Guiding Explanation-based NLI through Symbolic Inference Types

Anonymous ACL submission

Abstract

This work investigates localised, quasi-symbolic inference behaviours in distributional representation spaces by focusing on Explanation-based Natural Language Inference (NLI), where two explanations (premises) are provided to derive a single conclusion. We first establish the connection between natural language and symbolic inferences by characterising quasi-symbolic NLI behaviours, named symbolic inference types. Next, we establish the connection between distributional and symbolic inferences by formalising the Transformer encoder-decoder NLI model as a rule-based neural NLI model - a quasi-symbolic NLI representation framework. We perform extensive experiments which reveal that symbolic inference types can enhance model training and inference dynamics, and deliver localised, symbolic inference control. Based on these findings, we conjecture the different inference behaviours are encoded as functionally separated subspaces in the latent parametric space, as the future direction to probe the composition and generalisation of symbolic inference behaviour in distributional representation spaces.

1 Introduction

Explanatory sentences (Jansen et al., 2018b) can encode hierarchical, taxonomic, and causal relations between concepts (Gardenfors and Zenker, 2015). By understanding and reasoning over these concepts expressed by explanations, humans can make intricate decisions, which is significant in scientific, cognitive, and AI domains. In this work, we focus on the Explanation-based Natural Language Inference (NLI) task where two explanations (premises) are provided to derive a single conclusion. Within this task, a central challenge involves achieving localised and (quasi-)symbolic inference behaviour. E.g., given the two premises: *milk is a kind of liquid* and *liquids can flow*, one may derive the conclusion *milk can flow* by localising and substituting the concept *liquids* with *milk*.

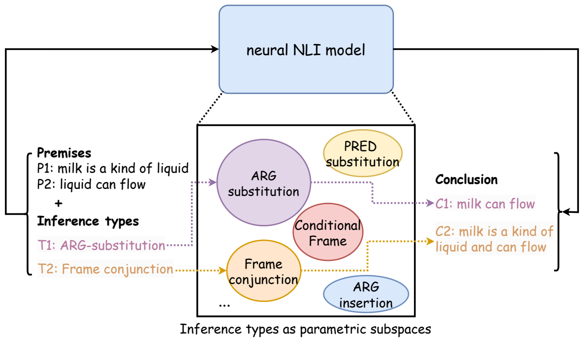

Figure 1: Conceptual visualisation for the proposed *Quasi-symbolic NLI representation* framework. Inference types can be encoded as functional subspaces, which are separated or disentangled in parametric space. Thus, by manipulating the inference types, we can deliver localised, symbolic inference control.

A key question then arises: How can we train current Transformer-based NLI models to learn and generalise this quasi-symbolic behaviour in the distributional representation space? Investigating this question allows us to shorten the gap between deep latent semantics and formal linguistic representations (Gildea and Jurafsky, 2000; Banarescu et al., 2013), integrating the flexibility of distributional-neural models with the properties of linguistically grounded representations, facilitating both interpretability (i.e., compositionality (Dankers et al., 2022; Marcus, 2003)) and generative control.

Recent studies have demonstrated that the predicate-argument structure and semantic roles from explanatory sentences (Argument Structure Theory - AST representations) (Jackendoff, 1992) can be effectively represented, localised, and disentangled in the latent space of transformer-based models (Zhang et al., 2024a,c). A particular instance of an AST representation is the Abstract Meaning Representation (AMR) (Banarescu et al., 2013), which represents the relations between semantic variables, allowing us to first establish the connection between natural and symbolic language inferences. Specifically, we leverage the AMR to *systematically characterise quasi-symbolic inference behaviours, named symbolic inference types, grounded on AMR symbolic graphs.* Using the explanation-based NLI dataset (EntailmentBank, Dalvi et al. (2021)), we identify ten categories of symbolic transformations and provide annotations for 5,134 premise-conclusion pairs in Section 3.

Next, we establish the connection between distributional and symbolic inferences from the perspective of neural representation spaces (see Section 4). An ideal neuro-symbolic NLI model should demonstrate two core representational capabilities: (i) the capacity to encode and to systematically apply inference rules and (ii) the ability to elicit syntactic-semantic features (Valentino, 2022). Motivated by this, we propose *quasi-symbolic NLI representation* conceptual framework over a Transformer-based encoder-decoder NLI architecture (Figure 1), in which *the symbolic inference types are injected to guide the formation of inference behaviours within the latent parametric space.* As for the former, explicit supervision on inference types should align the model's reasoning trajectory with target inference behaviours. By varying different inference types, the model should perform rule-based inference behaviour. With respect to the latter, we introduce a feature space (i.e., abstract sentence bottleneck) in the centre of the encoder-decoder architecture. Ideally, this low-dimensional feature space encodes sufficiently abstract, high-level semantic representations during inference.

We provide extensive experiments to evaluate both capabilities, including the training and inference (Section 5.1), localised inference control (Section 5.2), and feature representation with explanation inference retrieval task (Section 5.3). Experimental results reveal that the symbolic inference type can assist model training, inference, and deliver localised inference control, indicating the possibility of neural NLI models to learn and generalise the inference rules in the distributional space.

In summary, this work provides a complete initial step in investigating the quasi-symbolic inference over distributional semantic spaces, with the following contributions: **(1)** We first establish the connection between natural and symbolic language inferences from the perspective of linguistics by systematically characterising quasi-symbolic inference behaviours, named symbolic inference types, grounded on the AST/AMR representations. **(2)** We establish the distributional-symbolic connection from the perspective of neural representation space by proposing the quasi-symbolic NLI representation conceptual framework where the formation of inference behaviours is guided via our symbolic inference types in the latent space. Experimental results showed that the symbolic inference type supervision can improve model training, inference, and localisation. Based on those findings, we conjecture that different inference types are encoded as functional subspaces which are separated or disentangled in the parametric space, as a future direction to probe the composition of symbolic inference behaviours in distributional representation spaces.

Interpreting and controlling the NLI process from the perspective of the distributional space is a largely promising approach in NLP. To our knowledge, this is the first study to explore the quasi-symbolic NLI behaviour, targetting a more universal NLI control and interpretation, rather than a strict symbolic representation or architectural modification. The experimental pipelines are released[1].

2 Related Work

In this section, we review the related work around two topics: *neuro-symbolic representations* and *semantic control over latent spaces*, to highlight the current research limitation and elucidate the motivation underlying our work.

Neuro-symbolic representations. A longstanding goal in NLP is to blend the representational strengths of neural networks with the interpretability of symbolic systems to build more robust NLI models. Current methods usually inject symbolic behaviour through explicit symbolic representations, including graph (Khashabi et al., 2018; Khot et al., 2017; Jansen et al., 2017; Kalouli et al., 2020; Thayaparan et al., 2021), linear programming (Valentino et al., 2022b; Thayaparan et al., 2024), adopting iterative methods, using sparse encoding mechanisms (Valentino et al., 2020; Lin et al., 2020), synthetic quasi-natural language expression (Clark et al., 2020; Yang and Deng, 2021; Yanaka et al., 2021; Fu and Frank, 2024; Weir et al., 2024), symbolic-refined LLMs (Olausson et al., 2023; Quan et al., 2024), etc. Those studies ignore the underlying neuro-symbolic behaviour in neural representation space.

[1] https://anonymous.4open.science/r/Inference_type-5E07/

From an Explainable AI perspective, many studies have shown that neural networks can encode sparse neural-symbolic concepts without explicit symbolic injection across areas like image embedding (Ren et al., 2022; Deng et al., 2021; Li and Zhang, 2023), word embedding (Ethayarajh et al., 2018; Allen et al., 2019; Ri et al., 2023), contextual embedding (Gurnee et al., 2023; Nanda et al., 2023; Li et al., 2024), and LLM interpretation (Park et al., 2024; Templeton et al., 2024). By understanding the symbolic behaviour within neural networks, their decision-making logic can be better interpreted and controlled (Chen et al., 2024).

In this work, we draw on quasi-symbolic NLI objectives within distributional neural models, targetting better controllability and interpretability.

Semantic control over latent spaces. Latent variable models, such as VAE (Kingma and Welling, 2013) and Diffusion (Dhariwal and Nichol, 2021), have shown the capability of symbolic representation, control, and interpretation over the distributional space, which are widely deployed in the NLP domain, such as disentangled representation learning (Zhang et al., 2024a) and style-transfer (Liu et al., 2023a; Gu et al., 2023; Zhang et al., 2024b). Guided by semantic annotation, such as labels (Carvalho et al., 2023) and classifiers (Ho and Salimans, 2022), distinct semantic features can be geometrically separated and composed in the latent space, enhancing localisation and interpretability. However, this concept remains under-explored in the NLI domain. Thus, we propose the quasi-symbolic NLI representation conceptual framework and inference types as an initial step to probe the localised, quasi-symbolic NLI behaviour.

In the next section, we start by defining the symbolic inference types for semantically bridging the natural language and symbolic inferences.

3 Defining Symbolic Inference Types

Valentino et al. (2021) has demonstrated that stepwise explanation-based NLI cannot be directly framed as pure logical reasoning. Explanatory chains, while looking plausible at first inspection, commonly have subtler incompleteness and consistency problems from a logical point of view. Meanwhile, explanatory chains corresponding to definable inference patterns and symbolic operations can be localised over the sentence structure. Motivated by this middle ground between logical representations and lexico-semantic inference patterns, we introduce granular inference types based on explanatory sentences, using AMR to define the symbolic operations involved in step-wise inference, linking transformations from premises to conclusions [2]. Table 1 describes the AMR-grounded inference types and examples from the EntailmentBank corpus. Next, we define each lexico-semantic inference type and the corresponding symbolic forms.

