LLMs as Verbal Translation Layer for TSC Explanations

Félix Martí-Pérez[2], Christian Bontveit[1], Cèsar Ferri[2], and Jan Arne Telle[1]

[1] Department of Informatics, University of Bergen, Norway
[2] VRAIN, Universitat Politècnica de València, Spain

Abstract. Explaining time-series classifiers is difficult because the relevant evidence is often temporal, visual, and hard to name. Prototype explanations show representative examples, but still require users to infer the class logic visually. We study whether large language models can reduce this burden by translating labelled time-series prototypes into concise natural-language rules. We propose a prototype-to-rule pipeline and evaluate the generated rules through forward simulation, using an LLM proxy across four UCR datasets and a small exploratory human study with 20 participants. The results are dataset dependent. Rules help when class differences are clear and localised, but provide limited benefit when describing trends across multiple temporal regions. Our analysis suggests a vagueness-precision trade-off. Qualitative rules are close to how humans describe temporal patterns, but this same vagueness can make them hard to apply consistently. We conclude that LLM-generated rules are not final executable explanations, but promising human-centred intermediate representations between visual prototypes and formal decision rules. Code is available at `https://github.com/femartip/TSC_LLM_rules.git`

Keywords: Time-series classification · Explainable AI · LLMs for explainability · Forward simulation · Rule induction

1 Introduction

Time-series classification (TSC) is used in domains where model decisions often need to be inspected, explained, or communicated to users. This is difficult because the evidence used by a classifier is temporal and visual. Unlike text or images, time-series patterns are often not associated with concrete and easily identifiable features. A user may see that two curves differ, but still struggle to describe the difference in a way that can be remembered and applied to new examples.

Existing explanation methods for TSC include visual explanations, prototypes, shapelets, symbolic rules, and surrogate models. These approaches expose different aspects of the classifier's behaviour, but they also place different demands on the user. Prototype explanations are particularly intuitive because they show representative examples. However, they still require the user to infer the class logic visually.

This paper studies whether large language models (LLMs) can help reduce this burden. We use an LLM as a verbal translation layer between labelled time-series prototypes and natural-language class rules. Given a small set of prototypes for each predicted class of a reference classifier, the LLM generates concise rules describing the temporal patterns that distinguish the classes. These rules are not intended to be executable symbolic classifiers. They are qualitative descriptions, using terms such as 'early baseline', 'sharp drop', 'plateau', or 'slow recovery', closer to how humans often describe temporal shapes.

We propose a prototype-to-rule pipeline and evaluate it through forward simulation. In this setting, an explanation is useful if it helps a subject predict the label assigned by the reference classifier on held-out instances. We first use an LLM proxy to evaluate rules at scale across four UCR datasets. We then run a small exploratory human study with 20 participants across two datasets, contrasting an easier visual task with a harder one.

The results are dataset-dependent. Generated rules help when class differences are clear and localised, but provide limited benefit when discrimination depends on diffuse trends across multiple temporal regions. The human study further suggests that rules can improve perceived explanation quality and confidence. But also reveals a risk as qualitative descriptions may be readable while still being too vague to apply consistently near a decision boundary.

The contribution of this work is therefore not a claim that LLM-generated rules solve TSC interpretability. Rather, we provide preliminary evidence that they can be useful under favourable conditions, and we characterise some of the conditions under which they fail. More broadly, we argue that natural-language rules are best understood as human-centred intermediate representations, descriptions that sit between visual prototypes and formal decision rules, and whose usefulness depends on both their readability and their grounding.

2 Related Work

Explainability for time series classification has developed into a distinct area, with explanation methods commonly organised according to whether they highlight time points, subsequences, examples, or model-level structures [20]. This distinction matters because explanations for time series must respect temporal structure: perturbing or interpreting points independently can be misleading when the class evidence depends on trends, alignments, or shape patterns [18]. Recent work, therefore, includes a broad range of time-series-specific explanation formats, including saliency-style methods, simplifications, prototypes, counterfactuals, shapelets, and symbolic rules [12]. Our work is situated in this space but focuses on a specific question: whether visual prototype explanations can be translated into natural-language class rules useful for forward simulation.

A central line of work in interpretable TSC is based on shapelets, discriminative subsequences that can be inspected directly. Shapelets were introduced as a primitive for time-series data mining [22] and later scaled through the shapelet transform [15]. More recent methods continue to use shapelets as explanatory

primitives, either to identify relevant subsequences for a prediction [1] or to construct counterfactual explanations [11]. Rule-based time-series explanations are also closely related. LASTS, for example, explains arbitrary time-series classifiers using exemplars, counter-exemplars, and factual or counterfactual shapelet-based rules [8]. These approaches provide precise and often executable explanations. Our goal is different: we do not extract shapelet rules or executable symbolic conditions, but study whether an LLM can express prototype differences as qualitative, class-level natural-language rules.

Prototype-based explanations form a second relevant family. In image classification, ProtoPNet introduced a "this looks like that" form of reasoning, where predictions are explained by similarity to learned prototypical parts [3]. For sequential data, ProSeNet uses prototypes as case-based explanations and allows users to steer the model by editing prototypes [17]. Learned prototypes have also been used directly for explaining deep time-series classifiers, including ECG, respiration, and audio examples [7]. These methods show why prototypes are attractive: they are concrete and visually inspectable. However, prototypes do not automatically specify the relevant class distinction. A user may still need to infer which temporal region matters and how a new instance should be compared to the examples. This motivates our use of LLMs as a verbal translation layer that converts prototypes into explicit rules.

