

NLI under the Microscope: What Atomic Hypothesis Decomposition Reveals

Neha Srikanth

University of Maryland, College Park
nehasrik@umd.edu

Rachel Rudinger

University of Maryland, College Park
rudinger@umd.edu

Abstract

Decomposition of text into atomic propositions is a flexible framework allowing for the closer inspection of input and output text. We use atomic decomposition of hypotheses in two natural language reasoning tasks, traditional NLI and defeasible NLI, to form *atomic sub-problems*, or granular inferences that models must weigh when solving the overall problem. These atomic sub-problems serve as a tool to further understand the structure of both NLI and defeasible reasoning, probe a model’s consistency and understanding of different inferences, and measure the diversity of examples in benchmark datasets. Our results indicate that LLMs still struggle with logical consistency on atomic NLI and defeasible NLI sub-problems. Lastly, we identify *critical atomic sub-problems* of defeasible NLI examples, or those that most contribute to the overall label, and propose a method to measure the *inferential consistency* of a model, a metric designed to capture the degree to which a model makes consistently correct or incorrect predictions about the same fact under different contexts.

1 Introduction

Atomic decomposition involves breaking sentences down into *atomic propositions*, or granular facts that are explicitly supported by the original text. This style of decomposition has widespread applications, including assessing the factual precision of generated text (Min et al., 2023), claim verification (Chen et al., 2024), and multihop QA (Perez et al., 2020), since it allows for careful, finer-grained inspection of text.

We use atomic decomposition as tool to dive deeper into two types of natural language reasoning: traditional NLI (Giampiccolo et al., 2007) and defeasible inference (Rudinger et al., 2020), a mode of reasoning where inferences may change in light of new evidence. In both tasks, atomic decomposition of hypotheses into *atoms* breaks complex

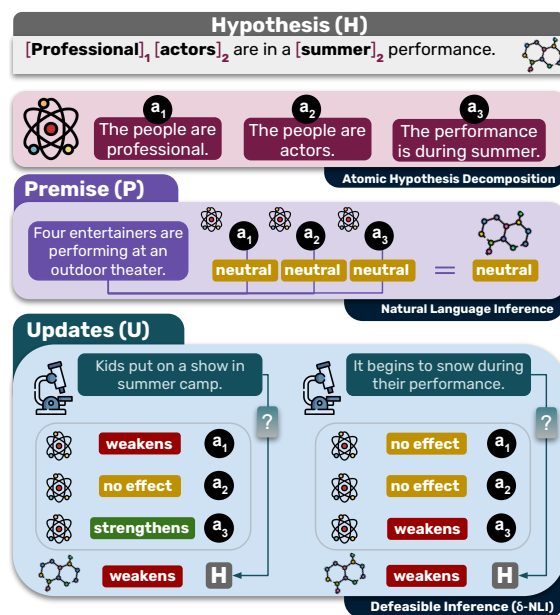


Figure 1: Top: Atomic hypothesis decomposition breaks down hypotheses (H) into entailed propositional “atoms” ($a_1 - a_3$). Middle: Pairing the premise (P) with each atom yields a set of NLI sub-problems ($P + a$); the sub-problem labels predict the full NLI problem ($P + H$) label. Bottom: Paired with an update (U), each atom yields a defeasible NLI sub-problem ($P + a + U$); the set of sub-problem labels are predictive of the full problem ($P + H + U$) label, but the non-monotonic relationship is more complex than for traditional NLI.

sentences into granular pieces of information that models must weigh when drawing higher level inferences, producing *atomic sub-problems*. We use these sub-problems not only for better insight into the structure and nuances of NLI and defeasible NLI, but also to assess accompanying benchmarks and to more deeply probe the robustness of models’ situational understanding.¹

Consider the example in Figure 1 of a premise and hypothesis with a neutral relationship. When predicting an overall neutral relation, a model must

¹Code and data available at <https://github.com/nehasrikn/nli-atoms>.

determine the relation between the premise and each of the three atomic propositions that together form the hypothesis — in this case, all of which are neutral. If the model predicts an entailment or a contradiction between the premise and one of the atoms, its understanding of the situation may be called into question.

We first present our two tasks of interest (§2) and discuss the utility of atomic sub-problems and their construction (§3). Then, we analyze the behavior of large language models (LLMs) on atomic sub-problems in traditional NLI (§4) where we evaluate their logical consistency between each original SNLI instance and its corresponding sub-problems. We find that despite high accuracy, LLMs still struggle with logical consistency. Then, we study atomic sub-problems in defeasible NLI (§5) and propose a framework to pinpoint the inference(s) evaluated in each example by way of the *question under discussion* of examples, a well-studied linguistic phenomenon (Benz and Jasinskaja, 2017; Wu et al., 2023). Finally, we present a method to group defeasible NLI examples based on related atomic sub-problems (§6) and measure the *inferential consistency* of a model, a metric capturing the likelihood that its prediction for a particular inference will remain **consistently** correct or incorrect under different contexts.

2 Background

2.1 Natural Language Inference

Natural language inference (Giampiccolo et al., 2007; MacCartney, 2009; Bowman et al., 2015) is the task of determining whether a premise P entails, contradicts, or is neutral in relation to a hypothesis, H . For example, the premise “A little girl in a lush green field” contradicts the hypothesis “A little girl rides her ox in a desert”, as “desert” directly contradicts the “lush green field”.

SNLI. The first large-scale NLI dataset, SNLI, uses image captions as premises paired with hypotheses elicited from crowdworkers. Though this benchmark has been well-saturated by LLMs over the past few years, it serves as a useful resource for studies in model robustness (Srikanth and Rudinger, 2022; Kaushik et al., 2019) and annotation artifacts (Gururangan et al., 2018).

2.2 Defeasible Inference

Defeasible reasoning is a form of non-monotonic reasoning in which inferences may be altered in

light of new evidence (Reiter, 1980). For example, given the premise “A group of people sitting around a rectangular table”, the inference “they have a work meeting” is weakened upon learning that “the people are toddlers.”

δ -NLI. Rudinger et al. (2020) introduce the task of defeasible natural language inference and an accompanying benchmark, δ -NLI. Given a P and H pair with a *neutral* relation, a third **update** sentence U *strengthens* H if, upon reading U , H is more likely to be true, and *weakens* H if H is less likely to be true upon reading U . Defeasible NLI is then a binary classification task of predicting a strengthener or weakener label for a (P, H, U) set. δ -NLI is built on top of three popular commonsense reasoning datasets: SNLI (Bowman et al., 2015), ATOMIC (Sap et al., 2019), and SOCIAL-CHEM-101 (Forbes et al., 2020). For a P and H pair (or, in the case of SOCIAL-CHEM-101, just H), crowdworkers write an update sentence for a target label, but are not instructed to target a particular part of H when doing so. The authors ensure that the train, development, and test splits of the data are split at the P - H level to avoid leakage. Crowdworkers may not write updates that directly contradict information in the premise. For simplicity, we focus on the SNLI-derived split of δ -NLI, or δ -SNLI, which selects neutral P - H SNLI pairs.

2.3 Related Work

Atomic decomposition has been used in fact checking (Min et al., 2023; Glover et al., 2022; Yuan and Vlachos, 2024), claim verification (Chen et al., 2024), summarization (Nenkova and Passonneau, 2004), and text-to-image generation (Cho et al., 2023) among others. Kamoi et al. (2023) construct a dataset of claims and sub-claims for claim checking where sub-claims, analogous to our atoms, are labeled with respect to evidence. Most relevant to our work, Stacey et al. (2022) train a span-based NLI model to make *span-level* decisions on SNLI and SICK examples that are composed to produce an overall label. In contrast, our work measures the logical consistency of LLMs that may have seen NLI data during pretraining, but that are not explicitly trained to weigh atomic inferences. Their followup (Stacey et al., 2024) *trains* another NLI system using LLM-generated atoms, however their study focuses primarily on *premise* decomposition. To the best of our knowledge, our study is the first to explore atoms in defeasible inference.

