AbstractReasoner at SemEval-2026 Task 11: Reducing Content Effects via Knowledge Distillation and Structured Reasoning Prompts

Akash Chowdhury* **Vlad Pavlovich*** **Julius Dunfoy** **Sophia Yang** **Abhiram Borra**

[achowd10, vpavlovi, jdunfoy, syang255, abborra] @ucsc.edu

University of California, Santa Cruz

Abstract

Syllogistic reasoning serves as a critical diagnostic for evaluating whether Large Language Models (LLMs) perform genuine logical inference or rely on semantic shortcuts. SemEval-2026 task 11 explores "content effects"—where model judgments are biased by world knowledge rather than logical form. Recent work has illustrated that LLM optimization techniques have provided substantial performance gains in mitigating content effect. To contribute to this research domain, this paper performs a systematic study of different intervention strategies: zero-shot chain of thought, symbolic representation, activation-steering, and supervised fine-tuning along with prompting optimization during inference. We achieved the best performance with our largest model (Phi-4 14B) by fine-tuning with chain of thought distillation, symbolic abstractions and LLM as optimizer prompting (FT_Optim) evaluated on the held-out split derived from the training data. This approach achieved the highest Combined Smooth Score (CSS) of 31.16. Additionally, Llama 3.1 provided noteworthy performance with 31.01 CSS under the same FT_Optim approach, indicating the performance gain was LLM-agnostic.

1 Introduction

Understanding whether large language models (LLMs) can perform *content-independent* logical reasoning remains an open challenge in natural language understanding research. Prior work (Bertolazzi et al., 2024; Eisape et al., 2024) shows that LLMs often rely on surface-level heuristics or world knowledge priors rather than genuine deductive reasoning, making them susceptible to content effects and biased inference. SemEval-2026 Task 11 (Valentino et al., 2026) directly examines this issue by evaluating how well models can execute syllogistic reasoning across languages and under varying degrees of content distraction.

Syllogistic reasoning involves deriving a logically valid conclusion from two categorical premises: (a) a major premise, (b) a minor premise, and (c) a conclusion statement. For example: *All humans are mortal; All Greeks are humans; therefore, All Greeks are mortal*. This form of structured reasoning is conceptually simple yet diagnostic for distinguishing genuine logical generalization from semantic or associative shortcuts. Task 11 operationalizes this evaluation through four subtasks: (1) syllogistic reasoning in English, (2) syllogistic reasoning with irrelevant premises, (3) multilingual syllogistic reasoning, and (4) multilingual syllogistic reasoning with irrelevant premises. These settings collectively test a model's robustness to linguistic variation and content-based distractions. Performance is measured using Accuracy, Total Content Effect (TCE), and the Combined Smooth Score (CSS). The detailed definitions and formulas for these metrics are provided in the Appendix D.

The task includes a diverse suite of languages-English, German, French, Italian, Dutch, Portuguese, Russian, Chinese, Swahili, Bengali, and Telugu-enabling an investigation into whether language models reason consistently across linguistic structures. Irrelevant premises probe reasoning robustness by testing whether LLMs can ignore distractors and focus on the syllogism's logical structure.

In this work, we explore four complementary approaches to improving content-independent logical reasoning: (i) Zero-shot Chain-of-Thought (ZS CoT) prompting (Kojima et al., 2022); (ii) kNN-based Conditional Activation Steering (K-CAST) (Valentino et al., 2025); (iii) symbolic and abstract representations (including few-shot settings) (Kim et al., 2025; Ranaldi et al., 2025); and (iv) parameter-efficient fine-tuning with QLoRA using CoT distillation (Hinton et al., 2015; Dettmers

*These authors contributed equally.

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 2495–2505

et al., 2023; Hu et al., 2022), where reasoning traces generated by GPT-5.1 serve as the teacher model.

Our goal is to understand how prompting style, reasoning representations, activation-level control, and fine-tuning strategies influence LLM performance on content-independent logical inference. Using Phi 4 14B model (FT_Optim) , our system achieved a Combined Smooth Score (CSS) of 36.82 on the submission dataset (subtask 1), which serves as the official primary ranking metric for SemEval-2026 Task 11 *(32nd place out of 103 teams)*. This performance corresponds to an Accuracy of 89.01% and a Content Effect Bias (TCE) of 3.12, indicating both strong logical correctness and high robustness to plausibility-driven distractions. The low TCE combined with high accuracy demonstrates that our approach effectively promotes content-independent reasoning rather than relying on semantic heuristics.

2 Related Works

Research on logical reasoning in large language models (LLMs) has increasingly focused on syllogistic inference as a diagnostic testbed for content-independent reasoning. Early studies showed that LLMs exhibit *content effects*, where reasoning accuracy varies with semantic plausibility despite identical logical form (Lampinen et al., 2024). Subsequent analyses confirmed that models often rely on heuristic or associative shortcuts rather than strictly deductive structure (Eisape et al., 2024). These findings establish syllogistic reasoning as a controlled setting for probing internal decision mechanisms.

Several works systematically benchmark LLMs on classical syllogisms. Bertolazzi et al. (2024) characterize LLMs as "soft reasoners," highlighting inconsistencies across syllogistic figures, while Seals and Shalin (2024) demonstrate failures when linguistic content conflicts with logical validity. Dataset-driven efforts such as NeurBaroco (Ozeki et al., 2024) and SylloBio-NLI (Wysocka et al., 2025) extend evaluation to bias-sensitive and domain-specific settings.

