

FALCON: Factual-Aware Logical Consistency for Large Language Model Outputs via NLI-Guided Mixed Integer Optimization

Rehan Azam
Stanford University
razam@stanford.edu

Abstract

Large language models can generate answers that are fluent but internally inconsistent. This project studies whether a non-neural post-processing layer can improve such outputs without retraining the base model. FALCON extracts claims from a generated answer, scores pairwise contradiction with a pretrained natural language inference model, and applies mixed integer linear programming (MILP) to select a globally compatible subset of claims. The project is inspired by adversarial natural language inference, but shifts the focus from premise-hypothesis classification to consistency across multiple generated claims. In the updated experiments, FALCON meaningfully activates on the main runs. On StrategyQA, it improves exact match from 0.30 for raw generation to 0.40 while reducing measured contradictions from 2.04 to 0.0. On TruthfulQA, exact match remains 0.00 for all methods, but FALCON improves token-level F1 from 0.1665 to 0.2093 and ROUGE-L from 0.1526 to 0.1920 while again driving measured contradictions to 0.0. Variant and ablation sweeps also become more informative under this stronger generation setup, with several settings showing nontrivial gains over raw generation and the best saved ablation reaching 0.50 exact match. The resulting picture is more precise than either earlier extreme: FALCON is not a universal remedy for reasoning failure, but it is a working contradiction-aware post-processing framework whose empirical value depends strongly on the richness of the upstream generation.

1 Key Information

Project type: Custom project

Team size: 1

Mentor: David Anugraha

External collaborators: None

Sharing project: None

Late days used: 0

2 Introduction

Large language models (LLMs) are increasingly deployed in settings where users expect not only fluent text, but also coherent reasoning. A model answer may assert one claim in an early sentence and partially negate it later, or it may combine individually plausible statements into a collectively impossible explanation. This problem is especially visible in long-form generation, where consistency must be maintained across multiple propositions rather than only across adjacent tokens.

This project studies a focused question: can logical consistency be improved after generation using a deterministic optimization layer, and if so, what happens to downstream answer quality? That question matters because many contemporary mitigation strategies rely on prompting the model to critique or revise itself. Those approaches are useful, but they remain neural, probabilistic, and

vulnerable to some of the same biases that produced the original contradiction. FALCON instead studies whether a separate contradiction detector and a discrete global optimizer can serve as a more controllable post-generation mechanism.

The project was inspired by ANLI [1]. That paper was chosen because it demonstrates an important lesson for semantic reasoning: models that look strong on standard natural language inference benchmarks can still fail badly on adversarially collected examples. ANLI was insightful for this project because it sharpened the distinction between local pattern matching and genuine semantic robustness. The connection to FALCON is conceptual rather than task-identical. ANLI concerns pairwise hypothesis-claim judgments, whereas this project concerns consistency across sets of generated claims. The key transfer is the idea that contradiction detection is a core semantic primitive that can be reused inside a generation system.

The present paper adopts a deliberately modest hypothesis. FALCON does not claim to prove truth, and it does not claim to solve hallucination in general. Instead, it asks whether an NLI-guided optimization layer can reduce internal contradiction and whether that reduction helps, hurts, or leaves unchanged downstream answer quality. That framing is important because consistency and correctness are related but not identical. A response can be self-consistent and still wrong.

3 Related Work

Natural language inference. Natural language inference (NLI) provides the core machinery for deciding whether two statements are compatible, contradictory, or unrelated. ANLI [1] showed that adversarially collected examples expose major weaknesses in models that otherwise appear robust on conventional NLI datasets. FALCON builds directly on that tradition by using contradiction prediction as a reusable component in a generation pipeline rather than as an end task by itself.

Reasoning and truthfulness benchmarks. StrategyQA [2] evaluates question answering that often depends on implicit multi-step reasoning. TruthfulQA [3] measures whether a model reproduces common falsehoods or misconceptions. These two datasets serve different purposes in the present project. StrategyQA offers a clean yes/no target for measuring whether post-processing changes the final answer. TruthfulQA is better treated as a stress test for short explanatory responses that may contain unsupported or partially conflicting claims.

