LSEG: A Fine-tuning Free Method for NL2FOL via Logic-Structure and Entropy Guided Inference Controlling

Yuwei Huang[1,2], **Shi Wang**[1,2,*], **Kangli Zi**[1,2†], **Tianyu Luo**[1,2], **Zhixiao Jiang**[1,2], **Ruiheng Wang**[1,2], **Yufei Wang**[1,2]
[1]State Key Laboratory of AI Safety, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
[2]University of Chinese Academy of Sciences, Beijing 100049, China
{huangyuwei23s, wangshi, zikangli, luotianyu23s, jiangzhixiao24s, wangruiheng25s, wangyufei25s}@ict.ac.cn

Abstract

Large language models have shown strong generative and reasoning capabilities, yet they still struggle with natural language to first order logic (NL2FOL) translation due to logical hallucination. We propose LSEG (Logic Structure and Entropy Guided), a fine-tuning free framework designed to improve logical consistency during inference. The core idea of LSEG is to correct hidden state deviation by leveraging logical stability across logic preserving perturbations of the input. Such deviation is especially harmful in NL2FOL, as even small drifts can flip quantifier scope or logical operators, producing formulas that are syntactically valid yet logically incorrect. First, LSEG constructs perturbation-averaged direction vectors that approximate a stable logical center. Second, it derives layer-wise correction directions by contrasting original and perturbed representations. Lastly, LSEG uses an entropy-guided adaptive mechanism to inject these directions only when the model exhibits unstable or overconfident reasoning states, thereby preserving fluency while correcting logical drift. Experiments on the FOLIO and MALLS benchmarks show that LSEG consistently improves logical equivalence scores over strong baselines, despite requiring no training or parameter updates. Further evaluation on LogicLLaMA demonstrates LSEG's architecture-agnostic effectiveness.

1 Introduction

Large language models (LLMs) have achieved human-competitive performance in natural language generation, code understanding, and reasoning tasks (Brown et al., 2020; Ouyang et al., 2022; Wei et al., 2022). Despite these advances, maintaining precise semantic alignment and formal logical consistency in natural language to first-order logic (NL2FOL) remains a fundamental challenge. In this task, LLMs often generate formulas that are syntactically valid yet logically inconsistent, or they exhibit semantic drift when translating complex statements (Pan et al., 2023; Xu et al., 2024). Figure 1 illustrates a representative case in which subtle representational deviations lead to erroneous logical forms.

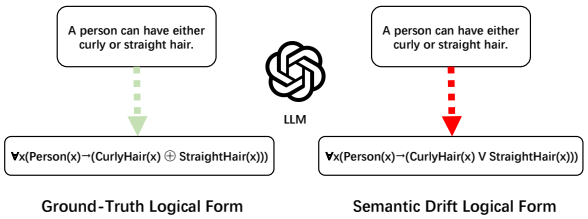

Figure 1: An example of semantic drift in NL2FOL translation. The LLM-generated logical form deviates from the correct logical intent, illustrating a typical case of inconsistency.

NL2FOL, which aims to map natural language statements to first-order logic, serves as a bridge between linguistic understanding and symbolic reasoning (Abzianidze, 2017; Bos and Markert, 2005). Traditional rule-based and grammar-driven methods (Zettlemoyer and Collins, 2012; Cao et al., 2019) provided accurate mappings in constrained domains but suffered from limited scalability. Neural semantic parsers later framed NL2FOL as a sequence-to-sequence translation problem (Lu et al., 2022), enabling end-to-end automation while still frequently producing syntactically correct yet semantically misaligned formulas due to the absence of explicit logical constraints. With the emergence of LLMs, recent work has explored few-shot prompting, supervised fine-tuning, and reinforcement learning to improve logical translation (Han et al., 2024; Tian et al., 2021; Yang et al., 2024; Liu, 2025).

However, even these enhanced LLM-based approaches struggle to maintain logical consistency,

* Corresponding author.
† Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 24236–24245
July 2-7, 2026

especially for sentences requiring fine-grained semantic composition or complex quantifier interactions. This raises a fundamental question: why do LLMs fail at NL2FOL even when given strong supervision?

Recent findings suggest that the source of such errors lies not only in decoding procedures, but also in representation drift the gradual divergence of internal hidden states from the intended logical structure during forward computation(Pan et al., 2023; Xu et al., 2024; Turner et al., 2023; Zou et al., 2023). Because NL2FOL is highly sensitive to quantifier scope and operator attachment, even minor representational deviations can alter logical relations, resulting in inconsistent formulas. These observations motivate a shift from external optimization toward representation level correction, where internal activations are guided to remain logically aligned throughout generation.

