Requirements Engineering manuscript No.
(will be inserted by the editor)

Systematic LLM Prompt Development for Human-Centric Interpretation of Linear Temporal Logic

Authors

Received: November 24, 2025/ Accepted:

Abstract Formal methods for requirements engineering have existed for decades; yet, these techniques are rarely used if not required by certification because they are challenging for non-experts (e.g., novices and non-technical stakeholders in multi-disciplinary teams) to interpret and apply. To enable non-experts to participate in collaborative software teams, we envision using artificial intelligence (AI) to assist in interpreting formal notations. Our research investigates how and to what extent generative AI with large language models (LLMs) can be used to assist non-experts in interpreting formal requirements.

Prior work explored whether LLM-generated explanations could improve novices' comprehension of LTL, and while the results suggested promise, the investigation could have benefited from a rigorous analysis of correctness and a systematic construction of the associated LLM prompts. In this article, we conduct a deeper, systematic investigation of LLM-generated explanations. We curate an expanded corpus of 77 formulae with gold-standard explanations. We introduce a factor-level prompt engineering methodology, adding and assessing individual prompt fragments to identify the components that most strongly affect quality (i.e., correctness, clarity, and length). Using these prompts, we compare the explanations generated by six LLMs.

Our results show that prompt structure has a significant effect on overall explanation quality and that explanations vary widely across models. These findings provide empirical foundations for integrating LLM-generated explanations into RE practice and offer practical guidance on designing prompts that support non-experts in understanding formal specifications.

Address(es) of author(s) should be given

1 Introduction

Formal methods (FMs) and formal notations have been long established as powerful tools in requirements engineering (RE) [58]. The primary advantage of FMs is their ability to provide a mathematically precise specification, reducing ambiguity and enabling early verification of requirements. Techniques such as model checking, theorem proving, and formal refinement allow for automated analysis, reducing the risk of defects in the early stages of system development [70]. Moreover, formal specifications can facilitate reasoning about correctness properties, ensuring that critical system requirements (i.e., safety and security) are rigorously validated before implementation [14].

Despite their benefits, the mathematical nature of FMs can make them difficult for non-specialists to understand [33, 72, 31]. In practice, this leads to a gap between formal specifications and real-world applications, particularly when communicating with non-technical stakeholders. To address this, hybrid approaches that combine FMs with semi-formal techniques (e.g., controlled natural languages and graphical representations), have been explored to improve usability while retaining precision [38]. However, even in hybrid approaches that increase accessibility, the underlying formal semantics often remain opaque, leaving non-experts unable to fully interpret, evaluate, or validate formal specifications. As modern software development becomes increasingly interdisciplinary, effective communication between formal-methods specialists and other stakeholders is more critical than ever.

LLMs are also increasingly used to support formal reasoning tasks. A notable example is Rango, which integrates LLMs with retrieval-augmented proving to enhance proof synthesis in the Coq proof assistant [63].

By dynamically retrieving relevant premises and similar proofs, Rango improves automation; however, it provides no explanation for how proofs are constructed, leaving users to trust results without interpretive support. This highlights the broader trend that LLMs are becoming powerful assistants for generating or verifying formal artifacts, yet far less work examines their potential for explaining those artifacts to human readers.

Recent advances in generative AI, particularly large language models (LLMs), offer a promising new path. LLMs have demonstrated strong abilities in natural-language explanation, paraphrasing, and domain adaptation. Early findings show that LLMs can translate unstructured natural language into formal specifications [12, 17] and detect inconsistencies or dependencies in large requirement sets [1,66]. LLMs have also been evaluated for their ability to reason about formal languages directly. While their fluency approaches human-level, their accuracy in producing correct logical forms remains limited [44].

Halili et al. [32] explored whether LLM-generated explanations could support novices learning Linear Temporal Logic (LTL) and provided initial empirical evidence that LLM explanations could improve comprehension, confirming errors linked to well-known misconceptions in temporal logic [31]. Yet, these results raised several open research questions: How should prompts be engineered to reduce semantic ambiguity? Which prompt components drive linguistic quality and correctness? How consistently do different LLMs behave across explanation tasks? These open questions motivate the broader and deeper investigation we present in this article. LTL is widely used in formal RE, software verification, hardware design, and robotics [54]. Its syntax is compact, but even small structural changes have substantial semantic consequences, making it an ideal domain for studying whether LLMs can reliably produce explanations that support novices, interdisciplinary teams, and experts reviewing their own specifications.

1.1 Motivating Example

Consider the operations of a microwave: *start*, *cook*, and *stop*. One can write a Linear Temporal Logic (LTL) formula (see Section 2.1 for an overview) to describe the function of a microwave as in Formula 1, where once *start* is activated, the microwave should *cook* until the user presses *stop*. Yet, a non-expert stakeholder may give the same meaning to Formula 2 as Formula 1, not realizing the critical role of the parentheses. In Formula 2 the global operator G only applies to *start* and there is no connection with the consequent of the implication. Further, in Formula 2, the consequent of the until operator U (i.e., *stop*) depends on a future state of the antecedent (i.e., X *cook*). With explanations, we can help individuals better interpret the correctness of a formula.

$$G\ (start \implies X\ (cook\ U\ stop)) \tag{1}$$

$$G\ start \implies X\ cook\ U\ stop \tag{2}$$

1.2 Problem

Despite recent progress, LLMs exhibit notable limitations when applied to formal requirements. They often produce plausible-sounding but incorrect interpretations, struggle with complex nested expressions, and fail to distinguish subtle semantic differences in formal notations, as the choice of prompt format, training examples, and even the order of the training examples can cause accuracy to vary [74]. These challenges highlight the importance of systematic evaluation and careful application when leveraging LLMs as assistive tools in FMs. Some research suggests that combining LLMs with traditional FMs could enhance their reliability and effectiveness in practical applications [62,23].

Ferrari and Spoletini's roadmap for LLMs in formal RE [25] emphasizes that, while LLMs show promise in automating tasks such as elicitation, analysis, and validation, their use as *explanatory tools* for formal specifications is far less explored and substantially less understood. This roadmap identifies explanation, semantic alignment, and correctness as open research challenges, particularly when natural language phrasing diverges from formal semantics. These unresolved issues directly motivate the need for systematic studies of explanation behavior under controlled conditions, and are especially salient in LTL, where small syntactic differences (e.g., operator scope, parentheses) yield large semantic consequences, and where humans already exhibit systematic misconceptions [31]. Therefore, before LLM-generated explanations can be responsibly integrated into RE practice, we require: (1) empirically grounded methods for evaluating LLM-generated explanations of formal artifacts; (2) systematic prompt-engineering strategies tailored to known semantic pitfalls in LTL; and (3) cross-model evidence clarifying how different LLMs behave under varying prompt structures.

Our work addresses these needs by developing, analyzing, and refining prompts for generating explanations of LTL formulae, evaluating their quality across multiple metrics, and comparing their performance across several LLMs.

1.3 Contributions

We investigate how and to what extent generative AI with LLMs can be used to assist non-experts (i.e., stakeholders and novices) in interpreting formal requirements written in Linear Temporal Logic (LTL) [54]. LTL is widely used in formal RE and is applicable to safety-critical domains for ensuring that systems operate correctly over time. Despite this, many misconceptions exist about the interpretation of LTL and its English meaning [31]. This makes LTL an ideal case study for examining whether LLMs can reliably produce explanatory text that supports comprehension without sacrificing precision.

This article substantially broadens both the methodological depth and empirical grounding of the contributions Halili et al. [32]. We make the following research contributions.

- Expanded and Diversified Corpus. We extend the original 50 formula dataset to a more diverse and challenging 77 formula corpus, adding 27 new formulae. This corpus supports a more comprehensive evaluation of explanation quality.
- Systematic, Factor-Level Prompt Engineering. We introduce a structured prompt-engineering methodology in which individual components or factors of a prompt are isolated, evaluated independently, and combined. This produces a refined aggregate prompt that substantially improves quality.
- Multi-Metric Linguistic Evaluation. We conduct an expanded analysis of LLM-generated explanations by evaluating quality in terms of correctness, clarity, and explanation length. This analysis goes beyond that of Halili et al. [32] and provides a more nuanced understanding of how explanation quality varies across prompt configurations.
- Cross-Model Comparison of LLM Behavior. We compare six LLMs, identifying model-specific strengths and weaknesses, and investigate whether LLM choice in prompt development impacts applicability of prompts quality across LLMs.
- Refinement of Correctness Semantics. We discuss how linguistic vagueness and temporal boundary conditions contribute to explanation errors.

This research addresses the longstanding challenge of making formal requirements more accessible without sacrificing their precision, with implications for education, industrial practice, and tool development in requirements engineering.

1.4 Organization

While the remainder of this paper is organized into eleven sections, we grouped the sections by lines of inquiry. Sections 2 and 3 provide the reader with the specific constructs, notations, definitions, and datasets used throughout the paper. Sections 4 and 5 introduce and evaluate our initial prompt (LLM_Init) through a classroom study. Sections 6 and 7 consider a second improved prompt (LLM_Exp) and compare it with LLM_Init, but critique it as wanting. Section 8 investigates which factors contribute to prompt quality and produces an aggregate prompt (LLM_Agg). Section 9 compares six LLMs and probes whether improvements made in creating LLM_Agg apply to other LLMs. Finally, Sections 10, 11 and 12 conclude the paper by discussing limitations and implications of this work, and positioning it within the literature.

2 Background

2.1 Linear Temporal Logic (LTL)

LTL is a modal temporal logic that extends propositional logic with operators expressing properties over paths in time [54]. Introduced to reason about concurrent programs, LTL has become fundamental in formal verification, particularly for specifying behavioral properties of reactive and concurrent systems [14].

LTL formulae are built upon atomic propositions using logical operators (i.e., and &, or |, implication $\rightarrow$, not !) and temporal operators. We use four LTL temporal operators in our work:

- Xa Next: The property a holds in the next state.
- Fa Eventually: The property a will hold at some future state.
- Ga Globally: The property a holds at every state.
- aUb Until: The property a holds until b becomes true.

The semantics of LTL are defined over infinite sequences of states (traces), where each state represents a set of atomic propositions that hold true at that point. For example, the formula $G(request \rightarrow F\ response)$ specifies that globally (at every state), if a *request* occurs, then eventually a *response* will follow—capturing behavioral properties in a reactive system.

2.2 Prompting Strategies

The interaction with LLMs is predominantly facilitated through *prompting*, where users provide specific information to the LLM to guide the model toward generating desired outputs. Several prompting strategies have emerged in the literature [45,68,21,67]:

- *Zero-shot prompting*: The model is asked to perform a task without examples.
- *One-shot prompting*: The model is provided with a single example before performing the task.
- *Few-shot prompting*: Multiple examples are provided to establish a pattern.
- *Chain-of-thought (CoT) prompting*: The model is guided to break down complex reasoning into intermediate steps.
- *Tree-of-thought prompting*: Extends chain-of-thought by exploring multiple reasoning branches.
- *Role prompting*: The model is instructed to assume a specific role or persona; in this case, different LLM instances can assume distinct personas to engage in collaborative tasks.

For tasks involving formal notations (e.g., LTL), structured prompts have proven effective, as they use the model's ability to recognize patterns and follow defined rules [65,68]. We use one-shot prompting, a guided decomposition strategy for eliciting reasoning, and role prompting strategies. Our approach differs from CoT prompting because we define an explicit structure for the reasoning steps within the prompt itself, guiding the model to follow our specified reasoning.

3 Preliminaries: Notation, Data, and Metrics

3.1 LTL Notation Standardization

There are multiple syntactic variations to represent LTL across the academic literature [6]. To prioritize notational consistency across the multiple stages of our project, we adopt a uniform alphabetic format (i.e., F, G, U, and X) for temporal operators and character format (i.e., &, |, !, =>) for logical operators, rather than alternative symbolic representations (e.g., $\Diamond$, $\Box$, $\circ$, $\wedge$, $\vee$, and $\neg$). This standardization was intended to make the notation more accessible and easier to parse by both LLMs and human readers. In our classroom study (see Section 5), we needed students to be able to write LTL formulae in plain text. In subsequent analysis, we needed the formulae to appear naturally across multiple tools and file formats (e.g., .csv). A downside to this approach is that implication => required two characters. For readability, we use $\rightarrow$ for implication in what follows.

3.2 50F Dataset and Gold Standard

To systematically evaluate the effectiveness of our prompts, we created a testing corpus of 50 LTL formulae, which we call the *50F dataset* in this paper. We selected 25 formulae from the dataset by Cherukuri et al. [13] and wrote an additional 25 formulae based on educational texts that teach LTL (e.g., [41]). We curated a diverse set of LTL formulae representing a spectrum of complexity levels. The test set included basic logical operators, simple temporal operators, compound formulae, and complex nested expressions. This range in aspects of LTL syntax allows for a comprehensive evaluation of LLM capabilities in this domain.

Two researchers collaborated to create a natural language rephrasing of each formula in the 50F dataset. Each one-sentence *formula summary* acted as our gold standard of a human readable explanation of the LTL formula, see Section 10 for a discussion of the limitations of our gold standard. Our 50F dataset and gold standards are available as part of our online open science supplement[1]. We use the 50F dataset in Section 5 and Section 7.

3.3 Expanded 77F Dataset

An early critique of the 50F dataset [32] was that it contained relatively simple formula. To mitigate this, we expanded the 50F dataset to include 27 formulae from the work of Greenman et al. [31], which we refer to as the 77F dataset. We used the 77F dataset to evaluate the generated LLM explanations in Sections 8–10.

3.4 Operationalization of Quality Construct

In this paper, we operationalize our construct of *quality* into three metrics: correctness, clarity, and length. We prioritize correctness over the other factors, as clarity is counterproductive if the clearly conveyed message is misleading. An ideal explanation being of minimal length without sacrificing correctness or clarity. Prior to calculating clarity and length, the LLM outputs are pre-processed to remove Markdown formatting for data in Section 6 and Section 7. We perform this step to mitigate the impacts of non-semantic visual tags on clarity metrics and word count, as we are most interested in the content of the explanation.

Correctness. In evaluating correctness, we consider the syntax and semantic meaning of each sentence in the explanation.

[1] See *https://github.com/ArtiRepo/REJ25N* for supplement.

Definition 1 (Correctness of an LTL Explanation) *Given an LTL formula and LLM explanation, the LLM explanation is correct if and only if (i) the refined natural language phrasing (i.e., the summary) is a correct interpretation of the LTL formula, and (ii) no individual statement in the explanation is objectively false.*

In order to score correctness, two researchers independently adjudicated each explanation as either *satisfying* or *violating* Definition 1. After ratings were complete, a third researcher arbitrated the disagreements in consultation with the original two raters (see Section 7.3 and Section 10.1 for calculation of inter-rater reliability).

