needs, techniques, applications, and future direction," *Neurocomputing*, vol. 599, p. 128111, Sep. 2024.

[2] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. 31st Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 4768–4777.

[3] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

[4] M. Tyrovolas, G. Nápoles, and C. Stylios, "Backpropagation-based counterfactual explanations for quasi-nonlinear fuzzy cognitive maps," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 56, no. 3, pp. 1706–1720, Mar. 2026.

[5] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual explanations without opening the black box: Automated decisions and the GDPR," *Harvard J. Law Technol.*, vol. 31, no. 2, pp. 841–887, 2018.

[6] A. Zytek, S. Pidò, and K. Veeramachaneni, "LLMs for XAI: Future directions for explaining explanations," 2024, *arXiv:2405.06064*.

[7] M. Silvestri, V. Vineis, E. Gabrielli, F. Giorgi, F. Veglianti, F. Silvestri, and G. Tolomei, "A survey on explainable AI narratives based on large language models," *TechRxiv*, Nov. 2025.

[8] P. Mavrepis *et al.*, "XAI for all: Can large language models simplify explainable AI?" *Comput. Sci. Inf. Technol.*, vol. 14, no. 22, p. 127, 2024.

[9] G. Castellano, M. G. Miccoli, R. Scaringi, G. Vessio, and G. Zaza, "Using LLMs to explain AI-generated art classification via grad-cam heatmaps," in *Proc. 5th Italian Workshop on Explainable Artificial Intelligence (XAI.it)*, ser. CEUR Workshop Proceedings, vol. 3839, 2024, pp. 65–74. [Online]. Available: https://ceur-ws.org/Vol-3839/paper5.pdf

[10] X. Zeng, "Enhancing the interpretability of SHAP values using large language models," 2024, *arXiv:2409.00079*.

[11] A. T. Kalai, O. Nachum, S. S. Vempala, and E. Zhang, "Why language models hallucinate," 2025, *arXiv:2509.04664*.

[12] A. Castelnovo *et al.*, "Augmenting XAI with LLMs: A case study in banking marketing recommendation," in *Explainable Artificial Intelligence*, ser. Communications in Computer and Information Science. Cham: Springer, 2024, pp. 211–229.

[13] N. Shirvani-Mahdavi and C. Li, "Rule2Text: A framework for generating and evaluating natural language explanations of knowledge graph rules," 2025, *arXiv:2508.10971*.

[14] T. Ichmoukhamedov, J. Hinns, and D. Martens, "How good is my story? towards quantitative metrics for evaluating LLM-generated XAI narratives," 2024, *arXiv:2412.10220*.

[15] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," in *Proc. 36th Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2022.

[16] T. B. Brown *et al.*, "Language models are few-shot learners," in *Proc. 34th Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2020.

[17] Y. Zhang, Y. Yuan, and A. C.-C. Yao, "Meta prompting for AI systems," 2023, *arXiv:2311.11482*.

[18] F. Ding, M. Hardt, J. Miller, and L. Schmidt, "Retiring adult: New datasets for fair machine learning," in *Proc. 35th Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2021.

[19] S. Agarwal *et al.*, "gpt-oss-120b & gpt-oss-20b Model Card," 2025, *arXiv:2508.10925*.

[20] D. Guo *et al.*, "Deepseek-r1 incentivizes reasoning in LLMs through reinforcement learning," *Nature*, vol. 645, pp. 633–638, 2025.

[21] A. Yang *et al.*, "Qwen3 Technical Report," 2025, *arXiv:2505.09388*.

[22] J. P. Kincaid, R. P. Fishburne, R. L. Rogers, and B. S. Chissom, "A readability formula that saves time," Naval Air Station Memphis, Tech. Rep. Research Branch Report 8-75, 1975.

[23] C. Moslonka *et al.*, "Learned hallucination detection in black-box LLMs using token-level entropy production rate," 2025, *arXiv:2509.04492*.

[24] A. Panickssery, S. R. Bowman, and S. Feng, "Llm evaluators recognize and favor their own generations," in *Proc. 38th Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2024, p. 2197.