LLMs have recently been explored as tools for making explanations more accessible. Some work uses LLMs to simplify or adapt existing XAI outputs for different audiences [16], while other work studies LLMs as generators of interpretable rule-based systems [21]. In the time-series setting, XForecast evaluates natural-language explanations for forecasting and argues that such explanations can be more accessible than raw temporal importance scores, while also showing that evaluating them is difficult [2]. Recent work on LLM-based time-series explanation and anomaly diagnosis similarly highlights the promise and difficulty of grounding free-form text in numerical temporal evidence [14]. Our work differs from these directions in two ways: it focuses on classification rather than forecasting or anomaly detection, and it evaluates rules generated from labelled prototypes rather than explanations derived directly from model outputs or attribution methods.

Finally, our evaluation follows the idea that explanations should be tested by whether they help users predict model behaviour. Doshi-Velez and Kim distinguish application-grounded, human-grounded, and functionally grounded evaluations of interpretability [6]. Hase and Bansal evaluate explanation methods through forward simulation, measuring whether users can better predict a model's outputs when given an explanation [9]. Human studies of interpretability also show that subjective preference and objective usefulness need not coincide, and that explanation complexity affects whether users can actually apply an explanation [13]. We adopt this forward-simulation view: an explanation is useful if it helps a subject imitate the reference classifier on held-out instances. Our LLM proxy provides a scalable screening step, but the human study is needed because proxy success does not necessarily imply human usefulness.

3 Prototype-to-Rule Pipeline

The goal of the pipeline is not to train a new classifier. The goal is to produce a verbal explanation of a fixed TSC's behaviour, a compact, natural-language description of what distinguishes each class, derived from a small set of representative examples. Given a reference classifier f, we first select a small set of class prototypes and then use an LLM as a verbal translation layer: it receives the labelled prototypes as visual inputs and produces class-wise natural-language rules. The rules are then evaluated by forward simulation: a subject receives the explanation and classifies held-out time series, and we measure agreement with f.

This framing matters because prototypes and rules place different cognitive demands on the user. A prototype explanation asks the user to infer class regularities from curves; a rule explanation externalises that inference step into explicit verbal predicates such as "early baseline near zero" or "sharp recovery after the first valley" descriptions that can be read, remembered, and applied without memorising images.

3.1 Prototype Selection

We select k representative time series per predicted class of the reference classifier. Prototypes are selected class-wise with k-medoids using dynamic time warping (DTW) as the distance measure [10]. Selection is performed on the training split; held-out test instances are never used for prototype selection. All series are normalised to zero mean and unit variance before DTW computation. Class imbalance is handled implicitly by selecting prototypes class-wise independently, so each class contributes exactly k prototypes regardless of its frequency.

All selection and evaluation uses the labels assigned by the reference classifier, not the original dataset labels. The task is therefore model imitation, not ground-truth classification. This choice is important for interpretability: the explanation should teach the model's behaviour that the user is trying to understand, not a potentially different notion of class membership.

3.2 Rule Generation

The rule-generation LLM receives the labelled class prototypes as line-plot images. The prompt instructs it to compare the classes and produce concise rules for each class, written as class-wise subrules denoted R1, R2, etc. Each subrule describes one main concept and may use either an approximate numeric comparison or a descriptive temporal-shape term. The full prompt is given in Appendix B.

The generated rules are deliberately not restricted to executable symbolic syntax. A strict condition such as "at time step t, value $> \theta$" is easy to evaluate programmatically but less natural for a non-expert reader. We instead allow fuzzy temporal language. Terms like *early*, *plateau*, *steep drop*, *slow recovery* are the kind of language users can read, remember, and apply without specialist

training. Early experiments with a less-constrained prompt showed that LLMs tend to produce long, overlapping logical clauses. The current multi-step prompt (Appendix B) manages to enforce an atomic rule structure.

This flexibility also introduces a risk: a rule that is too vague may sound right but fail to support consistent classification. The evaluation, therefore, tests applicability to unseen instances, not just plausibility.

3.3 Using Rules for Forward Simulation

We evaluate two teaching formats. The prototype-only format shows the selected prototypes and asks the subject to classify new instances. The prototype-plus-rules format shows the same prototypes together with the LLM-generated class rules. In both cases, the subject sees the held-out time series and predicts the classifier label.

The same forward-simulation protocol is used with an LLM proxy and with human participants. The LLM proxy gives a scalable first check of whether the rules contain discriminative information. The human study gives a small human-grounded check of whether the rules also support understanding for non-domain experts.

4 Experimental Setup

We use four UCR datasets [4] (see Table 1) for the LLM proxy experiments and two of them for the human study. The human study uses Chinatown and ECG200 because they give a useful contrast: Chinatown has a relatively clear visual distinction, while ECG200 is harder to separate by visual inspection.

Table 1. Datasets used in the experiments. Sony1 denotes SonyAIBORobotSurface1.