A complementary line of work examines internal reasoning mechanisms. Kim et al. (2025) identify activation-level "reasoning circuits" associated with logical inference in transformers. Building on this, Valentino et al. (2025) propose activation steering methods to mitigate content bias without weight updates.

Another direction explores symbolic or quasi-symbolic interventions. Ranaldi et al. (2025) introduce quasi-symbolic abstractions to encourage role-based reasoning, while work on faithful Chain-of-Thought (CoT) reasoning shows that unconstrained CoT often produces unfaithful or hallucinated steps (Lyu et al., 2023; Xu et al., 2024). Hybrid symbolic-verification approaches further combine LLM generation with formal checking mechanisms (Quan et al., 2024).

Together, prior research suggests that (i) LLM reasoning is systematically influenced by content and internal activation patterns, and (ii) structural interventions—such as symbolic abstraction or activation steering—can reduce these biases. Our work builds on these insights by jointly evaluating symbolic prompting, reflective prompting, and parameter-efficient symbolic distillation within the unified evaluation framework of SemEval-2026 Task 11.

3 System Overview & Experiment Details

In this section, we describe the models, data splits, prompting setups, and training configurations used to evaluate content-independent syllogistic reasoning on SemEval-2026 Task 11.

3.1 Models

We focus on a set of widely used open-source instruction-tuned LLMs spanning different sizes and architectures. Unless otherwise stated, we use the official instruct variants released by the respective model providers like Llama 3.2 3B Instruct, Llama 3.1 8B Instruct (Touvron et al., 2023), Qwen 2.5 3B Instruct, Qwen 2.5 7B Instruct (Qwen et al., 2025), Phi-4 4B (Phi-4-mini-instruct), and Phi-4 14B (Abdin et al., 2024). These models are used both as zero-shot reasoners and as backbones for activation steering and parameter-efficient fine-tuning. Variation in model size and architecture allows us to test whether the observed reasoning effects generalize across families and scales.

3.2 Dataset and Splits

Our experiments use the English subset (similar to subtask 1) of the official SemEval-2026 Task 11 training dataset. Each instance consists of two premises and a conclusion, while Subtasks 2 and 4 include an additional irrelevant premise to test robustness against distractors. We randomly split the provided training data into 70% training, 15% vali-

dation, and 15% test sets (Train-test-data). Training and validation are used for prompt design and fine-tuning, while the test set is reserved for final evaluation. All text is used as provided, and tokenization is handled by the respective model tokenizers. We evaluate Subtasks 3 and 4 via zero-shot cross-lingual transfer, leveraging the models' pre-trained multilingual capabilities.

3.3 Prompting & Baselines

Our first set of experiments establishes prompting-based baselines using Zero-shot Chain-of-Thought (ZS-CoT) prompting (Kojima et al., 2022). For each model, we use a standardized instruction that (i) explains the task (validity judgment over syllogisms), (ii) asks the model to reason step by step, and (iii) requires an explicit final label (VALID or INVALID). No task-specific examples are provided in the context for the zero-shot setting. We evaluate ZS-CoT, measuring Accuracy, Total Content Effect (TCE), and Combined Smooth Score (CSS). The resulting scores serve both as competitive baselines and as reference points for the impact of activation steering, symbolic prompting, and fine-tuning. Following work framing LLMs as token-space optimizers (Yang et al., 2024), we interpret ZS-CoT as an implicit one-step search over reasoning trajectories without iterative refinement. We further test whether alternative prompts influence this token-space search.

3.4 Activation Steering Setup

To mitigate content effects, we apply kNN-based Conditional Activation Steering (K-CAST) (Valentino et al., 2025) at selected transformer layers during inference. K-CAST operates without modifying model weights or altering prompts, instead constructing a datastore of hidden activations from logically equivalent syllogisms that differ only in surface content, thereby separating representations of logical validity from content plausibility. At inference time, the model's current hidden state is compared to this memory using k-nearest neighbors, and a context-dependent steering vector is injected at a chosen layer to bias the representation toward content-independent reasoning. Because the intervention is conditional and dynamically selected rather than static, it selectively corrects plausibility-driven trajectories while minimally disrupting valid reasoning paths.

3.5 Symbolic Representations

We next investigate whether abstracting away lexical content helps models focus on logical form. We prompt the LLMs to derive symbolic representations in which entities and predicates are mapped to schematic variables (e.g., `All A are B`, `All B are C` $\rightarrow$ `All A are C`) (Kim et al., 2025; Ranaldi et al., 2025). We enforce this via few-shot settings, comparing their performance against zero-shot prompts to quantify the benefits and limitations of abstraction.

3.6 Fine-tuning Protocol

To study whether formal reasoning ability can be transferred through learning, we perform parameter-efficient fine-tuning (Dettmers et al., 2023; Hu et al., 2022) via knowledge distillation (Hinton et al., 2015). Specifically, we fine-tune smaller open-weight language models using reasoning traces generated by GPT-5.1 (OpenAI, 2025) as supervision.

Symbolic Distillation: For each training instance, GPT-5.1 is prompted to produce a structured JSON output consisting of: (i) an explicit set-theoretic abstraction (e.g., mapping terms to A, B, C), (ii) formal logical rewriting using $\forall$, $\exists$, $\rightarrow$, and $\neg$, (iii) a short entailment or countermodel argument, and (iv) a binary validity label. These teacher-generated traces are treated as gold targets for student models. The training corpus is curated from ZS-CoT prompts combined with symbolic abstraction outputs generated by GPT-5.1, ensuring that supervision reflects content-independent reasoning rather than world knowledge heuristics.