To address this challenge, we propose LSEG (Logic-Structure and Entropy Guided), a fine-tuning free and plug-and-play framework for improving logical consistency in NL2FOL translation. LSEG first constructs a stable logical anchor from logic preserving variants of the input and derives layer-wise correction directions by contrasting original and perturbed representations. Principal component analysis is then applied to extract robust intervention vectors that generalize across examples. During inference, an entropy-guided adaptive mechanism selectively injects these correction vectors only when the model exhibits abrupt entropy drops—signals of over-confident or unstable reasoning—thus preserving fluency while correcting internal drift.

Compared with existing supervised or fine-tuned approaches, LSEG requires no parameter updates, introduces minimal computational overhead, and generalizes across architectures. It provides an interpretable means of stabilizing hidden-state dynamics, achieving a better balance among logical consistency, semantic stability, and reasoning transparency.

The main contributions of this work are summarized as follows:

- We introduce LSEG, a fine-tuning free, direction-vector-based framework that corrects internal representation drift and improves logical consistency without any fine-tuning.
- We propose an entropy-guided adaptive mechanism that dynamically adjusts intervention strength, preventing over-correction and preserving generation fluency.
- We demonstrate significant improvements on FOLIO and MALLS benchmarks and show that LSEG generalizes to LogicLLaMA models, confirming its robustness, interpretability, and architecture-agnostic applicability.

2 Related Work

Natural Language to First-Order Logic.

The task of NL2FOL aims to convert natural language statements into formal logical expressions, serving as a bridge between linguistic understanding and symbolic reasoning (Abzianidze, 2017; Bos and Markert, 2005). Early approaches relied on rule-based and grammar-driven methods, such as CCG and lambda calculus, which explicitly defined mappings between linguistic structures and logical forms to achieve precise semantic composition (Zettlemoyer and Collins, 2012; Cao et al., 2019). However, these methods heavily depended on manually crafted rules and exhibited limited scalability. Later, neural semantic parsers framed NL2FOL translation as a sequence-to-sequence task (Lu et al., 2022), achieving end-to-end automation but often producing syntactically correct yet semantically inconsistent formulas due to the lack of explicit logical constraints.

As NL2FOL tasks expanded beyond narrow domains, evaluating logical consistency became increasingly challenging. Early benchmarks lacked scale and structural diversity, making it difficult to measure whether models genuinely understood formal semantics. To enable systematic and comprehensive evaluation in open-domain settings, recent datasets such as FOLIO (Han et al., 2024) and MALLS (Yang et al., 2024) were introduced. With the rapid development of large language models (LLMs), researchers began leveraging their strong linguistic generalization and few-shot learning capabilities to improve logical translation. Representative models, including LogicLLaMA (Yang et al., 2024) and Code4Logic (Liu, 2025), combined reinforcement learning and logic-aware fine-tuning to enhance formal reasoning accuracy. Nevertheless, these methods still rely on extensive annotation and high computational cost, and logical inconsistency remains a persistent issue. Recent studies have further shown that logical errors often arise from internal representation drift rather than decoding-

level mistakes (Pan et al., 2023; Xu et al., 2024; Yang et al., 2026), motivating a shift from external optimization toward internal representation guidance.

Representation Steering and Logical Consistency Enhancement

Representation steering has recently emerged as a promising approach for improving model interpretability and consistency. Studies have demonstrated that hidden representations of LLMs encode rich semantic dimensions that can be modulated via steering vectors to enhance or suppress specific attributes (Ilharco et al., 2022; Turner et al., 2023; Meng et al., 2022; Zou et al., 2023; Liu et al., 2024). These methods typically manipulate intermediate activations through linear interpolation or directional adjustments, enabling semantic control without retraining(Xue et al., 2022). At the same time, entropy-guided mechanisms have been introduced to dynamically evaluate model uncertainty by monitoring token-level entropy fluctuations, allowing adaptive correction when instability is detected (Das et al., 2025). Together, these works provide a theoretical foundation for fine-tuning free interventions that adjust internal representations to enhance logical consistency. Building upon these insights, our LSEG framework leverages perturbation-averaged direction vectors to capture semantic stability centers and integrates entropy-guided adaptive decoding for lightweight, interpretable, and architecture-agnostic logical correction.