Dataset	Length	Classes	Balanced	Proxy	Human	Role in the experiments
Chinatown	24	2	No	Yes	Yes	Simple visual structure
UMD	150	3	Yes	Yes	No	Visual task; high human accuracy
ECG200	96	2	No	Yes	Yes	Difficult; high variance
Sony1	70	2	Yes	Yes	No	Difficult visual separation

The reference classifier is MiniRocket [5] trained on the UCR train split. For each dataset, we first obtain the classifier predictions on the test split and then select prototypes within each predicted class from the training data. Each

explanation condition uses k prototypes per class. The MiniRocket classifiers achieve competitive accuracy on these datasets (98% on Chinatown, 94% on UMD, 96% on ECG200 and 90% on SonyAIBORobotSurface1). The pipeline's task is to explain the classifier's behaviour, not to validate its accuracy.

All LLM calls use GPT 5.1 with High reasoning. The implications for evaluation bias are discussed in Section 7.

We also report a previous human prototype-only baseline from Håvardstun et al. [12] as contextual reference. That study evaluated time-series simplifications using a human-grounded forward-simulation task: participants were shown labelled class prototypes and then asked to classify 10 held-out instances. We chose results where the participants saw the original prototype samples. We therefore include its reported human accuracies for the same four datasets in Table 3. This row is not part of the controlled comparison in the present paper, as it comes from a separate study with different participants and a different experimental goal.

4.1 LLM Proxy Validation.

The proxy evaluation compares direct prototype prompting against rule-based prompting. In the direct condition, the LLM receives the prototypes and a held-out test series and predicts the classifier label. In the rule-based condition, an LLM first generates rules from the prototypes; a separate application prompt then asks an LLM proxy to classify held-out series using the generated rules.

The direct prototype baseline uses $k = 3$ prototypes per class, averaged over 10 runs, where each run evaluates the LLM on 100 randomly sampled held-out instances. The rule-based condition uses the same prototype budget with 3 runs (each regenerating a fresh ruleset) and also evaluates on 100 randomly sampled held-out instances per run.

This proxy experiment is not a substitute for a human evaluation. It is a screening test. If the rules cannot help an LLM apply the class distinction, they are unlikely to be useful teaching material. If they do help, the rules still need a human-grounded check.

4.2 Human Survey and Interview.

The human study is an exploratory, small-sample evaluation. It uses 20 computer science (CS) graduate students or similar technically trained participants. Participants are not domain experts in time series and do not have any domain information on the datasets. Each participant completes two classifier-imitation tasks: one on Chinatown and one on ECG200.

The study compares prototypes only against prototypes plus LLM-generated rules. In the rules condition, we use the best-performing ruleset from the $k = 3$ runs generated in the proxy evaluation (Section 5). Each task shows three prototypes per class and asks the participant to classify 10 held-out instances. Participants also give a confidence score after each prediction on a 1–5 scale.

The study uses four counterbalanced groups (see Table 2), each with five participants. This design reduces task-order effects and avoids a contrast bias in which a participant who first sees more information might attribute worse performance in the reduced-information condition to the explanation rather than to the task.

Table 2. Counterbalanced human-study task order.

Group	Task 1	Task 2
A	Chinatown, prototypes only	ECG200, prototypes plus rules
B	Chinatown, prototypes plus rules	ECG200, prototypes only
C	ECG200, prototypes only	Chinatown, prototypes plus rules
D	ECG200, prototypes plus rules	Chinatown, prototypes only

Before the tasks, participants reported familiarity with ML, time-series plots, and XAI or rules. After each task, they answer explanation-quality questions on a 5-point Likert scale. After both tasks, we conduct a short interview asking what visual features they used, whether the rules helped, whether any rule was confusing, and whether they preferred prototypes only, rules only, or both. Full survey layout is available in Appendix C.

A limitation is that all participants are CS graduate students, who are likely more comfortable parsing structured text and plots than other regular users.

4.3 Metrics

For the proxy and human classification tasks, the main metric is agreement with the reference classifier. We also report confidence for human predictions. For the post-task questionnaire, we report a quality composite and, where relevant, individual patterns. The human study statistics are descriptive because the sample is small. We report confidence intervals and a paired participant-level check, but we do not treat the survey as a definitive user study.

For rule quality, we additionally inspect word count, number of rules, and rule style. We also use an exploratory embedding analysis of generated rulesets as support for qualitative coding, but not as a primary performance metric.

5 Results

Table 3 shows the main proxy-forward-simulation results for $k = 3$. The first row reports the external human prototype-only baseline from Håvardstun et al. [12]. We include it only as context for how humans performed when taught with prototypes in a related forward-simulation study. The two LLM rows correspond to the direct-prototype condition and the generated-rule condition.

Table 3. Forward-simulation accuracy for $k = 3$ prototype teaching and generated rules. Sony1 denotes SonyAIBORobotSurface1.

Configuration	Chinatown	UMD	ECG200	Sony1
Human prototype baseline [12]	66%	96%	72%	72%
LLM, prototypes only	$70\% \pm 8\%$	$38\% \pm 6\%$	$70\% \pm 19\%$	$58\% \pm 2\%$
LLM, rules from prototypes	$97\% \pm 3\%$	$92\% \pm 1\%$	$67\% \pm 12\%$	$57\% \pm 11\%$

The rule-based condition substantially improves over direct prototype prompting in Chinatown and UMD. In Chinatown, the LLM successfully identifies the early-region contrast between classes and encodes it as a precise verbal criterion. On UMD, the rule condition recovers much of the structure that the direct prototype prompt fails to exploit. This suggests that the LLM does not spontaneously extract discriminative features when classifying directly from prototypes, but can do so when explicitly prompted to verbalise them as rules.