Optimization Setup: Parameter-efficient fine-tuning is performed using QLoRA (Dettmers et al., 2023) (4-bit quantization with LoRA adapters (Hu et al., 2022)). Models are trained as causal language models on the prompt and distilled JSON output with prompt-token loss masking. Base parameters are frozen and only low-rank adapters are updated, enabling efficient symbolic-style reasoning transfer with minimal memory.

LLM-as-Optimizer Inference Prompt: At inference time, we additionally evaluate a lightweight reflective trigger (Yang et al., 2024): We refer to this variant as FT_Optim. This intervention does not modify fine-tuned model parameters; instead, it encourages the model to explicitly follow the

structured reasoning trajectory learned during distillation. Empirically(Section 4), this self-reflective prompt often reduces Total Content Effect (TCE) and improves Combined Smooth Score (CSS), suggesting improved adherence to formal reasoning patterns. Model fine-tuning specific hyperparameters are reported in Appendix A.4.

4 Results & Discussions

Results: Across models, parameter-efficient fine-tuning (see Table 1) achieves the strongest overall performance. For Llama 3.1 8B, accuracy improves from 58.33 (ZS-CoT) to 86.11, while TCE decreases from 49.15 to 5.17, increasing CSS to 30.55 (from baseline 11.87). Llama 3.2 3B improves from 57.64 to 78.47 accuracy with a substantial reduction in TCE. Fine-tuned Qwen 2.5 7B achieves the highest overall accuracy (88.19), and fine-tuned Phi-4 14B attains the highest CSS (30.61; TCE = 5.13). Few-shot symbolic prompting improves robustness for Qwen 2.5 7B (TCE = 14.18; CSS = 20.12), though gains are smaller than fine-tuning. Finally, FT_Optim yields the highest CSS observed overall (Phi-4 14B: 31.16; TCE = 4.96).

Results Across Subtasks: When focusing exclusively on robustness as measured by Combined Smooth Score (CSS), fine-tuned models consistently achieve the strongest results. For Subtask 1, the highest CSS is obtained by Phi-4 14B (FT ZS-CoT), reaching 35.96 with very low content effect (TCE = 3.13). For Subtask 3, Llama 3.1 8B (FT ZS-CoT) achieves the overall highest CSS observed across all subtasks (36.42), driven by minimal content sensitivity (TCE = 3.13). In Subtask 4, which involves premise-level reasoning, Phi-4 14B (FT_Optim) attains the best robustness score (16.56), suggesting that inference-time reflective prompting provides additional stability for more complex structural judgments. See appendix D.4 for further details.

Discussions: The results reveal a clear hierarchy among reasoning interventions. Symbolic distillation through parameter-efficient fine-tuning consistently yields the largest improvements in both accuracy and robustness to content effects, suggesting that structured supervision encourages models to rely on formal reasoning patterns rather than surface-level semantic heuristics. In contrast, prompting-only strategies such as ZS-CoT or symbolic exemplars produce more variable gains: while they can partially activate latent reasoning capabilities, they do not reliably suppress content bias across models. The reflective trigger used in *FT_Optim* further stabilizes reasoning behavior but is most effective when combined with fine-tuning. Overall, these findings suggest that mitigating content effects in LLM reasoning requires modifying internal representations through structured supervision, rather than relying solely on inference-time prompting strategies.

5 Ablation Studies

To identify which components improve content-independent reasoning, we conduct ablations along three axes: representation, prompting, and supervision. For representation, we compare natural-language syllogisms with symbolic abstractions (mapping entities to schematic variables such as A, B, C), using the same reasoning template. This tests whether removing lexical content reduces TCE and improves structural consistency. Symbolic abstraction reduces total content effect across most model families and sizes (except Llama 3B & Phi 14B). For prompting, without parameter updates, we evaluate: (i) ZS-CoT, (ii) LLM-Optim prompting (reflective trigger). These settings isolate whether structured exemplars or self-reflection alone can mitigate content effects. LLM-Optim prompting alone doesn't help in reducing content effect but helps in boosting the accuracy. For supervision, we fine-tune models using symbolic reasoning traces generated by larger models. An additional FT_Optim variant applies the reflective prompt at inference time. This separates improvements from training-time symbolic transfer and those from inference-time stabilization.

6 Limitations

While our study compares multiple approaches to content-independent reasoning, several limitations remain. A practical limitation is that some runs did not consistently satisfy the strict output schema required for evaluation in subtasks (2 & 4) with irrelevant-premise annotations. Activation steering showed limited improvements in our experiments. We hypothesize this is due to the small datastore size and the sensitivity of steering vectors to layer selection. Moreover, results are based on syllogistic reasoning as a controlled testbed and may not directly transfer to more complex reasoning tasks. Finally, because LLM generation is inher-