Clarity. The field of linguistics defines quality as three criteria: (1) Fluency & Readability (*Readability*) [52]: Does the sentence sound natural and smooth? (2) Coherence & Logical Flow (*Coherence*) [8]: Does it maintain a clear structure and make sense? (3) Lexical & Syntactic Quality [7]: Is the word choice, grammar, and structure appropriate? Although slightly different than correctness, we argue that our definition of correctness above is a sufficient measure of the Lexical & Syntactic Quality of an explanation, and therefore exclude it from this analysis. Thus, we consider readability and coherence as proxies for clarity.

To compute readability, we use the Flesch Reading Ease score (FRE) [26] and the Dale-Chall (DC) score [18]. After considering the many scoring systems available in the literature, we decided to compare these two because they are examples of the two different types of metrics. The FRE listed in Formula 3 is directly computed based on the total number of words, syllables, and sentences. The highest FRE score is 121 and negative scores are possible, although most texts fit within the range of 10 to 100. A lower score indicates a higher reading level, with 60–70 considered as around an eighth or ninth-grade reading level. Unlike the FRE score, the Dale-Chall (DC) score compares the words present in the document with a list of easily understood words. It then computes a score indicative of grade level, typically ranging from 5 or lower (readable by fourth graders and lower), to 10 (college student understanding).

$$206.835 - 1.015\left(\frac{total\ words}{total\ sentences}\right) - 84.6\left(\frac{total\ syllables}{total\ words}\right) \quad (3)$$

To compute coherence, we developed a custom metric that measures paragraph coherence based on documentation of the TERA [35,61] tool, which measures coherence using the Coh-Metrix [47]. Our approach quantifies the logical flow and semantic consistency between sentences by first extracting individual sentences from each paragraph and generating context-aware embeddings using a pre-trained BERT model [19]. These embeddings capture deep semantic relationships beyond simple word overlap. We then compute pairwise cosine similarity scores between sentence embeddings and average them to produce an overall coherence score. This score ranges from -1 to 1, with higher values indicating stronger coherence.

LLM_Init

```
This is an example of an LTL translation to structured NL:
X ! e & X f means "In the next moment of the system
execution, e should not hold and, in the next moment of the
system execution, f has to hold."
Please show how you would reason to get to a translation
of what this LTL formula means:
[INPUT_FORMULA]
I am more interested in understanding how to read it rather
than reaching the correct answer. Do so in a concise way,
following this structure:
 - Write down original LTL formula                          A
 - Break the subformula into a parsing tree (breakdown).
 - Rebuild meaning from the bottom of the tree up
 - Refine the natural language phrasing for clarity.
```

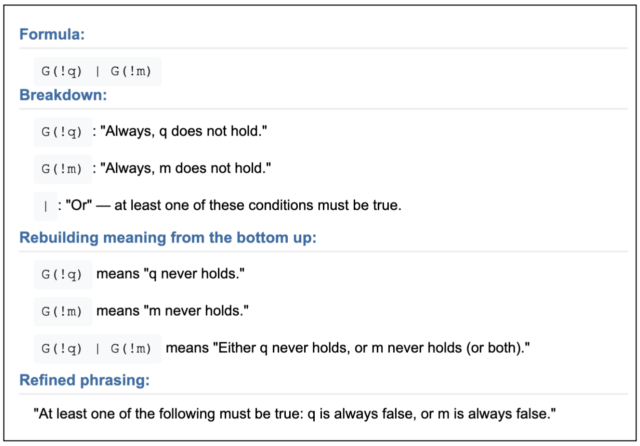
Fig. 1 Initial LLM LTL Prompt (LLM_Init)

In summary, we use the Flesch Reading Ease score (FRE), Dale-Chall (DC) score, and our custom Coherence metric to evaluate clarity.

Length. Finally, we define the *length* metric as the number of words (or distinct tokens) contained in a given explanation once markup language instructions relating to formatting have been removed.

3.5 Formula Explanations vs. Summaries

Within our prompts, we ask the LLM to follow a specific structure (see fragment A in Figure 1). In this paper, we define the formula *explanation* as the entire result produced by the LLM. Conversely, we define the formula *summary* as a natural language refinement of the formula, requested in the last statement in Figure 1. For example, Figure 2 shows a lightly formatted LLM explanation. In this case, the formula explanation is the entire text of Figure 2, whereas the summary is only the text immediately after “Refined phrasing:”. Our gold standard described above in Section 3.2 contains only summaries for each formula. Similarly, in the classroom study (see Section 5), students were asked to write formula summaries via the prompt, “Please explain the meaning of the formula in your own words (1 sentence)”.



Formula:

`G(!q) | G(!m)`

Breakdown:

`G(!q)` : "Always, q does not hold."

`G(!m)` : "Always, m does not hold."

`|` : "Or" — at least one of these conditions must be true.

Rebuilding meaning from the bottom up:

`G(!q)` means "q never holds."

`G(!m)` means "m never holds."

`G(!q) | G(!m)` means "Either q never holds, or m never holds (or both)."

Refined phrasing:

"At least one of the following must be true: q is always false, or m is always false."

Fig. 2 LLM-based Explanation for $G(!q) \mid G(!m)$ using LLM_Init

4 Initial Prompt (LLM_Init) and LLM Selection

Using an iterative design paradigm [69], we proposed an initial prompt with the expectation that empirical insights from the classroom study (see Section 5) would expose areas of improvement. The prompt template, listed in Figure 1, was constructed to elicit explanations that break down LTL formulae into comprehensible parts.

This prompt structure integrates several methodological considerations (see Figure 1). First, it implements the *one-shot* prompting paradigm (see Section 2.2) as it is initiated with a single clear example translation to establish the expected output format, where the example is excluded from any test dataset, reflecting the nature of in-context learning which operates without parameter updates [48]. Simultaneously, a structured step-by-step reasoning approach is used by providing a pre-defined template within the prompt, which guides the model to break down the formula into components before constructing the meaning. This reflects established teaching methodologies for formal notations. Additionally, this prompt emphasizes interest in the reasoning process rather than the mere correctness of the translation, encouraging a more explanatory reasoning over black-box answers. Finally, a clear set of tasks was defined for the LLM, including using a bottom-up approach to construct the meaning of a formula from each of its constituent parts, mirroring how humans break down complex formulae.

In early 2025, we informally tested our initial prompt with three versions of OpenAI's LLMs: GPT-4o, GPT-4o-mini, and GPT-o3-mini [51]. Among the models tested, GPT-4o demonstrated the best results in accurately interpreting and explaining LTL formulae. While all models followed our structured prompting approach, GPT-4o produced more concise and semantically correct explanations, particularly when handling complex nested temporal operators, which is helpful for non-expert users. GPT-4o was also more consistent in creating parse trees. For these reasons, we selected GPT-4o (i.e., *gpt-4o-2024-08-06*) for use in the classroom study in Section 5 and our initial evaluation in Section 7. We used the default values for all parameters except temperature, which we lowered to 0.2 to achieve more deterministic results. In Section 9, we examine how the selection of an LLM impacts the quality of generated LTL explanations.

5 Classroom Evaluation

In this section, we describe our study to explore to what extent the LLM explanations assisted the students in interpreting LTL formulae and whether students found them helpful.

In this investigation, we answered three research questions:

- RQ1 Do LLM explanations affect students' understanding of LTL formulae?
- RQ2 Do students find the explanations helpful?
- RQ3 How do students perceive the LTL intervention?

5.1 Methodology

We conducted our investigation in February 2025 at *Anonymous Institution*, and our protocol was reviewed by our Institutional Review Board. Supplemental materials and study data are available online[1].

5.2 Study Rationale and Experimental Design.

We investigated how LLM-based descriptions can assist non-experts in interpreting formal notations through a classroom study. Our evaluation was designed as an educational intervention with a single sequence. Each student participated in a short lecture (Period 0, see Table 1), and then completed the study intervention. The intervention consisted of three periods. In Period 1, students received instructions on how to complete the survey instrument and reviewed a cheatsheet of logical and temporal operators. Students were given both physical and electronic copies of the cheatsheet for ease of reference in the remaining periods. In Period 2, students considered a series of eight LTL formulae, which are listed in Table 1. In half of the cases (see LLM Explanation column in Table 1), students were given the

Table 1 Study Protocol & List of Formulae

Period	Details		
0	Introduce LTL		
1	Review Cheatsheet and Collect PII		
2	Answer Questions about Formulae		
	ID	Formula	LLM Explanation
	A	$G(!a)$	No
	B	$F(!e)$	Yes
	C	$G(!p \rightarrow Xs)$	Yes
	D	$G(c \rightarrow X\ G\ r)$	No
	E	$F(!p\ \&\ !u)$	No
	F	$G(!q) \mid G(!m)$	Yes
	G	$G(!a \rightarrow F\ z)$	Yes
	H	$G(a \rightarrow F(v\ U\ w))$	No
3	Reflection		

LLM explanation generated using the prompt we discuss in Section 4. For example, Figure 2 shows the explanation for the Formula F (in Table 1) as it appeared in our study instrument. For each formula and explanation, students were asked 4-5 questions. In Period 3, students reflected on learning LTL and the explanations.

To create our questionnaire, we began with the dataset from Cherukuri et al. [13] and reviewed the work of van Lamsweerde [41]. From these sources, we curated a set of thirteen formulae that represent common temporal patterns, covering a range of logical and temporal operators. We discussed and selected eight of the original thirteen formulae, looking for pairs at a similar level of difficulty (see Table 1 for the final list). Given the threat of carryover and a learning effect, we varied the order of which formula received the explanation, as we increased the level of difficulty with each pair (i.e., A/B, C/D, E/F, and G/H). We then created a list of types of questions (e.g., applying a sequence to a formula), and varied the question types throughout each part. This initial educational intervention[1] (i.e., study instrument) is not complete in that it does not ask each type of question for each formula, but we made this design choice to ensure that our intervention with the short lecture would fit into a single 75-minute class block. Additionally, we wanted the students to not unquestioningly take the LLM explanations as correct, so for each explanation, we asked the question, "Do you think the explanation of the formula above is correct (Yes/No)? Why or Why not?", which we call the *LLM-Belief* question in the remainder of this paper.

5.3 Study Context and Participant Information.

The evaluation was conducted during a 75-minute class session as part of a required second-year undergraduate computer science (CS) theory course. The previous and next course topics were pushdown automaton and Turing machines, respectively. The course instructor presented the introductory materials (i.e., the lecture in Period 0) similar to other classes, which required 25 minutes. A student researcher, acting as an in-class teaching assistant, was on hand to help answer and document student questions.

After the lecture, students were given access to a Qualtrics survey for the intervention in Periods 1–3. We originally envisioned the intervention instrument as a paper handout, but the course instructor recommended using Qualtrics because he had used it frequently for in-class activities and students were quite familiar with it. Students were encouraged to use other resources as part of the session and were able to ask questions of the instructor and the assistant. Students could leave at any time, and the first student left after 40 minutes.

Eighteen (18) students completed the questionnaire. Of these, we gained informed consent from 17 students; thus, we exclude the data of one student. Informed consent was collected during a previous class by one of the researchers without the presence of the course instructor. Students were assured that non-participation would not affect their standing in the course. By separating the consent process from the study itself, we achieved a more natural class setting.

No participant had prior experience with LTL, but one student (P) indicated awareness of other notations from work in the industry. All questions were voluntary (i.e., no question was mandatory). With the exception of Student Q, who started fifteen minutes after the rest of the participants, all students answered all questions. Student Q skipped the questions for formula H (see Table 1) and instead completed Period 3.

5.4 Data Processing.

After the in-class session, the collected data was anonymized and divided into questions with a definitive answer, qualitative response, and student feedback (see codebook online[1]). The qualitative questions were independently adjudicated for correctness by two researchers, achieving good inter-rater reliability $\kappa = 0.57$ (Cohen's kappa coefficient [15, 57]). Conflicts between the two adjudications were then evaluated by a third researcher and discussed among all the researchers. The adjudicated qualitative data was then aggregated with the questions with definitive answers to create the students'

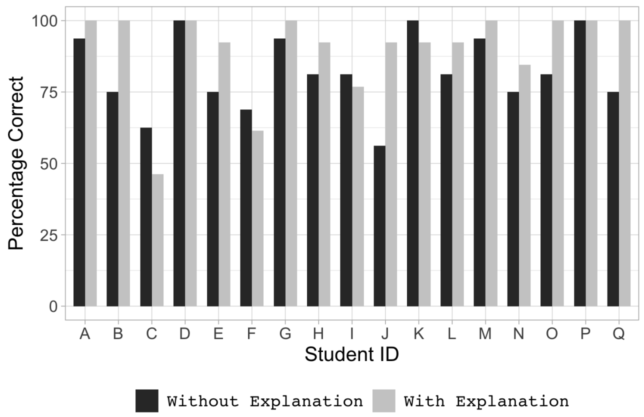


Fig. 3 Graph of the student percentages scores of questions for formula without (black) and with (grey) explanations.

score on the educational intervention (i.e., one point per question). The student feedback collected in Period 3 was analyzed by one researcher and verified by a second researcher.

5.5 Results

5.6 RQ1: LLM Explanations and Scores.

We evaluated students understanding of the formulae in Table 1, and students performed well overall with a median score of 29 (IQR 27 to 32). The highest score was 33/33 (Student P), and the lowest score was 19/33 (Student C). Formula F had the greatest percentage of correct responses (97% correct), while formula H had the least correct responses (72% correct without Student Q).

Null Hypothesis 1 *There is no observable difference between the median of the distributions of students' scores with and without the LLM explanations.*

We compared the scores with and without an associated explanation. Students' scores were higher on formulae with an explanation. This phenomenon exists with and without the LLM-Belief questions. Figure 3 shows the percentage of correct responses for the formulae questions (without the LLM-Belief questions) of each student, with and without explanations. Two students (D, P) scored equally well with perfect scores, and three students scored better without the explanations (C, F, K[2]); the remainder all favored better with the explanations. We find this difference in score to be significant at the $\alpha = .05$ level ($p = .032$, Wilcoxon paired signed rank test [16]), rejecting Hypothesis 1. The median improvement in students' scores is 6% (IQR 0% to 17%) with the explanation, meaning most students improved by at least one question, although we do not make claims about the true magnitude of the effect size since we are comparing percentages because the total number of questions was not the same for each formula (see Section 5.1). **To answer RQ1, we found a difference between students' scores with and without explanations, indicating an effect on student understanding.**

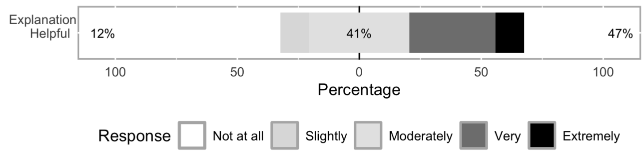


Fig. 4 Graph of student responses to the question, "to what extent were the explanations helpful in understanding the LTL formula?"