The results are not uniform. On ECG200 and SonyAIBORobotSurface1, rules do not improve the average proxy accuracy. These datasets remain difficult under both direct and rule-based prototype prompting. This is useful evidence for the paper's main position: LLM-generated rules are a promising teaching representation, but they are not a general solution. The benefit depends on whether the prototypes exhibit a class distinction that the LLM can express as a stable, applicable rule.

Additional experiments show that increasing the number of prototypes does not consistently improve proxy accuracy. The full k-sensitivity results are reported in Appendix A.

The human study provides a user-grounded check of whether the generated rules support genuine understanding, beyond what the LLM proxy can assess. Across 20 participants and 40 participant-task observations, mean agreement with the reference classifier increases from 0.784 with prototypes only to 0.861 with prototypes plus rules. Perceived confidence increases markedly from 2.902 to 3.942, and perceived explanation quality from 2.750 to 3.700. Table 4 summarises the full results.

Participant background provides one additional clue about when the explanation format is effective. Among the three self-reported familiarity measures collected in the pre-task questionnaire, only prior experience reading time-series plots was meaningfully associated with performance. Time-series plot familiarity correlated positively with mean classification accuracy ($r = 0.506$, $p = 0.023$), whereas general ML familiarity ($r = 0.087$, $p = 0.716$) and prior familiarity with XAI/rule-based explanations ($r = 0.087$, $p = 0.717$) showed no such relationship. This suggests that success in the task depends on prior exposure to time series rather than on prior experience in machine learning or explainability methods.

The dataset-level pattern is more informative than the aggregate average (Figure 1). In Chinatown, accuracy increases from 0.798 to 0.930. This matches the intended cognitive story: the rule text provides participants with a concise

Table 4. Human-study results by explanation condition. Confidence and quality are on 1 to 5 scales.

Condition	Tasks	Accuracy	Confidence	Quality composite
Prototypes only	20	0.784 [0.729, 0.839]	2.902 [2.452, 3.319]	2.750 [2.464, 3.029]
Prototypes plus rules	20	0.861 [0.811, 0.905]	3.942 [3.667, 4.214]	3.700 [3.407, 3.964]

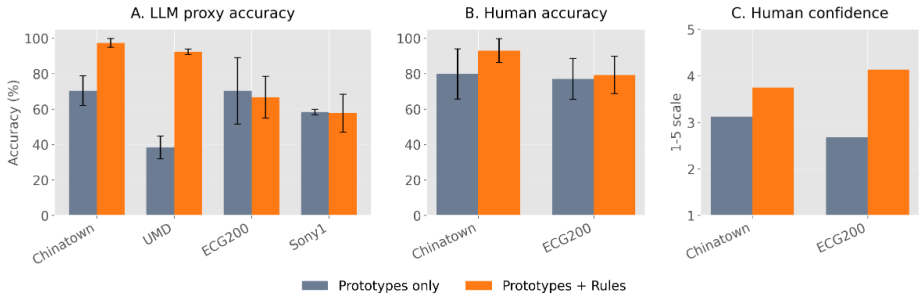

Fig. 1. Results overview. Panel A compares LLM proxy forward-simulation accuracy under direct prototype prompting and rule-based prompting. Panels B and C summarise the human study: rules improve classifier-imitation accuracy on Chinatown and increase confidence on both datasets, while ECG200 accuracy remains unchanged—illustrating the dataset-dependency of the verbal translation layer.

description of what to look for, so they do not need to memorise the prototype curves as images. On ECG200, the gain is much smaller, increasing only from 0.770 to 0.793, while confidence still rises substantially from 2.679 to 4.133. This dissociation, where we see a large confidence gain with only a small accuracy gain, should make us cautious. In the context of this study, the effect appears to reflect participants' feeling more structured and guided. However, in a high-stakes deployment, explanations that increase confidence more than correctness could promote overconfidence in ambiguous cases. This remains an important limitation of the approach on harder datasets and a point future work should address explicitly.

A participant-level paired summary also favours the rules condition. The mean paired delta is 0.077, with 13 participants improving, 4 unchanged, and 3 decreasing. A Wilcoxon signed-rank test gives $p = 0.017$. Given the small sample, we still treat this primarily as evidence of a promising trend rather than a definitive population-level conclusion, but the updated result is more supportive than in the earlier pilot.

The post-task ratings are consistent with the same interpretation. Rules score higher on understanding, sufficiency, ease of application, visible-pattern match, and confidence. The only less positive rating concerns information load. After reversing the "too much detail" item, the score decreases slightly from 4.150 to

4.000 with rules. In other words, participants found the rules somewhat more detailed, though the overall score remains high.

5.1 Qualitative Human Feedback

The interviews help explain when and why the rules were useful. Participants mostly reported using shape-level features: slopes, dips, recovery speed, bumps, and start/end behaviour. Most respondents said the rules helped them focus on the relevant part of the plot or clarified the contrast between classes. The preferred format was the combination of prototypes and rules rather than rules alone. This is consistent with the idea that the verbal translation layer *complements* rather than replaces the visual prototype.

