Towards Interpretable Formal Software Requirements: Empirical Assessment of Open-Source LLMs for LTL to NL Translation

APPENDIX

A OPEN-SOURCE LLMs

This section summarizes the open-source large language models (LLMs) evaluated in this study. We focus exclusively on models that are publicly available and can be executed locally without relying on external APIs, in order to support transparent, reproducible, and resource-controlled experimentation.

The evaluated models span a wide range of architectures, training objectives, and parameter scales, including base, instruction-tuned, and chat-oriented variants. They are drawn from several widely used open-source model families, such as LLaMA, Gemma, Qwen, Falcon, Mistral, Phi, Bloomz, and StarCoder. This diversity enables a broad assessment of LTL to NL translation behavior across different model choices and computational regimes.

Table 1 lists all 40 open-source LLMs evaluated in this study, along with their reported parameter counts (in Billions) and BF16 model sizes (in GB). This table serves as a reference for model selection and categorization throughout this empirical evaluation.

B PHASE-1 SCREENING

The objective of Phase-1 is to perform a broad capability screening of open-source LLMs to determine whether they can meaningfully interpret and translate LTL formulas into coherent NL descriptions. Since LTL to NL translation is a logic-intensive task and LLMs are not explicitly trained for formal temporal reasoning, not all models are expected to produce interpretable or semantically faithful outputs. Phase-1 therefore, serves as a filtering stage to identify models that demonstrate a minimum level of LTL interpretability before subjecting them to a more detailed and computationally expensive evaluation.

Phase-1 Experimental Setup. All candidate models are evaluated using a uniform zero-shot prompting setup to ensure a fair and unbiased comparison across models. The following prompt is used for all Phase-1 experiments:

> *"Transform the following LTL (Linear Temporal Logic) formula into a clear, precise NL (natural-language) specification, acting as a professional LTL expert. Just give the final NL output without any explanation."*

This prompt is intentionally minimal to assess the intrinsic ability of each model to interpret LTL formulas without assistance from in-context examples or explicit reasoning guidance. Each model is prompted independently with the same fixed set of 20 LTL formulas sampled across four datasets: Drone Planning (D1), CleanUp World (D2), OpenStreetMap (D3), and Synthetic NL-to-LTL (D4), ensuring coverage of diverse temporal operators, structural patterns, and atomic proposition vocabularies. Model outputs that are incoherent, grammatically incorrect, or fail to preserve the intended temporal semantics are considered unsuccessful. Models that consistently fail to produce interpretable NL descriptions are excluded from further evaluation.

Out of the 40 open-source LLMs evaluated, 33 models were able to consistently generate coherent and semantically interpretable NL descriptions for the given set of 20 LTL formulas under the zero-shot prompting setup. The remaining 7 models were excluded due to persistently incoherent outputs or their failure to preserve the intended temporal semantics of the given 20 LTL formulas.

Evaluation Criteria and Ranking. To rank the remaining 33 models during Phase-1 screening within each model size category, we combine three complementary evaluation criteria that capture translation accuracy, semantic quality, and general reasoning capability:

- *Translation Accuracy:* Assessed using manual binary judgments (correct / incorrect) by the first author, based on whether the generated NL description preserves the intended logical meaning of the corresponding reference NL description.

Table 1: Open-source LLMs evaluated. Model size are approximate and reported.

Model No.	Model Name	Parameters (B)	Model Size (GB, BF16)
M1	Llama-2-7B-Chat	7	14
M2	Llama-3.2-1B-Instruct	1	2.5
M3	Llama-3.2-3B-Instruct	3	6.5
M4	Meta-Llama-3-8B-Instruct	8	16.5
M5	Llama-2-13B-Chat	13	26
M6	Bloomz-Alpaca-560M-GGUF	0.6	1.13
M7	Bloomz-7B1-Instruct-GGUF	7	14.1
M8	Gemma-2-2B-it	2	5
M9	Gemma-3-12B-it	12	25
M10	Gemma-2-9B-it	9	19
M11	Gemma-2-2B	2	5
M12	StarCoder2-15B	16	50
M13	StarCoder2-7B	7	15
M14	StarCoder2-3B	3	7
M15	MobiLlama-1B	1	4
M16	MobiLlama-1B-Chat	1	4
M17	Mistral-7B-Instruct-v0.3	7	15
M18	Llama-3.2-3B	3	6.5
M19	Gemma-3-27B-it	27	54
M20	Qwen3-14B	14	30
M21	Qwen2.5-1.5B-Instruct	1.5	3
M22	Mistral-Small-3.2-24B-Instruct-2506	24	49
M23	Falcon3-1B-Instruct	1	3.5
M24	Falcon-H1-0.5B-Instruct	0.5	1.04
M25	Falcon-H1-34B-Instruct	34	67
M26	Falcon-H1-7B-Instruct	7	15
M27	EleutherAI-GPT-NEOX-20B	20	42
M28	CodeLlama-7B-Instruct	7	13.5
M29	Falcon-H1-1.5B-Deep-Instruct	1.5	3.2
M30	DeepSeek-LLM-67B-Chat	67	136
M31	Falcon-180B-Chat	180	361
M32	Mixtral-8x22B-Instruct-v0.1	141	295
M33	Llama-2-70B-Chat	70	140
M34	Mistral-Large-Instruct-2407-123B	123	247
M35	Llama-4-Scout-17B-16E-Instruct-109B	109	220
M36	Phi-4-14B	14	26
M37	Phi-1.5-1B	1	2.84
M38	Qwen1.5-72B	72	148
M39	Qwen1.5-110B-Chat	110	237
M40	Qwen2.5-72B-Instruct	72	145

- *Cosine Similarity:* Measures semantic alignment between the generated and reference NL descriptions using sentence-level embedding similarity.
- *Benchmark Average:* Computed as the average of commonly reported general benchmark scores (e.g., MMLU, GSM8K, ARC-C, MATH) within each model size category, serving as a proxy for general reasoning robustness.

These three criteria's are aggregated into a single composite score, which is used to rank models and select the top-performing 50% models within each size category for Phase-2 evaluation. The composite score is computed as follows:

$\text{Composite\_Score} = 0.25 \times \text{Cosine\_Similarity} + 0.50 \times \text{Accuracy} + 0.25 \times \text{Benchmark\_Average}$.

Table 2, 3, 4, and 5 summarize the Phase-1 screening results, including benchmark performance, cosine similarity, translation accuracy, composite scores, and within-category rankings. All reported scores are normalized to the $[0, 100]$ range for ease of comparison.

