Translating Biomedical Observations into Signal Temporal Logic with LLMs using Structured Feedback

Hanna Krasowski*
University of California, Berkeley
krasowski@berkeley.edu

Lauren E. Malek*
Princeton University
lm4677@princeton.edu

Sanjit A. Seshia
University of California, Berkeley
sseshia@berkeley.edu

Murat Arcak
University of California, Berkeley
arcak@berkeley.edu

Abstract

Biomedical literature contains valuable knowledge that can be used to validate or monitor machine learning models. To leverage this knowledge for machine learning, we propose an LLM-based approach that translates natural language statements into formal Signal Temporal Logic (STL) specifications, guided by semantic and syntactic feedback. To capture temporal and logical dependencies in biomedical sentences, we design an STL grammar and apply structured syntax checking alongside embedding-based cosine similarity to ensure syntactic validity and semantic alignment. Evaluating sentences from nine biomedical publications on COVID-19, we find that our approach generates semantically correct STL specifications, with GPT-4o achieving the strongest performance. The resulting specifications can be flexibly applied to monitor model outputs or incorporated into training objectives or constraints, enabling interpretable and specification-aware learning.

1 Introduction

Deep learning is accelerating scientific discovery by transforming data into weights and biases of artificial neural networks. For example, Lin et al. [19] trained a large protein language model that efficiently and more accurately predicts the atomic resolution structure of proteins. However, the success of deep learning depends on the quality of the training data, e.g., identifying biases can be critical to obtain usable predictions for underrepresented species [13]. Additionally, trained neural networks are not directly interpretable so systematic errors are difficult to detect.

In other domains, e.g., robotics and programming languages, formal specifications are used to mitigate these challenges by verifying learned models or model outputs [30, 33], or adding abstract loss functions to the data that robustify models in low-data regimes [22]. Recently, Large Language Models (LLMs) have been successfully used to automate the translation from natural language instructions or requirements to formal languages such as Signal Temporal Logic (STL) [8, 9, 16]. These translation approaches commonly require data sets of natural language sentences and formal specification pairs for fine-tuning or lifting to simpler translations since the target specification language will be underrepresented or absent in the training set of the LLM [8, 20]. Thus, these data sets are usually synthesized and consequently do not fully represent the diversity of natural language.

*Equal contribution.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Biosecurity Safeguards for Generative AI.

In this paper, we present an LLM-based translation of natural language sentences from biomedical literature into STL by leveraging structured semantic and syntactic feedback without fine-tuning the LLM. STL is a suitable specification language for biomedical observations since it expresses temporal and logical dependencies, e.g., how cytokine levels evolve over the course of a disease or medical study. Additionally, STL is built on continuous signals, which can be flexibly used to represent concentrations or changes in species, e.g., cytokines or cells. The translated STL specifications can be applied to safeguard machine learning models [33, 12, 2] or to support training in low-data regimes by using the STL specifications as part of the objective function [18, 17, 26]. Our evaluation of biomedical sentences from nine papers on COVID-19 shows the impact of ambiguity in natural language sentences on translation quality for three LLMs. Our main contributions are:

- We propose an LLM-based translation of natural language sentences from biomedical literature to STL that ensures syntactically correct STL candidate specifications through syntactic and semantic feedback.
- We define a compact STL grammar to reduce the nestedness of STL for better interoperability and alignment with natural language sentence structures.
- We conduct an ablation of the introduced hyperparameters related to the syntactic and semantic feedback.
- We test our method on sentences from biomedical publications and observe that semantically correct sentences and less ambiguously formulated sentences lead to higher cosine similarity between the sentence embeddings of the natural language statement and the STL candidate specifications.

1.1 Related work

Our proposed approach formalizes natural language sentences from biomedical literature with LLMs while using formal tools to check syntactic and semantic correctness. Thus, we briefly discuss related literature on automatic formalization, programmatic verification of LLM outputs, and the applications of temporal logic in a biological context.

Specification mining with LLMs Formalizing natural language with LLMs is a powerful means of translating tasks [9, 7, 27] and requirements [23] for automation, focusing on applications such as robotics and software engineering. To achieve high accuracy in translations, these methods often use synthesized data sets for LLM fine tuning [20] as well as lifting techniques that simplify the temporal logic or natural language statements [8, 20]. For example, an early work by Chen et al. [8] used a synthesized dataset and lifting to translate sentences to multiple temporal logic languages and achieve a testing accuracy of above 95 percent. Compared to existing works [8, 16, 20], the natural language specifications in biomedical literature are usually more ambiguous as they are informing a human reader instead of a concise communication between engineers (i.e., requirements) or imperative statements describing expected performance (i.e., tasks). Additionally, many studies [10, 37] require a human to correct or guide the translations, which can be difficult for operators without knowledge of formal languages. A parallel study [35] introduces a similar approach that generates linear temporal logic specifications for systems biology with an LLM that is specifically trained for this application. However, our approach does not require fine-tuning an LLM and approaches the translation to temporal logic with a feedback perspective.

Formal tools combined with LLMs Combining software tools with LLMs is becoming a common way to improve the reliability of the generated output [29]. Specifically for formal methods, there have been LLMs proposed that use model checkers [23, 38], satisfiability checkers [24, 39], or syntax checkers [8, 7, 24]. The checkers produce a set of valid candidates from all generated candidates [8, 20] or provide feedback for improving the generation [7, 24]. For example, Chen et al. [7] include rule-based syntactic feedback and LLM-based semantic feedback to generate a task specification and high-level motion plan. In contrast, our work leverages syntax checking as feedback to the LLM.

Temporal logic for biological processes Formal specifications for machine learning are commonly used for verification [33], and more recently as loss functions, e.g., specifying rewards based on formal languages or using their quantitative semantics as part of the loss function [18]. In the context of biological processes, temporal logic specifications have been used as loss functions [17, 26] or

synthesized from data [3]. For example, [17] proposes to learn biomolecular models from signal temporal logic specifications by leveraging the quantitative semantics of STL as a loss function for parameter optimization and a genetic algorithm. While it is feasible to manually engineer specifications for smaller biological processes, for a large number of scientific papers or reports, this is fairly tedious and also requires expert knowledge in formal languages. Thus, our work proposes to automate this process by extracting specifications from literature.

