CONTEXT-AWARE EXPLAINABLE SELECTIVE REGRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Regression models can have low average error yet make large errors on individual samples, which can lead to suboptimal decisions. Selective regression mitigates this issue by rejecting unreliable predictions, but existing methods rely primarily on numerical data. Text associated with each sample can carry signals of unreliability that numerical data do not capture, and can thus provide evidence for rejection. We introduce *context-aware explainable selective regression*, a post-hoc framework that equips a fixed regressor with a context-aware rejector. Given a numerical input, the associated text, and the prediction from the regressor, the rejector outputs a score for threshold-based rejection. For each rejected sample, it also extracts evidence from the text and generates rationales explaining why the evidence indicates low reliability. We implement the rejector with a fine-tuned large language model. Across 16 real-world datasets, it outperforms existing methods on average, with clear gains from text on tabular data and domain-dependent effects on time-series data. While the evidence is only partially faithful to the scoring behavior of the rejector, the evidence and rationales appear plausible in case studies, which suggests their potential to support human review.

1 INTRODUCTION

Regression models support real-world decisions in domains such as finance (Holvoet et al., 2025), transportation (Gammelli et al., 2022), and supply-chain management (Huber & Stuckenschmidt, 2021; Aloini et al., 2025). Although advances in regression have improved predictive accuracy (Wu et al., 2021; Nie et al., 2023; Liu et al., 2024b; Wu et al., 2023; Holzmüller et al., 2024; Wang et al., 2025; Hollmann et al., 2025), low average error can coexist with large errors on individual samples. Such errors can lead to suboptimal decisions, which makes it necessary to assess whether an individual prediction is reliable enough to inform a decision.

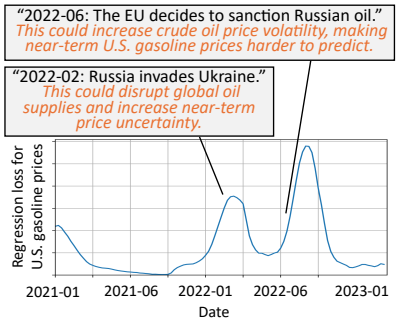


Figure 1: Conceptual illustration. Text may help assess prediction reliability.

A natural way to address this issue is to abstain from unreliable predictions, an approach known as *selective prediction* (Chow, 1970). In its regression counterpart, selective regression (Geifman & El-Yaniv, 2019; Zaoui et al., 2020), rejected samples can be routed to human review before they affect downstream decisions. Existing selective regression methods assess prediction reliability mainly from numerical data and regressor-derived signals, including representations, outputs, and uncertainty estimates (Zaoui et al., 2020; Cheng et al., 2023; Brusokas et al., 2025).

Textual context may inform rejection decisions, as illustrated in Figure 1. News (Liu et al., 2024a), event descriptions (Williams et al., 2025), user reviews (Mráz et al., 2025), and other text may reveal disruptions or unusual conditions not yet reflected in numerical data. Even when such information does not make the target easier to predict, it may help identify samples where the regressor is likely to make a large error. Prior work on multimodal regression has primarily studied whether such context improves predictive accuracy (Wang et al., 2024; Arazi et al., 2025; Zhang et al., 2025; Li

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

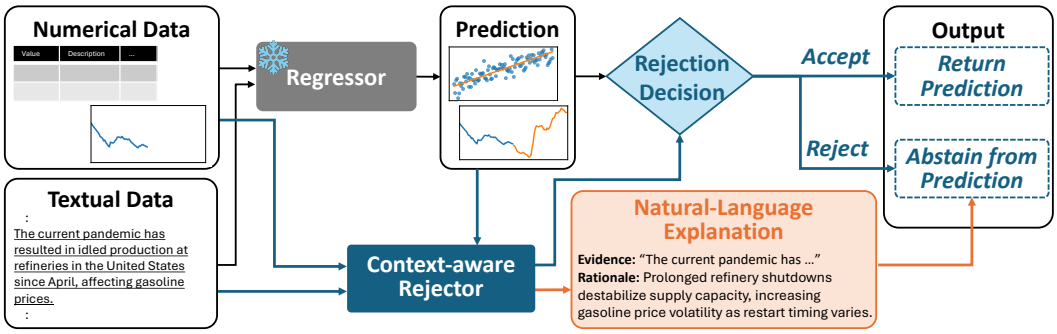


Figure 2: Overview of context-aware explainable selective regression. The rejector uses text to identify unreliable predictions and provide explanations for rejection decisions.

et al., 2026; Wang et al., 2026). We instead ask whether text can help identify unreliable predictions from a fixed regressor. Although multimodality has been studied for selective classification (Liu et al., 2026), its role in selective regression remains largely unexplored.

Text can also provide a basis for explaining rejection decisions. A rejection decision alone does not tell a human reviewer why a prediction was deemed unreliable. Prior work explains rejection decisions using examples in classification and feature attributions in selective regression (Artelt et al., 2023; Pugnana et al., 2023). We investigate whether text can provide evidence for rejection and support natural-language explanations of why a prediction may be unreliable.

To this end, we propose *context-aware explainable selective regression*, a post-hoc framework that equips a fixed regressor with a context-aware rejector, as illustrated in Figure 2. To leverage semantic cues in text, the rejector uses a large language model (LLM) to estimate sample-wise regression loss from the numerical data, the associated text, and the prediction from the regressor. The estimated loss is used for threshold-based rejection. The LLM can be fine-tuned for this task. For each rejected sample, it extracts evidence from the text and explains why the evidence indicates low reliability.

Our contributions are threefold:

- We propose context-aware explainable selective regression, which uses text both to identify unreliable predictions and to provide explanations for their rejection.
- We develop an LLM-based rejector that estimates sample-wise regression loss, applies threshold-based rejection, and generates explanations for rejection decisions.
- Across 16 real-world regression datasets with associated text, we show that the proposed rejector outperforms existing methods on average. We also evaluate the faithfulness of the extracted evidence and qualitatively examine the plausibility of the evidence and rationales.

2 Related Work

Selective prediction (Chow, 1970) addresses the trade-off between the proportion of accepted samples and their average loss by allowing a model to abstain from unreliable predictions. Geifman & El-Yaniv (2019) propose a deep learning framework that jointly trains a predictor and a rejector. Although selective prediction has been studied primarily for classification (Geifman & El-Yaniv, 2017; Ziyin et al., 2019; Liang et al., 2024; Heng & Soh, 2026), extensions to regression include conditional variance estimation (Zaoui et al., 2020), cost-based rejection (Cheng et al., 2023), post-hoc error prediction (Li et al., 2024; Altieri et al., 2024), and the use of conformal prediction interval widths to make rejection decisions (Johansson et al., 2024; Sokol et al., 2026). Selective regression methods for time-series data use energy-based models (Brusokas et al., 2025), estimates of conditional variance (Feng et al., 2025; Stradiotti et al., 2026), or estimated error percentiles (Inácio et al., 2026) to reject unreliable predictions. Most of these approaches construct scores from numerical data and model-derived quantities.

Multimodal regression methods combine text with numerical data to improve predictive accuracy in tabular regression (Shi et al., 2021; Tang et al., 2024) and time-series regression (Liu et al., 2024a;

Wang et al., 2024; Williams et al., 2025; Li et al., 2026; Wang et al., 2026). Their success suggests that text can complement numerical data, although its predictive value depends on the dataset and setting (Arazi et al., 2025; Zhang et al., 2025). In our setting, we use text post hoc to identify unreliable predictions from a fixed regressor rather than to improve its predictive accuracy. Ma et al. (2021) and Bose et al. (2026) use multimodal inputs for uncertainty estimation and conformal prediction, respectively. Liu et al. (2026) exploit confidence discrepancies between unimodal and fused predictions for selective classification. Torres et al. (2026) propose a framework that combines text and time-series data for selective classification in finance. However, using associated text to identify unreliable predictions in selective regression remains underexplored.

Existing work on explaining rejection decisions includes example-based explanations in classification (Artelt et al., 2023) and feature attributions for regression (Pugnana et al., 2023). Subramanyam et al. (2025) combine failure detection with attribute-based explanations. None of these studies provides natural-language explanations for rejection decisions in selective regression.

3 PRELIMINARIES

Let $(\mathbf{X}, \mathbf{S}, \mathbf{Y})$ denote a sample, where $\mathbf{X} \in \mathcal{X}$ is the numerical data, $\mathbf{S} \in \mathcal{S}$ is the associated text, which serves as context for the numerical data $\mathbf{X}$, and $\mathbf{Y} \in \mathcal{Y}$ is the regression target. We consider a multimodal regressor $f : \mathcal{X} \times \mathcal{S} \to \mathcal{Y}$ and denote its prediction by $\widehat{\mathbf{Y}} = f(\mathbf{X}, \mathbf{S})$. We treat f as a fixed black box: we can query its predictions but cannot access or modify its weights or hidden states. The loss for each sample is

$$L = \ell(\widehat{\mathbf{Y}}, \mathbf{Y}), \tag{1}$$

where $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a regression loss function. At inference time, $\mathbf{Y}$ and hence L are unavailable.

A context-aware rejector $g : \mathcal{X} \times \mathcal{S} \times \mathcal{Y} \to \{0, 1\}$ decides whether to accept or reject each sample based on the numerical data, the associated text, and the prediction. Outputs of 1 and 0 indicate acceptance and rejection, respectively. The selective regressor $(f, g) : \mathcal{X} \times \mathcal{S} \to \mathcal{Y} \cup \{\perp\}$ is

$$(f, g)(\mathbf{X}, \mathbf{S}) = \begin{cases} \widehat{\mathbf{Y}}, & \text{if } g(\mathbf{X}, \mathbf{S}, \widehat{\mathbf{Y}}) = 1, \\ \perp, & \text{if } g(\mathbf{X}, \mathbf{S}, \widehat{\mathbf{Y}}) = 0, \end{cases} \tag{2}$$

where $\perp$ denotes abstention. In practice, g is induced by a context-aware scoring function $s : \mathcal{X} \times \mathcal{S} \times \mathcal{Y} \to \mathbb{R}$, with higher scores indicating less reliable predictions. For a prespecified threshold $\gamma \in \mathbb{R}$, the resulting rejector is defined by

$$g_\gamma(\mathbf{X}, \mathbf{S}, \widehat{\mathbf{Y}}) = \mathbb{I}\left[s(\mathbf{X}, \mathbf{S}, \widehat{\mathbf{Y}}) \leq \gamma\right], \tag{3}$$

where $\mathbb{I}[\cdot]$ is the indicator function.

Coverage is the proportion of samples accepted, and selective risk is the average loss among accepted samples. Varying γ yields a risk–coverage curve that summarizes how effectively s ranks predictions by their unreliability. At a fixed coverage level, lower selective risk indicates a better ranking.

For each sample rejected by g, an explanation should identify evidence in $\mathbf{S}$ that supports the high score assigned by s and explain in natural language why that evidence suggests low reliability.

4 PROPOSED METHOD

We propose *context-aware explainable selective regression*, a post-hoc framework that equips the fixed regressor with a context-aware rejector. We implement this rejector using an LLM. Let the information available to the rejector be $\mathbf{Z} = (\mathbf{X}, \mathbf{S}, \widehat{\mathbf{Y}})$. The framework comprises three steps. First, the rejector estimates the regression loss for each sample from $\mathbf{Z}$ and uses this estimate as a score. Second, the rejector compares this score against a prespecified threshold to accept or reject the prediction. Third, for a rejected prediction, the rejector extracts evidence from $\mathbf{S}$ and explains why it suggests low prediction reliability. We can adapt the LLM for the loss-estimation task through supervised fine-tuning (SFT), followed by group relative policy optimization (GRPO) (Shao et al., 2024). We separate scoring from explanation generation since SFT supervises only loss estimation and does not include explanation targets. For vector-valued or multi-horizon targets, the rejector

accepts or rejects the entire prediction; we do not consider partial rejection of individual dimensions or horizons (Stradiotti et al., 2026).

4.1 Scoring with an LLM

We instantiate the regression loss in Section 3 as the sample-wise mean squared error (MSE)

$$L = \ell(\widehat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{d}\|\mathbf{Y} - \widehat{\mathbf{Y}}\|_2^2, \tag{4}$$

where d is the dimension of $\mathbf{Y}$. For a fixed f, the optimal fixed-coverage rejection rule ranks samples by their conditional expected loss $\mathbb{E}[L \mid \mathbf{Z}]$ and accepts those with the smallest values (Franc et al., 2023; Zaoui et al., 2020). Because L is unavailable at inference time, we prompt an LLM $\mathcal{M}_\theta$ to estimate it from $\mathbf{Z}$. The loss estimate serves as a score to determine rejection.

Let $\mathcal{P}$ be a function that serializes the task instruction and $\mathbf{Z}$ into a prompt. We define a score function as

$$s_\theta(\mathbf{Z}) = \text{str-to-num}(\mathcal{M}_\theta(\mathcal{P}(\mathbf{Z}))), \tag{5}$$

where str-to-num parses the generated string into a numerical value. We provide a prompt template for this task in Appendix A. The prompt asks the LLM to predict the realized regression loss for each sample. Consequently, a larger $s_\theta(\mathbf{Z})$ indicates a less reliable prediction.

Fine-tuning for Loss Estimation. To adapt the LLM for loss estimation, we can fine-tune it using a training set $\mathcal{D}_{\text{train}}^{\text{rej}} = \{(\mathbf{X}_i, \mathbf{S}_i, \mathbf{Y}_i)\}_{i=1}^n$, which is disjoint from the data used to train f. This adaptation is intended to improve sample-wise loss estimation and, consequently, rejection decisions. For each training sample, we compute the target as $L_i = \frac{1}{d}\|\mathbf{Y}_i - \widehat{\mathbf{Y}}_i\|_2^2$.