5.7 RQ2: Helpfulness of Explanations.

We directly asked students if they found the LLM explanations helpful. Figure 4 visualizes to what extent students found the explanations helpful in understanding the LTL formulae. The vast majority of students (15/17 or 88%) found the explanations at least moderately helpful. The two students (E and J) who found the explanations only slightly helpful both performed better with the explanations, with Student J having the largest percentage difference between the two measures (see Figure 3). Further, we asked students how they would improve the explanations, and five (5) students thought the descriptions were good and didn't make suggestions about how to improve them. The twelve remaining students wanted one or more changes: five (5) wanted more examples, five (5) wanted more visualizations, three (3) wanted them to be shorter, and two (2) wanted more detailed breakdowns. **We find that students preferred formulae with explanations, with roughly 40% of them finding the explanations helpful without any recommended modifications, answering RQ2.**

5.8 RQ3: Students' Perceptions.

In Period 3 of the intervention, we gauged the students' subjective experiences. Three-quarters of students reported learning LTL as similarly challenging to other course topics, with the remainder reporting it as less challenging (no one reported it as more challenging). When asked which temporal operator was the most

[2] Students C and F had the lowest overall scores. Student K missed only one question on the assessment.

challenging to understand, eight (8) students found Until U to be the most difficult temporal operator. Of the remaining students, four (4) selected Eventually F, four (4) selected Globally G, and one (1) selected Next X. We also inquired about what aspects made the questions challenging to answer, gave a few examples, and allowed students to write independently. From this data, we find that eleven (11) selected "interpreting the implications of formulae" as the hardest part, while eight (8) selected "applying the formula to sequences", with five (5) students selecting both of these. Only one student (B) selected "none, I did not find the questions challenging".

To understand engagement and whether students saw the applicability of LTL, we asked them how they could "use LTL (or this kind of reasoning) in a computer science project?". Ten (10) students produced answers that showed clear applicability. For example, students reported examples such as video games, autonomous cars, simulations, and software with timed events. Student G explained how each temporal operator would be used. The remaining seven (7) students were either unsure or gave vague answers (e.g., "by simplifying the logic behind some algorithms" [Student L]). **To summarize our data for RQ3, students perceived the LTL intervention as an appropriate level of difficulty for their class and varied in which parts they found most challenging.**

5.9 Discussion and Limitations

Qualitative Observations. While we invited students to use outside resources, none of the students appeared to use any other resources than the lecture slides and cheatsheet. In Period 3, multiple students noted that they had asked either the student researcher or instructor a question. During Period 2 of the study, students were confused with how to think through true/false questions about the satisfiability of an example sequence, how to write infinite sequences, and what the meaning of the first value in the sequence is. For example, Student P recommended changing the wording of the question from "[i]n a sequence where..., the formula is violated" to "the formula is *always* violated". Future iterations should add more concrete explanations of the notation around sequences of states.

As mentioned in Section 4, our prompt was necessarily insufficient. For example, in the explanation for $F(!e)$ (i.e., Formula B) and $G(!a \rightarrow F\ z)$ (Formula G), the LLM explanation identified the breakdown as "F(...): "Finally" (F) — at some point in the future"; yet, F stands for *eventually.* The resulting refined phrasing was correct in both cases, but this may have caused confusion about using F. Two students (E and N) correctly identified this error in terminology.

Identifying Points of Confusion. Students particularly struggled with the Until (U) operator, which eight students identified as the most difficult temporal concept to grasp. Eleven students cited interpreting implications as their primary challenge, while eight found difficulty applying the formulae to sequences. Students also expressed confusion about when temporal constraints begin and end, particularly in nested expressions. One participant questioned "whether F would be separate for each instance" [Student A] and how a single satisfaction of F impacts multiple trigger conditions. Additionally, students requested improvements to the explanations, with five wanting more examples, five seeking better visualizations, and three preferring more concise descriptions. These observations are consistent with the findings in Greenman et al. [31].

Threats to Validity. Our study is only a preliminary investigation and was limited by small sample size and educational constraints, which prohibited a control group and limited interactions with students to one class session. Learning and carry-over effects likely occurred as students progressed through the formulae. Our only attempt to mitigate this was to vary which formulae included explanations across different difficulty levels. Within each pair (e.g., C/D and G/H) and across the sequence, students may have learned about the operators from the explanations. Both of these effects may have confounded our results for RQ1, and further investigation is required to verify our claims, as well as more precisely measure their effect size. Additionally, the constructs of helpfulness and understanding are threatened by our use of a unipolar scale and our tool-agnostic design. Students may perform differently when interacting directly with an LLM, and may have perceived the helpfulness of the explanations differently if presented with multiple levels of unhelpfulness [11]. The single-session format also meant students may have experienced fatigue toward the end of the session, potentially affecting their performance on more complex formulae. Moreover, the voluntary nature of participation could have introduced some self-selection bias, as students more interested in formal notations may have shown more engagement. While students are one of our target populations for this work, CS undergraduates may not generalize to the broader population of stakeholders. The classroom setting and relatively simple LTL formulae may not fully capture the complexity of interpreting formal notations in all industrial contexts, and the helpfulness of our LTL explanations may not scale to complex nested formulae. Consequently, while the prompt was designed to elicit explanations that break down LTL formulae into comprehensible parts and emphasize the reasoning process, for these simpler formulae, the explanations might have appeared to students as providing direct solutions rather than extensively guiding them toward independently understanding and translating more complex formulae. In future interventions, the provided LTL materials could be more comprehensive, for example, clarifying operator precedence and associativity or introducing common equivalences. Our focus on specific LTL temporal operators may not generalize to other formal notations or more complex formula structures. Our small sample size (17 students) limits the statistical power of our analysis. The subjective evaluation of student responses also introduces potential measurement variability, which we attempted to mitigate through independent evaluation by multiple researchers.

6 Expanded Prompt (LLM_Exp)

Next, we describe our methodology for creating our expanded prompt for LLM-based explanations of LTL formulae. The updated prompt (i.e., LLM_Exp) is shown in Figure 5.

Incorporating Points of Confusion. Our LLM_Init prompt (see Section 4) can be improved by incorporating points of confusion. First, we integrate the points we discussed in our classroom study (see Section 5.9 for details). Second, as part of creating the gold standard for the 50F dataset (see Section 3.2), we identified common misconceptions. For example, explanations may not describe when temporal constraints begin and end, the satisfaction conditions for complex operators, the implications for state transitions, and satisfying and violating sequences.

For state transitions involving Until (e.g., $p\ U\ q$), explanations frequently fail to emphasize that once q becomes true, the entire formula is satisfied regardless of the value of p. When explaining implication (e.g., $a \rightarrow b$), explanations may state only that "if a holds, then b must hold" without making explicit what happens when a does not hold. Further, when breaking down more complex formulae (e.g., $(d \mid !b)\ U\ !a$), the explanation may list conditions randomly instead of prioritizing the part that determines when the formula terminates.

Prompt Refinement. The LLM_Exp prompt template is initialized with system contextualization. This involves defining the LLM's role as an expert in formal RE, particularly in Linear Temporal Logic, aiming to draw attention to relevant knowledge that the LLM

LLM_Exp

A

Task: You are an expert in formal requirements engineering with a focus on Linear Temporal Logic (LTL). Your goal is to translate formal LTL formulas into natural language, providing clear, precise explanations that account for when the formula is satisfied and when it imposes constraints on the system over time.

B

Instructions: For this formula [INPUT_FORMULA], your explanation must:
Specify clearly when the formula imposes a constraint on variables and when it does not.
Describe how the formula is evaluated at the moment it is checked and how its truth value may evolve.
Distinguish between the immediate satisfaction of the formula and the ongoing constraints that apply over time.
For temporal operators, explain when the formula imposes constraints, and when those constraints no longer apply.
Address both satisfying and violating sequences to demonstrate the conditions under which the formula holds and when it does not.
Highlight potential misunderstandings, especially when constraints apply and when they cease.

C

Key Guidelines for Specific Temporal Operators:
X(p): "p must be true in the very next state."
G(p): "p must be true in every state of the execution."
F(p): "There must exist at least one future state where p is true."
Until (U): Break down the behavior into two phases:
a U b: "b must eventually become true, and until that specific moment, a must remain true in every state."
Once b becomes true, a no longer needs to hold.
Negated Operators: Address negated operators explicitly:
!(G(p)): "There must be at least one state where p is false."
!(F(p)): "p must be false in all states of the execution."

D

Example Formula: (c & b) U a
Immediate Evaluation: Explain when (c & b) must hold (in the current state or in future states).
Constraints Over Time: Describe that a must become true eventually, and until it does, both c and b must hold. Clarify that once a becomes true, c and b are no longer constrained by the formula.
Explain Transition: Make clear that the formula does not impose constraints on c and b after a becomes true.
Satisfying and Violating Sequences: Illustrate how the formula is satisfied and when it is violated.

The translation would be: a has to be true either in the moment in which the formula is evaluated -and in that case the formula does not constrain the value of c and b- or at some point in the future -and in that case, c and b have both to be true from the moment the formula is evaluated until the instant before a is true (the formula does not constrain the value of c and b in the moment in which a is true or after it).

Conclude your response with a summary of the translation of this formula based on your explanation and analysis.

Fig. 5 Extended LLM LTL Prompt Template (LLM_Exp)

Table 2 Data Table for RQ4: Prompt Correctness, Clarity (Flesch Reading Ease (FRE), Dale-Chall (DC), and Coherence score), and Length for Explanations with the LLM_Init and LLM_Exp Prompts

Correctness	LLM_Init	94% (47/50)	LLM_Exp	82% (41/50)	
	LLM_Init		LLM_Exp		Comparison
	Median	IQR	Median	IQR	P-Value
FRE	50.47	(46.14 to 55.30)	54.90	(50.49 to 59.66)	.012
DC	7.98	(7.73 to 8.30)	6.85	(6.66 to 7.07)	<.001
Coherence	0.52	(0.50 to 0.54)	0.56	(0.54 to 0.57)	<.001
Length	232	(208 to 247)	475	(443 to 505)	<.001

may already have in this domain (see fragment (A) in Figure 5).

While our first approach focused on structural decomposition through parse trees, our analysis showed that effective LTL explanations need to address temporal semantics and constraint boundaries more explicitly and provide clearer distinctions between when constraints apply. For example, we instruct the LLM in fragment (B) to “distinguish between the immediate satisfaction of the formula and the ongoing constraints that apply over time”. Next in fragment (C), we provide specific guidelines for explaining each temporal operator, with particular emphasis on their semantic meanings and constraint boundaries, which address the points of confusion discussed above.

We retained the one-shot prompting paradigm, guiding the LLMs reasoning with an example. The updated example gives a complete reasoning process for the Until (U) operator—which students found most challenging—demonstrating precisely how constraint boundaries should be explained. We use the formula $(c\ \&\ b)\ U\ a$ in our prompt, see fragment (D) in Figure 5 for the recommended translation.

This approach leverages the LLM’s ability to recognize patterns and apply similar reasoning structures across different formulae while remaining focused on the semantic nuances of temporal logic.

7 Evaluation of Explanation and Summary Quality

In this section, we analyze the quality of the explanations and summaries generated by the LLMs and compare them with our gold standard and student responses. We ask two research questions:

RQ4 To what extent has the resulting explanations’ quality changed with the LLM_Exp prompt?

RQ5 How does the quality of the LLM-generated formula summaries compare with summaries written by students and experts?

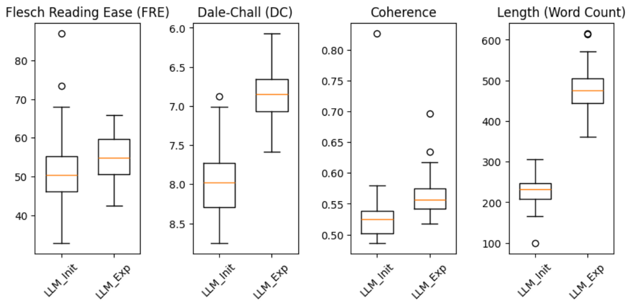


Fig. 6 Box plots of the Flesch Reading Ease (FRE) score, Dale-Chall (DC) score, and Coherence score, and Length for the full explanations of each formula in the 50F dataset, using the initial (LLM_Init) and expanded (LLM_Exp) prompts.

See Section 3.5 for differentiation between explanations and summaries of LTL formulae. For consistency, we used GPT-4o to generate both the summaries and explanations in this section, see Section 4 for details.

7.1 RQ4: Changes in Explanation Quality

We first address RQ4, considering how the quality of the resulting formula explanations changed with our expanded prompt. We compare our expanded prompt (identified as LLM_Exp, see Section 6) with our initial prompt (identified as LLM_Init) described in Section 4 using the quality criteria discussed in Section 3.4.

Null Hypothesis 2 *There is no observable difference between the quality scores based on whether they were generated with the LLM_Init or LLM_Exp prompt.*

The data for RQ4 are listed in Table 2 and shown in Figure 6.

Correctness. We adjudicated 94% (47/50) of the explanations generated by LLM_Init to be correct. However, only 82% (41/50) of the explanations generated by LLM_Exp were adjudicated to be correct. We found a decrease in quality of the explanations.

Clarity. We computed the Flesch Reading Ease (FRE), Dale-Chall (DC), and the Coherence scores for the 50F

Table 3 Data Table for RQ5: Prompt Correctness, Clarity (Flesch Reading Ease (FRE) and Dale-Chall (DC) score), and Length for Summaries

	Correctness	FRE		DC		Length	
	Percent (Score)	Median	IQR	Median	IQR	Median	IQR
LLM_Init	94% (47/50)	57.6	(43.1 to 71.4)	8.84	(8.33 to 9.71)	22	(17 to 25)
LLM_Exp	86% (43/50)	68.0	(62.1 to 74.7)	7.76	(7.41 to 8.26)	53	(47 to 65)
Gold	94% (47/50)	63.0	(33.8 to 68.3)	7.72	(7.17 to 8.42)	31	(24 to 77)
Students	81% (68/84)	84.0	(74.4 to 92.6)	7.57	(6.98 to 8.38)	16	(13 to 19)

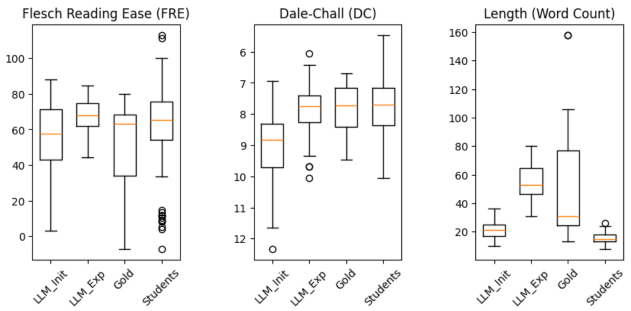


Fig. 7 Box plots of the Flesch Reading Ease (FRE) score, Dale-Chall (DC) score, and Length for the summaries of each formula in the 50F dataset, generated by the initial (LLM_Init) prompt, the expanded (LLM_Exp) prompt, and included in our gold standard. FRE and DC scores of all student responses from the classroom study discussed in Section 5 included for benchmarking purposes.

dataset using the prompts LLM_Init and LLM_Exp. Figure 6 shows box plots for each of the scores, grouped by LLM_Init and LLM_Exp. Recall the scales of FRE and DC are reversed, meaning a higher score on FRE and a lower score on DC indicate easier fluency and readability. From the box plots in Figure 6, it is evident that the median score for the expanded prompt (LLM_Exp) is better than the initial prompt (LLM_Init) across all three measures, accounting for both readability and coherence. We find that the difference in scores is significant at the $\alpha = .05$ level across all three scores (FRE: $p = .012$, DC: $p < .001$, and Coherence: $p < .001$, via the Wilcoxon paired signed rank test).