2 LLM-based Translation to STL

Our method translates natural language specifications from biomedical literature to STL with LLMs using a compact STL grammar. Natural language is inherently ambiguous. In the context of this paper, an ambiguous sentence can be translated into more than one formal specification while all being semantically valid interpretations. We denote the source natural language statement as $\nu \sim English$, the target STL grammar as Λ, the predicted STL specification is $\hat{\Phi}_\nu \sim \Lambda$ for the natural language statement ν, and $\kappa_\Phi \sim English$ is the literal backtranslation of the STL specification Φ.

2.1 Compact Signal Temporal Logic

STL [21] is a formal language that can express spatial and temporal properties. The standard grammar consists of Boolean and temporal operators that are evaluated over signal traces. In the context of biomedical literature, the signal traces are concentrations of species in the system of interest, e.g., cytokine concentrations over time for an autoimmune disease. The standard STL grammar allows for deep nesting of temporal and logical statements, but this is unrealistic to appear in observations. Thus, we define a compact STL grammar with reduced nestedness and temporal operations that commonly appear in biomedical literature. First, let us define the atomic propositions:

$$\mu := s_t < c \,|\, s_t > c \,|\, s_t = c \,|\, \dot{s}_t > \dot{c} \,|\, \dot{s}_t < \dot{c} \,|\, \dot{s}_t = \dot{c} \text{ with} \tag{1}$$

$$c := s(t) \,|\, c_{\text{low}} \,|\, c_{\text{mid}} \,|\, c_{\text{high}} \tag{2}$$

$$\dot{c} := 0 \,|\, \dot{c}_{\text{low}} \,|\, \dot{c}_{\text{high}} \,|\, -\dot{c}_{\text{low}} \,|\, -\dot{c}_{\text{high}} \tag{3}$$

where s_t is a time-dependent signal, the proposition $s_t = c$ is interpreted as the signal being close to c , the constant $s(t)$ is the value of a signal at time point t, and the real-valued magnitude of the constants c and $\dot{c}$ are depending on species signal and thus we use a discretization that reflects common descriptions in the test (e.g., "TNF-α levels are elevated" would be represented with c_{high}). Additionally, let us define the grammar for allowed sentences based on the vocabulary

$$\phi = \mu \,|\, \mu_1 \wedge \mu_2 | \, \mu_1 \implies \mu_2 \tag{4}$$

$$\psi = F_{[t_1,t_2]} G\phi \,|\, F_{[t_1,t_2]}\phi \,|\, G_{[t_1,t_2]}\phi \tag{5}$$

$$\Phi = \phi \,|\, \psi \,|\, \Phi_1 \wedge \Phi_2 \,|\, \psi_1 \implies \psi_2 \tag{6}$$

We use the Boolean operations conjunction $\wedge$ and implication $\implies$ and the temporal operators $F_{[t_1,t_2]}G$, $G_{[t_1,t_2]}$, and $F_{[t_1,t_2]}$ since they reflect typical structure of natural language expressions in biomedical literature. The temporal operations can be translated to "eventually in the time interval t_1 and t_2, ϕ holds and continues to hold from there on", "always in the time interval t_1 and t_2", and "eventually in the time interval t_1 and t_2", respectively.

2.2 LLM-based Translation

Our approach involves repeated sampling n_z times of STL from LLMs, filtering the responses through a combination of their syntactic and semantic properties. For one sample, our overall architecture consists of an initial prompt and two feedback prompts for syntactic errors and semantic improvement, which are based on syntax checking and semantic evaluation of the cosine similarity between the natural language statement and the backtranslation of the STL specification (see Fig. 1). All syntactically correct STL specifications $\hat{\Phi}_\nu$ are gathered as an output set. In our experiments, we investigate this output set for semantic correctness (see Sec. 3). The following paragraphs detail the syntax checking, the semantic evaluation, and the prompting.

Syntactic Feedback Until a valid STL is produced or up to n_x times, we provide Λ and the model's produced candidate STL $\hat{\Phi}$ to a parser [34], which returns the location of the first erroneous character,

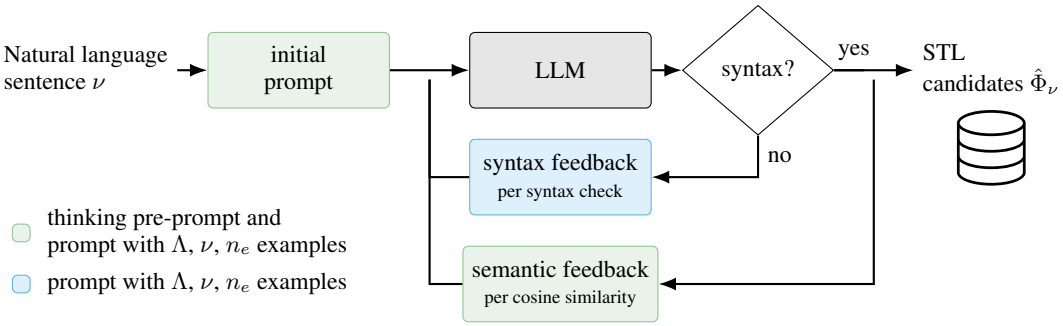

Figure 1: Overview of automated translation from natural language sentence ν to predicted STL specifications $\hat{\Phi}_\nu$. The syntax and semantic feedback loops can be run at a maximum of n_x and n_y times, respectively. The STL specification database gathers the syntactically correct specifications from all n_z samples. Examples for the prompts are provided in Appendix A.

if any, validating syntactic correctness of the candidate. As additional layers of validation and to provide more specific syntactic feedback, we further check species names, and balanced parentheses. As in [24], we attempt to remove the portion of the candidate immediately surrounding the character flagged as incorrect by the parser and prompt the models to fill the resulting hole with correct STL. Specifically, starting from the erroneous character, we search to the left and right for open and closing brackets/parentheses, respectively, and mark the portion between them as a hole. Though if there are no bounding parentheses or brackets for the syntax error character, we default to general feedback with the error message of the parser.

