KRIT: Knowledge-Reasoning Intelligence in vision-language Transformer

Kezhen Chen[§‡*] **Qiuyuan Huang**[‡] **Daniel McDuff**[‡]
Yonatan Bisk[§†] **Jianfeng Gao**[‡]
[§]Northwestern University [‡]Microsoft Research, Redmond [†]CMU
[§]kezhenchen2021@u.northwestern.edu, [†]ybisk@cs.cmu.edu,
[‡]{qihua,damcduff,jfgao}@microsoft.com

Abstract

Transformer-based pretraining techniques have achieved impressive performance on learning cross-model representations for various multi-modality tasks. However, most off-the-shelf models do not take advantage of commonsense knowledge and logical reasoning that are crucial to many real-world tasks. To this end, we introduce a new variant of the Transformer model for representation learning, **K**nowledge **R**easoning **I**ntelligence in Vision-Language **T**ransformer (KRIT). It utilizes a reasoning module and the commonsense knowledge embeddings extracted from text and detected image object tags to perform knowledge-grounded representation learning to improve model generalization and interpretability. KRIT is pretrained on a large image-text corpus and automatically extracted knowledge embeddings, and then finetuned on several downstream vision-language tasks. Experiments show that KRIT not only achieve the significant result on the OK-VQA task, but also for the first time, to the best of our knowledge, make the transformer model interpretable by illustrating its reasoning via a set of rules.

1 Introduction

Large-scale pretrained models have dramatically improved the quality of natural language processing (NLP) and vision-language models. Although these methods use image and text information as inputs and learn image-text alignments via well designed pretraining tasks, most still lack the external commonsense knowledge necessary for many tasks. The external knowledge is usually hard or impossible to be learned from standard datasets. More specifically, existing models often disregard the shared and complementary information provided by different modalities and do not leverage effectively the structure of knowledge graphs and commonsense reasoning. To address these challenges, we argue that modeling should leverage not only data of multiple modalities (i.e., vision and language) but also the rich structural and logical information embedded in commonsense knowledge bases. In this paper, we develop a general-purpose vision-language pretraining method, **K**nowledge **R**easoning **I**ntelligence in Vision-Language **T**ransformer (KRIT). We leverage a knowledge graph (Wikidata5m (Wang et al. 2020)) which has about five million entities and corresponding relations. The knowledge graph provides rich information useful for many vision-language understanding tasks. Figure 1 shows two visual question-answering examples that require external knowledge.

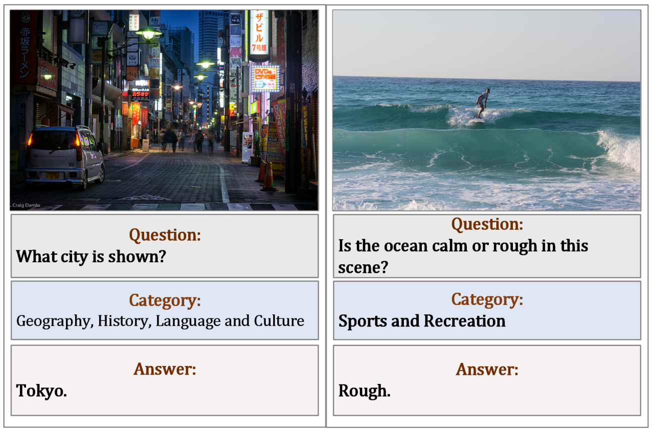


Figure 1: Examples of visual questions that requires knowledge.

To demonstrate the effectiveness of KRIT, we apply it to two downstream tasks: visual question answering (VQA) (Antol et al. 2015) and knowledge-based VQA (OK-VQA) (Marino et al. 2019). Experiments show that KRIT is effective in leveraging external knowledge for vision language understanding and reasoning tasks, and achieves the new significant on OK-VQA and competitive performance on VQA.

To summarize our contribution: i) We develop a knowledge-reasoning self-supervised Transformer using a knowledge graph to learn multi-modal representations, which includes physical properties, and ontological qualia/relations that are hard or impossible to recover from standard datasets; ii) We adapt knowledge-reasoning-patches besides use text and image bounding box features. Our approach enables the model to identify the types of knowledge and the space of entities, etc. that we are interested in and which may not be captured by detected objects. We promote these newer representations to handle a broader space of visual semantics than previous methods. iii) We leverage the Wikidata5M knowledge graph as the commonsense knowledge base, which includes entities, corresponding relations and information for general-purpose applications on multi-modal tasks. We

[*] Work done when Kezhen interned at Microsoft Research, Redmond.