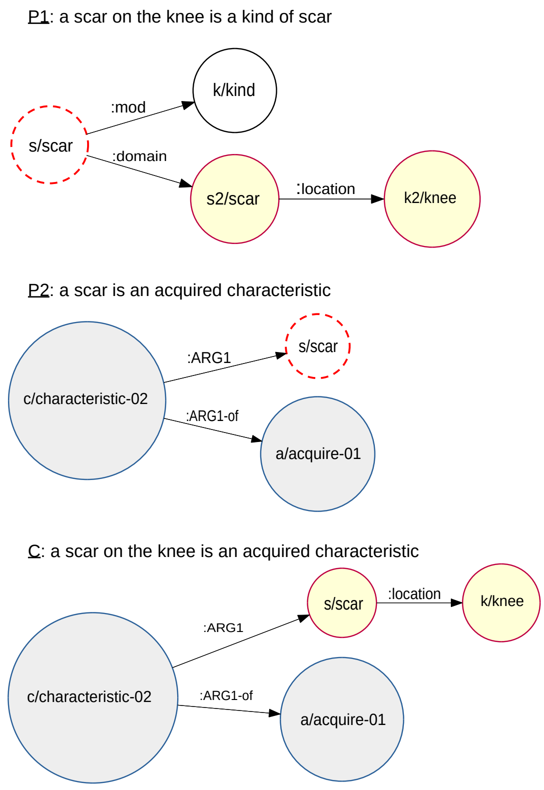

Figure 2: AMR argument substitution: the inference behaviour is defined as subgraph substitution.

The *substitution* category refers to obtaining a conclusion by replacing a predicate/argument term from one premise with a predicate/argument term from the other premise. Possible variations of this category include (1) *argument (ARG) substitution*, (2) *predicate (PRED) substitution*, and (3) *frame (PRED+ARG) substitution*. In this category, one premise is used to connect two terms which are usually connected by *is a kind of, is a source of*, etc. Conceptualising the AMR representation as a graph, this can be symbolically represented as a subgraph substitution operation over the premise graphs, as illustrated in Figure 2. The *PRED sub-*

[2]Please note that AMR is not used as a representation mechanism in the proposed architecture, but only to precisely ground these symbolic operations within a well-defined semantic representation structure.

Original type	Symbolic type	Prop.	Example entailment relation
Substitution	ARG substitution (ARG-SUB)	19%	P1: a scar on the knee is a kind of scar P2: a scar is an acquired characteristic C: a scar on the knee is an acquired characteristic
	PRED substitution (PRED-SUB)	5%	P1: food contains nutrients and energy for living things P2: to contain something can mean to store something C: food stores nutrients and energy for living things
	Frame substitution (FRAME-SUB)	20%	P1: the formation of diamonds requires intense pressure P2: the pressure is intense deep below earth 's crust C: the formation of diamonds occurs deep below the crust of the earth
Inference from Rule	Conditional frame insertion/substitution (COND-FRAME)	12%	P1: if something is renewable then that something is not a fossil P2: fuel wood is a renewable resource C: wood is not a fossil fuel
Further Specification or Conjunction	ARG insertion (ARG-INS)	18%	P1: solar energy comes from the sun P2: solar energy is a kind of energy P3: solar energy is a kind of energy that comes from the sun
	Frame conjunction (FRAME-CONJ)	6%	P1: photosynthesis stores energy P2: respiration releases energy C: photosynthesis stores energy and respiration releases energy
Infer Class from Properties	ARG/PRED generalisation (ARG/PRED-GEN)	1%	P1: rock is a hard material P2: granite is a hard material C: granite is a kind of rock
Property Inheritance	ARG substitution (Property Inheritance) (ARG-SUB-PROP)	0.4%	P1: blacktop is made of asphalt concrete P2: asphalt has a smooth surface C: a blacktop has a smooth surface
Causal Expression	Causality (IFT)	0.8%	an optical telescope requires visible light for human to use clouds / dusts block visible light if there is clouds or dusts, then the optical telescope cannot be used
Example-based Inference	Example (EXAMPLE)	0.9%	a shelter can be used for living in by raccoons some raccoons live in hollow logs an example of a shelter is a raccoon living in a hollow log

Table 1: Examples of symbolic inference types, with their corresponding abbreviations provided in parentheses and used consistently throughout the paper. The EntailmentBank utilised for this task comprises 5,134 instances, with our annotations covering 84% of the (premises, conclusion) cases. These annotations are planned for public release.

stitution category works in a similar manner, but replacing a predicate term. The two predicates are usually linked by the following patterns: "v_1 *is a kind of* v_2", "*to* v_1 *something means to* v_2 *something*", etc. The *frame (PRED+ARG) substitution* category combines both previous categories by replacing a frame (predicate subgraph) of one of the premises with one from the other premise.

The *further specification or conjunction* category allows for obtaining a conclusion by joining both premises. It includes (4) *ARG insertion* and (5) *frame conjunction*. For *ARG insertion*, the conclusion is obtained by connecting an argument from one of the premises to a frame of the other. As for *frame conjunction/disjunction*, the conclusion is obtained by joining the premises graphs through a conjunction/disjunction node (*and*) or (*or*).

The *inference from rule* category from Dalvi et al. (2021) encompasses a specific instance of insertion or substitution, identified as (6) *conditional frame insertion/substitution*. In this category, a frame is either inserted or replaced as an argument of a premise, following a conditional pathway present in the other premise. This process is illustrated in Figure 4 (Appendix A).

The inference type *infer class from properties* has been re-categorised as (7) *ARG or PRED generalisation*, where a new *:domain* relation frame is created if both premise graphs differ by a single predicate/argument term. (8) *Property inheritance*, on the other hand, is a special case of *ARG substitution*, where one of the premises describes a *is made of* relationship between the entity in the other premise and its replacement.

Finally, (9) *Causal Expression* and (10) *Example-based Inference* categories are defined according to the key lexical characteristic of the conclusion, as systematic AMR transformations which could be applied without rephrasing the underlying explanatory sentences could not be determined. More details about the annotation procedure are provided in Appendix A.

Thus far, we have established a connection between natural and symbolic language inferences through the AMR graph. In the next section, we aim to establish the distributional-symbolic NLI connection from the point of neural representation space.

4 Quasi-symbolic NLI Framework

In this section, we first formalise the concept of Quasi-symbolic NLI and then map it to the practical encoder-decoder architectures.

4.1 Quasi-symbolic NLI Formalisation

In this study, we formalise the concept of "quasi-symbolic NLI behaviour" as rule-based reasoning over neural representation, where discrete inference behaviours are implemented through differentiable transformations over continuous neural representations. This is achieved by characterising and manipulating quasi-symbolic inference behaviours, denoted by $\pi \in \Pi$, where Π represents the space of all possible inference rules.

The process involves three key stages: (i) Neural Encoding: The premises p_1 and p_2 are encoded into continuous vector representations ($\overrightarrow{p1}$ and $\overrightarrow{p2}$) in a neural space. (ii) Rule-Based Reasoning: The encoded representations are transformed using a reasoning function guided by the inference behaviour π. (iii) Neural Decoding: The resulting vector, $\overrightarrow{c}$, is decoded into a natural language conclusion c. Formally, the process can be described as follows:

$$\begin{aligned} &1.\ \overrightarrow{p1}, \overrightarrow{p2} = f_{encode}(p_1, p_2) \\ &2.\ \overrightarrow{c} = f_{reason}(\overrightarrow{p1}, \overrightarrow{p2}; \pi) \\ &3.\ c = f_{decode}(\overrightarrow{c}) \end{aligned}$$

Here, f_{encode}, f_{reason}, f_{decode} represent the encoding, reasoning, and decoding functions in a neural NLI model. The injection of π should exhibit two advantages:

1. Training Dynamics: During training, explicit supervision on π aligns the model's reasoning trajectory with target inference behaviours, improving conclusion prediction accuracy.

2. Inference Composition: By varying π during inference, the model can separate the semantics of the premises from the inference behaviour. This enables localised, quasi-symbolic NLI control, allowing for flexible and interpretable reasoning.

4.2 Quasi-symbolic NLI Representation

We focus on encoder-decoder architectures (e.g., T5) due to their inherent separation of reasoning and decoding phases, which naturally accommodates quasi-symbolic reasoning. From a representational perspective, we propose the concepts of latent rule space and feature space to align with the function of the neuro-symbolic NLI model.

Latent rule space. The latent rule space refers to the functional parameter space (i.e., models' weights), which captures the structured, rule-based reasoning behaviours $\pi \in \Pi$. We further propose that rule-based reasoning is primarily materialised in the encoder of the encoder-decoder NLI model:

Proposition: *The inference behaviour is instantiated at the encoder and can be controlled by the injection of the associated inference type labels.*

Latent feature space. The latent feature space refers to the input or output embedding space. To evaluate the feature representation capability, we next describe the methodological framework behind the construction of the abstract sentence representation within T5 (named T5 bottleneck).

As for the encoder's final layer output embedding, we compute dimension-wise mean pooling over token embeddings, followed by a multi-layer perceptron to obtain sentence embeddings. As for the decoder's input embedding, we reconstruct token embeddings via linear projection, feeding them into the decoder's cross-attention mechanism. Here, we only describe the optimal setup. We provide a systematic way to choose the best setup in the Appendix B.

5 Empirical Analysis

The experiment addresses three key questions: Section 5.1: (i) Do symbolic inference types enhance model training and inference performance? Section 5.2: (ii) Can these inference types be used for prescriptive inference control? Section 5.3: (iii) Does the incorporation of a sentence bottleneck contribute to improved feature representation? All experimental details are provided in Appendix B.

5.1 Training and Inference Evaluation

Firstly, we evaluate (i) if symbolic inference types enhance model training and inference performance. We consider three mechanisms to conditionally inject the symbolic inference types into the latent space, which are described below, where *p1*, *p2*, and *con* are the premises and conclusion, respectively, and </s> is a special token for sentence separation: **i.** The inference type as the prefix for the premises at the Encoder: *the inference type is [type] </s> p1 </s> p2* **ii.** The inference type as the prefix for the conclusion in the Decoder: *</s> the inference type is [type]. con* **iii.** The inference type at the end of the conclusion in the Decoder: *</s> con. the inference type is [type].*

Training dynamics. We first evaluate training performance based on five metrics: Loss (cross-entropy), perplexity (PPL), BLEURT (Sellam et al., 2020), BLEU (Papineni et al., 2002), and cosine similarity against sentenceT5 (Ni et al., 2021). By comparing the predicted and golden conclusions, we can fairly evaluate the accuracy of the NLI performance. For the baseline, we choose the T5, Bart (Lewis et al., 2019), GPT2 (Radford et al., 2019), our T5 bottleneck and Optimus (Li et al., 2020) with 768 latent dimensions as testbed. The performances are measured from the Entailment test set.