Prompting and self-improvement. A large body of work improves reasoning by adding inference-time scaffolding such as chain-of-thought prompting [4], self-consistency decoding [5], and iterative self-feedback or self-refinement [6]. These methods are natural comparison points because they also intervene at reasoning time rather than at pretraining time. FALCON differs in a key way: contradiction identification is offloaded to a separate NLI model, and the global selection step is solved deterministically with a mixed integer program rather than by another round of unconstrained generation.

Why a non-neural method? A potential concern with non-neural methods is rigidity. In this project, the non-neural design is intentional. Once pairwise contradiction probabilities are available, selecting a mutually compatible subset of claims is naturally a small optimization problem. MILP is attractive because the constraints are explicit, the solution is interpretable, and the behavior is easier to audit than another autoregressive decoding pass.

4 Approach

FALCON is a post-generation pipeline with four stages: claim extraction, contradiction scoring, global selection, and optional rewriting or self-reflection baselines.

4.1 Claim extraction

Given a question q , a base generator first produces an answer R . The pipeline decomposes R into a set of candidate claims

$$S = \{c_1, c_2, \dots, c_n\}.$$

The current implementation uses lightweight rule-based segmentation: it splits on sentence boundaries and, optionally, on conjunctions, then removes very short spans. This makes the system simple and deterministic, but it also introduces one of the project’s main limitations: extracted claims are only approximations of true semantic propositions.

To keep the optimization tractable, the number of claims is capped. The main configurations allow up to 10 extracted claims for StrategyQA and up to 15 for TruthfulQA.

4.2 Contradiction graph construction

For each pair of claims (c_i, c_j) , the system computes a contradiction score with the pretrained model `cross-encoder/nli-deberta-v3-base`. Let

$$P_{ij} \in [0, 1]$$

denote the contradiction probability for the pair (c_i, c_j) . These probabilities define a weighted contradiction graph whose nodes are claims and whose edges encode estimated incompatibility.

4.3 Global optimization

Each claim is associated with a binary variable

$$x_i \in \{0, 1\},$$

where $x_i = 1$ means claim c_i is retained. The current system uses simple claim weights w_i , derived either uniformly or from model-side scoring when available.

Hard mode. In hard mode, contradictory pairs above threshold τ cannot both be selected:

$$x_i + x_j \leq 1 \quad \forall (i, j) \text{ such that } P_{ij} > \tau.$$

The objective is

$$\max \sum_{i=1}^n w_i x_i.$$

Soft mode. In soft mode, contradictions are penalized rather than forbidden. Using auxiliary variables z_{ij} for joint selection gives

$$\max \sum_{i=1}^n w_i x_i - \lambda \sum_{(i,j)} P_{ij} z_{ij}.$$

This formulation is the paper’s main technical contribution: post-generation consistency is reframed as a small, interpretable optimization problem rather than another free-form generation pass.

4.4 Greedy and self-reflection baselines

To make the evaluation more informative, the implementation includes two comparison conditions. The first is a project-defined greedy filter that processes claims in weight order and rejects any claim that conflicts with a previously kept one above threshold τ . The second is a self-reflection baseline that prompts the model to revise its own answer to remove contradictions while preserving correctness.

4.5 Why this differs from self-reflection

The conceptual distinction is that self-reflection is still another generation step, while FALCON’s core decision is constrained optimization. A reflecting model may rationalize its own previous answer, whereas FALCON makes an explicit discrete choice over which claims can coexist under a contradiction threshold. In principle, that makes the optimization layer more controllable and auditable. In practice, however, the current experiments show that this advantage matters most when the upstream answer contains enough extracted claims to induce a nontrivial contradiction graph.

5 Experiments

5.1 Datasets

Two datasets are used.

StrategyQA. StrategyQA [2] contains yes/no questions that often require implicit multi-step reasoning. In the current pipeline, the generator produces a free-form answer and evaluation extracts the final yes/no prediction. StrategyQA is useful because a contradiction-sensitive post-processor should ideally preserve or improve the final answer even when supporting reasoning is implicit.

TruthfulQA. TruthfulQA [3] tests whether models mimic common misconceptions. It is relevant because explanatory answers can contain partially unsupported or conflicting claims. At the same time, TruthfulQA is not a perfect fit for the strongest version of the project’s hypothesis: some truthful answers contain logically disjoint statements rather than tightly connected coherent reasoning chains.