Model	Metric	ZS-CoT	LLM-Optim	FS-Symbolic	K-CAST	FT (ZS)	FT_Optim
Llama 3.2 3B Instruct	Acc	57.64	60.14	46.48	60.42	78.47	**79.17**
	TCE	29.81	27.68	37.69	46.88	15.86	**6.37**
	CSS	13.02	13.81	9.98	12.41	20.52	**26.41**
Llama 3.1 8B Instruct	Acc	58.33	61.97	55.56	63.19	**86.11**	84.03
	TCE	49.15	49.23	28.68	45.83	5.17	**4.53**
	CSS	11.87	12.60	12.65	13.04	30.55	**31.01**
Qwen 2.5 3B Instruct	Acc	65.28	70.14	62.50	77.78	73.91	**75.89**
	TCE	19.86	29.01	25.75	28.99	**11.99**	13.22
	CSS	16.17	15.93	14.58	17.67	20.74	**20.76**
Qwen 2.5 7B Instruct	Acc	74.31	75.00	74.83	79.86	**88.19**	82.52
	TCE	31.20	35.51	14.18	20.59	8.00	**6.99**
	CSS	16.62	16.31	20.12	19.61	**27.59**	26.80
Phi-4 4B Mini-Instruct	Acc	70.83	70.63	58.33	48.61	**81.94**	72.92
	TCE	44.61	50.64	33.93	50.00	**14.25**	20.15
	CSS	14.70	14.28	12.81	9.86	**22.00**	18.00
Phi-4 14B	Acc	83.33	79.86	82.64	50.00	86.11	**86.81**
	TCE	14.89	21.45	16.37	50.00	5.13	**4.96**
	CSS	22.13	19.42	21.44	10.14	30.61	**31.16**

Table 1: Test-set performance for syllogistic validity judgment under different reasoning interventions for Train-test-data. Results are reported using Accuracy (↑), Total Content Effect (TCE, ↓), and Combined Smooth Score (CSS, ↑). Columns correspond to the following strategies: standard Zero-Shot Chain-of-Thought prompting (ZS-CoT); LLM-Optim prompting, which adds a lightweight self-reflection trigger to encourage more deliberate reasoning; Few-Shot Symbolic prompting (FS-Symbolic), where demonstrations show abstraction of natural-language terms into symbolic variables (e.g., A, B, C). K-CAST activation steering(single layer); and parameter-efficient fine-tuning (FT). FT (ZS) denotes parameter-efficient fine-tuning evaluated using standard ZS-CoT decoding at inference time, while FT_Optim further applies the self-reflection trigger prompt during decoding.

ently stochastic, outputs may vary across runs depending on sampling settings and random seeds.

7 Future Work

Future work will extend this analysis to broader classes of logical inference, including propositional reasoning (Evans et al., 2018), relational reasoning (Sinha et al., 2019; Li et al., 2017), and rule-based or theorem-proving settings (Campero et al., 2018; Rocktäschel and Riedel, 2017). Such extensions will help determine whether the benefits of symbolic abstraction, activation steering, and fine-tuning generalize beyond syllogistic logic. Future work will improve constrained decoding and schema-aligned generation to ensure reliable evaluation on subtasks (2 & 4) requiring structured premise-level outputs. In addition, we intend to explore alternative activation steering methods beyond K-CAST and investigate multi-layer steering strategies to determine whether coordinated interventions across transformer blocks can produce more stable and scalable reductions in content bias. Furthermore, future work will examine whether these interventions remain effective and stable when applied to larger-scale models.

8 Conclusion

We investigated content-independent syllogistic reasoning in the SemEval-2026 Task 11 framework across six LLMs, comparing zero-shot, symbolic, activation-steering, and fine-tuning interventions. Our results show that distillation-based fine-tuning using symbolic abstraction reasoning traces provides the most robust gains, and Phi-4 14B achieving the overall best result with 86.81% accuracy and a 4.96 TCE. Combining fine-tuning with a lightweight self-reflection trigger (FT_Optim) further stabilized reasoning trajectories, yielding our highest CSS of 31.16. While ZS-CoT is a viable baseline, symbolic prompting provides the strongest training-free gains for large-scale models. Overall, we identify a clear trade-off: fine-tuning offers the most reliable robustness and correctness, whereas symbolic prompting and activation steering serve as efficient, low-cost alternatives for large and resource-constrained settings respectively.

Acknowledgments

We thank Darian Lee and Judith Clymo for their insightful feedback and suggestions. We thank the University of California, Santa Cruz (UCSC) for providing the computational resources used in this study. We also thank the organizers of SemEval-2026 Task 11 for providing the evaluation framework and datasets.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, and 8 others. 2024. Phi-4 technical report. *Preprint*, arXiv:2412.08905.

Leonardo Bertolazzi, Albert Gatt, and Raffaella Bernardi. 2024. A systematic analysis of large language models as soft reasoners: The case of syllogistic inferences. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13882–13905, Miami, Florida, USA. Association for Computational Linguistics.

Andres Campero, Aldo Pareja, Tim Klinger, Josh Tenenbaum, and Sebastian Riedel. 2018. Logical rule induction and theory learning using neural theorem proving. *Preprint*, arXiv:1809.02193.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems*, volume 36, pages 10088–10115. Curran Associates, Inc.

Tiwalayo Eisape, Michael Tessler, Ishita Dasgupta, Fei Sha, Sjoerd Steenkiste, and Tal Linzen. 2024. A systematic comparison of syllogistic reasoning in humans and language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8425–8444, Mexico City, Mexico. Association for Computational Linguistics.