Length. The median length of the explanations produced by LLM_Init was 232 words, while LLM_Exp had a median length of 475 words. We find that the resulting length of LLM_Exp explanations are significantly longer than those of LLM_Init ($p < .001$, via the Wilcoxon paired signed rank test).

Synthesis. We reject Hypothesis 2 as there are statistically significant differences across all metrics. The clarity of the explanations improved overall; yet, undesirably, the correctness decreased and the length increased. Therefore, we found that the expanded prompt LLM_Exp is easier to read but not ideal.

7.2 RQ5: Comparing Machine and Human Summaries

Next we investigate RQ5: How does the quality of the LLM-generated formula summaries compare with summaries written by students and experts? We compare the summaries given by the initial and expanded prompts (LLM_Init and LLM_Exp, respectively) directly with the summaries in our gold standard (Gold) for the 50F dataset. In addition to the 50F dataset, we evaluate all student responses (Students) in our classroom study (see Section 5) to all questions where students were asked to write summaries for a given formula (see question wording in Section 3.5). The data for RQ5 are listed in Table 3 and shown in Figure 7.

Correctness. After independent assessment, we adjudicated 94% (47/50) of the LLM_Init prompt summaries to be correct, and 86% (43/50) of the LLM_Exp prompt summaries to be correct. After analyzing RQ4, this data is not surprising because, by definition, summary correctness cannot be worse than the full explanation correctness, as the summaries are just the last sentence of the explanation. The LLM_Init prompt generated the same number of correct summaries as our initial gold standard (Gold), which we adjudicated to be 94% (47/50) correct upon independent evaluation. The LLM_Exp prompt generated fewer correct summaries. Finally, we adjudicated 81% (68/84) of student summaries to be correct, which is inline with the LLM summaries.

Clarity. For this research question, we measure clarity in terms of readability only. Coherence can only be measured across multiple sentences and since most of the summaries contain only one sentence, our coherence metric does not apply. To measure readability, we again use the Flesch Reading Ease (FRE) and the Dale-Chall (DC) scores. Figure 7 shows box plots for each of the FRE and DC scores, grouped by data source. In comparing LLM_Init and LLM_Exp, we see a similar trend as in RQ4 where the expanded prompt (LLM_Exp) produces more readable results than the initial prompt (LLM_Init); yet, here we investigate how the LLM outputs compare with those generated by humans. The results of the Flesch Reading Ease (FRE) scores show variations between groups of humans. For example, the gold standard (Gold) created by an LTL expert had a median FRE score of 63 whereas the students' responses (Students) had a median FRE score of 84, meaning the expert used a higher fluency (or grade-level) than the students. The expanded prompt (LLM_Exp) produced a median of 68, which was between the results of the Gold standard and the Students. However, the results of the Dale-Chall (DC) scores showed similarity between groups. The medians of LLM_Exp, Gold, and Students were 7.8, 7.7, and 7.6, respectively. See Table 3 for complete listing of scores and interquartile range (IQR). Depending on the measure, the clarity of the expanded prompt was similar to humans (i.e., between novice students and an expert).

Length. We found variability in the length of the summaries (see Length column in Table 3, and box plot in Figure 7). First, the median and interquartile ranges (IQR) for LLM_Init and Students were similar. While the students' responses were shorter, we believe that since they saw multiple LLM_Init summaries before being asked to write one, they mirrored the examples in terms of length. Second, similarly with RQ4, we find that the summaries produced by LLM_Exp were double the length as those produced by LLM_Init and triple the length of those produced by the students (Students). The median gold standard (Gold) length was double the length of the median of the student responses. Finally, the IQR for the gold standard extends the range of both LLM prompts, which do not overlap with each other. The IQR of the gold standard is in a different range than the IQR of the students.

Synthesis. We found mixed results for RQ5. Similar to RQ4, the summaries produced by LLM_Exp have better clarity than those produced by LLM_Init, but are less correct and longer. The clarity of the LLM summaries is within the range of the students responses and the gold standard. Student responses were shorter, but less likely to be correct, while the gold standard was more likely to be correct with a wide range in response length.

7.3 Discussion and Limitations

In RQ4, we found that updating our prompt improved the clarity of both explanations. Based on the metrics used in RQ5, the expanded prompt was either at the level of fluency of novices or in between experts and novices. We found the empirical evidence of RQ4 and RQ5 to be compelling and think the expanded prompt produces better explanations and summaries. However, the expanded prompt produced longer responses that were less likely to be correct.

While there are various limitations and threats to our approach, the most significant design flaw in creating the expanded prompt (LLM_Exp) is that we changed too many factors of the prompt at the same time. Post-analysis, we cannot assert which of the changes had a positive or negative impact on correctness, clarity or length. We mitigate this limitation in the next section by systematically analyzing features of the prompt.

To evaluate correctness in this section, we used Definition 1 (see Section 3.4). Two authors independently adjudicated the LLM_Init and LLM_Exp explanations and summaries for correctness, with a third author weighing in to resolve disagreements. We achieved very good inter-rater reliability with $\kappa = 0.63$ for the explanations and $\kappa = 0.72$ for the summaries (Cohen's kappa coefficient). Similarly, we re-adjudicated the correctness of the student responses using Definition 1 and achieved excellent inter-rater reliability $\kappa = 0.92$. In Section 3.2, we introduced the 50F dataset and our gold standard. Upon investigation, we found that 3/50 summaries in our gold standard contained errors, which were likely the result of fatigue and the repetitive nature of the original creation process. These errors highlight the importance of verifying our gold standard, and suggest that even experts make mistakes and may benefit from external support.

8 Systematic Evaluation of Prompt Components

Recall that thus far, we defined two prompts (LLM_Init and LLM_Exp) and compared their results with each other, as well as the human created summaries (see Section 7). Since we changed multiple aspects of the prompt between LLM_Init and LLM_Exp, we cannot assert which aspect improved the prompt, as we did not isolate the study of each part. In this section, we perform an experiment of prompt components (called factors) in an attempt to identify the optimal prompt for creating LTL formula explanations.

We aim to answer the research question:

RQ6 How does the inclusion of prompt factors affect the quality of generated LLM explanations?

8.1 Prompt Factors

To answer RQ6, we independently consider and evaluate five prompt factors: role, guidelines, specific operators, example, and style. Figure 8 maps the prompt and prompt factors discussed in this section. The prompts, LLM_Base and LLM_Agg, are shown on the left and right

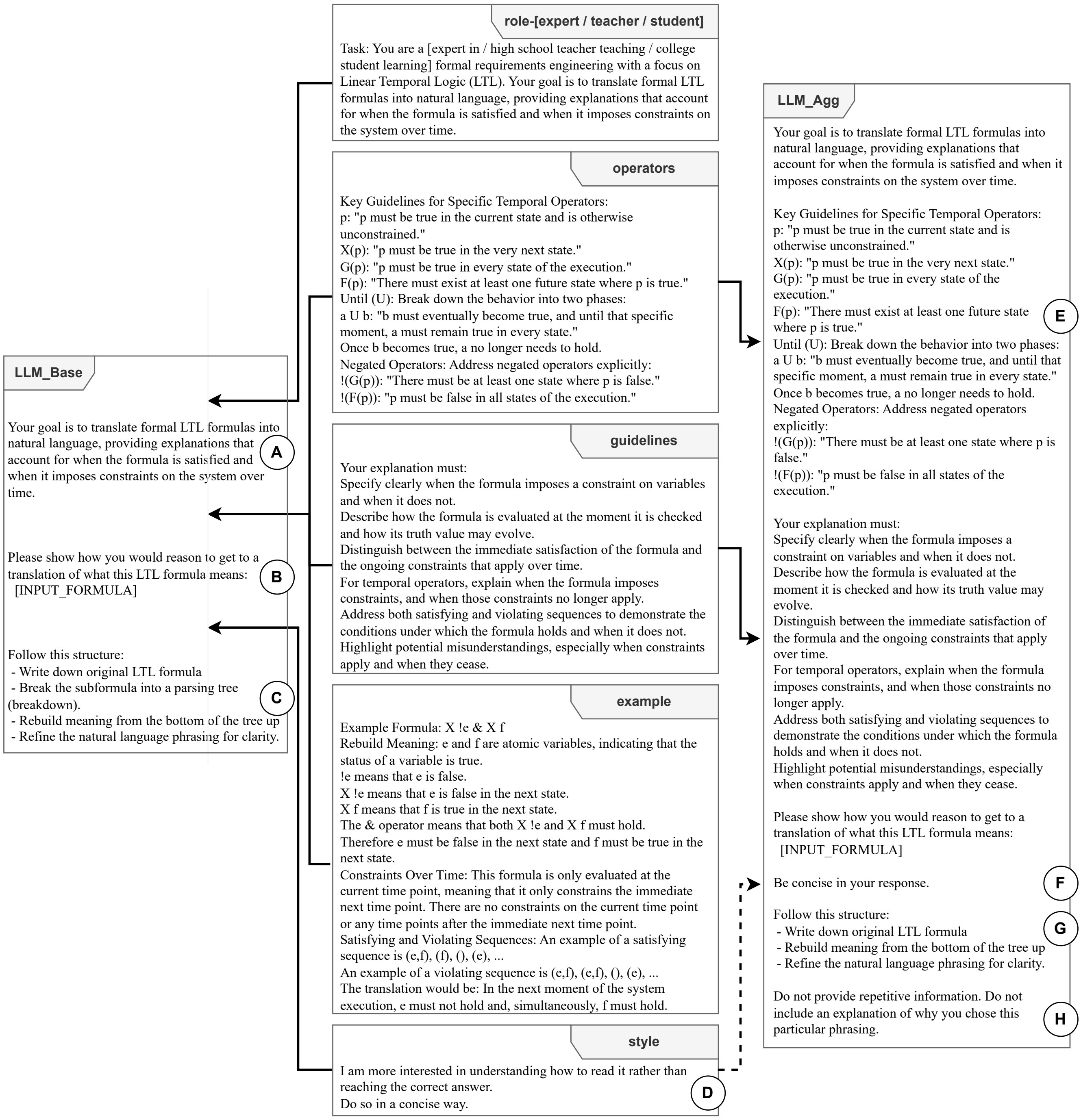


Fig. 8 Prompt Factors and their Relationship to the LLM_Base and LLM_Agg Prompts

side of Figure 8, respectively, while the middle column lists the individual factors.

Base Prompt (LLM_Base). We begin by creating a LLM_Base or minimal prompt for analysis. We reviewed each fragment of the LLM_Init prompt (introduced in Section 4) and the LLM_Exp prompt (introduced in Section 6) aiming to isolate required vs. optional fragments for analysis. The LLM_Base prompt shown on the left-hand side of Figure 8 contains three parts. The top statement (see Ⓐ in Figure 8) is taken from LLM_Exp instructing the LLM to consider when the statement is satisfied, the middle statement Ⓑ is the original question from LLM_Init, and the bottom statement Ⓒ is a set of tasks for the LLM to complete to ensure standardize outputs.

Role (role). We examine whether prescribing a role to the LLM in the prompt affects the clarity of the prompt. We add a sentence prescribing a role that the LLM must assume and experiment with including three roles: "expert" (role-expert), "high school teacher" (role-teacher), and "college student" (role-student). This sentence occurs at the beginning of the prompt, before fragment (A) the LLM_Base prompt (see Figure 8). The LLM_Exp prompt (see Section 6) asked the LLM to assume the role of an expert.

Specific Operators (operators). We experiment with providing example explanations of the LTL operators in a prompt called operators, which consists of the LLM_Base prompt plus explanations for each of the temporal operators. The inclusion of explanations of specific operators gives the LLM an example of a concise way to explain each relationship. Additionally, the explicit statement that the absence of a temporal operator only applies to the current state aims to address the most common error with the LLM_Base prompt. The operators factor is added to the LLM_Base prompt between fragments (A) and (B), see Figure 8.

Guidelines (guidelines). The guidelines prompt consists of the content in the LLM_Base prompt, plus the addition of guidelines about the output that were present in LLM_Exp. The guidelines are added between fragments (A) and (B) in the LLM_Base prompt. These guidelines were based on the points of confusion and ambiguities identified in the initial class room study from Section 5.

Example (example). Inspired by one-shot learning, we provide an example of a clear, concise response for the formula X !e & X f. We chose a new example from the one used in LLM_Exp (see fragment (D) in Figure 5), because the original example also belonged to the 50F dataset and we wanted to ensure uniqueness between examples and our datasets. The example is positioned between fragment (A) and (B) in the LLM_Base prompt.

Style (style). The last feature we added to the LLM_Base prompt was a sentence explicitly prompting the LLM to provide a response with the attributes of length and clarity that we measure. Given that the LLM responses are intended to help their audience learn LTL in general rather than answer any one specific question, we emphasize the importance of the explanation over the final answer. Additionally, to reduce fatigue from reading long explanations, we emphasize concision in the style prompt. The style factor is added to the LLM_Base prompt between fragments (B) and (C), see Figure 8.

8.2 Experiment Methodology

As already introduced in the previous subsection, we tested the LLM_Base prompt first and then add each factor one-by-one to the base prompt to test their impact (e.g., LLM_Base+role-exp or LLM_Base+style), for a total of seven experimental conditions. In this experiment, we use the 77F dataset introduced in Section 3.3 and evaluated the resulting explanation for each formula in the dataset using the quality factors introduced in Section 3.4. We ran each combination of formula and test prompt on DeepSeek, as DeepSeek performed the best in our head-to-head comparison of LLMs (see Section 9 for details). Our data, analysis, and generation scripts are available online[1].

Null Hypothesis 3 *There is no observable difference in quality between LLM_Base and any factor prompt (each including one of role-[expert/teacher/student], operators, guidelines, example, and style), where quality is evaluated as (a) correctness, (b) clarity, and (c) length.*

8.3 RQ6: Analyzing Prompt Factors

The results of our analysis are shown in Table 4 and Figure 9. Table 4 lists the results of each experimental condition for the five quality criteria: correctness, clarity (via the FRE, DC, and Coherence Scores), and length. Additionally, we list the results from LLM_Init and LLM_Exp for reference and comparison. Note that the values for LLM_Init and LLM_Exp in Table 4 differ from those reported in Table 2 because we recalculated each criteria using DeepSeek on the 77F dataset.