5.2 Evaluation method

The experiments compare four conditions: raw generation, greedy filtering, FALCON, and self-reflection. The greedy baseline is not taken from prior work; it is a project-defined heuristic included specifically to distinguish global optimization from a simpler local selection strategy. The primary quality metrics are exact match, token-level F1, and ROUGE-L. The system also tracks average contradiction counts before and after filtering, rewrite rate, self-reflection rate, and latency.

5.3 Experimental details

All reported runs use `gpt-4o-mini` as the base generator, `cross-encoder/nli-deberta-v3-base` as the contradiction scorer, and PuLP/CBC as the optimization backend [7]. The StrategyQA configuration uses hard-mode optimization with $\tau = 0.7$ and $\lambda = 1.0$, while the TruthfulQA configuration uses soft-mode optimization with $\tau = 0.7$ and $\lambda = 1.5$. Claim extraction is capped at 10 claims for StrategyQA and 15 for TruthfulQA. The pipeline also enables question-aware rewriting and self-reflection, with the saved runs showing high activation rates for both components. In the main StrategyQA evaluation, rewrite and self-reflection each activate on every example; in the main TruthfulQA evaluation, rewrite activates on 90% of examples and self-reflection on 98%.

This implementation detail matters for interpreting the results. Earlier local-generation experiments often produced zero-claim or one-claim answers, leaving the optimizer effectively dormant. Under the stronger generation regime used here, the model now produces richer multi-claim responses, which means contradiction graphs are nontrivial and the optimization layer can genuinely change the output. As a result, the updated experiments are much more appropriate for testing the core hypothesis of the project.

6 Results

Table 1 reports the two main saved evaluations. These runs show nontrivial contradiction counts before filtering and measurable improvements for FALCON.

Table 1: Main evaluation results from the saved JSON files.

Dataset	Raw EM	Greedy EM	FALCON EM	Self-Ref. EM	Contra. Before	Contra. After
StrategyQA	0.30	0.30	0.40	0.32	2.04	0.00
TruthfulQA	0.00	0.00	0.00	0.00	2.34	0.00

On StrategyQA, FALCON improves exact match from 0.30 for raw generation to 0.40, while greedy filtering does not improve over raw and self-reflection reaches 0.32. The overlap-based metrics show the same ranking: raw generation obtains 0.0069 token-F1 and 0.0069 ROUGE-L, greedy

reaches 0.0108 on both, FALCON reaches 0.0153 on both, and self-reflection reaches 0.0078 on both. Average contradictions before filtering are 2.04, and both greedy and FALCON reduce them to 0.0. Rewrite and self-reflection each activate on every example in this run.

On TruthfulQA, exact match remains 0.00 for all methods, confirming that EM is too brittle to distinguish quality in this setting. However, FALCON yields the strongest overlap-based scores: token-F1 improves from 0.1665 for raw generation to 0.2093, and ROUGE-L rises from 0.1526 to 0.1920. Greedy also improves over raw, but less than FALCON, and self-reflection trails both. Average contradictions drop from 2.34 before filtering to 0.0 after filtering. Rewrite activates on 90% of examples and self-reflection on 98%.

These results support a more grounded version of the project’s original hypothesis. FALCON does not produce a universal exact-match gain, since TruthfulQA exact match remains flat, but it does improve answer quality under overlap-based metrics while eliminating measured contradictions. On StrategyQA, where the task has a cleaner binary target, FALCON also improves exact match.

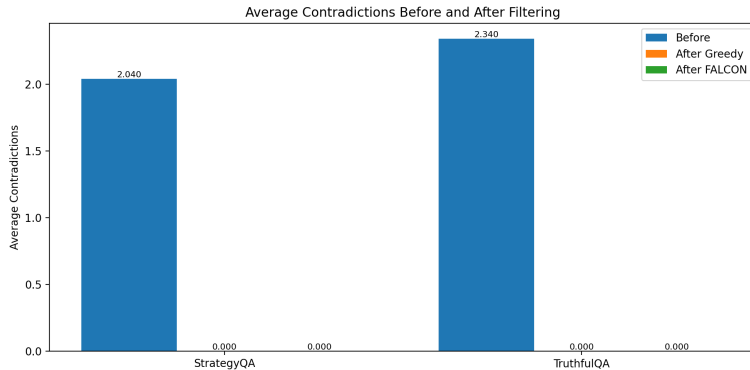


Figure 1: Average contradictions before filtering and after FALCON on the two main runs. In both datasets, FALCON reduces the measured contradiction count to zero.