For SFT, we convert each target loss L_i to a fixed-precision string and tokenize it as $\mathbf{a}_i = (a_{i,1}, \ldots, a_{i,m_i})$ is the resulting token sequence. The SFT objective is

$$\mathcal{L}_{\text{SFT}}(\theta) = -\sum_{i=1}^{n}\sum_{t=1}^{m_i} \log p_\theta(a_{i,t} \mid \mathcal{P}(\mathbf{Z}_i), \mathbf{a}_{i,<t}). \tag{6}$$

This stage trains the model to estimate regression loss in the required output format. The token-level objective, however, does not encode the numerical distance between a generated value and L_i.

We apply GRPO with a sequence-level reward in the optional second stage. Starting from the SFT model, we optimize the LLM parameters to maximize the following numerical reward:

$$R(L_i, s_\theta(\mathbf{Z}_i)) = -\left(L_i - s_\theta(\mathbf{Z}_i)\right)^2. \tag{7}$$

This reward aligns the values represented by complete generated strings with the numerical loss-estimation objective. We omit reward scaling to preserve differences in difficulty across samples. We do not use the token-level on-policy distillation objective proposed by Chen et al. (2026) to simplify the implementation. We provide detailed settings for SFT and GRPO in Appendix B.3.

4.2 Rejection Decision

For a target coverage $c \in (0, 1)$, let $\mathcal{D}_{\text{cal}} = \{(\mathbf{X}_i, \mathbf{S}_i, \mathbf{Y}_i)\}_{i=1}^m$ be a calibration set disjoint from the data used to train f and the rejector. Following Li et al. (2024), we set γ_c to the $\lceil c(m+1)\rceil$-th smallest value of the scores $s_\theta(\mathbf{Z})$ computed on $\mathcal{D}_{\text{cal}}$. The resulting rejector is

$$g_{\gamma_c}(\mathbf{X}, \mathbf{S}, \widehat{\mathbf{Y}}) = \mathbb{I}[s_\theta(\mathbf{Z}) \le \gamma_c]. \tag{8}$$

Thus, predictions with scores above γ_c are rejected, while the remaining predictions are accepted.

4.3 Explanation Generation for Rejected Samples

For each rejected sample, the LLM is prompted again to identify which parts of the input text support the high score assigned to that sample. Let $\mathcal{T}(\mathbf{S}) = (T_1, \ldots, T_m)$ denote the segmentation of $\mathbf{S}$ into sentences. The explanation consists of an ordered list of k evidence sentences $\mathbf{S}' = (T_{j_1}, \ldots, T_{j_k})$

and corresponding rationales $\mathbf{R} = (r_1, \ldots, r_k)$. Each r_l explains why T_{j_l} suggests a high score, and the evidence sentences are ordered by their estimated importance. k is a hyperparameter.

Let $\mathcal{P}_{\text{ex}}$ denote the explanation-generation instruction and $[\cdot]$ denote prompt concatenation. We provide the prompt template in Appendix A. We generate the explanation as

$$(\mathbf{S}', \mathbf{R}) = \mathcal{M}_\theta([\mathcal{P}(\mathbf{Z}), \text{num-to-str}(s_\theta(\mathbf{Z})), \mathcal{P}_{\text{ex}}]) \,. \tag{9}$$

We do not fine-tune the LLM for explanation generation. The extracted evidence is intended to direct a human reviewer to text that warrants scrutiny. We note that generating explanations after scoring does not guarantee that these explanations are faithful to the scoring behavior of the rejector (Jacovi & Goldberg, 2020; Madsen et al., 2024).

5 Experiments

We evaluate the proposed method on tabular and time-series regression datasets with associated text. The experiments assess selective regression performance and the faithfulness of the extracted evidence, and qualitatively examine the plausibility of the evidence and rationales.

5.1 Experimental Settings

Datasets. For tabular regression, we use seven datasets from TextTabBench (Mráz et al., 2025): Airbnb, Beer, Houses, Laptops, Mercari, Permits, and Wine. For time-series regression, we use nine domains from Time-MMD (Liu et al., 2024a): Agriculture, Climate, Economy, Energy, Environment, Health, Security, Social Good, and Traffic. We provide details of datasets, preprocessing, and data splits in Appendix B.1 and B.2.

Models. The regressors are TabSTAR (Arazi et al., 2025) for tabular tasks and MMTSFlib (Liu et al., 2024a) or TaTS (Li et al., 2026) for time series. The rejector uses Qwen3.5-9B (Qwen Team, 2026), Llama-3.1-8B-Instruct (Grattafiori et al., 2024), or GPT-5.4-nano (OpenAI, 2026). We evaluate all three LLMs without task-specific adaptation. For Qwen3.5-9B, we also evaluate variants trained with SFT alone and with SFT followed by GRPO. Fine-tuning and inference settings are provided in Appendix B.3 and Appendix B.4, respectively. Appendix B.7 describes the input modifications used in our ablation variants.

Baselines. We compare against two post-hoc selective regression methods that estimate loss using only numerical data: SR-Kernel, based on kernel regression, and SR-kNN, based on k-nearest-neighbor regression (Li et al., 2024; Zaoui et al., 2020). For time-series tasks, we additionally include the Time-Energy Model (TEM) (Brusokas et al., 2025). These baselines are unimodal and do not use textual data. We therefore construct multimodal baselines by using TabSTAR, MMTSFlib, or TaTS to predict sample-wise realized losses of the fixed regressor. We denote the resulting baselines by SR-TabSTAR, SR-MMTSFlib, and SR-TaTS, respectively. Note that they are not previously proposed selective regression methods. Details are provided in Appendix B.6.

Evaluation metrics. We assess selective regression using three metrics: the normalized area under the risk–coverage curve (NAURC) (Cattelan & Silva, 2024); empirical coverage, the fraction of accepted samples; and normalized selective risk, the MSE over accepted samples divided by the full-coverage risk. Lower NAURC and normalized selective risk are better, and empirical coverage should be closer to the target. To assess the faithfulness of the extracted evidence to the scoring behavior of the rejector, we report normalized comprehensiveness, the relative score decrease after removing the selected evidence, and normalized sufficiency, the relative score decrease when retaining only that evidence (DeYoung et al., 2020). Higher comprehensiveness and lower sufficiency indicate greater faithfulness. Appendix B.8 gives the formal definitions of the metrics.

5.2 Results

5.2.1 Selective Regression Performance

Tabular regression. Table 1 reports NAURC on the tabular regression datasets. Qwen3.5-9B (SFT+GRPO) achieves the lowest average NAURC and the best average rank. Task-specific fine-

Table 1: NAURC on the tabular regression datasets. Lower is better. Bold indicates the best result, and underlining indicates the second-best.

Method	Airbnb	Beer	Houses	Laptops	Mercari	Permits	Wine	Avg.	Avg. Rank
Unimodal baselines									
SR-Kernel	0.408	0.913	0.342	0.530	0.643	0.206	0.496	0.505	7.29
SR-kNN	0.501	0.847	0.223	0.505	0.715	0.214	1.696	0.671	6.29
Multimodal baselines									
SR-TabSTAR	**0.271**	**0.576**	0.364	0.377	0.239	**0.120**	0.244	0.313	2.57
LLM-based rejectors (Proposed)									
GPT-5.4-nano	0.616	1.031	0.615	0.907	0.626	0.265	0.570	0.661	9.71
LLaMA-3.1-8B-Instruct	1.127	1.003	1.117	1.005	1.023	1.046	1.490	1.116	13.00
Qwen3.5-9B (Base)	0.989	0.991	1.174	0.951	0.946	0.606	1.131	0.970	11.29
Qwen3.5-9B (SFT)	0.644	0.890	0.354	0.325	0.259	0.157	0.439	0.438	5.14
Qwen3.5-9B (SFT+GRPO)	0.296	0.729	**0.134**	0.338	**0.201**	0.126	**0.236**	**0.294**	**1.86**
Ablations									
Qwen3.5-9B (SFT, w/o text)	0.452	0.940	0.388	0.940	0.348	0.132	0.558	0.537	6.71
Qwen3.5-9B (SFT+GRPO, w/o text)	0.476	0.903	0.383	0.956	0.323	0.134	0.558	0.534	6.86
Qwen3.5-9B (SFT, w/o $\widehat{\mathbf{Y}}$)	0.968	1.000	0.528	0.801	0.441	0.913	1.145	0.828	10.00
Qwen3.5-9B (SFT, w/o text and $\widehat{\mathbf{Y}}$)	0.994	0.984	1.806	0.903	0.991	0.920	0.708	1.044	11.43
Qwen3.5-9B (SFT, random text)	0.593	1.079	1.194	0.315	0.284	0.132	0.370	0.567	6.57
Qwen3.5-9B (SFT, rephrased text)	0.887	1.110	0.301	**0.307**	0.246	0.153	0.457	0.495	6.00

tuning contributes to this performance gain. SFT lowers the average NAURC of Qwen3.5-9B, and subsequent GRPO yields a further reduction. Consistent with these gains, Table 8 in the Appendix shows that the fine-tuned variants achieve lower loss-estimation error and higher Spearman rank correlation between estimated and true losses on most datasets. Together, these results indicate that fine-tuning improves the ability of the rejector to estimate and rank prediction losses, thereby enhancing selective regression performance on the tabular datasets.

The input ablations indicate that both text and predictions from a regressor are important for identifying unreliable predictions. Removing text from the SFT model increases its average NAURC. Removing the predictions also degrades performance, and removing both text and the predictions yields the worst average NAURC among the ablations. Random text performs similarly to omitting text, whereas rephrased text performs better than either variant and comparably to the original text. This pattern suggests that the rejector uses textual context to assess prediction reliability.

Table 2 shows that Qwen3.5-9B (SFT+GRPO) attains the lowest normalized selective risk in 11 of the 21 settings, including one tie with a baseline, while its empirical coverage is within 0.05 of the target in 20 settings. The NAURC results and the results at prespecified target coverage levels support the effectiveness of the proposed method on the tabular datasets.

Table 2: Normalized selective risk and empirical coverage for the tabular regression tasks. Empirical coverage is shown in parentheses.

Dataset	Target Coverage	Qwen3.5-9B (SFT+GRPO)	SR-TabSTAR	SR-Kernel
Airbnb	0.1	0.170 (0.12)	**0.089** (0.10)	0.350 (0.13)
	0.5	0.297 (0.48)	**0.263** (0.52)	0.450 (0.47)
	0.9	**0.599** (0.89)	0.722 (0.89)	0.723 (0.91)
Beer	0.1	0.659 (0.10)	**0.438** (0.11)	0.900 (0.11)
	0.5	0.774 (0.51)	**0.611** (0.50)	0.882 (0.52)
	0.9	0.905 (0.91)	**0.833** (0.90)	0.922 (0.92)
Houses	0.1	**0.027** (0.08)	0.221 (0.07)	0.203 (0.09)
	0.5	**0.158** (0.46)	0.310 (0.52)	0.298 (0.50)
	0.9	**0.321** (0.88)	1.018 (0.90)	0.967 (0.92)
Laptops	0.1	0.314 (0.07)	0.431 (0.08)	**0.156** (0.23)
	0.5	**0.324** (0.55)	0.425 (0.46)	0.745 (0.61)
	0.9	**0.609** (0.87)	0.784 (0.95)	1.020 (0.98)
Mercari	0.1	**0.029** (0.07)	**0.029** (0.10)	0.571 (0.15)
	0.5	**0.124** (0.52)	0.179 (0.50)	0.519 (0.56)
	0.9	0.661 (0.93)	**0.628** (0.92)	0.751 (0.90)
Permits	0.1	**0.000** (0.06)	0.066 (0.13)	0.077 (0.08)
	0.5	**0.023** (0.29)	0.039 (0.57)	0.087 (0.54)
	0.9	0.592 (0.93)	**0.387** (0.94)	0.556 (0.90)
Wine	0.1	**0.116** (0.12)	0.364 (0.09)	0.235 (0.20)
	0.5	0.175 (0.51)	**0.163** (0.53)	0.445 (0.74)
	0.9	0.590 (0.88)	**0.397** (0.93)	1.006 (0.99)

Time-series regression. Table 3 reports NAURC on the time-series datasets using MMTSFlib as a regressor (Table 10 in the Appendix shows results for TaTS). Qwen3.5-9B (SFT+GRPO) achieves lower average NAURC and better average rank than the baselines. Table 9 in the Appendix shows that fine-tuning lowers loss-estimation MAE in most domains, although improvements in Spearman rank correlation vary across domains. These results support improved selective regression performance on average, with gains that vary across domains. Meanwhile, the variants without text achieve lower average NAURC than their full-input counterparts under both SFT and SFT+GRPO, although text improves performance in some domains. The benefit of text therefore varies across time-series domains. As on the tabular datasets, the SFT variant with random text has a higher average NAURC than the variant with the

Table 3: NAURC on the time-series regression datasets. Lower is better. Bold indicates the best result, and underlining indicates the second-best.