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

present experiments and analysis to demonstrate the effectiveness of our approach. iv) We extend a Reasoner Patch Module to learn inference rules that can perform reasoning on extracted knowledge entities. This module enhances the interpretation of the existing deep learning model, optimizes the limitations of the internal small dataset and unifies the abundant external knowledge.

2 Related Work

Vision-Language Transformer. Multi-modal representation learning is essential for vision-language tasks, such as image-captioning, visual question answering and visual commonsense reasoning. Large-scale architectures based on Transformer (Vaswani et al. 2017) have achieved impressive performance by pretraining representations for natural language processing (NLP) tasks (Peters et al. 2018; Devlin et al. 2018; Yang et al. 2019; Liu et al. 2019; Radford et al. 2019; Brown et al. 2020). Recent works on vision-language pretraining (VLP) have shown that these large-scale pretraining methods can also lead to effective cross-modal representations (Lu et al. 2019a; Tan and Bansal 2019; Zhou et al. 2019; Chen et al. 2019; Alberti et al. 2019; Li et al. 2020a, 2019, 2020b; Zhang et al. 2021; Kim, Son, and Kim 2021). Most methods have two stages: Firstly, the model architecture is pretrained using a large set of image-text pairs. Then they are finetuned on task-specific vision-language tasks. For example, Lu et al. (2019a); Tan and Bansal (2019) propose multi-stream Transformer-based frameworks with co-attention to fuse these modalities. Zhou et al. (2019); Chen et al. (2019); Alberti et al. (2019); Li et al. (2020a, 2019, 2020b); Zhang et al. (2021) propose unified pretrained architectures to work on both visual-language understanding and visual-language generation tasks. Kim, Son, and Kim (2021) introduces a pretraining approach to learn self-attention representations directly on image patches. Although these models achieve impressive results on standard vision-language tasks, they do not use information from external knowledge graph. Our proposed KRIT architecture shows how the knowledge and reasoning information extracted from text and image facilitates learning more robust and knowledge-aware representations for vision-language tasks. Gardères et al. (2020) uses ConceptNet knowledge graph as the knowledge base to facilitate commonsense vision-language question-answering. However, it does not pretrain a Transformer model to unify multi-model inputs.

Language Transformer Models with External Knowledge. Numerous researches have injected knowledge into language pretraining models (Yu et al. 2020; Xu et al. 2021; Rosset et al. 2021; Zhou et al. 2020; He et al. 2020a; Xiong et al. 2019; He et al. 2020b; Agarwal et al. 2021) with an emphasis on NLP tasks. For example, Yu et al. (2020) extracts knowledge graph information of language inputs from Wikipedia, and use them to help the pretraining progress. Xu et al. (2021) injects domain-specific knowledge in pretraining language model for NLP tasks. Although, these studies work on using the knowledge in pretraining, these methods only focus on language tasks, and have not been applied to multi-modal transformers (e.g. for vision and language). Additionally, some proposed structures and representations are domain-specific and are hard to extend to new tasks. In this paper, we introduce a knowledge-based pretraining model using the Transformer architecture for multi-modal understanding and reasoning. The knowledge representations in our method can be easily extracted from massive data.

3 KRIT model

When humans reason about the world, they process multiple modalities and combine external knowledge related to these inputs. Inspired by this idea, we introduce a new pretraining approach, **K**nowledge **R**easoning **I**ntelligence in Vision-Language **T**ransformer (KRIT), which uses a multi-layer Transformer model to learn unified representations on external knowledge and vision-language inputs. Given an image-text pair, we extract the knowledge information from the text and image, and convert them to knowledge embedding vectors using a reasoner patch module. These embeddings are used as additional inputs for pretraining. Figure 2 shows the illustration of KRIT. In this section, we first present how we extract the external knowledge from the knowledge base and then we introduce the details of our pretraining approach.

3.1 Extracting Knowledge

For our experiments we choose the Wikidata5M knowledge base (Wang et al. 2020) as a source of external knowledge. Wikidata5M is a knowledge graph created upon Wikidata (Vrandecic and Krotzsch 2014) and contains about 5 million relevant and important real-world entities, where each entity has a corresponding text description. Given a piece of text T with n words $\{w_0, ..., w_n\}$, we first perform named entity recognition on T based on the Wikidata5M knowledge graph and generate an entity set E, which has m named entities $\{e_0, ..., e_m\}$. Each entity has a span in T with length of one or more words. Therefore, after the named entity recognition, words in T can be separated into two subsets P and Q. The first subset P has p words $\{w_0, ..., w_p\}$, that construct the recognized entities. The second subset Q has q words $\{w_0, ..., w_q\}$, which are the remaining words excluding the recognized entities. Next, a natural-language description is extraceted for each recognized entity. These knowledge descriptions are used in our pretraining stage.