3 Methodology

3.1 Problem Definition

The goal of NL2FOL is to map a natural language sentence x to a logically equivalent first order expression y. We formalize this transformation as a conditional generation problem:

$$y = \arg\max_{y'} p_\theta(y' \mid x), \tag{1}$$

where θ denotes model parameters. Let $h^{(l)}(x)$ denote the hidden representation at layer l.

Logical hallucination arises when $h^{(l)}(x)$ drifts away from an internal state that preserves the intended logical structure of the input, such as quantifier scope and operator attachment. In NL2FOL, even small representational deviations may alter quantifier scope or logical relations and lead to inconsistent formulas. For example, a slight shift in internal logical structure can incorrectly swap quantifier scope in sentences such as "Every student read a book", changing the meaning from "each student read some (possibly different) book" to "there exists one book that every student read". Recent studies suggest that such errors often originate from representation drift during forward computation rather than decoding mistakes, causing hidden states to gradually misalign with the intended logical structure.

3.2 Framework

To address the challenges above, we propose LSEG(Logic-Structure and Entropy Guided), a fine-tuning free and plug and play framework equipped with entropy guided adaptive decoding. LSEG consists of two complementary stages: (1) construction of logical correction directions, and (2) entropy guided inference control. Figure 2 provides an overview of the pipeline.

The observations above indicate that the core difficulty is not surface level generation, but maintaining a stable logical anchor within the model's internal representations. Accordingly, LSEG introduces an explicit logical reference so that hidden states remain aligned with the intended logical structure during forward computation. With this reference, corrective perturbations can be applied directly to internal activations to counteract logical displacement and stabilize reasoning.

This motivates a representation level solution: instead of retraining the model or imposing additional decoding constraints, we directly correct hidden state drift by realigning internal activations toward a logic-aligned region at inference time. Our objective is therefore to estimate and apply corrective perturbations that reduce logical inconsistency during inference without modifying model parameters.

Concretely, LSEG operates in two stages:

1. **Direction Vector Construction:** building logical correction directions using perturbation based representation contrast.

2. **Entropy Guided Inference Control:** selectively injecting these directions based on entropy signals that indicate unstable reasoning states.

This design ensures that correction signals are derived from logic preserving perturbations and that interventions occur only when the model becomes internally unstable.

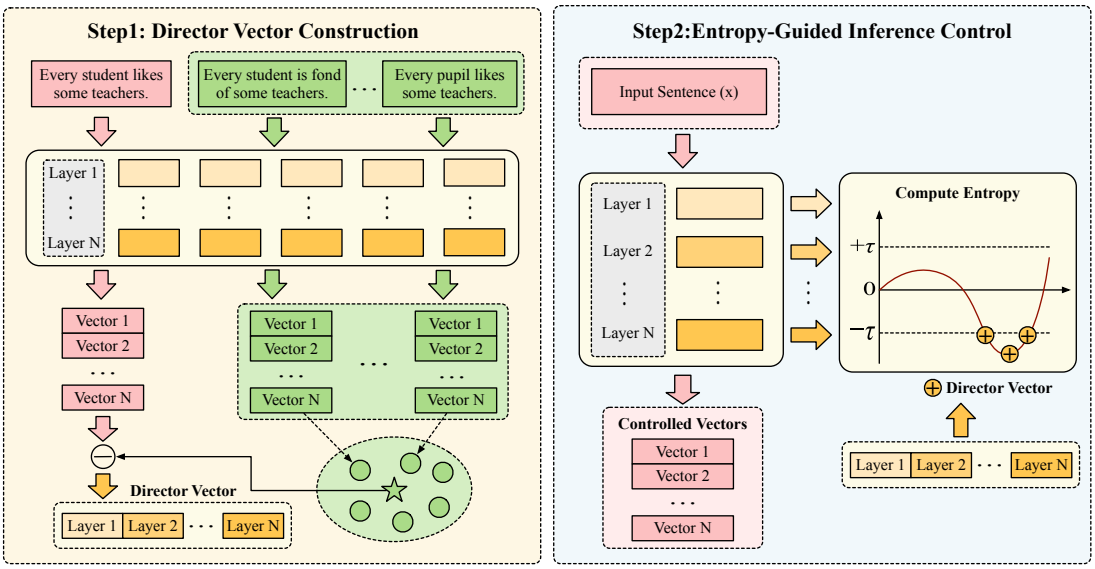

Figure 2: Overall framework of the proposed LSEG method. (a) Direction Vector Construction derives logical correction directions from logic preserving variants. (b) Entropy Guided Inference dynamically injects these directions based on entropy change.