3 Atomic Sub-Problems in NLI

Reasoning about situations often involves weighing multiple pieces of information to draw inferences. Consider the last row in Table 1. A human determining that H contradicts P will attribute the contradiction to the fact that P mentions a father and daughter, but H mentions two men. Implicitly, they will have also weighed the fact that H ’s mention of “cutting grass” is entailed by P ’s mention of a lawnmower, and so it does not contribute to the contradiction. We treat these two determinations as distinct *atomic sub-problems*.

Hypotheses in SNLI, and in turn, δ -SNLI, can be complex sentences, and while solving inference problems, models must weigh all pieces of information in both P and H . We expect humans to make inferences about constituent pieces of information in a manner that is consistent with their overall judgment, an equally desirable property in models. Not only does it signal holistic understanding of the situation described in the problem, but it can help pinpoint exactly what types of inferences models struggle with. Identifying atomic sub-problems also allows us to understand the granular inferences that are evaluated in benchmark datasets. In turn, this helps to understand the *diversity* in the dataset: despite there being thousands of examples, certain inferences may come up repeatedly.

To identify the constituent sub-problems, we break *hypotheses* in SNLI and δ -SNLI into atomic propositions (Wanner et al., 2024) to use in subsequent analyses. Each atomic decomposition represents a single piece of information. Formally, given an SNLI example with P and H , we generate atomic decompositions of H represented by $a_1 \dots a_n$. Each atomic sub-problem then involves predicting the relation between a (P, a_i) tuple (§4.1). Given a δ -NLI example with P , H , and U , atomic sub-problems involve a (P, a_i, U) tuple where the task is to determine whether U strengthens, weakens, or has no effect on a_i (§5).

3.1 Generating Atomic Propositions

To generate atomic propositions, we draw on Neo-Davidsonian event-based semantic representations of sentences (Castañeda, 1967; Parsons, 1990). Sentences can be represented in first-order logical form as conjunctions of predicates representing entities, where actions are explicitly represented with event variables and predicate arguments are mapped to semantic roles (Dowty, 1991). For ex-

ample, the sentence “*The juggler performs at a party*” could be represented as:

$$\begin{aligned} \exists x_1 \exists e \text{ (Juggler}(x_1) \wedge \text{Perform}(e) \wedge \text{Agent}(e, x_1) \wedge \\ \exists x_2 \text{ (Party}(x_2) \wedge \text{At}(e, x_2))) \end{aligned}$$

Each conjunct can then be mapped to a natural language expression, called an *atom*. This ensures that both arguments of actions *and* the actions themselves are included as separate atoms.

We draw on this intuition to carefully hand-construct exemplars, a methodology shown to improve the atomicity and groundedness of decompositions (Wanner et al., 2024). We prompt llama-3-8b-instruct with these exemplars (Appendix A) to generate atoms for each example in the δ -SNLI test set (henceforth, δ -SNLI-TEST), as well as for 1000 randomly sampled examples in the SNLI test set (SNLI-TEST-SAMPLE). See Table 5 for dataset statistics.

3.2 Validating Atomic Decompositions

Valid atomic decompositions of hypotheses must be logically entailed from the hypothesis they were decomposed from. For our experiments on SNLI-TEST-SAMPLE (§4), we do not validate atom entailment ourselves, letting each model determine whether H entails each a_i itself (§4.1), and only measuring consistency on the atomic sub-problems that the model itself admits as “valid”.

However, we *do* validate all generated atoms in δ -SNLI-TEST, since non-monotonic reasoning does not give rise to clear constraints between an original problem and its constituent atomic sub-problems. Our two-step validation process involves pruning decompositions with a strong, finetuned NLI model followed by human validation.

Pruning. For each example in δ -SNLI-TEST, we use a DEBERTA-large model finetuned on popular NLI datasets² and remove all generated atoms that are not entailed by the hypothesis. By design, P - H pairs in δ -SNLI have a neutral relation, and updates strengthen or weaken propositions in the *hypothesis*. Hence, we run a secondary pruning stage to retain only those atoms that are *not* entailed by the premise (see Table 5). See Appendix A for a discussion of coverage.

Human Validation. An author annotated all atoms that survived pruning as either *invalid* or

²MNLI (Williams et al., 2018), Fever-NLI (Thorne et al., 2018), Adversarial NLI (Nie et al., 2020), LingNLI (Parrish et al., 2021), and WANLI (Liu et al., 2022)

Premise (P)	Hypothesis (H)	Update (U)	Atoms
A man in a white t-shirt and jeans is holding a mallet and chisel next to his abstract sculpture which stands on several bricks.	A man is trying to finish his sculpture for a church	The man has taken his first strike against the granite. (Weaker)	a_1 : The thing the person is trying to do is finish. (-1) a_2 : The thing the person is trying to finish is a sculpture. (+1) a_3 : The sculpture is for a church. (0)
There is a green trash truck in road with a person sweeping sidewalk.	The garbage man sweeps up where the can spilled.	The person is wearing a city uniform. (Strengthen)	a_1 : The person is a garbage man. (+2) a_2 : The thing being swept is up. (invalid) a_3 : There is a can. (0) a_4 : The can has spilled. (+1) a_5 : The person is sweeping up a spill. (+2)
Two young men climb a tree overlooking a rural setting, with one of them out far on a limb and clutching a white helmet.	Two brothers are climbing a tree to get down their Frisbee. (N)	N/A	a_1 : There are two people. (E) a_2 : There are two people who are brothers. (N) a_3 : There are people climbing. (E) a_4 : There are people climbing a tree. (E) a_5 : There is a purpose for people climbing a tree. (N) a_6 : The purpose for people climbing a tree is to get something. (N) a_7 : The thing people are trying to get is a Frisbee. (N) a_8 : The thing people are trying to get is down. (invalid)
A father and his daughter are riding a lawn mower down a street while dressed in American colors.	two men cut grass by hand (C)	N/A	a_1 : There are two people. (E) a_2 : There are two people who are men. (C) a_3 : There are people cutting. (E) a_4 : There are people cutting grass. (E) a_5 : There is a method of cutting grass. (E) a_6 : The method of cutting grass is by hand (C)

Table 1: δ -SNLI (Rows 1 and 2) and SNLI (Rows 3 and 4) instances, along with atomic decompositions of hypotheses and the label of the sub-problem involving that atom.

valid (see Table 1 for examples of invalid atoms). Valid δ -SNLI atoms had to (1) be grammatical, (2) entail from H , (3) not entail from P . Atoms introducing new information were considered invalid, including those that were *pragmatic* inferences of H (Jeretic et al., 2020; Srikanth et al., 2024b).

95.7% of pruned atoms were determined as valid by the author annotator. An external annotator also annotated a sample of 100 atoms for validity for an agreement of $\kappa = 0.82$ measured by Cohen’s Kappa (Cohen, 1960). The remaining analysis in this work is done on the set of *valid* δ -SNLI atoms.

4 Atoms in Traditional NLI

While performing strongly on various benchmarks, LLMs still struggle with many types of consistency including paraphrastic consistency (Srikanth et al., 2024a; Verma et al., 2023), hypothetical consistency (Chen et al., 2023), or even preferential consistency (Zhao et al., 2024). When LLMs make entailment judgments, another desirable property is *logical consistency*. Namely, when an LLM itself deems a set of atoms $a_1 \dots a_n$ entailed by H , we can hold it accountable to maintain consistency between its judgments on each (P, a_i) sub-problem and its overall (P, H) judgment in a logical way. This gives us necessary, but not sufficient, evidence to help signal that it has “understood” the situation.