Established Baseline. The baseline prompt (LLM_Base) was found to be correct for 91% of formulae in the 77F dataset with a median length of 440 words. Four out of the 7 errors were attributed to the explanation applying a global operator where none was present. The median clarity scores were 59.61 for FRE, 11.54 for DC, and 0.516 for Coherence. These results should not be interpreted directly but act as a baseline for comparison with the prompt factors, though we note that clarity and length scores for LLM_Base were better than LLM_Init and worse than LLM_Exp, while correctness was better than both LLM_Base and LLM_Init (see Table 4).

Correctness. In comparing each independent addition to the LLM_Base prompt, we find that every factor reduced the number of formulae where the LLM produced an erroneous explanation. The top three operators that improved correctness were guidelines, operators, and style. In particular, the guidelines prompt produced correct explanations for all 77 formula, producing the only statistically significant correctness result (odds ratio, ∞; 95% CI, 1.51 to ∞; $p = 0.014$, Fisher's Exact Test). See Section 10.1 for our discussion on inter-rater reliability. Thus, we reject Hypothesis 3(a), as the inclusion of guidelines improves correctness.

Table 4 Data Table for RQ6: Quality Criteria for each Prompt using DeepSeek on the 77F Dataset

		FRE			DC		
	Correctness	Median	IQR	P-Value	Median	IQR	P-Value
LLM_Base	91% (70/77)	59.61	(54.30 to 66.75)	-	11.54	(11.13 to 12.03)	-
role-expert	92% (71/77)	58.28	(53.56 to 64.40)	†.006	11.64	(11.12 to 12.20)	.113
role-teacher	92% (71/77)	59.21	(52.94 to 63.01)	.041	11.60	(11.30 to 12.26)	.035
role-student	94% (72/77)	58.69	(53.55 to 63.22)	.022	11.62	(11.10 to 12.22)	.275
guidelines	100% (77/77)	59.28	(54.39 to 64.32)	.188	11.68	(11.21 to 12.15)	.036
operators	97% (75/77)	62.78	(58.53 to 66.56)	.022	11.17	(10.79 to 11.71)	<.001
example	92% (71/77)	62.71	(57.12 to 67.53)	.031	11.59	(10.87 to 11.99)	.187
style	96% (74/77)	59.60	(55.83 to 67.73)	†.362	11.63	(10.99 to 11.98)	.235
LLM_Agg	97% (75/77)	62.08	(58.00 to 67.21)	.007	10.52	(10.12 to 11.15)	<.001
LLM_Init	90% (69/77)	54.35	(49.12 to 59.83)		12.27	(11.88 to 12.67)	
LLM_Exp	88% (68/77)	60.36	(56.03 to 64.50)		11.30	(10.84 to 11.67)	

	Coherence			Length		
	Median	IQR	P-Value	Median	IQR	P-Value
LLM_Base	0.516	(0.508 to 0.525)	-	440	(373 to 500)	-
role-expert	0.517	(0.508 to 0.523)	†.004	426	(383 to 490)	.999
role-teacher	0.513	(0.507 to 0.519)	†.172	437	(393 to 490)	†.999
role-student	0.515	(0.510 to 0.524)	†.254	441	(383 to 492)	.136
guidelines	0.524	(0.514 to 0.532)	.026	620	(533 to 681)	<.001
operators	0.519	(0.510 to 0.530)	.303	404	(337 to 486)	.059
example	0.521	(0.513 to 0.527)	.483	433	(376 to 510)	.143
style	0.515	(0.509 to 0.522)	.271	353	(315 to 402)	<.001
LLM_Agg	0.525	(0.517 to 0.53)	.004	315	(256 to 368)	<.001
LLM_Init	0.516	(0.508 to 0.523)		291	(258 to 342)	
LLM_Exp	0.528	(0.520 to 0.536)		496	(451 to 580)	

P-Values report whether the median for each factor is significantly different from the LLM_Base prompt. P-Values calculated using Wilcoxon paired signed-rank test, unless denoted with a †. A † denotes data with asymmetric distributions about the median, violating the assumptions of the Wilcoxon test, and the use of the *Sign Test* [16] to calculate P-Values. P-Values for correctness excluded for space considerations; readers may calculate the odds ratio from table data using Fisher's Exact Test.

Clarity. Looking first at readability, five tests showed a significant difference in the FRE scores at the $\alpha = .05$ level (see P-Value column for FRE in Table 4). However, the direction of these changes varied, with only operators and example improving the FRE score from 59 to 62, putting them in the eighth or ninth-grade reading level. The operators factor also *slightly* improved the DC score and was found to be significant at the $\alpha = .05$ level ($p < .001$). The example factor worsened the DC score but was not significant. These results suggest that readability is improved by adding the operators factor alone.

Next, examining coherence, role-expert and guidelines showed significant improvements ($p = .004$ and $p = .026$, respectively). Yet, the magnitude of the improvement of role-expert is almost indistinguishable from rounding error and did not improve the values of the interquartile range (IQR). Similarly in the box plots in Figure 9, it is evident that the guidelines factor improves the results of coherence over the LLM_Base. These results show that adding the guidelines improve coherence. Therefore, adding the operators and guidelines improves clarity, rejecting Hypothesis 3(b).

Length. Two factors have a significant impact on length: guidelines and style. However, guidelines significantly increased the word count ($p < .001$), with the median length increasing by 180 words. The style factor decreased the median word count by 87 words ($p < .001$); thus, style was found to improve the length of the explanations, rejecting Hypothesis 3(c).

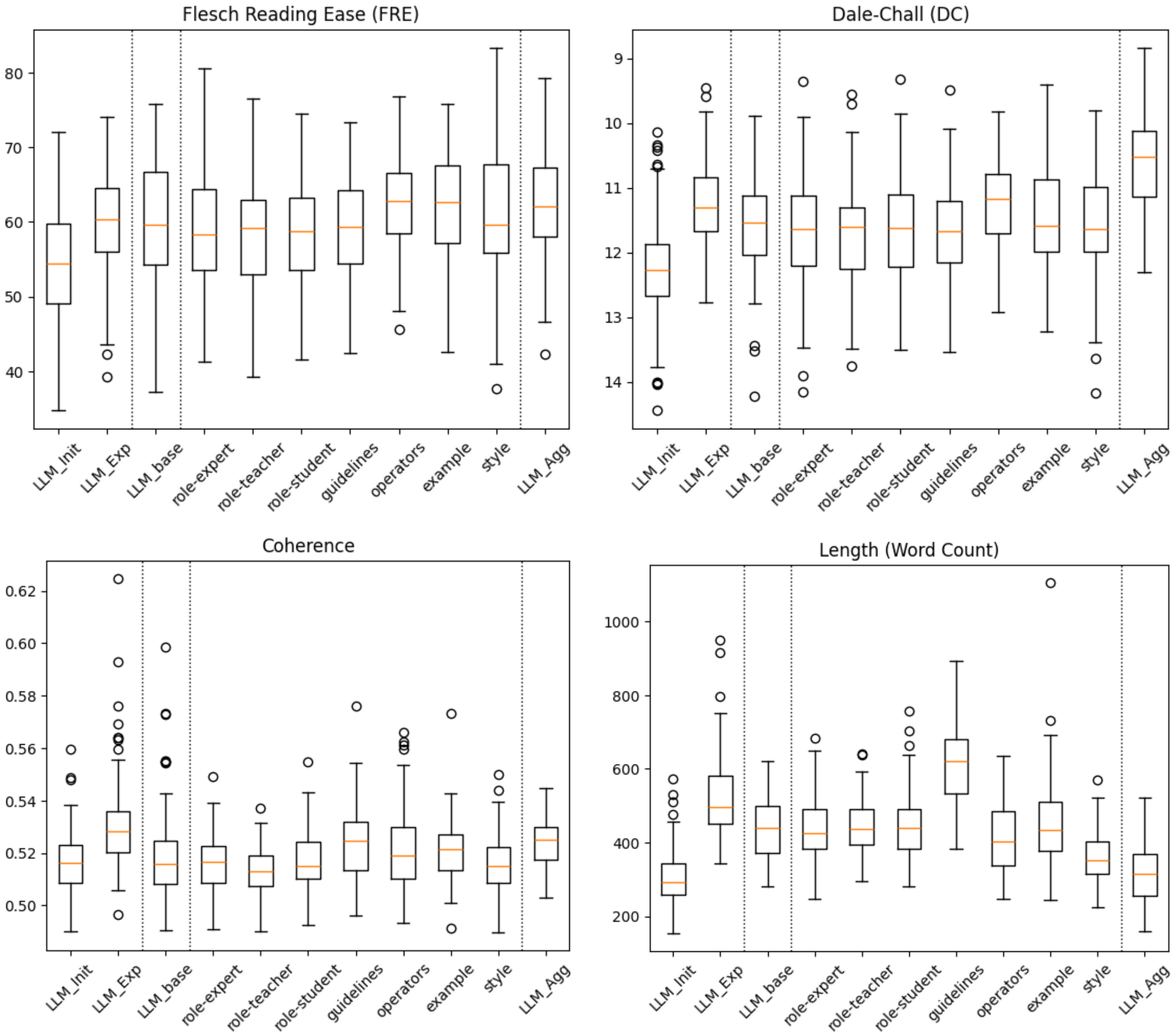


Fig. 9 Box Plots for RQ6: Quality Criteria for each Prompt using DeepSeek on the 77F Dataset (Data from Table 4)

8.4 Finalized Prompt (LLM_Agg)

Regarding RQ6, we find that including operators and guidelines impact the clarity, including style impacts the length, and each of these three positively influence the correctness, with guidelines achieving total correctness. From our analysis of quality criteria, we define our aggregate prompt (LLM_Agg) as the LLM_Base prompt with the additional language of the guidelines, operators, and style factors. Figure 8 right-hand side shows the LLM_Agg and how each factor contributes to the prompt.

While checking the outputs for correctness, we made three qualitative observations. First, the LLM created parse trees without a standardized format. While some formats were intuitive, other parse trees took considerable effort and time to interpret. We also found that the instructions in the LLM_Base prompt (see fragment C in Figure 8) produced redundancies between the 'parsing tree' and the 'rebuilding' steps. Thus, in an effort to decrease the length of the response, we removed the 'parsing tree' step (i.e., removed the second bullet point of fragment C in the base prompt of Figure 8, to produce fragment G). Second, we observed that some LLM responses conclude with a justification for why particular phrases were chosen for the explanation, which is not relevant to understanding the formula. We explicitly added an instruction at the end of the LLM_Agg (see fragment H in Figure 8) to ensure that the LLM avoids this behavior. Finally, as already discussed above, style was only a significant factor in reducing the length and did not impact clarity. Upon review, we conflated two aspects in our style factor: concision and understanding. Thus, we shortened the style factor to only include concision in the LLM_Agg (see fragment F in Figure 8).

Using the LLM_Agg prompt, we achieved 97% (75/77) correctness. The LLM_Agg prompt significantly improved all three clarity metrics and significantly reduced the prompt length. The resulting FRE score was improved by 2.47, a small improvement ($p = .007$); the DC score decreased by 1.02, a grade level ($p < .001$); and the Coherence improved 0.009, a small change ($p = .004$). These changes taken together show an overall improvement in clarity; yet, what is most surprising is that they were accomplished in combination with improving the correctness and significantly decreasing the median length by 125 words ($p < .001$), which is inline with our stated objective for the ideal explanation (see Section 3.4).

9 LLM Comparison

Finally, in this section, we investigate the variability of large language models (LLMs) for the task of generating LTL explanations. We aim to answer two research questions:

RQ7 Does the correctness of LTL explanations generated with the LLM_Base prompt vary by LLM choice?

RQ8 To what extent do prompt improvements generalize across LLMs?

We investigated six candidate LLMs. We selected popular models from several different LLM families [49], aiming to sample models trained on different materials. We restricted model versions to those that were free to access.

- **DeepSeek** DeepSeek: R1 0528, released May 28, 2025
  openrouter.ai/deepseek/deepseek-r1-0528:free
  Accessed: July 31 - November 21, 2025
- **Gemini** Gemini 2.5 Flash, released June 17, 2025
  openrouter.ai/google/gemini-2.5-flash:free
  Accessed: September 10 - September 16, 2025
- **Llama** Meta Llama 4 Scout, released April 5, 2025
  openrouter.ai/meta-llama/llama-4-scout:free
  Accessed: September 10 - September 16, 2025
- **Mistral** Mistral-Medium-2508, released August 12, 2025
  docs.mistral.ai/models/mistral-medium-3-1-25-08
  Accessed: September 11 - September 16, 2025
- **GPT** OpenAI: gpt-oss-20b, released August 5, 2025
  openrouter.ai/openai/gpt-oss-20b:free
  Accessed: September 12 - September 16, 2025
- **Qwen** Qwen: Qwen3 30B A3B, released April 28, 2025
  openrouter.ai/qwen/qwen3-30b-a3b:free
  Accessed: September 12 - September 16, 2025

For each of these models, we set the temperature to 0 to improve replicability.

9.1 RQ7: Comparing LLMs with the LLM_Base Prompt

To compare LLMs, we used the LLM_Base prompt (see Figure 8 and Section 8.1) on each of the six candidate LLMs and experiment with each of the formulae in the 77F dataset introduced in Section 3.3. Table 5 lists the data for both RQ7 and RQ8. For RQ7, we focus exclusively on correctness. From the LLM_Base column in the Correctness table in Table 5, we find that DeepSeek produced the most correct answers at 91% correctness. DeepSeek far outperformed the rest of the models with the next best model being Gemini at 79% and the worst model being Qwen at 65%. To answer RQ7, correctness does vary by LLM choice; and we therefore use DeepSeek in our analysis of prompt factors in Section 8.

9.2 RQ8: Prompt Generalization Across LLMs

Next we explore RQ8: To what extent do prompt improvements generalize across LLMs? Recall in the previous subsection, we found that DeepSeek outperformed other LLMs for correctness of LTL explanations using the LLM_Base prompt. We then used DeepSeek in our explanation of the prompt factors in Section 8, which resulted in the development of our aggregate prompt (LLM_Agg). In this subsection, we continue to explore variability between LLMs. We consider whether using DeepSeek in our experiment biased the resulting prompt to only be effective with DeepSeek; or conversely, whether improvements in the LLM_Agg prompt resulted in improved quality metrics across the other models. Therefore, we explore each model independently using the LLM_Base and LLM_Agg prompt with the 77F dataset. We also aim to verify whether DeepSeek is still the best choice of LLM after the prompt has changed.

