Neuro-symbolic Training for Reasoning over Spatial Language

Anonymous ACL submission

Abstract

Spatial reasoning is a fundamental aspect of human cognition, essential for everyday tasks. It is also crucial for machines to interact with their environment in a human-like manner. However, recent research shows that even state-of-the-art language models struggle in spatial reasoning, especially in unobserved situations with complex input compositions. This is attributed to not achieving the right level of abstraction required for their generalizability. To alleviate this issue, we propose training with neuro-symbolic techniques that exploit the spatial logical rules as constraints, providing the model with additional supervision. Training models to adhere to spatial reasoning rules guides them to make more effective abstractions for generalizability and transfer learning. We evaluate our approach on various spatial reasoning benchmarks over text. Our results demonstrate the effectiveness of our neuro-symbolic training in domain transfer and complex multi-hop spatial reasoning across multiple language model backbones.

1 Introduction

Spatial reasoning is essential for humans cognition and also plays a crucial role in many AI applications, including language grounding (Zhang et al., 2021), computer vision (Zhang and Kordjamshidi, 2022a; Liu et al., 2023), robotics (Sisbot et al., 2007; Yadollahi et al., 2023; Zhang and Kordjamshidi, 2022b) and even more specific fields such as medical domain (Atif et al., 2007; Datta et al., 2020; Gong et al., 2023). Large Language models (LLMs) have been widely applicable in many problems in these areas and, in some cases, show human level performance (OpenAI, 2023; Brown et al., 2020). However, recent studies highlight the shortcomings in multi-hop reasoning over text (Bang et al., 2023; Yang et al., 2023; Mirzaee and Kordjamshidi, 2023) and in many downstream applications (Liu et al., 2023; Chen et al., 2024) which calls for more attention to this topic. This paper focuses on one essential domain: spatial reasoning over text. Improvements in this area could encourage advancements in more complex domains.

In utilizing explicit logic to address multi-hop spatial reasoning in textual tasks, Yang et al. 2023 integrated LLMs with a neuro-symbolic framework by passing formal logical representations extracted by LLMs to the Answer Set Programming for conducting logical reasoning. Similarly, Mirzaee and Kordjamshidi 2023 passed formal representations to Prolog to solve reasoning tasks. These approaches rely on external tools and require formal representations at inference time. Moreover, formalizing commonsense knowledge for all types of reasoning is particularly infeasible in realistic domains, limiting their effectiveness only to synthetic benchmarks. We propose a feasible and effective method to address the limitations of utilizing spatial logic. Our approach uses a more generic neuro-symbolic framework to enhance spatial reasoning in LLMs, especially in complex and unobserved scenarios. By fine-tuning the language models with our techniques, we leverage spatial logical rules to guide the level of abstraction captured during training.

C: There are three blocks called A, B, and C. A is to the left of B and B is to the left of C. In A, there is a small yellow square near and to the left of a large yellow circle. The large circle is near and to the left of a large blue triangle. In B, there is a small yellow square near and below a large blue triangle. There is a medium black triangle near and above the large blue triangle. In C, there is a small black square far to the left and above a small blue circle. The blue circle is near and to the left of a large yellow square."

q1: Is the small black thing to the right of medium black thing?

a1:Yes

q2: What is the position of small black thing relative to medium black thing.

a2: Right

Figure 1: An example of story and questions of SQA. q_1 is Yes-No and q_2 is Find relation type of questions.

In particular, we train the models to minimize both cross-entropy loss and violation from logical constraints. We hypothesize that incorporating logical knowledge as supervision enriches the models' abstraction capabilities, further enhancing generalizability. A key advantage of our approach is that it does not require full access to logical knowledge. Any partially available knowledge can be utilized during training. Additionally, formal knowledge is not needed at inference time, which is crucial because generating formal representations during inference is a challenging task itself. Our technique also avoids reliance on external reasoning tools, which can add computational complexity in real-time applications.

We select three benchmarks, SpartQA-Human (Mirzaee et al., 2021), ReSQ (Mirzaee and Kordjamshidi, 2022), and StepGame (Shi et al., 2022) to evaluate our proposed method. Our improved results confirm our hypothesis about the impact of neuro-symbolic training on generalizability. The contribution[1] of this paper can be summarized as follows, **(1)** We propose to exploit the available logical spatial knowledge in tuning language models to deal with their lack of spatial reasoning indicated by previous research. **(2)** We provide extensive experimental results and analysis on the existing benchmarks. Our results consistently confirm the effectiveness of our proposed approach in both encoder-based and generative language models and their generalizability for transferring knowledge to different domains. **(3)** We show that the smaller models with our proposed method generalize better on out-of-domain and complex reasoning compared to using prompt-engineered larger models.

2 Related Works

Investigating the reasoning capability of NLP models has become a trending research topic given the instability in the performance of LLMs (Gendron et al., 2023; Valmeekam et al., 2023; Feng et al., 2024; Chen et al., 2024; Bang et al., 2023). Usually, this ability is measured via question-answering benchmarks (Weston et al., 2015; Zhou et al., 2020; Tan et al., 2023). However, fewer studies are focusing on spatial reasoning over text. Recent benchmarks, such as SpaRTUN (Mirzaee and Kordjamshidi, 2022) and StepGame (Shi et al., 2022; Li et al., 2024) datasets, are created to address this issue by providing evaluation resources.

Some studies report the performance of LLMs on benchmarks mentioned above and highlight their struggle with reasoning, even on synthetic datasets (Yang et al., 2023; Bang et al., 2023), especially in multi-hop reasoning tasks. Several works have attempted to improve LLMs' spatial reasoning capabilities. Mirzaee and Kordjamshidi 2022 employs fine-tuning on synthetic data, improving multi-hop reasoning of language models even when applied to realistic domains. Another approach designed a specialized memory network based on a recurrent neural network for deep reasoning tasks, but it was less effective than tuning pre-trained language models (PLMs)(Shi et al., 2022). Some papers also focus on enhancing in-context learning. For example, Sharma 2023 proposes prefix-based prompting that retrieves specific few-shot examples to transfer knowledge from simple to complex spatial relations. Meanwhile, Hu et al. 2023 nodifies the Chain-of-Thought (CoT) (Wei et al., 2023),by replacing the textual explanation with symbolic representation. While this method improved spatial understanding in simple environments, it is less applicable to settings with diverse, complex relations. Some studies employ neuro-symbolic modeling using a pipeline of LLMs and external logical solvers. For instance, Yang et al. 2023 used Answer Set Programming (ASP) as the logical solver, while Mirzaee and Kordjamshidi 2023 employed Prolog. These approaches extract relations and obtain predicate-argument logical forms using language models, separating the extraction from the reasoning, which is handled by external solvers. While this method significantly improves multi-hop reasoning, it requires formal representations at inference time, which is challenging. Furthermore, formalizing commonsense knowledge for all types of reasoning is infeasible in realistic domains.

Unlike the previous studies, we exploit spatial logical knowledge in fine-tuning language models. Our approach utilizes synthetic data and logical representations solely during training, allowing generalizability to realistic domains. Despite models with external tools in the pipeline, our neuro-symbolic method does not require a comprehensive set of reasoning rules; any partial set of rules can effectively supervise the model. Many studies emphasize the usefulness of logical knowledge to solve question-answering tasks (Prager et al., 2004; Hu et al., 2020; Amizadeh et al., 2020; Sun

[1] all the code will be available upon acceptance

et al., 2022). We utilize logical knowledge in the form of logical constraints. Similar approaches have been used to solve multiple NLP problems. For example, Lu et al. 2021 uses a heuristic function inspired by the A* algorithm to restrict the generator given a set of constraints. While Qin et al. 2022 uses a sampling method looking for the most optimal solution that satisfies the constraints. However, both of these techniques apply the constraints during inference, which differs from what we propose for using them during the training. There is also the benchmark that investigates the benefits of logical constraints in several NLP problems (Faghihi et al., 2023) under the DomiKnowS framework (Faghihi et al., 2021). Though we use techniques previously proposed to integrate the logic, our approach utilizes constraints solely during training, eliminating the need for access to logical constraints during inference where using them might be problematic for realistic domains and real-time applications.

3 Methodology

3.1 Problem Definition

For the spatial QA task, the inputs are a textual context C and a question Q about the spatial relations between objects in the scene description. The output is the answer(s) to the question, denoted as A. There are two types of questions/answers: Yes/No (YN) and find-relation (FR). The answer domain is restricted to $A = \{Yes, No\}$ for YN questions. The answers to FR questions depend on the dataset. An example of the answer domain for FR is $A = \{left, right, above, below, \ldots\}$. More details on the FR answers are provided in the experimental section.

3.2 Backbone Language Models

We selected two types of language models: encoder-based models from the BERT family (Devlin et al., 2019) and generative models like Flan-T5 (Chung et al., 2022) to assess the effectiveness of our proposed fine-tuning approach. Additionally, we compare the performance of LLMs, including the GPT family and Llama3-8B, on these tasks with our proposed approach.

3.2.1 Encoder-based Language Models

We utilize BERT as the backbone architecture, consistent with previously reported results (Mirzaee and Kordjamshidi, 2022). We fine-tune BERT by adding classification layers for QA tasks. We employ a binary classification layer for the YN domain while using multiple binary classification layers for the FR domain. The intention behind using multiple binary classifications instead of multi-label classification is that this task involves multi-label answers; one question can have multiple labels as responses. The number of binary classification layers in the FR domain depends on the number of possible answers, $|A|$. The input to the BERT model is constructed by concatenating the question, Q, separate token, and the context C. After feeding the input to BERT, we use the $[CLS]$ token from the last layer as the input to the classification layer(s) for final answer prediction. Note that we are aware that BERT is not the strongest baseline in the BERT family. Therefore, we provide results from the same setting as the BERT using RoBERTa in the Appendix. The overall results are slightly lower than BERT for transfer learning. So, we decided to use only BERT for the remaining experiments.

3.2.2 Generative Language Models

We select Flan-T5 (Chung et al., 2022) as the generative baseline and use this open-source model for fine-tuning. To reduce computational costs by decreasing the number of training parameters, we employ the LoRA adapter (Hu et al., 2021). In the YN setting, processing the output is straightforward, as the first token can represent the response directly generated by the model. We select the highest probability between the Yes and No tokens from the first token of the Flan-T5 output as the answer to the input question. The input prompt to the Flan-T5 model is structured as follows: "Answer based on the context: C Q." However, this approach is not feasible in the FR setting due to the multi-label questions requiring multiple answers.