Method	Agriculture	Climate	Economy	Energy	Environment	Health	Security	Social Good	Traffic	Avg.	Avg. Rank
Unimodal baselines											
SR-Kernel	0.670	**0.775**	**0.166**	0.694	0.626	0.932	1.882	2.410	0.651	0.978	8.67
SR-kNN	1.206	0.823	0.250	0.644	0.749	0.944	0.987	1.414	0.604	0.847	7.33
TEM	0.375	0.977	0.215	0.698	1.300	0.751	1.433	1.869	0.818	0.937	8.78
Multimodal baselines											
SR-MMTSFlib	1.072	0.987	0.447	0.737	0.692	0.583	1.840	1.486	1.135	0.998	10.56
SR-TaTS	1.156	1.023	0.443	0.682	0.719	0.586	1.841	1.335	1.145	0.992	10.11
LLM-based rejectors (Proposed)											
GPT-5.4-nano	0.574	1.068	0.512	0.759	0.835	0.761	1.192	0.942	1.055	0.855	10.11
LLaMA-3.1-8B-Instruct	0.721	1.032	1.325	0.710	1.112	0.802	**0.965**	0.993	1.154	0.979	11.00
Qwen3.5-9B (Base)	0.900	1.061	0.567	0.700	0.600	0.591	1.130	0.906	**0.581**	0.782	7.56
Qwen3.5-9B (SFT)	0.567	1.106	0.754	0.720	**0.334**	0.521	1.141	0.755	0.834	0.748	7.78
Qwen3.5-9B (SFT+GRPO)	0.401	1.119	0.642	0.672	**0.334**	**0.467**	1.223	0.712	0.770	0.705	**5.22**
Ablations											
Qwen3.5-9B (SFT, w/o text)	0.412	1.124	0.694	0.695	0.389	0.514	0.983	0.844	0.770	0.714	6.44
Qwen3.5-9B (SFT+GRPO, w/o text)	**0.373**	1.146	0.582	**0.612**	0.397	0.482	1.114	**0.696**	0.893	**0.699**	5.44
Qwen3.5-9B (SFT, w/o $\widehat{\mathbf{Y}}$)	0.448	1.236	0.258	0.966	0.987	0.504	1.300	1.817	0.841	0.929	10.56
Qwen3.5-9B (SFT, w/o text and $\widehat{\mathbf{Y}}$)	0.437	1.225	0.255	0.944	1.025	0.513	1.302	1.525	0.625	0.873	9.56
Qwen3.5-9B (SFT, random text)	0.541	1.125	0.662	0.784	0.511	0.488	1.230	0.951	0.914	0.801	9.56
Qwen3.5-9B (SFT, rephrased text)	0.475	1.110	0.819	0.679	0.376	0.526	1.211	0.815	0.796	0.756	7.33

original text, whereas rephrased text yields a similar average NAURC. This pattern is consistent with the rejector using sample-specific textual context. We hypothesize that differences across datasets in the information that text provides about prediction reliability may explain why text does not consistently improve selective regression performance.

Fixed-coverage evaluation reveals an additional practical challenge. Table 4 shows that Qwen3.5-9B (SFT+GRPO) achieves lower normalized selective risk than the baselines in most settings, yet empirical coverage often deviates substantially from the target across methods. Such deviations may arise from differences between the calibration and test score distributions, because a threshold calibrated on one distribution need not attain the target coverage under another. The Q–Q plots in Figure 7 reveal such distribution shifts on most time-series datasets. We hypothesize that these shifts reflect temporal nonstationarity exposed by the chronological split; Appendix C.2 provides further analysis. These findings highlight the importance of accounting for potential nonstationarity to maintain target coverage when applying selective regression to real-world time series, whose distributions can evolve over time.

Table 4: Normalized selective risk and empirical coverage for the time-series regression tasks. Empirical coverage is shown in parentheses.

Dataset	Target Coverage	Qwen3.5-9B (SFT+GRPO)	TEM	SR-Kernel
Agriculture	0.1	**0.308** (0.25)	1.316 (0.35)	3.400 (0.00)
	0.5	**0.352** (0.44)	1.316 (0.35)	1.148 (0.07)
	0.9	0.646 (0.63)	1.164 (0.59)	**0.490** (0.75)
Climate	0.1	1.250 (0.05)	1.015 (0.05)	**0.678** (0.08)
	0.5	1.025 (0.18)	0.970 (0.43)	**0.801** (0.32)
	0.9	1.044 (0.73)	0.998 (0.88)	**0.934** (0.76)
Economy	0.1	**0.491** (0.14)	0.924 (0.24)	1.383 (0.04)
	0.5	**0.752** (0.35)	1.188 (0.40)	1.263 (0.12)
	0.9	**0.850** (0.67)	1.275 (0.50)	1.391 (0.23)
Energy	0.1	**0.907** (0.03)	1.482 (0.35)	4.382 (0.07)
	0.5	**0.770** (0.21)	1.293 (0.35)	5.407 (0.15)
	0.9	**0.710** (0.53)	1.324 (0.59)	3.534 (0.40)
Environment	0.1	**0.509** (0.38)	0.731 (0.03)	0.772 (0.08)
	0.5	**0.614** (0.61)	0.885 (0.61)	0.854 (0.34)
	0.9	**0.886** (0.89)	0.977 (0.93)	1.216 (0.92)
Health	0.1	**0.562** (0.16)	0.778 (0.14)	1.488 (0.09)
	0.5	**0.646** (0.43)	0.897 (0.42)	1.278 (0.39)
	0.9	**0.690** (0.66)	0.978 (0.88)	1.273 (0.92)
Security	0.1	1.157 (0.25)	nan (0.00)	**0.576** (0.01)
	0.5	1.237 (0.44)	1.828 (0.06)	**0.726** (0.07)
	0.9	1.156 (0.63)	1.029 (0.64)	**0.691** (0.19)
Social Good	0.1	**0.465** (0.10)	1.273 (0.30)	4.142 (0.26)
	0.5	0.866 (0.53)	**0.772** (0.57)	3.079 (0.38)
	0.9	0.916 (0.77)	**0.745** (0.84)	2.705 (0.47)
Traffic	0.1	**0.065** (0.01)	0.304 (0.12)	0.294 (0.04)
	0.5	**0.141** (0.07)	0.748 (0.39)	0.329 (0.09)
	0.9	1.085 (0.37)	1.016 (0.86)	**0.397** (0.30)

Dataset-dependence of the text benefit. These results reveal a contrast across dataset types. Text provides a clear average benefit on the tabular datasets, whereas its effect on the time-series datasets varies across domains. Two factors may contribute to this contrast. In tabular datasets, text describes instance-specific attributes, such as property details and product descriptions, that may not be represented by the numerical data. In time-series datasets, by contrast, the textual context consists mainly of news or reports, only some of which may describe events that make the target difficult to predict. Moreover, several Time-MMD domains contain only a few hundred observations, which may be insufficient to learn reliable associations between textual cues and realized regression losses. The usefulness of text for selective regression may therefore depend both on whether it complements the numerical data and on whether sufficient training data are available to learn these associations. This interpretation parallels findings in multimodal regression, where gains depend on both complementary textual signals and the amount of training data (Zhang et al.,

Table 5: Normalized comprehensiveness (Comp.) and normalized sufficiency (Suff.) on the tabular datasets, as the maximum number k of extracted evidence sentences varies.

	Airbnb		Beer		Houses		Laptops		Mercari		Permits		Wine	
k	Comp.	Suff.	Comp.	Suff.	Comp.	Suff.	Comp.	Suff.	Comp.	Suff.	Comp.	Suff.	Comp.	Suff.
3	**-0.025**	**-0.165**	0.026	0.027	0.234	-0.012	0.025	**0.018**	**-0.071**	**-0.090**	**-0.025**	-0.008	-0.003	0.017
7	-0.086	-0.099	**0.037**	**0.020**	0.296	-0.011	**0.030**	0.033	-0.109	-0.081	-0.033	**-0.013**	-0.001	**0.008**
10	-0.080	-0.055	**0.037**	0.021	**0.334**	**-0.036**	-0.023	0.041	-0.107	-0.082	-0.029	-0.012	**0.006**	0.018

Table 6: Normalized comprehensiveness (Comp.) and normalized sufficiency (Suff.) on the time-series datasets, as the maximum number k of extracted evidence sentences varies.

	Agriculture		Climate		Economy		Energy		Environment		Health		Security		Social Good		Traffic	
k	Comp.	Suff.	Comp.	Suff.	Comp.	Suff.	Comp.	Suff.	Comp.	Suff.	Comp.	Suff.	Comp.	Suff.	Comp.	Suff.	Comp.	Suff.
5	0.376	0.465	-0.009	0.024	**0.238**	**0.187**	-0.067	**-0.075**	**0.071**	**0.041**	**-0.398**	-0.418	**0.238**	0.128	**-0.314**	-0.490	0.176	0.197
10	**0.379**	0.456	**-0.001**	0.019	0.220	0.233	-0.067	**-0.075**	0.070	0.043	-0.407	-0.425	0.209	0.190	-0.339	-0.484	**0.178**	0.204
30	0.362	**0.437**	-0.004	**0.009**	0.233	0.218	**-0.058**	-0.065	0.064	0.045	-0.404	**-0.430**	0.218	**0.126**	-0.399	**-0.492**	0.169	**0.194**

2025). Automatically identifying informative text and learning more effectively from limited data may therefore enhance the benefit of text for selective regression in time-series data.

5.2.2 Analysis of Explanations

Tables 5 and 6 report normalized comprehensiveness and sufficiency for samples rejected by Qwen3.5-9B (SFT+GRPO) at a target coverage of 0.7, as the maximum number k of extracted sentences varies. Figure 9 in the Appendix presents the results with standard deviations. These results suggest that the extracted evidence is only partially faithful to the scoring behavior of the rejector. On Houses, removing the evidence lowers the score, whereas retaining the evidence as the only textual data raises the score only slightly on average, which supports faithfulness under both metrics. Other datasets do not consistently exhibit this pattern. On Agriculture, removing the evidence lowers the score by approximately 37%, but retaining only the evidence also lowers it by approximately 45%; the selected evidence therefore does not suffice to reproduce the original score. On Health, the score increases on average both when the evidence is retained and when it is removed, which shows that a high score with the evidence alone is insufficient to establish faithfulness. These findings are consistent with prior work showing that retrospective self-explanations generated by LLMs can be unfaithful and that their faithfulness depends on the task (Madsen et al., 2024; Turpin et al., 2023).

Increasing the extraction limit k does not consistently improve faithfulness over the evaluated range: comprehensiveness does not consistently increase, nor does sufficiency consistently decrease. Thus, allowing more evidence sentences does not by itself ensure closer alignment with the scoring behavior. Because the evidence is extracted in a separate prompt without training for faithfulness, the selection is not constrained to reflect the contribution of each evidence sentence to the score. Training the extraction step to improve this alignment is therefore a direction for future work.

Qualitative examples. Figure 3 shows samples rejected at a target coverage of 0.7 from the Houses and Energy datasets. In Figure 3(a), the rejector extracts two sentences from the "Summary" and provides a rationale for each. The generated rationales interpret the extracted text as suggesting a smaller living area than predicted; the ground-truth value is indeed below the prediction. The output provides a reviewer with explicit textual evidence and natural-language explanations of why the prediction may warrant scrutiny. In Figure 3(b), the regressor predicts a decline, whereas the realized series first rises and then falls sharply, exhibiting substantial variation over the prediction horizon. The extracted sentences refer to high crude oil prices, gasoline production falling short of demand and thereby reducing inventories, and proposed legislation targeting price gouging. These statements identify conditions that could make the prediction unreliable. We provide examples for the other datasets in Appendix C.4.

Together, these examples illustrate how the method can direct reviewers to potentially relevant text and explain why a prediction may warrant rejection. Although the extracted evidence and generated rationales appear plausible, these case studies do not establish their correctness or usefulness to human reviewers. A dedicated human evaluation of whether such explanations improve expert decision-making is therefore an important direction for future work.

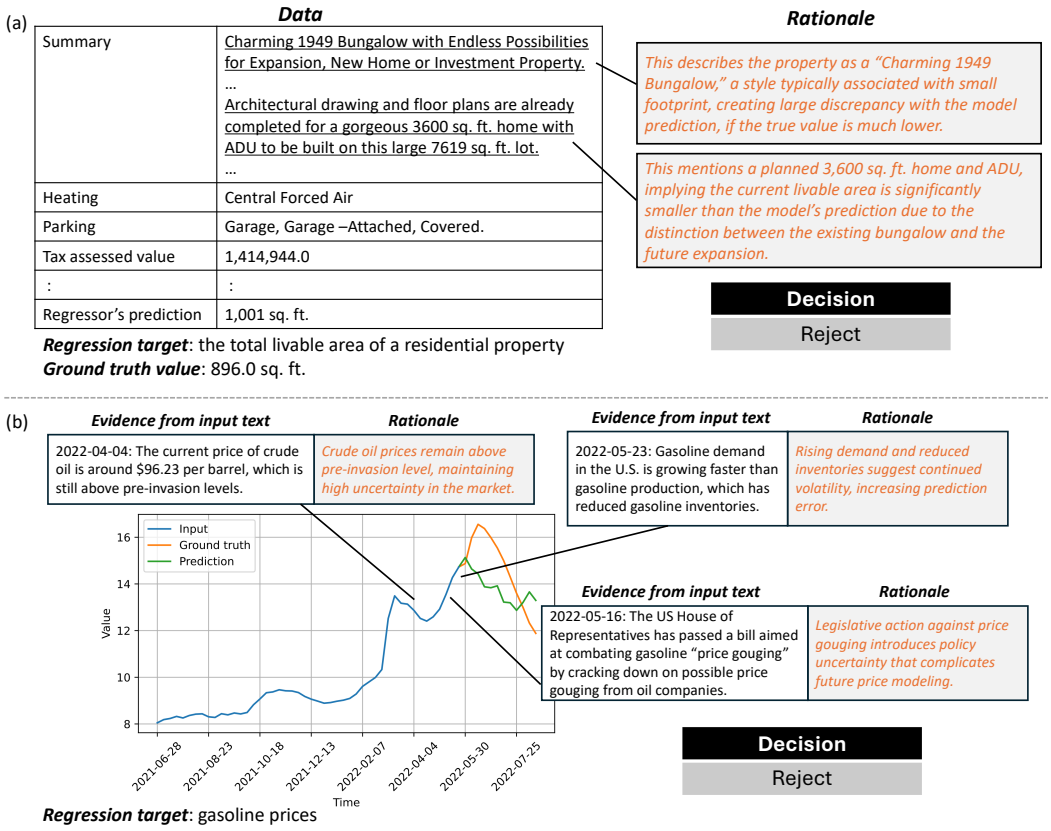


Figure 3: Examples from (a) Houses and (b) Energy. The rejector extracts sentences from the input text and generates a rationale explaining why each sentence indicates low prediction reliability.