3.2 Input

KRIT represents each image-text pair as six parts (w, k^w, t, p, v, k^t), where w is the sequence of word embeddings of the text, t is the word embedding sequence of the image object tags, p is the entity patches (alias of entities) extracted from Wikidata, v is the set of region feature vectors of the image, k^w is a set of KR-patches extracted from the text, and k^t is a set of KR-patches extracted from the object tags.

For each image-text pair, most of the existing VLP models represent the input pair as a sequence of word embeddings of the text, and a set of region vectors of the image. Inspired by Li et al. (2020b) and Zhang et al. (2021), we adopt an additional input, a sequence of object tags, which are used as anchor points to ease the learning of image-text alignment.

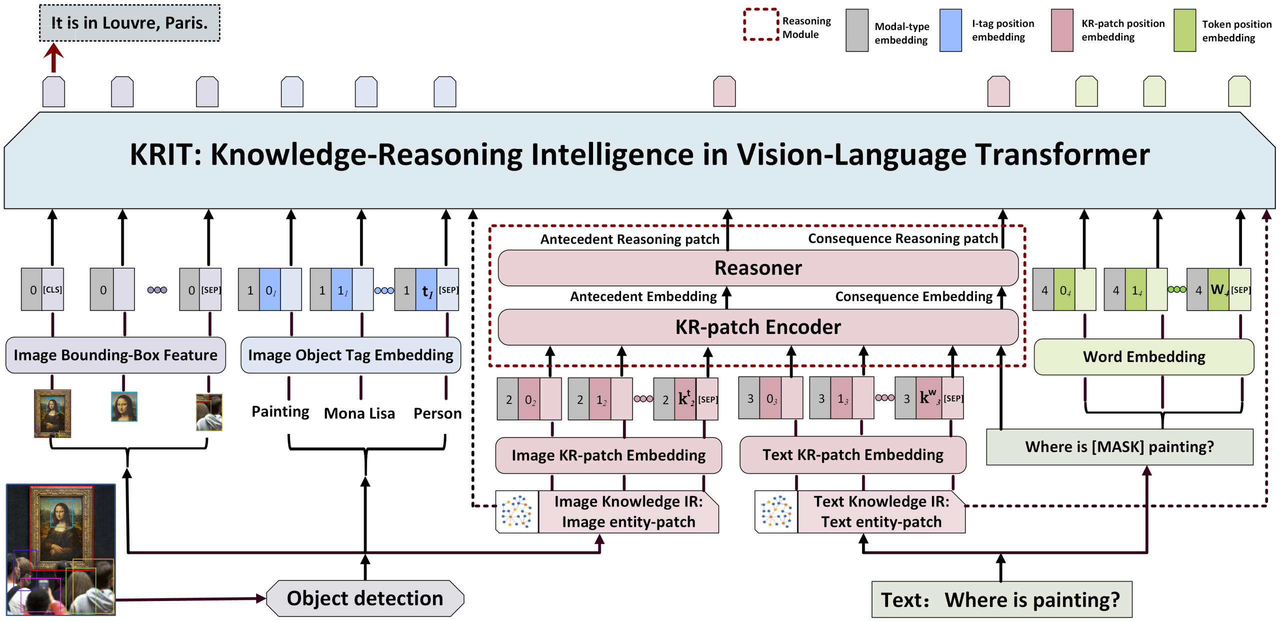


Figure 2: The KRIT model: Given an image-text pair, the input is represented as a tuple (w, k^w, v, t, k^t, p), where w is the sequence of embedding for the text, k^w is the sequence of text-KR-patch embeddings for the text entities extracted from the knowledge base, v is the sequence of embeddings for the image region bonding-box features, t is the sequence of embeddings for the object tags, and k^t is the sequence of image-KR-patch embeddings for image entities extracted from the knowledge base, p is the sequence of entity-patches.