LTL Formulas Used in Phase-1. The following 20 examples illustrate representative LTL formulas used for Phase-1 screening.

1. $F(landmark\_1) \,\&\, G(green\_room)$
   Go to landmark one by navigating only within the green room.
2. $F(orange\_room \,\&\, !\, red\_room)$
   Go to the orange room without going to the red room.
3. $F(red\_room)$
   Go to the red room.
4. $F((red\_room \,|\, yellow\_room) \,\&\, F(blue\_room))$
   Go through either the yellow room or the red room to eventually reach the blue room.
5. $F(Bookstore \,\&\, F(Science\_Library))$
   Find the Bookstore and then find the Science Library.
6. $G(!\, Brook\_St) \,\&\, F(Bookstore \,\&\, F(Science\_Library))$
   Do not enter Brook Street; first find the Bookstore and then find the Science Library.
7. $G(!(bxgEyRbc))$
   It will never happen that *bxgEyRbc* occurs.
8. $G(gghQaxvS \rightarrow F(wShUSJdl \,\&\, zLWapQMs))$
   After *gghQaxvS*, eventually both *wShUSJdl* and *zLWapQMs* occur.
9. $G(JFcwdUeIa \,|\, gdHweLFWxDqsFy)$
   It is always either *JFcwdUeIa* or *gdHweLFWxDqsFy* holds.
10. $F(GdiDl\_O \,|\, aUzaJLsVxl\_p)$
    Eventually, either GdiDl_O or aUzaJLsVxl_p.
11. $G((VjFeAl \,\&\, pgCCaAqWFWIVqV) \rightarrow F(fXWrgVJZjSEHK \,\&\, oaDVhvA))$
    After both VjFeAl and pgCCaAqWFWIVqV, eventually both fXWrgVJZjSEHK and oaDVhvA occur.
12. $F(bar\_is\_up \,\&\, elevator\_falls \,\&\, motorbike\_started)$
    Eventually, the bar is up, the elevator falls, and the motorbike has started.
13. $G(elevator\_is\_open \rightarrow G(train\_derails \rightarrow F(semaphore\_is\_green)))$
    Whenever the elevator is open, if the train derails, then eventually the semaphore is green.
14. $G(engine\_starts \rightarrow G(car\_starts \rightarrow F(train\_is\_crossing)))$
    The engine starts and, as a consequence, whenever a car starts then at some point in time the train is crossing.
15. $G((manager\_collect\_claims \,\&\, bridge\_closes) \rightarrow G(!(elevator\_falls \,\&\, bar\_is\_up)))$
    Whenever both a manager collects claims and a bridge closes, it is never the case that both the elevator falls and the bar is up.
16. $G(!(computer\_starts \,\&\, monitor\_lights\_up \,\&\, cpu\_is\_cooled\_down))$
    Absolutely never, the computer starts, the monitor lights up, and the CPU is cooled down.
17. $G(mouse\_clicks \rightarrow F(computer\_stops \,\&\, model\_is\_overfitted))$
    If a mouse clicks then at some point in time the computer stops and the model is overfitted.
18. $G((model\_underfits \,\&\, mouse\_captures\_input) \rightarrow G(fridge\_is\_empty \,\&\, model\_overfits))$
    After both the model underfits and the mouse captures input, it is always the case that the fridge is empty and the model overfits.
19. $G((car\_starts \,|\, elevator\_falls) \rightarrow G(sensor\_captures\_data \,|\, engine\_breaks))$
    Whenever either the car starts or the elevator falls then all the time a sensor captures data or the engine breaks.
20. $G(!bridge\_closes) \,|\, F(bridge\_closes \,\&\, F(table\_is\_brown \,\&\, bar\_is\_down))$
    Initially, the bridge does not close; later, after the bridge closes, and then, at some point both the table is brown and the bar is down.

Table 2: Phase-1 screening results for Tiny and Small categories. General Benchmarks include MMLU, GSM8K, MATH, and ARC-C. BA denotes Benchmark Average, CS denotes Cosine Similarity, and Acc denotes Translation Accuracy. Models are ranked within each category using the composite score (Total).

Category	Model No.	General Benchmarks				BA	CS	Acc	Total	Rank
		MMLU	GSM8K	MATH	ARC-C					
Tiny Models	M2	49.3	44.4	30.6	59.4	45.93	73.21	85	72.28	4
	M8	43.3	19.5	17.6	46.3	31.68	77.49	95	74.79	3
	M11	42.3	17.7	11.8	42.1	28.48	76.76	85	68.80	6
	M21	60.9	68.5	35.0	54.7	54.78	78.47	85	75.81	2
	M23	43.9	38.6	11.0	45.9	34.85	75.55	80	67.60	7
	M24	53.4	68.3	48.4	37.8	51.98	79.27	85	71.31	5
	M29	66.1	82.3	67.8	43.8	65.00	85.17	100	87.54	1
	M37	37.6	40.2	37.7	44.4	39.98	67.40	60	56.84	8
Small Models	M1	45.3	53.4	14.6	61.3	43.65	74.00	80	69.41	7
	M3	63.4	77.7	48.0	78.6	66.93	71.29	85	77.05	5
	M4	68.4	79.6	30.0	78.6	64.15	77.53	95	82.92	3
	M7	–	–	–	–	–	70.96	80	–	–
	M10	71.3	68.6	36.6	68.4	61.23	81.34	95	83.14	2
	M13	–	40.4	–	–	–	78.53	90	–	–
	M17	60.1	52.2	13.1	55.5	45.22	78.53	90	75.93	6
	M18	58.0	77.7	48.0	69.1	63.20	76.37	85	77.39	4
	M26	76.8	81.6	73.4	59.9	72.93	82.20	100	88.78	1

Table 3: Phase-1 screening results for the Medium category. General Benchmarks include MMLU, GSM8K, MATH, HumanEval, GPQA, and MBPP.

Category	Model No.	General Benchmarks						BA	CS	Acc	Total	Rank
		MMLU	GSM8K	MATH	HumanEval	GPQA	MBPP					
Medium Models	M5	54.8	28.7	3.9	18.3	31.6	30.6	27.98	81.24	85	69.80	7
	M9	71.9	94.4	83.8	85.4	25.4	60.4	70.22	76.97	90	81.79	6
	M12	–	–	–	–	–	–	–	80.56	85	–	–
	M19	76.9	95.9	89.0	87.8	24.3	65.6	73.25	81.76	90	83.75	3
	M20	81.0	92.4	62.0	72.2	39.9	73.4	70.15	81.64	100	87.94	2
	M22	80.5	85.3	69.0	82.9	44.4	68.3	71.73	82.64	100	88.59	1
	M25	84.0	83.6	83.8	87.2	41.5	83.8	77.32	77.63	90	83.73	4
	M36	84.8	92.2	80.4	82.6	56.1	85.2	81.61	71.50	90	83.27	5

Table 4: Phase-1 screening results for the Large category. General Benchmarks include MMLU, GSM8K, MATH, HumanEval, ARC-C, and MBPP.