Semantic Feedback Next to the syntactic feedback, we also provide semantic feedback to better align the meaning of ν and syntactically correct STL candidates. To this end, we use the cosine similarity between embeddings [1, 31] of ν and the literal backtranslation $\kappa_{\hat{\Phi}_\nu}$ of the STL candidate $\hat{\Phi}_\nu$ to quantify their semantic alignment. The backtranslation is grammar-specific, and temporal adverbs, comparison statements, and time intervals are appended to the growing backtranslation as the STL candidate is processed operator-by-operator. For the semantic feedback, we track the STL candidate with the highest cosine similarity $\hat{\Phi}_{best}$ within the current sample run (i.e., for one n_z), and provide $\hat{\Phi}_{best}$ and its back translation in the semantic feedback prompt. We avoid using translations from previous runs to not bias the run towards a translation with potentially low semantic correctness.

Prompting Prompts consist of the natural language statement ν, the target STL grammar Λ and n_e examples. We use more informal language to describe Λ in the prompts as shown in Appendix A, as this helped to guide the models towards more correct usage of the operators. For the examples, we provide an STL specification that conforms to Λ and the corresponding natural language statement from the systematic back translation to natural language. Further, we use constrained decoding to ease extraction of the LLM output. In particular, we define a simple JSON schema that is usable independent of the specific STL grammar. Specifically, we request the original natural language sentence ν, an STL translation $\hat{\Phi}_\nu$, and an explanation for the translation.

We observed that using the same few examples in the prompt led to the LLMs copying the examples, but without examples the constrained decoding rate and the syntax correctness was significantly reduced. Consequently, we define a synthetic set of STL-NL pairs with balanced representation of the components of Λ, and we sample randomly from the set to obtain examples for the prompts. To generate the set, we sample STL specifications from Λ and use our backtranslation to obtain the corresponding natural language statement. Then, from this larger set, we curate an equally balanced representation of the different operations and atomic propositions. For the initial prompt and the semantic feedback, we pre-prompt the model to think about the natural language statement ν and how it could be restructured for the STL translation. We add this additional step since it improves the reliability of the models generating a meaningful output.

Table 1: Ablation for syntactic n_x and semantic n_y feedback

	n_x				n_y	
Model	1	2	3	4	1	2
Qwen	0.34	0.42	**0.49**	0.46	**0.30**	0.29
DeepSeek	0.27	0.34	0.36	**0.39**	0.14	**0.19**
GPT-4o	0.56	0.60	**0.62**	0.62	**0.33**	0.26

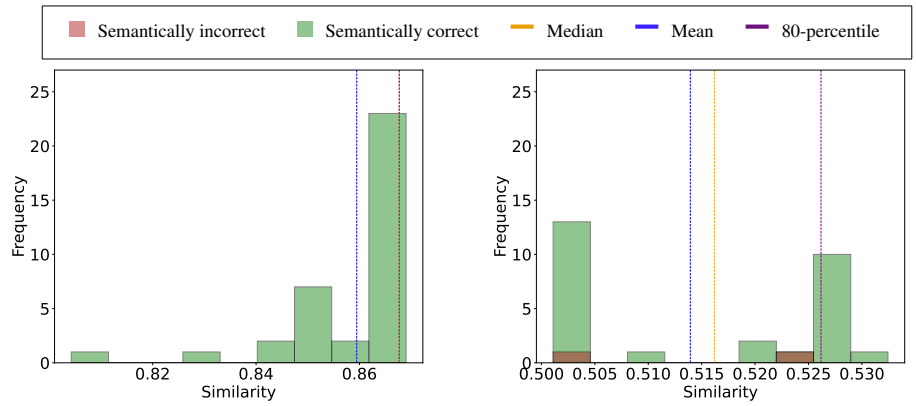

Figure 2: GPT-4o Histograms of sentence embedding cosine similarity for simpler sentence (left, Appendix B.1.3) and more ambiguous sentence (right, Appendix B.1.7) with $n_x = 3, n_y = 1, n_z = 18$.

3 Experiments

We evaluate our proposed translation approach on sentences from biomedical literature on COVID-19 [4, 6, 11, 14, 15, 28, 32, 36, 41]. Specifically, we use a validation set of 8 sentences and a test set of 16 sentences that reflect the diversity of sentences of interest (see Appendix B for details). We run our approach across three models: Qwen3-1.7B, DeepSeek-R1-Distill-Qwen-1.5B, and gpt-4o-2024-08-06. The Qwen and DeepSeek models are queried with an NVIDIA GeForce RTX 3070 Ti GPU with 8GB of memory using vLLM.

First, we conduct an ablation study on the validation set for the semantic and syntactic feedback prompts, specifically investigating the maximum number of feedback n_x and n_y. For the ablation across n_x, we set $n_y = 2$ and $n_z = 18$. The n_x ablation metric is the fraction of syntactically correct STL specifications over the maximum number of STL specifications that could have been generated (i.e., 432 for the setting of n_y and n_z and 8 natural language sentences) and is reported in Table 1. We observe that the number of optimal syntactic feedback attempts highly depend on the specific LLM, and that GPT-4o performs better than the DeepSeek and Qwen models.

Second, we investigate an appropriate value for n_y. Here, we use $n_x = 3$ and $n_z = 18$. As a metric, we compute the fraction of STL specifications where the semantic feedback led to a higher cosine similarity compared to the initial generated STL specification. The results are reported in Table 1. For DeepSeek and GPT-4o, two semantic feedback attempts lead to the best fraction of improved STL while for Qwen one semantic feedback attempt is optimal.

Since the cosine similarity of the original natural language sentence and the backtranslation of the generated STL specification is only an approximation of semantic correctness, we manually label the generated STL specifications as correct or incorrect and investigate their distribution over the cosine similarity. Exemplary resulting histograms for a simple and more ambiguous sentence to translate are shown in Fig. 2 and 4. Generally, we observe that the distribution of semantically correct and incorrect STL is often overlapping while the semantically correct translations have higher cosine similarity. Additionally, for more empirically ambiguous sentences, we obtain lower mean semantic similarity compared to more clearly written sentences.