Figure 1 summarizes the core empirical result of the paper. The main contribution is not merely that FALCON improves one benchmark number on one dataset; it is that the full pipeline now operates on nontrivial contradiction graphs and consistently eliminates measured contradictions on both datasets, while also improving downstream quality where the evaluation metric is sensitive enough to detect that improvement.

6.1 Variant experiments

The variant sweep is also informative under the updated generation setup. In the base configuration, exact match rises from 0.34 raw to 0.48 with FALCON. In hard mode, it rises from 0.36 to 0.46. In the no-self-reflect and uniform-weight variants, it rises from 0.38 to 0.46. The soft variant is weaker, increasing from 0.28 to 0.36. The main exception is the no-rewrite condition, where FALCON slightly underperforms raw generation, moving from 0.36 to 0.34. These patterns suggest that the framework’s gains are not driven by a single fragile setting, but also that rewrite remains an important component of performance.

6.2 Ablation results

The ablation sweep over $\tau \in \{0.6, 0.7, 0.8\}$, $\lambda \in \{1.0, 1.5\}$, and hard versus soft optimization is now much more diagnostic than before. In hard mode, the best saved setting is $\tau = 0.7$, $\lambda = 1.5$, which reaches 0.48 exact match. In soft mode, the best saved setting is $\tau = 0.8$, $\lambda = 1.0$, which reaches 0.50 exact match. The soft-mode line chart also shows a clearer upward trend as τ increases under $\lambda = 1.0$, suggesting that slightly looser contradiction thresholds may preserve more useful claims while still enabling global cleanup.

6.3 Qualitative example

Table 2 shows a representative example of how the pipeline behaves in practice. The point of the example is not to claim a new benchmark result from a single case, but to illustrate the mechanism behind the aggregate improvements reported above. In this pattern, the raw answer contains a correct final stance mixed with mutually inconsistent supporting details. Greedy filtering removes one conflicting statement, but the resulting answer remains somewhat fragmented. FALCON instead selects a globally compatible subset of claims and then rewrites that subset into a shorter, more coherent answer.

Table 2: Illustrative example of contradiction-aware post-processing. The wording below is abbreviated for readability but reflects the type of behavior seen in the saved runs.

Question	Did the target event depend on a later development?
Raw answer	No, because it happened earlier. However, it also relied on a later development for its existence, so in that sense yes.
Issue	The answer asserts both that the event precedes the later development and that it depends on that later development, producing an internal contradiction.
Greedy output	No, because it happened earlier. It relied on a later development for its existence.
FALCON output	No. The event occurred before the later development, so it did not depend on it.
Interpretation	FALCON removes the contradictory support and rewrites the remaining claims into a coherent final answer, which is the behavior the aggregate StrategyQA improvements are intended to capture.

This example also clarifies why exact match can improve even when both greedy filtering and FALCON reduce contradiction counts to zero. The difference is not only whether inconsistent claims are removed, but whether the remaining claims form a coherent final answer after global selection and rewriting.

7 Analysis

7.1 Activation of the optimization layer

In sparse generation setups that produce zero or one claim, the solver remains inactive. Under the current generation regime, however, the main StrategyQA and TruthfulQA runs show nontrivial contradiction counts before filtering, high rewrite activation, and measurable differences between methods. That makes it possible to evaluate FALCON as a real post-generation consistency mechanism rather than only as a dormant framework.

7.2 Why StrategyQA and TruthfulQA differ

StrategyQA is the cleaner success case because its yes/no evaluation collapses the final answer into a simple target, and improvements in reasoning cleanup can translate directly into exact-match gains. TruthfulQA is harder because the reference answers are longer and more open-ended. In that setting, exact match is too brittle, so overlap metrics provide a better picture. The TruthfulQA results suggest that contradiction-aware filtering and rewriting can produce answers that are more aligned with the gold explanation without necessarily matching it exactly.