Category	Model No.	General Benchmarks						BA	CS	Acc	Total	Rank
		MMLU	GSM8K	MATH	HumanEval	ARC-C	MBPP					
Large Models	M33	68.9	56.8	35.2	29.9	57.4	45.0	48.86	65.19	75	66.01	2
	M40	86.1	95.8	83.1	86.6	86.3	88.2	87.68	77.34	90	86.25	1
	M30	71.1	84.1	32.6	73.8	64.1	61.4	64.51	53.30	70	64.45	4
	M38	77.5	79.5	34.1	41.5	59.2	53.4	59.31	61.59	70	65.22	3

Table 5: Phase-1 screening results for the Ultra-Large category. General Benchmarks include MMLU, GSM8K, MATH, HellaSwag, ARC-C.

Category	Model No.	General Benchmarks					BA	CS	Acc	Total	Rank
		MMLU	GSM8K	MATH	HellaSwag	ARC-C					
Ultra-Large	M35	69.4	74.0	95.0	90.0	90.3	83.74	69.61	80	78.33	2
	M31	70.6	65.4	74.9	87.0	63.0	72.18	62.95	85	76.28	3
	M39	76.5	84.5	42.0	87.5	69.6	72.02	57.89	60	62.47	4
	M32	74.0	89.1	47.4	88.7	70.7	73.98	64.14	90	79.53	1

C PHASE-2 EVALUATION

This section documents the LTL formulas used in Phase-2 of the empirical evaluation. Phase-2 focuses on a detailed comparison of selected 16 models under multiple prompting strategies. To ensure transparency and reproducibility, we report the exact LTL formulas evaluated in this phase, together with their corresponding NL reference descriptions.

A total of 80 LTL formulas were used in Phase-2. These formulas were sampled from the same 4 datasets: D1, D2, D3, and D4 with a broader coverage of temporal structures and atomic proposition vocabularies. The selected examples span navigation tasks, safety constraints, response patterns, abstract temporal behaviors, and include both symbolic and human-readable atomic propositions. For clarity, the Phase-2 formula–description pairs are grouped by dataset below.

Drone Planning Dataset (D1).

- $F(red\_room \,\&\, F(landmark\_3))$
  Go to the red room, then go to landmark three.
- $F(third\_floor \,\&\, F(green\_room))$
  Go through the third floor and navigate to the green room.
- $!(purple\_room)\, U\, (third\_floor)$
  Go to the third floor while avoiding the purple room.
- $!(green\_room)\, U\, (landmark\_2)$
  Do not go into the green room until reaching landmark two.
- $F(third\_floor \,\&\, !green\_room)$
  Always avoid the green room and navigate to the third floor.
- $F(landmark\_1 \,\&\, !red\_room)$
  Avoid the red room and go to landmark one.
- $F(landmark\_3) \,\&\, G(third\_floor)$
  Go to landmark three and remain on the third floor.
- $F(landmark\_1) \,\&\, G(green\_room)$
  Go to landmark one by navigating only within the green room.
- $F(green\_room)$
  Go to the green room.
- $F(landmark\_1)$
  Navigate directly to landmark one.

CleanUp World Dataset (D2).

- $F(green\_room \,\&\, F(blue\_room))$
  Enter the blue room through the green room.
- $F(green\_room \,\&\, F(blue\_room))$
  Enter the blue room through the green room.
- $F((red\_room \,|\, yellow\_room) \,\&\, F(green\_room))$
  Go through the yellow or red room to reach the green room.
- $F(blue\_room) \,\&\, G(!green\_room)$
  Go to the blue room without going to the green room.
- $F(red\_room) \,\&\, G(!yellow\_room)$
  Go to the red room and never enter the yellow room.
- $F(green\_room) \,\&\, G(!blue\_room)$
  Go to the green room but avoid the blue room.
- $F(red\_room)$
  Go to the red room.
- $F(blue\_room)$
  Enter the blue room.
- $F((red\_room \,|\, yellow\_room) \,\&\, F(blue\_room))$
  Go through the yellow or red room to reach the blue room.
- $F((red\_room \,|\, blue\_room) \,\&\, F(green\_room))$
  Go through the red or blue room to reach the green room.

OpenStreetMap Dataset (D3).

- $F(Citizens\_Bank \,\&\, F(Chipotle))$
  Find Citizen's Bank and then find Chipotle.
- $F(CVS \,\&\, F(Watson\_Center))$
  Go to CVS and then go to Watson Center.
- $F(Barus\_Building \,\&\, F(CVS))$
  Find Barus Building and then find CVS.
- $G(!Angell\_St) \,\&\, F(CVS)$
  Do not enter Angell Street and find CVS.
- $G(!Angell\_St) \,\&\, F(Barus\_Building)$
  Stay away from Angell Street and find Barus Building.
- $G(!Brook\_St) \,\&\, F(Bookstore \,\&\, F(Science\_Library))$
  Avoid Brook Street and first go to the Bookstore and then the Science Library.
- $G(Nelson\_Fitness\_Center) \,\&\, F(Chipotle \,\&\, F(Nelson\_Fitness\_Center))$
  Always visit Nelson Fitness Center, and find Chipotle after that.
- $G(Science\_Library) \,\&\, F(Starbucks \,\&\, F(Science\_Library))$
  Stay in the Science Library and then go to Starbucks.
- $G(Barus\_Building) \,\&\, F(Watson\_Center \,\&\, F(Barus\_Building))$
  Remain in Barus Building and then visit Watson Center.
- $G(!Brook\_St) \,\&\, F(Science\_Library)$
  Do not enter Brook Street but find the Science Library.
- $G(!Angell\_St) \,\&\, F(FedEx\_Office)$
  Avoid Angell Street and then find the FedEx Office.
- $F(Bookstore)$
  Go to the Bookstore.
- $F(Chipotle)$
  Find Chipotle.
- $G(Science\_Library) \,\&\, F(CVS)$
  Always stay in the Science Library and eventually go to CVS.
- $G(Watson\_Center) \,\&\, F(Starbucks)$
  Remain at Watson Center and eventually go to Starbucks.