As illustrated in Table 2, all baselines with inference types always have lower test losses and PPLs, which means the inference type can help the model training. This finding suggests that during training, explicit supervision on inference types aligns the model's reasoning trajectory with target inference behaviours, improving conclusion prediction accuracy **(finding1)**. A similar observation is reflected in the test loss curve shown in Figure 8.

Moreover, across all baseline models, incorporating inference types into the encoder consistently results in improved performance as measured by BLEU, Cosine, and BLEURT metrics, indicating the inference behaviour is instantiated at the encoder (*Proposition*) **(finding2)**.

Furthermore, previous studies have revealed that the LLM evaluation can be consistent with the results obtained by expert human evaluation (Chiang and Lee, 2023; Liu et al., 2023b; Wang et al., 2023; Huang et al., 2023). Thus, we also conduct a quantitative analysis to measure whether the generated conclusion contradicts the premises through LLM evaluators, including ChatGPT4o as the baseline and GPT4o-mini for comparison. Table 3 indicates that EP can consistently result in improved LLM agreement scores. A qualitative evaluation based on the manual check is presented in appendix C (Tables 14 and 15).

In-context learning. Next, we quantitatively evaluate the symbolic inference types within in-context learning (ICL) in contemporary large language models (LLMs). As illustrated in Table 4, prompting with inference types can improve the performance of ICL in both seq2seq and causal LLMs. Besides, within the context of causal LLMs, an increase in few-shot examples[3], improves the performance. This finding indicates that our inference types can be generalised across various checkpoints and architectures, ultimately enhancing the reasoning capabilities of LLMs **(finding3)**.

Baseline	INJ	BLEU	Cosine	BLEURT	Loss ↓	PPL ↓
seq2seqLM: encoder-decoder architecture						
T5 original (small)	DE	0.55	0.96	0.30	0.53	1.44
	DP	0.59	0.96	0.34	0.58	1.57
	EP	**0.65**	**0.97**	**0.45**	**0.52**	**1.41**
	NO	0.54	0.96	0.22	0.69	2.22
T5 original (base)	DE	0.46	0.96	0.23	0.49	1.33
	DP	0.53	0.96	0.25	0.51	1.38
	EP	**0.61**	**0.97**	**0.39**	**0.45**	**1.22**
	NO	0.57	0.96	0.33	0.61	1.65
Bart (base)	DE	0.44	0.94	0.03	0.55	1.49
	DP	0.38	0.93	-0.42	**0.48**	**1.30**
	EP	**0.57**	**0.96**	**0.23**	0.58	1.57
	NO	0.54	0.96	0.17	0.63	1.71
T5 original (large)	DE	0.60	0.97	0.46	**0.40**	**1.49**
	DP	0.64	0.97	0.44	0.46	1.58
	EP	**0.67**	**0.97**	**0.50**	0.59	1.80
	NO	0.57	0.96	0.31	0.61	1.84
Flan-T5 (large)	DE	0.01	0.73	-1.34	6.91	10.2
	DP	0.01	0.73	-1.34	7.00	15.4
	EP	**0.21**	**0.87**	**-1.04**	**1.30**	**3.66**
	NO	0.20	0.87	-1.14	1.34	3.81
CausalLM: decoder only architecture						
GPT2 (large)	DE	0.02	0.87	-1.15	0.73	2.07
	DP	**0.08**	**0.90**	**-0.91**	**0.73**	**2.07**
	NO	0.07	0.90	-0.93	0.76	2.06
GPT2 (xl)	DE	0.20	0.88	-1.10	0.63	1.87
	DP	**0.28**	**0.91**	**-0.90**	**0.60**	**1.82**
	NO	0.27	0.90	-0.97	0.68	1.97
seq2seqLM with sentence bottleneck						
T5 bottleneck (base)	DE	0.35	0.91	-0.15	**0.84**	**2.31**
	DP	0.39	0.91	-0.13	0.86	2.36
	EP	**0.42**	**0.92**	**-0.07**	1.23	3.42
	NO	0.35	0.91	-0.20	1.24	3.45
Optimus (BERT-GPT2)	DE	**0.26**	**0.80**	**-1.11**	0.87	2.38
	DP	0.25	0.79	-1.14	**0.85**	**2.33**
	EP	0.09	0.74	-1.17	1.11	3.03
	NO	0.07	0.74	-1.20	1.13	3.09

Table 2: Quantitative evaluation on testset, where best results are highlighted in **bold**. Specification for abbreviation. INJ: ways for injecting the information of inference types into the model, it includes DE: decoder end, DP: decoder prefix, EP: encoder prefix, NO: no inference type. PPL is perplexity, Loss is cross entropy.

Baseline	INJ	ChatGPT4o	GPT4o-mini
T5 original (large)	DE	0.85	0.83
	DP	0.86	0.83
	EP	**0.91**	**0.85**
	NO	0.84	0.82

Table 3: Agreement scores for the quantitative analysis using LLMs on the test set. We also provide a qualitative manual evaluation in appendix C (Tables 14 and 15), with the prompt being provided in Table 17.

[3] We randomly sample the examples with the same inference type as the current test example from the training set. We perform ten times and calculate the average for each metric.

Baseline	INJ	Num	BLEU	Cosine	BLEURT
Seq2seqLLM: encoder-decoder architecture					
CoT-T5 (11b) (Kim et al., 2023)	Yes	10	0.51	0.97	0.39
	Yes	5	0.51	0.97	0.39
	Yes	0	0.50	0.97	0.36
	NO	0	**0.46**	**0.96**	**0.31**
Flan-T5 (xl)	Yes	10	0.49	0.96	0.40
	Yes	5	0.48	0.96	0.39
	Yes	0	0.52	0.96	0.39
	NO	0	**0.44**	**0.95**	**0.24**
Flan-T5 (xxl)	Yes	10	0.51	0.97	0.41
	Yes	5	0.53	0.97	0.43
	Yes	0	0.50	0.96	0.37
	NO	0	**0.48**	**0.96**	**0.36**
CausalLLM: decoder only architecture					
GPT-3.5-turbo-0125	Yes	10	0.52	0.96	0.40
	Yes	5	0.48	0.96	0.35
	Yes	0	0.46	0.96	**0.31**
	NO	0	**0.42**	0.96	0.33
GPT-4-0613	Yes	10	0.53	0.97	0.50
	Yes	5	0.52	0.97	0.47
	Yes	0	0.52	0.97	0.50
	NO	0	**0.47**	**0.96**	**0.40**
llama3-8b-8192	Yes	10	0.48	0.96	0.33
	Yes	5	0.45	0.96	0.32
	Yes	0	0.37	0.95	0.22
	NO	0	**0.34**	**0.95**	**0.19**
llama3-70b-8192	Yes	10	0.54	0.97	0.54
	Yes	5	0.52	0.97	0.52
	Yes	0	0.51	0.97	0.47
	NO	0	**0.44**	**0.96**	**0.40**

Table 4: ICL evaluation of test cases, where worst results are highlighted in **bold**. The prompt is *"performing natural language inference [where the inference type is type, description],* $[p1; p2; c]_{\times num}$*. p1, p2, what is the conclusion?"*. num is the number of examples. The *description* is based on the definition of inference types in Section 3.

5.2 Quasi-symbolic NLI Evaluation

Secondly, we evaluate (ii) if these inference types can be used for prescriptive inference control.

Qualitative evaluation. We qualitatively evaluate the quasi-symbolic NLI behaviour on the generation of conclusions by systematically intervening on the inference type prior to the encoder. As illustrated in Table 5, we can observe that the associated linguistic properties of the conclusion can be controlled consistently with the inference type modifications and this inference control is independent of the semantics of premises, which indicates that the representation mechanisms can improve inference control with regard to symbolic/lexico-semantic properties (**finding4**). For example, when the type is ARG substitution (ARG-SUB), the model can generate *the blacktop is made of a smooth surface* by replacing the argument *asphalt concrete* with *smooth surface*. The conclusions are changed to *asphalt and blacktop have the same surface* when the inference type is the conjunction (FRAME-CONJ). Additional examples are provided in Table 16.

P1: blacktop is made of asphalt concrete
P2: asphalt has a smooth surface

ARG-SUB: the blacktop is made of smooth surface
ARG-SUB-PROP: blacktop has a smooth surface
ARG/PRED-GEN: a blacktop is a kind of asphalt
ARG-INS: asphalt concrete blacktop has a smooth surface
FRAME-CON: asphalt and blacktop have the same surface
IFT: if the asphalt has a smooth surface then the blacktop will have a smooth surface

Table 5: Controllable generation over original T5 (base) (ARG-SUB: argument substitution, ARG/PRED-GEN: argument/predicate generalisation. ARG-SUB-PROP: property inheritance. ARG-INS: argument insertion, FRAME-CON: frame conjunction, IFT: casual expression.). The example of the T5 bottleneck is provided in Table 12 (Appendix C).

Quantitative analysis. Next, we perform an automated quantitative analysis using LLMs, including ChatGPT4o and GPT4o-mini. For each pair of premises in the EntailmentBank test set, we apply various inference types to generate a diverse set of conclusions using the fine-tuned T5 (base) model. We then assess the resulting (premises, conclusion, inference type) tuples based on two criteria: (i) whether the generated conclusion contradicts the premises, and (ii) whether the (premises, conclusion) pair is consistent with the specified inference type. Utilising the prompt detailed in Table 17 (Appendix C), we report the model agreement score for each criterion. As illustrated in Table 6, the T5 (base) model with controlled symbolic inference types achieves consistency and alignment scores exceeding 60% for both evaluated dimensions.

Evaluators	consistency	alignment
ChatGPT4o	0.67	0.63
GPT4o-mini	0.65	0.62

Table 6: Automated quantitative analysis scores.

5.3 Latent Feature Space Evaluation

Finally, we evaluate (iii) whether the incorporation of feature space (i.e., abstract sentence bottleneck) contributes to improved feature, concept representation in the NLI task.

We especially select the VAE baselines due to their analogous encoder-bottleneck-decoder architecture, wherein the compressed and orthogonal sentence bottleneck captures high-level, generalised semantics (concepts) (Mercatali and Freitas, 2021; Zhang et al., 2024a). This structural similarity is essential for facilitating human-like inference and cognition (LCM team, 2024).