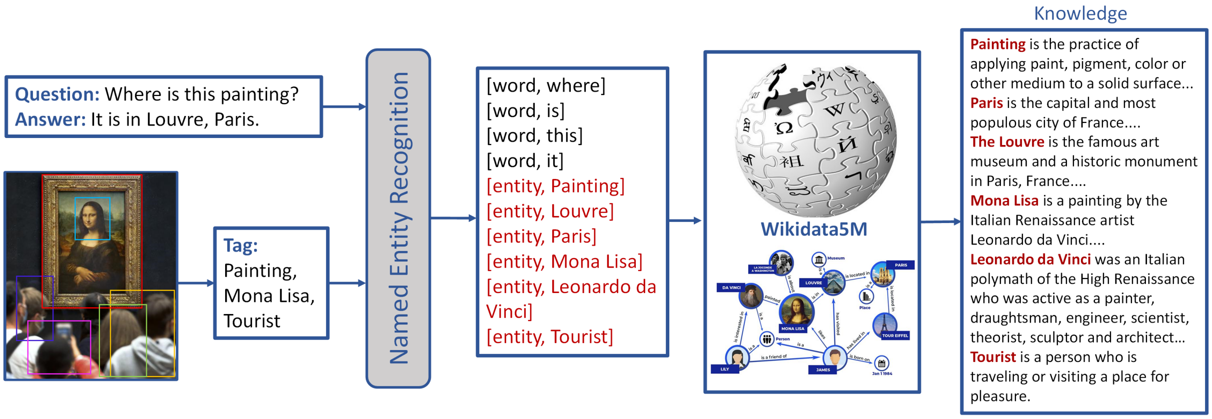
These object tags are the category names or semantically similar words of detected objects in the image. For generating v, we used a X152-C4 architecture as the object detection model (OD), which is initialized from an ImageNet-5K checkpoint (Deng et al. 2009). The OD model is pretrained on four vision datasets including Visual Genome (Krishna et al. 2016), COCO (Lin et al. 2014), Objects365 (Shao et al. 2019) and OpenImagesV5 (Kuznetsova et al. 2020). Given an image, the pretrained OD model generated the set of detected object names and the set of region features. Each region feature contains an vector of the image feature with 2048 dimension and a positional encoding of the region with 6 dimension. The image feature vector is concatenated with the positional encoding to construct the vectors in v, where each region vector in v has 2054 dimension. In pretraining, t uses the object tags in image captioning datasets and answer text in visual question answering datasets.

For each text-image pair, we also extract the KR-patches k^w and k^t from both the text and the image, where each KR-patch is an entity description corresponding an entity from Wikidata5M. One text side, the text in each pair is used for knowledge extraction and construct the k^w. On image side, we use the tags of objects in each image for knowledge extraction and generate the k^t. The k^w and k^t are concatenated together to construct a set of KR-patches k as inputs for the reasoner patch module. As each KR-patch has a corresponding entity, we adapt the similar ideas as using object tags. For the extracted entities, we use the alias of each entity as a entity patch and concatenate all entity patches to construct the p. These entity patches can be used as anchor points to facilitate alignment learning.

3.3 Reasoner Patch Module

The reasoner patch module contains a KR-patch encoder and a reasoner. This module takes two inputs, the sequence of words w of the text input and the sequence of KR-patches k. We use a pretrained BERT-Base model as the KR-patch encoder. w and k are encoded by the encode and the embedding vectors corresponding special token [CLS] are used as the outputs. The encoded vector r^w for w is regarded as antecedent vector and the encoded vector r^k for k is regarded as consequence vector, which are passed to reasoner. Reasoner module has b rule blocks, where each block learns to apply an inference rule on the KR-patches based on different types of text input. Inspired by classic inference systems, each rule block has three main components, an antecedent MLP f_i^a, a consequence MLP f_i^s and a learned rule matching vector h_i. The rule matching vector is used to check whether the antecedent is satisfied to apply the rule. Thus, we compute the attention weights between antecedent vector and the rule matching vectors of all rule blocks. The attention weights decide whether rules should be applied or not. Antecedent MLP in each block takes the antecedent vector r^w as input and outputs a vector a_i. Consequence MLP in each block takes the consequence vector r^k as input and outputs a vector s_i. The outputs of the reasoning module a and s are generated by computing the weighed sums of each a_i and c_i based on the rule attention weights. The reasoner module is presented in Figure 4 Finally, w, t, p, v, a and s are concatenated together as a sequence and passed as the input for the pretraining Transformer model. The following formulas describe the details of the reasoner patch module:

$$r^w = KRPatchEncoder(w) \tag{1}$$


Question: Where is this painting?
Answer: It is in Louvre, Paris.

Painting is the practice of applying paint, pigment, color or other medium to a solid surface... **Paris** is the capital and most populous city of France.... **The Louvre** is the famous art museum and a historic monument in Paris, France.... **Mona Lisa** is a painting by the Italian Renaissance artist Leonardo da Vinci.... **Leonardo da Vinci** was an Italian polymath of the High Renaissance who was active as a painter, draughtsman, engineer, scientist, theorist, sculptor and architect... **Tourist** is a person who is traveling or visiting a place for pleasure.

Figure 3: Overview of extracting knowledge on a text piece: given a text-image sample, we first perform named entity recognition on both text and image tags and detect a set of entities and rest of words. Then, we use these recognized entities to extract the text descriptions (KR-patches) corresponding to these entity.