4.1 Atomic and Overall Label Consistency

We construct a set of rules to establish the relationship between atomic sub-problems and overall problem labels.

1. If H is entailed by P : Each valid a_i must be entailed by P .
2. If H contradicts P : At least one valid a_i must contradict P .
3. If H is neutral with respect to P : At least one valid a_i must be neutral with respect to P , all others may be either neutral or entailed.

4.2 Experimental Setup

We experiment with six LLMs: gpt-4o (OpenAI, 2024), gpt-4o-mini (OpenAI, 2024), gpt-3.5-turbo-0125 (Ouyang et al., 2022), llama-3-8b-instruct (Dubey et al., 2024), llama-3-70b-instruct (Dubey et al., 2024), and gemma-2-9b-instruct (Team et al., 2024).

First, we benchmark each models’s performance on SNLI-TEST-SAMPLE original examples (Table 2) using Prompt B.1 adapted from Liu et al. (2023). We use 12 in-context *original* SNLI examples from the dev split evenly distributed over the three NLI labels. Then, for each SNLI-TEST-SAMPLE example, we have each model predict the relation between H and each generated atom a_i from §3.1 using the same prompt and exemplar set.

	Full Example Accuracy	Overall Logical Consistency	Consistency on Correct Exs	Consistency on Incorrect Exs	Logical Consistency by Label			Induced Atom Label Accuracy
					entailment	neutral	contradiction	
gpt-4o-mini-2024-07-18	89.8	84.0	86.8	59.8	78.8	90.1	82.6	81.3
gpt-4o-2024-08-06	88.5	87.9	89.6	74.8	88.9	92.7	79.2	81.9
llama-3-70b-it	87.7	84.2	89.5	46.3	82.1	81.3	89.5	84.8
llama-3-8b-it	85.2	81.2	88.2	41.2	76.0	85.0	85.1	82.9
gemma-2-9b-it	84.2	80.9	85.9	54.4	84.0	82.4	73.8	78.9
gpt-35-turbo-0125	82.1	74.4	80.4	46.9	70.4	90.9	64.2	74.9

Table 2: Accuracy (col. 1) and logical consistency of LLMs full SNLI examples (col. 2). We also report logical consistency on examples where the full prediction was correct (col. 3) and incorrect (col. 4.), as well as stratify by predicted label (cols. 5–7). Finally, we logically compose atomic sub-problem labels to see if this more interpretable method outperforms full example accuracy (col. 8).

For each atom that the LLM predicts as entailed, we have it predict the relation between P and a_i using the same prompt and exemplar set.

We report overall logical consistency as the percent of examples where the full prediction was logically consistent with the predicted labels for sub-problems as dictated by the rules in §4.1.

Results. Despite higher *accuracy* numbers, LLMs seem to struggle with *logical consistency* (Table 2). Interestingly, a model’s accuracy is not fully indicative of its logical consistency. When models incorrectly predict the full example’s label, they are more prone to logical inconsistencies between atomic sub-problems and the full problem (Table 2, Columns 3 and 4). Though not the top performing model, gpt-4o outperformed other models on logical consistency **even on examples where its full prediction was incorrect**. Such logical consistency indirectly captures the *reliability* of a model’s full prediction: when two LLMs achieve similar accuracies, logical consistency serves as another point of comparison.

We also stratify our results by the predicted overall label and report logical consistency within each class (Table 2, Columns 5–7). All models exhibit consistency gaps *within the 3 labels*, and different models struggle with different example classes.

Lastly, we experiment with *atomic inference* (Stacey et al., 2024), or inducing an overall label via logical rules over the predicted atomic sub-problem labels, to understand whether a setting in which models only provide granular inferences can be more effective. We induce an overall label with similar logical rules to those in §4.1: (1) if all a_i are predicted as entailed by P , we predict entailment, (2) if at least one a_i is predicted as contradicting P , we predict a contradiction, (3) otherwise, predict neutral. While this strategy does not yield competitive performance with full exam-

ple accuracy (Table 2, Column 1 versus Column 8), it does offer a more *interpretable* framework for LLMs that otherwise seem to struggle with logical consistency. Atom judgments may more difficult than overall judgments for a variety of reasons (see Appendix B.1), and in turn, inducing a label from individual atomic predictions may be less reliable.

5 Atoms in Defeasible Inference

We now turn to defeasible inference to explore how atomic sub-problems can help us better understand the complexities of the task, model performance, and the knowledge evaluated in the δ -SNLI dataset.

In traditional NLI (§4), labels of atomic sub-problems function like terms in an equation—the overall relation between P and H can be computed from strict logical rules over relations between P and individual a_i . In contrast, defeasible inference functions akin to fuzzy logic (Castro et al., 1998). Determining the overall effect of the update U on H involves a softer weighing of the direction and magnitude of its effect on each atom.

For example, consider the second row in Table 1. A human reading the update U (“The person is wearing a city uniform”) would conclude that U *strengthens* their belief in H (“The garbage man sweeps up where the can spilled”) as opposed to weakens it. Looking at this problem through the lens of atomic decomposition helps pinpoint why. H consists of five atoms, each representing pieces of evidence *not* present in P ripe for targeting by updates. Two of the five atoms are most strongly supported by U : the person is wearing a city uniform strongly strengthens our belief that they may be a garbage man (a_1) as well as that they may be sweeping up a spill (a_5). However, U has no effect on our belief that there is a can in the scenario (a_3). The co-occurrence of a_1 and a_5 compound the strengthening effect, leading to an overall strengthening effect of U on H .

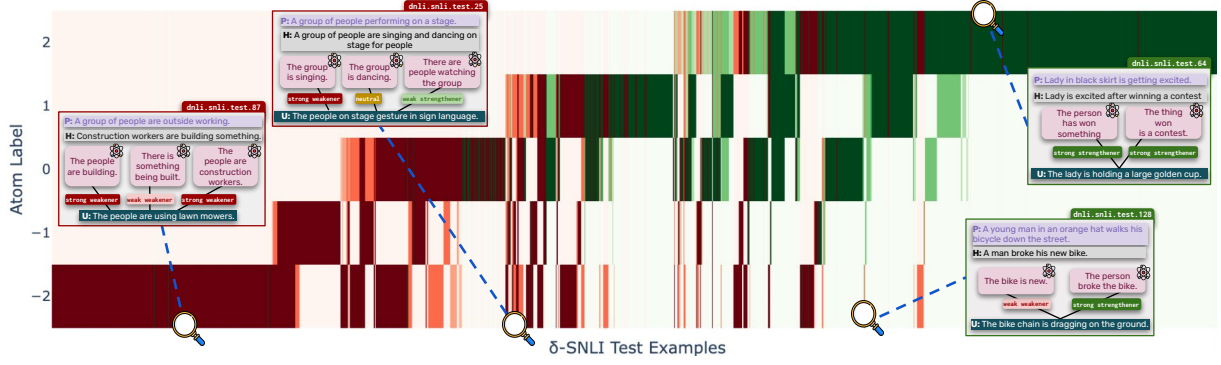


Figure 2: A rug plot visualization of 1,761 δ -SNLI instances and their corresponding distribution of atomic sub-problem labels. Each vertical slice represents one full δ -SNLI instance. Slice color (red or green) represents the full instance label (weakeners or strengtheners). For each δ -SNLI problem, we manually label each corresponding atomic sub-problem on a -2 (strongly weakens) to +2 (strongly strengthens) scale. Each vertical slice uses shading (light/dark) to represent the resulting distribution of atomic sub-problem labels (-2 to +2). Slices are ordered left to right by proportion of weakener labels, showing relatively high separation between red and green instances. When atomic sub-problems contain a mix of positive and negative labels, the full problem label may be a strengthener or a weakener, as illustrated by the two center-most exemplars.