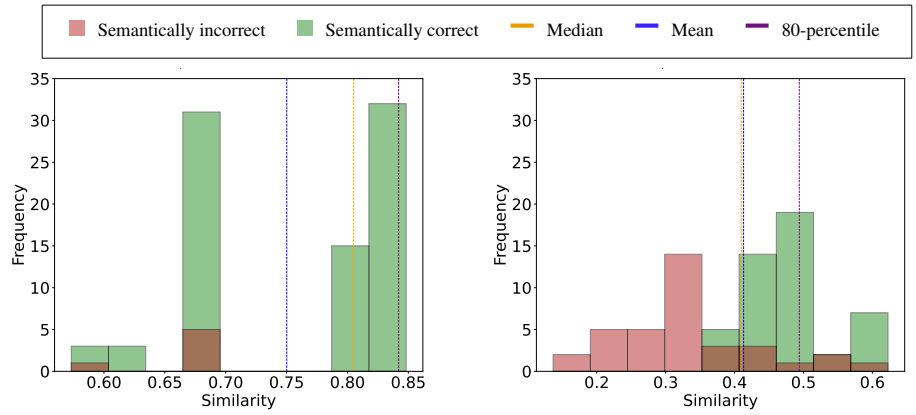

Figure 3: Histograms of sentence embedding cosine similarity for three test sentences with highest (left) and lowest (right) median cosine similarity for GPT-4o with $n_x = 3, n_y = 1, n_z = 18$.

Based on the result in Table 1, we set $n_x = 3$ and $n_y = 1$ for the test experiments. We run our LLM-based translation for each of the 16 natural language sentences 18 times. Overall, the LLM models produce syntactically correct STL for 72%, 50%, and 44% of the 576 samples for GPT-4o, Qwen, and DeepSeek, respectively. The fractions of semantically correct sentences per syntactically correct sentences are 74%, 31%, and 19%. As in the validation setting, we observe that a low mean or median of the cosine similarity reflects high ambiguity in the natural language sentence and vice versa. This becomes apparent from the histograms in Fig. 3 and 5, where we depict the distribution of cosine similarities for the three sentences with the highest median and the lowest median per LLM. For GPT-4o in Fig. 3, we observe a median of 0.81 and 0.41 for the top and bottom sentences. Further, the number of semantically correct sentences is significantly higher for the top-3 sentences. To illustrate the translations for the test sentences, we report six semantically correct and incorrect sentences for GPT-4o in Appendix C. The runtime for our translation experiments depends on the setting of n_x and n_y. For example, for $n_x = 2, n_y = 2$, and $n_z = 18$ on the test set, the runtime with Qwen, DeepSeek, and GPT-4o was around 2.5h, 2.5h, and 2h, respectively. Note that on average across the three models, there are about 200 syntactically correct STL specifications generated in this configuration.

4 Discussion and Limitations

Our numerical results show that automated translation of biomedical sentences to STL is feasible across a limited number of COVID-19 papers. We focus on COVID-19 since there is a large body of literature, which will allow us to also investigate consistency between publications in the future. We also compared the translation performance when starting from literally backtranslated STL, and observed a better semantic success rate for Qwen and GPT-4o. This underpins the higher ambiguity of natural language sentences directly taken from biomedical publications than sentences specifically written for STL. Nevertheless, a larger set of natural language sentences taken from diverse biomedical literature should be investigated to further investigate the scalability and effectiveness of our approach.

Our results suggest that a metric based on the distribution of cosine similarities would be a valid option to automatically decide semantic correctness for GPT-4o. For example, we added the 80-percentile limit to the histogram plots, which, if used as a criterion for semantic correctness, has a low false positive rate for GPT-4o. We also investigate if the SentenceTransformer model [31] used for the sentence embeddings is a critical factor for our approach. To this end, we compute the correlation based on the Spearman rank with respect to the sentence embeddings models Qwen3-Embedding[40] and Nomic Embed [25] by comparing the cosine similarities for the syntactically correct sentences produced in the validation setting $n_x = 3, n_y = 1, n_z = 18$. We obtain rank values around 0.90 and above for about 200 samples, i.e., a strong correlation between embedding models. Thus, evaluating semantic correctness with respect to statistical measures such as the 80-percentile, suggests that our approach is robust with respect to the selected sentence embedding model.

The DeepSeek model performed consistently worse. From analyzing the responses, we notice that it is more prone to include species names in STL candidates that were not part of the original natural language sentence. This suggests that adding an additional component to the syntactic feedback, which cross-references species names in STL candidates and natural language sentences, may be useful. Further, more LLMs should be tested to determine if an ensemble of multiple models [5], for which the outputs are consolidated, would be most effective.

While GPT-4o exhibits the best performance, the Qwen model also consistently produces semantically correct STL specifications. This is exciting since in medical and research settings, data privacy and budget constraints might limit the use of commercial models. Overall, translating biomedical literature into STL specifications makes this knowledge more accessible and can consequently robustify machine learning or be used for verifying and monitoring of machine learning models.

5 Conclusion

To leverage knowledge from biomedical literature for guiding and monitoring learning-based models, we introduce an LLM-based approach that uses syntactic and semantic feedback to translate biomedical sentences into STL. Our evaluation on realistic sentences from COVID-19 literature generated by three different LLMs shows our approach achieves syntactically correct STL specification candidates. The semantic correctness of the candidates is reasonable on GPT-4o, and overall suggests that cosine similarity alone is not a reliable indicator of semantic correctness. We found that the ambiguity of a natural language sentence has an indirect relationship with the average similarity of its STL translations, oftentimes regardless of semantic correctness. Future work should develop more robust metrics for automatically detecting semantic correctness, improve scalability through ensemble methods or automated consolidation of similar specifications, and expand the evaluation across larger sets of sentences and models.

Acknowledgments and Disclosure of Funding

We thank the SCALE Research Lab at UC Berkeley, Alex Beaudin, Sina Booeshaghi, Eric Palanques-Tost, Federico Mora Rocha, Ameesh Shah, and Beyazit Yalcinkaya for their helpful feedback. This work was funded in part by the Air Force Office of Scientific Research grant FA5590-23-1-0529 and by the Summer Undergraduate Program in Engineering Research at Berkeley (SUPERB).

References

[1] Mikel Artetxe and Holger Schwenk. "Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond". In: *Transactions of the association for computational linguistics* 7 (2019), pp. 597–610.

[2] Mehmet Emin Bakir et al. "Automatic selection of verification tools for efficient analysis of biochemical models". In: *Bioinformatics* 34.18 (2018), pp. 3187–3195.

[3] Ezio Bartocci, Luca Bortolussi, and Guido Sanguinetti. "Data-Driven Statistical Learning of Temporal Logic Properties". In: *Formal Modeling and Analysis of Timed Systems*. 2014, pp. 23–37.