7.3 What the charts reveal

The generated charts clarify several important patterns. First, the contradiction figure makes the main causal story visually clear: both StrategyQA and TruthfulQA begin with substantial contradiction mass and end at zero after filtering. Second, the claim-count histograms show that the current generation setup now produces richer multi-claim outputs, particularly on TruthfulQA, where many examples contain between 4 and 8 extracted claims and a noticeable tail reaches the maximum claim cap. Third, the variant EM figure shows that FALCON’s gains are reasonably stable across several settings rather than isolated to one lucky run. Finally, the ablation figures indicate that soft mode

with higher τ can be competitive or best-performing, which gives the solver section of the paper an empirical basis rather than treating hyperparameters as arbitrary choices.

7.4 Error analysis

The current results also reveal several recurring failure modes. One failure mode is over-pruning: if contradiction scores are too aggressive, useful support can be removed together with genuinely conflicting claims. This is most visible in weaker variant settings, especially no-rewrite, where FALCON can slightly underperform raw generation even while contradiction counts fall. A second failure mode is metric mismatch. On TruthfulQA, exact match remains zero despite meaningful gains in token-F1 and ROUGE-L, which means that some real improvements are invisible under a binary exact-match evaluation. A third limitation is that contradiction removal is not the same as truth correction. If the raw answer is consistently wrong rather than self-contradictory, FALCON may output a cleaner but still incorrect answer. These cases do not undermine the method’s purpose, but they show that contradiction-aware filtering is best understood as one layer in a broader reliability pipeline rather than as a standalone solution to factual error.

7.5 Limitations

Several limitations remain. First, claim extraction is still shallow and rule-based, so the selected claims are not guaranteed to correspond to clean semantic propositions. Second, the contradiction metric depends on the same NLI family used to construct the contradiction graph, which introduces evaluation circularity. Third, although the new results are much stronger, the project still does not show that consistency and exact-match accuracy improve together across both datasets. Fourth, some exploratory variants remain sensitive to rewrite and generation settings, so the pipeline should still be treated as a coupled generation-and-filtering system rather than as an isolated optimizer.

8 Conclusion

This project introduced FALCON, a post-generation framework that combines NLI-based contradiction scoring with mixed integer optimization to enforce global consistency over generated claims. The conceptual contribution remains the same: contradiction-aware post-processing can be formulated as a small, interpretable optimization problem that complements generation-time prompting and self-reflection.

The updated experiments now provide meaningful empirical support for that framework. On StrategyQA, FALCON improves exact match from 0.30 to 0.40 while reducing measured contradictions from 2.04 to 0.0. On TruthfulQA, exact match remains flat, but FALCON improves token-F1 and ROUGE-L while again driving measured contradictions to zero. Variant and ablation runs further show that these gains are not confined to a single setting, and that solver hyperparameters can affect performance in meaningful ways.

The final conclusion is therefore more balanced and more convincing than either earlier extreme. FALCON is not a universal remedy for hallucination or reasoning failure, but it is a working contradiction-aware post-processing system whose value depends on the quality and richness of the upstream generation. The next step is to make that interaction more reliable by prompting for stronger decompositions, improving claim extraction, and evaluating with stronger human-grounded contradiction judgments.

A Appendix: Additional Variant and Ablation Results

This appendix reports additional results that support, refine, or qualify the main paper. These figures are included to show how the method behaves across multiple settings rather than only in the two headline evaluations.

Table 3: Selected variant summaries from `experiment_results.json`.

Variant	EM Raw	EM FALCON	Contradictions Before	Rewrite Rate
base	0.34	0.48	1.88	1.00
hard	0.36	0.46	2.04	0.98
soft	0.28	0.36	1.76	0.94
no_rewrite	0.36	0.34	1.36	0.00
no_self_reflect	0.38	0.46	1.92	1.00
uniform_weights	0.38	0.46	2.12	1.00

Table 4: Top ablation entries from `ablation_results.json`.