Breaking down δ -SNLI hypotheses and forming atomic sub-problems in this manner gives us a framework to understand the intricacies of defeasible inference. We begin by benchmarking recent LLMs to understand the state of defeasible inference capabilities of models. Then, we introduce the idea of a *critical atom*, or the primary piece of information an update acts on. Finally, we use critical atoms as a way to better understand and interpret model behavior, as well as argue that critical atoms serve as a useful representation for measuring the type of knowledge evaluated in δ -SNLI.

5.1 Understanding Defeasible Inference with Atomic Sub-Problems

Benchmarking LLMs on δ -NLI. We benchmark a suite of recent models on full examples from δ -SNLI-TEST, including encoder models and prompt-based models, open-source and proprietary systems, as well as models of various sizes. We fine-tune encoder-only models (roberta-large and deberta-v3-large) on the train set of δ -SNLI for 2 epochs with a learning rate of $2e-5$ and a batch size of 32. For all prompt-based models, we do few-shot evaluation with Prompt C.1 and 10 in-context examples evenly split between strengtheners and weakeners.

Many of the models in our suite surpass the human performance benchmarked by [Rudinger et al. \(2020\)](#), with gpt-4o as the top performing model at 92% accuracy (Table 3). However, since it remains unclear whether this accuracy is indicative of a holistic understanding of situations in δ -SNLI, we

turn to studying performance on the atomic reasoning problems that compose each δ -SNLI example to better contextualize these results.

Annotating Atoms. Each atomic sub-problem in δ -SNLI is (P, a_i, U) tuple capturing the effect of the update on a specific atom a_i . An author annotated all valid (as determined in §3.2) atomic sub-problems for each example in δ -SNLI-TEST according to the five-point scale used in [Rudinger et al. \(2020\)](#) for validation (Table 5) ranging from *strongly weakens* (-2) to *strongly strengthens* (+2) with a midpoint value of *no effect* (0) for atoms on which U had no effect. The same external annotator from §3.2 annotated a random sample of 100 *valid* atomic sub-problems on the same -2 to +2 scale (Appendix E), obtaining an agreement of $\tau = 0.79$ with Kendall’s Tau ([Kendall, 1938](#)).

Ground Truth Label Distribution. Figure 2 visualizes the label distribution (-2 to +2) of atomic sub-problems of each δ -SNLI-TEST example as a thin vertical strip. Strips are green if the original example is a strengthener and red for weakeners. While some examples have all atomic labels of the *same* polarity, **a significant chunk of the dataset includes atomic sub-problems with no effect or the opposite polarity**. Examples depicted across the spectrum in Figure 2 illustrate the non-monotonicity of defeasible inference.

Atomic Sub-Problem Performance. We first measure the performance of models on atomic sub-problems using annotated ground-truth labels.

	Full Example Accuracy	Atom Accuracy	Critical Atom Accuracy	P(Full ✓ Critical ✓)	P(Full ✗ Critical ✗)
llama-3-8b-it	80.1	65.3	77.0	90.9	45.9
gpt-3.5-turbo	81.5	66.1	76.4	91.8	49.7
gemma-2-9b-it	82.3	68.5	72.1	91.9	60.5
Human	83.6	—	—	—	—
gpt-4o-mini-2024-07-18	86.9	74.8	79.4	93.1	66.4
roberta-large	87.4	83.4*	87.8*	94.0	44.9
llama-3-70b-it	88.0	73.8	81.9	93.6	64.5
deberta-v3-large	91.1	87.4*	91.5*	95.0	55.5
gpt-4o-08-06	92.6	77.2	83.5	96.5	75.5

Table 3: LLM accuracies on full δ -SNLI examples (col. 1), all atomic sub-problems (col. 2), and just critical atom sub-problems (col. 3), using our manual atomic labels. We also report full δ -SNLI example accuracy conditioned on accurate (col. 4) and inaccurate (col. 5) prediction of the corresponding critical atom sub-problem label.

The original δ -SNLI dataset was designed as a binary prediction task. However, as Figure 2 depicts, updates may also have no effect on atoms. We adapt Prompt C.1 to accommodate this ternary task (Prompt C.2) and use atoms in exemplars instead of full hypotheses. Since the train set of δ -NLI only admits binary labels, we reuse the finetuned models (deberta and roberta), and report atom accuracy on non-neutral atoms (80% of atoms, Figure 5).

Across the board, models perform worse on atomic sub-problems than on full examples (Table 3, Column 2). Since updates often act on multiple parts U (Table 1), we hypothesize that this compounding effect may contribute to higher performances on full examples.

We observe that some atomic sub-problems are more critical contributors to the overall effect of U on H . Consider the example in row 2 of Table 1. Since U acts most strongly on a_1 and a_5 , **we can assume that they are critical in determining the overall effect of U on H** , and correctly understanding the effect of U on a_3 or a_4 is not essential to the overall problem. We formalize this below.

5.2 Critical Atoms as Questions Under Discussion (QUD)

As established, updates vary in which atom they most strongly affect. Consider H and the three updates in Figure 3. Each U targets a distinct (or *critical*) atom without having an effect on the others. We formalize this notion by recognizing that hypotheses (or more broadly, sentences in discourse) serve as an answer to a large space of possible questions: all three questions in the right-most column could be answered with H . However, when updates target particular atoms, the *strategy* by which they do so favors a particular question Q , making it more likely that H is the answer to Q as opposed to any other question.

Hypothesis: A man in an orange vest stands on a dock .		
Atom	Example Update	H answers this QUD
a_1	The vest that the person wears is orange .	What color is the vest that the man is wearing?
a_2	The person stands .	What is the man doing on the dock?
a_3	The person is on a dock .	Where is the man?

Figure 3: Updates (U) may act on the same hypothesis H in different ways by targeting different atoms. Here, each U strongly targets a different atom, while having no effect on the other atoms derived from H (e.g. the U in the first row has no effect on a_3 in the last row). We refer to the atom(s) which an update most strongly affects as the “critical” atom of the (P, H, U) δ -NLI example. Critical atoms help identify the *question under discussion* of the example.

These questions function as *questions under discussion* (QUD), a well-studied linguistic phenomenon (Benz and Jasinskaja, 2017). The update “*The man smiles on his boat*” has no effect on the atom about vest color and hence does not pick out the QUD associated with that atom. We call the atoms above the *critical atom* for each corresponding update, as they uniquely pick out particular QUDs over others. **Critical atoms correspond to the particular inference or piece of knowledge a defeasible NLI example aims to test.** As such, we use them as a framework to measure consistency as well as the diversity of the δ -SNLI dataset.

Identifying Critical Atoms of Updates. In order to identify the critical atom for a defeasible NLI example, we identify the subset of its *valid* atoms with the strongest labels that match its overall polarity. The majority of δ -SNLI-TEST examples have one critical atom (Figure 6), but it is possible for example to have multiple if the effect is equally strong, as in row 2 of Table 1.

Performance on Critical Atomic Sub-Problems versus Full Examples. Across the board, models are stronger on the subset of critical atomic sub-problems than on all atomic problems (Table 3). We hypothesize that LLMs may be better at modeling stronger, **direct** inferences such as those in critical sub-problems, but may struggle when the effects are *indirect* or weaker.³ Such nuanced distinctions require a robust understanding of the multiple factors that control the underlying inference, a skill that even larger models seem to struggle with.

We also measure the probability that a model correctly predicts the label for the full example *given* that it has correctly (Column 4) and incorrectly (Column 5) solved all critical atomic sub-problems (Table 3). Correctly solving all atomic sub-problems is a strong indicator that a model is likely to predict the full problem correctly. However, some models still have as high as a 75% probability of predicting the full answer even having incorrectly predicted critical sub-problems, calling into question the robustness of their reasoning process in the face of such inconsistency.