The main criticism was not the rule length but vagueness near a decision boundary. Participants noticed terms such as "often", "tends to", or qualitative descriptions of drop magnitude. This is the central tension in the approach: fuzzy natural-language rules are more readable than rigid symbolic conditions, but they must still be precise enough to support consistent application. This tension connects directly to the rule analysis in the next section.

6 Rule Analysis

A rule can be predictive without being useful to a human, and readable without being precise enough to apply consistently. We therefore analyse the generated rules along three dimensions: length, linguistic form, and the type of temporal pattern they describe.

6.1 Rule Length and Complexity

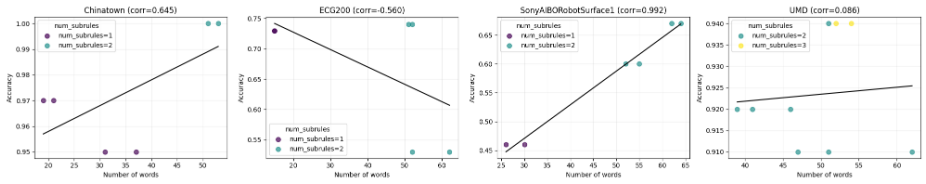

Fig. 2. Relation between ruleset word count and LLM-proxy accuracy. Each point corresponds to a class-level ruleset from one run. The relation is not monotonic: concise rules can be highly effective, but longer rules are not automatically better.

A natural first proxy for rule complexity is word count. Shorter rules should be easier to read and remember, but too much compression can remove the temporal detail needed for classification. Conversely, longer rules can describe more cases, but may become harder to use as teaching material.

Figure 2 shows the relation between ruleset word count and LLM-proxy accuracy.

The plot does not give a simple "more words give better rules" interpretation. Chinatown achieves high performance with compact early-region rules, whereas ECG200 and SonyAIBORobotSurface1 often require longer descriptions but do not achieve the same gain. Word count is at best a rough proxy for complexity: what matters is whether the rule identifies the discriminative temporal region and expresses it in a form that can be applied consistently. Complexity for human-facing explanations should be treated as a combination of length, number of conditions, temporal regions referenced, and linguistic vagueness.

6.2 Exploratory Rule Semantics

To complement the length and style analyses, we ran an exploratory embedding analysis across the 27 class rulesets from Chinatown, ECG200, SonyAIBORobotSurface1, and UMD. Each class ruleset was embedded using the all-mpnet-base-v2 sentence-transformer model [19] and clustered with k-means. The best silhouette score selected three clusters. We use this analysis as a qualitative aid, not as a primary performance result.

The clustering suggests three coarse semantic families, though the small number of rulesets (27) and the exploratory nature of the analysis mean these should be treated as a qualitative characterisation rather than a quantitative result. The first family, *early-baseline rules*, contains the Chinatown rulesets (mean proxy accuracy 0.973) and describes whether the early part of the series stays close to zero or starts from a higher level. The second family, *shape-trajectory rules*, contains the UMD and SonyAIBORobotSurface1 rulesets, as well as some ECG200 rulesets (mean proxy accuracy 0.753), describing changes across several regions, such as valleys, plateaus, and tails. The third family, *middle-threshold rules*, contains two ECG200 rule sets (mean proxy accuracy 0.730) that use threshold-like descriptions of the mean value in the middle segment.

This characterisation is useful for understanding the rule space, but the cluster-level accuracy differences should not be over-interpreted; they reflect both the type of rule and the inherent difficulty of the dataset, which are confounded here. Early-baseline rules are compact because the discrimination is simple, not simply because the rule type is inherently better.

A small stability check gives a second view of rule quality. Some dataset-class pairs produce semantically similar rules across runs. For example, UMD class 1 has a mean pairwise cosine similarity of 0.824. ECG200 is less stable, with mean similarities 0.493 and 0.523 for its two classes. This instability matters for interpretability. If repeated generations describe the same class in different ways, the method may still find useful rules, but it is less reliable as a fully automatic rule generator.

6.3 Representative Rules

The best Chinatown rules illustrate the strongest case for the pipeline. The prototypes contain a clear early-region difference, and the LLM turns this visual contrast into a short rule about baseline level and early variation.

Chinatown, high-performing ruleset ($k = 3$, proxy accuracy $= 1.00$)

Class 0

R1: In the early region, average values are clearly above zero (around 0.1 or higher); the series starts at a moderate level before dipping.
R2: Within the early region, there is a strong downward drop (the first value is at least about 0.15 higher than the early minimum), giving a pronounced descending shoulder.

Class 1

R1: In the early region, average values stay very close to zero (below about 0.1); the series begins on a low baseline.
R2: Within the early region, variation is small (the difference between the highest and lowest early values is less than about 0.15), so this part appears almost flat near zero.

These rules are useful because they reduce the prototype comparison to a small number of visible checks. They do not ask the user to remember the whole curve. They tell the user where to look and what contrast to apply.

ECG200 shows a more difficult case. The best available ruleset describes recovery after an early valley and the behaviour of the late segment.

ECG200, best available ruleset ($k = 3$, proxy accuracy $= 0.74$)

Class 0:

R1: After the early valley, the early middle region stays close to that minimum and well below the later plateau, forming a long, gradual upward slope (slow recovery).
R2: The late region sits at or above the middle level, often showing an upward bump near the end rather than a smooth, sustained decline.