6 Conclusion

We introduced context-aware explainable selective regression, which is a post-hoc framework that equips a fixed regressor with a context-aware rejector. Across 16 real-world datasets, it outperforms existing methods on average, with clear gains from text on tabular data and domain-dependent effects on time-series data. While the extracted evidence is partially faithful to the scoring behavior of the rejector, the evidence and generated rationales appear plausible in case studies. These results highlight the potential of text to identify unreliable predictions and support human review.

Limitations and future work. Our study has several limitations. First, although the proposed method performs better than the baselines on average, the benefit of text varies across domains and may depend on the information contained in the text. A quantitative assessment of which types of text are useful for identifying unreliable predictions would be important. Second, our implementation uses a simple LLM-based rejector. Developing mechanisms that improve both performance and faithfulness, including fine-tuning the LLM for evidence extraction, is a promising direction for future work. Third, our framework does not explicitly account for shifts in score distributions in time-series data, which may arise from nonstationarity. As our experiments illustrate, such shifts can cause empirical coverage to deviate from the target. Addressing these shifts is an important challenge when deploying the framework on real-world time series, including in streaming settings.

AI use statement

In this work, we used generative AI tools for implementing methods, assisting with translation, designing or providing feedback on research methodology or experiments, and cleaning and reformatting datasets. We have not used generative AI tools for generating synthetic datasets, helping

develop theoretical models or conceptual frameworks, proposing or refining hypotheses, supporting qualitative and thematic data analysis, and interpreting results. Formulating mathematical claims, providing critical ingredients for proving mathematical claims, and assisting in the writing of proofs are not applicable to this work. Additionally, we used generative AI tools for creating or editing software code, discovering research topics or identifying gaps, editing a research paper to improve readability, identifying relevant literature, formatting references, and suggesting a structure for a research paper. We reviewed all AI-assisted work; we reviewed all LLM-generated code and text and manually checked AI-suggested references for potential plagiarism and misunderstanding. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Ethics statement

This study involves no human participants and uses only existing benchmark datasets. The proposed rejector can help identify unreliable predictions, but it does not guarantee error-free predictions, target coverage under distribution shift, or faithful explanations. Both the scores and generated rationales may reflect biases or inaccuracies in the input text or the model. The rationales may also reproduce sensitive content. The rationales should not be treated as causal explanations. Neither the scores nor the rationales should be used as the sole basis for consequential decisions. Deployment in high-stakes settings should include domain-specific validation, monitoring for performance and coverage disparities across relevant groups, privacy-preserving data handling, and meaningful human oversight.

Reproducibility statement

We provide the detailed settings for the proposed method in the main text and the Appendix. The proposed method and its training objectives are specified in Section 4, and the experimental protocol and results are described in Section 5. The appendix provides the complete prompt templates (Appendix A) and details the datasets and preprocessing, data splits, fine-tuning and inference settings, base regressors, baselines, and evaluation metrics (Appendix B). We also include the implementation in the supplementary materials.

References

Davide Aloini, Elisabetta Benevento, Riccardo Dulmin, Emanuele Guerrazzi, and Valeria Mininno. Unlocking real-time decision-making in warehouses: A machine learning-based forecasting and alerting system for cycle time prediction. *Transportation Research Part E: Logistics and Transportation Review*, 2025.

Andres Altieri, Marco Romanelli, Georg Pichler, Florence Alberge, and Pablo Piantanida. Beyond the norms: detecting prediction errors in regression models. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Alan Arazi, Eilam Shapira, and Roi Reichart. TabSTAR: A tabular foundation model for tabular data with text fields. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

André Artelt, Roel Visser, and Barbara Hammer. "I do not know! but why?" — local model-agnostic example-based explanations of reject. *Neurocomputing*, 2023.

Alexis Bose, Jonathan Ethier, and Paul Guinand. Conformal prediction for multimodal regression, 2026. URL https://arxiv.org/abs/2410.19653.

Jonas Brusokas, Seshu Tirupathi, Dalin Zhang, and Torben Bach Pedersen. The time-energy model: Selective time-series forecasting using energy-based models. *Transactions on Machine Learning Research*, 2025.

Luís Felipe Prates Cattelan and Danilo Silva. How to fix a broken confidence estimator: Evaluating post-hoc methods for selective classification with deep neural networks. In *The 40th Conference on Uncertainty in Artificial Intelligence*, 2024.

Ming Chen, Sheng Tang, Rong-Xi Tan, Ziniu Li, Jiacheng Chen, Ke Xue, and Chao Qian. Beyond token-level supervision: Unlocking the potential of decoding-based regression via reinforcement learning. In *Forty-third International Conference on Machine Learning*, 2026.

Xin Cheng, Yuzhou Cao, Haobo Wang, Hongxin Wei, Bo An, and Lei Feng. Regression with cost-based rejection. In *Advances in Neural Information Processing Systems*, 2023.

C. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 1970.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

Ninghui Feng, Songning Lai, Xin Zhou, Jiayu Yang, Kunlong Feng, Zhenxiao Yin, Fobao Zhou, Zhangyi Hu, Yutao Yue, Yuxuan Liang, Boyu Wang, and Hang Zhao. Towards reliable time series forecasting under future uncertainty: Ambiguity and novelty rejection mechanisms, 2025.

Vojtech Franc, Daniel Prusa, and Vaclav Voracek. Optimal strategies for reject option classifiers. *Journal of Machine Learning Research*, 2023.

Daniele Gammelli, Yihua Wang, Dennis Prak, Filipe Rodrigues, Stefan Minner, and Francisco Camara Pereira. Predictive and prescriptive performance of bike-sharing demand forecasts for inventory management. *Transportation Research Part C: Emerging Technologies*, 2022.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017.

Yonatan Geifman and Ran El-Yaniv. Selectivenet: A deep neural network with an integrated reject option. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Aaron Grattafiori et al. The llama 3 herd of models, 2024.

Alvin Heng and Harold Soh. Know when to abstain: Optimal selective classification with likelihood ratios. In *The Fourteenth International Conference on Learning Representations*, 2026.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 2025.

Freek Holvoet, Katrien Antonio, and Roel Henckaerts. Neural networks for insurance pricing with frequency and severity data: A benchmark study from data preprocessing to technical tariff. *North American Actuarial Journal*, 2025.

David Holzmüller, Leo Grinsztajn, and Ingo Steinwart. Better by default: Strong pre-tuned MLPs and boosted trees on tabular data. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Jakob Huber and Heiner Stuckenschmidt. Intraday shelf replenishment decision support for perishable goods. *International Journal of Production Economics*, 2021.

Ricardo Inácio, Vitor Cerqueira, Marília Barandas, and Carlos Soares. Selective time series forecasting via metalearning, 2026.

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

Ulf Johansson, Cecilia Sönströd, and Henrik Boström. Conformal regression with reject option. In *Proceedings of the Thirteenth Symposium on Conformal and Probabilistic Prediction with Applications*, 2024.

Xiaocheng Li, Shang Liu, Chunlin Sun, and Hanzhao Wang. When no-rejection learning is consistent for regression with rejection, 2024.

Zihao Li, Xiao Lin, Zhining Liu, Jiaru Zou, Ziwei Wu, Lecheng Zheng, Dongqi Fu, Yada Zhu, Hendrik Hamann, Hanghang Tong, and Jingrui He. Language in the flow of time: Time-series-paired texts weaved into a unified temporal narrative. In *The Fourteenth International Conference on Learning Representations*, 2026.

Hengyue Liang, Le Peng, and Ju Sun. Selective classification under distribution shifts. *Transactions on Machine Learning Research*, 2024.

Haoxin Liu, Shangqing Xu, Zhiyuan Zhao, Lingkai Kong, Harshavardhan Kamarthi, Aditya B. Sasanur, Megha Sharma, Jiaming Cui, Qingsong Wen, Chao Zhang, and B. Aditya Prakash. Time-mmd: Multi-domain multimodal dataset for time series analysis. In *Advances in Neural Information Processing Systems*, 2024a.

Moru Liu, Hao Dong, Olga Fink, and Mario Trapp. Adaptive confidence regularization for multimodal failure detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024b.

Huan Ma, Zongbo Han, Changqing Zhang, Huazhu Fu, Joey Tianyi Zhou, and Qinghua Hu. Trustworthy multimodal regression with mixture of normal-inverse gamma distributions. In *Advances in Neural Information Processing Systems*, 2021.

Andreas Madsen, Sarath Chandar, and Siva Reddy. Are self-explanations from large language models faithful? In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.

Martin Mráz, Breenda Das, Anshul Gupta, Lennart Purucker, and Frank Hutter. Towards benchmarking foundation models for tabular data with text. In *ICML 2025 Workshop on Foundation Models for Structured Data*, 2025.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023.

OpenAI. Introducing GPT-5.4 mini and nano, 2026. URL `https://openai.com/index/introducing-gpt-5-4-mini-and-nano/`.

Andrea Pugnana, Carlos Mougan, and Dan Saattrup Nielsen. Model agnostic explainable selective regression via uncertainty estimation, 2023.

Qwen Team. Qwen3.5: Towards native multimodal agents, 2026. URL `https://qwen.ai/blog?id=qwen3.5`.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024.

Xingjian Shi, Jonas Mueller, Nick Erickson, Mu Li, and Alex Smola. Multimodal autoML on structured tables with text fields. In *8th ICML Workshop on Automated Machine Learning*, 2021.

Anna Sokol, Nuno Moniz, and Nitesh V. Chawla. Conformalized selective regression. *Discover Data*, 2026.

Luca Stradiotti, Laurens Devos, Anna Monreale, Jesse Davis, and Andrea Pugnana. Bounded-abstention multi-horizon time-series forecasting, 2026.

Rakshith Subramanyam, Kowshik Thopalli, Vivek Narayanaswamy, and Jayaraman J Thiagarajan. Decider: Leveraging foundation model priors for improved model failure detection and explanation. In *European Conference on Computer Vision*, 2025.

Zhiqiang Tang, Haoyang Fang, Su Zhou, Taojiannan Yang, Zihan Zhong, Cuixiong Hu, Katrin Kirchhoff, and George Karypis. Autogluon-multimodal (automm): Supercharging multimodal automl with foundation models. In *Proceedings of the Third International Conference on Automated Machine Learning*, 2024.

Dorothy Torres, Wei Cheng, and Henan Huang. Finabstain: Uncertainty-calibrated multimodal rag for selective financial forecasting, 2026.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Shiyu Wang, Jiawei LI, Xiaoming Shi, Zhou Ye, Baichuan Mo, Wenze Lin, Ju Shengtong, Zhixuan Chu, and Ming Jin. Timemixer++: A general time series pattern machine for universal predictive analysis. In *The Thirteenth International Conference on Learning Representations*, 2025.

Siyuan Wang, Peng Chen, Yihang Wang, Wanghui Qiu, Chenjuan Guo, Bin Yang, and Yang Shu. Unlocking the value of text: Event-driven reasoning and multi-level alignment for time series forecasting. In *The Fourteenth International Conference on Learning Representations*, 2026.

Xinlei Wang, Maike Feng, Jing Qiu, Jinjin Gu, and Junhua Zhao. From news to forecast: integrating event analysis in llm-based time series forecasting with reflection. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, 2024.

Andrew Robert Williams, Arjun Ashok, Étienne Marcotte, Valentina Zantedeschi, Jithendaraa Subramanian, Roland Riachi, James Requeima, Alexandre Lacoste, Irina Rish, Nicolas Chapados, and Alexandre Drouin. Context is key: A benchmark for forecasting with essential textual information. In *Forty-second International Conference on Machine Learning*, 2025.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with Auto-Correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023.

Ahmed Zaoui, Christophe Denis, and Mohamed Hebiri. Regression with reject option and application to knn. In *Advances in Neural Information Processing Systems*, 2020.

Xiyuan Zhang, Boran Han, Haoyang Fang, Abdul Fatir Ansari, Shuai Zhang, Danielle C. Maddix, Cuixiong Hu, Andrew Gordon Wilson, Michael W. Mahoney, Hao Wang, Yan Liu, Huzefa Rangwala, George Karypis, and Bernie Wang. When does multimodality lead to better time series forecasting?, 2025.

Liu Ziyin, Zhikang T. Wang, Paul Pu Liang, Ruslan Salakhutdinov, Louis-Philippe Morency, and Masahito Ueda. Deep gamblers: learning to abstain with portfolio theory. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019.

