Large Language Model with Curriculum Reasoning for Visual Concept Recognition

Yipeng Zhang
DCST, Tsinghua University
Beijing, China
zhang-yp22@mails.tsinghua.edu.cn

Xin Wang*
DCST, BNRist, Tsinghua University
Beijing, China
xin_wang@tsinghua.edu.cn

Hong Chen
DCST, Tsinghua University
Beijing, China
h-chen20@mails.tsinghua.edu.cn

Jiapei Fan
Alibaba Group
Hangzhou, China
jiapei.fjp@alibaba-inc.com

Weigao Wen
Alibaba Group
Hangzhou, China
weigao.wwg@alibaba-inc.com

Hui Xue
Alibaba Group
Hangzhou, China
hui.xueh@alibaba-inc.com

Hong Mei
MoE Lab, Peking University
Beijing, China
meih@pku.edu.cn

Wenwu Zhu*
DCST, BNRist, Tsinghua University
Beijing, China
wwzhu@tsinghua.edu.cn

Abstract

Visual concept recognition aims to capture the basic attributes of an image and reason about the relationships among them to determine whether the image satisfies a certain concept, and has been widely used in various tasks such as human action recognition and image risk warning. Most existing works adopt deep neural networks for visual concept recognition, which are black-box and incomprehensible to humans, thus making them unacceptable for sensitive domains such as prohibited event detection and risk early warning etc. To address this issue, we propose to combine large language model (LLM) with explainable symbolic reasoning via curriculum reweighting to increase the interpretability and accuracy of visual concept recognition in this paper. However, realizing this goal is challenging given that i) the performance of symbolic representations are limited by the lack of annotated reasoning symbols and rules for most tasks, and ii) the LLMs may suffer from knowlege hallucination and dynamic open environment. To address these issues, in this paper, we propose CurLLM-Reasoner, a curriculum reasoning method based on symbolic reasoning and large language model for visual concept recognition. Specifically, we propose a novel rule enhancement module with a tool library, which fully leverage the reasoning capability of large language models and can generate human-understandable rules without any annotation. We further propose a curriculum data resampling methodology to help the large language model accurately extract from easy to complex rules at different reasoning stages. Extensive experiments on various datasets demonstrate that CurLLM-Reasoner can achieve the state-of-the-art visual concept recognition results with explainable rules while free of human annotations.

CCS Concepts

• **Computing methodologies** → **Knowledge representation and reasoning**; *Computer vision*; Natural language processing.

Keywords

Large Language Models; Reasoning; Curriculum Learning; Visual concept recognition

ACM Reference Format:
Yipeng Zhang, Xin Wang, Hong Chen, Jiapei Fan, Weigao Wen, Hui Xue, Hong Mei, and Wenwu Zhu. 2024. Large Language Model with Curriculum Reasoning for Visual Concept Recognition. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3637528.3671653

1 Introduction

Visual concept recognition, such as human action recognition [7, 39, 60] and image risk warning, aims to determine whether the image satisfies a certain concept by extracting the basic attributes from the image and reasoning about the relationships among these attributes. Compared to the basic object detection or instance segmentation visual tasks that only involve visual attribute extraction, visual concept recognition remains a more challenging and worth exploring problem.

On the one hand, traditional deep learning based methods have achieved remarkable results in visual concept recognition tasks [7, 18, 23]. Nevertheless, due to their black box and unexplainable characteristics, the traditional deep learning based methods face challenges in being accepted for sensitive tasks involving sensitive data or specific objectives, such as healthcare and risk warning. On the other hand, symbolic reasoning [2], by emulating the cognitive reasoning process of humans, symbolizes various conditional statements and employs mathematical theorems for deduction and

*Corresponding Authors.

KDD '24, August 25–29, 2024, Barcelona, Spain

ACM ISBN 979-8-4007-0490-1/24/08
https://doi.org/10.1145/3637528.3671653

inference. By formalizing the reasoning process, symbolic reasoning methods ultimately yield a human-interpretable reasoning rule. Some previous researches [1, 13, 14, 17, 22, 26, 30, 49, 53, 62] integrate the perceptual capability of neural networks with the reasoning ability of symbolic methods. The neural perception component is expected to guide the learning of logical rules, while the logical reasoning component, in turn, supervises the learning of neural perception by generating logical formulas. The goal is to achieve mutual enhancement between these two components. This approach has made great success with traditional reasoning tasks such as some numeric calculation [13, 53]. However, more complex tasks such as visual concept recognition often require more sophisticated domain knowledge and symbolic operators, necessitating the composition of multiple layers of logic to accomplish the tasks. Yet, there is generally a lack of high-quality annotated and defined reasoning rules and symbolic representations, which limits the further development of symbolic methods.

Considering the success of large language models [12, 19, 27, 32, 34, 42, 48, 52, 68], a natural approach is to use a large language model to define reasoning rules instead of relying on human input and annotation. However, current approaches that employ large language models for reasoning [58, 60, 64] often conduct self-evaluation on the generated rules, which leads to the following issues: (1) The potential knowledge hallucination of the large language model undermines the trustworthiness of the confidences. Due to some security and privacy concerns, many companies or agencies are compelled to use open-source language models, exacerbating this problem. (2) Simple self-evaluation overlooks the impact of the data from the dynamic and open environment, e.g., when encountering unseen multimodal data, the large language model may struggle to accurately evaluate the generated rules. (3) As reasoning progresses, the generated rules by the large language model become more complex. Using the same data for evaluation during the whole rule generating process may fail to generate more complex rules.

To address these issues, we propose CurLLM-Reasoner, a novel curriculum reasoning method based on large language models. Specifically, we utilize a large language model and devise an iterative approach to generate reasoning rules. To aid the comprehension of multi-modal information such as images and text in the dynamic and open environment, we introduce a tool library that dynamically selects appropriate tools to assist in constructing reasoning rules. With the knowledge of the tools, the hallucination problem can be largely alleviated and the unseen multi-modal information can be tackled by the corresponding multi-modal tools. As the generated rules at different iterations vary in difficulty, we employ a curriculum learning approach to gradually generate easy to complex rules. This involves dynamically evaluating the difficulty of the data and resampling to choose easy to hard data for easy to complex rule generation in an accurate way. Extensive experiments are conducted on several datasets to demonstrate that our proposed CurLLM-Reasoner is able to generate accurate and human-readable rules and outperforms several state-of-the-art baselines.

Our main contributions are summarized as follows:

- We propose a new CurLLM-Reasoner approach that leverages large models together with a novel tool library to generate human-interpretable reasoning rules for visual concept recognition.
- We introduce a novel framework that combines curriculum learning with LLM reasoning, allowing for adaptive adjustment of difficulty levels based on the reasoning stages, which improves the performance of the reasoning method.
- We conduct extensive experiments on several real-world datasets to show that the proposed CurLLM-Reasoner can be applied to different visual concept recognition tasks and achieves SOTA performance.

2 Related Works

LLM for Reasoning Tasks. Large language models (LLM) [12, 16, 19, 48, 52, 70] exhibit a high capacity to understand, generate, and manipulate textual information, making them valuable tools for various natural language processing tasks such as machine translation, text generation, sentiment analysis, and question-answering [27, 32, 34, 42].

As the extensive knowledge repository possessed by LLMs across various domains, some researchers have tried to enable LLMs to engage in reasoning tasks through several novel techniques such as prompt engineering. The Chain-of-thought [58] (CoT) method serves as a representative method in which large language models are guided through prompts to provide rationales or justifications before generating answers. Tree-of-thoughts [64] expands upon the CoT approach by transforming the reasoning process from a linear chain to a tree structure, which enables large language models to better handle the reasoning tasks. And many previous works [10, 20, 45, 50, 60, 65] have extended large language models reasoning to more complex tasks, such as Symbol-LLM [60], which applies LLM to reason human action recognition tasks and derive human-interpretable reasoning rules for this task.

However, these methods rely on self-evaluation by the LLMs, which is not entirely reliable in sensitive domains like risk assessment due to the existence of knowledge hallucination. Additionally, the high computational cost associated with training multimodal LLMs poses challenges in accurately evaluating and processing multimodal data in dynamic and open environments.