Null Hypothesis 4 *There is no observable difference between the quality scores of the LLM_Base and LLM_Agg prompts for each of (a) DeepSeek, (b) Gemini, (c) Llama, (d) Mistral, (e) GPT, and (f) Qwen.*

Our results for RQ8 are listed in Table 5, with associated box plots shown in Figure 10. We note that the results in each DeepSeek row in Table 5 are identical to the respective LLM_Base and LLM_Agg rows in Table 4, as DeepSeek was used for all data in Table 4.

Correctness. All 6 LLMs made fewer errors with the LLM_Agg prompt as compared to the LLM_Base prompt, but the difference was only statistically significant for GPT (odds ratio, 3.64; 95% CI, 1.43 to 10.19; $p = .004$). Aside from the significant improvement in GPT, the relative order of other 5 models remained the same in terms of correctness between the LLM_Base and LLM_Agg

Table 5 Data for RQ7: Quality Criteria Results Comparing LLMs with LLM_Base and LLM_Agg.

Correctness	LLM_Base		LLM_Agg		Comparison
	Percent (Score)		Percent (Score)		P-Value
DeepSeek	91% (70/77)		97% (75/77)		0.167
Gemini	79% (61/77)		88% (68/77)		0.189
GPT	70% (54/77)		90% (69/77)		0.004
Llama	66% (51/77)		69% (53/77)		0.863
Mistral	73% (56/77)		82% (63/77)		0.248
Qwen	65% (50/77)		79% (61/77)		0.072
FRE	LLM_Base		LLM_Agg		Comparison
	Median	IQR	Median	IQR	P-Value
DeepSeek	59.61	(54.30 to 66.75)	62.08	(58.00 to 67.21)	.007
Gemini	61.05	(55.63 to 64.94)	62.93	(57.52 to 66.56)	.008
GPT	58.93	(55.19 to 65.48)	60.81	(56.74 to 65.48)	.190
Llama	63.43	(57.42 to 70.43)	65.30	(60.69 to 70.50)	.251
Mistral	62.49	(56.40 to 67.30)	63.58	(57.92 to 68.05)	.345
Qwen	61.09	(54.42 to 68.45)	67.00	(61.81 to 72.73)	<.001
DC	LLM_Base		LLM_Agg		Comparison
	Median	IQR	Median	IQR	P-Value
DeepSeek	11.54	(11.13 to 12.03)	10.52	(10.13 to 11.15)	<.001
Gemini	10.16	(9.65 to 10.57)	9.74	(9.25 to 10.21)	<.001
GPT	11.86	(10.43 to 11.31)	10.27	(9.77 to 10.85)	<.001
Llama	10.22	(9.58 to 10.79)	9.71	(9.04 to 10.42)	<.001
Mistral	10.83	(10.32 to 11.34)	10.50	(9.66 to 11.12)	.001
Qwen	10.60	(9.93 to 10.89)	9.73	(8.91 to 10.13)	<.001
Coherence	LLM_Base		LLM_Agg		Comparison
	Median	IQR	Median	IQR	P-Value
DeepSeek	0.516	(0.508 to 0.525)	0.525	(0.517 to 0.530)	.004
Gemini	0.537	(0.523 to 0.556)	0.643	(0.613 to 0.698)	<.001
GPT	0.512	(0.507 to 0.521)	0.513	(0.504 to 0.524)	0.598
Llama	0.611	(0.540 to 0.668)	0.639	(0.591 to 0.680)	†.012
Mistral	0.528	(0.516 to 0.537)	0.542	(0.524 to 0.552)	.001
Qwen	0.516	(0.508 to 0.523)	0.550	(0.529 to 0.580)	<.001
Length	LLM_Base		LLM_Agg		Comparison
	Median	IQR	Median	IQR	P-Value
DeepSeek	440	(373 to 500)	315	(256 to 368)	<.001
Gemini	510	(410 to 586)	384	(328 to 494)	<.001
GPT	357	(310 to 452)	235	(166 to 314)	<.001
Llama	327	(302 to 395)	155	(116 to 281)	†<.001
Mistral	501	(451 to 579)	193	(148 to 244)	<.001
Qwen	262	(235 to 307)	120	(100 to 144)	<.001

P-Values for Correctness report significance of the odds ratio for correct scores via Fisher's Exact Test. P-values for FRE, DC, Coherence, and Length calculated using Wilcoxon paired signed-rank test, unless denoted with a †. A † denotes data with asymmetric distributions about the median, violating the assumptions of the Wilcoxon test, and the use of the *Sign Test* to calculate P-Values.

prompts. See Section 10.1 for our discussion on inter-rater reliability.

Clarity. The clarity scores for all six LLMs improved between the LLM_Base and LLM_Agg prompts across all three metrics. While all FRE scores improved, only

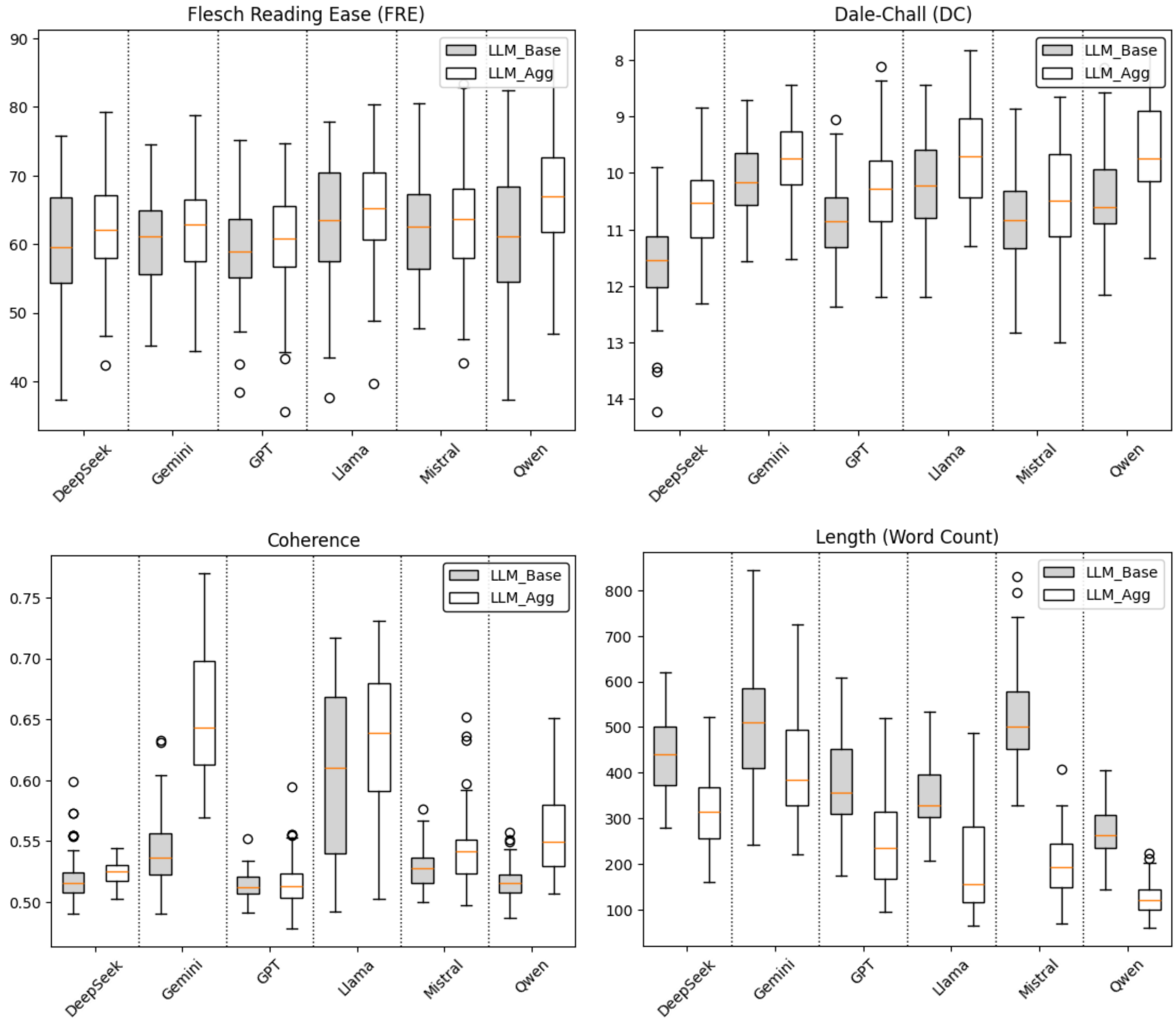


Fig. 10 Box Plots for RQ7: Quality Criteria Results Comparing LLMs with LLM_Base and LLM_Agg (Data from Table 5)

DeepSeek, Gemini, and Qwen were found to be significant at the $\alpha = .05$ level, while the improvements for GPT, Llama, and Mistral were not. The changes in all six DC scores were found to be significant, with improvements ranging from one-third of a grade-level for Mistral to one and a half grade levels for GPT. Finally, the Coherences scores were all improved and all but GPT were found to be significant. The Coherence box plots in Figure 10 illustrate the range of scores for each model. Some models (e.g., DeepSeek and GPT) are relatively consistent in terms of coherence, while other models (e.g., Gemini and Llama) have more variation. It is notable that the variability in coherence increased substantially with the LLM_Agg prompt for Gemini, and to a lesser extent for Qwen. In summary, we find that the clarity of all models were improved with the LLM_Agg prompt.

Length. The LLM_Agg prompts produced shorter explanations than the LLM_Base prompt across all six models, and this reduction was found to be significant at the $\alpha = .05$ level. The reduction in explanation length results in median explanations that were between a quarter and a half the length of the LLM_Base length. With the exception of Gemini, this decrease was more impactful across the other models than DeepSeek, see Figure 10 for box plots.

Overall. For each part in Hypothesis 4, we compare the results of the medians for the LLM_Agg prompt and the LLM_Base prompt. We reject Hypotheses 4(a) and 4(b) because we found significant differences in FRE, DC, Coherence, and Length for DeepSeek and Gemini, respectively. We reject Hypothesis 4(c) because there was significant difference in Correctness, DC, and Length for GPT. We reject Hypotheses 4(d) and 4(e) because Llama and Mistral, respectively, had significant differences in terms of DC, Coherence, and Length. Finally, we reject Hypothesis 4(f) because we observed significant changes in FRE, DC, Coherence, and Length for Qwen.

The LLM_Agg prompt was observed to yield a significantly different quality of explanations than LLM_Base for each of the LLMs in this study. Though we are unable to statistically prove that the LLM_Agg prompt led to the improved metrics, this study does indicate a preference for the LLM_Agg prompt for every LLM tested. The LLM_Agg prompt corresponded with improvements over the LLM_Base prompt in every metric for every LLM, and the improvements were statistically significant for at least three of the five metrics for each LLM.

10 Discussion and Limitations

In Section 5.9, we discussed the specific limitations of our classroom study. In Section 7.3, we critiqued the limitation of our process to create the LLM_Exp prompt. In this section, we discuss topics that cross multiple section boundaries and the limitations, and future directions of our work more generally.

10.1 Inter-rater Reliability and Adjudication of Quality

Inter-rater Reliability. In Section 8 and Section 9 we evaluated a series of prompts and LLMs, and for each combination we tested the prompt on a given LLM with the 77F dataset. Each explanation was then independently adjudicated for correctness according to Definition 1 (see Section 3.4). Given these combinations, this meant reviewing 1661 explanations totaling 615911 words (which is over 1200 single spaced pages or longer than Tolkien's "The Lord of the Rings" series, including "The Hobbit" [46]). Two authors adjudicated the 1661 explanations and achieved a good inter-rater reliability with $\kappa = 0.57$ (Cohen's kappa coefficient).

Clarity Metrics. Our evaluation of explanation quality relied on readability metrics (e.g., FRE), but these may not fully capture all aspects of effectiveness. While these automated measures assess sentence structure and complexity, they do not account for semantic accuracy or how well an explanation supports learning. Additionally, we used an online tool [64] to calculate FRE and DC scores in Section 7, but given the prospect of manually entering over one-thousand explanations into an online tool, we instead calculated the FRE and DC scores using the Textstat[3] library for Section 8 and Section 9.

Correctness of Student Summaries. In Table 3 we scored the summaries written by the students in our classroom study and found the 68/84 summaries satisfied Definition 1. Yet, over this same data when we scored the student responses for understanding in Section 5.6, we found that 73/84 summaries were sufficient. This underscores the importance of carefully defining evaluation criteria and how different pedagogical approaches have different goals.

[3] A Python library, see *https://textstat.org*

10.2 Definition 1 and Strict Correctness.

In Section 3.4, we formalized our notion of correctness with Definition 1. In this definition, we assumed a *relaxed* interpretation, which tolerates the informal linguistic conventions often used for introducing LTL in requirements engineering books (e.g., [40]) and research papers (see Section 2.1). In adjudicating the LLM explanations, we observed many examples where the explanation satisfied our definition of correctness but may confuse novices as to whether the F or U operators can be satisfied by the initial state. For example, is it correct to say that Fa means "there exists at least one future state where a holds" or must it explicitly be stated that "there exists at least one future state (including the current state) where a holds"? Under our assumption in Definition 1 both would be correct. This assumption was also present in our operator factor (see Figure 8).

In order to probe this effect, we created the LLM_Strict prompt, which is exactly the same as the LLM_Agg prompt as shown in Figure 5, except we updated the definition of the F and U operators, as denoted by the following bolded text (see fragment E in Figure 5 for comparison).

> F(p): "There must exist at least one state **(now or in the future)** where p is true."
> Until (U): Break down the behavior into two phases:
> a U b: "b must eventually become true, and until that specific moment, a must remain true in every state."
> Once b becomes true, a no longer needs to hold.
> **If b is true in the current state, there is no constraint on a.**

We created explanations for the LLM_Agg and LLM_Strict prompts with the 77F dataset on DeepSeek. We then evaluated these explanations using both a relaxed and strict interpretation of the boundary-explicit operator. This distinction is clearly reflected in the correctness results. The LLM_Agg prompt produced 75/77 relaxed-correct explanations but only 60/77 strict-correct explanations, while the LLM_Strict prompt produced 73/77 strict-correct explanations and 76/77 relaxed-correct explanations.

The relaxed scores suggests near-perfect performance; yet, the strict score reveals that many explanations fail precisely at the semantic boundaries left implicit by the informal phrasing. This gap demonstrates that relaxed correctness can mask errors that emerge when boundary conditions, such as satisfaction in the initial state or immediate fulfillment of $a\ U\ b$, are evaluated with precision. The effectiveness of boundary-explicit operator descriptions becomes evident when comparing strict correctness across the two prompt versions. After refining the operator definitions in the LLM_Strict prompt, strict correctness increased from 60 to 73. Relaxed correctness rose only slightly (from 75 to 76), confirming that the primary impact lies in reducing ambiguity at semantic edges. The improvement of thirteen explanations with the strict definition highlights the substantial effect of making boundary conditions explicit, even when relaxed correctness was already near perfect.