Class 1:

R1: After the early valley, the early middle region jumps quickly up to near the plateau level, forming a sharp upward step (fast recovery).
R2: From the middle into the late region the series tends to drift downward, or any late peak is brief and followed by a return to the middle plateau level.

This ruleset is still human-readable, but it is harder to apply. It refers to several regions and uses boundary-sensitive terms such as “often”, “sharp”, and “tends”. This matches the human-study feedback, where participants generally found the rules useful, but the main source of confusion was the vagueness around thresholds and borderline shapes.

In sum, the verbal translation layer succeeds when the class difference is concentrated in a clear, localised temporal region, but fails when discrimination requires integrating multiple shape cues or when the class boundary is inherently fuzzy. Which is precisely the conditions that separate Chinatown and UMD from ECG200 and SonyAIBORobotSurface1.

7 Discussion

The proxy and human results provide evidence that LLMs can generate natural-language rules containing discriminative information. As we have mentioned in the paper, this is not trivially guaranteed, since a rule could be fluent and still fail to separate classes. We show that on datasets with a clear, localized temporal contrast, such as Chinatown or UMD, rules improve substantially over direct prototype prompting.

However, the benefit is not universal. On ECG200 and SonyAIBORobot-Surface1, rules do not improve over direct prototype prompting, and human accuracy on ECG200 is mostly unchanged between conditions. We think these errors come from an inherent vagueness-precision trade-off. When discrimination requires integrating multiple temporal regions or the class boundary is fuzzy, the LLM produces rules that participants can read but cannot apply consistently. Terms such as “often”, “tends to”, and “sharp” are semantically meaningful but underspecified, especially near the decision boundary. This is somewhat shown on the embedding analysis in Section 6.2.

Still, our methodology comes with many limitations. First, the results reinforce that an LLM-based proxy is not a substitute for human evaluation. Since rule generation and proxy evaluation use the same LLM family, the proxy may favour rules for that model rather than for its usefulness. Second, rule stability remains an issue. On ECG200, repeated generation produces semantically different rule sets (mean cosine similarity 0.49 and 0.52). This means that selecting a good rule could still be guided by a proxy evaluation step. Finally, larger studies with more datasets and more human participants are needed before making broader claims about generalizability.

7.1 Position: Rules as an Intermediate Representation

The limitations above do not imply that natural-language rules are a poor representation. Rather, they point to a trade-off that is central to the representation itself. LLM-generated rules are often vague. This vagueness can reduce classification reliability near difficult decision boundaries. But it makes the rules closer to how humans often describe visual and temporal patterns. Human explanations

rarely take the form of fully specified symbolic descriptions. They more often refer to approximate shapes, relative changes, salient regions, and qualitative tendencies.

This suggests a different role for LLM-generated rules. They should not be treated as final executable classifiers, but as human-centred descriptions. Their value may lie in making prototype differences easier to discuss, inspect, and refine. In this sense, vagueness is not simply noise to be removed. It is also part of what makes the representation readable and cognitively natural.

The main challenge is therefore not to eliminate this vagueness. A useful system should preserve the qualitative form of human descriptions while grounding them when needed. For example, these vague expressions could be linked to specific regions.

We think this direction deserves further study. LLMs may be useful not because they produce exact symbolic rules, but because they can generate intermediate descriptions that sit between raw prototypes and formal decision rules. Exploring this space requires better measures of rule interpretability, stability, complexity and human usability, as well as comparisons against manually written rules, region-statistic rules, and shapelet-based symbolic rules.

8 Conclusion

We have studied LLMs as a verbal translation layer for time-series classification explanations. Given a small set of labelled class prototypes, an LLM generates concise natural-language rules. These rules are then evaluated by an LLM proxy and by human participants in a forward-simulation task.

On datasets with clear, localised temporal contrasts, LLM-generated rules outperform direct prototype prompting and improve both human accuracy and perceived explanation quality. These rules manage to make the discriminative region explicit and express the relevant contrast in words, relieving the user of the need to infer the rule visually. However, on harder datasets, the method does not provide the same benefit. The user study suggests that this failure may come from the vagueness in the descriptions of the rules.

The contribution is therefore preliminary empirical evidence that LLM-based rule generation can be a useful component of XAI for time series under favourable conditions, together with an initial characterisation of when and why it fails. More broadly, the results suggest that the interest of these rules lies in their closeness to human description. They express temporal differences through approximate shapes, salient regions, relative changes, and qualitative tendencies rather than through fully specified descriptions. This makes them imperfect as executable rules, but potentially useful as human-centred intermediate representations. The present work is a first step toward understanding when this kind of representation is useful, and what is needed to make it reliable.

Acknowledgments. During the preparation of this work, the authors used Opus 4.8 and GPT-5.5 for drafting sections that were later edited. We take full responsibility for all content.