Explanation-based NLI. We first evaluate the NLI performance of different baselines on the Entailment test set. A more effective feature space can enhance generation performance (Zhang et al., 2024c). Consequently, the same generation-related metrics can be applied to evaluate the quality of the feature space.

The baseline includes the state-of-the-art Transformer VAE model: Optimus (Li et al., 2020) and Della (Hu et al., 2022) with two different sentence dimensions (32 and 768), and five LSTM language autoencoders with 768 latent dimensions: denoising AE (Vincent et al. (2008), DAE), β-VAE (Higgins et al., 2016), adversarial AE (Makhzani et al. (2015), AAE), label adversarial AE (Rubenstein et al. (2018), LAAE), and denoising adversarial autoencoder (Shen et al. (2020), DAAE).

Table 7 illustrates that the T5 bottleneck can outperform all baselines on generation-related metrics, indicating better abstract sentence representations are learned to guide the decoding process.

Baseline	BLEU	Cosine	BLEURT	Loss ↓	PPL ↓
Optimus(32)	0.07	0.74	-1.20	1.13	2.31
Optimus(768)	0.08	0.74	-1.21	**0.82**	**2.27**
DELLA(32)	0.08	0.85	-1.23	1.69	5.41
DELLA(768)	0.09	0.87	-1.09	1.54	4.66
DAE(768)	0.15	0.89	-0.95	1.33	3.78
AAE(768)	0.11	0.88	-0.95	1.35	3.85
LAAE(768)	0.09	0.74	-1.12	1.38	3.97
DAAE(768)	0.07	0.74	-1.20	1.43	4.17
β-VAE(768)	0.07	0.74	-1.20	1.43	4.17
T5 bottleneck	**0.35**	**0.91**	**-0.20**	1.24	3.45

Table 7: Comparison of different baselines on EntailmentBank testset, T5 bottleneck has 768 dimensions.

Explanation inference retrieval. We next evaluate the abstract sentence embedding using as an associated explanation retrieval task (explanation-regeneration - i.e. retrieving the associated explanatory facts relevant to a claim) (Valentino et al., 2022a). Given a question-and-answer pair, it reconstructs the entailment tree by searching the explanations from a fact bank (i.e., WorldTree (Jansen et al., 2018a)) in an iterative fashion using a dense sentence encoder. In this framework, we can replace the sentence embeddings with the proposed T5 bottleneck baseline to evaluate its abstract sentence embeddings. We compare the T5 bottleneck with abstract sentence representation baselines: Optimus and five LSTM VAEs, and evaluate them via mean average precision (MAP). As illustrated in Table 8, the T5 bottleneck outperforms all baselines, indicating that it can deliver a better abstract representation of explanatory sentences and entailment relations in a retrieval setting **(finding5)**.

depth	t=1	t=2	t=3	t=4
DAE(768)	30.27	31.74	30.65	30.74
AAE(768)	29.13	30.47	29.33	29.14
LAAE(768)	19.13	20.86	18.32	18.01
DAAE(768)	13.16	15.42	14.30	13.97
β-VAE(768)	10.03	10.07	10.05	10.05
Optimus(768)	28.21	29.35	28.35	28.27
T5 bottleneck(768)	**34.47**	**35.28**	**34.50**	**34.47**

Table 8: Explanatory inference retrieval task where t represents the depth of entailment tree.

6 Conclusion and Future Work

This study serves as a foundational step in exploring the quasi-symbolic NLI behaviour within distributional semantic spaces. We establish the connection between natural and symbolic language inferences by characterising quasi-symbolic inference behaviours based on AMR graphs. Then, we propose the quasi-symbolic NLI representation framework. Experimental results reveal that integrating symbolic inference types enhances training dynamics, inference precision, and explanation retrieval, suggesting the potential for neuro-symbolic NLI.

Based on these findings, we hypothesise that distinct inference types can be represented as separated functional subspaces within the parametric space. During the training phase, explicit supervision on inference types aligns the model's reasoning trajectory with target inference behaviours, improving conclusion prediction accuracy. By manipulating various inference types during the inference stage, the NLI model can separate the semantics of the premises from the inference behaviour, which enables localised, quasi-symbolic NLI control.

In future research, we will examine this hypothesis and investigate the composition and generalisation of quasi-symbolic inference behaviours within latent spaces, targetting an explainable and controllable neuro-symbolic NLI model.

Limitations

Automatic NLI evaluation. In the domain of LLM automatic evaluation, the prevailing strategy is to select the most advanced LLM as the automatic evaluator (Chiang and Lee, 2023; Liu et al., 2023b; Wang et al., 2023; Huang et al., 2023). We perform a quantitative analysis of the inference consistency in the deductive reasoning process of LLMs, such as ChatGPT-4o. However, this assessment may be unreliable due to the inherent limitations of LLMs in logical reasoning. Human evaluation presents a potential alternative, yet the rigorous design of a protocol to systematically verify the logicality of NLI remains an under-explored area in this field. Although we perform a qualitative manual check for LLM evaluation in Table 14 and 15, this assessment is not systematic or rigorously structured. A promising direction for improving automatic NLI evaluation is the integration of symbolic theorem provers with LLMs.

Mechanism analysis. This study empirically explores quasi-symbolic inference behaviours within distributional semantic spaces. Our findings indicate that symbolic inference types can enhance model training, facilitate inference processes, and enable localised inference control. However, we have not yet provided a formal explanation for these observations. We hypothesise that quasi-symbolic inference behaviour arises from the geometrical separation of inference types within the parametric space. This hypothesis may be linked to the finding presented in Ortiz-Jimenez et al. (2023), which demonstrated that different tasks are disentangled in the visual embedding space of CLIP (Radford et al., 2021). By incorporating distinct task directions in the weight space, the model can achieve multi-task performance via task arithmetic. Future research will address this hypothesis by examining the geometric properties of the parametric space with the target of better composition, arithmetic, generalisation, and interpretation in the neuro-symbolic NLI domain.

References

Carl Allen, Ivana Balazevic, and Timothy Hospedales. 2019. What the vec? towards probabilistically grounded embeddings. *Advances in neural information processing systems*, 32.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Danilo S. Carvalho, Yingji Zhang, Giangiacomo Mercatali, and Andre Freitas. 2023. Learning disentangled representations for natural language definitions. In *Findings of the European chapter of Association for Computational Linguistics (Findings of EACL)*. Association for Computational Linguistics.

Lu Chen, Yuxuan Huang, Yixing Li, Yaohui Jin, Shuai Zhao, Zilong Zheng, and Quanshi Zhang. 2024. Alignment between the decision-making logic of llms and human cognition: A case study on legal llms. *arXiv preprint arXiv:2410.09083*.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631, Toronto, Canada. Association for Computational Linguistics.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2020. Transformers as soft reasoners over language. *arXiv preprint arXiv:2002.05867*.

Bhavana Dalvi, Peter Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. 2021. Explaining answers with entailment trees. *arXiv preprint arXiv:2104.08661*.

Marco Damonte, Shay B. Cohen, and Giorgio Satta. 2017. An incremental parser for abstract meaning representation. In *Proceedings of EACL*.

Verna Dankers, Elia Bruni, and Dieuwke Hupkes. 2022. The paradox of the compositionality of natural language: A neural machine translation case study. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4154–4175, Dublin, Ireland. Association for Computational Linguistics.

Huiqi Deng, Qihan Ren, Hao Zhang, and Quanshi Zhang. 2021. Discovering and explaining the representation bottleneck of dnns. *arXiv preprint arXiv:2111.06236*.

Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794.

Kawin Ethayarajh, David Duvenaud, and Graeme Hirst. 2018. Towards understanding linear word analogies. *arXiv preprint arXiv:1810.04882*.

Le Fang, Tao Zeng, Chaochun Liu, Liefeng Bo, Wen Dong, and Changyou Chen. 2021. Transformer-based conditional variational autoencoder for controllable story generation. *arXiv preprint arXiv:2101.00828*.

Xiyan Fu and Anette Frank. 2024. Exploring continual learning of compositional generalization in nli. *arXiv preprint arXiv:2403.04400*.

Peter Gardenfors and Frank Zenker. 2015. Applications of conceptual spaces: the case for geometric knowledge representation.

Daniel Gildea and Daniel Jurafsky. 2000. Automatic labeling of semantic roles. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL '00, page 512–520, USA. Association for Computational Linguistics.

Yuxuan Gu, Xiaocheng Feng, Sicheng Ma, Lingyuan Zhang, Heng Gong, Weihong Zhong, and Bing Qin. 2023. Controllable text generation via probability density estimation in the latent space. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12590–12616, Toronto, Canada. Association for Computational Linguistics.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. 2023. Finding neurons in a haystack: Case studies with sparse probing. *arXiv preprint arXiv:2305.01610*.

Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. 2016. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*.

Jonathan Ho and Tim Salimans. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*.

Jinyi Hu, Xiaoyuan Yi, Wenhao Li, Maosong Sun, and Xing Xie. 2022. Fuse it more deeply! a variational transformer with layer-wise latent variable inference for text generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 697–716, Seattle, United States. Association for Computational Linguistics.

Fan Huang, Haewoon Kwak, and Jisun An. 2023. Is chatgpt better than human annotators? potential and limitations of chatgpt in explaining implicit hate speech. In *Companion Proceedings of the ACM Web Conference 2023*, WWW '23, page 294–297. ACM.

Ray S Jackendoff. 1992. *Semantic structures*, volume 18. MIT press.

Peter Jansen, Rebecca Sharp, Mihai Surdeanu, and Peter Clark. 2017. Framing qa as building and ranking intersentence answer justifications. *Computational Linguistics*, 43(2):407–449.

Peter Jansen, Elizabeth Wainwright, Steven Marmorstein, and Clayton Morrison. 2018a. WorldTree: A corpus of explanation graphs for elementary science questions supporting multi-hop inference. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Peter A Jansen, Elizabeth Wainwright, Steven Marmorstein, and Clayton T Morrison. 2018b. Worldtree: A corpus of explanation graphs for elementary science questions supporting multi-hop inference. *arXiv preprint arXiv:1802.03052*.

Aikaterini-Lida Kalouli, Richard Crouch, and Valeria de Paiva. 2020. Hy-NLI: a hybrid system for natural language inference. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5235–5249, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Daniel Khashabi, Tushar Khot, Ashish Sabharwal, and Dan Roth. 2018. Question answering as global reasoning over semantic abstractions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2017. Answering complex questions using open information extraction. *arXiv preprint arXiv:1704.05572*.