Synthetic NL-to-LTL Dataset (D4). The remaining Phase-2 instances are drawn from the Synthetic NL-to-LTL dataset and include both restricted (symbolic atomic propositions) and unrestricted (human-readable atomic propositions) variants. These examples cover safety constraints, response patterns, nested temporal dependencies, and are used to evaluate generalization beyond navigation domains.

Restricted Variant.

- $G((iwBoSK \,\&\, NpPUaHQZqPQa) \rightarrow G(!IM\_CcZj))$
  After both iwBoSK and NpPUaHQZqPQa occur, absolutely never the case IM_CcZj occurs.
- $G(!PXjEPOci)$
  Under no condition PXjEPOci occur.
- $G(!(ckojym\_vNUR \,|\, mOqug)) \,|\, F(ckojym\_vNUR \,\&\, F(mOqug))$
  Either neither ckojym_vNUR nor mOqug ever occurs, or ckojym_vNUR eventually occurs and mOqug occurs at some later point.
- $F(qeBtdFQHmb)$
  Eventually, qeBtdFQHmb occurs.
- $G(aWpSUYgiTCy \rightarrow G(CAcfkLQXAL \rightarrow F(dvWRlxPo)))$
  Whenever aWpSUYgiTCy holds, if CAcfkLQXAL occurs then dvWRlxPo eventually will happen.
- $G((sYCGfBAVVp \,|\, QkktvCMxcVGI) \rightarrow F(QXjvGmzx))$
  Whenever either sYCGfBAVVp or QkktvCMxcVGI occurs, QXjvGmzx eventually occurs.
- $G(vUnUHzC)$
  It is always the case that vUnUHzC holds.
- $G(qYWhgQe \rightarrow G(mIEYuyhBx \,\&\, vWAYect))$
  Whenever qYWhgQe then every time both mIEYuyhBx and vWAYect hold.
- $G(GgvGADnCuwzUs \rightarrow F(IgbAcsPKo \,|\, cenfclGT))$
  Whenever GgvGADnCuwzUs occurs, then in the future either IgbAcsPKo or cenfclGT occurs.
- $G(paPQJQPqQOPwgn \rightarrow F(GXjrgYRkt\_Gn\_ \,|\, FEsX))$
  Whenever paPQJQPqQOPwgn occurs, eventually either GXjrgYRkt_Gn_ or FEsX occurs.
- $G(!(sTqTp \,|\, zEmMXnN \,|\, ideleiA\_fgg))$
  It will not happen that sTqTp, zEmMXnN, or ideleiA_fgg.

- $G(tjVXmfFhIx \mid AQogUdFHEZkp \mid mdPPpTdDtZ)$
  It is always the case that either tjVXmfFhIx, AQogUdFHEZkp, or mdPPpTdDtZ holds.
- $F(AjZXYF\_zM \mid KfDeQSGCPTEw \mid BPZqUyExRRq)$
  Eventually, either AjZXYF_zM, KfDeQSGCPTEw, or BPZqUyExRRq occurs.
- $G(oGeoKTIhZKz\_W \rightarrow F(lOGrS \,\&\, BgSIpcmj))$
  If oGeoKTIhZKz_W occurs, then it will happen that both lOGrS and BgSIpcmj occur.
- $G((dbQVDo \,\&\, uotjbfVtn) \rightarrow G(!syBUlnQ))$
  Whenever both dbQVDo and uotjbfVtn occur, implies that it will not happen that syBUlnQ.
- $G(!JLOlbbwc) \mid F(JLOlbbwc \,\&\, F(TkfnNgDBvtbM))$
  Either JLOlbbwc never occurs, or JLOlbbwc eventually occurs and TkfnNgDBvtbM occurs at a later time.
- $G(zFVJLgTpxjAEHN \rightarrow G(Z\_AuRh \rightarrow F(nKIlX)))$
  Whenever zFVJLgTpxjAEHN occurs, if Z_AuRh occurs then nKIlX eventually occurs.
- $F(cfZvISle)$
  Eventually, cfZvISle.
- $G(ERIzBflUWqY \mid ETbuydMOmBLV \mid JVygMUXkXT)$
  Every time either ERIzBflUWqY, ETbuydMOmBLV, or JVygMUXkXT holds.
- $G(!(ApZGIwL \mid dXwhbCPbJVMuwUb \mid vAvKToY))$
  It is never the case that ApZGIwL, dXwhbCPbJVMuwUb, or vAvKToY occurs.

Unrestricted Variant.

- $G(!(escalator\_moves \,\&\, house\_is\_open \,\&\, engine\_starts))$
  It is never the case that the escalator moves, the house is open, and the engine starts.
- $G(car\_enters \,\&\, train\_derails \,\&\, car\_stops)$
  It is always the case that a car enters, a train derails, and the car stops.
- $F(table\_has\_been\_moved \,\&\, train\_has\_been\_launched \,\&\, motorbike\_falls\_down)$
  Eventually, the table has been moved, the train has been launched, and the motorbike falls down.
- $G((car\_starts \,\&\, engine\_starts) \rightarrow F(constructor\_instantiate\_objects \,\&\, engine\_stops))$
  Whenever both the car starts and the engine starts, then at a certain moment the constructor instantiates objects and the engine stops.
- $G((brake\_is\_released \mid table\_is\_brown) \rightarrow G(!(constructor\_instantiate\_objects \,\&\, house\_is\_open)))$
  Whenever either the brake is released or the table is brown, then never the constructor does not instantiate objects and house is not open.
- $G(!(table\_is\_brown \mid motorbike\_has\_started \mid motorbike\_has\_stopped))$
  It is never the case that the table is brown, the motorbike has started, or the motorbike has stopped.
- $G(escalator\_speeds\_up \,\&\, brake\_is\_pressed \,\&\, engine\_starts)$
  It is always the case that the escalator speeds up, the brake is pressed, and the engine starts.
- $F(motorbike\_falls\_down \,\&\, motorbike\_catches\_fire \,\&\, engine\_starts)$
  Eventually, the motorbike falls down, catches fire, and the engine starts.
- $G((bar\_is\_down \,\&\, brake\_is\_released) \rightarrow F(train\_derails \,\&\, brake\_is\_released))$
  Whenever both the bar is down and the brake is released then at some point a train derails and the brake remains released.
- $G((sensor\_captures\_data \mid brake\_is\_released) \rightarrow G(brake\_is\_released \mid motorbike\_has\_started))$
  Whenever either the sensor captures data or the brake is released, it is always the case that the brake is released or the motorbike has started.
- $G(!(brake\_is\_pressed \mid motorbike\_has\_started)) \mid F((brake\_is\_pressed \mid motorbike\_has\_started) \,\&\, F(table\_is\_brown \mid motorbike\_has\_started))$
  Either neither the brake is pressed nor the motorbike has started ever occurs, or eventually one of them occurs and later the table becomes brown or the motorbike has started.
- $G(semaphore\_is\_green \rightarrow G(sensor\_retrieves\_data \rightarrow F(house\_is\_built)))$
  When a semaphore is green then always when the sensor retrieves data then at some point the house is built afterwards.
- $G((semaphore\_is\_broken \,\&\, train\_is\_crossing) \rightarrow F(bridge\_opens \,\&\, train\_has\_been\_launched))$
  Whenever the semaphore is broken and the train is crossing then it will happen that the bridge opens and the train has been launched.
- $G(sensor\_gathers\_information \mid house\_is\_built)$
  It is always the case that either the sensor gathers information or the house is built.

