Compositional Consistency-Guided Decoding for Three-Way Logical Question Answering

Tianyi Huang [1] **Ming Hou** [2] **Jiaheng Su** [2] **Yutong Zhang** [2] **Ziling Zhang** [2]

Abstract

Logical question answering offers a rare advantage for evaluating LLM reasoning: for some input transformations, the required relationship between outputs is exact. In three-way logical QA, a consistent premise set assigns a hypothesis H one of TRUE, FALSE, or UNKNOWN, and the label for the negated hypothesis $\neg H$ must follow a fixed negation map. Yet black-box LLMs can treat these paired queries as unrelated, producing inconsistent labels or retreating to UNKNOWN even when the premises decide one side. We turn this built-in structure into a test-time decoding rule. CGD-PD queries H and $\neg H$, projects candidate labels onto the negation-consistency constraint, and invokes targeted binary entailment probes only when the three-way signal remains ambiguous. The method is training-free, solver-free, and compatible with ordinary structured prompting. On FOLIO's first-order logic fields, CGD-PD converts a simple consistency diagnostic into a practical reliability layer: it reduces epistemic abstention while preserving the task's genuine UNKNOWN option. The results suggest a broader design principle for LLM reasoning systems: when a task exposes known compositional relations, decoding should use them rather than evaluate each prompt in isolation.

1. Introduction

Compositionality—the ability to construct and reason about complex objects from reusable parts and their relations—is often treated as a central ingredient for systematic generalization (Fodor & Pylyshyn, 1988; Lake & Baroni, 2018; Hupkes et al., 2021). For foundation models, a basic test of compositional behavior is whether predictions on related inputs respect known relationships between those inputs. In logical question answering (QA), such relationships are especially useful because they are not merely stylistic perturbations: they impose hard constraints on the outputs. This makes logic-oriented QA a useful setting for studying whether a model's answers are accurate, stable, and semantically consistent under simple logical transformations (Bowman et al., 2015; Williams et al., 2018; Ribeiro et al., 2020; Cho et al., 2025).

This paper focuses on three-way logical QA, where a set of assumptions S and a hypothesis H are paired with one of three labels, $\{$TRUE, FALSE, UNKNOWN$\}$, corresponding to $S \models H$, $S \models \neg H$, or neither. Three-way labeling is attractive because it distinguishes contradiction from underspecification. However, in LLM-based pipelines, UNKNOWN can play two roles. It can represent genuine logical underspecification, but it can also act as an implicit abstention when the model is uncertain, inconsistent, or sensitive to phrasing (Kadavath et al., 2022; Cho et al., 2025). This makes evaluation subtle: a model that predicts UNKNOWN frequently may appear conservative, while still failing to answer cases whose label is logically determined.

A second issue is more structural. For consistent premises, the label for H determines the label for $\neg H$ through a fixed negation map. Under this map, TRUE and FALSE swap, while UNKNOWN remains UNKNOWN. Thus, three-way logical QA contains a simple compositional consistency relation between two views of the same underlying inference problem. Standard prompting usually ignores this relation by treating H and $\neg H$ as independent inputs.

From a diagnostic to a decoder. The coupling induced by negation creates a constrained two-view inference problem: asking about H and $\neg H$ provides two noisy observations of the same underlying logical state. If one view is decisive and the other is ambiguous, the decisive view can be mapped back through the negation relation; if both views are ambiguous, a narrower entailment question can test whether either side is supported. This suggests an inference-time layer that is more structured than repeated sampling of a single prompt, but much lighter than external proof search.

[1]Ryquo [2]App-In Club. Correspondence to: Tianyi Huang <tianyi@ryquo.com>.

Accepted to the 2nd Workshop on Compositional Learning at ICML 2026, Seoul, South Korea. Copyright 2026 by the author(s).

Approach overview. We propose **Consistency-Guided Decoding with Proof-Driven Disambiguation (CGD-PD)**. Given (S, H), CGD-PD first queries the same 3-way classifier on H and on a mechanically negated form $\neg H$. It accepts pairs that already satisfy the negation map, applies a targeted UNKNOWN re-evaluation only to sides that abstain, uses binary entailment probes when both sides remain UNKNOWN, and adjudicates only when both sides are decisive but contradictory. The final decision is always interpreted through the negation-consistency constraint. CGD-PD is training-free, uses no external solver, and can be applied to black-box models that support structured prompting.

Empirical focus. We evaluate CGD-PD on FOLIO's first-order logic fields (Han et al., 2024a), where negation can be applied mechanically and the three-way label is well defined. This controlled setting isolates whether a known compositional relation can improve inference-time reliability when the negation operation is unambiguous. Across two instruction-tuned API models under the same strict prompt family, CGD-PD improves validation accuracy and reduces epistemic UNKNOWN. The diagnostics show that the gains are concentrated on determinate examples that the single-call classifier had treated as UNKNOWN, while the main remaining tension is protecting genuinely underspecified examples from over-resolution.

Contributions. This work makes three contributions:

- We formulate three-way logical QA as a compositional consistency problem in which predictions for H and $\neg H$ must satisfy a deterministic label relation.
- We introduce CGD-PD, a lightweight test-time wrapper that combines neural 3-way classification, symbolic negation projection, targeted UNKNOWN re-evaluation, and binary entailment probes.
- We provide a focused evaluation on FOLIO's formal fields, including accuracy, abstention, coverage, and gold-UNKNOWN diagnostics that separate epistemic recovery from overconfident resolution.