3.3 Logical Direction Construction

Given an input sentence x, we construct a set of n logically equivalent variants $\{x'_1, x'_2, \ldots, x'_n\}$ using lightweight logic-preserving perturbations, such as synonym substitution or minor syntactic reshaping. Importantly, these perturbations are designed to keep the target logical form unchanged under the NL2FOL schema, even if surface wording differs. In practice, we generate perturbations from a small curated calibration set: we manually select 50 sentences whose logical structures are relatively complex or information-rich (e.g., involving multiple quantifiers, negation, implication or conditionals, and multi-clause compositions), and rewrite each sentence into n surface-diverse variants using a large language model with explicit constraints that the logical structure must remain unchanged, including quantifier scope, operator attachment, and predicate–argument relations. We then perform a lightweight manual check and discard any rewrite that introduces logical ambiguity or changes the intended logical form. For example, for "Every student read a book", acceptable rewrites include "Each student has read at least one book" or "For every student, there exists a book that the student read", while "There is a book that every student read" is rejected because it changes the quantifier scope.

For each variant x'_i, the model produces hidden representations across L layers: $\{h^{(1)}(x'_i), \ldots, h^{(L)}(x'_i)\}$. At each layer l, we compute the logic-anchored mean representation as:

$$h^{(l)}_{\text{logic}} = \frac{1}{n} \sum_{i=1}^{n} h^{(l)}(x'_i). \tag{2}$$

The original representation is:

$$h^{(l)}_{\text{orig}} = h^{(l)}(x). \tag{3}$$

We then define the logic displacement vector at layer l as:

$$\Delta^{(l)} = h^{(l)}_{\text{logic}} - h^{(l)}_{\text{orig}}. \tag{4}$$

Intuitively, $\Delta^{(l)}$ captures how the representation of the original input should shift to better align with a logic-stable internal state implied by logically equivalent variants. Averaging across logic-preserving perturbations reduces lexical and syntactic noise while retaining invariant logical structure, producing a robust direction for correcting logical drift.

3.4 Layer-wise Intervention Vector Extraction

To obtain correction directions that generalize across examples, we collect displacement vectors $\{\Delta^{(l)}_1, \ldots, \Delta^{(l)}_N\}$ from training samples and perform principal component analysis at each layer:

$$v^{(l)} = \text{PCA}_1(\{\Delta^{(l)}_n\}^N_{n=1}), \tag{5}$$

where PCA_1 denotes the first principal component.

The vector $v^{(l)}$ captures the dominant pattern of logical deviation at layer l, providing a statistically stable correction direction. This is preferred over random perturbations or gradients because PCA isolates consistent displacement trends across samples. Empirically, mid-to-upper transformer layers show the strongest drift, and thus interventions focus primarily on these layers.

3.5 Entropy-Guided Intervention Mechanism

During inference, LSEG regulates when and where to apply correction vectors using token-level entropy as an internal stability signal. At decoding step t and layer l, we first obtain a layer-specific token distribution by projecting the hidden state $h_t^{(l)}$ to vocabulary logits using the shared output projection (i.e., the same unembedding matrix as the final LM head) and applying softmax:

$$p_t^{(l)} = \text{softmax}(W_{\text{out}} h_t^{(l)}), \tag{6}$$

where W_{out} denotes the output projection matrix. We then compute the layer-wise entropy as:

$$H_t^{(l)} = -\sum_k p_{t,k}^{(l)} \log p_{t,k}^{(l)}. \tag{7}$$

Entropy change between adjacent layers is defined as:

$$\Delta H_t^{(l)} = H_t^{(l)} - H_t^{(l-1)}, \tag{8}$$

and its standardized variation is:

$$z_t^{(l)} = \frac{\Delta H_t^{(l)} - \mu_{\Delta H}}{\sigma_{\Delta H}}, \tag{9}$$

where $\mu_{\Delta H}$ and $\sigma_{\Delta H}$ are the mean and standard deviation of ΔH used for normalization.

We treat layers with large-magnitude entropy fluctuations as unstable:

$$\mathcal{L}_t = \{l \mid |z_t^{(l)}| > \tau\}. \tag{10}$$

For each $l \in \mathcal{L}_t$, we compute the intervention strength as:

$$\alpha_l = \alpha_{\max} \frac{|\Delta H_t^{(l)}|}{\max_{j \in \mathcal{L}_t} |\Delta H_t^{(j)}|}. \tag{11}$$

The updated hidden state is:

$$h_t'^{(l)} = h_t^{(l)} + \alpha_l v^{(l)}. \tag{12}$$

Entropy thus serves as an internal diagnostic signal: when reasoning is stable, layer-wise entropy variations diminish and the intervention strength naturally decays toward zero.