- $G(!(car\_enters \mid car\_stops \mid engine\_starts))$
  Never, the car enters, the car stops, or the engine starts.
- $G(!(cpu\_cools\_down \,\&\, monitor\_lights\_up \,\&\, model\_is\_underfitted))$
  It is never the case that the CPU cools down, the monitor lights up, and the model is underfitted.
- $G(model\_overfits \,\&\, computer\_stops \,\&\, monitor\_lights\_up)$
  It is always the case that the model overfits, the computer stops, and the monitor lights up.
- $F(monitor\_lights\_up \,\&\, model\_overfits \,\&\, fridge\_is\_switched\_on)$
  Eventually, the monitor lights up, the model overfits, and the fridge is switched on.
- $G(model\_underfits \rightarrow F(plane\_lands \mid computer\_starts))$
  After a model underfits, sooner or later either the plane lands or the computer starts.
- $G(!(computer\_starts \,\&\, model\_is\_underfitted)) \mid F((computer\_starts \,\&\, model\_is\_underfitted) \,\&\, F(model\_is\_underfitted \,\&\, cpu\_is\_cooled\_down))$
  Either the computer never starts while the model is underfitted, or eventually both occur and later the CPU is cooled down.
- $G(cpu\_overheats \rightarrow G(mouse\_captures\_input \rightarrow F(cpu\_is\_overheated)))$
  When the cpu overheats then whenever a mouse captures input then at some point in time the cpu is overheated afterwards.
- $G((computer\_stops \mid model\_underfits) \rightarrow F(cpu\_overheats \,\&\, mouse\_captures\_input))$
  If the computer stops or the model underfits, eventually the CPU overheats and the mouse captures input.
- $G(computer\_stops \,\&\, monitor\_shows\_output \,\&\, fridge\_is\_switched\_on)$
  It is always the case that the computer stops, the monitor shows output, and the fridge is switched on.
- $G(computer\_stops \mid mouse\_clicks) \rightarrow G(!(plane\_lands \,\&\, fridge\_is\_switched\_on))$
  Always either the computer stops or the mouse clicks, implies that it will not happen that a plane lands and the fridge is switched on.
- $G(!(computer\_starts \mid fridge\_is\_empty \mid monitor\_shows\_output))$
  Under no circumstances a computer starts, the fridge is empty, or the monitor shows output.

D PROMPTING TEMPLATES

This section presents the complete prompt templates used in Phase-2 to evaluate the impact of different prompting strategies on LTL to NL translation. Each prompt takes an LTL formula as input and instructs the model to generate an NL description of the formula. Depending on the prompting strategy, some prompts additionally request explanations or step-by-step reasoning in the output.

The seven prompting strategies considered in this study span zero-shot, few-shot, explanation-augmented, and chain-of-thought prompting, as well as their combinations. These strategies are adapted from prior work on NL to LTL translation and formal reasoning tasks. To suit the LTL to NL translation setting, the original prompt formulations were modified primarily by adjusting task instructions and output constraints to emphasize faithful NL interpretation of LTL formulas while preserving the core structure of the original prompts used in earlier studies.

For all few-shot variants, a fixed set of seven in-context examples drawn across 4 datasets was used consistently across all prompts. Reusing the same examples across prompting strategies avoids introducing variability due to example selection and ensures a fair comparison. The following lists the exact prompt templates used for each of the seven prompting strategies (P1-P7).

P1: Zero-shot. Transform the following LTL formula into a clear, precise natural-language specification, acting as a professional LTL expert.
Requirements: Preserve the original logical structure and all temporal / boolean constraints. Ensure that every atomic proposition and operator (`G`, `F`, `X`, `U`, `&`, `|`, `!`, $\rightarrow$) is faithfully reflected in the wording. Output only the English description; do not include any extra commentary, LTL syntax, or other formatting.

P2: Zero-shot explanation-augmented. Translate the following LTL formula into a clear, concise natural-language sentence and explain your translation. Use only the operators `G`, `F`, `X`, `U`, `&`, `|`, `!`, `->`, and parentheses.
Output Format.

1. Natural Language Translation: `<English sentence here>`
2. Explanation: `<Explanation of how the LTL formula maps to the English description>`

P3: Few-shot. Your task is to translate Linear Temporal Logic (LTL) formulas into clear, unambiguous English utterances. Use the examples below as guidance, then translate the input formula.
In-context Examples.

1. LTL: $!(\textit{green\_room})\ U\ (\textit{landmark\_2})$
   Utterance: Don't go into the green room until going to landmark two.
2. LTL: $F(\textit{green\_room})\ \&\ G(!\textit{blue\_room})$
   Utterance: Go to the green room but avoid the blue room.
3. LTL: $G(!\textit{Brook\_St})\ \&\ F(\textit{Bookstore}\ \&\ F(\textit{Science\_Library}))$
   Utterance: Do not enter Brook St; first find Bookstore and then find Science Library.
4. LTL: $G(\textit{JFcwdUeIa} \mid \textit{gdHweLFWxDqsFy})$
   Utterance: Always, JFcwdUeIa or gdHweLFWxDqsFy must hold.
5. LTL: $G((\textit{VjFeAl}\ \&\ \textit{pgCCaAqWFWIVqV}) \rightarrow F(\textit{fXWrgVJZjSEHK}\ \&\ \textit{oaDVhvA}))$
   Utterance: After both VjFeAl and pgCCaAqWFWIVqV, eventually both fXWrgVJZjSEHK and oaDVhvA occur.
6. LTL: $G(!(\textit{computer\_starts}\ \&\ \textit{monitor\_lights\_up}\ \&\ \textit{cpu\_is\_cooled\_down}))$
   Utterance: Absolutely never that the computer starts, the monitor lights up, and the CPU is cooled down.
7. LTL: $G((\textit{car\_starts} \mid \textit{elevator\_falls}) \rightarrow G(\textit{sensor\_captures\_data} \mid \textit{engine\_breaks}))$
   Utterance: Whenever either the car starts or an elevator falls then, all the time a sensor captures data or the engine breaks.