Appendix

A Prompt Templates

The proposed method first queries the LLM with $\mathcal{P}(\mathbf{Z})$ to estimate the regression loss and, for rejected samples, queries it again with $\mathcal{P}_{\text{ex}}$ to extract evidence from the input text and explain why the evidence indicates low reliability. Figures 4 and 5 present the loss-estimation prompts for tabular and time-series regression, respectively, and Figure 6 presents the explanation-generation prompt. For time-series data, following the direct-prompting format of CiK (Williams et al., 2025), we serialize each series as a sequence of "(date, value)" pairs. We can ask the LLM to estimate the root mean squared error (RMSE) of each sample, rather than its MSE, and square the resulting estimate to calculate the numerical reward. The resulting targets have a more stable numerical scale and yield better SFT performance on average. All main results incorporate RMSE-based loss estimation; corresponding ablations are reported later in this appendix. Because the square-root transformation preserves order, we can make rejection decisions using RMSE-based scores.

```
# Task
Estimate the prediction error of a pretrained multimodal regression model
for a given instance.

The true target value is unavailable. Using the numerical features,
categorical features, textual contexts, dataset metadata, and the model
prediction, estimate the conditional expected root squared error between
the prediction and the true target value.

# Output requirement
Return exactly one non-negative finite floating-point number. Round the
number exactly four decimal places. Enclose it in <answer> and </answer>
tags. Do not output explanations, comment, units, or any other text.

# Given instance
<input>
<metadata>target_variable: {variable_name}</metadata>
<numerical>{numerical_features}</numerical>
<categorical>{categorical_features}</categorical>
<context>{context}</context>
<prediction>{prediction}</prediction>
</input>
```

Figure 4: Prompt template of the loss estimation task for the tabular regression datasets.

```
# Task
Estimate the prediction error of a pretrained multimodal time series
forecasting model for a given instance.

The true target sequence is unavailable. Using the historical time series
, textual contexts, dataset metadata, and the model prediction, estimate
the conditional expected root squared error between the prediction and
the true target sequence.

# Output requirement
Return exactly one non-negative finite floating-point number. Round the
number exactly four decimal places. Enclose it in <answer> and </answer>
tags. Do not output explanations, comment, units, or any other text.

# Given instance
<input>
<metadata>domain: {dataset_name}, target_variable: {variable_name},
frequency: {frequency}, sequence_length: {seq_len}, prediction_length: {
pred_len}</metadata>
<history>{historical_time_series}</history>
<context>{context}</context>
<prediction>{predicted_time_series}</prediction>
</input>
```

Figure 5: Prompt template of the loss estimation task for the time-series datasets.

```
Here is a follow-up task for the same instance.

# Task
Select up to {k} sentences that most strongly suggest high prediction
error of this instance from the text enclosed by <context> tags in the
previous question. The context is provided again below with sentence IDs:
{text_with_IDs}

# Selection instruction
- Select up to {k} distinct sentence IDs whose associated sentence most
strongly suggest that the prediction error of this instance is high.
- For each selected sentence, provide exactly one concise sentence
explaining why it increases prediction error.
- Base each rationale only on information explicitly stated in the
selected sentence.
- Sort them in descending order of the magnitude of their impacts on
prediction error.
- Enclose each selected sentence IDs and its rationale in <selected> and
 </selected> tags.
- If there is no suitable sentence, output <selected>NA</selected>

# Output format
<selected_sentences>
<selected>[sentence_id]: One-sentence rationale</selected>
...
<selected>[sentence_id]: One-sentence rationale</selected>
</selected_sentences>

Output only the tagged section, with no introduction, summary or
additional text.
```

Figure 6: Prompt template of the explanation generation task.

B Detailed Experimental Settings

B.1 Datasets

Tabular regression datasets. We use seven regression datasets from TextTabBench (Mráz et al., 2025), which is available at `https://github.com/mrazmartin/TextTabBench`. We cap the number of records in each dataset at 3,000. The column counts below refer to the input features retained after the benchmark's preprocessing.

(1) *Airbnb* contains 32 categorical, 13 numerical, and 11 text columns describing Seattle listings, including property and room types, coordinates, host attributes, listing summaries, neighborhood overviews, and amenities. The target is the listing price. (2) *Beer* contains 3 categorical, 13 numerical, and 5 text columns covering beer styles, alcohol content, bitterness and flavor attributes, product descriptions, and tasting notes. The target is the overall review score. (3) *Houses* contains 21 categorical, 5 numerical, and 4 text columns describing properties, including lot size, year built, school quality, and free-text descriptions of appliances, flooring, and heating. The target is total interior livable area; price-related columns are excluded to avoid leakage. (4) *Laptops* contains 29 categorical, 1 numerical, and 26 text columns covering specifications and descriptions, such as processor brand, RAM, screen size, sales-package contents, and additional features. The target is price. (5) *Mercari* contains 2 categorical, 1 numerical, and 4 text columns describing marketplace listings, including item condition, shipping information, product and brand names, product descriptions, and hierarchical category information. The target is item price. (6) *Permits* contains 12 categorical, 12 numerical, and 5 text columns describing San Francisco building-permit applications, including permit and construction types, estimated cost, project descriptions, and locations. The target is approval time in days. (7) *Wine* contains 7 categorical, 3 numerical, and 6 text columns covering grape variety, vintage, alcohol content, sensory descriptions, and marketing text. The target is price.

For the numerical-only regression baselines, including SR-Kernel and SR-kNN, we one-hot encode the categorical columns and concatenate the resulting indicator variables with the numerical columns to form a single numerical input vector. For the LLM-based rejectors, we serialize each input column as a column-name–value pair and insert the resulting pairs into the corresponding numerical, categorical, or context block of the prompt. For tabular datasets, we conducted all experiments using four different data splits and report results averaged across these splits.

Time-series regression datasets. We use all nine domains from Time-MMD (Liu et al., 2024a), which is available at `https://github.com/AdityaLab/MM-TSFlib`. We perform univariate forecasting of the designated target in each domain. The domains are as follows: (1) *Agriculture* contains 496 monthly observations, with the retail broiler composite as the target; (2) *Climate* contains 496 monthly observations, with drought level as the target; (3) *Economy* contains 423 monthly observations, with international trade balance as the target; (4) *Energy* contains 1,479 weekly observations, with gasoline prices as the target; (5) *Environment* contains 15,248 daily observations from 1982 through 2023, with the air quality index as the target; (6) *Health* contains 1,389 weekly observations, with the proportion of influenza patients as the target; (7) *Security* contains 297 monthly observations, with disaster and emergency grants as the target; (8) *Social Good* contains 892 monthly observations, with the unemployment rate as the target; and (9) *Traffic* contains 531 monthly observations, with travel volume as the target.

Because the original Environment series is substantially longer than the other series, we restrict it to the 2,922 daily observations from October 1, 2015, through September 30, 2023. We select this eight-year window to obtain a conveniently sized series of approximately 3,000 observations, close to the 3,000-record cap used for the TextTabBench datasets. For Security, we apply a logarithmic transformation to the numerical series to reduce and stabilize its scale. For the Energy dataset, we sort the observations by date to ensure the correct chronological order.

The input sequence lengths are 24, 48, and 96 time steps for the monthly, weekly, and daily datasets, respectively. We use prediction lengths of $\{6, 8, 10, 12\}$ for the monthly datasets, $\{12, 24, 36, 48\}$ for the weekly datasets, and $\{48, 72, 96, 120\}$ for the daily dataset. We report performance averaged over the four prediction lengths for each dataset. For all time-series datasets, we use the `fact` column as the textual context and exclude the `pred` column.

B.2 Data Splits

We divide each dataset into five subsets in a 3:1:3:1:2 ratio: regressor training and validation sets, rejector training and validation sets, and a test set. We split tabular data randomly and time-series data chronologically. After training the regressor on the first two subsets, we keep it fixed and treat it as a black box. We use the next two subsets to train and validate the rejector, with the rejector validation set also serving as the calibration set $\mathcal{D}_{\text{cal}}$, and reserve the final subset for evaluation.

B.3 Fine-tuning Settings

B.3.1 Hyperparameters

For the loss-estimation task, we fine-tuned a LLM-base rejector using 16-bit LoRA (Hu et al., 2022). Table 7 summarizes the hyperparameters for the SFT and GRPO stages.

Table 7: Hyperparameters for task-specific fine-tuning.

Parameter	Value
Weight precision	16-bit
LoRA target modules	`all-linear`
LoRA rank r	8
LoRA scaling α	16
LoRA dropout	0.0
Optimizer	`adamw_torch_fused`
Learning rate	1e-4 (SFT) 5e-6 (GRPO)
Learning-rate scheduler	`cosine`
Warmup ratio	0.05 (SFT) 0.1 (GRPO)
Weight decay	0.01 (SFT) 0.01 (GRPO)
Effective batch size	16 (SFT) 16 (GRPO)
Training epochs	3 (SFT) 5 (GRPO)
Group size G	NA (SFT) 4 (GRPO)
Rollout temperature	NA (SFT) 0.7 (GRPO)
Clipping parameter ϵ	NA (SFT) 0.2 (GRPO)
KL coefficient β	NA (SFT) 0.0 (GRPO)
Reward scaling	NA (SFT) None (GRPO)

B.3.2 GRPO Details

We describe the GRPO objective used in the second adaptation stage. For each training sample i, we draw a group of G completion sequences from the policy before the current update:

$$\widetilde{\mathbf{a}}_i^{(j)} = \left(\widetilde{a}_{i,1}^{(j)}, \ldots, \widetilde{a}_{i,m_{i,j}}^{(j)}\right) \sim p_{\theta_{\text{old}}}(\cdot \mid \mathcal{P}(\mathbf{Z}_i)), \qquad j = 1, \ldots, G. \tag{10}$$

After parsing a completion as $\widetilde{L}_i^{(j)} = \text{str-to-num}(\widetilde{\mathbf{a}}_i^{(j)})$, we assign the sequence-level reward introduced in Equation 7:

$$R_i^{(j)} = -\left(L_i - \widetilde{L}_i^{(j)}\right)^2. \tag{11}$$

The same reward is assigned to every token in the completion. If a formatting error prevents a completion from being parsed as a number, we assign it a fixed negative reward. We form a group-

relative advantage by subtracting the mean reward within the group:

$$\widehat{A}_i^{(j)} = R_i^{(j)} - \frac{1}{G}\sum_{h=1}^{G} R_i^{(h)}. \tag{12}$$

Unlike standard GRPO, we do not divide this centered reward by the within-group standard deviation. This omission preserves information about the absolute scale of numerical errors, which standard reward scaling can remove in regression (Chen et al., 2026).

For token t of completion j, the importance ratio is

$$\rho_{i,t}^{(j)}(\theta) = \frac{p_\theta\left(\widetilde{a}_{i,t}^{(j)} \mid \mathcal{P}(\mathbf{Z}_i), \widetilde{\mathbf{a}}_{i,<t}^{(j)}\right)}{p_{\theta_{\text{old}}}\left(\widetilde{a}_{i,t}^{(j)} \mid \mathcal{P}(\mathbf{Z}_i), \widetilde{\mathbf{a}}_{i,<t}^{(j)}\right)}. \tag{13}$$

We maximize the clipped surrogate objective

$$\mathcal{J}_{\text{GRPO}}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{G}\sum_{j=1}^{G}\frac{1}{m_{i,j}}\sum_{t=1}^{m_{i,j}}\min\Big(\rho_{i,t}^{(j)}(\theta)\widehat{A}_i^{(j)}, \\ \text{clip}\Big(\rho_{i,t}^{(j)}(\theta), 1-\epsilon, 1+\epsilon\Big)\,\widehat{A}_i^{(j)}\Big), \tag{14}$$

where ϵ controls the clipping range. This objective applies the sequence-level numerical signal directly to the sampled output, while the preceding SFT stage supplies token-level supervision for the required numerical format. As stated in Section 4, we do not add the on-policy distillation loss proposed by (Chen et al., 2026).

B.4 Inference Settings

For loss estimation, we use greedy decoding for all LLM-based rejectors. For explanation generation, we use each model's default sampling configuration, except for Qwen3.5-9B, for which we follow its recommended non-thinking configuration with a temperature of 0.7 and a top-p value of 0.8, which is provided at `https://huggingface.co/Qwen/Qwen3.5-9B`. We disable thinking for both loss estimation and explanation generation. For each input and task, we generate a single completion and perform neither self-consistency sampling nor reranking.

B.5 Base Regressors

TabSTAR (Arazi et al., 2025) is a tabular foundation model that combines a pretrained language encoder with a Transformer-based tabular architecture and target-aware representations. We fine-tune its pretrained checkpoint on each downstream dataset using LoRA. We use the official implementation available at `https://github.com/alanarazi7/TabSTAR` and retain its default hyperparameters.

MMTSFlib (Liu et al., 2024a) is a representative baseline framework for time-series forecasting with textual information. We use BERT as the language embedding model and iTransformer (Liu et al., 2024b) as the time-series forecasting model, with all hyperparameters set to the values used in the original implementation available at `https://github.com/AdityaLab/MM-TSFlib`.

TaTS (Li et al., 2026) is a text-enhanced time-series forecasting method that incorporates text embeddings as additional variables in a multivariate time series. As with MMTSFlib, we use BERT as the language embedding model and iTransformer as the time-series forecasting model, with all hyperparameters set to the values used in the original implementation available at `https://github.com/iDEA-iSAIL-Lab-UIUC/TaTS`.

B.6 Baselines

We compare with unimodal baselines that do not use textual context.

SR-Kernel is a post-hoc method proposed by Li et al. (2024). Given a fixed base regressor, it fits an RBF-kernel regressor to predict the base regressor's squared error on data disjoint from those used to train the base regressor. Its output serves as a score for threshold-based rejection.

SR-kNN uses k-nearest-neighbor regression to predict squared error of a fixed regressor and uses the resulting estimate as a score for threshold-based rejection. Zaoui et al. (2020) introduce this approach as a Bayes-optimal rejection rule when the base regressor is the Bayes regressor. Because our base regressors are not restricted to Bayes regressors, we follow Li et al. (2024) and use SR-kNN as a simple baseline for predicting the squared error of a fixed base regressor.

Time-Energy Model (TEM) (Brusokas et al., 2025) uses an energy-based model (EBM) to estimate the joint distribution of the input time series and the base regressor's forecast, and uses the resulting energy as a score. TEM makes rejection decisions by energy bin rather than by directly thresholding individual scores. Specifically, it partitions the training samples into energy bins, computes the MSE within each bin, and accepts bins in ascending order of MSE until the target coverage is reached. The original method reuses the encoder of the base regressor in the EBM. In our setting, however, the base regressor is treated as a black box. To maintain the same information-access conditions across methods, we use neither its weights nor its training data. Instead, we train a separate numerical-only regressor on the rejector training set and reuse its weights in the EBM. We train the EBM using the input time series and forecasts from the evaluated base regressor, and calibrate the energy bins using that regressor's realized losses. We compute NAURC from the sample-wise energies and corresponding realized losses. We use Autoformer (Wu et al., 2021) as a numerical-only encoder with all hyperparameters set to the values used in the original implementation available at `https://github.com/JonasBrusokas/Time-Energy-Model`.

We construct multimodal baselines by using TabSTAR, MMTSFlib, or TaTS to predict the fixed regressor's sample-wise realized squared error, denoting the resulting methods by ***SR-TabSTAR***, ***SR-MMTSFlib***, and ***SR-TaTS***, respectively. We note that they are not previously proposed selective regression methods.