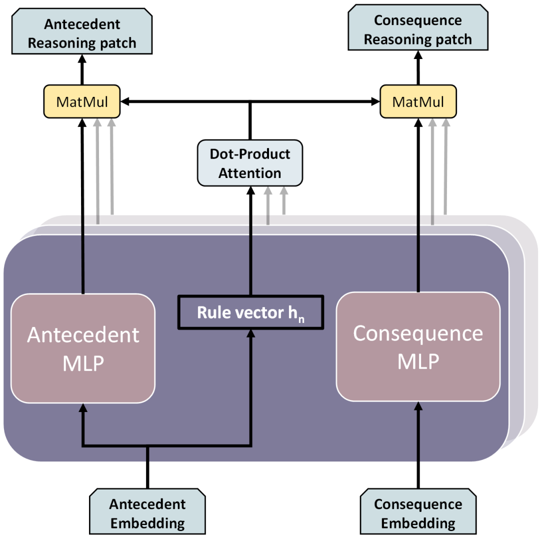


Figure 4: Reasoner in the Reasoning Module

$$r^k = KRPatchEncoder(k) \tag{2}$$

$$\beta_1, \beta_2, ..., \beta_n = Attention(r^w, h_1, h_2, ..., h_n) \tag{3}$$

$$a_i = f_i^a(r^w) \qquad s_i = f_i^s(r^k) \tag{4}$$

$$a = \sum_{i=1}^{n} \beta_i a_i \qquad s = \sum_{i=1}^{n} \beta_i c_i \tag{5}$$

3.4 Pretraining Objective

KRIT is pretrained with two types of objectives: sequence-level and token-level. Sequence-level objective distinguishes the representations of the text, image and the external knowledge. Token-level objective distinguishes the semantic space of inputs. Thus, we propose the novel KRIT pretrainig loss $\mathcal{L}_{pretraining}$ as in Equation 6, where $\mathcal{L}_{SL}$ is the loss from sequence-level pretraining and $\mathcal{L}_{TL}$ is the loss from token-level pretraining. Next, we introduce the details for each loss.

$$\mathcal{L}_{pretraining} = \mathcal{L}_{SL} + \mathcal{L}_{TL} \tag{6}$$

Sequence-Level Objective. The sequence-level loss $\mathcal{L}_{SL}$ is a four-way contrastive loss. Given the input tuples (w, k^w, t, p, v, k^t) from dataset D, we construct negative inputs by polluting the tuples to compute the loss. At each time, we keep the correct tuple or replace one of tuple elements including the text, tags or knowledge with a random element from other documents, which results three different types of polluted tuples: $(w_{neg}, k^w, t, p, v, k^t)$, $(w, k^w, t_{neg}, p, v, k^t)$ and $(w, k^w, t, p_{neg}, v, k^t)$. In 50% of the time, the correct tuple stays unchanged. In the rest of 50% of the time, the three types of negative samples have equal probabilities to be generated. During pretraining, KRIT model aims to predict whether the tuple is correct or polluted. Following the tradition of Transformer-based pretraining model, the encoding of [CLS] token can be used as the representation of the tuple input. We passed this encoding of [CLS] to a fully-connected layer $f(.)$ and predict four classes: the tuple is correct (c=0), w is unmatched (c=1), t is unmatched (c=2) or p are unmatched (c=3). Then the sequence-level loss is defined as:

$$\mathcal{L}_{SL} = -\mathbb{E}_{(w,t,p,v,k^w,k^t;c)\sim D} \log p(c|f(w, t, p, v, k^w, k^t)) \tag{7}$$

Token-Level Objective. The token-level loss $\mathcal{L}_{TL}$ uses the masked token loss $\mathcal{L}_{MTL}$ (Devlin et al. 2018) on each text element in (w, t and p), and thus we have $\mathcal{L}^w_{MTL}$, $\mathcal{L}^t_{MTL}$ and $\mathcal{L}^p_{MTL}$. We asked the model to predict the original token for each masked token. Then, we compute cross-entropy loss for all prediction as the $\mathcal{L}_{MTL}$. Based on this design, the token-level loss is defined as: $\mathcal{L}_{TL} = \mathcal{L}^w_{MTL} + \mathcal{L}^t_{MTL} + \mathcal{L}^p_{MTL}$

3.5 Knowledge Based Context

Existing notions of semantic scope for MLM based pretraining objectives assume that the local sentence context provides all necessary meaning to a word. While visual pretraining expands that notion of scope (Bisk et al. 2020), semantics is still restricted to within a sentence. Our Knowledge Base representations capture long-distance semantic links generally only recoverable by document level understanding (e.g. linking both the artist and the location of a painting). The Knowledge embeddings provide a complementary source of information which we force the model to integrate into its contextualized lexical (and visual) representations of objects, reshaping them to place otherwise disparate concepts near each other in space. Specifically, existing VLP models rely heavily on a shared initial embedding space to make cross-modal connections. A single projection layer is used to convert visual features to 720 word embeddings in a pretrained linguistic space. This has two effects: 1. It assumes that visual information can be mapped to an existing "hub" in the BERT embedding space and 2. It allows the transformer to operate on a single embedding space throughout contextualization. Our approach makes a minor but fundamentally different and more general decision, by allowing several input embedding manifolds which the initial layers of the transformer must reconcile. In this way, the attention mechanism works over heterogenous data sources to extract relevant knowledge when appropriate and opens up a larger research question about how and where additional data can be integrated into multi-modal transformers.