P4: Few-shot with reasoning. Translate the given LTL formula into a clear, concise English sentence. You are provided with worked examples below, each including step-by-step explanations. For the input formula, output only the final English translation; do not include any explanations.
In-context Examples

1. LTL: $!(\textit{green\_room})\ U\ (\textit{landmark\_2})$
   Explanation:
   - "*green_room*" means entering the green room.
   - "$!$*(green_room)*" means do not enter the green room.
   - "*landmark_2*" means reaching landmark two.
   - "U" means until.

   Final English: Don't go into the green room until going to landmark two.
2. LTL: $F(\textit{green\_room})\ \&\ G(!\textit{blue\_room})$
   Explanation:
   - "*green_room*" means going to the green room.
   - "$F(\textit{green\_room})$" means eventually go to the green room.
   - "*blue_room*" means entering the blue room.
   - "$!$*blue_room*" means do not enter the blue room.
   - "$G(!\textit{blue\_room})$" means always avoid the blue room.
   - "&" means and.

   Final English: Go to the green room but avoid the blue room.
3. LTL: $G(!\textit{Brook\_St})\ \&\ F(\textit{Bookstore}\ \&\ F(\textit{Science\_Library}))$
   Explanation:
   - "$!$*Brook_St*" means do not enter Brook St.
   - "$G(!\textit{Brook\_St})$" means always avoid Brook St.
   - "$F(\textit{Science\_Library})$" means eventually find the Science Library.
   - "*Bookstore* & $F(\textit{Science\_Library})$" means find the Bookstore and then eventually find the Science Library.
   - "$F(\cdot)$" means eventually.
   - "&" means both conditions must hold.

   Final English: Do not enter Brook St; first find Bookstore and then find Science Library.
4. LTL: $G(\textit{JFcwdUeIa} \mid \textit{gdHweLFWxDqsFy})$
   Explanation:
   - "*JFcwdUeIa*" means JFcwdUeIa holds.
   - "*gdHweLFWxDqsFy*" means gdHweLFWxDqsFy holds.
   - "$\mid$" means either.
   - "$G(\cdot)$" means always.

   Final English: Always, JFcwdUeIa or gdHweLFWxDqsFy must hold.
5. LTL: $G((\textit{VjFeAl}\ \&\ \textit{pgCCaAqWFWIVqV}) \rightarrow F(\textit{fXWrgVJZjSEHK}\ \&\ \textit{oaDVhvA}))$
   Explanation:
   - "*VjFeAl*" means VjFeAl occur.
   - "*VjFeAl*" and "*pgCCaAqWFWIVqV*" means both events occur.
   - "*fXWrgVJZjSEHK*" and "*oaDVhvA*" means fXWrgVJZjSEHK and oaDVhvA occur.
   - "$\rightarrow$" means if-then.
   - "$F(\cdot)$" means eventually.
   - "$G(\cdot)$" means always.

   Final English: After both VjFeAl and pgCCaAqWFWIVqV, eventually both fXWrgVJZjSEHK and oaDVhvA occur.

6. LTL: $G(!(computer\_starts \ \& \ monitor\_lights\_up \ \& \ cpu\_is\_cooled\_down))$
   Explanation:
   - "computer_starts" means the computer starts.
   - "monitor_lights_up" means the monitor lights up.
   - "cpu_is_cooled_down" means the CPU is cooled down.
   - "!" negates the conjunction.
   - "$G(\cdot)$" means always.

   Final English: Absolutely never that the computer starts, the monitor lights up, and the CPU is cooled down.
7. LTL: $G((car\_starts \mid elevator\_falls) \rightarrow G(sensor\_captures\_data \mid engine\_breaks))$
   Explanation:
   - "$car\_starts \mid elevator\_falls$" means either the car starts or an elevator falls.
   - "$sensor\_captures\_data \mid engine\_breaks$" means either a sensor captures data or the engine breaks.
   - "$G(sensor\_captures\_data \mid engine\_breaks)$" means always, either a sensor captures data or the engine breaks.
   - "|" denotes logical disjunction (either).
   - "$\rightarrow$" denotes logical implication.
   - "$G(\cdot)$" denotes the globally (always) temporal operator.

   Final English: Whenever either the car starts or an elevator falls then, all the time a sensor captures data or the engine breaks.

P5: Few-shot explanation-augmented. You are a Linear Temporal Logic (LTL) expert. Translate the following LTL formula into concise, precise natural-language (NL) sentences and explain your translation step by step. Remember that X means "next", U means "until", G means "globally" (always), and F means "finally" (eventually). Parentheses specify operator precedence and group subformulas together. You must answer in the following format: [
Explanation: `<Step-by-step reasoning>`
So the Final NL: `<Final natural-language translation>`
]
In-context Examples.

1. LTL: $!(green\_room) \ U \ (landmark\_2)$
   Explanation:
   - "$green\_room$" means entering the green room.
   - "$!(green\_room)$" means do not enter the green room.
   - "$landmark\_2$" means reaching landmark two.
   - "U" means until.

   Final NL: Don't go into the green room until going to landmark two.
   FINISH
2. LTL: $F(green\_room) \ \& \ G(!blue\_room)$
   Explanation:
   - "$green\_room$" means going to the green room.
   - "$F(green\_room)$" means eventually go to the green room.
   - "$blue\_room$" means entering the blue room.
   - "$!blue\_room$" means do not enter the blue room.
   - "$G(!blue\_room)$" means always avoid the blue room.
   - "&" means and.

   Final NL: Go to the green room but avoid the blue room.
   FINISH
3. LTL: $G(!Brook\_St) \ \& \ F(Bookstore \ \& \ F(Science\_Library))$
   Explanation:
   - "$!Brook\_St$" means do not enter Brook St.
   - "$G(!Brook\_St)$" means always avoid Brook St.
   - "$F(Science\_Library)$" means eventually find the Science Library.
   - "$Bookstore \ \& \ F(Science\_Library)$" means find the Bookstore and then eventually find the Science Library.
   - "$F(\cdot)$" means eventually.
   - "&" means both conditions must hold.