For In-context Learning, we select Llama3-8B (AI@Meta, 2024), GPT3.5 (Brown et al., 2020), and GPT-4 (OpenAI, 2023) as the backbone LLMs for prompt engineering. Then, we apply *Zero-shot*, *Few-shot* as the baseline of prompt-based learning.

Zero-shot. We give the prompt to LLMs for answers based on the scene description and the question without having any examples.

Few-shot. We randomly select four questions from the training set and append the information about the response format for each question with the corresponding label. We provide these examples, scene description, and the target question to

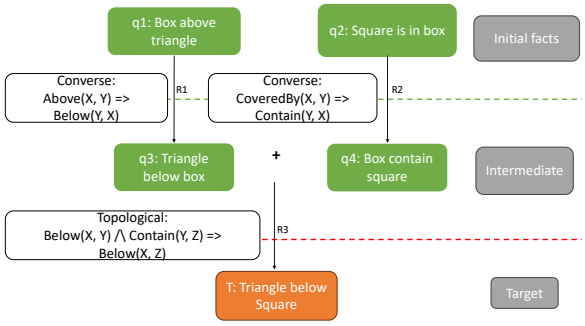

Figure 2: The Q-*chains* are created for all training examples and used in the training phase only. This figure shows one chain of reasoning (Q-*chain*). R_i refers to rule i, and q_j refers to factual sentence j. Each factual sentence is turned into a question like "Is a triangle below a square?" The initial facts are a part of the original context and are provided in the training dataset. The detail of Q-*chain* creation is provided in Section 3.3.

LLMs to query the answer.

3.3 Training with Spatial Logic

To address the challenge of multi-hop spatial reasoning, we employ symbolic knowledge during training. We assume a formal knowledge base (KB) of rules, referred to as *spatial logic*, is given during training time. This *spatial logic* does not need to be exhaustive or cover all reasoning aspects; any available knowledge can be used to enhance the reasoning capabilities of the language model. This work uses 79 reasoning rules from (Mirzaee and Kordjamshidi, 2022). These rules are divided into four categories: converse, inverse, transitive, and transitive + topological. The KB covers rules between 15 spatial relations, including directional and topological relations, such as "*If above*(x, y) *then below* (y, x)." $Above(x, y) \Rightarrow Below(y, x)$ can represent this example rule. Our main hypothesis is that supervision from high-level logical knowledge allows the model to capture more abstract patterns, thereby improving generalization to other domains.

To exploit the spatial logic in the training phase, we follow three steps: **(1)** Creating the chain of reasoning for each training example. **(2)** Obtaining consistency constraints from the chain of reasoning. **(3)** Incorporating constraints in the loss function.

Creating example-specific rules. We begin exploiting *Spatial Logic* by automatically augmenting the training questions with a chain of questions, denoted as Q-$Chain$. This process is illustrated in Figure 2. We infer the answer to the final query from the input context using the resolution tree, which outlines the logical steps for implication. Our synthetic training data (e.g., SpaRTUN) provides logical representations of the initial facts from the context. We begin building the tree with initial facts: "The box is above the triangle," denoted as q_1, and "The square is in the box," denoted as q_2. Next, we apply a forward chaining approach to identify the relevant rules: spatial converse rules R_1 for q_1 and R_2 for q_2. This approach yields the intermediate inferred facts: q_3, "The triangle is below the box," from applying R_1 to q_1, and q_4, "The box contains the square," from applying R_2 to q_2. Finally, we search for the appropriate spatial rule to combine q_3 and q_4, resulting in the target fact t: "The triangle is below the square." Please note that we only train using *Spatial Logic* when the dataset contains an annotation available to create Q-$Chain$ (e.g., SpaRTUN).

Obtaining consistency constraints. We then generate the consistency constraints between q_is given the Q-$Chain$. To explain the consistency constraints between questions, we denote the answer to the YN questions as $AnsYN(q_i)$, which will be True if the answer to q_i is True. We denote the answer to the FR questions as $AnsFR(q_i, relation)$, which will be True if the specified relation exists in the set of answers to q_i. We obtain a set of consistency rules per training example, as shown in Table 1. For example, in Figure 2, if q_1: "box above the square," is True, then q_3: "triangle below box," should be True. The corresponding constraints for YN will be $AnsYN(q_1) \Rightarrow AnsYN(q_3)$, and for the FR case, will be $AnsFR(q_1, above) \Rightarrow AnsFR(q_3, below)$.

Incorporating Constraints. Our goal is to minimize the violation of logical consistency constraints. To achieve this, we require a differentiable form of logic as a surrogate for the original logical constraints. There are three commonly used types of conversions for mapping logical constraints into differentiable soft logic: t-norm Product, t-norm Łukasiewicz, and t-norm Gödel (Li et al., 2019). We use the t-norm Product for our conversion, $\neg A$ is $1 - a$, $A \wedge B$ is ab, $A \vee B$ is $a+b-ab$, and $A \Rightarrow B$ is $\min(1, \frac{b}{a})$ where A, B are concepts with probabilities a, b, respectively. We then build on previous studies on optimizing an objective that combines task performance and soft constraints (Li et al., 2019; Asai and Hajishirzi, 2020; Étienne Bamas et al., 2020). Inspired by

these works, we use the following objectives,

$$\nabla_w \mathcal{L}(w) = \nabla_w L(w) + \sum_{k=1}^{K} \lambda_k \nabla_w h_k(w) \quad (1)$$

where $L(w)$ is the task-performance loss function, i.e., Cross-Entropy loss, K is the number of logical constraints related to current $Q\text{-}Chain$, h_k measures the violation from a given logical constraint in soft logic differentiable form, and λ_k is a learning parameter indicating the importance of obeying k^{th} rule. We set the λ_k to 1 for all constraints to ensure they have equal significance.

In fact, in this work, we utilize the implementation in DominKnowS (Faghihi et al., 2021) to integrate the constraints. The DomiKnowS framework provides a declarative language to incorporate symbolic knowledge as logical constraints. We provide the original logical forms, which are automatically converted into a differentiable form for inclusion in the loss function. Specifically, we employ an optimization based on a dual formulation of the original loss, as proposed in (Étienne Bamas et al., 2020), known as the Primal-Dual (PD) program. When implementing the problem with DomiKnowS, we had to declare a graph of concepts and relationships, and define logical rules between the concepts for the constraints using Python syntax. More details on the syntax can be found in the Appendix.

Rules	Constraints in YN
R_1	$AnsYN(q_1) \Rightarrow AnsYN(q_3)$
R_2	$AnsYN(q_2) \Rightarrow AnsYN(q_4)$
R_3	$AnsYN(q_3) \wedge AnsYN(q_4) \Rightarrow AnsYN(t)$
Rules	Constraints in FR
R_1	$AnsFR(q_1, above) \Rightarrow AnsFR(q_3, below)$
R_2	$AnsFR(q_2, coveredby) \Rightarrow AnsFR(q_4, contain)$
R_3	$AnsFR(q_3, below) \wedge AnsFR(q_4, contain) \Rightarrow AnsFR(t, below)$

Table 1: Example of logical constraints presented in $Q\text{-}chain$ of Figure 2, where R_i and q_i defined in the Figure 2. $AnsYN(q_i)$ and $AnsFR(q_i, relation)$ are defined in the Section 3.3.

3.4 In-context Learning with Spatial Logic

We utilize spatial logical reasoning to create few-shot examples in the in-context prompting approaches, including Chain-of-Thoughts (Wei et al., 2023), Chain-of-Symbols (Hu et al., 2023), and others. These experiments intend to analyze and compare the LLMs' performance against our proposed fine-tuning method.

Chain-of-Thought (CoT). We modify the instructions to require an explanation before answering to enable CoT. In this setting, we augment the answer response with reasoning explanations based on spatial logical rules. Then, we provide these CoT examples, modified instruction, and target questions to generate the target answer with an explanation. An example of a reasoning explanation of CoT is given in Table 2.

Logical Representation (LR). We transform the CoT explanation into a first-order logical form in this setting. The predicate-argument structure $R(obj_i, obj_j)$ represents the relation R between obj_i and obj_j, where obj_i and obj_j denote object i and object j, respectively. We follow the same pipeline as CoT but use logical explanations instead of natural language. An example LR counterpart of the CoT phrase is in Table 2.

Chain of Symbol (CoS). This approach was introduced by Hu et al. (2023). They present the advantage of symbolic representation over natural language regarding the number of tokens and performance effectiveness. We construct the symbolic explanations of our CoT with their proposed format. We explicitly follow the proposed process in Hu et al. (2023) for converting spatial expressions from natural language into symbolic representation. An example of the CoS counterpart of CoT expression is shown in Table 2.

Domain	Spatial context
CoT	large red square is to the left of a small green square
LR	Left(large red square, small green square)
CoS	(large, red, square) < (small, green, square)

Table 2: An example of a spatial relation used in Chain of Thought (CoT), Logical Representation (LR), and Chain of Symbol (CoS).

Step-by-Step Prompting. From the SpartQA-Human dataset, we observed that scene descriptions are often complex and lengthy. To simplify this, we introduced an additional prompting step using LLMs to break down the context before answering the question. Each generated line of context contains a simple spatial description. For example, given the original context, "There exists a big red square and a medium red square in a block called A. The big one is touching the right edge of the block." the expected context generated by the LLM would be "The big red square in block A. <newline> The big red square is touching the right edge of block A..." More examples are provided in Appendixx C. We refer to this prompting technique as *Step-by-Step*, which can be applied before any ICL method.

4 Experiment

4.1 Datasets

Our experiments are conducted on the following datasets: SpaRTUN, SpartQA-Human, ReSQ, and StepGame. We only evaluate models on three out of four datasets: SpartQA-Human, ReSQ, and StepGame. This decision is based on the previous results (Mirzaee and Kordjamshidi, 2022) that demonstrate the challenge of these datasets compared to testing on SpaRTUN, which will likely be solved with typical fine-tuning on SpaRTUN.

SpaRTUN (Mirzaee and Kordjamshidi, 2022) is a synthetic SQA dataset. This dataset contains both YN and FR types of questions that require multi-hop reasoning to answer. It covers a wide range of spatial relations.[2] The answer for YN is $A = \{Yes, No\}$. While the answer for the FR is the subset of 15 spatial relations, e.g., $left, right$, and $behind$. The dataset provides the chain of reasoning annotations we use to create the Q-$Chains$ for our proposed training method.

SpartQA-Human (Ver.1) (Mirzaee et al., 2021) is a small human-annotated SQA dataset. The dataset contains both YN and FR types of questions. We only use the YN question types for fine-tuning and evaluation. The YN questions require more reasoning steps and are shown to be more challenging based on previous research results in Mirzaee and Kordjamshidi (2022).