6 Measuring Inferential Consistency in δ -NLI

As established, identifying the critical atom(s) of a δ -NLI example allows us to pinpoint the knowledge or fact that the example is designed to evaluate (§5.2). The two different examples in Figure 4 share the same critical atom, representing two different contexts under which the model must directly evaluate the fact “*The people are friends.*” A model correctly predicting whether or not people are friends under some contexts but not others may indicate that it has not fully understood the factors that influence the inference. Because they contain independent examples, few datasets accommodate measuring a model’s *inferential consistency* (I_C), or the likelihood that its prediction for a particular inference will remain consistently correct or incorrect under different contexts (here, contexts refers to different (P, U) pairs). Correctly drawing an inference under a single context does not guarantee that the model will make a correct prediction for the same inference under a different context.

To quantify this, we group examples in δ -SNLI-TEST by their critical atom and report the inferential consistency of different models.

³For example, U in Row 2 of Table 1 has no effect on a_3 , but that effect is slightly strengthened upon mention of a spill

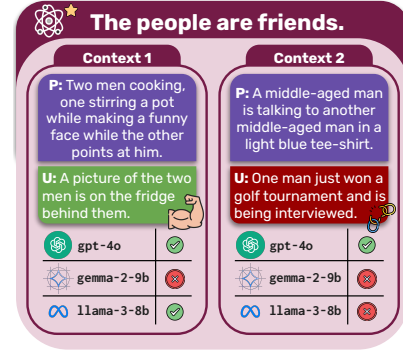


Figure 4: Grouping examples by their critical atom(s) allows us to understand under which *contexts* (premises and updates) a model has understood a piece of knowledge. Here, we show two δ -NLI examples that evaluate the same atom (top): one that *strengthens* it (left), and one that *weakens* it (right). A model that truly understands a fact and the factors that influence it (or, conversely does not) should yield consistently correct or incorrect predictions. However, some models have mixed accuracy among examples targeting the same atom, indicating that they only understand the inference under *some* contexts.

How many unique critical atoms does δ -SNLI-TEST evaluate? While δ -SNLI contains around 2K test examples and 88K training examples, how many distinct critical atoms underlie those examples? We quantify this by identifying semantically equivalent critical atoms in δ -SNLI-TEST by embedding each atom with NV-Embed-7B (Lee et al., 2024) and then computing pairwise cosine similarity between all critical atoms. We construct a graph G where nodes correspond to critical atoms and edges are drawn between nodes if their similarity is above a threshold ($\theta = 0.75$) and they are bidirectionally entailed as determined by the NLI model in §3.2. Finding semantically equivalent groupings then reduces to finding all maximal cliques (Tomita et al., 2006) in G .

The 1,761 δ -SNLI-TEST examples that contain at least one valid atom (from 193 unique P - H pairs) contain 429 unique atoms.⁴ Following the procedure above yields 349 unique cliques, or **349 unique critical atoms**.⁵ We bucket examples by these cliques, resulting in groups of examples that share a critical atom (which we call *critical atom*

in a_4 , since donning a city uniform is likelier at spill sites.

⁴Note that the number of unique atoms is upper-bounded by the SNLI hypotheses that δ -SNLI builds upon.

⁵We also experiment with generating the QUDs (Appendix D) for each critical atom and find that δ -SNLI-TEST evaluates 223 unique QUDs, around an 81% coverage of the available atoms.

buckets, or just *buckets*, represented as \otimes). We find that certain critical atoms arise frequently: top critical atoms include “*The others are friends of the person*” (21 examples) or “*The person is a man*” (20 examples).

Measuring Inferential Consistency (I_C). Srikanth et al. (2024a) introduce *paraphrastic consistency*, a metric capturing the probability that a model’s prediction for two paraphrases of the same NLI problem remain consistent (both incorrect or both correct). We adapt this metric to compute a model’s *inferential consistency*, or the probability that its predictions for two defeasible NLI examples e_i and e_j **that share the same critical atom** are either both correct or incorrect. As in Srikanth et al. (2024a), we define two terms:

- R_{\otimes} : a discrete random binary variable representing whether a model’s prediction for a full δ -NLI example is correct (1) or incorrect (0).
- θ_{\otimes} : $\mathbb{E}[R_{\otimes}]$, or the average correctness (i.e., accuracy) of the δ -NLI examples in a particular critical atom bucket.

For a binary classification task ($y \in \{0, 1\}$) and a model M , we define I_C as

$$I_C = \underbrace{P(M(e_i) = y, M(e_j) = y)}_{\text{prob. of both predictions correct}} + \underbrace{P(M(e_i) \neq y, M(e_j) \neq y)}_{\text{prob. of both predictions incorrect}} \quad (1)$$

We estimate I_C directly from the accuracies of critical atom buckets as:

$$I_C = \mathbb{E}[\theta_{\otimes}^2] + \mathbb{E}[(1 - \theta_{\otimes})^2] \quad (2)$$

Note that δ -NLI examples can have multiple critical atoms (Figure 6). In these cases, we divide the weight of the example e across all critical example buckets that share e when computing θ_{\otimes} .

Results. All models exhibit room for improvement in inferential consistency (Table 4), giving us a sense of how well the LLMs we analyze have internalized the critical atomic facts evaluated in δ -SNLI-TEST. One source of inconsistency we observe arises when certain contexts (premise-update pairs) demand more implicit reasoning, multi-hop reasoning, or background knowledge. For example, consider two different contexts under which the model must evaluate the same critical atom “*The people are tall*”:

Model	Inferential Consistency
llama-3-8b-it	74.5
gpt-3.5-turbo	75.8
gemma-2-9b-it	76.5
gpt-4o-mini-2024-07-18	81.7
roberta-large	82.6
llama-3-70b-it	83.5
deberta-v3-large	87.0
gpt-4o-08-06	88.7

Table 4: Inferential consistency of models (I_C) on δ -NLI examples. We group examples that share the same critical atom and compute the probability that two examples in the same group were both incorrectly or both correctly predicted by a model.

Context 1: P : Four people standing on a hiking trail in a forest with big tree logs on the ground, U : Their long legs step across several logs at once.

Context 2: P : Two men in orange uniforms stand before a train and do some work, U : They can easily touch the top of the train.

Several models struggle to draw the strengthening inference under the first context, but all models that we analyze successfully draw the strengthening inference under the second.

These analyses help us understand whether models have internalized certain pieces of knowledge and the factors that influence them, raising interesting questions about how best to collect updates to increase the coverage of a fact or situation. Defeasible reasoning systems must be able to deftly modulate inferences in a manner that is sensitive to diverse contexts. Future work may leverage the identification of critical atomic sub-problems to nudge annotators toward underrepresented critical atoms or contexts.

7 Conclusion

Decomposing hypotheses of NLI and defeasible NLI problems to form *atomic sub-problems* allows us to measure the logical consistency of models as well as better understand the structure of non-monotonic reasoning. We find that labels of atomic sub-problems in defeasible reasoning share a more complex relation to the full problem than in traditional NLI, even within the same label class. We introduce *critical atoms* as the primary fact or piece of knowledge evaluated by a defeasible NLI example, enabling us to group examples by shared critical atoms and measure the *inferential consistency* of LLMs across different contexts.

Limitations

This paper uses LLMs to generate atomic decompositions of hypotheses. While we did validate whether or not generated atoms were *valid* during manual annotation for δ -SNLI, we did not determine whether each family had missing decompositions of a particular granularity for SNLI-TEST-SAMPLE.