B.7 Ablation Variants

We construct the following variants by modifying the rejector's input prompt during both training and evaluation. The *w/o text*, *w/o* $\widehat{\mathbf{Y}}$, and *w/o text and* $\widehat{\mathbf{Y}}$ variants leave the textual context, the base regressor's prediction, or both fields blank, respectively. The *random text* variant replaces the textual context with text from another sample. For tabular datasets, we use text from a different record; for time-series datasets, we draw replacement text from the earliest subset used to train the base regressor to avoid temporal leakage. The *rephrased text* variant replaces the textual context with a version rephrased by Qwen3.5-9B that is not fine-tuned for the loss-estimation task.

B.8 Evaluation Metrics

Selective regression metrics. Consider a test set of n samples with realized losses $L_1, \dots, L_n$ and its scores $s_1, \dots, s_n$, where a larger score indicates a less reliable prediction. At a threshold γ, let $a_i(\gamma) = \mathbb{I}[s_i \le \gamma]$ indicate whether sample i is accepted. The empirical coverage and selective risk are, respectively,

$$\widehat{C}(\gamma) = \frac{1}{n}\sum_{i=1}^{n} a_i(\gamma), \qquad \widehat{R}(\gamma) = \frac{\sum_{i=1}^{n} a_i(\gamma) L_i}{\sum_{i=1}^{n} a_i(\gamma)}, \tag{15}$$

where the selective risk is defined whenever at least one sample is accepted. The full-coverage risk is $\widehat{R}_{\text{full}} = \frac{1}{n}\sum_{i=1}^{n} L_i$. At each target coverage, we report the normalized selective risk $\widehat{R}(\gamma)/\widehat{R}_{\text{full}}$ together with $\widehat{C}(\gamma)$. Lower normalized selective risk and empirical coverage closer to the target are preferred.

To define the area under the risk–coverage curve (AURC), let π be a permutation such that $s_{\pi(1)} \le \cdots \le s_{\pi(n)}$. Accepting the first k samples gives coverage $C_k = k/n$ and selective risk

$$R_k = \frac{1}{k}\sum_{j=1}^{k} L_{\pi(j)}. \tag{16}$$

Defining the empirical risk–coverage curve to take value R_k on $((k-1)/n, k/n]$, its area is

$$\text{AURC} = \int_0^1 \widehat{R}(c)\,\mathrm{d}c = \frac{1}{n}\sum_{k=1}^{n} R_k. \tag{17}$$

Let $\mathrm{AURC}_{\mathrm{oracle}}$ be the value obtained by ordering samples by increasing realized loss, and note that the expected AURC under a random ordering is $\widehat{R}_{\mathrm{full}}$. Following Cattelan & Silva (2024), we define the normalized AURC as

$$\mathrm{NAURC} = \frac{\mathrm{AURC} - \mathrm{AURC}_{\mathrm{oracle}}}{\widehat{R}_{\mathrm{full}} - \mathrm{AURC}_{\mathrm{oracle}}}. \tag{18}$$

Thus, an oracle ranking has NAURC $= 0$, while a random ranking has expected NAURC $= 1$; values greater than 1 indicate rankings worse than random. Lower values are better.

Faithfulness Metrics. For a rejected sample, let $\mathbf{S}'$ denote the selected evidence, $\mathbf{Z} = (\mathbf{X}, \mathbf{S}, \widehat{\mathbf{Y}})$ the original input to the scorer, $\mathbf{Z}_- = (\mathbf{X}, \mathbf{S} \setminus \mathbf{S}', \widehat{\mathbf{Y}})$ the input with the evidence removed, and $\mathbf{Z}_+ = (\mathbf{X}, \mathbf{S}', \widehat{\mathbf{Y}})$ the input retaining only the evidence. The comprehensiveness and sufficiency are defined as (DeYoung et al., 2020)

$$\mathrm{Comp} = s_\theta(\mathbf{Z}) - s_\theta(\mathbf{Z}_-), \qquad \mathrm{Suff} = s_\theta(\mathbf{Z}) - s_\theta(\mathbf{Z}_+). \tag{19}$$

We use normalized comprehensiveness and normalized sufficiency, both of which are normalized by $s_\theta(\mathbf{Z})$. We report each metric averaged over the evaluated rejected samples. Higher normalized comprehensiveness indicates that removing the evidence decreases the score more, whereas lower normalized sufficiency indicates that the evidence alone better preserves the original score.

B.9 Experimental Environment

All experiments were conducted on an AWS EC2 instance equipped with a single NVIDIA L40S GPU with 44 GiB of memory. We used Python 3.12.12, PyTorch 2.7.1 with CUDA 12.6, Transformers 5.5.0, TRL 0.24.0, and Unsloth 2026.7.3. For the locally hosted LLMs, we loaded the Qwen3.5-9B and Llama-3.1-8B-Instruct checkpoints from `https://huggingface.co/unsloth/Qwen3.5-9B` and `https://huggingface.co/unsloth/Meta-Llama-3.1-8B-Instruct`, respectively. For GPT-5.4-nano, we used the model snapshot of `2026-03-17`.

For explanation generation on the tabular datasets, each text column may contain multiple sentences. To construct $\mathcal{T}(\mathbf{S})$, we segment the text in each column into sentences using spaCy and assign each sentence an ID of the form `[column_name-index]`, combining the column name with the sentence's ordinal position within that column. For the time-series datasets, the text field generally contains a single sentence for each publication date. We therefore use the date as the sentence ID without additional sentence segmentation, such as processing with spaCy.

C Additional Experimental Results

C.1 Selective Regression Performance

Table 8 reports loss-estimation performance on the tabular regression datasets. Fine-tuning improves this performance on nearly all datasets. Table 9 reports loss-estimation performance on the time-series regression datasets using MMTSFlib as the base forecaster. Fine-tuning again reduces MAE on most datasets, but SFT and GRPO do not consistently improve Spearman rank correlation. This mixed result is consistent with the dataset-dependent improvements in selective regression performance observed on the time-series datasets.

Table 10 reports NAURC on the time-series regression datasets using TaTS as the base forecaster. Although the proposed method achieves the best average NAURC and the second-best average rank, its performance gains vary substantially across datasets, as they do with MMTSFlib. Table 12 further reports normalized selective risk and empirical coverage at several target coverage levels using TaTS. Empirical coverage again deviates from the target, suggesting that these deviations pose a challenge for time-series data regardless of the base forecaster.

Table 13 shows the comparison between LLM-based rejectors trained to output RMSE and MSE. Because predicting RMSE yields better SFT performance on average, we use RMSE as the LLM's prediction target in this study. For GRPO, RMSE estimates are squared before computing the MSE-based reward.

Table 8: MAE and Spearman rank correlation between the score and realized loss on the tabular regression datasets.

	Qwen3.5-9B (SFT+GRPO)		Qwen3.5-9B (SFT)		Qwen3.5-9B (Base)	
Dataset	MAE	Spearman	MAE	Spearman	MAE	Spearman
Airbnb	**2.25e1**	**0.338**	2.52e1	0.139	2.74e2	-0.015
Beer	**0.147**	**0.122**	0.189	0.066	0.275	-0.012
Houses	**1.90e2**	**0.518**	2.28e2	0.221	1.54e5	0.135
Laptops	6.94e3	0.270	**6.799e3**	**0.308**	4.49e4	-0.052
Mercari	**9.67**	**0.492**	1.19e1	0.317	1.77e1	0.021
Permits	**1.22e1**	**0.892**	1.28e1	0.889	5.48e1	0.135
Wine	**9.19e2**	**0.337**	1.06e3	0.204	1.14e4	0.140

Table 9: MAE and Spearman rank correlation between the score and realized loss on the time-series regression datasets using MMTSFlib as the base forecaster.

	Qwen3.5-9B (SFT+GRPO)		Qwen3.5-9B (SFT)		Qwen3.5-9B (Base)	
Dataset	MAE	Spearman	MAE	Spearman	MAE	Spearman
Agriculture	**0.390**	**0.445**	0.405	0.364	0.468	0.242
Climate	**0.202**	-0.121	0.216	-0.080	0.403	**-0.050**
Economy	**0.694**	0.330	0.845	0.247	0.847	**0.412**
Energy	**0.972**	**0.279**	1.134	0.215	1.131	0.211
Environment	0.330	**0.645**	**0.310**	0.630	0.520	0.349
Health	**0.471**	**0.388**	0.608	0.302	0.671	0.244
Security	**0.348**	-0.391	0.351	-0.237	0.372	**-0.192**
Social Good	**0.442**	**0.575**	0.469	0.518	0.685	0.129
Traffic	**0.314**	0.457	0.331	**0.461**	0.411	0.509

Table 10: NAURC on the time-series regression datasets using TaTS as the base forecaster. Lower is better. Bold indicates the best result, and underlining indicates the second-best.

Method	Agriculture	Climate	Economy	Energy	Environment	Health	Security	Social Good	Traffic	Avg.	Avg. Rank
Unimodal baselines											
SR-Kernel	0.619	**0.775**	**0.180**	0.596	0.696	0.685	2.116	2.094	**0.651**	0.934	4.67
SR-kNN	1.095	0.908	0.236	**0.561**	0.911	0.650	1.065	1.283	0.735	0.827	**4.44**
TEM	**0.381**	1.029	0.203	0.775	1.204	0.684	1.465	1.906	0.818	0.940	6.44
Multimodal baselines											
SR-MMTSFlib	1.093	0.984	0.436	0.698	0.627	0.542	1.911	1.441	1.149	0.987	5.78
SR-TaTS	1.109	1.022	0.440	0.690	0.655	0.540	1.910	1.305	1.253	0.980	6.22
LLM-based rejectors (Proposed)											
GPT-5.4-nano	0.497	0.993	0.677	0.641	0.850	0.854	1.329	0.808	1.025	0.853	5.67
LLaMA-3.1-8B-Instruct	0.746	1.053	1.339	0.677	0.762	0.701	**0.931**	1.108	1.156	0.941	6.33
Qwen3.5-9B (Base)	1.223	1.009	0.426	0.719	0.767	0.587	1.152	1.128	0.469	0.831	5.22
Qwen3.5-9B (SFT)	0.805	1.100	0.850	0.760	0.759	0.523	1.221	0.755	0.849	0.847	5.67
Qwen3.5-9B (SFT+GRPO)	0.403	1.171	0.675	0.770	**0.349**	**0.484**	1.362	**0.622**	0.816	**0.762**	4.56

Table 11: MAE and Spearman rank correlation between the score and realized loss on the time-series regression datasets using TaTS as the base forecaster.

	Qwen3.5-9B (SFT+GRPO)		Qwen3.5-9B (SFT)		Qwen3.5-9B (Base)	
Dataset	MAE	Spearman	MAE	Spearman	MAE	Spearman
Agriculture	**0.370**	**0.448**	0.404	0.393	0.430	0.231
Climate	**0.209**	-0.187	0.217	-0.118	0.309	**-0.067**
Economy	**0.644**	0.339	0.819	0.223	0.696	**0.464**
Energy	**0.868**	**0.247**	1.054	0.196	1.271	0.205
Environment	**0.423**	**0.437**	0.497	0.321	0.559	0.268
Health	**0.459**	**0.380**	0.551	0.326	0.587	0.190
Security	0.374	-0.427	0.367	-0.271	**0.364**	**-0.154**
Social Good	**0.436**	**0.594**	0.470	0.551	0.626	0.241
Traffic	**0.294**	0.480	0.301	0.441	0.442	**0.610**

Table 12: Normalized selective risk and empirical coverage for TaTS in the time-series regression datasets. Empirical coverage is shown in parentheses.

Dataset	Target Coverage	Qwen3.5-9B (SFT+GRPO)	TEM	SR-Kernel
Agriculture	0.1	**0.370** (0.27)	1.316 (0.35)	1.145 (0.02)
	0.5	**0.383** (0.38)	1.316 (0.35)	1.277 (0.07)
	0.9	**0.515** (0.58)	1.164 (0.59)	0.524 (0.73)
Climate	0.1	1.290 (0.03)	1.015 (0.05)	**0.667** (0.06)
	0.5	1.074 (0.17)	0.970 (0.43)	**0.811** (0.31)
	0.9	1.051 (0.72)	0.998 (0.88)	**0.944** (0.77)
Economy	0.1	**0.647** (0.13)	0.924 (0.24)	1.338 (0.04)
	0.5	**0.712** (0.37)	1.188 (0.40)	1.333 (0.12)
	0.9	**0.778** (0.59)	1.275 (0.50)	1.363 (0.22)
Energy	0.1	1.540 (0.03)	1.482 (0.35)	**1.323** (0.02)
	0.5	**0.896** (0.21)	1.293 (0.35)	3.365 (0.04)
	0.9	**0.773** (0.53)	1.324 (0.59)	3.601 (0.37)
Environment	0.1	**0.648** (0.37)	0.731 (0.03)	1.179 (0.12)
	0.5	**0.846** (0.73)	0.885 (0.61)	1.398 (0.47)
	0.9	**0.932** (0.91)	0.977 (0.93)	1.502 (0.91)
Health	0.1	**0.675** (0.21)	0.778 (0.14)	1.125 (0.06)
	0.5	**0.686** (0.45)	0.897 (0.42)	1.202 (0.35)
	0.9	**0.682** (0.69)	0.978 (0.88)	1.221 (0.74)
Security	0.1	**1.249** (0.36)	nan (0.00)	nan (0.00)
	0.5	**1.265** (0.59)	1.828 (0.06)	nan (0.00)
	0.9	1.184 (0.73)	**1.029** (0.64)	1.654 (0.01)
Social Good	0.1	**0.258** (0.12)	1.273 (0.30)	4.419 (0.21)
	0.5	**0.728** (0.55)	0.772 (0.57)	2.792 (0.37)
	0.9	0.921 (0.76)	**0.745** (0.84)	2.479 (0.58)
Traffic	0.1	0.659 (0.04)	0.304 (0.12)	**0.300** (0.03)
	0.5	**0.273** (0.06)	0.748 (0.39)	0.393 (0.12)
	0.9	0.674 (0.41)	1.016 (0.86)	**0.407** (0.26)

Table 13: NAURC comparison between LLM-based rejectors trained to output RMSE and MSE.