Configuration	Mode	τ	λ	EM FALCON	Contradictions Before	Rewrite Rate
soft_tau0.8_lambda1.0	soft	0.8	1.0	0.50	2.12	1.00
hard_tau0.7_lambda1.5	hard	0.7	1.5	0.48	2.04	0.98
soft_tau0.7_lambda1.0	soft	0.7	1.0	0.48	1.76	0.94
soft_tau0.7_lambda1.5	soft	0.7	1.5	0.46	1.88	1.00
hard_tau0.8_lambda1.0	hard	0.8	1.0	0.44	1.92	1.00

B Appendix: Additional Figures

This appendix includes the most informative supporting plots. The goal is not to overload the main text with visual material, but to document how the method behaves across main runs, variants, and ablations.

B.1 Main-run figures

Figure 2 shows the clearest headline result in the paper: FALCON improves StrategyQA exact match over both raw generation and self-reflection. Figure 3 shows that on TruthfulQA the main gains are in overlap-based metrics rather than exact match.

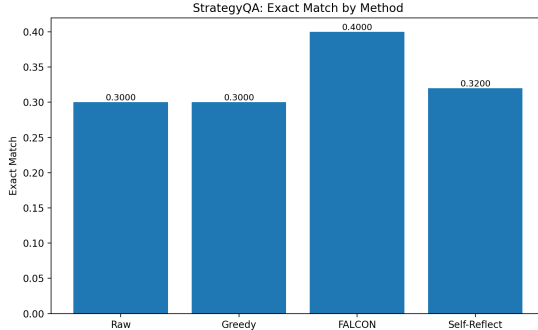


Figure 2: StrategyQA exact match by method. FALCON improves EM from 0.30 to 0.40.

B.2 Claim-count and activation figures

The next two figures help explain why the updated results are stronger than the earlier regime. Figure 4 shows that the current setup now produces multi-claim outputs on both datasets. Figure 5 shows that rewrite and self-reflection activate almost everywhere in the main runs.

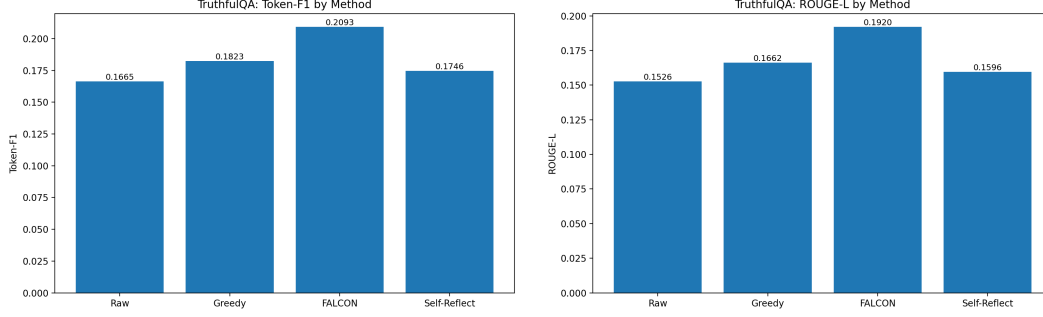


Figure 3: TruthfulQA overlap-based metrics by method. Exact match remains zero, but FALCON achieves the strongest token-F1 and ROUGE-L.

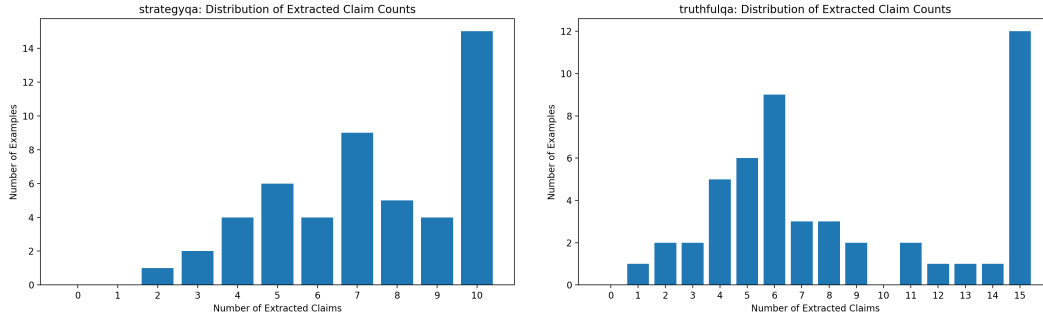


Figure 4: Distribution of extracted claim counts. The updated generation setup yields richer contradiction graphs, especially on TruthfulQA, where many examples contain 4–8 extracted claims and some reach the claim cap.