References

1. Adel, T.: Explaining time series predictions via relevant shapelets. In: Proceedings of the 8th International Conference on Advances in Artificial Intelligence (ICAAI). pp. 212–217 (2024). https://doi.org/10.1145/3704137.3704190
2. Aksu, T., Liu, C., Saha, A., Tan, S., Xiong, C., Sahoo, D.: Xforecast: Evaluating natural language explanations for time series forecasting. CoRR **abs/2410.14180** (2024). https://doi.org/10.48550/ARXIV.2410.14180, https://doi.org/10.48550/arXiv.2410.14180
3. Chen, C., Li, O., Tao, D., Barnett, A., Rudin, C., Su, J.K.: This looks like that: Deep learning for interpretable image recognition. In: Advances in Neural Information Processing Systems. vol. 32 (2019)
4. Dau, H.A., Bagnall, A., Kamgar, K., Yeh, C.C.M., Zhu, Y., Gharghabi, S., Ratanamahatana, C.A., Keogh, E.: The ucr time series archive. IEEE/CAA Journal of Automatica Sinica **6**(6), 1293–1305 (2019). https://doi.org/10.1109/JAS.2019.1911747
5. Dempster, A., Schmidt, D.F., Webb, G.I.: Minirocket: A very fast (almost) deterministic transform for time series classification. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. p. 248–257. KDD '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3447548.3467231, https://doi.org/10.1145/3447548.3467231
6. Doshi-Velez, F., Kim, B.: Towards a rigorous science of interpretable machine learning (2017)
7. Gee, A., Garcia-Olano, D., Ghosh, J., Paydarfar, D.: Explaining deep classification of time-series data with learned prototypes. CEUR workshop proceedings **2429**, 15–22 (08 2019)
8. Guidotti, R., Monreale, A., Spinnato, F., Pedreschi, D., Giannotti, F.: Explaining any time series classifier. In: 2020 IEEE second international conference on cognitive machine intelligence (CogMI). pp. 167–176. IEEE (2020)
9. Hase, P., Bansal, M.: Evaluating explainable AI: Which algorithmic explanations help users predict model behavior? In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 5540–5552. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.491, https://aclanthology.org/2020.acl-main.491/
10. Holder, C., Guijo-Rubio, D., Bagnall, A.: Clustering time series with k-medoids based algorithms. In: Advanced Analytics and Learning on Temporal Data: 8th ECML PKDD Workshop, AALTD 2023, Turin, Italy, September 18–22, 2023, Revised Selected Papers. p. 39–55. Springer-Verlag, Berlin, Heidelberg (2023). https://doi.org/10.1007/978-3-031-49896-1_4, https://doi.org/10.1007/978-3-031-49896-1_4
11. Huang, Q., Chen, W., Bäck, T., van Stein, N.: Shapelet-based model-agnostic counterfactual local explanations for time series classification. arXiv preprint arXiv:2402.01343 (2024)
12. Håvardstun, B., Marti-Perez, F., Ferri, C., Telle, J.A.: Evaluating simplification algorithms for interpretability of time series classification (2025), https://arxiv.org/abs/2505.08846
13. Lage, I., Chen, E., He, J., Narayanan, M., Kim, B., Gershman, S.J., Doshi-Velez, F.: Human evaluation of models built for interpretability. In: Proceedings of the AAAI Conference on Human Computation and Crowdsourcing (2019)

14. Lan, T., Le, H.D., Li, J., He, W., Wang, M., Liu, C., Zhang, C.: Axis: Explainable time series anomaly detection with large language models (2025)
15. Lines, J., Davis, L.M., Hills, J., Bagnall, A.: A shapelet transform for time series classification. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 289–297 (2012)
16. Mavrepis, P., Makridis, G., Fatouros, G., Koukos, V., Separdani, M.M., Kyriazis, D.: Xai for all: Can large language models simplify explainable ai? (2024)
17. Ming, Y., Xu, P., Qu, H., Ren, L.: Interpretable and steerable sequence learning via prototypes. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 903–913 (2019)
18. Mochaourab, R., Venkitaraman, A., Samsten, I., Papapetrou, P., Rojas, C.R.: Post hoc explainability for time series classification: Toward a signal processing perspective. IEEE Signal Processing Magazine **39**(4), 119–129 (2022)
19. Song, K., Tan, X., Qin, T., Lu, J., Liu, T.Y.: Mpnet: Masked and permuted pre-training for language understanding. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 16857–16867. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper_files/paper/2020/file/c3a690be93aa602ee2dc0ccab5b7b67e-Paper.pdf
20. Theissler, A., Spinnato, F., Schlegel, U., Guidotti, R.: Explainable AI for time series classification: a review, taxonomy and research directions. Ieee Access **10**, 100700–100724 (2022)
21. Warczyński, J., Lango, M., Dušek, O.: Leveraging large language models for building interpretable rule-based data-to-text systems. In: Proceedings of the 17th International Natural Language Generation Conference. pp. 622–630 (2024)
22. Ye, L., Keogh, E.: Time series shapelets: A new primitive for data mining. Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining pp. 947–956 (2009)

A Effect of the Number of Prototypes

In Table 5, we can observe how varying the value of k affects the performance of the LLM. Increasing the number of prototypes does not universally improve outcomes. For instance, performance degrades or plateaus in the Chinatown and ECG200 datasets. In contrast, k has a noticeable effect on the UMD dataset, where a single prototype ($k = 1$) is insufficient to capture class boundaries.

The results for SonyAIBORobotSurface1 are noteworthy: performance is stable between $k = 1$ and $k = 3$, but degrades significantly at $k = 5$. This suggests that redundant prototypes can make the class boundary less clear, and that the optimal k depends on the coherence of the discriminative structure in the dataset.