2. Problem Setting

Each instance consists of a premise set S and a hypothesis H. The task is to output one of three labels:

$$\begin{aligned} \text{TRUE}: &\quad S \models H, \\ \text{FALSE}: &\quad S \models \neg H, \\ \text{UNKNOWN}: &\quad S \not\models H \text{ and } S \not\models \neg H. \end{aligned}$$

We assume S is logically consistent, as is typical for curated benchmarks such as FOLIO (Han et al., 2024a).

2.1. Negation mapping and consistency

Negation induces a deterministic mapping on labels:

$$\begin{aligned} \mathsf{NegMap}(\text{TRUE}) &= \text{FALSE}, \\ \mathsf{NegMap}(\text{FALSE}) &= \text{TRUE}, \\ \mathsf{NegMap}(\text{UNKNOWN}) &= \text{UNKNOWN}. \end{aligned}$$

If a system outputs labels for both H and $\neg H$, negation consistency requires

$$y(\neg H) = \mathsf{NegMap}(y(H)). \tag{1}$$

Although (1) is elementary, LLMs can violate it when prompted independently. CGD-PD treats (1) as a hard constraint for its final decision.

2.2. Epistemic UNKNOWN versus genuine UNKNOWN

On datasets with an explicit UNKNOWN label, UNKNOWN can arise for two different reasons:

- **Genuine underspecification:** the premises do not entail either side.
- **Epistemic UNKNOWN:** the model returns UNKNOWN due to uncertainty, instability, or conservative behavior even when one side is entailed.

This distinction matters because epistemic UNKNOWN reduces accuracy and coverage without reflecting the task semantics. Our goal is to reduce epistemic UNKNOWN while preserving genuine underspecification whenever the available probes do not support a decisive label.

3. Method: CGD-PD

CGD-PD is a test-time wrapper around an LLM that can be prompted as a 3-way classifier. It uses three reusable components: (i) a base 3-way classifier, (ii) a targeted UNKNOWN fixer, and (iii) binary entailment probes used as a lightweight proof check. The final decoder composes these components with the symbolic negation map in (1).

3.1. Base 3-way classifier

We define $\mathsf{Classify}(S, H) \in \{\text{TRUE}, \text{FALSE}, \text{UNKNOWN}\}$ by prompting an LLM to output a structured label. In our experiments, the prompt requires a strict schema so that the model returns only one of the three labels. Because UNKNOWN can act as abstention, we enable a soft bias-away-from-UNKNOWN instruction in the base prompt. In the implementation this is controlled by an `unknown_penalty` configuration parameter, set to 0.5 in the reported runs; the scalar is not an API-level logit bias and is not exposed to the model as a numerical weight. Instead, it adds the qualitative instruction that UNKNOWN should be used only when

the premise block genuinely leaves both sides possible.[1] Appendix A provides the prompt templates for the base classifier, the UNKNOWN fixer, the binary entailment probe, and the adjudicator.

3.2. Composing hypothesis and negation probes

Given (S, H), CGD-PD first queries the base classifier on H and on a mechanically negated hypothesis $\neg H$. In practice, we represent negation with a canonical wrapper (e.g., `NOT:` H) and explicitly define its semantics in the prompt. We denote the two outputs by $y_H = \mathsf{Classify}(S, H)$ and $y_{\neg H} = \mathsf{Classify}(S, \neg H)$. If $y_{\neg H} = \mathsf{NegMap}(y_H)$ and at least one side is decisive, we accept the pair as negation-consistent and return y_H. Otherwise, we proceed to disambiguation or adjudication.

3.3. Targeted UNKNOWN fixing

When one side returns UNKNOWN, we do not immediately force a decision. Instead, we run a targeted prompt $\mathsf{FixUnknown}(S, H)$ that asks the model to re-evaluate the same premise block and hypothesis under the original 3-way decision rule. The fixer treats UNKNOWN as a last resort: it should return TRUE or FALSE when the premise block contains a decisive logical path, and return UNKNOWN only when a specific required fact is missing. When it keeps UNKNOWN, the prompt asks it to name the missing premise and, when possible, quote the part of the premise block that indicates the gap. When it returns a decisive label, the auxiliary fields are set to null.

After applying the fixer on the UNKNOWN side(s), we first check whether the updated pair is already negation-consistent. If so, we return the label for H. If exactly one side is decisive and the other remains UNKNOWN, we project the remaining side through (1) and return the corresponding label for H. This projection is rule-constrained but not a formal guarantee: it depends on the reliability of the decisive model call that triggered it.

3.4. Proof-driven disambiguation via binary entailment probes

If both sides remain UNKNOWN after targeted fixing, we use binary entailment probes $b_H = \mathsf{EntailsYesNo}(S, H) \in \{\text{YES}, \text{NO}\}$ and $b_{\neg H} = \mathsf{EntailsYesNo}(S, \neg H) \in \{\text{YES}, \text{NO}\}$. The binary probe asks a narrower question than 3-way classification and removes UNKNOWN as an output option. This can reveal cases where the 3-way prompt used UNKNOWN as a default, but it can also overcommit; the decoder therefore uses the two binary answers only when they agree with a consistent one-sided entailment pattern. Specifically, if $(b_H, b_{\neg H}) = (\text{YES}, \text{NO})$, CGD-PD returns TRUE; if $(b_H, b_{\neg H}) = (\text{NO}, \text{YES})$, it returns FALSE; otherwise, it keeps UNKNOWN. This includes the conflict case where both probes say YES; rather than privileging one side arbitrarily, the method abstains.