[4] Daniel Blanco-Melo et al. "Imbalanced Host Response to SARS-CoV-2 Drives Development of COVID-19". In: *Cell* 181.5 (2020), pp. 1036–1045.

[5] Yue Cao et al. "Ensemble deep learning in bioinformatics". In: *Nature Machine Intelligence* 2.9 (2020), pp. 500–508.

[6] Alice S. Chau et al. "The Longitudinal Immune Response to Coronavirus Disease 2019: Chasing the Cytokine Storm". In: *Arthritis & Rheumatology* 73.1 (2021), pp. 23–35.

[7] Yongchao Chen et al. "AutoTAMP: Autoregressive Task and Motion Planning with LLMs as Translators and Checkers". In: *arXiv* (2024).

[8] Yongchao Chen et al. "NL2TL: Transforming Natural Languages to Temporal Logics using Large Language Models". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 2023, pp. 15880–15903.

[9] Dan BW Choe et al. "Seeing, Saying, Solving: An LLM-to-TL Framework for Cooperative Robots". In: *arXiv* (2025).

[10] Matthias Cosler et al. “nl2spec: Interactively Translating Unstructured Natural Language to Temporal Logics with Large Language Models”. In: *Computer Aided Verification*. 2023, pp. 383–396.

[11] Diane Marie Del Valle et al. “An inflammatory cytokine signature predicts COVID-19 severity and survival”. In: *Nature Medicine* 26.10 (2020), pp. 1636–1643.

[12] Jyotirmoy V Deshmukh et al. “Robust online monitoring of signal temporal logic”. In: *Formal Methods in System Design* 51.1 (2017), pp. 5–30.

[13] Frances Ding and Jacob Steinhardt. “Protein language models are biased by unequal sequence sampling across the tree of life”. In: *bioRxiv* (2024).

[14] Sara Ghaffarpour et al. “Cytokine profiles dynamics in COVID-19 patients: a longitudinal analysis of disease severity and outcomes”. In: *Scientific Reports* 15.1 (2025).

[15] Jérôme Hadjadj et al. “Impaired type I interferon activity and inflammatory responses in severe COVID-19 patients”. In: *Science* 369.6504 (2020), pp. 718–724.

[16] Jie He et al. “DeepSTL: from English requirements to signal temporal logic”. In: *Proceedings of the 44th International Conference on Software Engineering*. 2022, pp. 610–622.

[17] Hanna Krasowski et al. “Learning Biomolecular Models using Signal Temporal Logic”. In: *Annual Learning for Dynamics and Control Conference (L4DC)*. 2025, pp. 1365–1377.

[18] Karen Leung, Nikos Aréchiga, and Marco Pavone. “Backpropagation through signal temporal logic specifications: Infusing logical structure into gradient-based methods”. In: *The International Journal of Robotics Research* 42.6 (2023), pp. 356–370.

[19] Zeming Lin et al. “Evolutionary-scale prediction of atomic-level protein structure with a language model”. In: *Science* 379.6637 (2023), pp. 1123–1130.

[20] Jason Xinyu Liu et al. “Grounding Complex Natural Language Commands for Temporal Tasks in Unseen Environments”. In: *7th Annual Conference on Robot Learning*. 2023.

[21] Oded Maler and Dejan Nickovic. “Monitoring temporal properties of continuous signals”. In: *International Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems*. 2004, pp. 152–166.

[22] Kumar Manas et al. “Uncertainty-Aware Trajectory Prediction via Rule-Regularized Heteroscedastic Deep Classification”. In: *Robotics: Science and Systems (RSS)*. 2025.

[23] Daniel Mendoza, Christopher Hahn, and Caroline Trippel. “Translating Natural Language to Temporal Logics with Large Language Models and Model Checkers”. In: *Formal Methods in Computer-Aided Design (FMCAD)*. 2024, pp. 1–11.

[24] Federico Mora et al. “Synthetic Programming Elicitation for Text-to-Code in Very Low-Resource Programming and Formal Languages”. In: *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. 2024.

[25] Zach Nussbaum et al. “Nomic Embed: Training a Reproducible Long Context Text Embedder”. In: *arXiv* (2025).

[26] Eric Palanques-Tost et al. “STL-based Optimization of Biomolecular Neural Networks for Regression and Control”. In: *Accepted for Proc. of the IEEE Conference on Decision and Control (CDC)*. 2025.

[27] Jiayi Pan, Glen Chou, and Dmitry Berenson. “Data-Efficient Learning of Natural Language to Linear Temporal Logic Translators for Robot Task Specification”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2023, pp. 11554–11561.

[28] Ritu Pasrija and Mohammad Naime. “The deregulated immune reaction and cytokines release storm (CRS) in COVID-19 disease”. In: *International Immunopharmacology* 90 (2021).

[29] Aske Plaat et al. “Agentic large language models, a survey”. In: *arXiv* (2025).

[30] Zachary Ravichandran et al. “Safety Guardrails for LLM-Enabled Robots”. In: *arXiv* (2025).

[31] Nils Reimers and Iryna Gurevych. “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks”. In: *Conference on Empirical Methods in Natural Language Processing*. 2019.

[32] Vivian A. Scheuplein et al. “High Secretion of Interferons by Human Plasmacytoid Dendritic Cells upon Recognition of Middle East Respiratory Syndrome Coronavirus”. In: *Journal of Virology* 89.7 (2015), pp. 3859–3869.

[33] Sanjit A. Seshia, Dorsa Sadigh, and S. Shankar Sastry. “Toward verified artificial intelligence”. In: *Commun. ACM* 65.7 (2022), pp. 46–55.

[34] Erez Shinan. *Welcome to Lark's documentation! — Lark documentation*. 2020. URL: `https://lark-parser.readthedocs.io/en/stable/` (visited on 08/19/2025).

[35] Difei Tang and Natasa Miskov-Zivanov. "Generating Bounded Linear Temporal Logic in Systems Biology with Large Language Models". In: *bioRxiv* (2025).

[36] Sophie Trouillet-Assant et al. "Type I IFN immunoprofiling in COVID-19 patients". In: *Journal of Allergy and Clinical Immunology* 146.1 (2020), pp. 206–208.