Table 5. Forward-simulation accuracy for the rule-based LLM with $k = 1$, $k = 3$, and $k = 5$. Sony1 denotes SonyAIBORobotSurface1.

k	Chinatown	UMD	ECG200	Sony1
1	$91\% \pm 1\%$	$69\% \pm 6\%$	$70\% \pm 10\%$	$60\% \pm 11\%$
3	$97\% \pm 3\%$	$92\% \pm 1\%$	$67\% \pm 12\%$	$57\% \pm 11\%$
5	$93\% \pm 3\%$	$92\% \pm 1\%$	$71\% \pm 7\%$	$47\% \pm 3\%$

B Prompt

Prompt used for rule extraction

You are a time-series classification expert analysing labelled prototypes for classes `{classes}` (`{num_images}` prototypes per class).
Follow these steps:

Step 1 — **Analyse differences between classes:** Identify which regions (early, middle, late) differ most, and whether differences are best described by thresholds, trends, peaks/troughs, plateaus, or temporal shape patterns.

Step 2 — **Build a feature summary:** Determine which of the following are most discriminative between classes:
- Region statistics (mean/min/max in early, middle, late)
- Trends (rising/falling)
- Peaks, troughs, plateaus
- Relative differences between regions

Step 3 — **Generate human-readable classification rules:** Each rule must:
- Describe one main concept
- Use either a numeric comparison or a descriptive shape term (e.g., upward peak, broad plateau, falling trend, rising tail)

– Be as concise as possible
Avoid: redundant rules, conditions shared by all classes, mathematical notation, ambiguity.

Step 4 — Validate internally: Check that the rules correctly distinguish the prototypes. Refine any non-discriminative rules. Prefer the smallest rule set that separates all classes.

Output format (strictly):
Class `<label>`:
R1: ...
R2: ...
...

The initial iteration of the prompt utilised a lightweight design consisting of only three basic constraints: providing a human-understandable rule for each class, ensuring each sub-rule covered only a single condition, and enforcing a specific output layout.

However, this approach resulted in weak prompt adherence. The model consistently generated long, convoluted, and overlapping rules with multiple combined conditions per line, making them difficult to parse.

To mitigate this behaviour, the prompt was systematically expanded into a multi-step workflow. The final prompt strictly enforces atomic rule constraints, forbids mathematical jargon, and integrates an internal validation loop to guarantee clarity.

C Survey Layout

Participants first completed a background form (1–5 familiarity scales for ML, time-series plots, and XAI). They then completed two classifier-imitation tasks (Chinatown and ECG200, counterbalanced as in Table 2), each with three prototypes per class and 10 held-out instances. After each prediction, participants reported confidence on a 1–5 scale. The target label was always the reference classifier label.

Post-task ratings (5-point Likert, strongly disagree to strongly agree). Positive items: I understood the difference between the classes. / The information shown was sufficient. / The explanation was easy to apply. / The explanation matched visible patterns in the plots. / I felt confident in my classifications. *Diagnostic items (interpreted separately or reverse-coded):* The explanation was too vague. / The explanation contained too much detail.

Final interview (semi-structured). What visual features did you use? / Did the rules help, and which part? / Were any rules confusing or vague? / Did you rely more on prototypes or rules? / Would you prefer prototypes only, rules only, or both? / Did the rules help you understand the classifier, or mainly help answer the test?

The full survey files and scoring scripts are available in the project repository. Figure 3 shows a representative page from the prototypes-plus-rules condition.

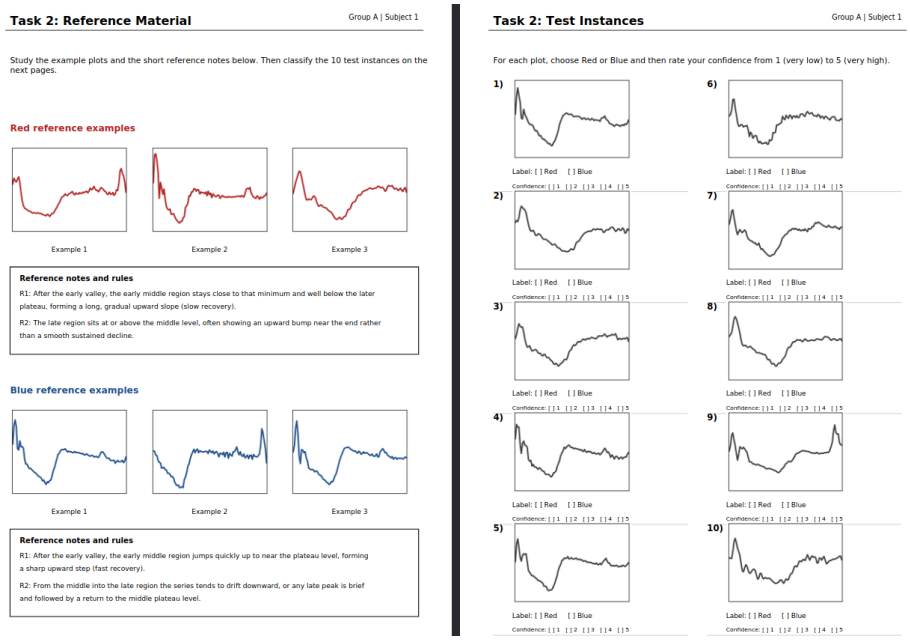

Fig. 3. Representative survey page shown to participants in the prototypes-plus-rules condition.