Both δ -NLI and SNLI have been shown to contain annotation artifacts (Gururangan et al., 2018), or particular statistical patterns between hypotheses (in the case of SNLI) and updates (in the case of δ -NLI). While prompt-based models are not directly trained on any of the datasets, the performance of finetuned roberta-large and deberta-v3-large may be inflated by the presence of annotation artifacts in update sentences. However, our methodology provides an opportunity for the collection of updates free from these artifacts by collecting updates that indirectly or lightly affect on hypotheses.

Finally, since the original train set of δ -NLI does not contain “no effect” or “neutral” updates, we report metrics for *finetuned* models on the subset of non-neutral atomic sub-problems. Future work may augment δ -NLI finetuning data with *contextual* neutral atomic sub-problems (akin to hard negatives in information retrieval) to enable this predictive ability in finetuned models.

Acknowledgments

We thank the anonymous reviewers as well as the members of the University of Maryland CLIP lab for their thoughtful and thorough feedback. We also thank Joe Stacey for feedback on an earlier draft of this paper. This work was supported by NIH Award No. R01MD016037 (Srikanth) and NSF CAREER Award No. 2339746 (Rudinger). The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health or the National Science Foundation. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the paper.

References

Anton Benz and Katja Jasinskaja. 2017. Questions under discussion: From sentence to discourse.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. [A large anno-](#)

[tated corpus for learning natural language inference](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

- Héctor-Neri Castañeda. 1967. Comments on d. davidson’s ‘the logical form of action sentences’.
- Juan Luis Castro, Enric Trillas, and Jose Manuel Zurita. 1998. Non-monotonic fuzzy reasoning. *Fuzzy Sets and Systems*, 94(2):217–225.
- Angelica Chen, Jason Phang, Alicia Parrish, Vishakh Padmakumar, Chen Zhao, Samuel R Bowman, and Kyunghyun Cho. 2023. Two failures of self-consistency in the multi-step reasoning of llms. *arXiv preprint arXiv:2305.14279*.
- Jifan Chen, Grace Kim, Aniruddh Sriram, Greg Durrett, and Eunsol Choi. 2024. [Complex claim verification with evidence retrieved in the wild](#). 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 3569–3587, Mexico City, Mexico. Association for Computational Linguistics.
- Jaemin Cho, Yushi Hu, Roopal Garg, Peter Anderson, Ranjay Krishna, Jason Baldridge, Mohit Bansal, Jordi Pont-Tuset, and Su Wang. 2023. Davidsonian scene graph: Improving reliability in fine-grained evaluation for text-image generation. *arXiv preprint arXiv:2310.18235*.
- Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.
- David Dowty. 1991. Thematic proto-roles and argument selection. *language*, 67(3):547–619.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Maxwell Forbes, Jena D Hwang, Vered Shwartz, Maarten Sap, and Yejin Choi. 2020. Social chemistry 101: Learning to reason about social and moral norms. *arXiv preprint arXiv:2011.00620*.
- Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and William B Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pages 1–9.
- John Glover, Federico Fancellu, Vasudevan Jaganathan, Matthew R. Gormley, and Thomas Schaaf. 2022. [Revisiting text decomposition methods for NLI-based factuality scoring of summaries](#). In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 97–105, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

- Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. [Annotation artifacts in natural language inference data](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.
- Paloma Jeretic, Alex Warstadt, Suvrat Bhooshan, and Adina Williams. 2020. [Are natural language inference models IMPPRESsive? Learning IMPLicature and PRESupposition](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8690–8705, Online. Association for Computational Linguistics.
- Ryo Kamoi, Tanya Goyal, Juan Diego Rodriguez, and Greg Durrett. 2023. [WiCE: Real-world entailment for claims in Wikipedia](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7561–7583, Singapore. Association for Computational Linguistics.
- Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2019. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*.
- Maurice G Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93.
- Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoenybi, Bryan Catanzaro, and Wei Ping. 2024. Nv-embed: Improved techniques for training llms as generalist embedding models. *arXiv preprint arXiv:2405.17428*.
- Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. 2022. [WANLI: Worker and AI collaboration for natural language inference dataset creation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6826–6847, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Hanmeng Liu, Ruoxi Ning, Zhiyang Teng, Jian Liu, Qiji Zhou, and Yue Zhang. 2023. Evaluating the logical reasoning ability of chatgpt and gpt-4. *arXiv preprint arXiv:2304.03439*.
- Bill MacCartney. 2009. *Natural language inference*. Stanford University.
- Sewon Min, Kalpesh Krishna, Xinxu Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. [FActScore: Fine-grained atomic evaluation of factual precision in long form text generation](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.
- Ani Nenkova and Rebecca Passonneau. 2004. [Evaluating content selection in summarization: The pyramid method](#). In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 145–152, Boston, Massachusetts, USA. Association for Computational Linguistics.
- Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. [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, Online. Association for Computational Linguistics.
- OpenAI. 2024. [Hello gpt-4o: A new model for openai’s future](#). Accessed: 2024-10-13.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- Alicia Parrish, William Huang, Omar Agha, Soo-Hwan Lee, Nikita Nangia, Alexia Warstadt, Karmanya Agarwal, Emily Allaway, Tal Linzen, and Samuel R. Bowman. 2021. [Does putting a linguist in the loop improve NLU data collection?](#) In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4886–4901, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Terence Parsons. 1990. Events in the semantics of english: A study in subatomic semantics.
- Ethan Perez, Patrick Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela. 2020. [Unsupervised question decomposition for question answering](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8864–8880, Online. Association for Computational Linguistics.
- Raymond Reiter. 1980. A logic for default reasoning. *Artificial intelligence*, 13(1-2):81–132.
- Rachel Rudinger, Vered Shwartz, Jena D. Hwang, Chandra Bhagavatula, Maxwell Forbes, Ronan Le Bras, Noah A. Smith, and Yejin Choi. 2020. [Thinking like a skeptic: Defeasible inference in natural language](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4661–4675, Online. Association for Computational Linguistics.
- Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3027–3035.
- Neha Srikanth, Marine Carpuat, and Rachel Rudinger. 2024a. How often are errors in natural language

- reasoning due to paraphrastic variability? *Transactions of the Association for Computational Linguistics*, 12:1143–1162.
- Neha Srikanth and Rachel Rudinger. 2022. [Partial-input baselines show that NLI models can ignore context, but they don’t](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4753–4763, Seattle, United States. Association for Computational Linguistics.
- Neha Srikanth, Rupak Sarkar, Heran Mane, Elizabeth Aparicio, Quynh Nguyen, Rachel Rudinger, and Jordan Boyd-Graber. 2024b. [Pregnant questions: The importance of pragmatic awareness in maternal health question answering](#). 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 7253–7268, Mexico City, Mexico. Association for Computational Linguistics.
- Joe Stacey, Pasquale Minervini, Haim Dubossarsky, Oana-Maria Camburu, and Marek Rei. 2024. [Atomic inference for NLI with generated facts as atoms](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10188–10204, Miami, Florida, USA. Association for Computational Linguistics.
- Joe Stacey, Pasquale Minervini, Haim Dubossarsky, and Marek Rei. 2022. [Logical reasoning with span-level predictions for interpretable and robust NLI models](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3809–3823, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.
- James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. [FEVER: a large-scale dataset for fact extraction and VERification](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.
- Etsuji Tomita, Akira Tanaka, and Haruhisa Takahashi. 2006. The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical computer science*, 363(1):28–42.
- Dhruv Verma, Yash Kumar Lal, Shreyashee Sinha, Benjamin Van Durme, and Adam Poliak. 2023. [Evaluating paraphrastic robustness in textual entailment models](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 880–892, Toronto, Canada. Association for Computational Linguistics.
- Miriam Wanner, Seth Ebner, Zhengping Jiang, Mark Dredze, and Benjamin Van Durme. 2024. A closer look at claim decomposition. *arXiv preprint arXiv:2403.11903*.
- Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. [A broad-coverage challenge corpus for sentence understanding through inference](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.
- Yating Wu, Ritika Mangla, Greg Durrett, and Junyi Jessy Li. 2023. [QUDeval: The evaluation of questions under discussion discourse parsing](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5344–5363, Singapore. Association for Computational Linguistics.
- Moy Yuan and Andreas Vlachos. 2024. [Zero-shot fact-checking with semantic triples and knowledge graphs](#). In *Proceedings of the 1st Workshop on Knowledge Graphs and Large Language Models (KaLLM 2024)*, pages 105–115, Bangkok, Thailand. Association for Computational Linguistics.
- Xiutian Zhao, Ke Wang, and Wei Peng. 2024. Measuring the inconsistency of large language models in preferential ranking. In *Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024)*, pages 171–176.