We call this step proof-driven because focused entailment questions provide lightweight evidence that one side is derivable from S. It is not a formal proof system, and we do not claim solver-level guarantees.

3.5. Adjudication for inconsistent decisive pairs

Finally, if y_H and $y_{\neg H}$ are both decisive but violate (1), we invoke a lightweight adjudicator prompt. It chooses between the two consistent assignments, returning either y_H or $\mathsf{NegMap}(y_{\neg H})$. The adjudicator is used only when the classifier has produced a contradictory decisive pair.

3.6. Algorithm and compute

Algorithm 1 summarizes CGD-PD. The wrapper makes two calls in the common case and up to six calls when both fixers and both binary probes are required. In our FOLIO validation runs, it uses 4–5 calls on average (Section 5).

4. Experimental Setup

Our experiments ask whether a minimal inference-time layer can improve task accuracy and reliability diagnostics. We focus on consistency under negation and epistemic abstention.

4.1. Dataset: FOLIO

We evaluate on FOLIO (Han et al., 2024a), which pairs short natural-language stories with first-order logic annotations for both the premises and the hypothesis. FOLIO labels each hypothesis as one of TRUE, FALSE, or UNKNOWN under standard first-order semantics. The FOL fields reduce ambiguity in negation, allowing us to mechanically represent $\neg H$ in a controlled way.

Input representation. Unless otherwise stated, we run CGD-PD on the FOL fields: premises are the dataset-provided FOL formulas, and hypotheses are the dataset-provided FOL conclusions. This choice makes the negation operation unambiguous and allows us to isolate the consistency mechanism. It also defines the scope of the evaluation: natural-language negation scope is an important and harder setting, and robustness there should be tested directly when negation must be generated or interpreted from text.

[1] Appendix C reports development-stage variants that helped motivate the final selective design.

Algorithm 1 CGD-PD: Consistency-Guided Decoding with Proof-Driven Disambiguation

Require: Premises S, hypothesis H, negation mapping NegMap
$y_H \leftarrow \mathsf{Classify}(S, H)$
$y_{\neg H} \leftarrow \mathsf{Classify}(S, \neg H)$ (mechanical negation)
if $y_{\neg H} = \mathsf{NegMap}(y_H) \wedge \neg(y_H = \textsc{Unknown} \wedge y_{\neg H} = \textsc{Unknown})$ **then**
  return y_H
end if
if $y_H = \textsc{Unknown}$ **then**
  $y_H \leftarrow \mathsf{FixUnknown}(S, H)$
end if
if $y_{\neg H} = \textsc{Unknown}$ **then**
  $y_{\neg H} \leftarrow \mathsf{FixUnknown}(S, \neg H)$
end if
if $y_H \neq \textsc{Unknown} \wedge y_{\neg H} \neq \textsc{Unknown} \wedge y_{\neg H} = \mathsf{NegMap}(y_H)$ **then**
  return y_H (consistent after fixing)
end if
if $y_H \neq \textsc{Unknown} \wedge y_{\neg H} = \textsc{Unknown}$ **then**
  return y_H (project via NegMap)
else if $y_H = \textsc{Unknown} \wedge y_{\neg H} \neq \textsc{Unknown}$ **then**
  return $\mathsf{NegMap}(y_{\neg H})$
end if
if $y_H = \textsc{Unknown} \wedge y_{\neg H} = \textsc{Unknown}$ **then**
  $b_H \leftarrow \mathsf{EntailsYesNo}(S, H)$
  $b_{\neg H} \leftarrow \mathsf{EntailsYesNo}(S, \neg H)$
  if $b_H = \textsc{Yes} \wedge b_{\neg H} = \textsc{No}$ **then**
    return $\textsc{True}$
  else if $b_H = \textsc{No} \wedge b_{\neg H} = \textsc{Yes}$ **then**
    return $\textsc{False}$
  else
    return $\textsc{Unknown}$
  end if
end if
return $\mathsf{Adjudicate}(S, H, y_H, y_{\neg H})$ (decisive conflict)

4.2. Models and decoding

We evaluate two instruction-tuned LLMs accessed via API: `gpt-5.2` and `claude-sonnet-4-5`. All prompts require structured outputs with an explicit schema, and temperature is set to 0 to reduce sampling variability. We do not include chain-of-thought in the main protocol, which reduces format variability and keeps the prompt family consistent across methods. This protocol is intentionally conservative and should not be interpreted as measuring the best achievable performance of either model on FOLIO; in particular, a more permissive prompt or explicit reasoning format could change the single-call baseline and the relative gain from CGD-PD.

4.3. Baselines and development-stage variants

Our primary comparison is against **Single**, which makes one call to $\mathsf{Classify}(S, H)$. Appendix C also reports two development-stage variants used during method design: **gate3**, a 3-call gating-style variant without binary entailment probes, and **CGR3**, a 3-call consistency-guided redundancy variant without the proof-driven disambiguation stage. CGD-PD is not intended to replace ensembling methods such as self-consistency (Wang et al., 2023); it addresses a complementary axis, namely logical coupling across related probes. Matched-call self-consistency is a natural complementary baseline for call-budget efficiency; accordingly, our claims focus on the reliability effects of using logically coupled probes rather than on optimal call allocation.