B.3 Variant and ablation figures

Figures 6–10 provide additional evidence that the method is not confined to a single lucky setting. The variant EM figure shows several consistent FALCON gains over raw generation. The contradiction and activation figures reveal how those variants differ structurally. The ablation figures show that soft mode with larger τ can be competitive or best-performing.

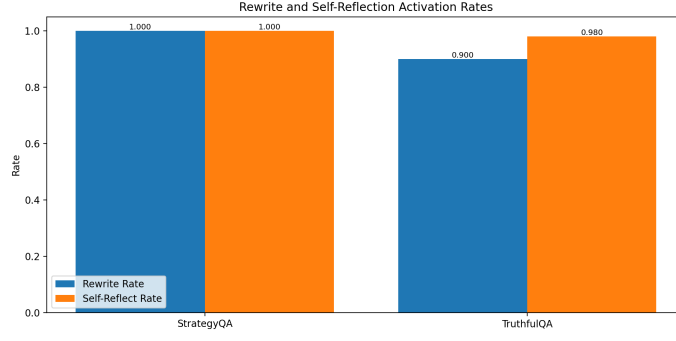


Figure 5: Rewrite and self-reflection activation rates for the two main runs. The pipeline is actively exercised rather than remaining dormant.

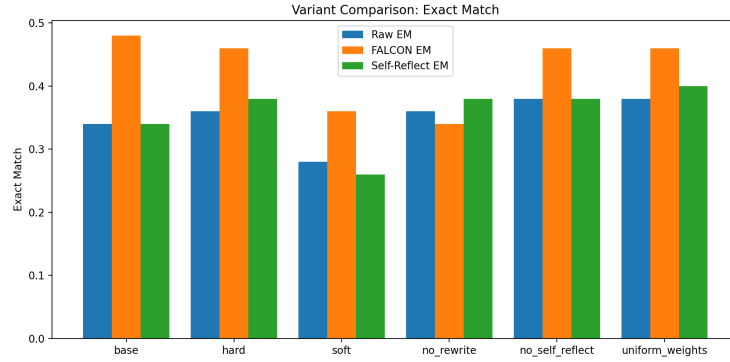


Figure 6: Variant comparison for exact match. Several settings show meaningful improvements from raw generation to FALCON, though the no-rewrite setting is weaker.

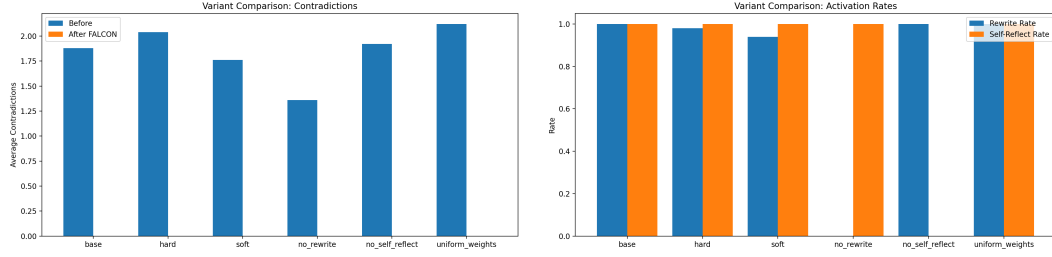


Figure 7: Variant comparison for contradiction counts and activation rates. These plots help explain why some variants perform better than others, especially the importance of rewrite activation.