SpartQA-Human (Ver.2), we follow the methodologies from (Mirzaee et al., 2021) to further extend the human annotation dataset with the new contexts and questions. Creation aims to enrich the SpartQA-Human with a more extensive test for diverse patterns and questions. Annotators included authors and an undergrad student who was paid as a research assistant. The extended version consists of two types of questions, YN and FR.

ReSQ (Kordjamshidi et al., 2017; Mirzaee and Kordjamshidi, 2022) is a small realistic SQA dataset with multi-hop reasoning. The depth of reasoning is smaller than other synthetic datasets, but it usually requires commonsense knowledge to infer the answer. It also provides the depth of reasoning annotation, denoted as k, which can be 1 or 2. The questions that required complex reasoning and commonsense knowledge are annotated as *unclassified* depth. 12.30% of the test examples fall in the k=1 split, 23.93% fall in the k=2 split, and the rest are unclassified. This dataset only has YN questions. We use this dataset to evaluate the performance of our models in realistic domains.

StepGame (Shi et al., 2022) is a synthetic SQA dataset containing extensive multi-hop reasoning questions over spatial relations until the depth of ten with annotation denoted as k in our tables of experimental results. This dataset contains only FR questions. The domain of answer is $a = \{left, right, above, below, overlap, lower\text{-}left, lower\text{-}right, upper\text{-}left, upper\text{-}right\}$. The advantage of utilizing this dataset is the possibility of evaluating the performance of the models at various depths of reasoning.

4.2 Experimental Models

BERT. This is the baseline architecture described in Section 3.2.1. We use bert-based-uncased as the initial checkpoint. This model is only fine-tune with the target dataset in each experiment.

BERT-T. We fine-tune the model with SpaRTUN first. Then, we further fine-tune with the target dataset of each experiment.

BERT-T + Q-Chain. We follow the methodology explained in section 3.3 to inject logical knowledge into the BERT-T model via training with constraints using the DomiKnowS framework (Faghihi et al., 2021).

Flan-T5. We follow the methodology described in section 3.2.2. We selected the flan-t5-base in this experiment as the starting checkpoint.

Flan-T5-T. We utilize the same setting as the Flan-T5 model. However, we further fine-tuned the model with SpaRTUN before finally fine-tuning it with the target dataset.

Flan-T5-T + Q-Chain. We inject the logical knowledge into Flan-T5-T model using proposed method in section 3.3. The backbone is the same as the baseline model without any changes.

Experimental Setup. We used the same hyper-parameters across all experiments. The learning rate was selected from $\{1e^{-5}, 8e^{-6}, 1e^{-6}\}$. For SpartQA-Human and ReSQ, the number of epochs was 100, while for StepGame, it was 30. For fine-tuning with SpaRTUN, we used 12 epochs for BERT and 8 epochs for Flan-T5. The final hyper-parameters were chosen using the development set of the target dataset. In all experiments, we used Cross-Entropy Loss as the loss function and Adam as the optimizer, with the *amsgrad* parameter set to *True*. The experiments were run on 8 A6000

[2] The dataset also includes Don't Know questions, but for simplicity, we apply the closed-world assumption and label those questions as No.

	SpartQA-Human		ReSQ				
Model	Ver.1	Ver.2	k=1	k=2	unclassified	All	#
BERT	54.54	53.57	70.67	56.85	60.66	60.98	1
BERT-T	55.94	58.03	**76.00**	54.79	**61.18**	61.15	2
BERT-T+Q-$Chain$ (Our)	**59.44**	**58.92**	72.00	**58.90**	59.90	**61.31**	3
Flan-T5	54.54	60.71	74.67	56.16	61.44	61.80	4
Flan-T5-T	49.65	57.14	81.33	54.79	61.44	62.30	5
Flan-T5-T+Q-$Chain$ (Our)	**55.94**	**61.61**	**81.33**	**57.53**	**63.75**	**64.43**	6
GPT3.5 (zero-shot)	58.04	58.03	74.67	60.95	66.58	66.22	7
GPT3.5 (few-shot)	62.23	58.92	84.00	68.49	68.12	70.16	8
GPT3.5 (LR)	64.33	59.82	-	-	-	-	9
GPT3.5 (CoS)	60.14	64.28	-	-	-	-	10
GPT3.5 (CoT)	65.73	71.43	**86.67**	67.12	68.64	70.49	11
GPT-4 (zero-shot)	**77.62**	68.75	84.00	73.97	**76.86**	**77.05**	12
Llama-3 (zero-shot)	61.54	50.89	80.00	64.38	67.35	68.20	13
Llama-3 (few-shot)	62.94	60.71	82.67	69.86	71.46	72.46	14
Llama-3 (LR)	60.14	**72.32**	-	-	-	-	15
Llama-3 (CoS)	67.13	61.60	-	-	-	-	16
Llama-3 (CoT)	67.83	70.54	82.76	**76.03**	67.10	71.15	17
PistaQ*	75.52	-	-	-	-	47.21	18

Table 3: Accuracy of SpartQA-Human and ReSQ with various models. For ReSQ, k is the number of the reasoning steps required to answer the questions. *Unclassified* indicates the cases in which k was challenging for human annotators to determine. The results from PistaQ* reported from Mirzaee and Kordjamshidi (2023).

GPUs, requiring approximately 100 GPU hours.

4.2.1 In-context Learning with LLMs.

We evaluate the performance of three selected LLMs: GPT3.5, GPT-4, and Llama3-8B, using ReSQ and SpartQA-Human to compare with our fine-tuning approach. The selected examples and prompts for each in-context learning method can be found in the Appendix C.

Zero-shot. We directly ask LLMs to answer the question given the corresponding context.

Few-shot. We provide LLMs with four randomly selected examples from the training data.

CoT. We augment the *few-shot* setting with reasoning explanations along with the *CoT* examples.

LR. We provide the few-shot examples with the described logical format in Section 3.4.

CoS. We further alter the *few-shot* examples with the CoS symbolic forms explained in Section 3.4.

4.3 Results

Realistic Domain. ReSQ serves as a realistic SQA domain. Table 3 shows that using the Q-$chain$ is effective for both BERT and Flan-T5, with notable improvement on Flan-T5. To better understand these results, we analyzed the performance across different splits of ReSQ, which are based on the manually annotated depth of reasoning required to answer the questions. The first two splits include questions requiring one or two hops of reasoning, denoted as $k = 1$ and $k = 2$, while the *unclassified* split covers questions where reasoning depth is hard to determine, relying more on commonsense knowledge. Our observations from Table 3 reveal that our model consistently improves performance on $k = 2$ but negatively impacts BERT on $k = 1$ and the *unclassified* categories. This suggests that logic-based tuning is more effective when deeper reasoning is needed but is less useful for questions requiring commonsense knowledge. LMs, on the other hand, outperform fine-tuning approaches on ReSQ, consistently showing 2-13% higher performance than Flan-T5+T+Q-$chain$ (lines 7-13). The improvement is most pronounced in the *unclassified* subset, even in *zero-shot* scenarios. This indicates that LLMs' superior performance is primarily due to their commonsense knowledge rather than complex reasoning, unlike our method, which targets multi-hop reasoning. However, logic-based fine-tuning improves more over Flan-T5 than BERT on the *unclassified* subset, suggesting that the Q-$chain$ can support complex reasoning when applied to a model with stronger commonsense capabilities. Note that LR and CoS have not been evaluated on ReSQ due to the challenges associated with extracting logical representations from realistic domains. This is evident in the results reported for PistaQ*, which follows a similar approach. PistaQ* achieved only 47.21% accuracy on ReSQ, despite significant improvements observed in synthetic domains such as SpartQA-Human.

Synthetic Domain with More Complex Logical Reasoning. SpartQA-Human and StepGame are synthetic domains used in our experiments. We consistently observe improvement with our proposed Q-$chain$ in this domain, which typically requires many more hops of reasoning. As observed in Table 3, Q-$chain$ consistently shows improvement in Flan-T and BERT compared to fine-tuning without it. Moreover, the gap between small PLMs and LLMs is much smaller in this dataset com-

Model	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
BERT	98.51	95.53	91.68	66.71	49.11	41.47	41.47	32.09	28.94	28.16
BERT-T	98.50	95.32	**93.26**	**76.78**	**66.36**	58.76	53.70	46.27	42.71	40.12
BERT-T+Q-$Chain$ (Our)	**98.70**	**96.45**	93.03	74.58	64.95	**59.04**	**54.38**	**49.23**	**45.36**	**44.05**
Llama-3 (few-shot)	38.01	27.87	24.15	21.27	19.75	18.03	16.88	15.52	15.17	14.70
GPT3 (few-shot)	55.00	37.00	25.00	30.00	32.00	29.00	21.00	22.00	34.00	31.00
GPT3 (CoT)	61.00	45.00	30.00	35.00	35.00	27.00	22.00	24.00	23.00	25.00
GPT3(d2) + ASP	92.60	89.90	89.10	93.80	92.90	91.60	91.20	90.40	89.00	88.30

Table 4: Accuracy of StepGame on several models including results of GPT3 reported from Yang et al. 2023.

Model	Raw Context	Step by step
GPT3.5 (zero-shot)	58.04	63.64
GPT3.5 (few-shot)	62.23	64.33
GPT3.5 (CoT)	65.73	67.83
GPT3.5 (LR)	64.33	59.44
GPT3.5 (CoS)	60.14	58.74
GPT-4 (zero-shot)	**77.62**	**78.32**

Table 5: The accuracy of LLMs on SpartQA-Human.

pared to the realistic domain (ReSQ). This is expected, as LLMs are better at commonsense than complex reasoning, as previously explained. The result is further supported when assessing the proposed method on StepGame. As can be observed in Table 4, the fine-tuning method consistently demonstrates significant positive differences in all reasoning steps compared to LLMs. The struggle of GPT3 on reasoning over StepGame is also investigated in (Yang et al., 2023). The reported results from this paper are in Table 4. Our proposed method consistently improves by 1-4% on a higher number of reasoning hops (k=6 to k=10), similar to the observation in ReSQ. These results confirm our primary hypothesis that our proposed method equips the models with a higher level of logical abstraction to conduct higher reasoning steps. We also included results from GPT-3(d2) + ASP and PistaQ*, both of which demonstrate significant effectiveness in multi-hop reasoning. These models perform relation extraction for a formal logical representation using the language model, while reasoning is handled separately using external tools. However, these pipelines face two key limitations. First, they require formal representations at test time, effective only on synthetic data. Second, they do not fine-tune the language models for better reasoning, relying instead on external tools. This approach contrasts with ours, which focuses on directly improving the language models' reasoning capabilities.