4.4. Metrics

We report:

- **Accuracy**: fraction of hypotheses whose predicted label matches the gold label.
- UNKNOWN **rate**: fraction of predictions that are UNKNOWN.
- **Epistemic** UNKNOWN **rate**: fraction of gold determinate examples (TRUE or FALSE) predicted as UNKNOWN.
- **Compute**: average number of model calls per example.

We report 95% confidence intervals for key deltas using paired bootstrap resampling over examples.

5. Results

5.1. Main quantitative results

Table 1 summarizes validation performance on FOLIO's FOL fields (204 examples). Using the same structured prompt family for the single-call and wrapped systems, CGD-PD improves accuracy on both models while reducing UNKNOWN predictions.

Confidence intervals. The paired bootstrap confidence intervals exclude zero for both models. For GPT-5.2, CGD-PD yields a +4.4 point accuracy gain (95% CI: +1.5 to +7.4) and a -3.4 point reduction in UNKNOWN rate (95% CI: -6.4 to -0.5). For Claude Sonnet 4.5, accuracy improves by +6.8 points (95% CI: +3.4 to +10.3) and UNKNOWN rate drops by -16.7 points (95% CI: -21.6 to -11.8). The Claude single-call baseline is notably conservative under this strict no-chain-of-thought schema, with 75.5% UNKNOWN predictions. We therefore interpret the Claude result as evidence that CGD-PD can recover from epistemic abstention under this prompt family, not as a claim about the model's best possible FOLIO performance.

6. Analysis

We analyze how CGD-PD changes predictions beyond the aggregate accuracy numbers. Because the method reduces

Table 1. FOLIO validation results on FOL fields (204 validation examples). Accuracy, UNKNOWN rate, and epistemic UNKNOWN rate are percentages; epistemic UNKNOWN is the fraction of gold TRUE/FALSE examples predicted as UNKNOWN. "Calls" is the mean number of wrapper model invocations per example.

Model	Method	Acc. (%)	UNKNOWN (%)	Epist. UNKNOWN (%)	Calls
GPT-5.2	Single	63.7	57.4	41.5	1.00
GPT-5.2	CGD-PD	**68.1**	**53.9**	**36.3**	4.36
Claude Sonnet 4.5	Single	42.2	75.5	72.6	1.00
Claude Sonnet 4.5	CGD-PD	**49.0**	**58.8**	**53.3**	4.91

abstention, we separately examine final accuracy, coverage, and the reliability of non-UNKNOWN decisions. We use "answered accuracy" to mean accuracy conditional on a non-UNKNOWN prediction; this is not probability calibration, but it tests whether the wrapper merely converts UNKNOWN into decisive labels or also preserves reliability among the labels it chooses to answer.

6.1. Confusion matrices: resolving epistemic UNKNOWN

Figure 1 in Appendix B shows row-normalized confusion matrices for GPT-5.2 and Claude Sonnet 4.5. Two patterns stand out:

- **Reduced abstention on determinate cases.** For both models, many gold TRUE and FALSE examples are predicted as UNKNOWN by the single classifier. CGD-PD moves some of these to the correct decisive label, reducing epistemic UNKNOWN.
- **Boundary between recovery and over-resolution.** For GPT-5.2, the fraction of gold UNKNOWN predicted as UNKNOWN is unchanged. For Claude Sonnet 4.5, CGD-PD predicts UNKNOWN less often even when the gold label is UNKNOWN, introducing additional TRUE/FALSE errors. The net effect remains positive because the recovered determinate cases dominate, but this diagnostic identifies where a future variant should be more selective.

6.2. Coverage and decisive-label reliability

Table 2 summarizes coverage, answered accuracy, and errors on gold-UNKNOWN examples. For GPT-5.2, CGD-PD increases coverage from 42.6% to 46.1% and answered accuracy from 79.3% to 83.0%. For Claude Sonnet 4.5, coverage increases more substantially, from 24.5% to 41.2%, while answered accuracy increases from 60.0% to 61.9%. These numbers suggest that CGD-PD is not simply replacing abstentions with lower-quality decisive labels in this setting. The gold-UNKNOWN columns also identify the key selectivity challenge: for Claude, the number of genuinely underspecified examples converted to a decisive label rises from 13 to 21. The final-output error profile is mildly asymmetric for Claude (13 TRUE errors vs. 8 FALSE errors on gold UNKNOWN), making probe-level label bias a useful target for follow-up analysis.

6.3. How often does proof-driven disambiguation trigger?

CGD-PD is designed to spend extra calls primarily when the model abstains or is inconsistent. On FOLIO validation, CGD-PD uses the maximum six calls on 54% of GPT-5.2 examples and 61% of Claude examples, reflecting the prevalence of UNKNOWN outputs in this task. On GPT-5.2, CGD-PD changes the single-model prediction on 15/204 examples; on Claude, it changes 34/204, largely by converting UNKNOWN into decisive outputs. These aggregate change counts do not isolate which internal branch caused each change, so they should be read together with the confusion matrices rather than as an adjudicator-specific audit.

6.4. What the current robustness evidence establishes

The paired bootstrap intervals and coverage diagnostics provide evidence that the observed gains are stable under resampling of this validation split and are not merely due to converting UNKNOWN into low-quality decisive answers. The evidence is strongest for the controlled setting studied here: FOL-formula inputs, mechanical negation, two API models, and one strict structured-output prompt family. The most informative next checks are therefore clear: allocate the same call budget to self-consistency or related inference-time baselines, and test natural-language FOLIO inputs where negation scope must be interpreted rather than mechanically specified.