Neural Symbolic Reasoning. Deep neural networks have achieved tremendous success in many tasks, but when it comes to tasks requiring reasoning ability and explainability, there still remain a lot of problems for deep neural networks. In contrast, symbolic reasoning which emulates human cognitive reasoning processes using symbolic operators, has shown promising results in various numerical reasoning tasks [13, 14, 22, 53]. However, tasks involving semantic understanding still pose significant challenges [5, 43, 44].

To combine the reasoning capabilities of traditional symbolic methods with the learning abilities of deep neural networks, some approaches introduce novel structures that enable neural networks to possess reasoning capabilities. Specifically, some works [1, 26, 30, 49, 62] employ modular networks with logical supervision, allowing deep networks to utilize specialized structures to obtain human-readable symbolic representations or reasoning rules, such

as programs [26] or trees [49]. Other works leverage the differentiability of deep networks and redesign specific tasks to transform symbolic systems into differentiable operations for optimization. For example, [40] extends Prolog [59], and [4] designs a differentiable Forth interpreter. Other works [2, 13, 17, 31, 38, 41, 67] treat the deep networks as perception modules and utilize other complex reasoning systems to handle symbolic problems. They employ DNNs to capture semantic information and abstract it into neural symbols, while a symbolic executor serves as the reasoning system to infer from these symbols and derive the final answers.

However, existing neural symbolic reasoning methods require accurate annotations of intermediate reasoning symbols and rules, preventing extending them to more complex and general tasks.

Curriculum Learning. Inspired by human learning processes, curriculum learning attempts to train deep neural networks by starting with the simple samples and gradually increasing the difficulty of the data samples [3, 37, 54, 57, 71]. It has achieved remarkable success in various fields, such as recommendation systems [8, 11, 55, 61], combinatorial optimization [69], neural architecture search [46, 66, 72], multimodal learning [73], and video grounding [9, 36].

The key aspect of curriculum learning is how to accurately assess the difficulty level of samples and a training scheduler to decide the input sequence or weights of data subsets, leading to the development of various methods. Baby Step [51] is one of the most intuitive approaches, utilizing simple predefined measures such as sentence length for NLP tasks. However, such a method requires strong domain knowledge and is not easily transferred to many other specific domains. Thus, several works have proposed automatic curriculum learning. Self-paced methods [6, 35] leverage the training loss to evaluate the difficulty of data. The transfer teacher approach [24] introduces a powerful teacher model and assesses the difficulty of data based on the performance of samples on the teacher model. The RL Teacher method [21] employs a reinforcement learning-based model as the teacher model and dynamically adjusts the difficulty of data based on the feedback from the current model.

In this work, we use the idea of curriculum learning to dynamically resample the dataset to adapt different levels of data difficulty during the rule generation process.

2.1 Methodology

The overall framework of our CurLLM-Reasoner is shown in Figure 1. To incorporate reasoning knowledge from various domains, we employ a large language model as reasoner. To fully leverage the reasoning capabilities of the large language model, we devise a rule enhancement methodology and introduce several auxiliary tools to dynamically generate human-readable rules. To more comprehensively and reasonably evaluate the quality of the generated rules based on the current generating state, we adopt a curriculum learning approach to dynamically resample the dataset.

In the following subsections, we first give the problem formulation and preliminaries. Then, we give a brief overview of the main parts of our method and detail the rule enhancement strategy based on a large language model to find the reasoning rules in section 2.3 and describe our curriculum resampling method in section 2.4.

2.2 Preliminary

Definition 1 (Visual Concept Recognition Through Reasoning). *Given a set of images as input, the task is generating* ***a set of reasoning rules*** *to determine whether visual contents or objects carried in these images belong to specific concepts.*

These concepts may vary depending on different specific tasks, such as recognizing whether particular visual objects are engaged in certain activities and whether particular visual contents may involve certain risks. Suppose $C = \{c_1, c_2, \cdots, c_K\}$ represents a collection of visual concepts, where each c_k is a concept, i.e., a text that describes a certain action of a person or qualities of an object in an image. Given a dataset $\mathcal{X} = \{x_1, x_2, ..., x_n\}$ and each x_i is an image, $\mathcal{Y} = \{y_1, y_2, ..., y_n\}$ is the set of labels of the given samples, where each $y_i \in \mathbb{R}^K$. y_i^k is the $k-th$ dimension of y_i and is a binary value. If $y_i^k = 1$, x_i satisfies concept c_k.

Our purpose is to find a list of human-readable reasoning rules, $\{R_k\}$, where each R_k is the disjunctive normal form of several rules $r_k^1 \vee r_k^2 \vee ... \vee r_k^M$ and is used to reason whether a give data $x \in \mathcal{X}$ satisfies the concept c_k. Each rule r_k^i is a conjunctive normal form of various symbolic conditions $s_i^1 \wedge s_i^2 \wedge ... \wedge s_i^Q \quad c_k$, which means that we conduct a reasoning process about the concept c_k from the conditions set $\{s_i^q\}$. Typically, a condition s_i^q is an executable program accompanied by a natural language description to describe its purpose.

After we get a set of rules, we can use logical reasoning to get the final result. For each condition s_i^q of the rule r_k^i, it accepts $x \in \mathcal{X}$ as input and outputs the probability p_i^q that this data satisfies this condition. Since the conditions are conjunctive normal form, it is required that all conditions hold true for the conclusion to be valid. Therefore, we consider the minimum value of all conditions, i.e., $\min_q p_i^q$, as the probability of rule r_k^i being valid. The different rules are organized as disjunctive normal form, if any of the rules is satisfied, the conclusion is considered valid. Hence, we take the maximum value of these rules' probabilities as the final probability for the given sample, i.e., $p = \max_i \min_q p_i^q$. If p exceeds a predefined threshold, we consider that the sample satisfies the visual concept c_k.

Taking the concept of "a person is drinking" as an example, one possible rule is "*there is a person in the image* $\wedge$ *there is a bottle in the image* $\wedge$ *the people is holding the bottle* $\wedge$ *the head of the people and the bottle is close a person is drinking*". One possible executable program for the first condition is to detect whether there is a person in the image and one possible executable program for the third condition is to segment the person and the bottle in the image and check whether the masks of these two objects overlap.

2.3 Rule Enhancement with LLM

The first key problem of visual concept recognition through reasoning is to find the rules. When humans conduct reasoning, they often draw upon knowledge from various domains. However, traditional machine learning approaches struggle to achieve the same goal without extensively annotated data, particularly when it is required to generate human-readable reasoning rules. Upon recognizing the effectiveness of large language models across multiple domains, we employ LLM as the core for generating rules.