Seungone Kim, Se June Joo, Doyoung Kim, Joel Jang, Seonghyeon Ye, Jamin Shin, and Minjoon Seo. 2023. The cot collection: Improving zero-shot and few-shot learning of language models via chain-of-thought fine-tuning. *arXiv preprint arXiv:2305.14045*.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Paul-Ambroise Duquenne Maha Elbayad Artyom Kozhevnikov Belen Alastruey Pierre Andrews Mariano Coria Guillaume Couairon Marta R. Costa-jussà David Dale Hady Elsahar Kevin Heffernan João Maria Janeiro Tuan Tran Christophe Ropers Eduardo Sánchez Robin San Roman Alexandre Mourachko Safiyyah Saleem Holger Schwenk LCM team, Loïc Barrault. 2024. Large Concept Models: Language modeling in a sentence representation space.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Chunyuan Li, Xiang Gao, Yuan Li, Baolin Peng, Xiujun Li, Yizhe Zhang, and Jianfeng Gao. 2020. Optimus: Organizing sentences via pre-trained modeling of a latent space. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4678–4699.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2024. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36.

Mingjie Li and Quanshi Zhang. 2023. Does a neural network really encode symbolic concepts? In *International Conference on Machine Learning*, pages 20452–20469. PMLR.

Bill Yuchen Lin, Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Xiang Ren, and William W Cohen. 2020. Differentiable open-ended commonsense reasoning. *arXiv preprint arXiv:2010.14439*.

Guangyi Liu, Zeyu Feng, Yuan Gao, Zichao Yang, Xiaodan Liang, Junwei Bao, Xiaodong He, Shuguang Cui, Zhen Li, and Zhiting Hu. 2023a. Composable text controls in latent space with ODEs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16543–16570, Singapore. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023b. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. 2015. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*.

Gary F Marcus. 2003. *The algebraic mind: Integrating connectionism and cognitive science*. MIT press.

Giangiacomo Mercatali and André Freitas. 2021. Disentangling generative factors in natural language with discrete variational autoencoders. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3547–3556, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ivan Montero, Nikolaos Pappas, and Noah A Smith. 2021. Sentence bottleneck autoencoders from transformer language models. *arXiv preprint arXiv:2109.00055*.

Neel Nanda, Andrew Lee, and Martin Wattenberg. 2023. Emergent linear representations in world models of self-supervised sequence models. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 16–30, Singapore. Association for Computational Linguistics.

Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B Hall, Daniel Cer, and Yinfei Yang. 2021. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. *arXiv preprint arXiv:2108.08877*.

Theo Olausson, Alex Gu, Ben Lipkin, Cedegao Zhang, Armando Solar-Lezama, Joshua Tenenbaum, and Roger Levy. 2023. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5153–5176, Singapore. Association for Computational Linguistics.

Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. 2023. Task arithmetic in the tangent space: Improved editing of pre-trained models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2024. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 39643–39666. PMLR.

Xin Quan, Marco Valentino, Louise A Dennis, and André Freitas. 2024. Verification and refinement of natural language explanations through llm-symbolic theorem proving. *arXiv preprint arXiv:2405.01379*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Jie Ren, Mingjie Li, Qirui Chen, Huiqi Deng, and Quanshi Zhang. 2022. Towards axiomatic, hierarchical, and symbolic explanation for deep models.

Narutatsu Ri, Fei-Tzin Lee, and Nakul Verma. 2023. Contrastive loss is all you need to recover analogies as parallel lines. *arXiv preprint arXiv:2306.08221*.

Paul K Rubenstein, Bernhard Schoelkopf, and Ilya Tolstikhin. 2018. On the latent space of wasserstein auto-encoders. *arXiv preprint arXiv:1802.03761*.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*.

Tianxiao Shen, Jonas Mueller, Regina Barzilay, and Tommi Jaakkola. 2020. Educating text autoencoders: Latent representation guidance via denoising. In *International Conference on Machine Learning*, pages 8719–8729. PMLR.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. 2024. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*.

Mokanarangan Thayaparan, Marco Valentino, and André Freitas. 2021. Explainable inference over grounding-abstract chains for science questions. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1–12.

Mokanarangan Thayaparan, Marco Valentino, and André Freitas. 2024. A differentiable integer linear programming solver for explanation-based natural language inference. *arXiv preprint arXiv:2404.02625*.

Marco Valentino. 2022. Explanation-based scientific natural language inference.

Marco Valentino, Ian Pratt-Hartmann, and André Freitas. 2021. Do natural language explanations represent valid logical arguments? verifying entailment in explainable nli gold standards.

Marco Valentino, Mokanarangan Thayaparan, Deborah Ferreira, and André Freitas. 2022a. Hybrid autoregressive inference for scalable multi-hop explanation regeneration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11403–11411.

Marco Valentino, Mokanarangan Thayaparan, and André Freitas. 2020. Explainable natural language reasoning via conceptual unification. *arXiv preprint arXiv:2009.14539*.

Marco Valentino, Mokanarangan Thayaparan, and André Freitas. 2022b. Case-based abductive natural language inference. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1556–1568.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 1096–1103, New York, NY, USA. Association for Computing Machinery.

Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023. Is ChatGPT a good NLG evaluator? a preliminary study. In *Proceedings of the 4th New Frontiers in Summarization Workshop*, pages 1–11, Singapore. Association for Computational Linguistics.

Nathaniel Weir, Kate Sanders, Orion Weller, Shreya Sharma, Dongwei Jiang, Zhengping Jiang, Bhavana Dalvi Mishra, Oyvind Tafjord, Peter Jansen, Peter Clark, and Benjamin Van Durme. 2024. Enhancing systematic decompositional natural language inference using informal logic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9458–9482, Miami, Florida, USA. Association for Computational Linguistics.

Hitomi Yanaka, Koji Mineshima, and Kentaro Inui. 2021. SyGNS: A systematic generalization testbed based on natural language semantics. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 103–119, Online. Association for Computational Linguistics.

Kaiyu Yang and Jia Deng. 2021. Learning symbolic rules for reasoning in quasi-natural language. *arXiv preprint arXiv:2111.12038*.

Yingji Zhang, Danilo Carvalho, and Andre Freitas. 2024a. Learning disentangled semantic spaces of explanations via invertible neural networks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2113–2134, Bangkok, Thailand. Association for Computational Linguistics.

Yingji Zhang, Danilo Carvalho, Marco Valentino, Ian Pratt-Hartmann, and Andre Freitas. 2024b. Improving semantic control in discrete latent spaces with transformer quantized variational autoencoders. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1434–1450, St. Julian's, Malta. Association for Computational Linguistics.

Yingji Zhang, Marco Valentino, Danilo Carvalho, Ian Pratt-Hartmann, and Andre Freitas. 2024c. Graph-induced syntactic-semantic spaces in transformer-based variational AutoEncoders. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 474–489, Mexico City, Mexico. Association for Computational Linguistics.

A Annotation Details

Annotation procedure. Annotation was performed manually for 5134 entailment triples (two premises, one conclusion) from the EntailmentBank (Dalvi et al., 2021), according to Algorithm 1. Graph subset relations and root matching are relaxed for non-argument (:ARG*, op*) edges, meaning relations such as *:manner* or *:time* can be ignored for this purpose. Two independent annotators with post-graduate level backgrounds in Computational Linguistics were used in this process, on a consensus-based annotation scheme where a first annotator defined the transformations and a second annotator verified and refined the annotation scheme, in two iterations. The annotation of the AMR graph is based on an off-the-shelf parser (Damonte et al., 2017). The descriptions for each inference type category are as follows:

ARG-SUB (Figure 2): the conclusion is obtained by replacing one argument with another argument.

PRED-SUB: the conclusion is obtained by replacing one verb with another verb.

FRAME-SUB: the conclusion is obtained by replacing a frame of one of the premises with one from the other premise.

COND-FRAM (Figure 4): the conclusion is obtained according to the conditional premise with keyword “if".

ARG-INS (Figure 3): the conclusion is obtained by connecting an argument from one of the premises to a frame of the other.

FRAME-CONJ: the conclusion is obtained by using connectives to connect two premises.

ARG/PRED-GEN (Figure 5): a new *:domain* relation frame is created in the conclusion if both premise graphs differ by a single predicate/argument term.

ARG-SUB-PROP (Figure 6): one of the premises describes a “*is made of*” relationship between the entity in the other premise and its replacement.

IFT: the conclusion should be a conditional sentence.

EXAMPLE: the conclusion should contain the keyword “example".

Unknown (UNK) category. In this work, our annotation occupies 84% based on the EntailmentBank corpus. As for other unknown categories, we do not further specify them, as they either require knowledge outside of the scope of the premises or do not have a consistent symbolic transformation expression. An additional subtype called *premise copy* was included for the cases where the conclusion has the same graph as one of the premises.

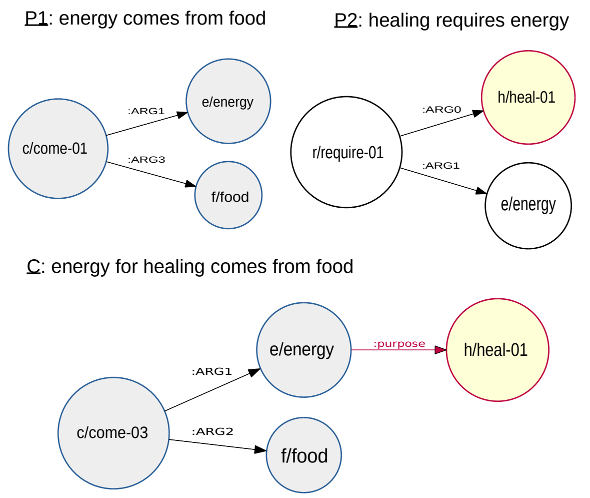

Figure 3: AMR argument insertion (ARG-INS).