7. Related Work

Compositionality and modular reasoning. Compositional generalization has long been studied as a challenge for neural systems (Fodor & Pylyshyn, 1988; Lake & Baroni, 2018; Hupkes et al., 2021; Keysers et al., 2020). A related line of work studies modular architectures that assemble reusable components for complex tasks (Andreas et al., 2017). CGD-PD is much narrower than these broad programs: it does not learn compositional representations or modules. Instead, it uses a fixed logical relation, nega-

Table 2. Coverage and decisive-label reliability on FOLIO validation (204 examples). Coverage, answered accuracy, and Gold U kept are percentages; Gold U→T/F are counts of genuinely underspecified examples predicted as decisive labels. T/F/U denote TRUE, FALSE, and UNKNOWN.

Model	Method	Cov.	Ans. acc.	Gold U kept	Gold U→T	Gold U→F
GPT-5.2	Single	42.6	79.3	88.4	3	5
GPT-5.2	CGD-PD	46.1	83.0	88.4	4	4
Claude Sonnet 4.5	Single	24.5	60.0	81.2	9	4
Claude Sonnet 4.5	CGD-PD	41.2	61.9	69.6	13	8

tion, to compose multiple inference-time views into a more consistent final decision.

Logic-oriented QA benchmarks. Standard NLI benchmarks such as SNLI and MultiNLI (Bowman et al., 2015; Williams et al., 2018) have been central to evaluating entailment, but natural-language datasets can conflate linguistic heuristics with logical competence. Logic-focused resources add structure to reduce ambiguity, including RuleTaker and ProofWriter (Clark et al., 2021; Tafjord et al., 2021) and more recent benchmarks targeting formal reasoning. FOLIO (Han et al., 2024a) is directly aligned with our setting because it provides first-order logic annotations for both premises and hypotheses together with explicit three-way labels, enabling controlled negation-based transformations. Related resources such as P-FOLIO (Han et al., 2024b) and LogicBench (Parmar et al., 2024) extend evaluation in complementary directions, including human-written reasoning chains and broader logical phenomena.

Consistency and invariance testing. Behavioral testing work such as CheckList (Ribeiro et al., 2020) argues that accuracy alone can hide brittle behaviors under perturbations. Metamorphic testing extends this idea by specifying transformations and expected relations between outputs (Cho et al., 2025). Our work shares the diagnostic motivation, but uses a small set of logic-specific metamorphic relations to guide inference-time decisions rather than only to evaluate failures.

Inference-time verification. Self-consistency and related sampling methods improve reasoning by aggregating repeated generations from the same prompt (Wang et al., 2023). Tree-of-Thoughts and related search-based approaches use larger inference-time budgets to branch and evaluate candidate steps (Yao et al., 2023). CGD-PD is complementary: it uses a small number of calls on logically linked prompts and applies a deterministic constraint to project onto a coherent assignment. The proof-driven disambiguation step is also related to generate-then-verify paradigms and inference-time intervention (Li et al., 2024).

Constrained decoding, abstention, and structured inference. Constrained decoding has a long history in sequence generation, including lexically constrained decoding (Hokamp & Liu, 2017; Post & Vilar, 2018) and incremental parsing constraints for text-to-SQL (Scholak et al., 2021). These methods primarily enforce syntactic or formal-language constraints. CGD-PD can be viewed as a lightweight structured inference layer for one semantic constraint. The explicit UNKNOWN label also resembles selective prediction and abstention, where models can defer on uncertain inputs (Geifman & El-Yaniv, 2017; Kadavath et al., 2022).

8. Scope and Limitations

CGD-PD is a structured inference layer, not a replacement for formal proof search. Its hard guarantee is that the returned label for H corresponds to a negation-consistent assignment over $(H, \neg H)$; label correctness still depends on the model judgments used by the wrapper and on the implementation of the mechanical negation convention. The current study isolates this mechanism on one FOLIO validation split, FOL-formula inputs, two API models, and a strict no-chain-of-thought prompt family. This controlled scope makes the mechanism easier to interpret, while leaving natural-language negation, additional benchmarks, and stronger prompting protocols as important next tests. CGD-PD also spends additional inference-time calls relative to a single classifier, so matched-budget comparisons with self-consistency and related inference-time baselines would sharpen efficiency claims. Finally, the present diagnostics are final-output diagnostics: coverage, answered accuracy, and gold-UNKNOWN error profiles characterize the predictions, but branch-level logs such as fixer changes, adjudicator overrides, and probe-level label biases would give a more detailed account of the mechanism.

9. Conclusion

We presented CGD-PD, a training-free inference-time wrapper for three-way logical QA that makes the negation relation explicit at decoding time. The central idea is simple: predictions for a hypothesis and its negation should not be treated as unrelated samples, because the task already specifies how their labels compose. On FOLIO's FOL fields, applying this constraint with targeted entailment

probes improves both evaluated LLMs and reduces epistemic UNKNOWN under a strict structured-output protocol. The analysis clarifies where the method helps—primarily by recovering determinate examples that were previously abstained on—and where care is still needed, especially around genuinely underspecified cases. More broadly, the work argues for a modest but practical form of reasoning reliability: before adding larger verifiers or new training, inference can often be strengthened by making the task's own compositional structure operational.