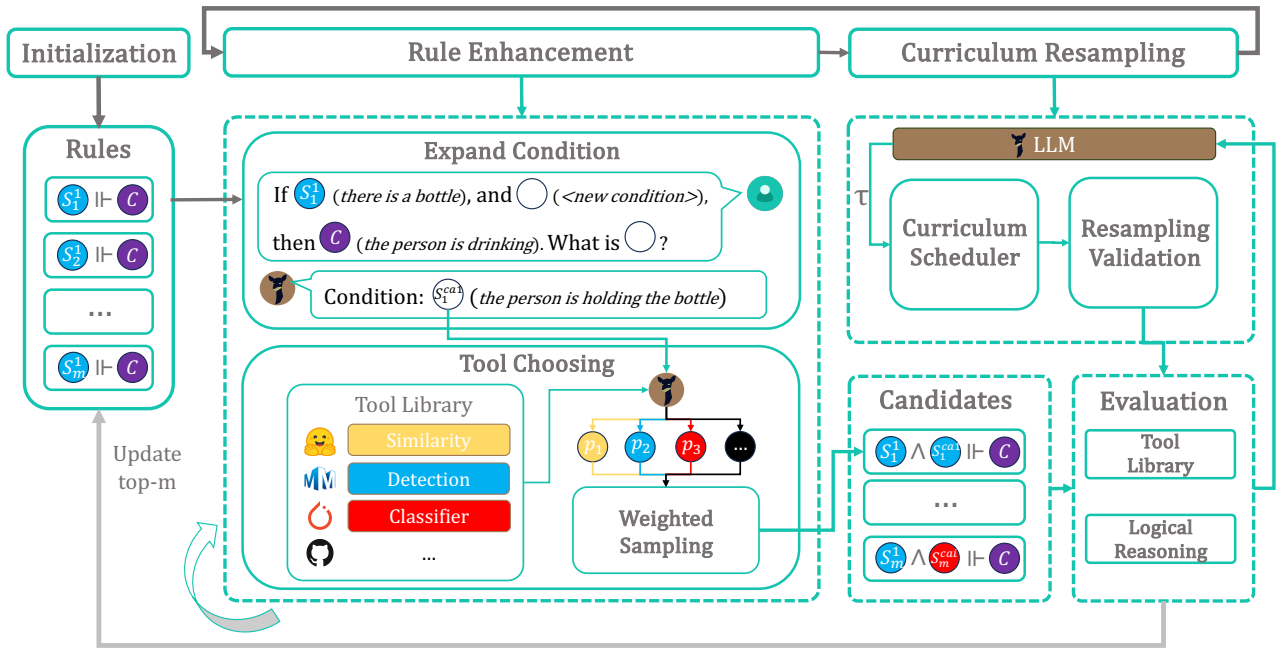

Figure 1: The framework of the proposed CurLLM-Reasoner. Our method iteratively generates the reasoning rules according to the present results. For each iteration, we first iterate through the current list of rules and expand on them to get various candidate rules. After that, we evaluate these candidates and select the top-m rules among them as the objects to be enhanced in the next iteration. At each iteration step, we dynamically adjust the difficulty level τ based on the selected data difficulty from the history as well as the evaluation metrics of the generated rules. This level τ is then passed into the curriculum scheduler for data resampling.

We consider a rule as a conjunction of multiple conditions and iteratively generate new conditions to expand our rule set. To accomplish this, we initialize a rule pool through positive samples of a certain concept. These initialized rules often represent certain intuitive attributes about the samples that satisfy the concept, such as the presence of specific elements in an image. Subsequently, we propose a method to enhance these initialized rules through interactions with a large language model. The objective is to capture the relationships between these intuitive attributes using the enhanced complex rules, thereby facilitating improved concept recognition.

Rule initialization. Suppose we are generating rules of concept c_k, and there is a positive dataset $X_k \subset X_{train}$, where $\forall x \in X_k$, x satisfies c_k. We perform captioning on the images to obtain textual descriptions for each positive sample. Subsequently, we conduct *part-of-speech* analysis on the textual descriptions to extract the nouns and calculate their frequencies across all positive samples. We select the top-m words with the highest frequencies. For each word w, we construct an initial rule that includes only one condition: "*there is a* w c_k" and thus get m initialized rules.

Expand Condition. After obtaining the initialized rules, we need to enhance them to obtain more complex rules. For each rule $s_i^1 \wedge s_i^2 \wedge ... \wedge s_i^Q$ c_k in the current rule set, we construct a prompt based on its conditions and target concept c_k to interact with the large language model and generate new conditions. According to [33], reasoning in the backward direction is significantly more efficient. And inspired by [60], the prompt can be formulated as:

- Please take a deep breath and answer the question. The reasoning chain is: if s_i^1, s_i^2, ..., s_i^Q and <condition>, then c_k. What is <condition>?

The <condition> output from the large language model represents the newly added condition s_i^{ca}.

Tool choosing. However, only relying on the large language model to conduct rule enhancement may suffer from the following two problems. On the one hand, large language models are often susceptible to knowledge hallucination and exhibit inherent instability, resulting in varying quality of their outputs. Therefore, it is necessary to apply filtering techniques to select reliable enhanced conditions. On the other hand, as a purely text-based model, the large language model lacks the ability to directly judge whether a given image satisfies a given condition, i.e., the large language model cannot determine whether the generated condition s_i^{ca} is a proper condition of the multi-modal input data, especially the image data. To address these challenges, we design a tool library, which integrates various pre-defined multi-modal function modules such as object detection modules. For each newly generated condition, we dynamically choose appropriate tools from the tool library to assist the large language model in verifying the generated rules effectively, which will be elaborated on in detail.

We employ the large language model to assist in tool selection. However, practical experience has shown that the large language model cannot directly output a specific tool, or it may be influenced by history, consistently yielding the same tool, lacking exploration of various tools, resulting in sub-optimal choices. Therefore, we utilize the large language model to provide probabilities for using each tool based on the currently generated new condition s_i^{ca}. To obtain these probabilities, we use the following prompt:

- You can solve problems with the following tools. [Tool 1]: [Description of the functionality of tool 1], ..., [Tool L]: [Description of the functionality of tool L]. Please take a deep breath and answer the question. There is an image and you need to find out whether s_i^{ca}. You will use the mentioned tools to solve the problem. Tell me the probabilities that each tool to be used. Please answer with the format as '[Tool 1]: <p_1>, [Tool 2]: <p_2>, ..., [Tool L]: <p_L>', where each <p> is a float value.

To further mitigate the potential impact of redundant choices and prevent repeatedly choosing the same tool, we reweight these probabilities based on the types and numbers of tools selected in history. For the tool l, we consider the output result p_l from the large language model as the original probability of choosing tool l. Then, we iterate through the rule set. For the condition s_i^q in these rules, if the tool used for s_i^q is l, we need to determine whether s_i^q is consistent with the current condition s_i^{ca} and get their similarity w_i^q. Afterward, we use the weighted probabilities to randomly select which tool to use at the current stage. The probability of choosing each tool can be represented as:

$$p_l = p_l(1 + \sqrt{\frac{2\log n_s}{\sum_i \sum_q w_i^q}}), \tag{1}$$

where n_s is the number of all conditions. The reweighing process is utilized to guarantee that the tool has been used in similar rules in history, we will decrease its probability of being chosen, so that the model can explore many more other tools, avoiding the problem of repeatedly choosing the same tools.

After expanding a new condition and tool choosing, we get a new candidate rule $s_i^1 \wedge s_i^2 \wedge ... \wedge s_i^Q \wedge s_i^{ca} \quad c_k$ with their tools. For example, for the concept "drinking", the origin rule may be "*there is a bottle (tool detection) the person is drinking*", and the enhanced rule may be "*there is a bottle (tool detection)* $\wedge$ *a person is holding the bottle (tool overlap) the person is drinking*". For each rule, we will repeat the rule enhancement process multiple times to obtain different candidate rules.

2.4 Curriculum Resampling

Once the set of candidate rules is obtained, the next key point is how to distinguish among various abilities of different generated rules. Since in the process of our proposed rule enhancement procedure, those earlier rules tend to emphasize the intuitive information of the data, while the later iterations for rule enhancement may focus more on the relationships among various features of the data. In other words, the initial rules are simpler, while the later enhanced rules become more complex. Therefore, when evaluating the rules, if we fix the validation dataset, all the simple rules in the earlier stages may fail to identify the difficult samples and may even lead to capturing false associations. Conversely, in the later stages, the simple data instances may prevent the model from generating complex rules, which results in sub-optimal performance.

Inspired by curriculum learning, we dynamically adjust the difficulty of each iteration step based on the performance of the current rules. We employ a superloss approach [6] to assess whether the data conforms to the specified difficulty level.

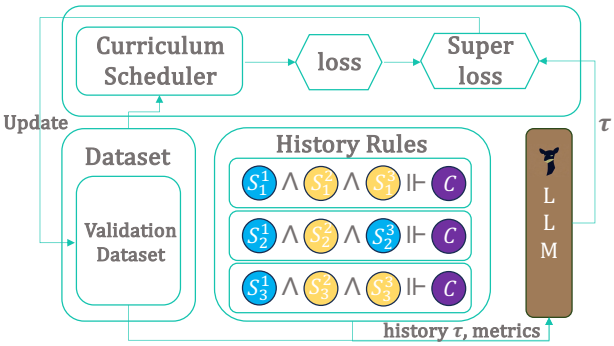

Figure 2: Curriculum resampling method. At the beginning of each iteration step, we determine the difficulty level τ for the current step based on the historical difficulty and rule scores. Then, we perform data resampling using the superloss method to obtain the validation dataset specifically for the current iteration step.