Richard Evans, David Saxton, David Amos, Pushmeet Kohli, and Edward Grefenstette. 2018. Can neural networks understand logical entailment? *Preprint*, arXiv:1802.08535.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *Preprint*, arXiv:1503.02531.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Geonhee Kim, Marco Valentino, and Andre Freitas. 2025. Reasoning circuits in language models: A mechanistic interpretation of syllogistic inference. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 10074–10095, Vienna, Austria. Association for Computational Linguistics.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Andrew K Lampinen, Ishita Dasgupta, Stephanie C Y Chan, Hannah R Sheahan, Antonia Creswell, Dharshan Kumaran, James L McClelland, and Felix Hill. 2024. Language models, like humans, show content effects on reasoning tasks. *PNAS Nexus*, 3(7):pgae233.

Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. 2017. Gated graph sequence neural networks. *Preprint*, arXiv:1511.05493.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 305–329, Nusa Dua, Bali. Association for Computational Linguistics.

OpenAI. 2025. Gpt-5.1 technical report. Proprietary large language model accessed via the OpenAI API.

Kentaro Ozeki, Risako Ando, Takanobu Morishita, Hirohiko Abe, Koji Mineshima, and Mitsuhiro Okada. 2024. Exploring reasoning biases in large language models through syllogism: Insights from the NeuBAROCO dataset. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 16063–16077, Bangkok, Thailand. Association for Computational Linguistics.

Xin Quan, Marco Valentino, Louise A. Dennis, and Andre Freitas. 2024. Verification and refinement of natural language explanations through LLM-symbolic theorem proving. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2933–2958, Miami, Florida, USA. Association for Computational Linguistics.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Leonardo Ranaldi, Marco Valentino, and Andre Freitas. 2025. Improving chain-of-thought reasoning via

quasi-symbolic abstractions. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 17222–17240. Association for Computational Linguistics.

Tim Rocktäschel and Sebastian Riedel. 2017. End-to-end differentiable proving. *Preprint*, arXiv:1705.11040.

S Seals and Valerie Shalin. 2024. Evaluating the deductive competence of large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8614–8630, Mexico City, Mexico. Association for Computational Linguistics.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. 2019. CLUTRR: A diagnostic benchmark for inductive reasoning from text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4506–4515, Hong Kong, China. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Marco Valentino, Geonhee Kim, Dhairya Dalal, Zhixue Zhao, and André Freitas. 2025. Mitigating content effects on reasoning in language models through fine-grained activation steering. *Preprint*, arXiv:2505.12189.

Marco Valentino, Leonardo Ranaldi, Giulia Pucci, Federico Ranaldi, and André Freitas. 2026. Semeval-2026 task 11: Disentangling content and formal reasoning in large language models. In *Proceedings of the 20th International Workshop on Semantic Evaluation (SemEval-2026)*.

Magdalena Wysocka, Danilo Carvalho, Oskar Wysocki, Marco Valentino, and Andre Freitas. 2025. SylloBio-NLI: Evaluating large language models on biomedical syllogistic reasoning. 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 7235–7258, Albuquerque, New Mexico. Association for Computational Linguistics.

Jundong Xu, Hao Fei, Liangming Pan, Qian Liu, Mong-Li Lee, and Wynne Hsu. 2024. Faithful logical reasoning via symbolic chain-of-thought. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13326–13365, Bangkok, Thailand. Association for Computational Linguistics.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2024. Large language models as optimizers. In *International Conference on Learning Representations*, volume 2024, pages 12028–12068.

A Fine-Tuning Details

A.1 Distillation Data Generation

We construct the fine-tuning dataset using a teacher-student distillation setup. A strong proprietary language model (GPT-5.1) is prompted to solve syllogistic validity tasks using explicit formal reasoning.

For each syllogism, the teacher model is instructed to: (i) rewrite the argument using abstract set-theoretic notation, (ii) reason about validity by checking for the existence of a countermodel, and (iii) output a structured JSON object containing a reasoning trace (`thought_process`), a binary validity label (`validity`), and a concise justification (`reason`).

The gold validity label provided in the dataset is enforced during generation, even in cases where the teacher identifies a semantic countermodel. This ensures consistency with the task definition while preserving exposure to explicit countermodel-based reasoning.

All teacher outputs are stored and used as supervision for student models. No filtering or post-editing of reasoning traces is applied.

A.2 Fine-tuning Objective and Loss Masking

Student models are fine-tuned as causal language models on sequences formed by concatenating the instruction prompt and the teacher-generated JSON output. To prevent the model from learning to reproduce the prompt, we apply **loss masking** over all prompt tokens.

Formally, given token sequence $x = (x_1, \ldots, x_T)$ and label sequence $y = (y_1, \ldots, y_T)$, loss is computed only for positions corresponding to the JSON output, while prompt positions are assigned an ignore index. This training strategy encourages the model to learn structured reasoning and decision generation rather than prompt memorization.

A.3 QLoRA Configuration

All student models are fine-tuned using QLoRA, loading the base model weights in 4-bit NF4

Model	Tr BS	Ev BS	Acc	Ep	LR
Llama 3.1 8B	2	2	8	50	2.0×10^{-4}
Llama 3.2 3B	8	8	4	50	2.0×10^{-4}
Qwen2.5 3B	6	6	4	50	2.0×10^{-4}
Qwen2.5 7B	2	2	8	50	2.0×10^{-4}
Phi 4 4B	6	6	4	50	2.0×10^{-4}
Phi 4 14B	2	2	8	50	1.5×10^{-4}

Table 2: Model-specific hyperparameters for QLoRA distillation fine-tuning. Tr BS = training batch size, Ev BS = evaluation batch size, Acc = gradient accumulation steps, Ep = epochs.

quantized form with double quantization. LoRA adapters are inserted into both attention and MLP projection layers. The following modules are adapted: `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`. The LoRA rank is set to $r = 64$ with scaling factor $\alpha = 16$ and dropout 0.05. All base model parameters remain frozen throughout training.