In-context Learning. We experiment with variations of in-context learning for comparison. As shown in Table 3 and Table 5, CoT proved to be the most effective approach for spatial reasoning in both GPT and Llama. We also tested the impact of using Formal Representations. Table 5 (line 4) shows that it slightly worsened performance, with natural language CoT being more effective in most cases. Additionally, we evaluate CoS symbolic representations, finding that CoT and LR outperformed CoS using the same examples, highlighting the advantage of natural language over symbolic representation in our specific setting. We further applied step-by-step prompting explained in Section 3.4 to evaluate more sophisticated prompting techniques on LLMs. Table 5 shows that this technique achieves 2-5% better results than zero-shot, few-shot, and CoT using raw context. The results demonstrate that more straightforward and shorter contexts improve LLMs' spatial understanding. Lastly, we compare open-source Llama and closed-source GPT models. Our results indicate that Llama-3 is competitive with GPT-3.5, showing only minor differences in the YN domain as shown in Table 3. Thus, we plan to prioritize open-source models in future research.

5 Conclusion

Given the importance of spatial reasoning in many real-world applications, we focus on improving this reasoning skill in language models. We equip LLMs with neuro-symbolic supervision to leverage logical knowledge during fine-tuning. This practice boosts their capability of capturing the necessary level of abstraction for spatial reasoning and enhances the generalizability for transferring knowledge across domains. Our constraint-based training technique achieves higher accuracy than other competitive Spatial Question-answering baselines across various benchmarks. Furthermore, the results indicate that our model performs better in scenarios requiring more reasoning steps. Lastly, we compare our models with state-of-the-art LLMs. Experimental comparisons show that while large LLMs like GPT3.5 perform better in commonsense reasoning, we achieve higher performance in multi-hop spatial question-answering with comparatively smaller language models like BERT.

6 Limitations

While we improve the reasoning capability of the models, our approach by no means solves the general reasoning problem. The trustworthiness and reliability of the LLMs are still a research challenge. Our models will need GPU resources to run, which might be expensive. Our research is based on open-source data and code, and the results will be reproducible. We do not see any ethical concerns in our research approach or the impact of our results. Our data, being limited to the spatial reasoning domain, does not include any specific type of bias that can harm minority people belonging to any particular gender or race.

References

AI@Meta. 2024. Llama 3 model card.

Saeed Amizadeh, Hamid Palangi, Oleksandr Polozov, Yichen Huang, and Kazuhito Koishida. 2020. Neuro-symbolic visual reasoning: Disentangling "visual" from "reasoning". *Preprint*, arXiv:2006.11524.

Akari Asai and Hannaneh Hajishirzi. 2020. Logic-guided data augmentation and regularization for consistent question answering. *Preprint*, arXiv:2004.10157.

Jamal Atif, Céline Hudelot, Geoffroy Fouquier, Isabelle Bloch, and Elsa D Angelini. 2007. From generic knowledge to specific reasoning for medical image interpretation using graph based representations. In *IJCAI*, pages 224–229.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *Preprint*, arXiv:2302.04023.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Danny Driess, Pete Florence, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. 2024. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. *Preprint*, arXiv:2401.12168.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *Preprint*, arXiv:2210.11416.

Surabhi Datta, Yuqi Si, Laritza Rodriguez, Sonya E Shooshan, Dina Demner-Fushman, and Kirk Roberts. 2020. Understanding spatial language in radiology: Representation framework, annotation, and spatial relation extraction from chest x-ray reports using deep learning. *Journal of Biomedical Informatics*, 108:103473.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Hossein Rajaby Faghihi, Quan Guo, Andrzej Uszok, Aliakbar Nafar, Elaheh Raisi, and Parisa Kordjamshidi. 2021. Domiknows: A library for integration of symbolic domain knowledge in deep learning. *Preprint*, arXiv:2108.12370.

Hossein Rajaby Faghihi, Aliakbar Nafar, Chen Zheng, Roshanak Mirzaee, Yue Zhang, Andrzej Uszok, Alexander Wan, Tanawan Premsri, Dan Roth, and Parisa Kordjamshidi. 2023. Gluecons: A generic benchmark for learning under constraints. *Preprint*, arXiv:2302.10914.

Chun Feng, Joy Hsu, Weiyu Liu, and Jiajun Wu. 2024. Naturally supervised 3d visual grounding with language-regularized concept learners. *Preprint*, arXiv:2404.19696.

Gaël Gendron, Qiming Bao, Michael Witbrock, and Gillian Dobbie. 2023. Large language models are not abstract reasoners. *Preprint*, arXiv:2305.19555.

Shizhan Gong, Yuan Zhong, Wenao Ma, Jinpeng Li, Zhao Wang, Jingyang Zhang, Pheng-Ann Heng, and Qi Dou. 2023. 3dsam-adapter: Holistic adaptation of sam from 2d to 3d for promptable medical image segmentation. *Preprint*, arXiv:2306.13465.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Hanxu Hu, Hongyuan Lu, Huajian Zhang, Yun-Ze Song, Wai Lam, and Yue Zhang. 2023. Chain-of-symbol prompting elicits planning in large langauge models. *Preprint*, arXiv:2305.10276.

Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. 2020. Harnessing deep neural networks with logic rules. *Preprint*, arXiv:1603.06318.

Parisa Kordjamshidi, Taher Rahgooy, Marie-Francine Moens, James Pustejovsky, Umar Manzoor, and Kirk Roberts. 2017. Clef 2017: Multimodal spatial role labeling (msprl) task overview. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 367–376, Cham. Springer International Publishing.

Fangjun Li, David C. Hogg, and Anthony G. Cohn. 2024. Advancing spatial reasoning in large language models: An in-depth evaluation and enhancement using the stepgame benchmark. *Preprint*, arXiv:2401.03991.

Tao Li, Vivek Gupta, Maitrey Mehta, and Vivek Srikumar. 2019. A logic-driven framework for consistency of neural models. *Preprint*, arXiv:1909.00126.

Fangyu Liu, Guy Emerson, and Nigel Collier. 2023. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics*, 11:635–651.

Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, Noah A. Smith, and Yejin Choi. 2021. Neurologic a*esque decoding: Constrained text generation with lookahead heuristics. *Preprint*, arXiv:2112.08726.

Roshanak Mirzaee, Hossein Rajaby Faghihi, Qiang Ning, and Parisa Kordjmashidi. 2021. Spartqa: : A textual question answering benchmark for spatial reasoning. *Preprint*, arXiv:2104.05832.

Roshanak Mirzaee and Parisa Kordjamshidi. 2022. Transfer learning with synthetic corpora for spatial role labeling and reasoning. *Preprint*, arXiv:2210.16952.

Roshanak Mirzaee and Parisa Kordjamshidi. 2023. Disentangling extraction and reasoning in multi-hop spatial reasoning. *Preprint*, arXiv:2310.16731.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

John Prager, Jennifer Chu-Carroll, and Krzysztof Czuba. 2004. Question answering using constraint satisfaction: QA-by-dossier-with-contraints. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 574–581, Barcelona, Spain.

Lianhui Qin, Sean Welleck, Daniel Khashabi, and Yejin Choi. 2022. Cold decoding: Energy-based constrained text generation with langevin dynamics. *ArXiv*, abs/2202.11705.

Manasi Sharma. 2023. Exploring and improving the spatial reasoning abilities of large language models. *Preprint*, arXiv:2312.01054.

Zhengxiang Shi, Qiang Zhang, and Aldo Lipani. 2022. Stepgame: A new benchmark for robust multi-hop spatial reasoning in texts. *Preprint*, arXiv:2204.08292.

Emrah Akin Sisbot, Luis F Marin, and Rachid Alami. 2007. Spatial reasoning for human robot interaction. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2281–2287. IEEE.

Haitian Sun, William W. Cohen, and Ruslan Salakhutdinov. 2022. Reasoning over logically interacted conditions for question answering. *Preprint*, arXiv:2205.12898.

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. 2023. Towards benchmarking and improving the temporal reasoning capability of large language models. *Preprint*, arXiv:2306.08952.

Karthik Valmeekam, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. 2023. Large language models still can't plan (a benchmark for llms on planning and reasoning about change). *Preprint*, arXiv:2206.10498.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M. Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *Preprint*, arXiv:1502.05698.

Elmira Yadollahi, Miguel Alexandre Monteiro, and Ana Paiva. 2023. Learning spatial reasoning in virtual vs. physical games with robots. In *Proceedings of the 11th International Conference on Human-Agent Interaction*, HAI '23, page 162170, New York, NY, USA. Association for Computing Machinery.

Zhun Yang, Adam Ishay, and Joohyung Lee. 2023. Coupling large language models with logic programming for robust and general reasoning from text. *Preprint*, arXiv:2307.07696.

Yue Zhang, Quan Guo, and Parisa Kordjamshidi. 2021. Towards navigation by reasoning over spatial configurations. *Preprint*, arXiv:2105.06839.

Yue Zhang and Parisa Kordjamshidi. 2022a. Lovis: Learning orientation and visual signals for vision and language navigation. *Preprint*, arXiv:2209.12723.

Yue Zhang and Parisa Kordjamshidi. 2022b. LOViS: Learning orientation and visual signals for vision and language navigation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5745–5754, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Ming Zhou, Nan Duan, Shujie Liu, and Heung-Yeung Shum. 2020. Progress in neural nlp: Modeling, learning, and reasoning. *Engineering*, 6(3):275–290.

Étienne Bamas, Andreas Maggiori, and Ola Svensson. 2020. The primal-dual method for learning augmented algorithms. *Preprint*, arXiv:2010.11632.

A Data Statistic

The data statistic of each dataset is shown in Table 6.

Dataset	Train	Test	Dev
SpaRTUN (YN)	20633	3232	3152
SpaRTUN (FR)	18400	2818	2830
Q-Chain (YN)	56987	-	-
Q-Chain (FR)	46750	-	-
SpartQA-Human(Ver.1)	161	143	51
SpartQA-Human(Ver.2)	200	112	60
ReSQ	1008	610	333
StepGame	50000	5000	100000

Table 6: Size of each SQA benchmark used in experiments

B RoBERTa results

We provide results from the same setting as the BERT using RoBERTa in Table 7 and found that the overall results are slightly lower than BERT for typical transfer learning. So, we decided to use only BERT for the remaining experiments.

C In-context Learning Examples

The example of each in-context learning prompt and example is indicated below.