Superloss. Under the concept c_k, we employ the trainable curriculum method for data resampling. Given the difficulty threshold τ, we assign a trainable parameter σ_i to each data, representing the confidence of the data being difficulty τ. For all data $x_i \in \mathcal{X}$, we organize their labels as $\{y_i^k\}$ and consider it as a binary classification task. The loss for each data can be calculated as $l_i = -(y_i^k \log \hat{y}_i^k + (1 - y_i^k)\log(1 - \hat{y}_i^k))$, where $\hat{y}_i^k$ is the output of the curriculum scheduler. The curriculum scheduler can be any classifier suitable for the task, and for efficiency, we employ CLIP as the feature extractor and train a two-layer MLP network as the classifier. Afterward, we compute the superloss as follows:

$$L_\lambda(l_i, \sigma_i) = (l_i - \tau)\sigma_i + \lambda(\log \sigma_i)^2, \tag{2}$$

$$L = \frac{\sum_{i=1}^n L_\lambda(l_i, \sigma_i)}{n}, \tag{3}$$

where λ is a regularization hyper-parameter and L is the training objective. After the training procedure is completed, we select the top $(\frac{1}{T} * 100)\%$ data points with the highest confidence and consider them as the data that conform to the difficulty level τ, where T represents the total number of iterations.

Dynamic difficulty adjustment. To dynamically adjust the difficulty based on the different datasets and different concepts, we leverage the reasoning ability of the large language models. For iteration t, we input the difficulty level from the previous $t-1$ steps together with the corresponding average scores of generated rules during these steps into the LLM to obtain the difficulty level τ_t for step t. The prompt used is as follows:

- You are a researcher of curriculum learning, which mimics the human process of learning a domain from easy to hard,

i.e., the difficulty gradually increases with training, but sometimes you need to dynamically adjust the difficulty according to the current effect. The difficulty of the 1-st iteration is τ_1 and the mAP (mean average precision) metric is m_1... The difficulty of the $(t-1)$-th iteration is τ_{t-1} and the mAP metric is m_{t-1}. What is the difficulty of the next iteration should be? Please answer in the format with 'difficulty: <p>'.

3 Experiments

In this section, we empirically evaluate the performance of the proposed CurLLM-Reasoner, analyze the roles of the proposed modules, and provide some examples of the generated rules. Next, we will describe the baselines and the datasets we adopt. We do all the experiments with a NVIDIA A100-SXM4-40GB GPU. In the experiments, for each dataset, we employ curriculum scheduler with learning rate of $1e-3$ and weight decay of $1e-4$, and an Adam optimizer for optimization.

Baselines. To better show the effectiveness of our proposed method, we compared it with different baselines depending on the specific scenarios. On the one hand, we selected several state-of-the-art models for visual recognition tasks and several effective approaches in image classification. On the other hand, considering that our model incorporates a large language model and some foundational models, we also introduced some approaches that use the same foundational models as baselines for comparison.

- ResNet [25]. ResNet is a widely recognized neural network that has significantly advanced the field of computer vision. It addresses the challenge of training deep architecture by introducing residual connections.
- DenseNet [28]. In order to facilitate optimal information flow across network layers, DenseNet establishes direct connections between all layers. This design ensures seamless integration and propagation of information throughout the network, facilitating effective feature extraction and representation learning.
- ViT [15]. ViT divides an image into patches of fixed sizes and linearly embeds them with positional embeddings. After that, this methodology enables effective performance by leveraging the Transformer architecture.
- RelViT [39]. RelViT has introduced a concept-feature dictionary that aids inference by capturing visual features and facilitating relational queries. This concept-feature dictionary enables the extraction of relevant visual information and supports the reasoning process by leveraging the captured visual features.
- Ram++ [29]. By leveraging foundational models in the field of computer vision, Ram++ is able to zero-shot capture common visual concepts.
- CLIP [47]. The CLIP model is a state-of-the-art deep learning architecture that combines vision and language understanding. It is trained in a contrastive learning framework with a large-scale dataset to learn joint representations of images and their associated textual descriptions.
- Symbol-LLM [60]. Symbol-LLM capitalizes on the recent advancements in large language models and introduces a novel symbolic system that exhibits superior performance in a wide range of activity recognition tasks.

For fair comparison, we adopt the ViT-L-14@336px CLIP model as the image-text modality alignment model and employ Vicuna-7b [12] as the large language model for all methods.

Datasets. To facilitate a comprehensive comparison of the variations among different methods, we have carefully chosen multiple datasets for visual concept recognition tasks to conduct our evaluations.

- Stanford40 [63]. It is a classic dataset for recognizing human actions in still images which contains 40 diverse daily human actions, and all the images are obtained from Google, Bing, and Flickr.
- HICO [7]. It is a human-object interaction dataset which has a diverse set of interactions with common object categories. It has a total of 47,774 images, covering 600 categories of human-object interactions.
- Alibaba Risk. The Alibaba Risk Dataset is a collection of risk warning data collected by the Alibaba Group from its e-commerce platforms. It involves assessing the risk associated with products listed by merchants on platforms such as Taobao and Tmall. The dataset utilizes the product images provided by the merchants to determine if these products violate certain rules and pose risks. We have chosen the following categories of risks, ALI-politics: risk of government policies, ALI-pornography: risk of pornography, and the dataset of other prohibited content ALI-other.

Tools. To strike a balance between time cost and performance, we have chosen a limited but sufficient set of tools to aid in accomplishing our tasks. Below, we will describe some of the selected auxiliary tools we have utilized.

- Detection. For computer vision tasks, information extraction from images necessitates the use of detection techniques. In our experiments, we employ the SOLOv2 [56] framework as the detection tool.
- Overlap. For the interaction between objects in an image, their relative positions often play a crucial role in judging whether they have a relation. We utilize this tool to assess the proximity of two objects in an image. Firstly, we employ the detection tool to obtain the bounding boxes of the two objects of interest. Subsequently, based on the results, we determine whether the bounding boxes overlapped.
- Similarity. For certain tasks, the descriptions of symbols provided by large language models may require the joint utilization of both textual and visual modalities. Hence, we incorporate the CLIP tool to compute the similarity between text and images, allowing us to determine whether the generated symbolic operators fulfill the desired criteria.
- Classifier. For certain generated symbol operators, a natural binary characteristic may exist, where positive samples contain the desired information embodied by the operator, while negative samples lack this information. To address the problem, we employ a straightforward classifier to satisfy the symbol. In this context, for training samples, we initially

extract information using CLIP and subsequently employ a simple SVM classifier for classification.

Additionally, in cases where the detection tool encounters out-of-distribution (OOD) issues, meaning that the objects to be detected are outside the scope of the pre-trained model's detection capabilities, we resort to using the similarity or classifier tool as an alternative solution. We will evaluate and select whether to use the similarity tool or the classifier tool on the validation dataset.

3.1 Main Results

We evaluate the proposed method on the mentioned datasets, and for all datasets, we choose mean average precision (mAP) as the evaluation metric. The results are shown in Table 1. We find that our method outperforms all the baselines in all of the tasks. Moreover, from the obtained results, we can get several observations.

For traditional image classification models such as ResNet and DenseNet, their performances on public datasets are comparatively inferior to other methods. This is due to the fact that for these tasks, there exist subtle differences in information between images belonging to some of the categories, such as "riding a horse" and "feeding a horse". To achieve accurate classification results, models need to gather information about the relationships between different objects in the images and reasoning based on such relationships. ViT outperforms these models, possibly due to the enhancement brought by its number of parameters. However, all of these results remain black-box and lack interpretability.