   Final NL: Do not enter Brook St; first find Bookstore and then find Science Library.
   FINISH
4. LTL: $G(JFcwdUeIa \mid gdHweLFWxDqsFy)$
   Explanation:
   - "$JFcwdUeIa$" means JFcwdUeIa holds.
   - "$gdHweLFWxDqsFy$" means gdHweLFWxDqsFy holds.
   - "|" means either.
   - "$G(\cdot)$" means always.

   Final NL: Always, JFcwdUeIa or gdHweLFWxDqsFy must hold.
   FINISH
5. LTL: $G((VjFeAl \ \& \ pgCCaAqWFWIVqV) \rightarrow F(fXWrgVJZjSEHK \ \& \ oaDVhvA))$
   Explanation:
   - "$VjFeAl$" means VjFeAl occur.

- "*VjFeAl*" and "*pgCCaAqWFWIVqV*" means both events occur.
- "*fXWrgVJZjSEHK*" and "*oaDVhvA*" means fXWrgVJZjSEHK and oaDVhvA occur.
- "*(VjFeAl & pgCCaAqWFWIVqV)* $\rightarrow$ $F(...)$" means if both VjFeAl and pgCCaAqWFWIVqV happen, then eventually both fXWrgVJZjSEHK and oaDVhvA happen.
- "$\rightarrow$" means if-then.
- "$F(\cdot)$" means eventually.
- "$G(\cdot)$" means always.

Final NL: After both VjFeAl and pgCCaAqWFWIVqV, eventually both fXWrgVJZjSEHK and oaDVhvA occur.
FINISH

6. LTL: $G(!(\textit{computer\_starts} \,\&\, \textit{monitor\_lights\_up} \,\&\, \textit{cpu\_is\_cooled\_down}))$
Explanation:
   - "computer_starts" means the computer starts.
   - "monitor_lights_up" means the monitor lights up.
   - "cpu_is_cooled_down" means the CPU is cooled down.
   - The conjunction represents all three events occurring together.
   - "!" negates the conjunction.
   - "$G(\cdot)$" means always.

   Final NL: Absolutely never that the computer starts, the monitor lights up, and the CPU is cooled down.
   FINISH
7. LTL: $G((\textit{car\_starts} \mid \textit{elevator\_falls}) \rightarrow G(\textit{sensor\_captures\_data} \mid \textit{engine\_breaks}))$
Explanation:
   - "$\textit{car\_starts} \mid \textit{elevator\_falls}$" means either the car starts or an elevator falls.
   - "$\textit{sensor\_captures\_data} \mid \textit{engine\_breaks}$" means either a sensor captures data or the engine breaks.
   - "$G(\textit{sensor\_captures\_data} \mid \textit{engine\_breaks})$" means always, either a sensor captures data or the engine breaks.
   - "$\mid$" means either.
   - "$\rightarrow$" means implication.
   - "$G(\cdot)$" means always.

   Final NL: Whenever either the car starts or an elevator falls then, all the time a sensor captures data or the engine breaks.
   FINISH

P6: Zero-shot CoT. You are a Linear Temporal Logic (LTL) expert. For each formula I give to you, "Think through the translation step by step", then output a concise natural-language instruction. Remember that *X* means "next", *U* means "until", *G* means "globally", *F* means "finally", and *GF* means "infinitely often". Parentheses specify operator precedence and group subformulas together.
Must answer in the following format: [
Explanation: `<Give explanation here>`
So the Final NL: `<Final natural-language translation>`
]

P7: Few-shot explanatory CoT. You are a Linear Temporal Logic (LTL) expert. Your answers must follow the output format below as indicated in the examples and adhere to the following rules:

1. Remember that *U* means "until", *G* means "globally / always", *F* means "finally / eventually", and *GF* means "infinitely often".
2. Only the following operators may appear: $\mid$, &, !, $\rightarrow$.
3. Atomic propositions must be interpreted consistently across the explanation and the final instruction.

Translate the following LTL formula into clear, step by step natural-language instructions. Show your chain-of-thought briefly, then provide the final NL instruction.

1. LTL formula: $!(\textit{green\_room})\ U\ (\textit{landmark\_2})$
Explanation dictionary:
   - "$!(\textit{green\_room})\ U\ (\textit{landmark\_2})$" : avoid the green room until landmark two is reached
   - "$\textit{green\_room}$" : green room
   - "$\textit{landmark\_2}$" : landmark two

   Final natural-language instruction: "Don't go into the green room until going to landmark two."
   FINISH
2. LTL formula: $F(\textit{green\_room}) \,\&\, G(!\textit{blue\_room})$
Explanation dictionary:
   - "$F(\textit{green\_room}) \,\&\, G(!\textit{blue\_room})$" : eventually reach the green room while always avoiding the blue room
   - "$\textit{green\_room}$" : green room
   - "$\textit{blue\_room}$" : blue room

   Final natural-language instruction: "Go to the green room but avoid the blue room." FINISH
3. LTL formula: $G(!\textit{Brook\_St}) \,\&\, F(\textit{Bookstore} \,\&\, F(\textit{Science\_Library}))$
Explanation dictionary:

- "$G(!Brook\_St)$" : always avoid Brook St
- "$F(Bookstore\ \&\ F(Science\_Library))$" : first go to the Bookstore, then eventually reach the Science Library

Final natural-language instruction: "Do not enter Brook St; first find Bookstore and then find Science Library." FINISH

4. LTL formula: $G(JFcwdUeIa \mid gdHweLFWxDqsFy)$
   Explanation dictionary:
   - "$G(JFcwdUeIa \mid gdHweLFWxDqsFy)$" : always either JFcwdUeIa or gdHweLFWxDqsFy must hold
   - "$JFcwdUeIa$" : JFcwdUeIa
   - "$gdHweLFWxDqsFy$" : gdHweLFWxDqsFy

   Final natural-language instruction: "Always, JFcwdUeIa or gdHweLFWxDqsFy must hold." FINISH
5. LTL formula: $G((VjFeAl\ \&\ pgCCaAqWFWIVqV) \rightarrow F(fXWrgVJZjSEHK\ \&\ oaDVhvA))$
   Explanation dictionary:
   - "$(VjFeAl\ \&\ pgCCaAqWFWIVqV)$" : both VjFeAl and pgCCaAqWFWIVqV occur
   - "$F(fXWrgVJZjSEHK\ \&\ oaDVhvA)$" : eventually both fXWrgVJZjSEHK and oaDVhvA occur
   - "$G((\cdot) \rightarrow (\cdot))$" : always, if the condition holds, then the consequence eventually holds