C.1 Few-Shot

system : You will be given a story and question. Then, you will answer either only Yes or No based on given story. Candidate answer: [Yes, No]

user : There exist a big red square, a big red triangle, a medium red square, and a small green circle in a block called A. The triangle and medium square are touching the bottom edge of the block. The big and medium square are touching the right edge of the block. And the circle is above the big square which is above the medium square. There is another block called B to the left of block A. A medium green square is touching block B's left edge and below a medium red square. The medium red square is above and to the left of a small green square. Also a medium red triangle is below and to the left of the small square. Is the medium red triangle below the red square in block B?

assistant : Yes

user : There are three blocks A, B and C. Block A has a medium white triangle and a small red rectangle. There exists a medium white rectangle to the left of the triangle and to the right of the red rectangle. Aove block A there is block B. It has a small white oval touching the bottom edge of the block and a medium white triangle to the right of it. Below block A there is block C. Its has a small green triangle touching the top edge of the block. A medium white triangle is below a small green rectangle. Another small green triangle is below and to the right of the white triangle. Are all rectangles above a white triangle?'

assistant : Yes

user : There is a block called A. Inside A, a black triangle above and to the left of a black circle. There is another block called B which is to the right of A. B has two yellow circles, a blue circle, and a blue square. The blue circle is touching the left edge of B. One of the yellow circles is to the right of the blue circle and to the left of the blue square. The blue square is touching the right edge of B and above the other yellow circle, Which also touches the right edge of B. Does block B have all of the circles inside it?

assistant : No

user : There is a block called A. A has a large blue square near and below a medium blue triangle. Far to the left of the medium blue triangle is a small yellow square. Far above the medium blue triangle is a small blue triangle. There is another block called B, which is to the right of A. In B, There is a small yellow circle touching the left edge of B. Far to the right of the circle is a large yellow square. Near and above the large square is a small yellow square. There is another block

	SpartQA-Human		ReSQ				
Model	Ver.1	Ver.2	k=1	k=2	unclassified	All	#
BERT	54.54	53.57	70.67	56.85	60.66	60.98	1
RoBERTa	54.54	-	76.00	56.16	58.87	60.33	2
BERT-T	55.94	58.03	76.00	54.79	61.18	61.15	2
RoBERTa-T	49.65	-	64.67	57.87	55.78	56.72	4
BERT-T+Q-$Chain$ (Our)	59.44	58.92	72.00	58.90	59.90	61.31	3

Table 7: Accuracy of SpartQA-Human and ReSQ with BERT and RoBERTa for comparison between these models.

called C, which is to the right of B. In C, there is a small blue circle near and to the left of a large yellow triangle. The large yellow triangle is near and to the left of a large blue triangle. The large blue triangle is far above a large blue circle. The large blue circle is touching the bottom edge of C. Are the large blue things in C to the left of the large blue thing in A?

assistant : No

user : There is a block called A. In A there is a red triangle Which is above a black circle. The black circle is touching the bottom of A. A yellow square is touching the top edge of A, and below that, a red oval is touching the right edge of A. There is a second block called B, to the left of A. Inside B are 3 squares: a brown, a green, and a red. The green square is above the red square and touching the left edge of B. The brown square is touching both the bottom edge and the right edge of B. The red square is touching the left edge of the brown square. Is the oval to the right of the brown thing?

C.2 Chain of Thought

system : You will be given story and question. Then, you will answer either Yes or No with explanation based on given story step by step. Candidate answer: [Yes, No]

user : There exist a big red square, a big red triangle, a medium red square, and a small green circle in a block called A. The triangle and medium square are touching the bottom edge of the block. The big and medium square are touching the right edge of the block. And the circle is above the big square which is above the medium square. There is another block called B to the left of block A. A medium green square is touching block B's left edge and below a medium red square. The medium red square is above and to the left of a small green square. Also a medium red triangle is below and to the left of the small square. Is the medium red triangle below the red square in block B?

assistant : From information provided:The medium red square is above a small green square in B.Therefore, the small green square is below the medium red square in B. The medium red triangle is below the small square in B, and the small green square is below the medium red square in B.Therefore, the medium red triangle is below the medium red square in B. Answer: Yes

user : There are three blocks A, B and C. Block A has a medium white triangle and a small red rectangle. There exists a medium white rectangle to the left of the triangle and to the right of the red rectangle. Aove block A there is block B. It has a small white oval touching the bottom edge of the block and a medium white triangle to the right of it. Below block A there is block C. Its has a small green triangle touching the top edge of the block. A medium white triangle is below a small green rectangle. Another small green triangle is below and to the right of the white triangle. Are all rectangles above a white triangle?'

assistant : From given information: A medium white triangle is below a small green rectangle in C.Then, small green rectangle is above medium white triangle in C.A small red rectangle in block A, a medium white triangle in C, and block C is below block A.Then, a medium white triangle is below a A small red rectangle. A medium white triangle is below a A small red rectangle. Therefore, a small red rectangle is above a medium white triangle.A medium white rectangle is in A,a medium white triangle in C, and block C is below block A.Then, a medium white triangle is below a A medium white rectangle. A medium white triangle is below a A medium white rectangle. Therefore, a medium white rectangle is above a medium white triangle.A medium white rectangle is above a medium white triangle.Thus, all rectangles from given story are above a white triangle. Answer: Yes'

user : There is a block called A. Inside A, a black triangle above and to the left of a black circle. There is another block called B which is to the

right of A. B has two yellow circles, a blue circle, and a blue square. The blue circle is touching the left edge of B. One of the yellow circles is to the right of the blue circle and to the left of the blue square. The blue square is touching the right edge of B and is above the other yellow circle, Which is also touching the right edge of B. Does block B have all of the circles inside it?

assistant : From given relation, A black triangle is above a black circle in A.Then, there is black circle in A.Therefore, not all circle are in B. Answer: No

user : There is a block called A. A has a large blue square near and below a medium blue triangle. Far to the left of the medium blue triangle is a small yellow square. Far above the medium blue triangle is a small blue triangle. There is another block called B, which is to the right of A. In B, There is a small yellow circle touching the left edge of B. Far to the right of the circle is a large yellow square. Near and above the large square is a small yellow square. There is another block called C, which is to the right of B. In C, there is a small blue circle near and to the left of a large yellow triangle. The large yellow triangle is near and to the left of a large blue triangle. The large blue triangle is far above a large blue circle. The large blue circle is touching the bottom edge of C. Are the large blue things in C to the left of the large blue thing in A?

assistant : Based on the provided content. Block B is to the right if block A.Block C is to the right of block B.Therefore, block C is to the right of block A.In block C, the yellow triangle is near and to the left of large blue triangle. Therefore, there is large blue triangle in C.Block A has a large blue square near and below a medium blue triangle.Therefore, there is large blue square in A.Then, large blue triangle in C is to the right of large blue square in A.So, large blue triangle in C is not to the left of large blue square in A. Answer: No

user : There is a block called A. In A there is a red triangle Which is above a black circle. The black circle is touching the bottom of A. A yellow square is touching the top edge of A, and below that, a red oval is touching the right edge of A. There is a second block call B Which is to the left of A. Inside B there are 3 squares, a brown one, a green one, and a red one. The green square is above the red square and touching the left edge of B. The brown square is touching both the bottom edge and the right edge of B. The red square is touching the left edge of the brown square. Is the oval to the right of the brown thing?

C.3 Logical Representation (LR)

system : You will be given story and question. Then, you will answer either Yes or No with explanation based on given story step by step. Candidate answer: [Yes, No]

user : There exist a big red square, a big red triangle, a medium red square, and a small green circle in a block called A. The triangle and medium square are touching the bottom edge of the block. The big and medium square are touching the right edge of the block. And the circle is above the big square which is above the medium square. There is another block called B to the left of block A. A medium green square is touching the left edge of block B and is below a medium red square. The medium red square is above and to the left of a small green square. Also a medium red triangle is below and to the left of the small square. Is the medium red triangle below the red square in block B?

assistant : Below(Medium red triangle in B, red square in block B) Answer: Yes

user : There are three blocks A, B and C. Block A has a medium white triangle and a small red rectangle. There exists a medium white rectangle to the left of the triangle and to the right of the red rectangle. Above block A there is block B. It has a small white oval touching the bottom edge of the block and a medium white triangle to the right of it. Below block A there is block C. Its has a small green triangle touching the top edge of the block. A medium white triangle is below a small green rectangle. Another small green triangle is below and to the right of the white triangle. Are all rectangles above a white triangle?'

assistant : Below(C, A) => Above(A, C) Contain(A, Small red rectangle) + Above(A, C) + Contain(C, Medium white triangle)=>Above(Small red rectangle, Medium White triangle) Above(A, C) & Contain(A, Medium white rectangle) & Contain(C, Medium white triangle)=>Above(Medium white rectangle, Medium White triangle) Below(Medium white triangle in C, small green rectangle in C) =>Above(small green rectangle, Medium white triangle)Answer: Yes

user : There is a block called A. Inside A, there is a black triangle that is above and to the left of a black circle. There is another block called B which

is to the right of A. In B, there are two yellow circles, a blue circle, and a blue square. The blue circle is touching the left edge of B. One of the yellow circles is to the right of the blue circle and to the left of the blue square. The blue square is touching the right edge of B and is above the other yellow circle, Which is also touching the right edge of B. Does block B have all of the circles inside it?

assistant : Contain(A, Black Circle) => ¬Contain(B, Black Circle) ¬Contain(B, Black Circle) => Some x, ¬Contain(B, x) where x is Circle. Answer: No

user : There is a block called A. In A, there is a large blue square near and below a medium blue triangle. Far to the left of the medium blue triangle is a small yellow square. Far above the medium blue triangle is a small blue triangle. There is another block called B, which is to the right of A. In B, There is a small yellow circle touching the left edge of B. Far to the right of the circle is a large yellow square. Near and above the large square is a small yellow square. There is another block called C, which is to the right of B. In C, there is a small blue circle near and to the left of a large yellow triangle. The large yellow triangle is near and to the left of a large blue triangle. The large blue triangle is far above a large blue circle. The large blue circle is touching the bottom edge of C. Are the large blue things in C to the left of the large blue thing in A?

assistant : Question: Left(large blue thing in C, large blue thing in A)Related relation from story: Right(B, A), Right(C, B), Contain(C, large blue triangle), Contain(A, large blue square)Steps: Right(C, B) + Right(B, A) => Right(C, A) Contain(C, large blue triangle) + Right(C, A) + Contain(A, large blue square) => Right(large blue triangle in C, large blue square in A)Right(large blue triangle in C, large blue square in A) => !Left(large blue thing in C, large blue thing in A) Answer: No

user : There is a block called A. In A there is a red triangle Which is above a black circle. The black circle is touching the bottom of A. A yellow square is touching the top edge of A and below that there is a red oval is touching the right edge of A. There is a second block call B Which is to the left of A. Inside B there are 3 squares, a brown one, a green one, and a red one. The green square is above the red square and touching the left edge of B. The brown square is touching both the bottom edge and the right edge of B. The red square is touching the left edge of the brown square. Is the oval to the right of the brown thing?