Method	Agriculture	Climate	Economy	Energy	Environment	Health	Security	Social Good	Traffic	Avg.
Qwen3.5-9B (SFT; RMSE)	**0.567**	1.106	**0.754**	0.720	0.334	**0.521**	1.141	**0.755**	0.834	**0.748**
Qwen3.5-9B (SFT; MSE)	1.219	**1.087**	0.850	**0.655**	**0.273**	0.689	**0.995**	0.783	**0.757**	0.812

C.2 Distribution Shift of Scores in Time-series Data

Figure 7 shows Q–Q plots of the scores estimated by the Qwen3.5-9B (SFT+GRPO) rejector on the calibration set against those on the test set. A Q–Q plot pairs the corresponding quantiles of two samples, so that all points lie on the line $y = x$ when the two distributions coincide, which is the condition under which the threshold γ_c calibrated on one set attains the target coverage on the other. For the tabular datasets, which are split randomly, the points lie close to the line for most datasets, with Permits as the only exception (Figure 7(a)). For the time-series datasets, which are split chronologically, the points deviate substantially from the line in most datasets and lie mostly above it, particularly in the upper tail; only Economy remains comparatively close to the line, although the plot lies slightly above it (Figure 7(b)). This contrast mirrors the coverage results in Tables 2 and 4: for Qwen3.5-9B (SFT+GRPO), Permits at a target coverage of 0.5 is the only tabular setting in which the empirical coverage deviates from the target by more than 0.05, whereas the empirical coverage deviates substantially in most time-series settings and falls below the target on all nine datasets at a target coverage of 0.9.

We attribute the shifts in the time-series datasets to nonstationarity of the data, which the chronological split exposes: the calibration and test sets come from different periods, so any change in the data over time appears as a shift of the score distribution. Figure 8 plots the portions of each time-series dataset used for training the rejector, calibrating the threshold, and testing. In most datasets, the calibration and test segments behave differently, with trends or level shifts that are particularly large in Agriculture, Climate, Economy, Energy, Security, and Social Good and smaller but visible in Health and Traffic. Such changes can alter the forecast errors and hence the realized losses that the scores estimate; consistently, the numerical-only baselines TEM and SR-Kernel also deviate substantially from the target coverage in Table 4. We note that the test set of every dataset consists of the most recent observations and includes data from 2020 onward (from 2022 onward for Environment), a period of major external shocks such as the COVID-19 pandemic and the Russian invasion of Ukraine. These events appear in the evidence and rationales in the case studies (Figure 3(b) and Figures 19, 20, and 21) and may have induced part of the nonstationarity in both the numerical series and the accompanying text. The short calibration segments, which span only about 30 to 90 time steps for the monthly datasets, add sampling variability to the thresholds, although variability alone would not produce shifts that are predominantly in one direction. Because a threshold calibrated on past data is applied to future data in deployment, maintaining target coverage for time series requires accounting for nonstationarity, for example by recalibrating the threshold with recent observations, which we leave for future work.

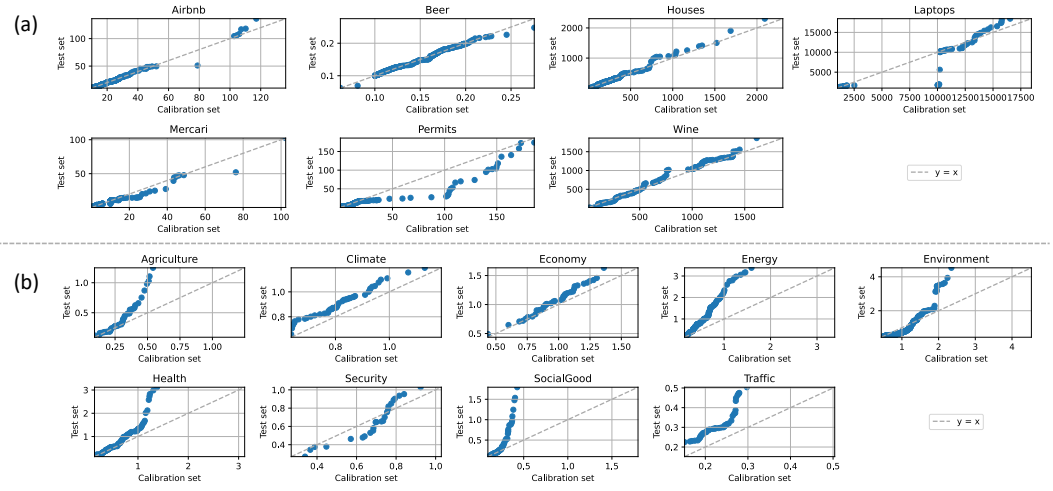


Figure 7: Q–Q plot of estimated scores by the Qwen3.5-9B (SFT+GRPO) rejector between calibration and test sets for (a) tabular datasets and (b) time-series datasets.

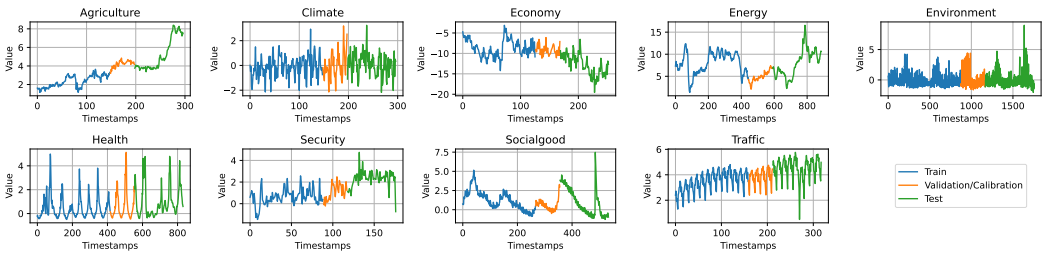


Figure 8: Time-series data used for training, validating, and testing rejectors and calibrating a threshold.

C.3 Faithfulness of Evidence

Figure 9 shows normalized comprehensiveness and sufficiency for the tabular and time-series datasets; the error bars indicate standard deviations. On Houses, the extracted evidence is faithful to the scoring behavior according to both metrics, but this pattern does not consistently hold for the other datasets.

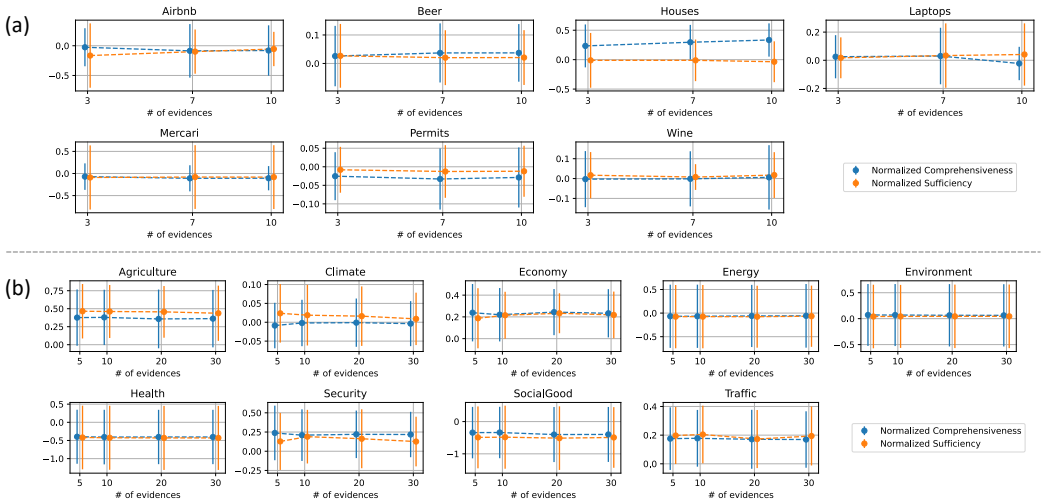


Figure 9: Normalized comprehensiveness and sufficiency for rejected samples as the maximum number of extracted evidence sentences varies for (a) the tabular datasets and (b) the time-series datasets. Error bar indicates standard deviation in each setting.

C.4 Case Studies

We present rejected examples from datasets not shown in the main text, together with the extracted evidence and generated rationales.

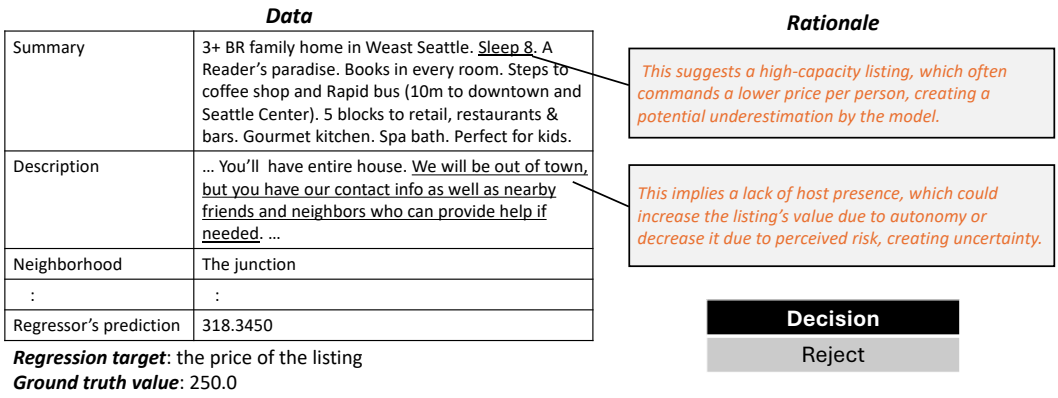


Figure 10: Rejected example from Airbnb.

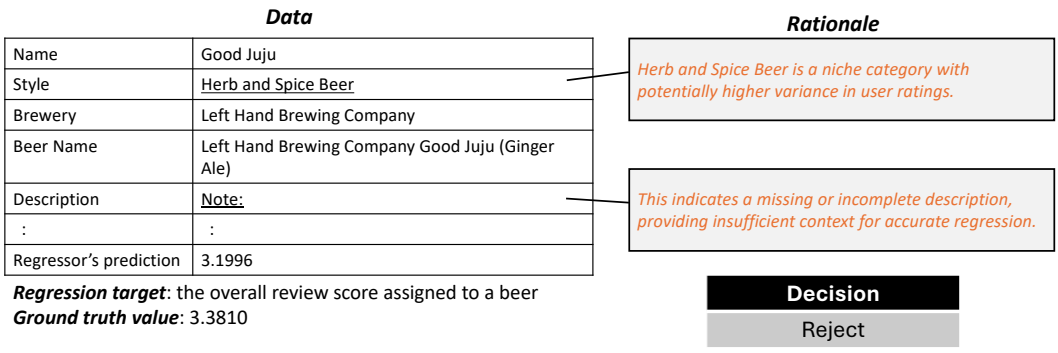


Figure 11: Rejected example from Beer.

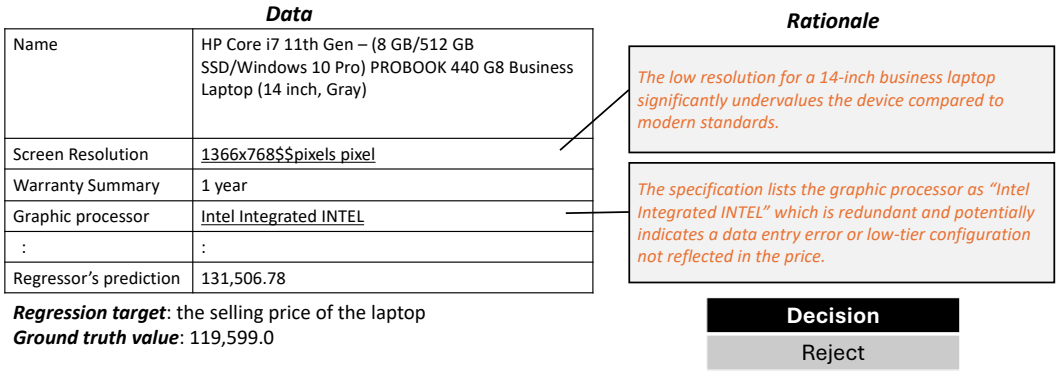


Figure 12: Rejected example from Laptops.

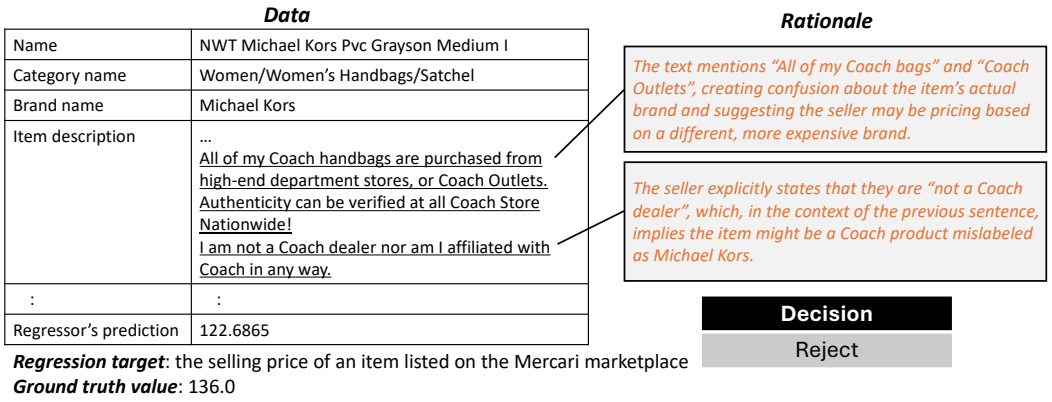


Figure 13: Rejected example from Mercari.