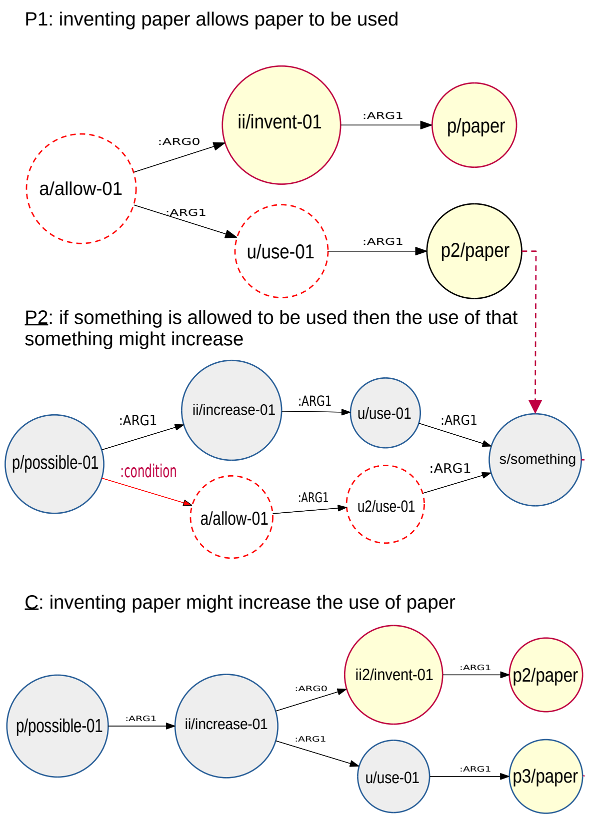

Figure 4: AMR conditional frame insertion (COND-FRAME).

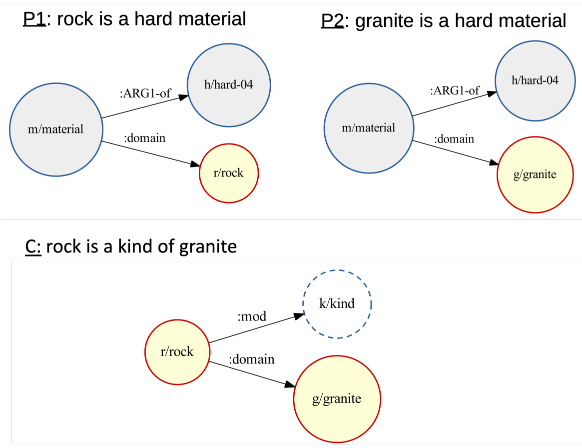

Figure 5: AMR argument generalisation (ARG-GEN).

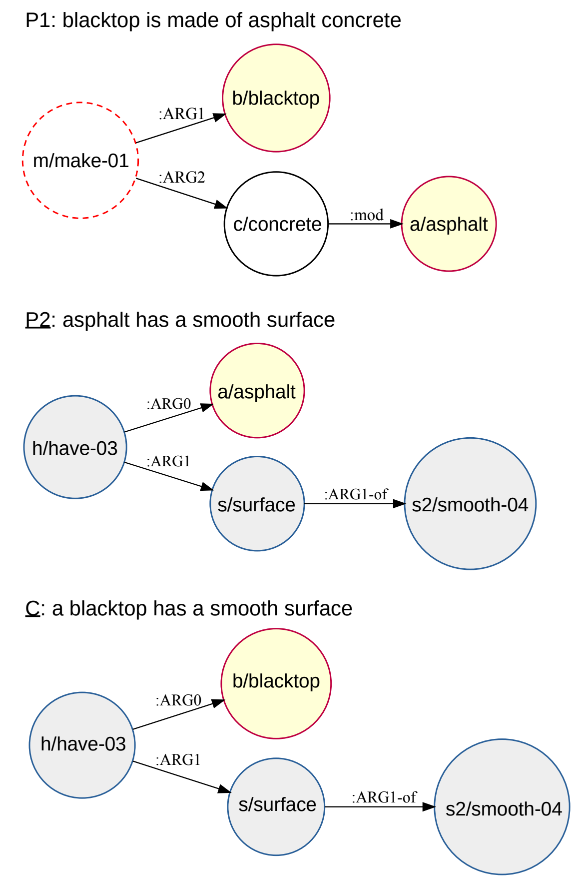

Figure 6: AMR argument substitution (property inheritance) (ARG-SUB-PROP).

B Experimental Details

B.1 Dataset

Table 9 describes the statistical information of the corpus used in the experiment. For experiments: Section 5.1, 5.2, and 5.3, the EntailmentBank dataset is split into train 60%, valid 20%, and test 20% sets. For the explanation inference retrieval task in Section 5.3, we follow the same experimental setup provided online. [4]

Corpus	Num data.	Avg. length
WorldTree (Jansen et al., 2018a)	11430	8.65
EntailmentBank (Dalvi et al., 2021)	5134	10.35

Table 9: Statistics from explanations datasets. WorldTree is used in the Explanation Inference Retrieval task.

B.2 T5 Bottleneck Architecture

Figure 7 shows the architecture of the T5 bottleneck for learning latent sentence space. It includes two stages: sentence embedding and decoder connection. The sentence embedding aims to transform token embeddings into a sentence (single) embedding. Decoder connection aims to connect the encoder and decoder.

Latent sentence space: While designing the sentence bottleneck, we compare the four most frequently used mechanisms to transform token embeddings into sentence embeddings:

(1) Mean pooling: calculating the mean of each dimension on all token embeddings and feeding the resulting vector into a multi-layer perceptron to obtain the sentence embedding. (2) multi-layer perceptron (MLP): applying an MLP to reduce the dimensionality of token embeddings, and the resulting embeddings are concatenated to form a single sentence embedding: $z = \text{concat}\left[\text{MLP}_1(x_1); ...; \text{MLP}_T(x_T)\right]$ where $\text{MLP}_i(x_i)$ represents the i-th neural network for input representation of token x_i, z is the latent sentence representation, and T is the maximum token length for a sentence. (3) multi-head attention: feeding each token embedding into the multi-head attention and considering the first output embedding as the sentence embedding (Montero et al., 2021): $z = \text{MultiHead}\left(XW^q, XW^k, XW^v\right)[0]$ where $X = [x_1, ..., x_T]$ and W^q, W^k, and W^v are the weights for learning q, k, v embeddings in self-attention, respectively. (4) Sentence T5: re-loading the pre-trained sentence T5 (S-T5, Ni et al. (2021)).

Conditional generation: Next, we consider four strategies to inject sentence embeddings into the decoder. (1) Cross-attention input embedding (CA Input): reconstructing the token embeddings from a sentence representation and directly feeding them

[4] `https://github.com/ai-systems/hybrid_autoregressive_inference`

into the cross-attention layers of the decoder: $\hat{Y} = \text{MultiHead}\left(YW^q, \text{MLP}(z)W^k, \text{MLP}(z)W^v\right)$ where $\hat{Y}$ is the reconstruction of decoder input sequence $Y = [y_1, ..., y_K]$. (2) Cross-attention KV embedding (CA KV): instead of reconstructing the token embeddings, it consists of directly learning the Key and Value (Hu et al., 2022; Li et al., 2020), which is formalised as $\hat{Y} = \text{MultiHead}\left(YW^q, \text{MLP}_k(z), \text{MLP}_v(z)\right)$, where MLP_k and MLP_v are neural layers for learning k v embeddings. (3) Non-cross-attention input connection (NCA Input): reconstructing the token embeddings and element-wisely adding them with the input embeddings of the decoder (Fang et al., 2021). (4) Non-cross-attention output connection (NCA Output): adding the reconstructed token embeddings to the output embedding of the decoder.

Train: architecture					
Decoder Connection		CA Input	CA KV	NCA Input	NCA Output
Sentence Embedding	Pooling	1.41	1.44	1.86	2.42
	MLP	1.71	1.94	2.09	2.62
	MHA	1.51	2.24	2.31	3.03
	S-T5	**1.24**	1.42	1.81	2.22

Table 10: Comparison of different setups on test loss via cross-entropy (CA: cross-attention, NCA: non-cross-attention), bottom: comparison of different baselines on EntailmentBank testset.

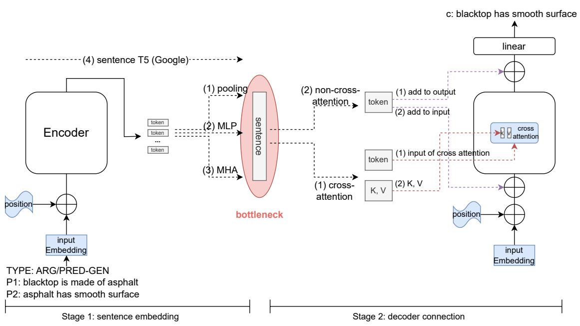

Figure 7: The architectural configuration of T5 bottleneck, it consists of two stages: sentence embedding and decoder connection.

B.3 Implementation Details

Hyper-parameters. 1. Size of Sentence Representation: in this work, we consider 768 as the size of the sentence embedding. Usually, the performance of the model improves as the size increases. **2.** Multi-head Attention (MHA): in the experiment, MHA consists of 8 layers, each layer containing 12 heads. The dimensions of Query, Key, and Value are 64 in each head. The dimension of token embedding is 768. Training hyperparameters are: **3.** For all models, the max epoch: 40, learning rate: 5e-5. During fine-tuning the T5 bottleneck, we first freeze the pre-trained parameters in the first epoch and fine-tune all parameters for the remaining epochs. **4.** All models are trained on a single A6000 GPU device.

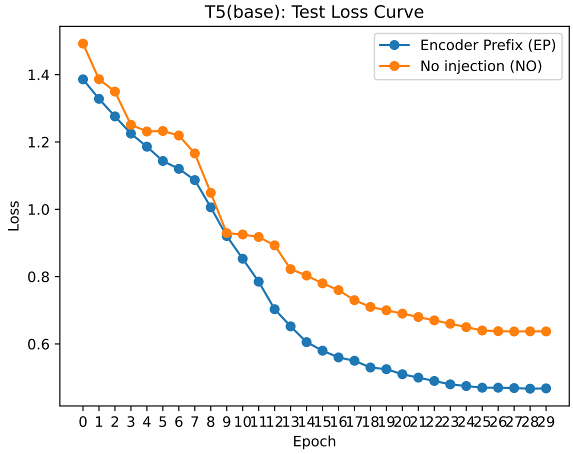

Figure 8: The test loss curve indicates that EP facilitates better convergence, indicating the supervision on inference types aligns the model's reasoning trajectory with target inference behaviours, improving conclusion prediction accuracy.