[37] Jun Wang et al. "ConformalNL2LTL: Translating Natural Language Instructions into Temporal Logic Formulas with Conformal Correctness Guarantees". In: *arXiv* (2025).

[38] Guangyuan Wu et al. "LLM Meets Bounded Model Checking: Neuro-symbolic Loop Invariant Inference". In: *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*. 2024, pp. 406–417.

[39] Xi Ye et al. "SatLM: Satisfiability-Aided Language Models Using Declarative Prompting". In: *Advances in Neural Information Processing Systems*. Vol. 36. 2023, pp. 45548–45580.

[40] Yanzhao Zhang et al. "Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models". In: *arXiv* (2025).

[41] Yan Zhao et al. "Longitudinal COVID-19 profiling associates IL-1RA and IL-10 with disease severity and RANTES with mild disease". In: *JCI Insight* 5.13 (2020).

A Prompt examples

Initial thinking

Give an explanation of how you would translate the following natural language sentence into a signal temporal logic (STL) statement: <natural language sentence ν>

What parts of the sentence make sense to translate and what parts don't? Think carefully, step-by-step through the process, and tell us about how STL could be used to capture the original semantic meaning of the natural language sentence. Your STL response must conform to the following rules: <Rules>

Initial STL generation

Now that you have thought about how you would translate this natural language sentence to STL, please output your STL translation. Here is the natural language sentence again: <natural language sentence ν>

Format your response in JSON. Include (1) your thinking process, (2) the input sentence, and (3) your STL response. Make sure to follow the STL rules that were specified in the last prompt.

Your STL response must conform to the following rules: <Rules>

Here are some examples of how to format your output, but don't copy the STL. Make sure the STL you produce is for the input sentence that you are currently being asked to translate. <Examples>

Syntactic feedback

Your previous response was: <model's STL response>

Your response is not formatted correctly as an STL statement. This was the error that was thrown when trying to parse your STL statement: <error message>

Fix your response so that it follows the STL rules, and look at the example STL statements to better understand how the STL syntax works.

As a reminder, the sentence you are trying to translate is: <natural language sentence ν>

Format your response in JSON. Include (1) your thinking process, (2) the input statement, and (3) your STL response. Your STL response must conform to the following rules: <Rules>

Here are some examples of how to format your output, but don't copy the STL. Make sure the STL you produce is for the input sentence that you are currently being asked to translate. <Examples>

Semantic thinking

You were asked to translate the following natural language sentence into STL: <natural language sentence ν>

In response, you produced the following STL statement: <model's STL response $\hat{\Phi}_\nu$>

This statement means: <backtranslation of model's STL response $\kappa_{\hat{\Phi}_\nu}$>

Give an explanation of how you would improve your STL statement so that it is closer in meaning to the natural language sentence you were asked to translate. Your STL response must conform to the following rules: <Rules>

Semantic STL generation

Now that you have thought about how you would improve your STL statement, please output your new and improved STL translation. As a reminder, here is the natural language sentence that you are trying to translate: <natural language sentence ν>

Format your response in JSON. Include (1) your thinking process, (2) the input sentence, and (3) your STL response. Your STL response must conform to the following rules: <Rules>

Here are some examples of how to format your output, but don't copy the STL. Make sure the STL you produce is for the input sentence that you are currently being asked to translate. <Examples>

<Rules>

u : less_than | greater_than | is | derivative_greater_than | derivative_less_than | derivative_is
less_than : s(t) < c # Species s is less than c
greater_than : s(t) > c # Species s is greater than c
is : s(t) = c # Species s is close to c
derivative_greater_than : d_s(t) > d_c # The rate of change of species s is greater than d_c
derivative_less_than : d_s(t) < d_c # The rate of change of species s is less than d_c
derivative_is : d_s(t) = d_c # The rate of change species s is close to d_c
c : s(t_a) | c(low) | c(mid) | c(high) # c is the level of a species, it can be a specific value or generally just low, moderate, or high
d_c : 0 # Rate of change is 0
| d_c(low) # Species is slowly increasing
| d_c(high) # Species is rapidly increasing
| -d_c(low) # Species is slowly decreasing
| -d_c(high) # Species is quickly decreasing
predicate : u | u1 and u2 | u1 implies u2 # You can combine predicates with Boolean operators
temporal_operator : eventually[t_a,t_b]globally(predicate) # This means that between day t_a and t_b, there is a point when the predicate becomes true for the rest of the interval
| globally[t_a,t_b](phi) # This means the predicate is true over the entire interval from day t_a to t_b
| eventually[t_a,t_b](phi) # This means there is at least 1 time between days t_a and t_b that the predicate is true
t_a : number | ∞ # Time in days
s : IL6 | IL12 | IL1β | IL1Ra | TNFα | IL8 | IFNα | IFNβ | SARSCoV2 | IL1RN # Species names you can use
d_s : d_IL6 | d_IL12 | d_IL1β | d_IL1Ra | d_TNFα | d_IL8 | d_IFNα | d_IFNβ | d_SARSCoV2 | d_IL1RN # Names for derivatives of the species

<Examples>

{"thinking:" "Hmm, first I should...",
"input_sentence:" "From day 11 to 12 eventually at every point in that interval IL-1Ra was above its high levels.",
"output_STL": "eventually[11,12]globally(IL1Ra(t) > c(high))"}

{"thinking:" "Hmm, first I should...",
"input_sentence:" "From day 13 onward at every point in that interval the rate of change of IL-1β was close to 0 and IL1RN was below its high levels.",
"output_STL": "globally[13,∞](d_IL1β(t)=0 and IL1RN(t) < c(high))"}

B Natural language statements

1. Validation sentences from: 1-6 [14], 7 [28], 8 [4]

    1. In the mild and moderate groups, IL-6 concentrations were at their highest level in the first week after the symptom onset and then exhibited a decreasing trend.
    2. Remarkably, in the mild group, the amount of these cytokines (IL-1β and IL-1Ra) increased at the day 1–7, reached a peak at the day 8–14, and diminished after >14 days.
    3. TNF-α levels elevated at the day 1–7 and 8–14 times intervals, then decreased at the day>14.
    4. We detected that IL-8 was significantly elevated in all COVID-19 subgroups at three studied time intervals compared to the control group.
    5. We found that although there was no difference in the production of IFN-β in all patients with COVID-19 compared to the control group at the day 1–7, IFN-β levels were higher in moderate, severe, and critical subjects at the day 8–14 or >14 compared to the healthy control and themselves at the day 1–7.
    6. IL-12 reached its maximum level at the day>14 in mild patients.