C.4 Chain of Symbol

system : You will be given story and question. Then, you will answer either Yes or No with explanation based on given story step by step. Candidate answer: [Yes, No]

user : There exist a big red square, a big red triangle, a medium red square, and a small green circle in a block called A. The triangle and medium square are touching the bottom edge of the block. The big and medium square are touching the right edge of the block. And the circle is above the big square which is above the medium square. There is another block called B to the left of block A. A medium green square is touching the left edge of block B and is below a medium red square. The medium red square is above and to the left of a small green square. Also a medium red triangle is below and to the left of the small square. Is the medium red triangle below the red square in block B?

assistant : Let analyze the relations given from context to answer question.B: 1. (medium, red, triangle) ↓, < (small, red, square)Based on the information, B: 1. (medium, red, triangle) ↓, < (small, red, square)2. (medium, red, triangle) ↓ (small, red, square)Answer: Yes

user : There are three blocks A, B and C. Block A has a medium white triangle and a small red rectangle. There exists a medium white rectangle to the left of the triangle and to the right of the red rectangle. Above block A there is block B. It has a small white oval touching the bottom edge of the block and a medium white triangle to the right of it. Below block A there is block C. Its has a small green triangle touching the top edge of the block. A medium white triangle is below a small green rectangle. Another small green triangle is below and to the right of the white triangle. Are all rectangles above a white triangle?'

assistant : Let consider the related relations to answer the questions, A: 1. (small, red, rectangle) 2. (medium, white, rectangle) C: 1. (small, green, rectangle) = edge 2. (medium, white, triangle)) ↓ (small, green, rectangle) C ↓ A Based on the information, let determined the relation of all rectangle and white triangles, 1. C ↓ A. Therefore, A ↑ C 2. A ↑ C, A: (small, red, rectangle), and C: (medium, white, triangle). Therefore, A: (small, red, rectangle) ↑ C: (medium, white, triangle) 3.

A ↑ C, A: (medium, white, rectangle), and C: (medium, white, triangle). Therefore, A: (medium, white, rectangle) ↑ C: (medium, white, triangle). 4. C: (medium, white, triangle)) ↓ (small, green, rectangle). Therefore, (small, green, rectangle) ↑ (medium, white, triangle).5. A: (small, red, rectangle) ↑ C: (medium, white, triangle), A: (medium, white, rectangle) ↑ C: (medium, white, triangle), and C: (small, green, rectangle) ↑ (medium, white, triangle), therefore, all rectangle above a white triangle. Answer: Yes

user : There is a block called A. Inside A, there is a black triangle that is above and to the left of a black circle. There is another block called B which is to the right of A. In B, there are two yellow circles, a blue circle, and a blue square. The blue circle is touching the left edge of B. One of the yellow circles is to the right of the blue circle and to the left of the blue square. The blue square is touching the right edge of B and is above the other yellow circle, Which is also touching the right edge of B. Does block B have all of the circles inside it?

assistant : Let consider the related relations to answer the questions,A: 1. (black, triangle) ↑ < (black, circle)B: 1. (yellow, circle1) right (blue, circle)2. (blue, square) ↑ (yellow, circle2) 3.(blue, circle) =left edge of B Based on the information, let determined the relation of all rectangle and white triangles, 1. A: 1. (black, triangle) ↑< (black, circle). Therefore, A: (black, circle)2. A: (black, circle). Therefore, B doesn't have all circle. Answer: No

user : There is a block called A. In A, there is a large blue square near and below a medium blue triangle. Far to the left of the medium blue triangle is a small yellow square. Far above the medium blue triangle is a small blue triangle. There is another block called B, which is to the right of A. In B, There is a small yellow circle touching the left edge of B. Far to the right of the circle is a large yellow square. Near and above the large square is a small yellow square. There is another block called C, which is to the right of B. In C, there is a small blue circle near and to the left of a large yellow triangle. The large yellow triangle is near and to the left of a large blue triangle. The large blue triangle is far above a large blue circle. The large blue circle is touching the bottom edge of C. Are the large blue things in C to the left of the large blue thing in A?

assistant : Let consider the related relations to answer the questions,B > A,C > BA: 1. (large, blue square) ∼ ↓ (medium, blue square)C: 1. (large, yellow, triangle)Based on the information, let determined the relation large blue things in C and the large blue thing in A,1. C > B and B > A. Therefore, C > A2. A: 1. (large, blue, square) ∼↓ (medium, blue square). Therefore, A: (large, blue, square)3. C > A, A: (large, blue, square), and C: (large, yellow, triangle).Therefore, C: (large, yellow, triangle) > A: (large, blue, square).4. C: (large, yellow, triangle) > A: (large, blue, square).Therefore, C: (large, yellow, triangle) !< A: (large, blue, square)large blue things in C is not to the left of the large blue thing in A. Answer: No

user : There is a block called A. In A there is a red triangle Which is above a black circle. The black circle is touching the bottom of A. A yellow square is touching the top edge of A and below that there is a red oval is touching the right edge of A. There is a second block call B Which is to the left of A. Inside B there are 3 squares, a brown one, a green one, and a red one. The green square is above the red square and touching the left edge of B. The brown square is touching both the bottom edge and the right edge of B. The red square is touching the left edge of the brown square. Is the oval to the right of the brown thing?

C.5 Step-by-Step Prompting

Raw context : There exist a big red square, a big red triangle, a medium red square, and a small green circle in a block called A. The triangle and medium square are touching the bottom edge of the block. The big and medium square are touching the right edge of the block. And the circle is above the big square which is above the medium square. There is another block called B to the left of block A. A medium green square is touching the left edge of block B and is below a medium red square. The medium red square is above and to the left of a small green square. Also a medium red triangle is below and to the left of the small square.

Spliting context : A big red square in block A.

A big red triangle in block A.

A medium red square in block A.

A small green circle in block A.

The triangle is touching the bottom edge of block A.

The medium square is touching the bottom edge of block A.

The big square is touching the right edge of block A.

The medium square is touching the right edge of block A.

The circle is above the big square in A.

The big square is above the medium square in A.

Block B is to the left of block A.

A medium green square is touching the left edge of block B.

A medium green square is below a medium red square in B.

The medium red square is above a small green square in B.

The medium red square is left of a small green square in B.

The medium red triangle is below the small square in B.

The medium red triangle is left the small square in B.

D Spatial Logical Rules

The conversion from spatial logical rules proposed in (Mirzaee and Kordjamshidi, 2022) to logical constraints used in our experiment is shown in Table 8.

E DomiKnowS Concepts and Graphs

Graph and concepts declaration of our work inside the DomiKnowS framework.

E.1 Graph for YN

```
story = Concept(name="story")
question = Concept(name="question")
story_contain, = story.contains(question)
answer_class = question(name="answer_class", ConceptClass=EnumConcept,
                        values=["yes", "no"])
symmetric = Concept(name="symmetric")
s_quest1, s_quest2 = symmetric.has_a(arg1=question, arg2=question)
ifL(andL(answer_class.yes('x'), existsL(symmetric('s', path=('x', symmetric)))),
    answer_class.yes(path=('s', s_quest2)))

reverse = Concept(name="reverse")
r_quest1, r_quest2 = reverse.has_a(arg10=question, arg20=question)
ifL(andL(answer_class.yes('x'), existsL(reverse('r', path=('x', reverse)))),
    answer_class.no(path=('r', r_quest2)))
ifL(andL(answer_class.no('x'), existsL(reverse('r', path=('x', reverse)))),
    answer_class.yes(path=('r', r_quest2)))

transitive = Concept(name="transitive")
t_quest1, t_quest2, t_quest3 = transitive.has_a(arg11=question, arg22=question, arg33=question)
ifL(andL(answer_class.yes('x'), existsL(transitive("t", path=('x', transitive))), answer_class.yes(path=('t', t_quest2))),
    answer_class.yes(path=('t', t_quest3)))

transitive_topo = Concept(name="transitive_topo")
tt_quest1, tt_quest2, tt_quest3, tt_quest4 = transitive_topo.has_a(arg111=question, arg222=question,arg333=question, arg444=question)
ifL(andL(answer_class.yes('x'), existsL(transitive("t", path=('x', transitive_topo))),
          answer_class.yes(path=('t', tt_quest2)), answer_class.yes(path=('t', tt_quest3))),
    answer_class.yes(path=('t', tt_quest4)))
```

E.2 Graph for FR on SpaRTUN

```
story = Concept(name="story")
question = Concept(name="question")
story_contain, = story.contains(question)

left = question(name="left")
right = question(name="right")
```