3.6 Pretraining Corpus

We use the public corpus of Zhang et al. (2021) for pretraining. This corpus contains image-text pairs from several existing vision-language datasets, including COCO (Lin et al. 2014), Conceptual Captions (Sharma et al. 2018), SBU captions (Ordonez, Kulkarni, and Berg 2011), Flickr30k (Young et al. 2014), GQA (Hudson and Manning 2019), VQA (Antol et al. 2015), VG-QAs, and a subset of OpenImages. The final corpus has about 5.65 million images, 2.5 million QA pairs, 4.68 million captions, and 1.67 million pseudo-captions.

3.7 Implementation Details

KRIT uses the Transformer architecture from BERT, initialized with parameters from BERT models. We use a linear projection matrix W_I to transform the image region features to the dimensionality of the BERT model. For knowledge encoder, we initialize it using BERT-Base model and use 30 rule blocks in the reasoner module. We keep up to 20 entities for each training sample and the maximum length of each entity description is 25 tokens. Thus, after adding the special token at the begining of the text piece [CLS], the total length of the knowledge text piece is 501. The AdamW optimizer is picked for model optimization and the learning rate is set to $5e^{-5}$. KRIT is trained for at least one million steps with a batch size of 240 on 16 V100 GPUs.

4 Adapting to Vision-Language Tasks

After pretraining, we apply KRIT to several downstream vision-language understanding tasks. Each task poses different knowledge and reasoning challenges.

VQA. VQA is one of the most widely used visual question answering datasets. Following Antol et al. (2015), the model is required to answer natural language questions based on an image. Given an image and a question, the task is to select the correct answer from a multi-choice list. We use the VQA v2.0 dataset (Antol et al. 2015) for our experiments. VQA v2.0 is constructed based on the COCO image corpus and the dataset is split into a training set with 83k images and 444k questions, a validation set with 41k images and 214k questions and a test set with 81k images and 448k questions. The model picks the corresponding answer from a shared set of 3,129 answer candidates.

For VQA, the model takes one input sequence, which contains the concatenation of a question, object tags, region features, and extracted knowledge from the question and tags. Then KRIT [CLS] token is fed to a linear classifier for predicting the answer. Following Li et al. (2020b), we treat VQA as a multi-label classification problem. Each answer is assigned a soft target score based on its relevancy to the human answer responses, and we finetune the model by minimizing the cross-entropy loss against those soft target scores. At inference, we simply use a Softmax function for prediction.

OK-VQA. Outside Knowledge Visual Question Answering (OK-VQA) (Marino et al. 2019) is a new dataset that asks models to draw upon outside knowledge to answer questions. This dataset has 14,055 open-ended questions on COCO images and each question has 10 human annotated answers. We filter for questions with high-confidence answers in which 5 out of 10 annotated answers are the same (leaving 7,400 questions). As OK-VQA is designed to test how models use external knowledge, there are a substantial number of highly dissimilar answer candidates. This differentiates it from simpler multi-choice settings like VQA. Thus, we treat answer selection as an image-text retrieval task. During training, we formulate the task as a binary classification problem. Given an aligned tuple containing the image, question, answer, tags and extracted knowledge, we randomly select a different image, different knowledge or a different answer to construct a misaligned tuple. The [CLS] token is then used as the input to a binary classifier to predict whether the input is aligned or misaligned. During testing, we use the probability score to rank each answer for a given image-question pair and top-K retrieval results are used as the metric for evaluation.