Baselines. In the experiment, we implement five LSTM-based autoencoders, including denoising AE (Vincent et al. (2008), DAE), β-VAE (Higgins et al., 2016), adversarial AE (Makhzani et al. (2015), AAE), label adversarial AE (Rubenstein et al. (2018), LAAE), and denoising adversarial autoencoder (Shen et al. (2020), DAAE). Their implementation relies on the open-source codebase available at the URL [5]. As for transformer-based VAEs, we implement Optimus (Li et al., 2020)[6] and Della (Hu et al., 2022)[7]. All baseline models undergo training and evaluation with the hyper-parameters provided by their respective sources. A latent dimension of 768 is specified to ensure a uniform and equitable comparative analysis.

Metrics. To evaluate the generated conclusions against the reference conclusions, we employ BLEU scores for 1- to 3-gram overlaps and report

[5] https://github.com/shentianxiao/text-autoencoders
[6] https://github.com/ChunyuanLI/Optimus
[7] https://github.com/OpenVLG/DELLA

the average score. Additionally, to assess semantic similarity, we calculate the cosine similarity between the generated and reference conclusions by encoding both using the pretrained Sentence-T5 model[8] and computing the cosine similarity of their resulting embeddings.

C Complementary Results

Ablation studies. We remove the inference types from the dataset and evaluate the T5 model performance using the same metrics. In this case, we can compare the model performance trained with or without that inference type. From Table 11, we can observe that the baselines (T5 small and base) achieve higher BLEU and BLEURT scores without the data with ARG-INS, COND-FRAME, and UNK inference type, respectively. This result indicates that the T5 cannot generalize well over those inference types. Also, removing the UNK inference type from data can achieve lower loss and PPL, which indicates that it has a negative impact on model training.

Remove	T5	BLEU	BLEURT	Cosine	Loss ↓	PPL ↓
FRAME-SUB	small	0.50	0.19	0.95	0.95	2.58
	base	0.60	0.33	0.96	0.72	1.95
ARG-INS	small	**0.54**	**0.27**	0.95	0.82	2.22
	base	**0.63**	**0.46**	0.97	0.64	1.73
FRAME-CONJ	small	0.53	0.26	0.96	0.84	2.28
	base	0.60	0.35	0.96	0.65	1.76
COND-FRAME	small	**0.55**	**0.25**	0.96	0.88	2.39
	base	**0.59**	**0.36**	0.96	0.69	1.87
UNK	small	**0.55**	**0.23**	0.95	<u>0.53</u>	<u>1.44</u>
	base	**0.62**	**0.40**	0.96	<u>0.58</u>	<u>1.57</u>
No	small	0.54	0.22	0.96	0.69	2.22
No	base	0.57	0.33	0.96	0.61	1.65

Table 11: Ablation study over inference type (No: no inference types are removed).

More controllable inference examples. We provide more controlled examples based on both the Original T5 and T5 bottleneck in Table 12, 13, and 16. All examples reveal that the inference type can provide quasi-symbolic inference control to language models.

Quasi-symbolic NLI control

P1: a pumpkin contains seeds
P2: fruit contains seeds

Original T5:
ARG-INS: a fruit in a pumpkin contains seeds
FRAME-CONJ: a pumpkin and fruit both contains seeds
FRAME-SUB: fruit is a kind of pumpkin

T5 bottleneck:
ARG-INS: fruit is a part of pumpkin that contains seeds
FRAME-CONJ: a fruit contains seeds
FRAME-SUB: a pumpkin is a kind of plant

Table 12: Controlled generation. original T5(base) (top) and T5 bottleneck (bottom).

Quasi-symbolic NLI control

P1: eating something has a negative impact on that something
P2: some animals eat cacti
ARG-INS: some animals have a negative impact on cacti by eating cacti
PRED-SUB: some animals may have a negative impact on cacti
FRAME-SUB: eating cacti has a negative impact on that cacti

ARG-INS: some animals have a negative impact on cacti by eating cacti
PRED-SUB: animals have a negative impact on cacti
FRAME-SUB: eating cacti has a negative impact on that cacti

Table 13: Controlled generation. original T5(base) (top) and T5 bottleneck (bottom).

Qualitative evaluation for LLM evaluators. We conduct a qualitative evaluation through manual inspection. However, this assessment is not systematic or rigorously structured as we discussed in the Limitations section. Tables 14 and 15 present examples with discrepancies in scores between ChatGPT4o and GPT4o-mini, as well as a comparison of predictions between encoder prefix injection (EP) and the absence of inference-type injection (NO), respectively.

From both tables, we observe that ChatGPT4o tends to be more accurate than GPT4o-mini and that EP outperforms NO in generating correct predictions.

[8] https://huggingface.co/sentence-transformers/sentence-t5-base

Premises	**Prediction(NO)**	**Golden**	**ChatGPT4o**	**GPT4o-mini**	**Human Check**
p1: the metal on the roof of a car is in contact with air p2: air contains oxygen and water vapor	the car roof contains water vapor and oxygen	the metal on the roof of a car is in contact with oxygen and water vapor	0	1	0 (invalid predicate "contains")
p1: friction occurs when the student is rubbing his hands together p2: friction causes the temperature of an object to increase	rubbing your hands together causes the temperature of the object to increase	friction causes the temperature of student's hands to increase	1	0	1 (replacing "friction" with "rubbing hands together")
p1: a caterpillar is a kind of insect p2: metamorphosis is when an insect changes from an immature form to an adult form	metamorphosis is when a caterpillar changes from an immature form to an adult form	an example of metamorphosis is when a caterpillar changes from an immature form to an adult form	1	0	1 (replacing "insect" with "metamorphosis")
p1: an increase in water has a positive impact on alligators p2: a flood is caused by an increase in water	a flood has a positive impact on alligators	a flood has a positive impact on alligators	1	0	1 (exact match)
p1: predators eat prey p2: eating prey requires catching prey	predators catching prey requires catching prey	predators must catch prey to eat prey	0	1	0 (fail to do substitution between "eating" and "catching")
p1: a leaf uses chlorophyll to produce carbohydrates p2: carbohydrates are made of sugars	a leaf uses chlorophyll to produce sugars	a leaf uses chlorophyll to produce sugar	0	1	1 (valid inference)
p1: salt is a kind of pure substance p2: pepper is a kind of substance	salt and pepper are kinds of substances	salt and pepper are two substances	1	0	1 (valid conjunction both "salt" and "pepper")
p1: some bacteria are good for humans by helping digestion p2: digestion is when stomach acid breaks down food	some bacteria help digestion by breaking down food	some bacteria are good for humans by helping to break down food	0	1	0 (without "good for human")
p1: a doorbell is a kind of electric device p2: closing an electric circuit causes an electrical device to function	closing a doorbell causes the doorbell to function	an electric circuit causes a doorbell to function	0	1	0 (incorrect replacing "electric circuit" with "doorbell")
p1: burning something is a kind of action p2: if something occurs by performing an action, then that action is required for that something to occur	burning something is required for something to occur	if something occurs by burning something else, then burning that something else is required for that something to occur	1	0	1 (valid inference by replacing "action" with "burning something")
p1: echolocation can be used for hunting prey p2: bats must emit sound in order to echolocate	bats must emit sound in order to hunt prey	bats must emit sound in order to hunt their prey	1	0	1 (valid inference by replacing "echolocate" with "hunting prey")
p1: different solids will have the same physical properties p2: an mixture is formed by two or more substances combined together physically	one solid will form a mixture	different solids that are combined will become a mixture	1	0	0 (incorrect "one solid")

Table 14: Qualitative evaluation for examples with discrepancies in scores between ChatGPT4o and GPT4o-mini (NO: no inference type injection, 0: invalid, 1: valid). We can observe that the ChatGPT4o tends to be more accurate than GPT4o-mini by human check.

Premises	Prediction(NO)	Prediction(EP)	Golden	ChatGPT4o	Human Check
p1: the metal on the roof of a car is in contact with air p2: air contains oxygen and water vapor	the car roof contains water vapor and oxygen	the car roof is in contact with oxygen and water vapor	the metal on the roof of a car is in contact with oxygen and water vapor	NO:0, EP:1	NO:0, EP:1
p1: a beak is used for catching food by some birds p2: eating food requires catching food	ads are used for eating by birds to catch food	a beak is used for eating by some birds	a beak is used for eating food by some birds	NO:0, EP:1	NO:0, EP:1
p1: predators must catch prey to eat prey p2: a predator is a kind of animal	animals must catch and eat prey	animals must catch prey to eat prey	some animals must catch prey to eat	NO:0, EP:1	NO:0, EP:1
p1: an adaptation is a kind of change p2: adapting is when a living thing responds to a change in an environment	an adaptation is something a living thing responds to a change in an environment	adaptation is when a living thing responds to a change in an environment	an adaptation is a kind of change in response to a change in an environment	NO:0, EP:1	NO:0, EP:0
p1: a doorbell is a kind of electric device p2: closing an electric circuit causes an electrical device to function	closing a doorbell causes the doorbell to function	closing an electric circuit causes a doorbell to function	an electric circuit causes a doorbell to function	NO:0, EP:1	NO:0, EP:1
p1: green plants are made of plant cells p2: green plants are a kind of producer	a producer is made of plant cells	producers are made of plant cells	producers are made of plant cells	NO:1, EP:0	NO:1, EP:1
p1: the iron nail has rusted p2: rusting is when iron chemically reacts with water and oxygen	iron nails rusting is when the iron nails chemically react with water and oxygen to form iron nail	the iron nail has rusted	a chemical reaction has happened on the iron nail	NO:0, EP:1	NO:0, EP:0
p1: wood burns p2: wood chips are made of wood	wood burns when introduced to wood	wood chips burn	wood chips burn	NO:0, EP:1	NO:0, EP:1
p1: some bacteria are good for humans by helping digestion p2: digestion is when stomach acid breaks down food	some bacteria help digestion by breaking down food	some bacteria are good for humans by helping digestion by breaking down food	some bacteria are good for humans by helping to break down food	NO:0, EP:1	NO:0, EP:1
p1: a lung is a kind of organ for directly breathing in gasses p2: smoke is a kind of gas	a lung is an organ directly for directly breathing in smoke	a lung is a kind of organ for directly breathing in smoke	lungs can be used for directly breathing in smoke	NO:0, EP:0	NO:1, EP:1
p1: the hot egg will decrease in temperature p2: the cold water will increase in temperature	the hot egg and cold water will increase in temperature	the hot egg will decrease in temperature and the cold water will increase in temperature	the water will increase in temperature and the egg will decrease in temperature	NO:0, EP:1	NO:0, EP:1
p1: plant reproduction requires pollinating animals for pollination p2: a bee can help on pollination by carrying pollen	plants reproduction requires bees that carry pollen	a bee can help plant reproduction by carrying pollen	a bee can help on pollination in plant reproduction by carry pollen	NO:0, EP:1	NO:0, EP:1
p1: a leaf uses chlorophyll to produce carbohydrates p2: carbohydrates are made of sugars	a leaf uses chlorophyll to produce sugars	a leaf uses chlorophyll to produce sugar	a leaf uses chlorophyll to produce sugar	NO:1, EP:1	NO:1, EP:1

Table 15: Qualitative evaluation for prediction through EP and NO (NO: no inference type, EP: encoder prefix), we can observe that EP outperforms NO in generating correct predictions.