	SNLI	δ -SNLI
# test examples	1000	1837
label distr	36/32/32% (E/C/N)	50/50% (S/W)
# unique hypotheses	1000	203
# unique generated atoms	3263	475
# unique valid atoms	–	429
# unique atomic sub-problems	3263	4079
% strengtheners	–	46.0
% weakeners	–	31.4
% no effect	–	22.6
# unique critical atoms	–	349

Table 5: Dataset statistics for both SNLI and δ -SNLI.

Model	Proportion
llama-3-8b-it	32.0
gpt-3.5-turbo	31.1
gemma-2-9b-it	25.5
gpt-4o-mini-2024-07-18	34.5
llama-3-70b-it	33.3
gpt-4o-08-06	27.6

Table 6: Proportion of examples in SNLI-TEST-SAMPLE (1000 examples) depending on a completeness assumption of atom generation.

Prompt A.1: Atom Generation

Prompt: You are an expert linguist. You are given a **sentence**. Generate a list of atomic facts that are strictly logically entailed from the given **sentence**. Keep each fact independent and self-contained. Each fact should make sense when read on its own. Only write facts that are directly described or supported by the **sentence**. End your response with [END].

SENTENCE: {sentence}

FACTS:

A.1 Coverage of Generated Atoms

Generated atoms must cover the information presented in the hypothesis. For example, for the hypothesis H in Figure 1, an atom generation model should produce atoms covering all three pieces of information: at least one mentioning *professional*, *actors*, and *summer*. Pruning should not reduce the coverage of the atom set. Here, we estimate the completeness, or coverage, of the generated atoms with respect to the hypothesis of NLI examples.

Completeness in the case of SNLI. We note that completeness of atoms only matters in particular cases. When the overall $P-H$ pair is predicted by the model to be entailed, the logical consistency check does not rely on completeness, as no single atom, if predicted by the model to be entailed, should be predicted as anything other than entailment. However, there are two possible scenarios of

inconsistency when $P-H$ is predicted to be neutral: (a) All atoms were predicted to be entailed by P (this does necessitate ensuring completeness), or (b) one or more atoms is predicted to be a contradiction (this does not require ensuring completeness). Consistency when $P-H$ is predicted to be a contradiction does require checking that all atoms cover the hypothesis. Table 6 shows the proportion of examples in SNLI-TEST-SAMPLE that are dependent upon a completeness assumption as per the description for each label above. These help contextualize our results in Table 2 by estimating an upper bound of logical consistency.

To measure completeness in SNLI, we randomly sample 50 SNLI examples and annotate for completeness, and find that in 49/50 examples, all pieces of information from the original hypothesis project into at least one atom. The only example in our random sample that was missing an atom was for the hypothesis shown in Table 8.

We also sample another set of 50 random examples from SNLI-TEST-SAMPLE where one of our models (gpt-4o-2024-08-06) predicted either contradiction or neutral *and* where consistency failed in order to understand of what percentage of errors are due to actual failures in consistency or are simply due to lack of completeness. Here, we study the atoms that the model deemed entailed from the hypothesis for each example, and annotate whether those cover all pieces of information in the hypothesis. In 6/50 examples, gpt-4o-2024-08-06 incorrectly judged that at least one of the generated atoms was not entailed by the hypothesis, hence omitting it and causing a completeness issue for the set of atoms over which we measure its logical consistency. However, we find that of these six cases, only two examples were missing an atom in the set of generated atoms, again indicating that our *atom generation* process reliably projects all information from the hypothesis into the generated set of atoms.

Completeness in the case of δ -NLI. We manually validated all generated 4,079 atomic subproblems in the entire test set of δ -SNLI-TEST. Lack of completeness is most likely when none of the atoms have a gold label in the direction of the gold label of the overall example, indicating that an atom may be missing. This happens in only 3% of examples (70/1761) in δ -SNLI-TEST. We annotate these 70 examples to understand whether or not an atom was indeed missing after our automatic pruning

Example	Premise: A female within the foreground is heading towards a large white colored pillar that is apart of a large building with people are loitering or waiting on the steps of said building.
	Hypothesis: The woman has an important meeting today in the building.
	Update: The woman is wearing a janitor’s uniform (<i>weaker</i>)
Atoms	a_1 : The person has something (0, no effect).
	a_2 : The thing the person has is a meeting. (0, no effect)
	a_3 : The meeting is important (0, no effect)
	a_4 : The meeting is today (0, no effect)
	a_5 : The meeting is in a building (0, no effect)

Table 7: An example of a lack of completeness in generated atoms.

Hypothesis	The man and woman are going to a movie in the city.
Atoms	a_1 : There are two people.
	a_2 : There are two people who are a man and a woman.
	a_3 : There are people going.
	a_4 : There are people going to a movie.
	a_5 : There is a destination for the people going.
	a_6 : The destination is a movie.
	a_7 : There is a city.
	a_8 : The people are going to the city.

Table 8: An example of a lack of completeness in generated atoms for SNLI. Here the missing atom is “*The movie is in the city.*”

step. We find that in only 28 of the 70 examples (representing only 1% of *all* examples), at least one atom was missing. Most of the cases where none of the atoms have a gold label in the direction of the gold label of the overall example are cases where the original example is flawed in some way (hinges on some stereotype, ambiguous language, faulty reasoning, or the original crowdworker who wrote the update did not understand the instructions) and our annotations do not propagate the flawed assumptions or reasoning from the original example. Table 7 shows an example. The original gold label propagates a stereotype or attitude towards janitors (i.e it is less likely they have an important meeting if they are a janitor). We choose not to propagate such attitudes or stereotypes in our annotation, hence, all atoms have the “no effect” label.

B SNLI

Prompt B.1: Traditional NLI

Prompt: You will be given a **premise** and a **hypothesis** about that **premise**. You need to decide whether the **hypothesis** is entailed by the **premise** by choosing one of the following answers: ‘e’: The **hypothesis** follows logically from the information contained in the **premise**. ‘c’: The **hypothesis** is logically false from the information contained in the **premise**. ‘n’: It is not possible to determine whether the **hypothesis** is true or false without further information. Read the **premise** and **hypothesis** and select the correct answer from the three answer labels (e, n, c). Also provide a single line of explanation in a new line. End your response with [END] and output nothing after.

Premise: {premise}
Hypothesis: {hypothesis}

Is the hypothesis entailed by, contradicted by, or neutral with respect to the premise?

B.1 Inconsistency in SNLI

We analyze the set of randomly selected sample of 50 examples from Appendix A.1 where one of our models (gpt-4o-2024-08-06) was logically inconsistent. We find that inconsistencies arise for a number of reasons, some of which we discuss here.

Misjudgment of atom entailment from the hypothesis. Measuring logical consistency in SNLI examples happens in two phases (1) the model determines whether the atom entails from the hypothesis, and (2) the model predicts the relationship between the premise and an atom. Depending on its strength, we find that there are inconsistencies that arise from the model’s misjudgment of atom entailment. For example:

Premise: A man on a bicycle, wearing cycle gear, riding at a fast past down paved trail surrounded by tree’s and grass.