7. It is reported that in recovered cases, within a few hrs of virus entry, both α and β-IFNs (at first day of infection) are rapidly produced and an antiviral state is soon reached.
8. By day 14, we detected no viral reads for SARS-CoV-2, and the observed cytokines returned to baseline, with the exception of IL-6 and IL1RN or IL1RA, which remained elevated, similar to results observed with MERS.

2. Test sentences from: 1 [6], 2 [28], 3 [36], 4-5 [32], 6-7 [15], 8-14 [41], 15-16 [11]

1. In patients with COVID-19, SARS-CoV-2-specific T-cells appear in peripheral blood within two weeks of symptom onset (31).
2. This is the reason that seroconversion (undetectable stage to production of IgM followed by IgG) in 100% of infected people (with positive virus-specific IgG) is achieved 17–19 days after commencement of indications [7].
3. Following day 10, IL-6 remains increased whereas IFN-α tapered.
4. Two days postinfection, permissive Vero cells produced high peak titers of $5 \times 10^6\,\mathrm{TCID\_50s/ml}$ and $1 \times 10^7\,\mathrm{TCID\_50s/ml}$ of MERS- and SARS-CoV, respectively (Fig. 1B, panel i).
5. In parallel, stimulation with CpG 2216 also resulted in lower, but clearly detectable, amounts of IFNs.
6. Circulating IL-1α also was not detected (fig. S9F).
7. Monocyte chemotactic factor chemokine(C-C motif) ligand 2 (CCL2) was increased in the blood of infected patients as well as the transcripts of its receptor CCR2; this was associated with low counts of circulating inflammatory monocytes (Fig. 4I), suggesting a rolefor the CCL2/CCR2 axis in the monocyte chemo-attraction into the inflamed lungs.
8. Then, in the mild group of patients, IP-10 levels declined from week 2 and returned back to normal on week 4.
9. IP-10 level was significantly elevated in COVID-19 patients in week 1 of onset of symptoms in both mild and severe groups when compared with healthy volunteer controls (P = 1.36 × 10–8 and 4.39 × 10–8, respectively).
10. Significantly higher levels of MCP-1 in severe cases were observed when compared with mild cases at early an time point of the infection (week 1 and 2; P = 0.047 and 8.62 × 10–5, respectively) but not at later time points (week 3 and 4; P = 0.136 and 0.030, respectively, Supplemental Table 1 and Figure 2).
11. We also found that IL-1 receptor antagonist (IL-1RA) levels were elevated in both severe and mild cases and remained at a high level during the 4 weeks of follow-up.
12. Most cytokines observed in previous publications of "cytokine storms" in association with disease severity (9, 10, 14) were observed only in the late stage of severe cases, mostly at 4 weeks after onset of symptom — for example, IL-6, IL-12, IL-1β, IFN-γ, IL-17, and IL-27.
13. In the first week, RANTES in the mild group (638.62 ± 174.81 pg/mL) was much higher than that in healthy controls (358.36 ± 123.44 pg/mL, P = 1.0 × 10–6) and remained high in mild cases during their recovery phase (630.57 ± 171.00 pg/mL in week 3 and 654.14 ± 162.86 pg/mL in week 4).
14. No elevation of RANTES was observed in the severe group during the disease progression, suggesting that RANTES may play an important role in protecting COVID-19 patients from developing severe illness (Supplemental Table 1 and Figure 5A).
15. We found that IL-6 (P<0.0001), IL-8 (P<0.0001) and TNF-α (P<0.0001) were significantly elevated in COVID-19 serum compared to healthy donor serum or plasma isolated from CAR T cell-treated patients with no CRS (Fig. 1).
16. In line with previous reports, IL-1β levels were mostly low or at the limit of detection of 0.1 pg ml^{-1}, even though the assay was able to detect various levels of recombinant control cytokines (Extended Data Fig. 1b).

3. Top-3 and Bottom-3 sentences with respect to mean cosine similarity on test set with $n_x = 2, n_y = 2, n_z = 18$

- Qwen: 3, 6, 16 (top) 2, 4, 7 (bottom)
- DeepSeek: 3, 11, 16 (top) 2, 7, 10 (bottom)
- GPT-4o: 3, 6, 16 (top) 2, 5, 7 (bottom)

C STL Correctness Examples

Examples of semantically correct and incorrect STL candidates from $n_x = 2, n_y = 2, n_z = 18$ experiment on GPT-4o model. Sentences are from test set, specifically 1 [36], 2 [11], 3 [15], 4 [32], 5 [28], 6 [15].

1. Natural language sentence: "Following day 10, IL-6 remains increased whereas IFN-α tapered."
   Semantically correct STL candidate:
   $G[10, \infty](IL6(t) > c_{\text{mid}}) \wedge F[10, \infty]G(\dot{IFN\alpha}(t) < 0)$
   Semantically incorrect STL candidate:
   $G[11, \infty](IL6(t) > c_{\text{mid}}) \wedge G[11, \infty](\dot{IFN\alpha}(t) < \dot{c}_{\text{low}})$
2. Natural language sentence: "In line with previous reports, IL-1β levels were mostly low or at the limit of detection of 0.1 pg ml^{-1}, even though the assay was able to detect various levels of recombinant control cytokines (Extended Data Fig. 1b)."
   Semantically correct STL candidate: $G[0, \infty](IL1\beta(t) < c_{\text{low}})$
   Semantically incorrect STL candidate: N/A
3. Natural language sentence: "Circulating IL-1α also was not detected (fig. S9F)."
   Semantically correct STL candidate: $IL1\alpha(t) < c_{\text{low}}$
   Semantically incorrect STL candidate: $IL1\beta(t) = c_{\text{low}}$
4. Natural language sentence: "In parallel, stimulation with CpG 2216 also resulted in lower, but clearly detectable, amounts of IFNs."
   Semantically correct STL candidate:
   $CpG2216(t) > c_{\text{low}} \implies IFN\alpha(t) > c_{\text{low}} \wedge IFN\beta(t) > c_{\text{low}}$
   Semantically incorrect STL candidate: $G[0, \infty](IFN\alpha(t) > c_{\text{low}} \wedge IFN\alpha(t) < c_{\text{mid}})$
5. Natural language sentence: "This is the reason that seroconversion (undetectable stage to production of IgM followed by IgG) in 100% of infected people (with positive virus-specific IgG) is achieved 17–19 days after commencement of indications [7]."
   Semantically correct STL candidate: $F[17, 19](IgM(t) > c_{\text{high}} \wedge IgG(t) > c_{\text{high}})$
   Semantically incorrect STL candidate: $G[17, 19](SARSCoV2(t) = c_{\text{high}})$
6. Natural language sentence: "Monocyte chemotactic factor chemokine(C-C motif) ligand 2 (CCL2) was increased in the blood of infected patients as well as the transcripts of its receptor CCR2; this was associated with low counts of circulating inflammatory monocytes (Fig. 4I), suggesting a rolefor the CCL2/CCR2 axis in the monocyte chemo-attraction into the inflamed lungs."
   Semantically correct STL candidate: $CCL2(t) > c_{\text{mid}} \wedge CCR2(t) > c_{\text{mid}}$
   Semantically incorrect STL candidate:
   $CCL2(t) = c_{\text{high}} \wedge CCR2(t) = c_{\text{high}} \implies IFN\gamma(t) < c_{\text{low}}$