Impact Statement

By studying how logically grounded consistency checks can expose and reduce specific failure modes in LLM reasoning, this work contributes to efforts toward more reliable and interpretable AI evaluation. Such methods may be useful in educational tools, analysis assistants, and verification-oriented systems where transparent abstention behavior matters. Consistency is not the same as truth, however: a system can satisfy a local logical relation and still be wrong about the underlying task. Deployment should therefore pair inference-time consistency layers with domain-appropriate safeguards and, when possible, independent verification. Extending these ideas beyond negation to richer logical transformations may support broader settings where abstention, consistency, and reproducibility are central concerns.

References

Andreas, J., Rohrbach, M., Darrell, T., and Klein, D. Neural module networks, 2017. URL https://arxiv.org/abs/1511.02799.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL https://aclanthology.org/D15-1075/.

Cho, S., Ruberto, S., and Terragni, V. Metamorphic testing of large language models for natural language processing. In *2025 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pp. 174–186. IEEE, sep 2025. doi: 10.1109/icsme64153.2025.00025. URL http://dx.doi.org/10.1109/ICSME64153.2025.00025.

Clark, P., Tafjord, O., and Richardson, K. Transformers as soft reasoners over language. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20, 2021. ISBN 9780999241165.

Fodor, J. A. and Pylyshyn, Z. W. *Connectionism and cognitive architecture: a critical analysis*, pp. 3–71. MIT Press, Cambridge, MA, USA, 1988. ISBN 0262660644.

Geifman, Y. and El-Yaniv, R. Selective classification for deep neural networks, 2017. URL https://arxiv.org/abs/1705.08500.

Han, S., Schoelkopf, H., Zhao, Y., Qi, Z., Riddell, M., Zhou, W., Coady, J., Peng, D., Qiao, Y., Benson, L., Sun, L., Wardle-Solano, A., Szabó, H., Zubova, E., Burtell, M., Fan, J., Liu, Y., Wong, B., Sailor, M., Ni, A., Nan, L., Kasai, J., Yu, T., Zhang, R., Fabbri, A., Kryscinski, W. M., Yavuz, S., Liu, Y., Lin, X. V., Joty, S., Zhou, Y., Xiong, C., Ying, R., Cohan, A., and Radev, D. FOLIO: Natural language reasoning with first-order logic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 22017–22031, Miami, Florida, USA, November 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1229. URL https://aclanthology.org/2024.emnlp-main.1229/.

Han, S., Yu, A., Shen, R., Qi, Z., Riddell, M., Zhou, W., Qiao, Y., Zhao, Y., Yavuz, S., Liu, Y., Joty, S., Zhou, Y., Xiong, C., Radev, D., Ying, R., and Cohan, A. P-FOLIO: Evaluating and improving logical reasoning with abundant human-written reasoning chains. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 16553–16565, Miami, Florida, USA, November 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.966. URL https://aclanthology.org/2024.findings-emnlp.966/.

Hokamp, C. and Liu, Q. Lexically constrained decoding for sequence generation using grid beam search, 2017. URL https://arxiv.org/abs/1704.07138.

Hupkes, D., Dankers, V., Mul, M., and Bruni, E. Compositionality decomposed: how do neural networks generalise? (extended abstract). In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20, 2021. ISBN 9780999241165.

Kadavath, S., Conerly, T., Askell, A., Henighan, T., Drain, D., Perez, E., Schiefer, N., Hatfield-Dodds, Z., DasSarma, N., Tran-Johnson, E., Johnston, S., El-Showk, S., Jones, A., Elhage, N., Hume, T., Chen, A., Bai, Y., Bowman, S., Fort, S., Ganguli, D., Hernandez, D., Jacobson, J., Kernion, J., Kravec, S., Lovitt, L., Ndousse, K., Olsson, C., Ringer, S., Amodei, D., Brown, T., Clark, J., Joseph, N., Mann, B., McCandlish, S., Olah, C., and Kaplan, J. Language models (mostly) know what they know, 2022. URL https://arxiv.org/abs/2207.05221.

Keysers, D., Schärli, N., Scales, N., Buisman, H., Furrer, D., Kashubin, S., Momchev, N., Sinopalnikov, D., Stafiniak, L., Tihon, T., Tsarkov, D., Wang, X., van Zee, M., and Bousquet, O. Measuring compositional generalization: A comprehensive method on realistic data, 2020. URL https://arxiv.org/abs/1912.09713.

Lake, B. and Baroni, M. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2873–2882. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/lake18a.html.

Li, K., Patel, O., Viégas, F., Pfister, H., and Wattenberg, M. Inference-time intervention: Eliciting truthful answers from a language model, 2024. URL https://arxiv.org/abs/2306.03341.

Parmar, M., Patel, N., Varshney, N., Nakamura, M., Luo, M., Mashetty, S., Mitra, A., and Baral, C. Logicbench: Towards systematic evaluation of logical reasoning ability of large language models, 2024. URL https://arxiv.org/abs/2404.15522.

Post, M. and Vilar, D. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation, 2018. URL https://arxiv.org/abs/1804.06609.

Ribeiro, M. T., Wu, T., Guestrin, C., and Singh, S. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4902–4912, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.442. URL https://aclanthology.org/2020.acl-main.442/.