A.4 Training Hyperparameters

Table 2 summarizes the model-specific hyperparameters used for fine-tuning. Batch size and gradient accumulation are adjusted per model to accommodate GPU memory constraints while keeping the effective batch size comparable across settings.

A.5 Optimization Details

We use the `paged_adamw_8bit` optimizer with a cosine learning-rate schedule and warmup ratio 0.03. Gradient checkpointing and mixed-precision training (FP16) are enabled for memory efficiency. Models are evaluated every 100 steps, and the best checkpoints are selected based on validation loss. All experiments are conducted on NVIDIA GPU(GeForce RTX 3090) with sufficient memory to support 4-bit quantized training. Code link: https://github.com/johnny22245/SemEval-2026-Task-11.git

B Prompt Details

B.1 Zero-Shot CoT

Figure 1 illustrates the zero-shot chain-of-thought (ZS-CoT) prompting template used as a baseline in our experiments. The prompt instructs the model to analyze each syllogism using explicit step-by-step reasoning and to produce a structured JSON output containing a reasoning trace, a binary validity judgment, and a concise justification. No task-specific fine-tuning or supervision is applied in this setting; models are evaluated directly using the prompt at inference time. This baseline serves as a reference point for assessing the impact of other experiments on formal reasoning performance.

System: You are a logic engine. Analyze the syllogism below.
User: Syllogism: {{syllogism}}
Instructions: (1) In `thought_process`: *Let's think step by step.*
(2) Output `validity` (true/false).
(3) Output a concise `reason`.
Assistant: *[model generates JSON]*

Figure 1: Zero-shot chain-of-thought (ZS-CoT) prompting template used as the baseline for syllogism validity reasoning.

B.2 Symbolic representations

Figure 2 presents the structured formal-reasoning prompt used in our experiments. The prompt enforces an explicit decomposition of syllogistic reasoning into abstraction, formalisation, explanation, and final answering stages. By requiring models to translate natural language arguments into symbolic representations and to reason about logical entailment step by step, this prompt encourages systematic formal analysis rather than surface-level pattern matching.

B.3 Few-Shot Settings

Figure 3 shows the few-shot prompting template used in our evaluation. The prompt provides demonstration examples illustrating the expected structured output format and the intended focus on formal validity rather than real-world plausibility. At inference time, we instantiate the placeholder with a fixed set of demonstrations and append the test syllogism, then evaluate the model's predicted validity based on its generated response.

C K-CAST Steering Hyperparameters

We evaluate K-CAST using activation-based steering with a k-nearest-neighbor datastore. For each model, hidden representations $\phi(x)$ are extracted from a single transformer block and stored alongside their gold labels (`valid`/`invalid`) from the training split. Class-wise means are computed, and the steering direction is defined as $\Delta_c = \mu_{\texttt{valid}} - \mu_{\texttt{invalid}}$.

For each model, we perform a grid of **25 ablations**, corresponding to all combinations of **five**

```
Use only logical form (ignore real-world meaning).
Follow these steps:
S1: abstract to A,B,C with All/No/Some.
S2: FOL for P1,P2,C using ∀, ∃, →, and ¬.
S3: 1-3 steps: entailment or countermodel.
```

Figure 2: Structured formal-reasoning prompt used for syllogism validity evaluation.

transformer layers and **five steering scales**. Candidate layers are selected from the last five blocks {-8, -7, -6, -5, -4}. The steering scale α is sampled uniformly at random from the range $[-3, 3]$ to obtain five fixed values, which are reused across all layers for that model.

At inference time, a forward hook is registered at the selected layer to apply the K-CAST steerer parameterized by (k, α). Predictions are obtained by comparing the final-token logits for the tokens "valid" and "invalid". For each model, the best-performing (layer, α) pair is selected based on validation accuracy and subsequently used for evaluation on the test split. Unless otherwise stated, we use k=32 nearest neighbors. Table 3 reports all 25 layer–α ablations per model; we select the best configuration by validation accuracy and use it for test evaluation.

D Evaluation Metrics for Subtask 1

We evaluate content-independent syllogistic reasoning using three complementary metrics: Overall Accuracy (ACC), Total Content Effect (TCE), and the Primary Ranking Metric, referred to as the Combined Smooth Score (CSS). These metrics jointly measure logical correctness and robustness to content-based bias.

D.1 Overall Accuracy (ACC)

Overall Accuracy measures the proportion of examples for which the model correctly predicts the gold validity label:

$$\text{ACC} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(\hat{y}_i = y_i),$$

where N is the total number of syllogisms, y_i is the gold label, $\hat{y}_i$ is the model prediction, and $\mathbb{I}(\cdot)$ denotes the indicator function. ACC captures basic logical competence but does not account for sensitivity to plausibility or semantic content.