   Final natural-language instruction: "After both VjFeAl and pgCCaAqWFWIVqV, eventually both fXWrgVJZjSEHK and oaDVhvA occur." FINISH
6. LTL formula: $G(!(computer\_starts\ \&\ monitor\_lights\_up\ \&\ cpu\_is\_cooled\_down))$
   Explanation dictionary:
   - "$computer\_starts\ \&\ monitor\_lights\_up\ \&\ cpu\_is\_cooled\_down$" : all three events occur simultaneously
   - "$!(\cdot)$" : not
   - "$G(!(\cdot))$" : always never allow this to happen

   Final natural-language instruction: "Absolutely never that the computer starts, the monitor lights up, and the CPU is cooled down." FINISH
7. LTL formula: $G((car\_starts \mid elevator\_falls) \rightarrow G(sensor\_captures\_data \mid engine\_breaks))$
   Explanation dictionary:
   - "$car\_starts \mid elevator\_falls$" : either the car starts or an elevator falls
   - "$G(sensor\_captures\_data \mid engine\_breaks)$" : always either a sensor captures data or the engine breaks
   - "$G((\cdot) \rightarrow G(\cdot))$" : always, if the condition occurs, then the consequence always holds

   Final natural-language instruction: "Whenever either the car starts or an elevator falls then, all the time a sensor captures data or the engine breaks." FINISH

E ANALYSIS OF PROMPTING STRATEGIES

This section presents a quantitative analysis of the effect of various prompting strategies on LTL to NL translation performance, independent of individual models. We compute prompt-wise average performance scores within each model size category. Specifically, for a given prompting strategy and model category, we calculate the *Average Total Score* by taking the arithmetic mean of the TS values obtained by all models belonging to that category under the same prompt.

For example, the average TS for prompt P1 in the Tiny category (Avg. Ts Tiny) is computed by averaging the TS values achieved by all Tiny models when evaluated using P1 prompting strategy. This procedure is applied consistently across all prompting strategies and model size categories, resulting in a prompt-wise, category-level summary of LTL to NL translation performance. These averages provide a quantitative basis for comparing prompting strategies while minimizing the influence of individual model variability. Table 6 reports the average TS achieved by each prompting strategy within each model size category, along with an overall average computed across all evaluated model size categories, using the Phase-2 evaluation results.

Across all model sizes, few-shot prompting strategies consistently outperform zero-shot and explanation-augmented alternatives. In particular, the standard few-shot prompt (P3) achieves the highest overall average TS (58.46), followed closely by few-shot prompting with reasoning (P4) (57.63). These gains are observed consistently across Tiny, Small, Medium, Large, and Ultra-Large models, indicating that providing concrete in-context examples is a robust and model-agnostic mechanism for improving LTL interpretability.

In contrast, zero-shot prompting strategies (P1, P2, and P6) yield substantially lower average TS values across all model categories, highlighting the limited effectiveness of prompts that do not provide in-context examples. Explanation-augmented few-shot strategies (P5 and P7) exhibit mixed behavior: while

Table 6: Prompt-wise average LTL to NL translation performance across model size categories. Each cell reports the Average Total Score for a given prompting strategy within a model size category. Overall Avg. TS column reports the average TS across all evaluated models for each prompting strategy.

Prompt	Avg. TS Tiny	Avg. TS Small	Avg. TS Medium	Avg. TS Large	Avg. TS Ultra-Large	Overall Avg. TS
P1	43.56	47.91	47.49	47.30	46.32	46.52
P2	42.73	48.03	49.64	44.41	48.06	46.58
P3	53.88	58.45	**61.33**	**59.19**	**59.48**	**58.46**
P4	**53.96**	**59.04**	57.87	58.05	59.23	57.63
P5	52.67	48.58	58.25	46.62	45.57	50.33
P6	43.38	49.15	53.18	41.57	38.72	45.20
P7	51.58	55.16	57.83	43.33	48.14	51.20

they achieve competitive performance for Medium-sized models, they consistently underperform relative to P3 and P4 for Large and Ultra-Large models. This suggests that jointly generating explanations and final NL descriptions may dilute model focus, introducing a trade-off between explanation richness and translation accuracy.

Overall, the results summarized in Table 6 provide quantitative evidence that few-shot prompting (P3) is the most reliable and effective strategy for LTL to NL translation across model scales, followed by few-shot prompting with reasoning (P4). These findings directly support the conclusions drawn for RQ2 and highlight the critical role of prompt design in LTL to NL translation tasks.

F REFERENCE EXPLANATION GENERATION USING CHATGPT

This section describes the procedure and prompt used to generate the ground-truth reference explanations for the LTL to NL translation task evaluated in Phase-2. Since none of the original datasets provide explanatory text for LTL formulas, we generated reference explanations in a controlled manner to enable a fair evaluation of the quality of explanations for the explanation-augmented prompting strategies (P2, P5, P6, P7).

We designed a structured prompt and used ChatGPT to generate explanations for each of the 80 LTL-NL pairs used in Phase-2. The prompt instructs the model to produce a logically precise explanation that clearly conveys the semantics of the LTL formula in plain English while remaining faithful to the given NL description. All generated explanations were manually reviewed and verified by the first author to ensure semantic alignment, logical correctness, and consistency with both the LTL formula and the reference NL description. These validated explanations are used as proxy ground-truth reference explanations for evaluating explanation-augmented prompting strategies in Phase-2. The following prompt was provided to ChatGPT to generate explanations for all 80 LTL formulas used in Phase-2.

> You are an expert in formal logic and natural language understanding.
> Given a Linear Temporal Logic (LTL) formula and its corresponding natural language (NL) description, generate a clear, logically precise explanation that expresses the meaning of the formula in plain English.
>
> The explanation should:
>
> - Be 4-5 sentences long, written as one coherent paragraph.
> - Clearly describe the overall meaning of the LTL formula.
> - Explain the role of each temporal and logical operator (e.g., F, G, X, U).
> - Map the key phrases in the NL description to their corresponding components in the LTL formula.
> - Maintain a formal, academic tone and avoid conversational or speculative phrasing.
> - Avoid adding assumptions or information that are not implied by the given LTL formula or NL description.
> - Do not hallucinate.
>
> Now generate the explanation for:
> LTL: `<insert LTL formula here>`
> NL: `<insert NL sentence here>`