The performance of RelViT heavily relies on the quality of the annotated concept-feature dictionary or reasoning rules provided. In the case of the HICO dataset, which includes expert-annotated files, RelViT outperforms the aforementioned classification models. However, for the Stanford40 dataset, where such annotated files are not provided, we could only generate some simple annotations through a detection model, resulting in worse performance. Additionally, due to the absence of annotations, RelViT cannot be applied to the Alibaba Risk Dataset. These limitations highlight the drawbacks of traditional reasoning-based approaches.

Pretrained foundational models based on extensive data such as CLIP, demonstrate superior performance in common scenarios, highlighting the ability of these foundational models, and enabling their application across a wide range of tasks. However, for some rare and sensitive scenarios, due to limitations in the available positive data, even after fine-tuning, the performance remains unsatisfactory.

Symbol-LLM outperforms all other baselines on public datasets, primarily because it not only leverages foundational models to acquire generalizable knowledge but also utilizes large language models to harness their powerful reasoning capabilities, thereby assisting the approach in achieving superior results. However, even after adjusting the prompts, Symbol-LLM still performs poorly on the risk warning tasks. Although Symbol-LLM can generate some good reasoning rules, it lacks the ability to determine whether the samples satisfy these rules. This limitation arises because Symbol-LLM relies entirely on the CLIP for evaluation and, compared to the large amount of pre-training data, it may struggle to capture crucial information due to the scarcity of various domains of rare risk samples.

Our method incorporates a tool library to enable accurate rule evaluation and utilize curriculum resampling, which adjusts the dataset according to different iterations, thereby getting the information of these rare risk samples and improving the performance.

3.2 Ablation Study

To better investigate the mechanisms of our proposed method and to demonstrate the necessity of certain experiment settings, we conducted the following ablation studies.

Proposed module effectiveness. In our method, the central components comprise the tool library, the reasoning module, and the curriculum learning method. We conduct an ablation study on these components, and the results are presented in Table 2

It can be observed that with the help of each module, there is a corresponding performance improvement. This indicates that each proposed component contributes positively to the reasoning result.

Additionally, we observe that for simpler tasks like the Stanford40 dataset, significant improvements can be achieved by solely employing the reasoning module. However, for more complex tasks, although specific rules can be obtained through the reasoning approach, it is challenging to accurately determine whether samples satisfy these complex rules using CLIP alone. Moreover, distinguishing the quality of the generated rules is also problematic. Therefore, incorporating the tool library module and the curriculum methods brings further improvement.

Hyper-parameters. We conducted an analysis of two hyperparameters that are prominently mentioned in the methodology section including the total number of iterations T and the top m rules to choose. In the following experiments, the ablation experiments about these two hyper-parameters are conducted on the Stanford40 dataset and the first 150 concepts of the HICO dataset. The results are shown in Table 3.

From the results, it can be seen that as the maximum number of iterations T increases, the performance first increases and then stabilizes. This indicates that using our method can find good results in a few steps. As m increases, the performance first increases and then decreases for the HICO dataset. On the one hand, this suggests that only a small number of rules are needed to capture the concepts, and an excessive number of rules may lead to sub-optimal rules being included in the rule set thus causing a decrease in the performance. On the other hand, simple logical reasoning may not work excellently on more complex datasets, and using a superior method such as ensemble learning to fuse different rules may be a future direction worth exploring.

Tools. We also explore several other tools, such as "number", which counts the quantity of a specific object in the image, "close", which calculates the distance between two objects and determines if it is below a certain threshold, and "calculate", which compares the quantity of two objects. However, during the experiments, we find that these tools are less frequently utilized during the generating process of the LLM and appeared less frequently in the top3 generated rules. The detailed numbers of those tools when generating the rules of Stanford40 datasets are provided in Table 4.

The poor performance of certain tools can be attributed to two factors. Firstly, some tools inherently involve challenging tasks that are difficult to generalize. Secondly, the difficulty lies in determining

Table 1: Model performance

Dataset	ResNet	DenseNet	ViT	RelViT	Ram++	CLIP	Symbol-LLM	**Ours**
Stanford40	0.8102	0.8263	0.9037	0.8143	0.8071	0.8472	0.9128	**0.9572**
HICO	0.1801	0.1747	0.4305	0.4496	0.4601	0.6337	0.6498	**0.6674**
ALI-pornography	0.3266	0.2022	0.1512	-	0.0344	0.0642	0.1030	**0.3396**
ALI-politics	0.1304	0.0402	0.0293	-	0.0086	0.0392	0.0405	**0.1577**
ALI-other	0.1795	0.0045	0.1712	-	0.0121	0.0235	0.0292	**0.2323**

Table 2: The impact of the proposed modules.

	Stanford40	HICO
CLIP	0.8472	0.6337
CLIP+Reasoning	0.9426	0.6376
CLIP+Reasoning+Curriculum	0.9533	0.6546
CLIP+Reasoning+Tools	0.9537	0.6493
Our	0.9572	0.6674

Table 3: The impact of the hyper parameters. $T = 0$ means using the initialized rules.

	Stanford40	25%HICO
$T = 0$	0.8530	0.6680
$T = 1$	0.9373	0.6751
$T = 2$	0.9572	0.6893
$T = 3$	0.9532	0.6902
$T = 4$	0.9531	0.6885
$m = 1$	0.9477	0.6905
$m = 2$	0.9501	0.6924
$m = 3$	0.9572	0.6893
$m = 4$	0.9531	0.6717
$m = 5$	0.9532	0.6625

Table 4: Number of tools that appeared in all rules throughout the generation process and that appeared in the top3 rules

	Total number	Top3 number	top3/total
detection	2015564	691992	0.3433
overlap	384440	137300	0.3571
number	1021512	313044	0.3065
close	131808	32592	0.2473
calculate	52462	9592	0.1832

suitable thresholds. For example, in the case of the 'close' task, it is challenging to define what constitutes "closeness" between two objects, as it often depends on the relative scale of objects in the image, leading to varying interpretations of the problem. And we discover that the LLM exhibits a significant preference for certain tools, such as detection and number. This preference can potentially be attributed to the higher frequency of these words in the majority of training data during the model's training process. This implies that 7B models, despite their limited capacity than other LLMs, still possess biases that prevent them from generalizing to all the cases.

Reasoning. In order to demonstrate the performance of reasoning, we construct the experiment that utilize some simple baselines which utilize all the tools. To obtain the scores, these tools can be combined in two different ways, conjunction manner and disjunction manner.

For certain concept c, conjunction manner means that data is predicted as a positive sample if and only if all the tools predicted the data as true $t_1 \wedge t_2 \wedge ... \wedge t_n \quad c$. Disjunction manner means that data is predicted as a positive sample if and only if there exists a tool that predicted the data as true $(t_1 \quad c) \vee (t_2 \quad c) \vee ... \vee (t_n \quad c)$.

Table 5: The impact of the reasoning method

	Conjunction	Disjunction	Our
stanford40	0.8703	0.9472	**0.9572**
HICO	0.5989	0.4745	**0.6674**

We can observe that for the stanford40 dataset, the disjunction baseline gets a better result, indicating that simple rules, which can be characterized by fewer premises per rule, are sufficient to achieve good performance for this easy dataset. Conversely, for the complex HICO dataset, the conjunction baseline outperforms the disjunction baseline, indicating that the need for more complex rules. This is the same as the findings presented in our paper and conforms to some intuitive understanding, where the difficulty of the task varied across different datasets, and we need to adaptively change the complexity of extracted rules according to the dataset, which shows the importance of reasoning with curriculum learning.

3.3 Case Study

Next, we will provide some generated rules and examples of how they can be applied for reasoning on given images.