D.2 Total Content Effect (TCE)

Total Content Effect quantifies a model's susceptibility to content-based bias by measuring accuracy variation across different logical–plausibility conditions. Let $\mathcal{C}$ denote the set of four content conditions in the English subtask, and let ACC_c be the accuracy under condition c. TCE is defined as:

$$\text{TCE} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \left| \text{ACC}_c - \overline{\text{ACC}} \right|,$$

$$\overline{\text{ACC}} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \text{ACC}_c.$$

Lower TCE values indicate stronger invariance to plausibility manipulations and greater content-independent reasoning.

D.3 Primary Ranking Metric: Combined Smooth Score (CSS)

The official ranking metric combines accuracy and robustness into a single score:

$$\text{CSS} = \frac{\text{ACC}}{1 + \ln(1 + \text{TCE})}.$$

This formulation rewards high accuracy while smoothly penalizing content sensitivity, favoring models that are both correct and robust to semantic distractions.

Summary: ACC measures correctness, TCE isolates content effects, and CSS provides a balanced criterion for model comparison. All results reported in this work use these metrics unless stated otherwise.

D.4 Sub-task Results

Note on Unreported Scores in Subtasks 2 and 4: Some configurations are marked with "–" in Subtask 2 and selected Subtask 4 settings. These runs produced outputs that could not be reliably evaluated due to generation failures rather than reasoning errors. In several cases, the models entered repetitive generation loops (e.g., repeated `\n` or `\t` tokens), resulting in incomplete or truncated responses that did not contain the required structured fields. Because Subtask 2 requires strict structured premise identification with exact field-level matching, and Subtask 4 expects a fixed output schema, such malformed outputs could not be parsed by the official evaluation script. These cases are therefore reported as unreported ("–") for transparency.

Model	Layer	α=2.3774	α=−1.0460	α=−1.8372	α=2.0123	α=−0.4846
Llama 3.1 8B	−8	0.4167	0.6528	**0.6667**	0.4444	0.6250
Llama 3.1 8B	−7	0.4167	0.6528	0.6597	0.4583	0.6250
Llama 3.1 8B	−6	0.4306	0.6528	0.6528	0.4653	0.6250
Llama 3.1 8B	−5	0.4236	0.6528	0.6597	0.4583	0.6181
Llama 3.1 8B	−4	0.4306	0.6458	0.6528	0.4653	0.6181
Llama 3.2 3B	−8	0.4792	0.5625	**0.6111**	0.5000	0.5486
Llama 3.2 3B	−7	0.4931	0.5625	0.6111	0.5000	0.5486
Llama 3.2 3B	−6	0.4722	0.5694	0.6042	0.5000	0.5486
Llama 3.2 3B	−5	0.4722	0.5694	0.6042	0.4931	0.5486
Llama 3.2 3B	−4	0.4722	0.5694	0.6042	0.4931	0.5486
Qwen 2.5 3B	−8	0.3125	0.7292	0.7500	0.3681	0.7292
Qwen 2.5 3B	−7	0.2778	0.7500	**0.7708**	0.2917	0.7361
Qwen 2.5 3B	−6	0.2917	0.7431	0.7500	0.3264	0.7292
Qwen 2.5 3B	−5	0.2847	0.7361	0.7569	0.3194	0.7292
Qwen 2.5 3B	−4	0.3125	0.7361	0.7361	0.3264	0.7292
Qwen 2.5 7B	−8	0.3958	0.6875	0.7500	0.4097	0.6667
Qwen 2.5 7B	−7	0.4028	0.7083	0.7639	0.4097	0.6806
Qwen 2.5 7B	−6	0.3542	0.7153	0.7986	0.3611	0.6806
Qwen 2.5 7B	−5	0.3542	0.7361	**0.8056**	0.3681	0.6806
Qwen 2.5 7B	−4	0.3889	0.7083	0.7639	0.4028	0.6667
Phi 4 4B	−8	**0.4861**	0.4861	0.4861	0.4861	0.4861
Phi 4 4B	−7	0.4861	0.4861	0.4861	0.4861	0.4861
Phi 4 4B	−6	0.4861	0.4861	0.4861	0.4861	0.4861
Phi 4 4B	−5	0.4722	0.4861	0.4861	0.4861	0.4861
Phi 4 4B	−4	0.4653	0.4861	0.4861	0.4861	0.4861
Phi 4 14B	−8	0.4861	0.4861	0.5000	0.4861	0.4861
Phi 4 14B	−7	0.4861	0.4861	0.5000	0.4861	0.4861
Phi 4 14B	−6	0.4931	0.4931	0.5000	0.4861	0.4861
Phi 4 14B	−5	0.4931	0.4931	**0.5069**	0.4931	0.4861
Phi 4 14B	−4	0.4931	0.4931	0.5069	0.4931	0.4861

Table 3: K-CAST validation accuracy across 25 hyperparameter settings per model.