Algorithm 1 LSEG: Logic Structure and Entropy Guided Inference for NL2FOL

Input: Language model f_θ, dataset $\{x_i\}_{i=1}^N$
Generate perturbations $\{x_i'\}_{i=1}^n$ for each x_i
Compute layer wise displacements $\Delta^{(l)}$
Perform PCA to obtain intervention vectors $v^{(l)}$
for each decoding step t **do**
    Compute $H_t^{(l)}$ and $z_t^{(l)}$ for all layers
    Identify unstable layers $\mathcal{L}_t = \{l \mid |z_t^{(l)}| > \tau\}$
    for $l \in \mathcal{L}_t$ **do**
        Compute α_l
        Update $h_t^{(l)} \leftarrow h_t^{(l)} + \alpha_l v^{(l)}$
    end for
end for

3.6 Algorithm

Algorithm 1 summarizes the full LSEG procedure. The direction vectors $\{v^{(l)}\}$ are estimated offline for each target model using a small calibration set and reused across inputs without parameter updates. During inference, entropy dynamics determine when and where to inject the learned directions, allowing LSEG to operate as a lightweight and architecture-agnostic plug-in.

4 Experiments

4.1 Comparison with State-of-the-Arts

Datasets and Evaluation. We evaluate LSEG on two standard benchmarks for NL2FOL: (1) FOLIO (Han et al., 2024), a large-scale human-annotated dataset covering quantifiers, negation, implication, and conditional structures; and (2) MALLS (Yang et al., 2024), a synthetic dataset generated by GPT-4 that includes deeper compositional semantics and long-distance dependencies. The evaluation metric is Logical Equivalence (LE), which measures whether the generated and reference logical forms are logically equivalent after normalization.

LSEG is evaluated under fixed decoding settings in our implementation, and it is applied purely during inference without any fine-tuning or parameter updates. In this paper, LSEG is implemented on the Qwen-3 8B model (Yang et al., 2025).

Compared Methods. We compare LSEG against recent state-of-the-art (SOTA) logical translation systems, which can be categorized into two groups: (1) supervised models such as LogicLLaMA (Yang

Model	FOLIO LE (%)	MALLS LE (%)	Training Cost
Flan-T5 (Chung et al., 2024)	70.67	68.45	Instruction Tuning
Claude-1 (Perez et al., 2023)	74.47	77.46	API
GPT-4 (Achiam et al., 2023)	85.53	84.38	API
LogicLLaMA (Yang et al., 2024)	84.90	81.34	Supervised + RLHF
Symbol-LLM-13B (Xu et al., 2024)	91.59	89.41	Supervised
Code4Logic (CodeGeeX) (Liu, 2025)	84.77	85.81	Few-shot
Code4Logic (GPT-3.5-Turbo-16k) (Liu, 2025)	**92.67**	90.92	Few-shot (API)
Qwen-3 8B (baseline, our setup)	88.00	89.70	–
LSEG (Ours, Qwen-3 8B) (Yang et al., 2025)	92.01	**91.34**	Fine-tuning Free

Table 1: Logical Equivalence (LE) results on FOLIO and MALLS. Scores for prior baselines are taken from Code4Logic (Liu, 2025) (and the original cited papers) for reference. The Qwen-3 8B baseline and LSEG are evaluated under the same prompt, decoding, and evaluation pipeline in our implementation.

et al., 2024) and Symbol-LLM (Xu et al., 2024), which incorporate explicit logical supervision or symbolic constraints; and (2) few-shot code generation methods, represented by Code4Logic (Liu, 2025), which reformulate NL2FOL translation as Python code generation and evaluate multiple LLM backbones. In addition, we include instruction-tuned general-purpose models such as Flan-T5 (Chung et al., 2024), Claude-1 (Perez et al., 2023), and GPT-4 (Achiam et al., 2023) as baseline references.

Results and Analysis. Table 1 reports the Logical Equivalence (LE) scores on FOLIO and MALLS. Existing state-of-the-art models mainly achieve performance gains through supervised fine-tuning or code-based reasoning. While these approaches improve logical consistency, they require additional training resources or costly API access and often depend on specific architectures. In contrast, LSEG requires no training and operates purely during inference by applying perturbation-averaged direction vectors with entropy-guided adaptive control. This lightweight and model-agnostic framework steers hidden representations toward logic-aligned regions, effectively mitigating logical hallucinations without compromising fluency. The results demonstrate that LSEG, implemented on Qwen-3 8B, achieves LE scores comparable to or exceeding those of existing SOTA methods under fine-tuning free conditions, highlighting its efficiency, transferability, and general applicability.