Scholak, T., Schucher, N., and Bahdanau, D. Picard: Parsing incrementally for constrained auto-regressive decoding from language models, 2021. URL https://arxiv.org/abs/2109.05093.

Tafjord, O., Dalvi, B., and Clark, P. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 3621–3634, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.317. URL https://aclanthology.org/2021.findings-acl.317/.

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., and Zhou, D. Self-consistency improves chain of thought reasoning in language models, 2023. URL https://arxiv.org/abs/2203.11171.

Williams, A., Nangia, N., and Bowman, S. R. A broad-coverage challenge corpus for sentence understanding through inference, 2018. URL https://arxiv.org/abs/1704.05426.

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T. L., Cao, Y., and Narasimhan, K. Tree of thoughts: Deliberate problem solving with large language models, 2023. URL https://arxiv.org/abs/2305.10601.

A. Prompt Templates

We use structured prompts and parse only the requested JSON fields. The templates below mirror the implementation used for the reported runs, with instance-specific content replaced by placeholders. In the main experiments, `STORY` contains the FOL premise block; a hypothesis prefixed with `NOT:` is treated as the mechanical negation of the remaining text. We do not request or parse chain-of-thought.

A.1. Common Classifier Rules

The base classifier, UNKNOWN fixer, and consistency adjudicator prepend the following shared instruction block before their task-specific instructions.

```
You are a careful, literal logician.
Task: 3-way logical classification of whether STORY entails HYPOTHESIS.

Valid labels: True, False, Unknown.

Decision rule:
1) True = STORY logically entails HYPOTHESIS.
2) False = STORY logically entails NOT(HYPOTHESIS).
3) Unknown = STORY entails neither HYPOTHESIS nor NOT(HYPOTHESIS).

Constraints:
- Use ONLY the STORY. Do not use outside knowledge or assumptions.
- Apply the decision rule literally.
- Do not use Unknown as a response to uncertainty about your own reasoning.
- Use Unknown only when the STORY genuinely leaves both HYPOTHESIS and NOT(HYPOTHESIS) unentailed.

Negation convention:
- If the HYPOTHESIS begins with 'NOT: ', treat it as the logical negation of the remainder.
```

A.2. Input Block

The implementation appends the premise and hypothesis block below to each prompt.

```
STORY:
{STORY}

HYPOTHESIS:
{HYPOTHESIS}
```

A.3. Base 3-way Classifier

```
[[TASK:single]]
<COMMON CLASSIFIER RULES>

Output format: Return ONLY a JSON object: {"label": <label>}
Do not include any extra keys, markdown, or commentary.
```

A.4. Targeted UNKNOWN Re-evaluation

```
[[TASK:fix_unknown]]
<COMMON CLASSIFIER RULES>

This prompt is used only after an earlier call returned Unknown.
Re-evaluate the same STORY and HYPOTHESIS under the decision rule above.

If the STORY entails HYPOTHESIS, return True.
If the STORY entails NOT(HYPOTHESIS), return False.
If neither is entailed, return Unknown.

If you return Unknown, provide:
- missing_premise: the specific fact that would be needed to decide the label,
- quote: an exact quote from the STORY showing the relevant gap, or null if no such quote exists.

If you return True or False, set missing_premise=null and quote=null.

Earlier model output:
- prior_label = {PRIOR_LABEL}

Output format: Return ONLY JSON with keys label, missing_premise, quote.
```

```
No extra keys, markdown, or commentary.
```

A.5. Binary Entailment Probe

```
[[TASK:entails_yesno]]
You are a careful, literal reasoner.
Return ONLY a JSON object with key 'answer' and value 'YES' or 'NO'.

Rules:
- Use ONLY the STORY. Do not use outside knowledge or assumptions.
- Answer YES only if the STORY logically entails the HYPOTHESIS.
- Otherwise answer NO.

STORY:
{STORY}

HYPOTHESIS:
{HYPOTHESIS}

Question: Does the STORY logically entail the HYPOTHESIS?
```

A.6. Consistency Adjudicator

```
[[TASK:cgr3_adjudicator]]
<COMMON CLASSIFIER RULES>

Consistency requirement across H and NOT:H:
- If the final label for H is True, then the corresponding label for NOT:H is False.
- If the final label for H is False, then the corresponding label for NOT:H is True.
- If the final label for H is Unknown, then the corresponding label for NOT:H is Unknown.

Earlier one-shot labels:
- label(H) = {Y_H}
- label(NOT:H) = {Y_NOT_H}

The earlier labels may be inconsistent. Use them only as auxiliary signals.
Your job is to output one final label for H that follows the STORY and satisfies the decision rule.

Output format: Return ONLY JSON: {"label": <label>}.
No extra keys, markdown, or commentary.
```

B. Confusion Matrices

Figure 1 shows row-normalized confusion matrices for the validation split, highlighting how CGD-PD shifts mass away from single-shot UNKNOWN predictions and, in some cases, corrects overconfident TRUE/FALSE errors.