Hypothesis: The man is riding on the sidewalk.

Generated Atoms (a_1) There is a person. (a_2) There is a person who is a man. (a_3) There is a person riding. (a_4) There is a person riding on something. (a_5) The thing the person is riding on is a sidewalk.

Based on the full premise and hypothesis, gpt-4o-2024-08-06 predicted that the hypothesis was *neutral* in relation to the premise. However, in its atomic predictions between the premise and each atom, the model incorrectly determined that the last atom was not entailed by the hypothesis, and hence it was not included in the set of atoms

that were used to measure consistency. It judged all other atoms as entailed from the premise, hence leading to the model behaving inconsistently in this example.

Use of hypernyms in atoms. We make sure to include hypernyms of entities in the set of atomic facts. For example, “*the man dances*” is decomposed into “*there is a person*”, “*the person is a man*”, “*the person dances*”. In some cases, models have difficulty on premise-atom judgments where the premise uses the hyponym (“man”) and the hypothesis uses the hypernym (“person”).

Out of domain syntactic constructions. The syntactic structures of many of our atoms differ, in some cases significantly, from the constructions in the original dataset. For example “*the thing the person is eating is a sandwich*” is a pseudo-cleft construction that is very rare in the SNLI dataset. As such, some sentences are out of domain for encoder models that were trained on SNLI or prompt-based models that have inadvertently seen SNLI training data in pretraining corpora.

Weaker effects based on annotation elicitation. Both SNLI and defeasible NLI are datasets created by crowdworkers writing hypotheses and updates conditioned on a label. One of the consequences of this process is that hypotheses and updates tend to strongly express the desired label. In contrast, atoms tend to express labels in a softer way, and their effects often compound when taken together. As such, these atoms may be out of distribution as compared to hypotheses that express the label with a higher magnitude. Since the effect of each atom is lighter than when they are taken together in a full sentence, atomic judgments are sometimes much more difficult than overall judgments.

Co-reference Effects. In some cases, atomic generations remove some of the implicit co-reference between the hypothesis and premise. We observe that changing the co-reference in atoms can result in inconsistencies between the overall example and each premise-atom judgment.

C Defeasible NLI

Prompt C.1: Defeasible Inference

Prompt: You are a reasoning system. You are given a description of a **situation** and a **hypothesis** about that **situation** that may or may not be true. Given some more **evidence** about the **situation**, output ‘more’ if the **hypothesis** seems more likely to be true after learning the **evidence**, or output ‘less’ if the **hypothesis** seems less likely to be true after learning the **evidence**. Also provide a single line of explanation in a new line. End your response with [END] and output nothing after.

Situation: {context}

Hypothesis: {hypothesis}

Evidence: {evidence}

Does the evidence make the hypothesis about the situation more or less likely to be true?

Prompt C.2: Defeasible Inference Atoms

Prompt: You are a reasoning system. You are given a description of a **situation** and a **hypothesis** about that **situation** that may or may not be true. Given some more **evidence** about the **situation**, output ‘more’ if the **hypothesis** seems more likely to be true after learning the **evidence**, output ‘less’ if the **hypothesis** seems less likely to be true after learning the **evidence**, or output ‘none’ if the likelihood of the **hypothesis** remains unchanged after learning the **evidence**. Also provide a single line of explanation in a new line. End your response with [END] and output nothing after.

Situation: {context}

Hypothesis: {hypothesis}

Evidence: {evidence}

Does the evidence make the hypothesis about the situation more or less likely to be true?

D QUD Generation for Understanding Diversity

In order to generate QUDs from critical atom propositions, we use gpt-4o-2024-08-06 with a temperature of 0.01 and 15 exemplars of critical atom to QUD mappings along with Prompt D.1. For example, the critical atom *The dog is brown.* translates to the QUD *What color is the dog?* Exemplars include critical atoms that do not use generics (e.g. “girl”) mapped to QUDs that use generics (“person”). This is to make sure that similar inferences are correctly mapped to the same QUD. Table 9 shows examples of QUDs generated by the

Critical Atom	QUD
The kids are waiting.	What are the people doing?
The dog is brown.	What color is the dog?
The thing the grandpa is wearing is a shirt.	What is the person wearing?
The two women are sisters.	What is the relationship between the two people?
The bike is brand new.	What is the condition of the object?
The girls are posing.	What are the people doing?

Table 9: QUDs generated from critical atoms in δ -SNLI-TEST by gpt-4o-2024-08-06.

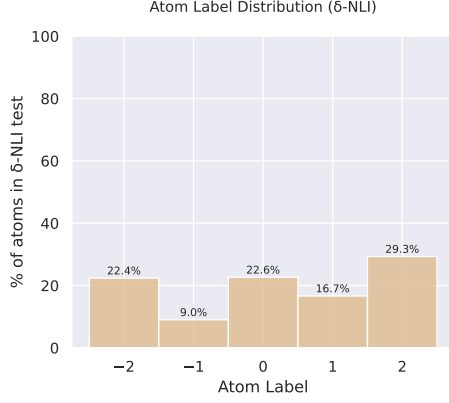


Figure 5: Distribution of fine-grained labels across all atoms in δ -SNLI-TEST.

model. We validate all generated QUDs by asking an external annotator if the QUD can be answered by the critical atom sentence, and find that in 92% of cases, the QUD generated by gpt4o correctly captures the critical atom.

Prompt D.1: QUD Generation

Prompt: You are an expert linguist. Given a short **sentence**, generate a **question** that is answered by the sentence. Read the whole sentence carefully before generating the question.
Sentence: critical atom
Question:

E Validation Instructions

Since atomic sub-problems mirror the original validation task for defeasible inference, we use the instructions provided to annotators from [Rudinger et al. \(2020\)](#) to ensure alignment.

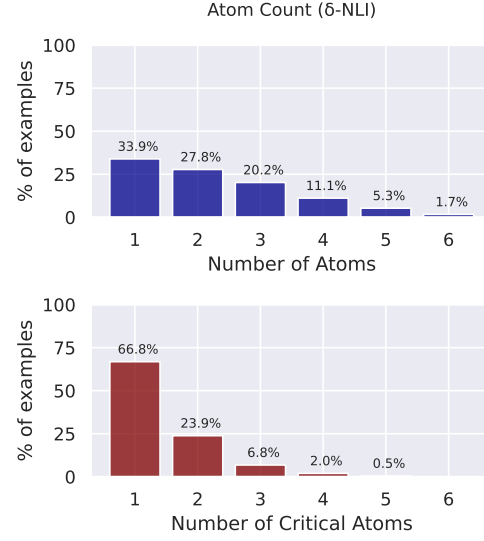


Figure 6: Number of atoms (top) and number of critical atoms (bottom) per example in δ -SNLI-TEST.

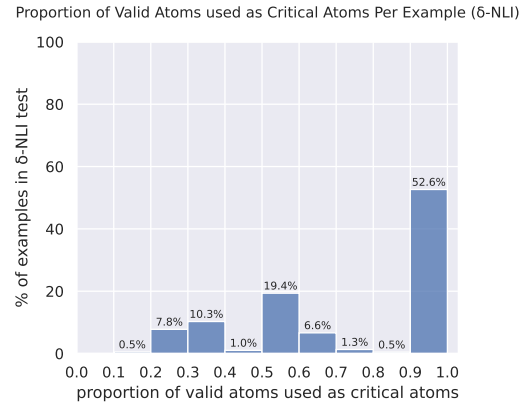


Figure 7: Proportion of valid atoms used as critical atoms per update in δ -SNLI-TEST. 52% of updates target all valid possible atoms at once in a single update sentence.