4.2 Cross-Model Validation and Effectiveness

Objective. To further assess the generality of LSEG beyond the Qwen-3 8B backbone used in Table 1, we apply it to LogicLLaMA (Yang et al., 2024), a logic-supervised model built on the LLaMA-2 architecture with 7B and 13B checkpoints. Since LogicLLaMA is trained with supervised data and RLHF objectives for NL2FOL, this experiment tests whether an inference-time, training-free intervention can still provide additional gains under a different architecture and training paradigm.

Setup. We directly apply LSEG to the released LogicLLaMA-7B and LogicLLaMA-13B checkpoints without any further training or parameter updates. We evaluate on the same FOLIO and MALLS benchmarks as in Table 1, using Logical Equivalence (LE) as the evaluation metric. Unless otherwise noted, we follow the official inference and evaluation pipeline provided by LogicLLaMA for these checkpoints. During decoding, LSEG injects layer-wise intervention vectors and modulates the intervention strength based on layer-wise entropy variations.

Results. Table 2 reports the LE scores before and after applying LSEG. LSEG yields consistent improvements for both model sizes. The absolute gains are around one point, suggesting that LSEG provides complementary benefits even for logic-tuned models.

Model	**FOLIO LE (%)**	**MALLS LE (%)**
LogicLLaMA (7B)	84.12	81.98
LogicLLaMA + LSEG (7B)	**85.24**	**83.06**
LogicLLaMA (13B)	85.81	82.37
LogicLLaMA + LSEG (13B)	**86.90**	**83.48**

Table 2: Logical Equivalence (LE) results of LogicLLaMA before and after applying LSEG on FOLIO and MALLS. All results are obtained using the official evaluation scripts. LSEG operates entirely during inference without any fine-tuning or additional training data.

Analysis. These results indicate that LSEG consistently improves logical equivalence across model scales and across a different backbone family. Intuitively, the perturbation-anchored intervention vectors help correct subtle representation drift in logic-relevant layers, while the entropy-guided gating reduces the risk of over-correction by activating interventions only when internal confidence shifts sharply. Overall, LSEG behaves as a plug-and-play inference-time module that can complement both general-purpose LLMs and logic-supervised models.

4.3 Ablation Study

To better understand the contribution of each component in LSEG, we conduct ablation experiments focusing on two major design choices: (1) the entropy-guided adaptive scheduling mechanism, and (2) the transformer layers at which intervention is applied. All experiments are performed on Qwen-3 8B using the FOLIO dataset, and we report Logical Equivalence (LE) as the evaluation metric.

Entropy-Guided Adaptive Scheduling. We compare LSEG with and without the entropy-guided activation mechanism. The fixed-strength variant injects a constant perturbation throughout decoding, while the full LSEG activates intervention only when a significant entropy drop is detected. As shown in Table 3, removing entropy guidance causes a noticeable reduction in LE on both datasets, indicating that uniform-strength intervention tends to over-correct high-confidence states. Entropy-guided LSEG maintains fluency while providing more stable correction.

Variant	FOLIO LE (%)	MALLS LE (%)
w/o Entropy Guidance (fixed α)	90.42	90.11
LSEG (fixed schedule)	91.02	90.83
LSEG (entropy-guided, full)	**92.01**	**91.34**

Table 3: Ablation on entropy-guided adaptive scheduling. Entropy-based triggering provides the most stable and consistent improvement.

Layer-Wise Injection Analysis. To understand where LSEG is most effective, we partition the transformer into non-overlapping depth intervals of four layers and apply LSEG only within a single interval in each run. Table 4 reports Logical Equivalence (LE) on FOLIO and MALLS, and Figure 3 overlays both datasets in a single plot for easier comparison. We observe that injecting in shallow layers typically degrades performance (e.g., L0–3), which is consistent with the intuition

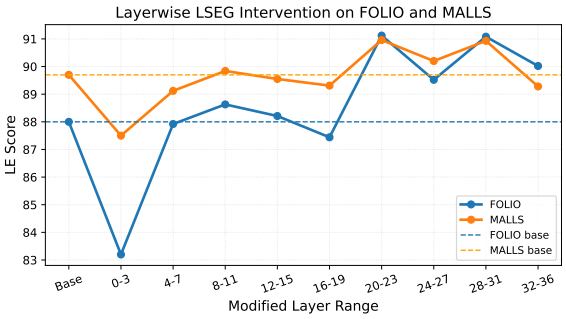

Figure 3: Layer-wise LSEG intervention on FOLIO and MALLS (combined). Shallow-layer injection reduces LE, while mid-to-upper layers yield the largest gains; entropy-guided full LSEG achieves the best performance on both benchmarks.