Model (FT Variant)	Subtask 1			Subtask 2		Subtask 3			Subtask 4			
	Acc	TCE	CSS	Acc	CSS	Acc	TCE	CSS	Acc	F1	TCE	CSS
Llama 3.1 8B (ZS)	84.29	14.87	22.39	–	–	88.02	3.13	36.42	66.67	14.56	12.99	11.16
Llama 3.1 8B (+Opt)	85.34	12.77	23.56	–	–	86.46	4.69	31.57	64.58	13.28	16.18	10.13
Llama 3.2 3B (ZS)	70.68	8.75	21.56	–	–	77.60	8.06	24.22	60.94	16.54	18.37	9.77
Llama 3.2 3B (+Opt)	67.54	7.71	21.34	–	–	75.52	8.97	22.89	62.50	21.94	16.20	10.98
Qwen 2.5 3B (ZS)	71.98	4.35	26.89	–	–	75.00	6.67	24.70	–	–	–	–
Qwen 2.5 3B (+Opt)	71.58	4.71	26.10	–	–	83.33	7.08	26.97	–	–	–	–
Qwen 2.5 7B (ZS)	90.00	5.47	31.38	–	–	80.10	13.38	21.85	–	–	–	–
Qwen 2.5 7B (+Opt)	86.63	6.24	29.07	–	–	78.42	4.94	28.19	–	–	–	–
Phi-4 4B (ZS)	70.53	15.63	18.51	–	–	75.00	18.96	18.78	–	–	–	–
Phi-4 4B (+Opt)	71.58	17.15	18.36	–	–	69.27	16.00	18.07	52.08	4.16	7.72	8.88
Phi-4 14B (ZS)	86.91	3.13	35.96	–	–	91.15	5.16	32.34	–	–	–	–
Phi-4 14B (+Opt)	85.86	7.29	27.56	–	–	88.54	6.16	29.83	77.08	10.42	4.17	16.56

Table 4: Results across all subtasks for fine-tuned (FT) models evaluated with ZS-CoT (ZS) and LLM-Optim (+Opt). Subtasks 1 & 3 report Accuracy, TCE, and CSS; Subtask 4 reports Accuracy, F1 over premises, TCE, and CSS. †

† Scores are from post-evaluation runs and may slightly differ from the official evaluation results due to stochastic LLM generation and environment differences. – Denotes unreported scores for runs where outputs could not be reliably parsed/evaluated due to format inconsistencies under the required structured prediction setting.

E Decoding Strategy and Inference Pipeline

All inference experiments are performed using the vLLM framework, which enables efficient large-scale generation with optimized GPU memory management. GPU resources are configured by setting the `CUDA_VISIBLE_DEVICES` environment variable. In our setup, inference runs across two GPUs using

tensor parallelism, allowing the model to distribute computation while maintaining synchronized decoding.

We use deterministic decoding settings to ensure stable and comparable outputs across models. The temperature is set to 0.0 and top-p to 1.0, producing fully deterministic generation. The maximum generation length is limited to 1024 tokens. A repetition penalty of 1.1 is applied to discourage repetitive loops. In addition, a small list of blocked tokens is provided through the `bad_words` parameter to prevent pathological generation patterns occasionally observed during long reasoning traces.

To enforce structured outputs, we use guided decoding supported by vLLM. Model responses are constrained to follow a predefined JSON schema with three fields: `thought_process`, `validity`, and `reason`. The schema is defined using a Pydantic model and automatically converted to JSON format. During decoding, the guided decoding mechanism ensures that generated tokens remain consistent with this structure, which reduces formatting errors and simplifies downstream evaluation. Prompt inputs are processed in batches and distributed across parallel workers to maximize GPU utilization. Each prompt is mapped to a unique identifier, and generated outputs are returned in the same order to preserve alignment with the evaluation dataset.

Finally, generated outputs are parsed and validated against the expected schema. If all required fields are present, the response is converted to structured JSON format. If parsing fails or required keys are missing, the instance is flagged while preserving the raw model output(used for manual parsing for evaluation). This ensures robustness during large-scale inference while maintaining consistency for evaluation.

Example 1
Syllogism:
Every wolf is a mammal.
Nothing that is a mammal is a reptile.
Therefore, no reptile is a wolf.

S1: P1: All A are B.
    P2: No B are C.
    C : No C are A.

S2: P1: $\forall x(A(x) \rightarrow B(x))$
    P2: $\forall x(B(x) \rightarrow \neg C(x))$
    C : $\forall x(C(x) \rightarrow \neg A(x))$

S3: From P2: $B \rightarrow \neg C$.
    Contraposition: $C \rightarrow \neg B$.
    With P1 ($A \rightarrow B$), derive C $\rightarrow \neg A$.

Validity: Valid
Reason: $A \subseteq B$ and $B \cap C = \emptyset \rightarrow C \subseteq \neg A$.

Example 2
Syllogism:
No paintings are machines.
Some sculptures are paintings.
Thus, some sculptures are machines.

S1: P1: No A are B.
    P2: Some C are A.
    C : Some C are B.

S2: P1: $\forall x(A(x) \rightarrow \neg B(x))$
    P2: $\exists x(C(x) \wedge A(x))$
    C : $\exists x(C(x) \wedge B(x))$

S3: From P2 obtain x with $C \wedge A$.
    P1 forces $\neg B$.
    No entailment to $C \wedge B$.

Validity: Invalid
Reason: Countermodel exists.

Example 3
Syllogism:
No robots are trees.
All appliances are robots.
Therefore, some appliances are trees.

S1: P1: No B are C.
    P2: All A are B.
    C : Some A are C.

S2: P1: $\forall x(B(x) \rightarrow \neg C(x))$
    P2: $\forall x(A(x) \rightarrow B(x))$
    C : $\exists x(A(x) \wedge C(x))$

S3: $A \rightarrow B$ and $B \rightarrow \neg C$ imply $A \rightarrow \neg C$.
    Hence $\neg\exists x(A(x) \wedge C(x))$.
    Conclusion contradicts this.

Validity: Invalid
Reason: Premises entail no A are C.

Figure 3: Structured formal reasoning examples produced under the logical abstraction prompt.