Rule Type	Rule	Constraints in YN	Constraints in FR
Converse	$above(x,y) \Rightarrow below(y,x)$ $below(x,y) \Rightarrow above(y,x)$ $left(x,y) \Rightarrow right(y,x)$ $right(x,y) \Rightarrow left(y,x)$ $front(x,y) \Rightarrow behind(y,x)$ $behind(x,y) \Rightarrow front(y,x)$ $coveredby(x,y) \Rightarrow cover(y,x)$ $cover(x,y) \Rightarrow coveredby(y,x)$ $inside(x,y) \Rightarrow contain(y,x)$ $contain(x,y) \Rightarrow inside(y,x)$	$AnsYn(q_i) \Rightarrow AnsYn(q_j)$	$AnsFR(q_i, above) \Rightarrow AnsFR(q_j, below)$ $AnsFR(q_i, below) \Rightarrow AnsFR(q_j, above)$ $AnsFR(q_i, left) \Rightarrow AnsFR(q_j, right)$ $AnsFR(q_j, right) \Rightarrow AnsFR(q_j, left)$ $AnsFR(q_i, front) \Rightarrow AnsFR(q_j, behind)$ $AnsFR(q_i, behind) \Rightarrow AnsFR(q_j, front)$ $AnsFR(q_i, coveredby) \Rightarrow AnsFR(q_j, cover)$ $AnsFR(q_i, cover) \Rightarrow AnsFR(q_j, coveredby)$ $AnsFR(q_i, inside) \Rightarrow AnsFR(q_j, contain)$ $AnsFR(q_i, contain) \Rightarrow AnsFR(q_j, inside)$
Symmetric	$near(x,y) \Rightarrow near(y,x)$ $far(x,y) \Rightarrow far(y,x)$ $touch(x,y) \Rightarrow touch(y,x)$ $disconnected(x,y) \Rightarrow disconnected(y,x)$ $overlap(x,y) \Rightarrow overlap(y,x)$	$AnsYn(q_i) \Rightarrow AnsYn(q_j)$	$AnsFR(q_i, near) \Rightarrow AnsFR(q_j, near)$ $AnsFR(q_i, far) \Rightarrow AnsFR(q_j, far)$ $AnsFR(q_i, touch) \Rightarrow AnsFR(q_j, touch)$ $AnsFR(q_i, disconnected) \Rightarrow AnsFR(q_j, disconnected)$ $AnsFR(q_i, overlap) \Rightarrow AnsFR(q_j, overlap)$
Transitivity	$left(x,y) \wedge left(y,z) \Rightarrow left(x,z)$ $right(x,y) \wedge right(y,z) \Rightarrow right(x,z)$ $above(x,y) \wedge above(y,z) \Rightarrow above(x,z)$ $below(x,y) \wedge below(y,z) \Rightarrow below(x,z)$ $behind(x,y) \wedge behind(y,z) \Rightarrow behind(x,z)$ $front(x,y) \wedge front(y,z) \Rightarrow front(x,z)$ $inside(x,y) \wedge inside(y,z) \Rightarrow inside(x,z)$ $contain(x,y) \wedge contain(y,z) \Rightarrow contain(x,z)$ $inside(x,y) \wedge coveredby(y,z) \Rightarrow inside(x,z)$ $contain(x,y) \wedge cover(y,z) \Rightarrow contain(x,z)$ $inside(x,y) \wedge left(y,z) \Rightarrow left(x,z)$ $inside(x,y) \wedge right(y,z) \Rightarrow right(x,z)$ $inside(x,y) \wedge above(y,z) \Rightarrow above(x,z)$ $inside(x,y) \wedge below(y,z) \Rightarrow below(x,z)$ $inside(x,y) \wedge behind(y,z) \Rightarrow behind(x,z)$ $inside(x,y) \wedge front(y,z) \Rightarrow front(x,z)$ $inside(x,y) \wedge near(y,z) \Rightarrow near(x,z)$ $inside(x,y) \wedge far(y,z) \Rightarrow far(x,z)$ $inside(x,y) \wedge disconnected(y,z) \Rightarrow disconnected(x,z)$ $coveredby(x,y) \wedge left(y,z) \Rightarrow left(x,z)$ $coveredby(x,y) \wedge right(y,z) \Rightarrow right(x,z)$ $coveredby(x,y) \wedge above(y,z) \Rightarrow above(x,z)$ $coveredby(x,y) \wedge below(y,z) \Rightarrow below(x,z)$ $coveredby(x,y) \wedge behind(y,z) \Rightarrow behind(x,z)$ $coveredby(x,y) \wedge front(y,z) \Rightarrow front(x,z)$ $coveredby(x,y) \wedge near(y,z) \Rightarrow near(x,z)$ $coveredby(x,y) \wedge far(y,z) \Rightarrow far(x,z)$ $coveredby(x,y) \wedge disconnected(y,z) \Rightarrow disconnected(x,z)$	$AnsYn(q_i) \wedge AnsYN(q_j) \Rightarrow AnsYN(q_k)$	$AnsFR(q_i, left) \wedge AnsFR(q_j, left) \Rightarrow AnsFR(q_k, left)$ $AnsFR(q_i, right) \wedge AnsFR(q_j, right) \Rightarrow AnsFR(q_k, right)$ $AnsFR(q_i, above) \wedge AnsFR(q_j, above) \Rightarrow AnsFR(q_k, above)$ $AnsFR(q_i, below) \wedge AnsFR(q_j, below) \Rightarrow AnsFR(q_k, below)$ $AnsFR(q_i, behind) \wedge AnsFR(q_j, behind) \Rightarrow AnsFR(q_k, behind)$ $AnsFR(q_i, front) \wedge AnsFR(q_j, front) \Rightarrow AnsFR(q_k, front)$ $AnsFR(q_i, inside) \wedge AnsFR(q_j, inside) \Rightarrow AnsFR(q_k, inside)$ $AnsFR(q_i, contain) \wedge AnsFR(q_j, contain) \Rightarrow AnsFR(q_k, contain)$ $AnsFR(q_i, inside) \wedge AnsFR(q_j, coveredby) \Rightarrow AnsFR(q_k, inside)$ $AnsFR(q_i, contain) \wedge AnsFR(q_j, cover) \Rightarrow AnsFR(q_k, contain)$ $AnsFR(q_i, inside) \wedge AnsFR(q_j, left) \Rightarrow AnsFR(q_k, left)$ $AnsFR(q_i, inside) \wedge AnsFR(q_j, right) \Rightarrow AnsFR(q_k, right)$ $AnsFR(q_i, inside) \wedge AnsFR(q_j, above) \Rightarrow AnsFR(q_k, above)$ $AnsFR(q_i, inside) \wedge AnsFR(q_j, below) \Rightarrow AnsFR(q_k, below)$ $AnsFR(q_i, inside) \wedge AnsFR(q_j, behind) \Rightarrow AnsFR(q_k, behind)$ $AnsFR(q_i, inside) \wedge AnsFR(q_j, front) \Rightarrow AnsFR(q_k, front)$ $AnsFR(q_i, inside) \wedge AnsFR(q_j, near) \Rightarrow AnsFR(q_k, near)$ $AnsFR(q_i, inside) \wedge AnsFR(q_j, far) \Rightarrow AnsFR(q_k, far)$ $AnsFR(q_i, inside) \wedge AnsFR(q_j, disconnected) \Rightarrow AnsFR(q_k, disconnected)$ $AnsFR(q_i, coveredby) \wedge AnsFR(q_j, left) \Rightarrow AnsFR(q_k, left)$ $AnsFR(q_i, coveredby) \wedge AnsFR(q_j, right) \Rightarrow AnsFR(q_k, right)$ $AnsFR(q_i, coveredby) \wedge AnsFR(q_j, above) \Rightarrow AnsFR(q_k, above)$ $AnsFR(q_i, coveredby) \wedge AnsFR(q_j, below) \Rightarrow AnsFR(q_k, below)$ $AnsFR(q_i, coveredby) \wedge AnsFR(q_j, behind) \Rightarrow AnsFR(q_k, behind)$ $AnsFR(q_i, coveredby) \wedge AnsFR(q_j, front) \Rightarrow AnsFR(q_k, front)$ $AnsFR(q_i, coveredby) \wedge AnsFR(q_j, near) \Rightarrow AnsFR(q_k, near)$ $AnsFR(q_i, coveredby) \wedge AnsFR(q_j, far) \Rightarrow AnsFR(q_k, far)$ $AnsFR(q_i, coveredby) \wedge AnsFR(q_j, disconnected) \Rightarrow AnsFR(q_k, disconnected)$
Transitivity + Topological	$inside(x,y) \wedge inside(h,z) \wedge left(y,z) \Rightarrow left(x,h)$ $inside(x,y) \wedge inside(h,z) \wedge right(y,z) \Rightarrow right(x,h)$ $inside(x,y) \wedge inside(h,z) \wedge above(y,z) \Rightarrow above(x,h)$ $inside(x,y) \wedge inside(h,z) \wedge below(y,z) \Rightarrow below(x,h)$ $inside(x,y) \wedge inside(h,z) \wedge behind(y,z) \Rightarrow behind(x,h)$ $inside(x,y) \wedge inside(h,z) \wedge front(y,z) \Rightarrow front(x,h)$ $inside(x,y) \wedge inside(h,z) \wedge near(y,z) \Rightarrow near(x,h)$ $inside(x,y) \wedge inside(h,z) \wedge far(y,z) \Rightarrow far(x,h)$ $inside(x,y) \wedge inside(h,z) \wedge disconnected(y,z) \Rightarrow disconnected(x,h)$ $coveredby(x,y) \wedge coveredby(h,z) \wedge left(y,z) \Rightarrow left(x,h)$ $coveredby(x,y) \wedge coveredby(h,z) \wedge right(y,z) \Rightarrow right(x,h)$ $coveredby(x,y) \wedge coveredby(h,z) \wedge above(y,z) \Rightarrow above(x,h)$ $coveredby(x,y) \wedge coveredby(h,z) \wedge below(y,z) \Rightarrow below(x,h)$ $coveredby(x,y) \wedge coveredby(h,z) \wedge behind(y,z) \Rightarrow behind(x,h)$ $coveredby(x,y) \wedge coveredby(h,z) \wedge front(y,z) \Rightarrow front(x,h)$ $coveredby(x,y) \wedge coveredby(h,z) \wedge near(y,z) \Rightarrow near(x,h)$ $coveredby(x,y) \wedge coveredby(h,z) \wedge far(y,z) \Rightarrow far(x,h)$ $coveredby(x,y) \wedge coveredby(h,z) \wedge disconnected(y,z) \Rightarrow disconnected(x,h)$ $left(x,y) \wedge left(h,z) \wedge contain(y,z) \Rightarrow left(x,h)$ $left(x,y) \wedge left(h,z) \wedge cover(y,z) \Rightarrow left(x,h)$ $right(x,y) \wedge right(h,z) \wedge contain(y,z) \Rightarrow right(x,h)$ $right(x,y) \wedge right(h,z) \wedge cover(y,z) \Rightarrow right(x,h)$ $above(x,y) \wedge above(h,z) \wedge contain(y,z) \Rightarrow above(x,h)$ $above(x,y) \wedge above(h,z) \wedge cover(y,z) \Rightarrow above(x,h)$ $below(x,y) \wedge below(h,z) \wedge contain(y,z) \Rightarrow below(x,h)$ $below(x,y) \wedge below(h,z) \wedge cover(y,z) \Rightarrow below(x,h)$ $behind(x,y) \wedge behind(h,z) \wedge contain(y,z) \Rightarrow behind(x,h)$ $behind(x,y) \wedge behind(h,z) \wedge cover(y,z) \Rightarrow behind(x,h)$ $front(x,y) \wedge front(h,z) \wedge contain(y,z) \Rightarrow front(x,h)$ $front(x,y) \wedge front(h,z) \wedge cover(y,z) \Rightarrow front(x,h)$ $near(x,y) \wedge near(h,z) \wedge contain(y,z) \Rightarrow near(x,h)$ $near(x,y) \wedge near(h,z) \wedge cover(y,z) \Rightarrow near(x,h)$ $far(x,y) \wedge far(h,z) \wedge contain(y,z) \Rightarrow far(x,h)$ $far(x,y) \wedge far(h,z) \wedge cover(y,z) \Rightarrow far(x,h)$ $disconnected(x,y) \wedge disconnected(h,z) \wedge contain(y,z) \Rightarrow disconnected(x,h)$ $disconnected(x,y) \wedge disconnected(h,z) \wedge cover(y,z) \Rightarrow disconnected(x,h)$	$AnsYn(q_i) \wedge AnsYN(q_j) \wedge AnsYN(q_k) \Rightarrow AnsYN(q_l)$	$AnsFR(q_i, inside) \wedge AnsFR(q_j, inside) \wedge AnsFR(q_k, left) \Rightarrow AnsFR(q_l, left)$ $AnsFR(q_i, inside) \wedge AnsFR(q_j, inside) \wedge AnsFR(q_k, right) \Rightarrow AnsFR(q_l, right)$ $AnsFR(q_i, inside) \wedge AnsFR(q_j, inside) \wedge AnsFR(q_k, above) \Rightarrow AnsFR(q_l, above)$ $AnsFR(q_i, inside) \wedge AnsFR(q_j, inside) \wedge AnsFR(q_k, below) \Rightarrow AnsFR(q_l, below)$ $AnsFR(q_i, inside) \wedge AnsFR(q_j, inside) \wedge AnsFR(q_k, behind) \Rightarrow AnsFR(q_l, behind)$ $AnsFR(q_i, inside) \wedge AnsFR(q_j, inside) \wedge AnsFR(q_k, front) \Rightarrow AnsFR(q_l, front)$ $AnsFR(q_i, inside) \wedge AnsFR(q_j, inside) \wedge AnsFR(q_k, near) \Rightarrow AnsFR(q_l, near)$ $AnsFR(q_i, inside) \wedge AnsFR(q_j, inside) \wedge AnsFR(q_k, far) \Rightarrow AnsFR(q_l, far)$ $AnsFR(q_i, inside) \wedge AnsFR(q_j, inside) \wedge AnsFR(q_k, disconnected) \Rightarrow AnsFR(q_l, disconnected)$ $AnsFR(q_i, coveredby) \wedge AnsFR(q_j, coveredby) \wedge AnsFR(q_k, left) \Rightarrow AnsFR(q_l, left)$ $AnsFR(q_i, coveredby) \wedge AnsFR(q_j, coveredby) \wedge AnsFR(q_k, right) \Rightarrow AnsFR(q_l, right)$ $AnsFR(q_i, coveredby) \wedge AnsFR(q_j, coveredby) \wedge AnsFR(q_k, above) \Rightarrow AnsFR(q_l, above)$ $AnsFR(q_i, coveredby) \wedge AnsFR(q_j, coveredby) \wedge AnsFR(q_k, below) \Rightarrow AnsFR(q_l, below)$ $AnsFR(q_i, coveredby) \wedge AnsFR(q_j, coveredby) \wedge AnsFR(q_k, behind) \Rightarrow AnsFR(q_l, behind)$ $AnsFR(q_i, coveredby) \wedge AnsFR(q_j, coveredby) \wedge AnsFR(q_k, front) \Rightarrow AnsFR(q_l, front)$ $AnsFR(q_i, coveredby) \wedge AnsFR(q_j, coveredby) \wedge AnsFR(q_k, near) \Rightarrow AnsFR(q_l, near)$ $AnsFR(q_i, coveredby) \wedge AnsFR(q_j, coveredby) \wedge AnsFR(q_k, far) \Rightarrow AnsFR(q_l, far)$ $AnsFR(q_i, coveredby) \wedge AnsFR(q_j, coveredby) \wedge AnsFR(q_k, disconnected) \Rightarrow AnsFR(q_l, disconnected)$ $left(q_0) \wedge left(q_1) \wedge contain(q_2) \Rightarrow left(q_3)$ $left(q_0) \wedge left(q_1) \wedge cover(q_2) \Rightarrow left(q_3)$ $right(q_0) \wedge right(q_1) \wedge contain(q_2) \Rightarrow right(q_3)$ $right(q_0) \wedge right(q_1) \wedge cover(q_2) \Rightarrow right(q_3)$ $above(q_0) \wedge above(q_1) \wedge contain(q_2) \Rightarrow above(q_3)$ $above(q_0) \wedge above(q_1) \wedge cover(q_2) \Rightarrow above(q_3)$ $below(q_0) \wedge below(q_1) \wedge contain(q_2) \Rightarrow below(q_3)$ $below(q_0) \wedge below(q_1) \wedge cover(q_2) \Rightarrow below(q_3)$ $behind(q_0) \wedge behind(q_1) \wedge contain(q_2) \Rightarrow behind(q_3)$ $behind(q_0) \wedge behind(q_1) \wedge cover(q_2) \Rightarrow behind(q_3)$ $front(q_0) \wedge front(q_1) \wedge contain(q_2) \Rightarrow front(q_3)$ $front(q_0) \wedge front(q_1) \wedge cover(q_2) \Rightarrow front(q_3)$ $near(q_0) \wedge near(q_1) \wedge contain(q_2) \Rightarrow near(q_3)$ $near(q_0) \wedge near(q_1) \wedge cover(q_2) \Rightarrow near(q_3)$ $far(q_0) \wedge far(q_1) \wedge contain(q_2) \Rightarrow far(q_3)$ $far(q_0) \wedge far(q_1) \wedge cover(q_2) \Rightarrow far(q_3)$ $disconnected(q_0) \wedge disconnected(q_1) \wedge contain(q_2) \Rightarrow disconnected(q_3)$ $disconnected(q_0) \wedge disconnected(q_1) \wedge cover(q_2) \Rightarrow disconnected(q_3)$