that early layers mainly encode lexical and local syntactic features and are more sensitive to representation perturbations. In contrast, the strongest single-interval gains emerge in mid-to-upper layers, with L20–23 and L28–31 yielding substantial improvements on both benchmarks, suggesting that logic-relevant information is consolidated later in the network and that these layers are also where larger entropy fluctuations tend to occur during decoding. Finally, enabling entropy-guided LSEG across all layers achieves the best overall results, indicating that combining layer-wise intervention with adaptive triggering produces more stable and effective corrections.

Layer Range	FOLIO LE (%)	MALLS LE (%)
Base (no LSEG)	88.00	89.70
L0–L3	83.20	87.50
L4–L7	87.92	89.12
L8–L11	88.63	89.84
L12–L15	88.21	89.55
L16–L19	87.44	89.31
L20–L23	91.12	90.96
L24–L27	89.52	90.20
L28–L31	91.08	90.93
L32–L36	90.02	89.28
Full LSEG (entropy-guided)	**92.01**	**91.34**

Table 4: Layer-wise ablation of LSEG. The strongest gains appear in mid-to-upper layers (L20–L31), and full entropy-guided LSEG achieves the final best performance.

4.4 Layer Entropy Profiling

To better understand where uncertainty changes arise during NL2FOL decoding, we profile token entropy across transformer layers. For each decoding step t and layer l, we obtain a layer specific token distribution by projecting the hidden state $h_t^{(l)}$ to vocabulary logits with the shared

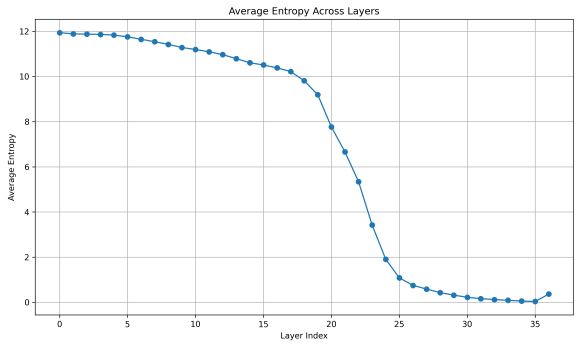

Figure 4: Average token entropy across transformer layers, computed from layer specific token distributions and averaged over generated tokens on 30 randomly sampled inputs. Entropy collapses in mid to upper layers, consistent with the layers where LSEG interventions are most effective.

output projection and applying softmax, $p_t^{(l)} = \mathrm{softmax}(W_{\text{out}} h_t^{(l)})$. We then compute layer entropy as $H_t^{(l)} = -\sum_k p_{t,k}^{(l)} \log p_{t,k}^{(l)}$. We randomly sample 30 inputs and average $H_t^{(l)}$ over generated tokens only (excluding the prompt). All runs use greedy decoding .

Figure 4 shows that entropy remains relatively high and changes slowly in early layers, but exhibits a pronounced collapse in the mid to upper layers and stays low thereafter. This depth region aligns with the layers that yield the largest gains in our layer range ablation (Table 4), suggesting that logic relevant decisions become consolidated deeper in the network and that large entropy shifts concentrate in those layers. Overall, this diagnostic analysis supports LSEG's design choice to use layer wise entropy fluctuations as an internal instability signal and to prioritize interventions in mid to upper transformer layers.

5 Conclusion

This work introduces LSEG, a lightweight and fine-tuning free framework for improving logical consistency in natural language to first-order logic translation. LSEG constructs logical correction directions from perturbation-averaged representations and applies them adaptively during inference using entropy-based uncertainty signals. This design allows the method to operate as a plug-and-play module that can be inserted into existing large language models without parameter updates or additional supervision.

Experiments on FOLIO and MALLS demonstrate that LSEG substantially enhances logical equivalence while maintaining fluent generation, achieving performance comparable to or exceeding state-of-the-art systems despite requiring no training. Additional evaluation on LogicLLaMA further shows that LSEG generalizes across different model architectures, serving as a broadly applicable inference-time correction mechanism. Ablation studies confirm the importance of both entropy-guided scheduling and layer-wise intervention, highlighting that logic-relevant information consolidation occurs primarily in mid-to-upper transformer layers where LSEG is most effective.