These results reinforce the broader insight that correctness failures may not stem from misunderstandings of *what* the temporal operators mean, but rather from how natural language frames their temporal scope. Expressions such as "in the future", "eventually", or "until" admit multiple plausible interpretations unless their boundaries are explicitly defined. This aligns with prior work, which shows that human misconceptions of LTL frequently cluster around initial-state inclusion, immediate satisfaction of $a\ U\ b$, and the interpretation of temporal distance [55,31]. Moreover, linguistic research has long documented that temporal expressions such as "in the future", "eventually", and "until" are inherently vague and context-dependent, often leading to divergent interpretations unless their boundaries are explicitly fixed [53,20].

The improvement in strict correctness also underscores that linguistic precision in operator definitions is essential when the goal is to produce explanations aligned with the strict semantics formalized in Section 3.4. These findings directly motivate the systematic prompt-design methodology presented in Section 8. By decomposing prompts into their constituent factors, analyzing each independently, and recombining them, we can construct prompts that explicitly anticipate and mitigate natural language ambiguities at semantic boundaries. This approach not only enhances semantic fidelity for LTL but also offers a principled framework for prompting across a broader range of formal notations.

10.3 Generalization of Prompt

In Section 8, we built the LLM_Agg prompt by analyzing a series of prompt factors for improvements to quality (i.e., correctness, clarity, and length) of explanations using the 77F dataset. We then verified the resulting quality of the LLM_Agg prompt on the same set of formulae. It is our understanding that each new instance of the DeepSeek model created to run these experiments did not *learn* from the repeated exposure to the 77F dataset via our multiple prompts. Nevertheless, we wanted to assure ourselves that the LLM_Agg prompt would work well for other formulae. In order to assess whether the prompt performs significantly better on the 77F dataset than other formulae, we generate explanations with DeepSeek and the LLM_Agg prompt for a new set of 20 formulae that were not previously used in any decisions leading to the design of the prompt. Only one of the 20 explanations was adjudicated incorrect, which is not significantly different from the odds of correctness found in the 77F dataset. Thus, the LLM_Agg prompt and our approach is likely generalizable to other formulae.

10.4 Additional Limitations and Improvements

While our datasets were diverse, they may not fully represent the complexity of real-world specifications. Future work can explore including domain-specific information as part of the formulae and prompt. For example, our investigation only used single letters as propositions rather than contextual information, such as Formulae 1 and 2 in our microwave example in Section 1.

Humans and LLMs have trouble translating formulae when they are not written in a natural way (e.g., $!X\ !(!g \mid !j))$; yet, these formulae can happen in the real world, especially when automatically generated by tools. In this case, future work could explore integrating our work with deductive methods to simplify the formulae and reason on their truth value. Further, some explanations generated by the LLM prompts were long and repetitive. These highly detailed explanations would be of use to novices and those learning LTL, but would be undesirable in an industry setting. As future work, we imagine personalized prompts for different purposes.

In this paper, we applied our approach to LTL, but our ultimate goal is to create a framework that can be used to generate explanations for a variety of formal languages and artifacts. In our initial approach, we started with easier formulae and elicited user feedback early in our process (via our classroom study, see Section 5). We iterated over our prompt and this approach proved successful for LTL. In future work, we will explore other formal notations. We can incorporate our framework into the vision presented by Ferrari and Spoletini [25], where our LLM-based explanations are just one part of their full cycle of FM-based development, including verification.

11 Related Work

Research at the intersection of LLMs, RE, and formal reasoning has experienced rapid growth in recent years. In this section, we first review how LLMs have been used to support RE tasks in general, then focus on work that involves formal languages and formal reasoning, followed by studies on LLM-generated explanations for RE and software artifacts. Finally, we summarize the cognitive and linguistic work on understanding temporal logic, which motivates our prompt-design methodology.

11.1 LLMs for RE Tasks.

LLMs have been used for elicitation, classification, and specification generation in RE. Ronanki et al. assessed ChatGPT's ability to generate software requirements, finding that it produces highly abstract and consistent requirements that align with structured RE processes [60]. Hymel and Johnson [34] compared LLM-generated requirements with those produced by human experts, and demonstrated that LLMs generate aligned and complete requirements faster and at a lower cost than humans; but, found that the models struggle with domain-specific nuances. Other work has shown that LLMs can identify dependencies and inconsistencies in large-scale requirement datasets with reasonable accuracy [1,66]. However, challenges remain in distinguishing between requirements relationships and traceability across evolving specifications [71].

Beyond individual case studies, a growing body of work seeks to position LLMs systematically within the RE landscape. Ferrari and Spoletini [25] present a two-way roadmap connecting formal requirements engineering and LLMs, highlighting how LLMs can both consume and generate formal artifacts, and outlining research challenges in correctness and explainability. Beg et al. provide a short survey on formalizing software requirements using LLMs [9], synthesizing emerging approaches for translating natural language requirements into formal or semi-formal representations. El-Hajjami and Salinesi propose Synthline, a product-line approach to generate synthetic RE datasets using LLMs [22], which is relevant to our own effort of constructing challenging LTL formula corpora. Ronanki et al. also offer prompt-engineering guidelines tailored to RE scenarios [59], emphasizing the importance of domain-specific prompt structure and evaluation. These works collectively underscore both the promise and the difficulty of reliably integrating LLMs into RE practice.

Our study is also connected to the broader vision of "usable formal methods," which aims to make formal notations accessible to practitioners and stakeholders [24,23]. While that line of work primarily focuses on improving the usability of formalisms and tools for humans, our contribution addresses a complementary challenge: using LLMs as intermediaries that provide semantically grounded explanations of existing formal artifacts. In particular, we investigate how LLMs can help bridge the gap between LTL formulae and natural language understanding, with a focus on correctness and mitigating the natural language ambiguities that often hinder stakeholder comprehension.

11.2 LLMs and Formal Languages.

Recent research has explored the capabilities of LLMs in working with formal languages. For example, Chen et al. [12] and Cosler et al. [17] show that LLMs can effectively generate formal logic representations with minimal training; yet, challenges remain in ensuring syntactic validity and logical coherence, as LLMs may generate incorrect or unverifiable specifications. To address these limitations, hybrid approaches combining pipeline decomposition and majority-voting mechanisms have been proposed [27], demonstrating accuracy improvements over both rule-based and end-to-end neural models. While LLMs approach human-level comprehension, they still struggle with accurately generating logical forms [50].

Several recent studies focus specifically on temporal logics and safety-critical specifications. For example, Li et al. introduce AutoSafeLTL, a self-supervised framework for automatically generating safety-compliant LTL specifications from natural language using an LLM, with counterexample-guided feedback to enforce safety properties [42]. Bellodi et al. systematically assess the (in)ability of LLMs to reason in interval temporal logic [10], providing complementary evidence that even advanced models struggle with temporal reasoning beyond simple point-based semantics. Jiang et al. evaluate whether LLMs excel in complex logical reasoning with formal languages [36], showing that performance remains fragile when deeper compositional reasoning is required. Gokhale et al. propose LogicGuard, where an LTL-based critic supervises an LLM-driven agent to improve safety and correctness in long-horizon tasks [30]. Thompson et al. introduce Rango [63], a retrieval-augmented LLM-powered prover for Coq, illustrating the emergence of hybrid LLM–formal methods tools that improve automation but do not address explainability. These works confirm that combining LLMs with temporal logic is increasingly common, but also highlight that correctness and semantic robustness are ongoing challenges.

To enhance LLM reasoning for complex tasks, various prompting strategies have been developed that leverage intermediate steps. For instance, Chain-of-thought (CoT) prompting facilitates reasoning by eliciting a series of intermediate steps, while Plan-and-Solve prompting [67] extends zero-shot CoT by explicitly instructing the LLM to devise its own plan for problem decomposition, addressing issues such as missing-step errors. More complex and computationally intensive paradigms involve multi-agent debate (MAD) [21], where multiple LLM instances collaboratively reason and reach consensus through a debate process.

In contrast, our approach employs a more prescriptive, structured prompt: rather than relying on the model to devise its own reasoning structure or engage in a multi-agent debate, we provide a predefined format and sequence for intermediate steps within the prompt itself. This guided decomposition is particularly important in the context of LTL, where a systematic breakdown of the formula structure is crucial for achieving precise and logically coherent explanations.

Ferrari et al. [25] and other recent position papers stress the need for hybrid approaches that combine LLMs with formal verification techniques [5], arguing that LLMs alone cannot guarantee correctness for safety- or mission-critical specifications. Our work aligns with this perspective: we do not aim to replace formal verification, but instead focus on enhancing the correctness and utility of natural-language explanations generated by LLMs for existing LTL formulae.

11.3 LLM-Generated Explanations for Software and RE Artifacts.

LLMs can provide automated explanations for requirements specifications, aiding in traceability and validation tasks. For example, Fuggitti and Chakraborti incorporate verification into their LLM translation of natural language descriptions into LTL formulae [28]. For RE specifications, several studies evaluate LLM-generated explanations for structured requirement formats. By transforming unstructured text into well-defined templates, LLMs can support validation tasks [39] and provide contextualized justifications for changes in evolving requirement documents [3]. Fuggitti and Chakraborti incorporate verification into their LLM-based translation from natural language descriptions into LTL formulae [28], offering explanations for correctness and counterexamples.

LLMs have also been explored in software development for explaining source code, automated code transformations, and debugging processes. Ali et al. propose combining retrieval-augmented generation with LLMs to establish traceability links between natural-language requirements and software artifacts, improving explanation quality and reducing hallucinations [2]. Research on LLM-based test case analysis examines structured explanation generation for software testing, including security-critical systems [37]. CodeEditor [43] integrates LLM-driven explanations into IDEs, providing insights into code transformations, dependency resolutions, and bug fixes.

Arora et al.'s recent REJ work on generative AI for change justification [4] shows that stakeholders value not only the correctness of LLM outputs, but also the quality and transparency of the accompanying explanations. At a broader level, Gemechu et al. analyze natural language reasoning capabilities of LLMs [29], highlighting systematic reasoning failures even when surface-level fluency is high. These findings support our focus on correctness-oriented explanations and motivate a careful examination of where and why LLM-generated explanations of formal properties may mislead readers.

Our research builds on these efforts by investigating how LLMs can enhance the explainability of formal specifications, with a focus on correctness and stakeholder comprehension for LTL formulae.

11.4 Cognitive and Linguistic Aspects of Temporal Logic Understanding

It is well documented that humans struggle with the semantics of temporal logic. Greenman et al.'s study [31] shows that misconceptions about LTL cluster around initial state inclusion, the possibility of immediate satisfaction of $a\ U\ b$, and the interpretation of temporal distance. Prasad et al. further developed a misconception-driven adaptive tutor for LTL [55], demonstrating that explicitly targeting such misconceptions in the design of instructional materials improves learning outcomes. These findings closely align with the types of errors we discussed when comparing relaxed and strict correctness of the LLM-generated explanations.

From a linguistic perspective, temporal expressions such as "in the future," "eventually," and "until" are known to be inherently vague and context-dependent. Foundational work in formal semantics by Partee [53] and Dowty [20] shows that the interpretation of tense and temporal adverbs relies heavily on discourse context and can diverge significantly from the precise semantics of formal logics. This linguistic vagueness provides theoretical grounding for our observation that explanations evaluated as relaxed-correct may still mislead readers under stricter semantic scrutiny. It also motivates our emphasis on boundary-explicit phrasing in operator definitions and on systematic prompt design to mitigate ambiguity.

12 Conclusions and Future Work

In this paper, we investigated how LLMs can generate natural-language explanations of LTL formulae and how prompt design influences the semantic quality of these explanations. Our classroom study provided initial evidence that LLM-generated explanations can support novice comprehension. Next, we substantially deepened that investigation by evaluating explanations independently of classroom performance, expanding the dataset to 77 formulae, and systematically analyzing how different prompt factors affect correctness, clarity, and concision. Using our curated gold-standard summaries, we compared explanations produced by both our base and aggregated prompts and examined how their quality varies across six LLMs. Our results show that careful prompt engineering, especially explicit structural decomposition, yields explanations that more reliably capture the semantics of LTL, even in the presence of operator precedence, nesting, and subtle boundary conditions. These findings demonstrate that LLMs can produce linguistically coherent and pedagogically meaningful explanations of formal specifications, and that these explanations can be systematically improved through targeted refinement of prompts.

Our work makes several novel contributions at the intersection of FMs and AI-assisted requirements engineering. While prior research has explored LLM capabilities in writing formal specifications, this study is among the first to rigorously evaluate the use of LLMs to explain formal notations. The potential impact of this research extends beyond educational settings to industrial practice, where FMs remain underused despite their benefits. By demonstrating that LLMs can help bridge the communication gap between formal specifications and non-expert stakeholders, our work could increase the adoption of FMs in multi-disciplinary teams, enhance requirements validation processes, and ultimately improve software quality without sacrificing the precision that makes formal methods valuable.

As next steps, we intend to examine the scalability of our approach to more deeply nested or structurally complex formulae and to evaluate how direct interaction with an LLM (rather than static explanations) affects the learning experience. Understanding how hallucinations or incorrect explanations influence trust and comprehension will be essential for assessing risks and designing safer LLM-assisted workflows.

Additionally, we plan to further refine our prompts and explore personalized prompting strategies tailored to different stakeholder groups. For example, we envision prompts designed for experts that highlight subtle semantic corner cases or assist in reviewing their own specifications. We also envision prompts for non-experts in interdisciplinary teams that emphasize conceptual clarity and domain grounding. Additionally, we aim to provide prompts for students that offer practice, feedback on common misconceptions, and progressively more challenging examples.

We also aim to integrate model-checking tools into the explanation-generation workflow, enabling the LLM to cross-check its reasoning against formal verification results, as proposed in recent work (e.g., [73, 56]). This hybrid approach has the potential to improve both correctness and trustworthiness of LLM-generated explanations.

Finally, we plan to investigate how to extend our methodology to other formal notations and develop a more comprehensive framework for generating semantically faithful explanations of formal specifications. This includes expanding our study population and exploring how explanation quality varies across domains and stakeholder needs.