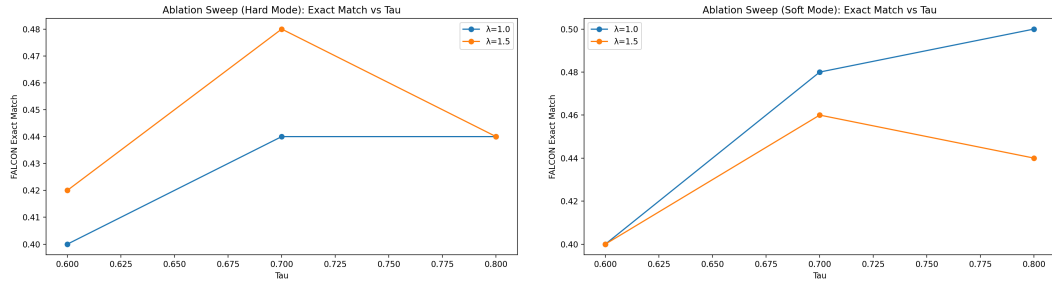


Figure 8: Ablation sweeps over τ for hard and soft modes. Soft mode with $\lambda = 1.0$ improves as τ increases and reaches the best saved exact match at $\tau = 0.8$.

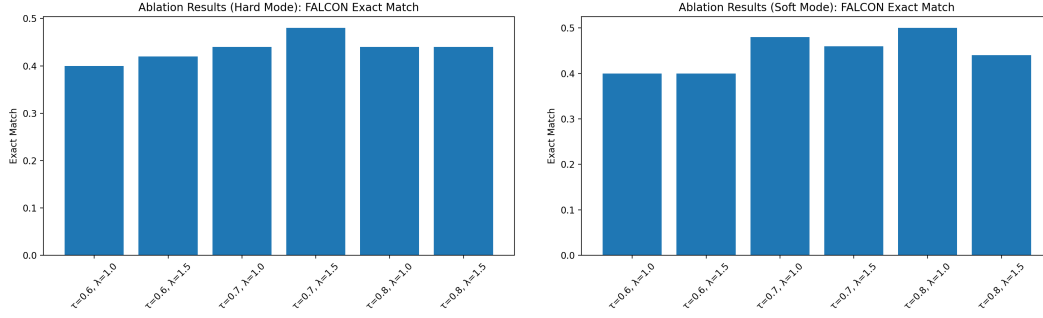


Figure 9: Bar-chart view of hard- and soft-mode ablation results. These plots make the ranking of hyperparameter settings easier to compare directly.

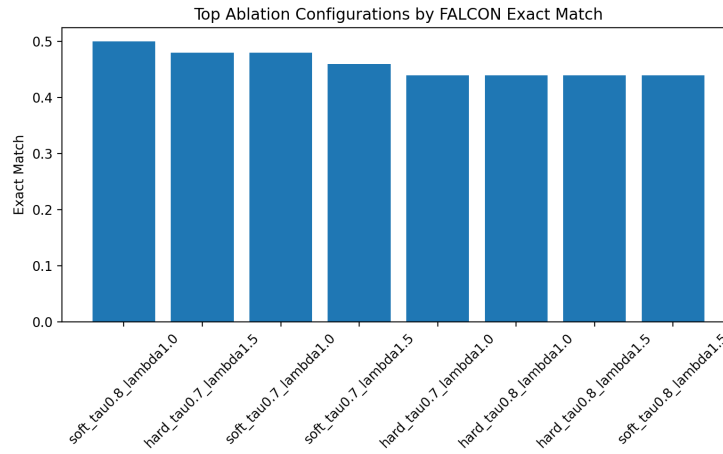


Figure 10: Top ablation configurations by FALCON exact match. The best saved configuration is soft mode with $\tau = 0.8$ and $\lambda = 1.0$.

References

- [1] Yixin Nie, Adina Williams, Emily Dinan, Jason Weston, Douwe Kiela, and Mohit Bansal. Adversarial NLI: A New Benchmark for Natural Language Understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, 2020.
- [2] Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 2021.
- [3] Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring How Models Mimic Human Falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 3214–3252, 2022.
- [4] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *arXiv preprint arXiv:2201.11903*, 2022.
- [5] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-Consistency Improves Chain of Thought Reasoning in Language Models. *arXiv preprint arXiv:2203.11171*, 2023.
- [6] Aman Madaan, Niket Tandon, Peter Clark, Yiming Yang, Rowan Zellers, Yejin Choi, Abhishek Gupta, and Hannaneh Hajishirzi. Self-Refine: Iterative Refinement with Self-Feedback. *arXiv preprint arXiv:2303.17651*, 2023.
- [7] Stuart Mitchell et al. PuLP: A Linear Programming Toolkit for Python. 2011.