Overall, LSEG offers a simple and effective approach for mitigating logical hallucination in NL2FOL translation. Future work includes extending LSEG to other structured reasoning tasks, exploring more fine-grained uncertainty signals, and integrating LSEG with symbolic or code-based reasoning frameworks to further improve reliability in high-stakes applications.

Limitations

LSEG is an inference time intervention approach, and its effectiveness depends on several practical factors.

First, the method relies on logic preserving surface variations to estimate stable correction directions. Although these variations are designed to preserve the underlying logical form, there can be cases where surface changes interact with task specific annotation conventions or parsing schemas, which may reduce the precision of the estimated directions.

Second, LSEG uses layer level uncertainty dynamics to decide when to intervene. While entropy based signals provide a lightweight proxy for internal instability, they are not a perfect indicator of logical correctness. Certain inputs may exhibit stable entropy patterns while still containing subtle semantic errors, and conversely some entropy fluctuations may reflect benign variability rather than genuine logical drift.

Acknowledgements

Shi Wang and Kangli Zi are corresponding authors. This work is supported by the National Key Research and Development Program of China (No. 2024QY210004) and the ICT Innovation Subject (No. E461010).

References

Lasha Abzianidze. 2017. Langpro: Natural language theorem prover. *EMNLP 2017*, page 115.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Johan Bos and Katja Markert. 2005. Recognising textual entailment with logical inference. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 628–635.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Ruisheng Cao, Su Zhu, Chen Liu, Jieyu Li, and Kai Yu. 2019. Semantic parsing with dual learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 51–64, Florence, Italy. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and 1 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Souvik Das, Lifeng Jin, Linfeng Song, Haitao Mi, Baolin Peng, and Dong Yu. 2025. Entropy guided extrapolative decoding to improve factuality in large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6589–6600, Abu Dhabi, UAE. Association for Computational Linguistics.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, Lucy Sun, Alexander Wardle-Solano, Hannah Szabó, Ekaterina Zubova, Matthew Burtell, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, and 16 others. 2024. FOLIO: Natural language reasoning with first-order logic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22017–22031, Miami, Florida, USA. Association for Computational Linguistics.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2022. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*.

Junnan Liu. 2025. Few-shot natural language to first-order logic translation via code generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10939–10960, Albuquerque, New Mexico. Association for Computational Linguistics.

Sheng Liu, Haotian Ye, Lei Xing, and James Zou. 2024. Reducing hallucinations in vision-language models via latent space steering. *Preprint*, arXiv:2410.15778.

Xuantao Lu, Jingping Liu, Zhouhong Gu, Hanwen Tong, Chenhao Xie, Junyang Huang, Yanghua Xiao, and Wenguang Wang. 2022. Parsing natural language into propositional and first-order logic with dual reinforcement learning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5419–5431, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. 2023. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3806–3824, Singapore. Association for Computational Linguistics.

Ethan Perez, Sam Ringer, Kamile Lukosiute, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, and 1 others. 2023. Discovering language model behaviors with model-written evaluations. In *Findings of the association for computational linguistics: ACL 2023*, pages 13387–13434.

Jidong Tian, Yitian Li, Wenqing Chen, Liqiang Xiao, Hao He, and Yaohui Jin. 2021. Diagnosing the first-order logical reasoning ability through LogicNLI. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3738–3747, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. 2023. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou,

and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Fangzhi Xu, Zhiyong Wu, Qiushi Sun, Siyu Ren, Fei Yuan, Shuai Yuan, Qika Lin, Yu Qiao, and Jun Liu. 2024. Symbol-LLM: Towards foundational symbol-centric interface for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13091–13116, Bangkok, Thailand. Association for Computational Linguistics.

Xiaojun Xue, Chunxia Zhang, Zhendong Niu, and Xindong Wu. 2022. Multi-level attention map network for multimodal sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 35(5):5105–5118.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Junqi Yang, Yuecong Min, Jie Zhang, Shiguang Shan, and Xilin Chen. 2026. Infact: A diagnostic benchmark for induced faithfulness and factuality hallucinations in video-llms. *Preprint*, arXiv:2603.11481.

Yuan Yang, Siheng Xiong, Ali Payani, Ehsan Shareghi, and Faramarz Fekri. 2024. Harnessing the power of large language models for natural language to first-order logic translation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6942–6959, Bangkok, Thailand. Association for Computational Linguistics.

Luke S Zettlemoyer and Michael Collins. 2012. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. *arXiv preprint arXiv:1207.1420*.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, and 1 others. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.