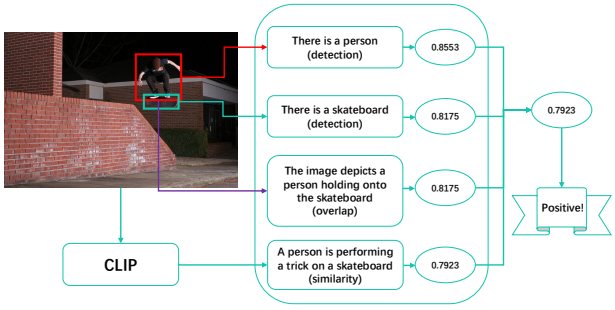

Figure 3: Jumping the Skateboard

The first example is from the HICO dataset, specifically the "jumping the skateboard" concept. The rule we generated for this concept is "*There is a person* $\wedge$ *There is a skateboard* $\wedge$ *The image*

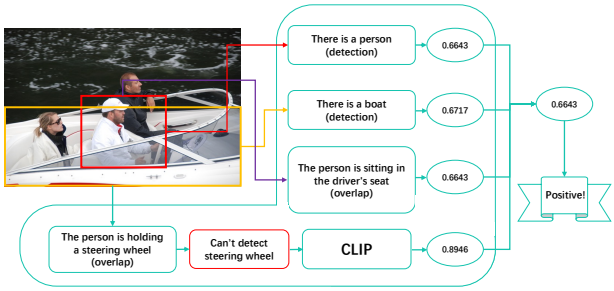

Figure 4: Driving the Boat

depicts a person holding onto the skateboard ∧ *A person is performing a trick on a skateboard* *A person is jumping the skateboard*."

The tools they employed are "detection", "detection", "overlap", and "similarity". For each tool, we evaluated the samples using the approach described in Section 3. For each conditional symbol, we obtain a probability value p indicating the likelihood of the current sample satisfying that condition. As the symbols are connected by conjunction, we take the minimum value among these probabilities as the final probability of the current sample passing the rule. If this value exceeds a certain threshold, we consider the sample as a positive example that satisfies the target concept.

When the specific target object cannot be detected by the present detection tool, we will choose to use either the similarity or classifier tool as a substitute based on the performance of the validation set as we have described in section 3. For example, in the case of "driving the boat" shown in Figure.4, the detection tool encounters an OOD problem and is unable to detect the "steering wheel". In response, we choose the classifier tool as a substitute, which utilizes CLIP to extract information from images and the description of conditions and performs simple binary categorization, and it also yields excellent results in this case.

From the experimental results and the provided case, it can be observed that our approach can provide effective and human-interpretable reasoning rules.

4 Discussion

In this section, we provide some discussion about the dataset of Alibaba and some corresponding tasks.

The task within the dataset is challenging due to the influence of policy regulations. Some risks are subtle and require specific conditions, such as the simultaneous presence of two particular objects or the absence of mosaic obscuring in critical areas, to be identified as risks. Consequently, there may be instances of misjudgment and noise during the annotation process. Other approaches, limited to data-level analysis, often exhibit inferior performance on complex datasets as they fail to capture the underlying decision rules. In addition, we conducted a fine-grained analysis where we extracted some typical sub-categories of ALI-politics. Based on this analysis, we have evaluated the strengths and the weaknesses of our method and some of the baselines. The experimental results are shown below.

The risk type "social hotspots", typically involves sensational social events, which means that it is usually new and unlikely to have appeared directly in the training data of the pre-trained model. As a result, the performance of the CLIP model is relatively poor in this aspect. However, these social events exhibit inherent patterns, and therefore, our method and Symbol-LLM can capture these patterns and form reasoning chains, leading to better performance. Besides, our approach introduces a tool library that utilizes the visual modality to assist reasoning and employs a curriculum method to further filter out excellent rules, resulting in improved performance.

Table 6: The results of some typical sub-categories

	CLIP	Symbol-LLM	Our
Social hotspots	0.1878	0.3167	**1.0000**
The 1989 events	0.0078	0.0051	**0.0061**

However, at the same time, current reasoning-based methods ,including our method, also have some limitations. For example, the risk "The 1989 events" or "The 1989 Tiananmen Square protests and massacre", refers to a bad event that occurred on June 4th, 1989. It is often restricted and banned in the Chinese internet environment. Therefore, it usually cannot be directly mentioned. Some individuals with malicious intent often try to circumvent these restrictions by using synonyms or homophones, such as referring to it as the "VIIV", where "VI" and "IV" represent the Roman numerals for six and four. These alternative references are diverse and constantly evolving. Additionally, the mere appearance of the numbers six and four does not necessarily indicate a violation of the risk. As a result, reasoning-based methods struggle to find consistently effective rules and, in the worst-case scenario, may degrade the performance of the pre-trained model they are based on.

Overall, most of the existing works are constrained by significant differences in data distributions between the pre-training, fine-tuning, and inference stages. These variations can impact the model's performance and limit its ability to generalize effectively across different data distributions and may lead to a struggle to fully comprehend the contextual information present in the current environment and extract relevant rules.

Our approach, on the other hand, leverages a tool library and only needs limited training on specific tools to assist large language models in comprehending unknown information. This supplementary training enables our method to capture rules more robustly and successfully tackle complex tasks.

5 Conclusion

In this paper, we propose CurLLM-Reasoner, a novel reasoning method based on large language models for visual concept recognition, which is able to generate accurate and human-readable reasoning rules. Experiments show that our proposed method can significantly improve the performance. The tool library module and curriculum resampling method enhance the accuracy of the extracted reasoning rules by large language models. A potential future work could focus on automatically generating tools which are semantically consistent, rather than only relying on predefined tools.

Acknowledgments

This work is supported by the National Key Research and Development Program of China No.2023YFF1205001, National Natural Science Foundation of China (No. 62222209, 62250008, 62102222), Beijing National Research Center for Information Science and Technology under Grant No. BNR2023RC01003, BNR2023TD03006, and Beijing Key Lab of Networked Multimedia.

References

[1] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Neural module networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 39–48.

[2] Yoshua Bengio et al. 2019. From system 1 deep learning to system 2 deep learning. In *Neural Information Processing Systems*.

[3] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*. 41–48.

[4] Matko Bošnjak, Tim Rocktäschel, Jason Naradowsky, and Sebastian Riedel. 2017. Programming with a differentiable forth interpreter. In *International conference on machine learning*. PMLR, 547–556.

[5] Patricia A Carpenter, Marcel A Just, and Peter Shell. 1990. What one intelligence test measures: a theoretical account of the processing in the Raven Progressive Matrices Test. *Psychological review* 97, 3 (1990), 404.

[6] Thibault Castells, Philippe Weinzaepfel, and Jerome Revaud. 2020. Superloss: A generic loss for robust curriculum learning. *Advances in Neural Information Processing Systems* 33 (2020), 4308–4319.

[7] Yu-Wei Chao, Zhan Wang, Yugeng He, Jiaxuan Wang, and Jia Deng. 2015. Hico: A benchmark for recognizing human-object interactions in images. In *Proceedings of the IEEE international conference on computer vision*. 1017–1025.

[8] Hong Chen, Yudong Chen, Xin Wang, Ruobing Xie, Rui Wang, Feng Xia, and Wenwu Zhu. 2021. Curriculum disentangled recommendation with noisy multi-feedback. *Advances in Neural Information Processing Systems* 34 (2021), 26924–26936.

[9] Houlun Chen, Xin Wang, Xiaohan Lan, Hong Chen, Xuguang Duan, Jia Jia, and Wenwu Zhu. 2023. Curriculum-listener: Consistency-and complementarity-aware audio-enhanced temporal sentence grounding. In *Proceedings of the 31st ACM International Conference on Multimedia*. 3117–3128.

[10] Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2023. Program of Thoughts Prompting: Disentangling Computation from Reasoning for Numerical Reasoning Tasks. *Transactions on Machine Learning Research* (2023).

[11] Yudong Chen, Xin Wang, Miao Fan, Jizhou Huang, Shengwen Yang, and Wenwu Zhu. 2021. Curriculum meta-learning for next POI recommendation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2692–2702.

[12] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality. https://lmsys.org/blog/2023-03-30-vicuna/

[13] Wang-Zhou Dai, Qiuling Xu, Yang Yu, and Zhi-Hua Zhou. 2019. Bridging machine learning and logical reasoning by abductive learning. *Advances in Neural Information Processing Systems* 32 (2019).