Premises	Inference Type	T5 original
P1: a pumpkin contains seeds	ARG-INS	a fruit in a pumpkin contains seeds
P2: fruit contains seeds	FRAME-CONJ	a pumpkin and fruit both contain seeds
	IFT	if a pumpkin contains fruit then the fruit may contain seeds
	EXAMPLE	fruit is an example of pumpkins being sown
	ARG/PRED-GEN	a pumpkin is a kind of fruit
	ARG-SUB	fruit can contain pumpkin seeds
	UNK	a pumpkin can contain seeds
	FRAME-SUB	fruit is a kind of pumpkin
P1: sunlight is a kind of solar energy	ARG-INS	solar energy is a kind of resource for plants that uses water
P2: water and sunlight are resources for plants	FRAME-CONJ	water and sunlight are resources for plants and are kinds of solar energy
	UNK	the resources for plants include water and solar energy
	ARG-SUB	water and solar energy are resources for plants
P1: to move something can mean to transfer something	ARG-SUB	flowing can mean to transfer energy
P2: flowing is a kind of movement for energy	INF	if something flows, then that energy will flow
	FRAME-CONJ	moving can transfer energy and mean flowing
	ARG-INS	flowing can be a kind of transfer of energy to another entity
	ARG/PRED-GEN	transferring energy with flowing can be seen as transferring energy
P1: if a pot is exposed to a stove then that pot may become hot	COND-FRAME	the pot will become hot
P2: the pot is exposed to a stove	EXAMPLE	an example of hot pot is a pot exposed to a stove
P1: eating something has a negative impact on that something	FRAME-SUB	eating cacti has a negative impact on that cacti
P2: some animals eat cacti	PRED-SUB	some animals may have a negative impact on cacti
	ARG-INS	some animals have a negative impact on cacti by eating cacti
	EXAMPLE	cooking cacti is an example of a negative impact on a cactus
	INF	if a cactus has a negative impact on an animal, that cactus could be devoured
P1: seeing requires light	ARG-SUB	reading requires light
P2: reading requires seeing	ARG-INS	light is a kind of requirement for reading
	INF	if light is moving then reading may be taken
	EXAMPLE	a light bulb will be used for reading
	UNK	light will help you read

Premises	Inference Type	T5 bottleneck
P1: a pumpkin contains seeds	ARG-INS	fruit is part of a pumpkin that contains seeds
P2: fruit contains seeds	FRAME-CONJ	a fruit contains seeds
	FRAME-SUB	a pumpkin is a kind of plant
P1: sunlight is a kind of solar energy	ARG-INS	water is a kind of resource that is used by plants for growth
P2: water and sunlight are resources for plants	FRAME-CONJ	plants and water are resources that require water and energy
	ARG-SUB	plants use water and sunlight to produce energy
P1: to move something can mean to transfer something	ARG-SUB	flowing can mean to transfer energy
P2: flowing is a kind of movement for energy	INF	if something flows, then that energy will flow
	FRAME-CONJ	moving can transfer energy and mean flowing
	ARG-INS	flowing can be a kind of transfer of something
	ARG/PRED-GEN	transferring energy with flowing can be seen as transferring energy
P1: if a pot is exposed to a stove then that pot may become hot	COND-FRAME	the pot may become hot
P2: the pot is exposed to a stove	ARG/PRED-GEN	the pot may be a source of heat
P1: eating something has a negative impact on that something	FRAME-SUB	eating cacti has a negative impact on that cacti
P2: some animals eat cacti	PRED-SUB	animals have a negative impact on cacti
	ARG-INS	some animals have a negative impact on cacti by eating cacti
P1: seeing requires light	ARG-SUB	reading requires light
P2: reading requires seeing	FRAME-CONJ	reading and feeling can both be used
	INF	if something is visible then that something will be seen

Table 16: controllable NLI via inference type (Top: original T5, bottom: T5 bottleneck).

Algorithm 1 Annotation procedure

1: Find premise P_x most similar to the conclusion C, $P_{\bar{x}}$ being the other premise.
2: $G_{x,\bar{x},C} \leftarrow$ AMR graph of $P_x, P_{\bar{x}}, C,$ respectively.
3: # - - - - - - - - - - - - - - - - - - - common ARG-SUB, PRED-SUB -
4: **if** $G_x = G_c$ or $G_{\bar{x}} = G_c$ **then**
5: $type = PREM\text{-}COPY$ # Comment: no reasoning happen.
6: **else if** P_x and C differ by one word w **then** # Comment: common ARG(PRED)-SUB.
7: **if** w is a verb **then**
8: $type = PRED\text{-}SUB$
9: **else**
10: $type = ARG\text{-}SUB$
11: **end if**
12: **else**
13: # - - - - - - - - - - - - - - - - - COND-FRAME, FRAME-SUB, ARG-SUB-PROP - - - - - - - - - - - - - -
14: Get AMR graphs G_1, G_2, G_c for P_1, P_2 and C respectively. $P_x \rightarrow G_x$.
15: **if** $\exists$:ARG*$(x, a) \in C$ and $a \in P_{\bar{x}}$ **then**
16: **if** $\exists$:condition($root(G_x)$, $root(G_{\bar{x}})$) **then**
17: # Comment: see Figure 4, two root nodes are connected by :condition edge
18: $type = COND\text{-}FRAME$
19: **else if** $root(a)$ is a noun **then**
20: **if** $root(G_{\bar{x}})$ = "make-01" and $\exists$:ARG*($root(G_{\bar{x}})$, a) **then**
21: # Comment: "make" as a trigger to classify ARG-SUB and property inheritance.
22: $type = ARG\text{-}SUB\text{-}PROP$
23: **else**
24: $type = ARG\text{-}SUB$ # ARG-SUB that was not caught by the simpler rule on line 10, due to Px differing from C by more than a single word
25: **end if**
26: **else**
27: $type = FRAME\text{-}SUB$
28: **end if**
29: # - - - - - - - - - - - - - - - - - - - Further-specification and Conjunction -
30: **else if** $G_x \subset G_c$ and $G_{\bar{x}} \subset G_C$ **then**
31: $type = FRAME\text{-}CONJ$
32: **else if** $\exists x, y$:domain($root(G_x)$, x) and :domain($root(G_{\bar{x}}$, y) and :op*("and", x) $\in G_c$ and :op*("and", y) $\in G_c$ **then** # Comment: using connectives 'and' to connect two premises
33: $type = FRAME\text{-}CONJ$
34: **else if** $G_x \subset G_c$ **then**
35: $d \leftarrow G_c - G_x$
36: **if** $root(d)$ is a noun **then**
37: $type = ARG\text{-}INS$ # Comment: inserting an argument.
38: **else**
39: $type = FRAME\text{-}INS$ # Comment: inserting a phase (also annotated as ARG-INS).
40: **end if**
41: # - ARG/PRED-GEN and Others -
42: **else if** $\exists$:domain($root(G_c)$, y) and ($root(G_c) \in G_x$ and $y \in G_{\bar{x}}$) or ($root(G_c) \in G_{\bar{x}}$ and $y \in G_x$) **then**
43: $type = ARG/PRED\text{-}GEN$
44: **else**
45: $type = UNK$
46: **end if**
47: **end if**

Prompts for automatic evaluation

Consistency:
You are a scoring expert in natural language reasoning. Given two premises and a conclusion, your goal is to evaluate whether the conclusion violates the premises. During your inference process, please only consider the information from the premises.
you can directly give your score (0 or 1) based on the following criteria:
0: the conclusion violates the premises.
1: the conclusion doesn't violate the premises.

The output format is just the score. You don't need to analyse the reasoning process.

Alignment:
You are a scoring expert. Given two premises, a conclusion, and an inference type, your goal is to evaluate whether the (premises, conclusion) pair is aligned with the inference type.

The following is the description of 10 inference types:
1. ARG-SUB: the conclusion is obtained by replacing one argument with another argument.
2. PRED-SUB: the conclusion is obtained by replacing one verb with another verb.
3. FRAME-SUB: the conclusion is obtained by replacing a frame of one of the premises with one from the other premise.
4. COND-FRAM: the conclusion is obtained according to the conditional premise with keyword "if".
5. ARG-INS: the conclusion is obtained by connecting an argument from one of the premises to a frame of the other.
6. FRAME-CONJ: the conclusion is obtained by using connectives to connect two premises.
7. ARG/PRED-GEN: a new ":domain" relation frame is created in the conclusion if both premise graphs differ by a single predicate/argument term.
8. ARG-SUB-PROP: one of the premises describes a "is made of" relationship between the entity in the other premise and its replacement.
9. IFT: the conclusion should be a conditional sentence.
10. EXAMPLE: the conclusion should contain the keyword "example".

When evaluating, some premises might not be able to deduce more than one conclusions. You can ignore those cases.

Finally, you can directly give your score (0 or 1) based on the following criteria:
0: the (premises, conclusion) pair is not aligned with the inference type.
1: the (premises, conclusion) pair is aligned with the inference type.

The output format is just the score. You don't need to analyse the reasoning process.

Table 17: Empirically designed prompt for automatically evaluating the controllability in Section 5.2.