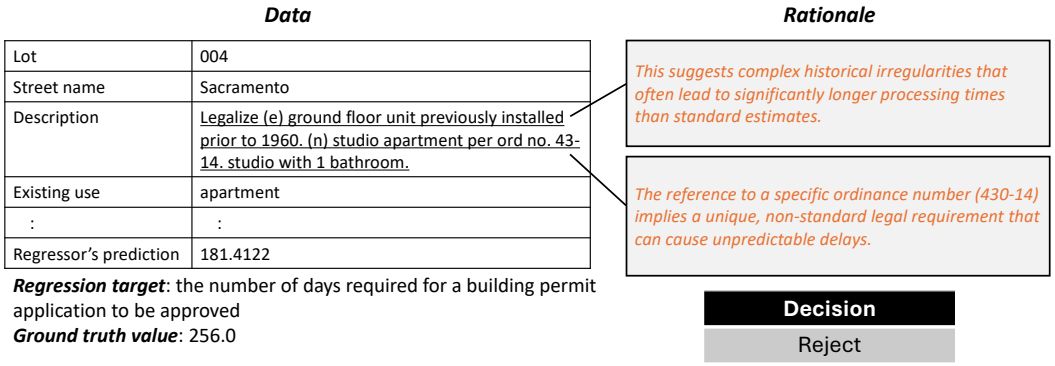


Figure 14: Rejected example from Permits.

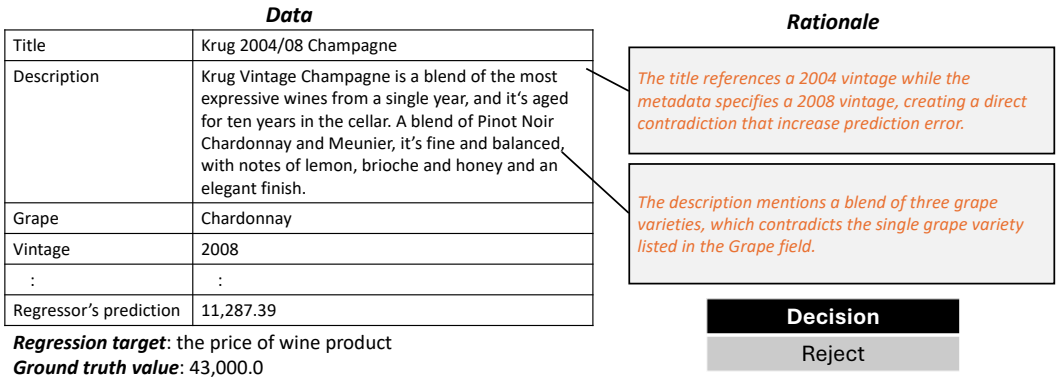
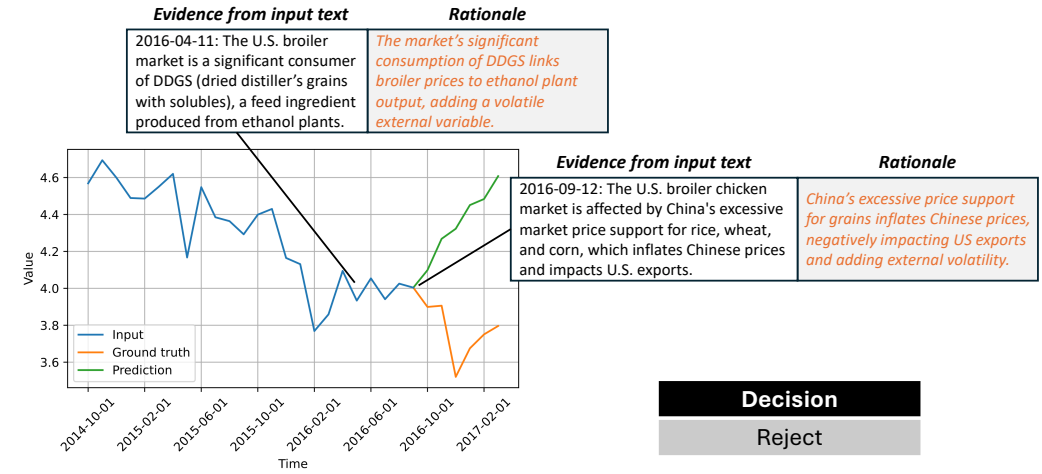
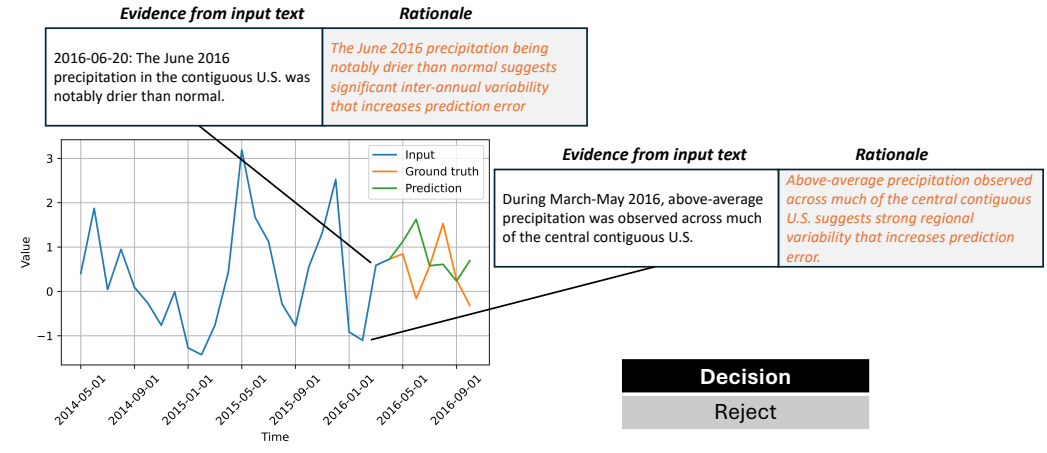


Figure 15: Rejected example from Wine.



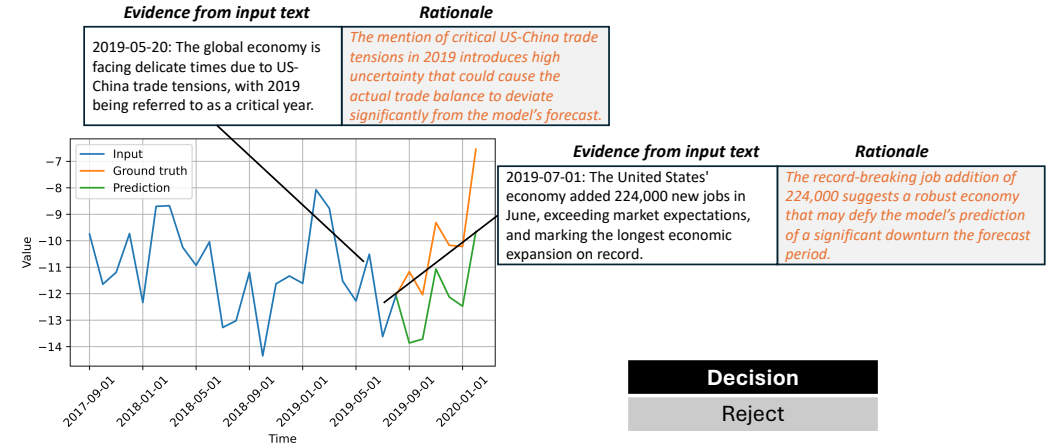
Regression target: retail broiler composite

Figure 16: Rejected example from Agriculture.



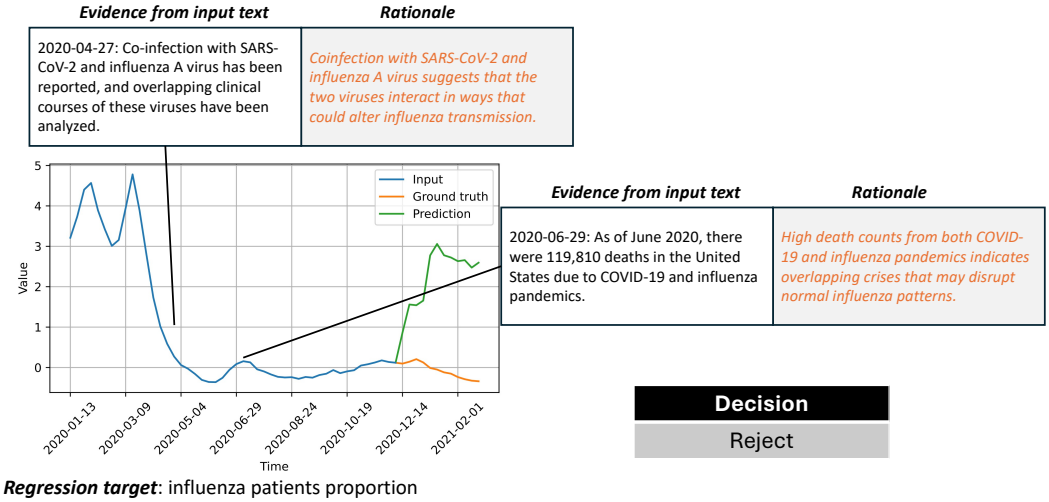
Regression target: drought level

Figure 17: Rejected example from Climate.



Regression target: international trade balance

Figure 18: Rejected example from Economy.



Regression target: influenza patients proportion

Figure 19: Rejected example from Health.

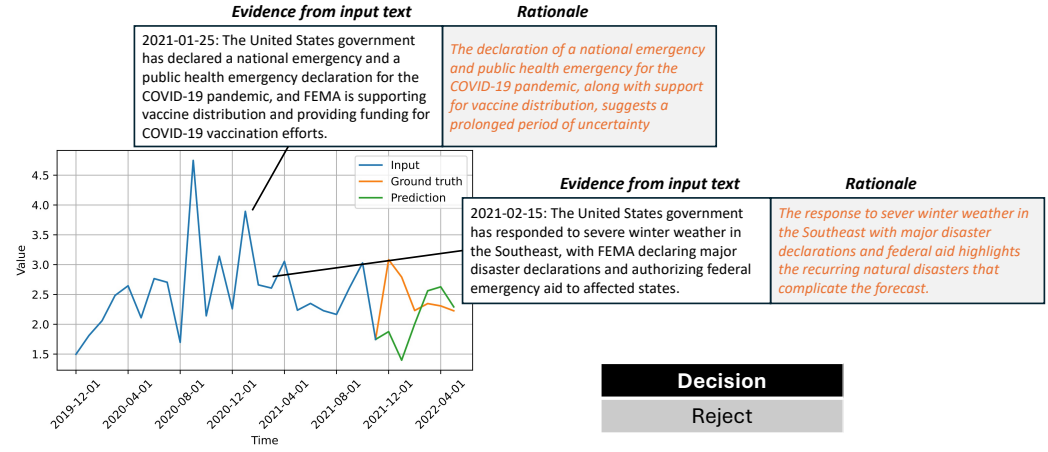


Figure 20: Rejected example from Security.

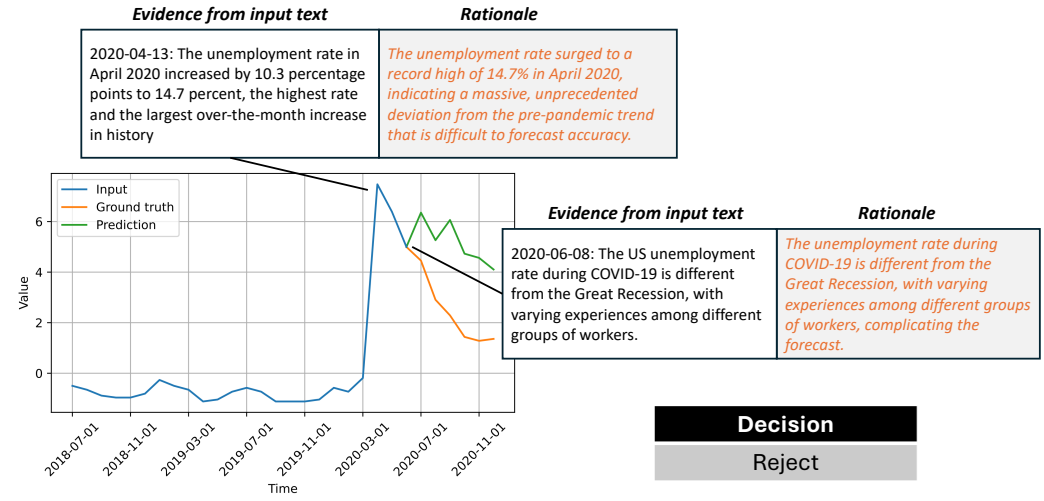


Figure 21: Rejected example from Social Good.

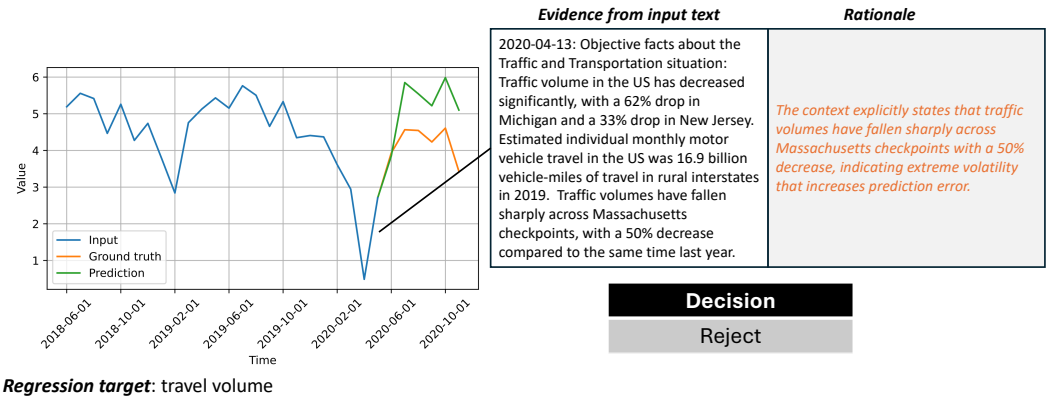


Figure 22: Rejected example from Traffic.