[14] Honghua Dong, Jiayuan Mao, Tian Lin, Chong Wang, Lihong Li, and Denny Zhou. 2018. Neural Logic Machines. In *International Conference on Learning Representations*.

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

[16] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. GLM: General Language Model Pretraining with Autoregressive Blank Infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 320–335.

[17] Xuguang Duan, Xin Wang, Peilin Zhao, Guangyao Shen, and Wenwu Zhu. 2022. DeepLogic: Joint Learning of Neural Perception and Logical Reasoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 4 (2022), 4321–4334.

[18] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. 2019. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*. 6202–6211.

[19] Luciano Floridi and Massimo Chiriatti. 2020. GPT-3: Its Nature, Scope, Limits, and Consequences. *Scope, Limits, and Consequences (November 1, 2020)* (2020).

[20] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. 2023. MME: A Comprehensive Evaluation Benchmark for Multimodal Large Language Models. *arXiv preprint arXiv:2306.13394* (2023).

[21] Alex Graves, Marc G Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. 2017. Automated curriculum learning for neural networks. In *international conference on machine learning*. Pmlr, 1311–1320.

[22] Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural turing machines. *arXiv preprint arXiv:1410.5401* (2014).

[23] Saurabh Gupta and Jitendra Malik. 2015. Visual semantic role labeling. *arXiv preprint arXiv:1505.04474* (2015).

[24] Guy Hacohen and Daphna Weinshall. 2019. On the power of curriculum learning in training deep networks. In *International conference on machine learning*. PMLR, 2535–2544.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[26] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. 2017. Learning to reason: End-to-end module networks for visual question answering. In *Proceedings of the IEEE international conference on computer vision*. 804–813.

[27] Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. 2023. VTimeLLM: Empower LLM to Grasp Video Moments. *arXiv preprint arXiv:2311.18445* (2023).

[28] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4700–4708.

[29] Xinyu Huang, Yi-Jie Huang, Youcai Zhang, Weiwei Tian, Rui Feng, Yuejie Zhang, Yanchun Xie, Yaqian Li, and Lei Zhang. 2023. Inject semantic concepts into image tagging for open-set recognition. *arXiv preprint arXiv:2310.15200* (2023).

[30] Drew A Hudson and Christopher D Manning. 2018. Compositional Attention Networks for Machine Reasoning. In *International Conference on Learning Representations*.

[31] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2901–2910.

[32] Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. 2023. ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and individual differences* 103 (2023), 102274.

[33] Seyed Mehran Kazemi, Najoung Kim, Deepti Bhatia, Xin Xu, and Deepak Ramachandran. 2022. Lambada: Backward chaining for automated reasoning in natural language. *arXiv preprint arXiv:2212.13894* (2022).

[34] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems* 35 (2022), 22199–22213.

[35] M Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. *Advances in neural information processing systems* 23 (2010).

[36] Xiaohan Lan, Yitian Yuan, Hong Chen, Xin Wang, Zequn Jie, Lin Ma, Zhi Wang, and Wenwu Zhu. 2023. Curriculum multi-negative augmentation for debiased video grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[37] Haoyang Li, Xin Wang, and Wenwu Zhu. 2023. Curriculum graph machine learning: a survey. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*. 667–6682.

[38] Qing Li, Siyuan Huang, Yining Hong, Yixin Chen, Ying Nian Wu, and Song-Chun Zhu. 2020. Closed loop neural-symbolic learning via integrating neural perception, grammar parsing, and symbolic reasoning. In *International Conference on Machine Learning*. PMLR, 5884–5894.

[39] Xiaojian Ma, Weili Nie, Zhiding Yu, Huaizu Jiang, Chaowei Xiao, Yuke Zhu, Song-Chun Zhu, and Anima Anandkumar. 2021. RelViT: Concept-guided Vision Transformer for Visual Relational Reasoning. In *International Conference on Learning Representations*.

[40] Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. 2018. Deepproblog: Neural probabilistic logic programming. *Advances in neural information processing systems* 31 (2018).

[41] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B Tenenbaum, and Jiajun Wu. 2018. The Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences From Natural Supervision. In *International Conference on Learning Representations*.

[42] Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. 2023. Recent advances in natural language processing via large pre-trained language models: A survey. *Comput. Surveys* 56, 2 (2023), 1–40.

[43] Allen Newell. 1980. Physical symbol systems. *Cognitive science* 4, 2 (1980), 135–183.

[44] Allen Newell and Herbert Simon. 1956. The logic theory machine–A complex information processing system. *IRE Transactions on information theory* 2, 3 (1956), 61–79.

[45] Gabriel Poesia, Kanishk Gandhi, Eric Zelikman, and Noah D. Goodman. 2023. Certified Reasoning with Language Models. (2023). arXiv:2306.04031 [cs.AI]

[46] Yijian Qin, Xin Wang, Ziwei Zhang, Hong Chen, and Wenwu Zhu. 2024. Multi-task graph neural architecture search with task-aware collaboration and curriculum. *Advances in neural information processing systems* 36 (2024).

[47] 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*. PMLR, 8748–8763.

[48] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. [n. d.]. Improving language understanding by generative pre-training. ([n. d.]).

[49] Amrita Saha, Shafiq Joty, and Steven CH Hoi. 2021. Weakly supervised neuro-symbolic module networks for numerical reasoning. *arXiv preprint arXiv:2101.11802* (2021).

[50] Abulhair Saparov, Richard Yuanzhe Pang, Vishakh Padmakumar, Nitish Joshi, Seyed Mehran Kazemi, Najoung Kim, and He He. 2023. Testing the General Deductive Reasoning Capacity of Large Language Models Using OOD Examples. *arXiv preprint arXiv:2305.15269* (2023).

[51] Valentin I Spitkovsky, Hiyan Alshawi, and Dan Jurafsky. 2010. From baby steps to leapfrog: How "less is more" in unsupervised dependency parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. 751–759.

[52] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[53] Andrew Trask, Felix Hill, Scott E Reed, Jack Rae, Chris Dyer, and Phil Blunsom. 2018. Neural arithmetic logic units. *Advances in neural information processing systems* 31 (2018).

[54] Xin Wang, Yudong Chen, and Wenwu Zhu. 2021. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 9 (2021), 4555–4576.

[55] Xin Wang, Zirui Pan, Yuwei Zhou, Hong Chen, Chendi Ge, and Wenwu Zhu. 2023. Curriculum co-disentangled representation learning across multiple environments for social recommendation. In *International Conference on Machine Learning*. PMLR, 36174–36192.

[56] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. 2020. Solov2: Dynamic and fast instance segmentation. *Advances in Neural information processing systems* 33 (2020), 17721–17732.

[57] Xin Wang, Yuwei Zhou, Hong Chen, and Wenwu Zhu. 2024. Curriculum Learning: Theories, Approaches, Applications, Tools, and Future Directions in the Era of Large Language Models. In *Companion Proceedings of the ACM on Web Conference 2024*. 1306–1310.

[58] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems* 35 (2022), 24824–24837.

[59] Jan Wielemaker, Tom Schrijvers, Markus Triska, and Torbjörn Lager. 2012. Swi-prolog. *Theory and Practice of Logic Programming* 12, 1-2 (2012), 67–96.

[60] Xiaoqian Wu, Yong-Lu Li, Jianhua Sun, and Cewu Lu. 2023. Symbol-LLM: Leverage Language Models for Symbolic System in Visual Human Activity Reasoning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

[61] Zihao Wu, Xin Wang, Hong Chen, Kaidong Li, Yi Han, Lifeng Sun, and Wenwu Zhu. 2023. Diff4Rec: Sequential Recommendation with Curriculum-scheduled Diffusion Augmentation. In *Proceedings of the 31st ACM International Conference on Multimedia*. 9329–9335.

[62] Yuan Yang and Le Song. 2019. Learn to Explain Efficiently via Neural Logic Inductive Learning. In *International Conference on Learning Representations*.