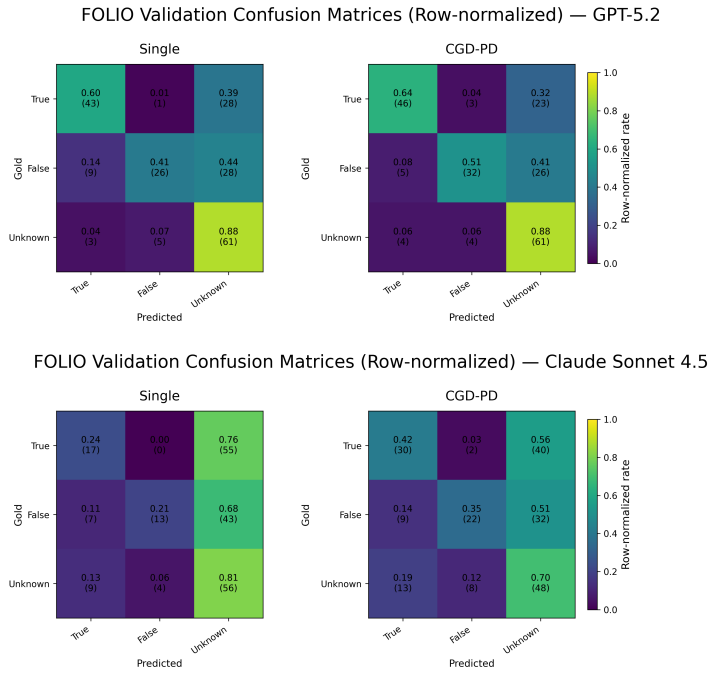

Figure 1. Row-normalized confusion matrices on the FOLIO validation split. For each model, the left heatmap is the single-call classifier and the right heatmap is CGD-PD; rows are gold labels and columns are predicted labels. Each cell shows the row-normalized rate with the raw count in parentheses.

C. Ablations and Negative Results

This appendix summarizes development-stage variants and negative results that informed the final design. For transparency, Table 4 reports train and validation results, and Table 3 shows representative validation cases.

What are gate3 and CGR3? Both variants were intermediate designs explored in development. **gate3** is a 3-call gating baseline that spreads compute across related prompts and applies a lightweight resolution step, but omits CGD-PD's proof-driven binary entailment probes. **CGR3** is a 3-call consistency-guided redundancy variant that probes logically related forms and applies a lightweight consistency-aware resolution rule, but omits proof-driven disambiguation. These variants help isolate what is gained by the final CGD-PD design.

Table 3. Representative GPT-5.2 validation cases illustrating two mechanisms of CGD-PD. The first case preserves a genuine UNKNOWN when neither side is entailed; the second uses the decisive negated probe to correct the final label for H. T/F/U denote TRUE/FALSE/UNKNOWN.

Premise excerpt	Hypothesis H	Gold	Single	Ours	Signals used by our method
The Picuris Mountains are a mountain range in New Mexico or Texas.	The Picuris Mountains are in Texas.	U	F	U	Probe labels: $y(H)$ = UNKNOWN and $y(\neg H)$ = UNKNOWN. CGD-PD then runs proof-driven entailment checks and finds $\mathsf{EntailsYesNo}(S, H)$ = NO and $\mathsf{EntailsYesNo}(S, \neg H)$ = NO, so it preserves UNKNOWN. This illustrates that the wrapper need not force a decisive answer when neither side is supported.
The Picuris Mountains are a mountain range in New Mexico or Texas.	The Picuris Mountains are in New Mexico.	F	T	F	Probe labels: $y(H)$ = UNKNOWN and $y(\neg H)$ = TRUE. Because the negated form is decisive, CGD-PD maps that decision back through the negation constraint and returns FALSE. This illustrates how asymmetric evidence from logically linked probes can correct a single-prompt error.

Table 4. Ablations and negative results. Acc = accuracy; Unk = UNKNOWN rate; Calls/ex = average model calls per example. Validation contains 204 examples in FOLIO v0.0, so `-max_examples` is an upper bound rather than an exact size control.

		Train (500)			Validation (204)		
Model	Method	Acc	Unk	Calls/ex	Acc	Unk	Calls/ex
GPT-5.2	Single	0.632	0.544	1.00	0.637	0.574	1.00
GPT-5.2	gate3	0.632	0.568	3.00	–	–	–
GPT-5.2	CGR3 (3 calls)	0.654	0.454	3.00	–	–	–
GPT-5.2	**CGD-PD (final)**	**0.682**	**0.396**	3.85	**0.681**	**0.539**	4.36
Claude Sonnet 4.5	Single	0.436	0.744	1.00	0.422	0.755	1.00
Claude Sonnet 4.5	**CGD-PD**	**0.490**	**0.546**	4.80	**0.490**	**0.588**	4.91

A simple 3-call gate did not reliably help. The initial gate-style variant (gate3) matched single-shot accuracy on GPT-5.2 (0.632) while increasing UNKNOWN (0.544 $\rightarrow$ 0.568) at three times the call budget. This suggests that spreading compute across probes is insufficient; the integration rule matters.

Redundancy helps only when decoding respects logical structure. A three-call variant (CGR3) improved GPT-5.2 train accuracy (0.632 $\rightarrow$ 0.654) and reduced UNKNOWN (0.544 $\rightarrow$ 0.454), but the gains were sensitive to the resolution rule. This motivated the final CGD-PD decoder, which exploits asymmetries between H and $\neg H$ while remaining conservative when neither side is supported.

Global pressure against UNKNOWN is brittle. Across early iterations, penalizing UNKNOWN in the prompt did not consistently reduce UNKNOWN and sometimes introduced errors on gold-UNKNOWN examples. CGD-PD instead applies additional pressure selectively, only when the probe pair provides evidence favoring one side.