D Semantic Evaluation Histograms

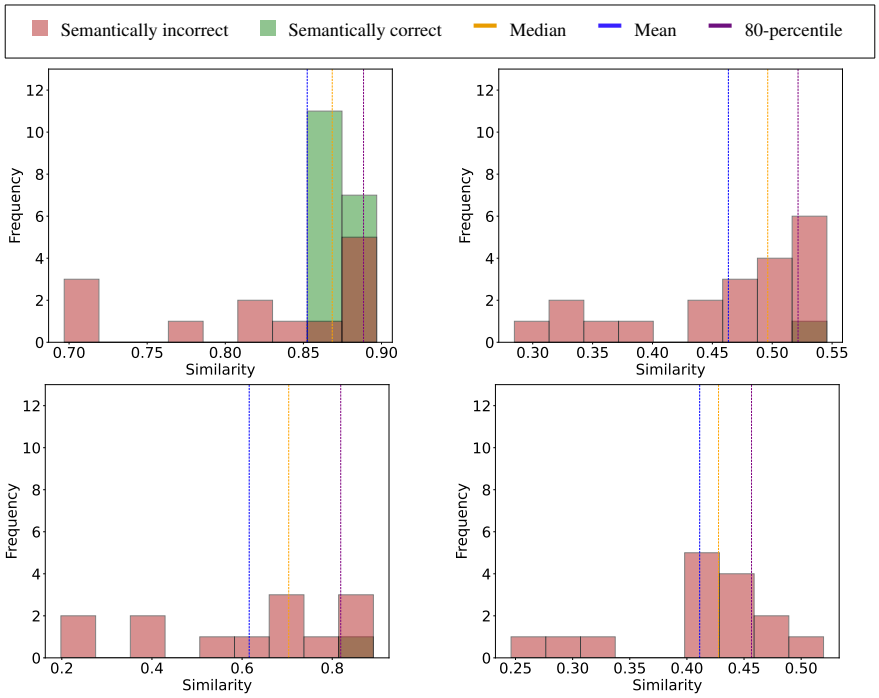

Figure 4: Histograms of sentence embedding cosine similarity for simpler sentence (left, Appendix B.1.3) and more ambiguous sentence (right, Appendix B.1.7) for Qwen (top) and DeepSeek (bottom) with $n_x = 3, n_y = 1, n_z = 18$.

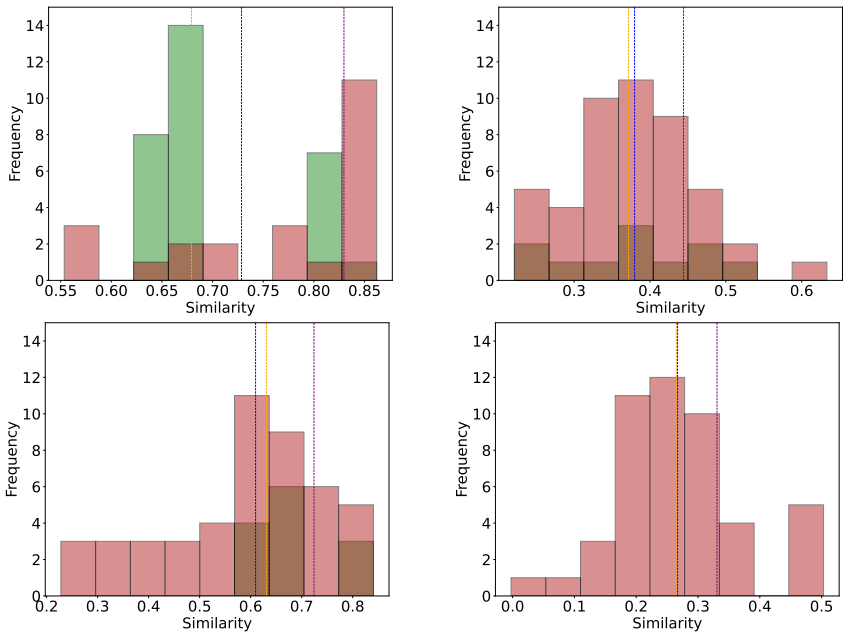

Figure 5: Histograms of sentence embedding cosine similarity for three test sentences with highest (left) and lowest (right) median cosine similarity for Qwen (top) and DeepSeek (bottom) with $n_x = 3, n_y = 1, n_z = 18$.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We list our contributions in Sec. 1, which are supported by the presentation of the method in Sec. 2 and the numerical evaluation with three different LLMs in Sec. 3.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations in Sec. 4.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide the implementation details in Sec. 3 and the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We will provide the code with an open license if the paper is accepted.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide the implementation details in Sec. 3 and the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We do not claim or report statistical significance as the amount of validation and test sentences is too small.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the resource details in Sec. 3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We conformed with the Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the societal impact in Sec. 4.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We do not provide such assets.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We provide references to software tools and the literature we build on.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We describe the usage of LLMs since they are central for our approach in Sec. 2 and 3.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.