Table 8: The conversion from spatial logical rules proposed in (Mirzaee and Kordjamshidi, 2022) to logical constraints used in our experiment.

```
above = question(name="above")
below = question(name="below")
behind = question(name="behind")
front = question(name="front")
near = question(name="near")
far = question(name="far")
disconnected = question(name="disconnected")
touch = question(name="touch")
overlap = question(name="overlap")
coveredby = question(name="coveredby")
inside = question(name="inside")
cover = question(name="cover")
contain = question(name="contain")
output_for_loss = question(name="output_for_loss")

# Only one label of opposite concepts
exactL(left, right)
exactL(above, below)
exactL(behind, front)
exactL(near, far)
exactL(disconnected, touch)

# Inverse Constrains
inverse = Concept(name="inverse")
inv_question1, inv_question2 = inverse.has_a(arg1=question, arg2=question)

# First inverse relation, allow inverse back and forth
inverse_list1 = [(above, below), (left, right), (front, behind), (coveredby, cover),
                 (inside, contain)]

for ans1, ans2 in inverse_list1:
    ifL(andL(ans1('x'), existsL(inverse('s', path=('x', inverse)))),
        ans2(path=('s', inv_question2)))

    ifL(andL(ans2('x'), existsL(inverse('s', path=('x', inverse)))),
        ans1(path=('s', inv_question2)))

# 2 PMD : = entropy + beta * constraint_loss ( Train with no-constraint first then working on)
# symmetric
```

```
inverse_list2 = [(near, near), (far, far), (touch, touch), (disconnected, disconnected), (overlap, overlap)]
for ans1, ans2 in inverse_list2:
    ifL(andL(ans1('x'), existsL(inverse('s', path=('x', inverse)))),
        ans2(path=('s', inv_question2)))

# Transitive constrains
transitive = Concept(name="transitive")
tran_quest1, tran_quest2, tran_quest3 = transitive.has_a(arg11=question, arg22=question, arg33=question)

transitive_1 = [left, right, above, below, behind, front, inside, contain]

for rel in transitive_1:
    ifL(andL(rel('x'),
             existsL(transitive("t", path=('x', transitive))),
             rel(path=('t', tran_quest2))),
        rel(path=('t', tran_quest3)))
# Transitive of cover and contain
transitive_2 = [(coveredby, inside), (cover, contain)]
for rel1, rel2 in transitive_2:
    ifL(andL(rel2('x'),
             existsL(transitive("t", path=('x', transitive))),
             rel1(path=('t', tran_quest2))),
        rel2(path=('t', tran_quest3)))

# Transitive of inside/cover with position
transitive_3_1 = [inside, coveredby]
transitive_3_2 = [left, right, above, below, behind, front, near, far, disconnected]
for rel1 in transitive_3_1:
    for rel2 in transitive_3_2:
        ifL(andL(rel1('x'),
                 existsL(transitive("t", path=('x', transitive))),
                 rel2(path=('t', tran_quest2))),
            rel2(path=('t', tran_quest3)))

# Transitive + topo constrains
tran_topo = Concept(name="transitive_topo")
tran_topo_quest1, tran_topo_quest2, tran_topo_quest3, tran_topo_quest4 = tran_topo.has_a(arg111=question, arg222=question, arg333=question, arg444=question)
# (x inside y) + (h inside z) + (y direction z) => (x direction h)
tran_topo_2_1 = [inside, coveredby]
tran_topo_2_2 = [left, right, above, below, behind, front, near, far, disconnected]
for rel1 in tran_topo_2_1:
    for rel2 in tran_topo_2_2:
        ifL(andL(rel1('x'),
                 existsL(tran_topo('to', path=('x', tran_topo))),
                 rel1(path=('to', tran_topo_quest2)),
                 rel2(path=('to', tran_topo_quest3))
                 ),
            rel2(path=('to', tran_topo_quest4)))

tran_topo_3_1 = [left, right, above, below, behind, front, near, far, disconnected]
tran_topo_3_2 = [contain, cover]
for rel1 in tran_topo_3_1:
    for rel2 in tran_topo_3_2:
        ifL(andL(rel1('x'),
                 existsL(tran_topo('to', path=('x', tran_topo))),
                 rel1(path=('to', tran_topo_quest2)),
                 rel2(path=('to', tran_topo_quest3))),
            rel1(path=('to', tran_topo_quest4)))
```

E.3 Graph for FR on StepGame

```
story = Concept(name="story")
question = Concept(name="question")
story_contain, = story.contains(question)

left = question(name="left")
right = question(name="right")
above = question(name="above")
below = question(name="below")
lower_left = question(name="lower-left")
lower_right = question(name="lower_right")
upper_left = question(name="upper_left")
upper_right = question(name="upper_right")
overlap = question(name="overlap")
```