References

1. Abualhaija, S., et al.: Automating Requirements Traceability with Large Language Models: Challenges and Opportunities. Journal of Requirements Engineering (2023)
2. Ali, S.J., Naganathan, V., Bork, D.: Establishing Traceability Between Natural Language Requirements and Software Artifacts by Combining RAG and LLMs. Conceptual Modeling, Lecture Notes in Computer Science **15238**, 295–314 (2025)
3. Arora, C., Grundy, J., Abdelrazek, M.: Advancing Requirements Engineering Through Generative AI: Assessing the Role of LLMs. Springer, Generative AI for Effective Software Development pp. 129–148 (2024)
4. Arora, G., Ghosh, A., Haller, A., Wiström, M.: Generative ai for change justification in requirements engineering. Requirements Engineering (2025)
5. Arora, R., Singh, A.: Prompt Engineering for LLMs in Requirements Engineering: A Systematic Analysis. In: Proceedings of the 30th IEEE International Requirements Engineering Conference (RE) (2023). URL `https://arxiv.org/abs/2310.13976`
6. Baier, Christel and Katoen, Joost-Pieter: Principles of Model Checking. MIT Press (2008)
7. Banerjee, S., Lavie, A.: METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In: Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pp. 65–72 (2005). DOI 10.3115/1073083.1073135
8. Barzilay, R., Elhadad, N., McKeown, K.R.: Inferring Strategies for Sentence Ordering in Multidocument News Summarization. In: Journal of Artificial Intelligence Research, vol. 17, pp. 35–55 (2002). DOI 10.1613/jair.989

9. Beg, A., O'Donoghue, D.P., Monahan, R.: A short survey on formalising software requirements using large language models. CoRR **abs/2506.11874** (2025). URL `https://arxiv.org/abs/2506.11874`
10. Bellodi, P., Casavecchia, P., Paparella, A., Sciavicco, G., Stan, I.E.: Assessing the (in)ability of LLMs to reason in interval temporal logic. In: 32nd International Symposium on Temporal Representation and Reasoning (TIME 2025), *Leibniz International Proceedings in Informatics (LIPIcs)*, vol. 355, pp. 4:1–4:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik (2025)
11. Bentler, P.M.: Semantic Space is (Approximately) Bipolar. Journal of Psychology **71**(1), 33–40 (1969)
12. Chen, Y., Gandhi, R., Zhang, Y., Fan, C.: NL2TL: Transforming Natural Languages to Temporal Logics using Large Language Models. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing EMNLP'23, pp. 15880–15903 (2023). DOI 10.18653/V1/2023.EMNLP-MAIN.985
13. Cherukuri, H., Ferrari, A., Spoletini, P.: Towards Explainable Formal Methods: From LTL to Natural Language with Neural Machine Translation. In: Proceedings of the 28th International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ'22), pp. 79–86 (2022)
14. Clarke, E.M., Grumberg, O., Peled, D.: Model Checking. MIT Press (1999)
15. Cohen, J.: A Coefficient of Agreement for Nominal Scales. Educational and Psychological Measurement **20**(1), 37–46 (1960)
16. Conover, W.J.: Practical Nonparametric Statistics. John Wiley & Sons (1999)
17. Cosler, M., Hahn, C., Mendoza, D., Schmitt, F., Trippel, C.: nl2spec: Interactively Translating Unstructured Natural Language to Temporal Logics with Large Language Models. In: Proceedings of the 35th International Conference on Computer Aided Verification (CAV'23), pp. 383–396 (2023). DOI 10.1007/978-3-031-37703-7_18
18. Dale, E., Chall, J.S.: A Formula for Predicting Readability. Educational Research Bulletin **27**(1), 11–28 (1948)
19. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1, pp. 4171–4186 (2019). DOI 10.18653/v1/N19-1423
20. Dowty, D.R.: Word Meaning and Montague Grammar: The Semantics of Verbs and Times in Generative Semantics and in Montague's PTQ. D. Reidel Publishing Co. (1979)
21. Du, Y., Li, S., Torralba, A., Tenenbaum, J.B., Mordatch, I.: Improving Factuality and Reasoning in Language Models Through Multiagent Debate. In: Proceedings of the 41st International Conference on Machine Learning (ICML'24) (2023)
22. El-Hajjami, A., Salinesi, C.: Synthline: A product line approach for synthetic requirements engineering data generation using large language models. In: Research Challenges in Information Science, Lecture Notes in Business Information Processing, pp. 208–225. Springer (2025)
23. Ferrari, A., Mazzanti, F., Basile, D., ter Beek, M.H.: Systematic Evaluation and Usability Analysis of Formal Methods Tools for Railway Signaling System Design. IEEE Transactions on Software Engineering **48**(11), 4675–4691 (2022). DOI 10.1109/TSE.2021.3075806
24. Ferrari, A., Mazzanti, F., Basile, D., ter Beek, M.H., Fantechi, A.: Comparing Formal Tools for System Design: A Judgment Study. In: Proceedings of the 42nd International Conference on Software Engineering (ICSE), pp. 62–74. ACM (2020). DOI 10.1145/3377811.3380370. URL `https://doi.org/10.1145/3377811.3380370`
25. Ferrari, A., Spoletini, P.: Formal Requirements Engineering and Large Language Models: A Two-way Roadmap. Information and Software Technology **181**, 107697 (2025). DOI https://doi.org/10.1016/j.infsof.2025.107697
26. Flesch, R.: A New Readability Yardstick. Journal of Applied Psychology **32**, 221–233 (1948). DOI 10.1037/h0057532
27. Fuggitti, A., et al.: Majority-voting for LTL synthesis: Leveraging Multiple Translation Strategies for Improved Formalization. Formal Methods in System Design (2023)
28. Fuggitti, F., Chakraborti, T.: NL2LTL - A Python Package for Converting Natural Language (NL) Instructions to Linear Temporal Logic (LTL) Formulas. In: AAAI Conference on Artificial Intelligence, vol. 37, p. 27068 (2023). DOI 10.1609/aaai.v37i13.27068
29. Gemechu, D., Ruiz-Dolz, R., Beyer, H., Reed, C.: Natural language reasoning in large language models: Analysis and evaluation. Findings of the Association for Computational Linguistics: ACL 2025 pp. 3717–3741 (2025)
30. Gokhale, A., Srivastava, V., Bullo, F.: Logicguard: Improving embodied LLM agents through temporal logic-based critics. CoRR **abs/2507.03293** (2025). URL `https://arxiv.org/abs/2507.03293`
31. Greenman, B., Saarinen, S., Nelson, T., Krishnamurthi, S.: Little Tricky Logic: Misconceptions in the Understanding of LTL. The Art, Science, and Engineering of Programming **7**(2) (2023)
32. Halili, S., Spoletini, P., Grubb, A.M.: Augmenting, Not Replacing: The Role of LLMs in Human-Centric Formal RE. In: Proceedings of the IEEE 33rd International Requirements Engineering Conference, pp. 218–230 (2025)
33. Hall, A.: Seven Myths of Formal Methods. IEEE Software **7**(5), 11–19 (1990)
34. Hymel, S., Johnson, T.: LLMs vs. Human Experts: A Comparative Study in Requirements Elicitation. In: Proceedings of the 46th International Conference on Software Engineering (ICSE'24) (2024)
35. Jackson, G.T., Allen, L.K., McNamara, D.S.: Common Core TERA Common Core TERA: Text Ease and Readability Assessor, p. 49. In "Adaptive Educational Technologies for Literacy Instruction", Routledge (2016)
36. Jiang, J., Wang, J., Yan, Y., Liu, Y., Zhu, J., Zhang, M., Cai, X., Gao, L.: Do large language models excel in complex logical reasoning with formal language? In: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP 2025), pp. 16889–16914. Association for Computational Linguistics, Suzhou, China (2025)
37. Jin, D., Jin, Z., Chen, X., Wang, C.: Mare: Multi-Agents Collaboration Framework for Requirements Engineering (2024)
38. Kovetz, A., Fuchs, N.E., Schwitter, R., Kaljurand, K.: Controlled Natural Language for Requirements Specification. Formal Aspects of Computing **29**(1), 133–162 (2017)
39. Krishna, M., Gaur, B., Verma, A., Jalote, P.: Using LLMs in Software Requirements Specifications: An Empirical Evaluation. Proc. of 2024 IEEE 32nd International Requirements Engineering Conference (RE) pp. 475–483 (2024)

40. van Lamsweerde, A.: Requirements Engineering in the Year 00: A Research Perspective. In: Proceedings of the 22nd International Conference on Software Engineering (ICSE'00), ICSE '00, pp. 5–19. ACM, New York, NY, USA (2000)
41. van Lamsweerde, A.: Requirements Engineering - From System Goals to UML Models to Software Specifications. Wiley (2009)
42. Li, J., Tian, M., Zhong, B.: Automatic generation of safety-compliant linear temporal logic via large language model: A self-supervised framework. CoRR **abs/2503.15840** (2025). URL `https://arxiv.org/abs/2503.15840`
43. Li, J., et al.: CodeEditor: Learning to Edit Source Code with Pre-Trained Models. ACM Transactions on Software Engineering and Methodology **32**(6) (2023)
44. Liu, J., Cao, S., Shi, J., Zhang, T., Nie, L., Hu, L., Hou, L., Li, J.: How Proficient Are Large Language Models in Formal Languages? An In-Depth Insight for Knowledge Base Question Answering. In: Findings of the Association for Computational Linguistics (ACL'24), pp. 792–815 (2024). DOI 10.18653/V1/2024.FINDINGS-ACL.45
45. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. ACM Computing Surveys **55**(9) (2023). DOI 10.1145/3560815
46. LotrProject: An Analysis of Tolkien's Books: Word Count and Density. Online at *http://lotrproject.com/statistics/books/wordscount* (2025)
47. McNamara, D.S., Graesser, A.C., McCarthy, P.M., Cai, Z.: Automated Evaluation of Text and Discourse with Coh-Metrix. Cambridge University Press (2014)
48. Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., Zettlemoyer, L.: Rethinking the Role of Demonstrations: What Makes In-Context Learning Work? In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pp. 11048–11064 (2022)
49. Minaee, S., Mikolov, T., Nikzad, N., Chenaghlu, M., Socher, R., Amatriain, X., Gao, J.: Large language models: A survey. arXiv (2025)
50. Mirzadeh, I., Alizadeh, K., Shahrokhi, H., Tuzel, O., Bengio, S., Farajtabar, M.: Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models (2024). URL `https://arxiv.org/abs/2410.05229`
51. OpenAI: GPT-4o Mini: Advancing Cost-Efficient Intelligence (2025). Available online at *https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/*, accessed 03/08/2025
52. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: A Method for Automatic Evaluation of Machine Translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318 (2002). DOI 10.3115/1073083.1073135
53. Partee, B.H.: Nominal and temporal anaphora. Linguistics and Philosophy **7**(3–4), 243–286 (1984)
54. Pnueli, A.: The Temporal Logic of Programs. In: Proceedings of the 18th IEEE Annual Symposium on Foundations of Computer Science (FOCS), pp. 46–57 (1977)
55. Prasad, S., Greenman, B., Nelson, T., Krishnamurthi, S.: A misconception-driven adaptive tutor for linear temporal logic. In: Computer Aided Verification (CAV 2025), *Lecture Notes in Computer Science*, vol. 15934, pp. 185–200. Springer (2025)
56. Ramani, K., Tawosi, V., Alamir, S., Borrajo, D.: Bridging llm planning agents and formal methods: A case study in plan verification. CoRR **abs/2510.03469** (2025). DOI 10.48550/arXiv.2510.03469. URL `https://arxiv.org/abs/2510.03469`
57. Regier, D.A., Narrow, W.E., Clarke, D.E., Kraemer, H.C., Kuramoto, S.J., Kuhl, E.A., Kupfer, D.J.: Dsm-5 field trials in the united states and canada, part ii: Test-retest reliability of selected categorical diagnoses. American Journal of Psychiatry **170**(1), 59–70 (2013). DOI 10.1176/appi.ajp.2012.12070999
58. Roggenbach, M., Cerone, A., Schlingloff, B.H., Schneider, G., Shaikh, S.A.: Formal methods for software engineering. Springer Nature Switzerland, Gewerbestrasse **11**, 6330 (2021)
59. Ronanki, K., Arvidsson, S., Axell, J.: Prompt engineering guidelines for using large language models in requirements engineering. In: Software Engineering and Advanced Applications, pp. 245–262. Springer (2025)
60. Ronanki, S., Gupta, R.: Can Large Language Models Generate Requirements? An Empirical Evaluation. arXiv preprint arXiv:2307.07381 (2023). URL `https://arxiv.org/abs/2307.07381`
61. Solet Lab: The Coh-Metrix Common Core Text Ease and Readability Assessor (T.E.R.A.). Available online at *https://soletlab.adaptiveliteracy.com*, accessed 02/15/2025
62. Spoletini, P., Ferrari, A.: The Return of Formal Requirements Engineering in theEra of Large Language Models. In: Proceedings of the 30th International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ'24), pp. 344–353 (2024)
63. Thompson, K., Saavedra, N., Carrott, P., Fisher, K., Sanchez-Stern, A., Brun, Y., Ferreira, J.F., Lerner, S., First, E.: Rango: Adaptive Retrieval-Augmented Proving for Automated Software Verification. In: Proceedings of the 47th International Conference on Software Engineering (ICSE'25) (2025)
64. Tyler, S.: Readability Analyzer. Online at *https://datayze.com/readability-analyzer* (2025)
65. Valmeekam, K., Olmo, A., Sreedharan, S., Kambhampati, S.: Large Language Models Still Can't Plan (A Benchmark for LLMs on Planning and Reasoning about Change). In: Proceedings of the Foundation Models for Decision Making Workshop (2022)
66. Vlasov, I., et al.: Investigating LLMs for Automated Requirements Classification and Refinement. IEEE Transactions on Software Engineering (2023)
67. Wang, L., Xu, W., Lan, Y., Hu, Z., Lan, Y., Lee, R.K.W., Lim, E.P.: Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL'23) (2023)
68. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E.H., Le, Q.V., Zhou, D.: Chain-of-thought Prompting Elicits Reasoning in Large Language Models. In: Proceedings of the 36th International Conference on Neural Information Processing Systems (NIPS'22) (2022)
69. Wieringa, R.: Design Science Methodology for Information Systems and Software Engineering. Springer (2014)
70. Wing, J.M.: A Specifier's Introduction to Formal Methods. IEEE Computer **23**(9), 8–24 (1990)
71. Winkler, S., et al.: Large Language Models for Requirements Engineering: A Systematic Assessment. In: Proceedings of the IEEE International Requirements Engineering Conference (RE) (2023)

72. Woodcock, J., Larsen, P.G., Bicarregui, J., Fitzgerald, J.: Formal Methods: Practice and Experience. ACM Computing Surveys **41**(4), 1–36 (2009)
73. Zhang, Y., Cai, Y., Zuo, X., Luan, X., Wang, K., Hou, Z., Zhang, Y., Wei, Z., Sun, M., Sun, J., Sun, J., Dong, J.S.: Position: Trustworthy AI agents require the integration of large language models and formal methods. In: Proceedings of the 42nd International Conference on Machine Learning (ICML 2025), vol. 267, pp. 82441–82459. Proceedings of Machine Learning Research (2025)
74. Zhao, Z., Wallace, E., Feng, S., Klein, D., Singh, S.: Calibrate Before Use: Improving Few-shot Performance of Language Models. In: Proceedings of the 38th International Conference on Machine Learning, vol. 139, pp. 12697–12706 (2021)