5 Experiments

5.1 Results

We conduct experiments on VQA and OK-VQA and compare our model against standard VLP baselines and knowledge VLP baselines. Table 1 shows the overall performance. The left part in Table 1 presents the results on VQA dataset and OK-VQA dataset comparing with standard VLP baselines, where these models do not use any external knowledge. On VQA dataset, KRIT achieves almost similar accuracy with VinVL model (Zhang et al. 2021), which provides evidence that KRIT has competitive visual understanding ability with other standard VLP models. On OK-VQA, we compare KRIT with the Oscar and VinVL because they are most similar with ours. On the filtered testing set (2710 samples), we

Standard VLP						Knowledge VLP	
	VQA		OK-VQA				OK-VQA
Methods (%)	**Dev**	**Test**	**R@1**	**R@5**	**R@10**	**Methods**	**Acc-full**
ViLBERT (Lu et al. 2019a)	70.63	70.92	–	–	–	–	–
VisualBERT (Li et al. 2019)	70.80	71.00	–	–	–	**Q only** (Marino et al. 2019)	14.93
LXMERT (Tan and Bansal 2019)	72.42	72.54	–	–	–	**MLP** (Marino et al. 2019)	20.67
12-in-1 (Lu et al. 2019b)	73.15	–	–	–	–	**BAN** (Marino et al. 2019)	25.1
UNITER-B (Chen et al. 2019)	72.27	72.46	–	–	–	**BAN+AN** (Marino et al. 2019)	25.61
ViLT (Kim, Son, and Kim 2021)	71.26	–	–	–	–	**MUTAN** (Marino et al. 2019)	26.41
Oscar-B (Li et al. 2020b)	73.16	73.44	34.50	63.95	73.47	**MUTAN+AN** (Marino et al. 2019)	27.84
VinVL (Zhang et al. 2021)	**75.95**	**76.12**	39.82	68.26	77.49	**ConceptBERT** (Gardères et al. 2020)	33.66
KRIT (ours)	75.44	75.86	**41.47**	**69.45**	**78.56**	**KRIT (ours)**	**33.73**

Table 1: Left: Results of KRIT on VQA and OK-VQA comparing to standard VLP baselines show that our model has competitive results on VQA and outperforms existing VLP baselines on OK-VQA. Right: KRIT achieves state-of-the-art performance on OK-VQA full testing set comparing to knowledge VLP baselines.

report the Recall@1, Recall@5 and Recall@10 metrics for comparison. The OK-VQA dataset requires models to use external knowledge to answer questions. As standard VLP models such as Oscar or VinVL do not use such information during pretraining, KRIT provides a significant improvement on this dataset. We also test our model on the whole testing set (5046 samples) and use the same evaluation method described in Marino et al. (2019) to provide a fair comparison with other vision-language models using external knowledge. The right part in Table 1 presents our result comparing with the models on OK-VQA leaderboard. Our result achieves the state-of-the-art performance, which provides more evidence that KRIT is promising on commonsense VQA tasks. More details on the finetuning setting are described in Appendix A and more experiments are presented in Appendix E.

5.2 Ablation Studies

In this paper, we perform ablation experiments to evaluate the effects of reasoner patch module and the number of rule blocks in the module. We compared two different settings with KRIT. To study the effect of our novel reasoner patch module, we perform experiment with a pretraining setting without the reasoner in the module. Given a set of entities, we extract the KR-patches of these entities. Then, the patches are encoded with the text encoder and the output vector is directly concatenated with text inputs and image inputs for pretraining. To study the effect of the number of rule blocks in reasoner module, we test two settings to compare 30 rule blocks and 50 rule blocks. We trained the two different settings and our KRIT 1 million steps with 4 V100 GPUs with same batch size and finetuned them on VQA and OKVQA datasets. Results for the three models are presented in the Table 2.

The Effect of Reasoning Module. Based on the results, the model without reasoner outperforms KRIT on VQA dataset but has lower performance on OK-VQA dataset. As VQA dataset is designed to learn the alignments between image and language, model does not require any external knowledge. Thus, the model without reasoner may be easier to learn the alignments. However, OK-VQA dataset is designed for commonsense visual question-answering, our reasoner module is helpful to interpret the external knowledge and learns how to use the knowledge. Thus, KRIT has better performance on OK-VQA dataset.

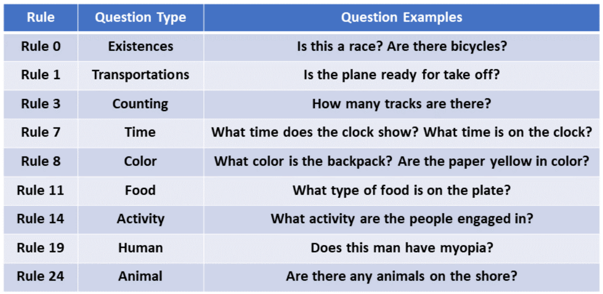
The Effect of the Number of Rule Blocks. Based on the results, the model with 30 rules in reasoner has better performance than the model with 50 rules on OK-VQA dataset. To have a better understanding on the reasoner module, we analyze the attention weights in the reasoner module with 30 rule-blocks and 50 rule-blocks. We randomly sampled 2,500 training samples from OK-VQA dataset and for each question, we only picked the top-5 selected rule with highest attention weights. After visualization, most of the samples select 5 rules in the reasoner with 30 rule-blocks. In the reasoner with 50 rule-blocks, the pattern of rule selection is less obvious than the reasoner with 30 rule-blocks. The model with 30 rule blocks seems enough to handle various tasks in our pretraining corpus. Thus, we choose 30 rule blocks in KRIT, which provides relatively clear rule selection results on samples to understand. The section B in Appendix shows the visualization results.