[63] Bangpeng Yao, Xiaoye Jiang, Aditya Khosla, Andy Lai Lin, Leonidas Guibas, and Li Fei-Fei. 2011. Human action recognition by learning bases of action attributes and parts. In *2011 International conference on computer vision*. IEEE, 1331–1338.

[64] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601* (2023).

[65] Yao Yao, Zuchao Li, and Hai Zhao. 2023. Beyond Chain-of-Thought, Effective Graph-of-Thought Reasoning in Large Language Models. *arXiv preprint arXiv:2305.16582* (2023).

[66] Yang Yao, Xin Wang, Yijian Qin, Ziwei Zhang, Wenwu Zhu, and Hong Mei. 2024. Data-Augmented Curriculum Graph Neural Architecture Search under Distribution Shifts. *Proceedings of the AAAI Conference on Artificial Intelligence* 38, 15 (Mar. 2024), 16433–16441.

[67] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. 2018. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. *Advances in neural information processing systems* 31 (2018).

[68] Zeyang Zhang, Xin Wang, Ziwei Zhang, Haoyang Li, Yijian Qin, and Wenwu Zhu. 2024. LLM4DyG: Can Large Language Models Solve Spatial-Temporal Problems on Dynamic Graphs?. In *Conference on Knowledge Discovery & Data Mining (ACM SIGKDD)*.

[69] Zeyang Zhang, Ziwei Zhang, Xin Wang, and Wenwu Zhu. 2022. Learning to solve travelling salesman problem with hardness-adaptive curriculum. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 9136–9144.

[70] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. arXiv:2306.05685 [cs.CL]

[71] Yuwei Zhou, Hong Chen, Zirui Pan, Chuanhao Yan, Fanqi Lin, Xin Wang, and Wenwu Zhu. 2022. Curml: A curriculum machine learning library. In *Proceedings of the 30th ACM International Conference on Multimedia*. 7359–7363.

[72] Yuwei Zhou, Xin Wang, Hong Chen, Xuguang Duan, Chaoyu Guan, and Wenwu Zhu. 2022. Curriculum-nas: Curriculum weight-sharing neural architecture search. In *Proceedings of the 30th ACM International Conference on Multimedia*. 6792–6801.

[73] Yuwei Zhou, Xin Wang, Hong Chen, Xuguang Duan, and Wenwu Zhu. 2023. Intra-and Inter-Modal Curriculum for Multimodal Learning. In *Proceedings of the 31st ACM International Conference on Multimedia*. 3724–3735.

A supplementary material

A.1 Implementation Details

We do all the experiments with a NVIDIA A100-SXM4-40GB GPU. In the experiments, for each dataset, we employ curriculum scheduler with learning rate of $1e-3$ and weight decay of $1e-4$, and an Adam optimizer for optimization. For ResNet [25] and DenseNet [28] models, a grid search is conducted to select the learning rate from $\{1e-4, 1e-3, 1e-2\}$, and weight decay from $\{0, 1e-5, 1e-4, 1e-3, 1e-3\}$. For ViT [15], we finetune on the google/vit-base-patch16-224-in21k model with learning rate of 1e-5. For Ram++ [29], we label the training set, computed the tf-idf values for each tag result of each class, and select the top distinctive terms with human-in-the-loop as indicators. Whenever these terms appeared in the tags of test data predicted by Ram++, the probability of that term is increased by $\frac{1}{3}$. In the case of CLIP [47], we finetune it using OpenAI's ViT-L-14@336px as a base model, and the probability of a data point satisfying the condition is determined by calculating the similarity between words and textual concepts. For Symbol-LLM [60], we employ Vicuna-7b as large language models.

A.2 CurLLM-Reasoner Algorithm

In this section, we provide the pseudo-code of the CurLLM-Reasoner in alg 1.

Algorithm 1 Curriculum Reasoning Method

Input: concept c_k, training dataset $\mathcal{X}_{train}$, training labels $\mathcal{Y}$
Output: Human-readable reasoning rules R_k

Get positive sample set $\mathcal{X}_k$ and labels $\mathcal{Y}_k$ for concept c_k
Initialize R_k^0 according to $\mathcal{X}_k$
$\tau_0 = 0$
$\mathcal{X}_{val}$ = Curriculum_Resampling($\tau_0, \mathcal{X}_{train}, \mathcal{Y}_k$)
R_k^0, m_0 = Evaluate_Rules($R_k^0, \mathcal{X}_{val}$)
for t = 1 to T **do**
  τ_t = Dynamic_Difficulty_Adjustment($\tau_0, m_0, ..., \tau_{t-1}, m_{t-1}$)
  $\mathcal{X}_{val}$ = Curriculum_Resampling($\tau_t, \mathcal{X}_{train}, \mathcal{Y}_k$)
  R_k^t = []
  for r_i in R_k^{t-1} **do**
    for j = 1 to enhancement_scale **do**
      $s_i^{ca_j}$ = Rule_Enhancement(r_i, c_k)
      $\text{tool}_i^{ca_j}$ = Tool_Choosing($R_k^t, R_k^{t-1}, s_i^{ca_j}$)
      $s_i^{ca_j} = (s_i^{ca_j}, \text{tool}_i^{ca_j})$
      R_k^t.append($r_i \wedge s_i^{ca_j}$)
    end for
  end for
  R_k^t, m_t = Evaluate_Rules($R_k^t, \mathcal{X}_{val}$)
end for
R_k^T = Outperform_Softrule(R_k^T)
return R_k^T

A.3 Module Analysis

In this section, we will provide further analysis regarding the tool library and curriculum resampling modules.

tool library. The tool library module aims to improve the evaluation process of data and obtains more accurate and discriminative rules. Without utilizing the tool library, to evaluate visual data with text description, we compute the similarity between images and corresponding textual descriptions of symbolic conditions using CLIP. However, large language models may provide some descriptions that are ambiguous for CLIP and make it challenging for CLIP to distinguish between similar samples. For instance, for the concept "riding the cow", one of the conditions generated by a large language model is "the people is on the cow". When the subjects in the text condition perfectly match the image, CLIP may overlook subtle relational details and provide consistent results between the positive sample and the negative sample as depicted in Figure 5. In such cases, incorporating our tool library introduces prior knowledge of human processing tasks under specific simple situation, enabling better differentiation between positive and negative examples and consequently enhancing the model's performance.

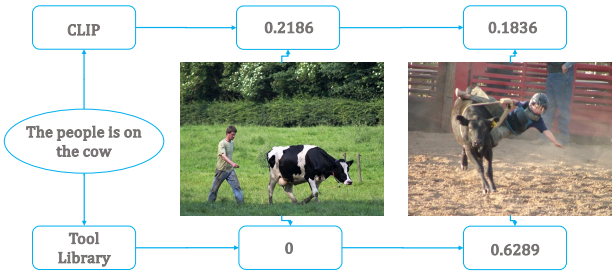

Figure 5: Case of riding the cow

curriculum resampling. The curriculum resampling methodology aims to capture more accurate rules by better adapting to the difficulty levels at different iterations. Without curriculum resampling, there is a possibility of learning false associations. For example, in the concept "cleaning the floor" from the Stanford40 dataset, without using the curriculum resampling, one of the generated rules is "*there is a floor* $\wedge$ *there is a house* $\rightarrow$ *the person is cleaning the floor*". This is because for the concept "cleaning the floor", almost all positive samples contain a house and a floor as background elements. This results in the dominance of information related to "house" and "floor" in the rule set at a very early iteration, which limits the reasoning capability of LLMs and makes it fail to fully understand the meaning of the concept itself and captures erroneous false associations. The use of curriculum resampling can mitigate this issue by dynamically adjusting the dataset during iterations. By modifying the composition of validation samples and increasing hard samples which may contain the concepts of "house" or "floor", the evaluation process restricts the rules that capture only these false associations, leading to more accurate results. After adopting currculum resampling, one of the captured rules is "*there is a broom* $\wedge$ *the broom is being held by the person* $\wedge$ *the broom is in contact with the floor* $\rightarrow$ *the person is cleaning the floor*".