Model (%)	VQA	OKVQA
No Reasoner	72.06	28.41
KRIT-50	71.95	29.63
KRIT-30	71.42	**31.02**

Table 2: Results of ablation studies.

5.3 Explainability

The reasoner module provides us explainability to understand what types of questions each rule block applies. Given a training sample, the question is used as antecedent to activate rule blocks. The attention weights between the antecedent and rule matching vectors show us which rule blocks are used on this question. To better understand the reasoner module, we randomly sampled 2500 samples from VQA training set and input the question in each sample into the pretrained KRIT with 30 rules. We only pick the rule block with the highest attention weight on each question and group them together if they choose the same rule. After inspection, we find that similar types of questions tend to choose same rule blocks. Figure 5 shows a selected set of rules, the corresponding question types and some examples. From the figure, each rule block tends to apply on one or two types of questions. Each type of questions asks some specific information such as color or time. Thus, if our system does not generate correct answer on a question, we might be able to examine the rules it selects to understand the reason why it fails. We also shows the question types in selected rules on OK-VQA dataset in Appendix B. We believe this direction is a interesting direction for future research.



Rule	Question Type	Question Examples
Rule 0	Existences	Is this a race? Are there bicycles?
Rule 1	Transportations	Is the plane ready for take off?
Rule 3	Counting	How many tracks are there?
Rule 7	Time	What time does the clock show? What time is on the clock?
Rule 8	Color	What color is the backpack? Are the paper yellow in color?
Rule 11	Food	What type of food is on the plate?
Rule 14	Activity	What activity are the people engaged in?
Rule 19	Human	Does this man have myopia?
Rule 24	Animal	Are there any animals on the shore?

Figure 5: Selected rules in reasoner and question types.

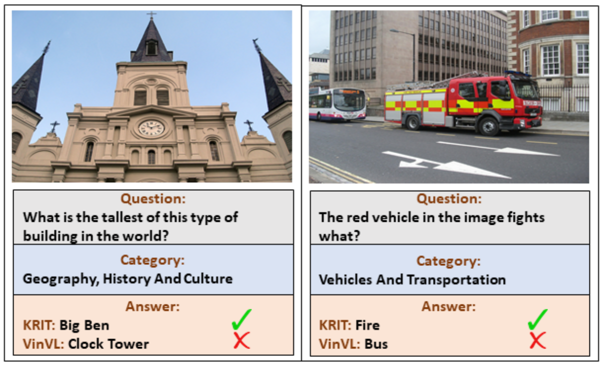
Question Type	VinVL	KRIT	Gain
Plants and Animals	32.12	35.58	+3.46
Science and Technology	27.97	28.97	+1.00
Sports and Recreation	35.95	40.39	+4.44
Geo, History, Lang, and Culture	27.06	29.79	+2.73
Brands, Companies, and Products	25.88	27.33	+1.45
Vehicles and Transportation	27.12	29.56	+1.44
Cooking and Food	33.53	31.51	-2.02
Weather and Climate	34.43	37.73	+3.30
People and Everyday	29.28	32.17	+2.89
Objects, Material and Clothing	37.62	36.21	-1.41
Other	35.06	35.27	+0.21

Table 3: Accuracy of question types in OK-VQA full testing set.

5.4 Qualitative Analysis

Category Results on OK-VQA. Here we present qualitative analyses to illustrate how external knowledge influences the output of the pretraining model. We choose OK-VQA dataset full testing set for the qualitative analysis because this dataset requires external knowledge. Based on the types of knowledge required, questions in OK-VQA are categorized into 11 categories and the accuracy results of each category are reported in Table 3. In most categories, KRIT outperforms the VinVL model. This observation illustrates that the external knowledge used in KRIT includes many different aspects. Specifically, on categories ”Plants and Animals”, ”Sports and Recreation” and ”Weather and Climate”, KRIT provides the most significant improvements.

Correct Examples from KRIT. Existing VLP models are not able to learn much additional knowledge from general vision language datasets. The knowledge embeddings used in KRIT provide extra information that cannot be reflected from image-text pairs. Figure 6 has two examples comparing the answers generated by KRIT and VinVL. From the example, we find that the VinVL model is limited to visual detection and KRIT has stronger visual understanding and reasoning ability. For example, in the first example, the generated answer from VinVL is ”Clock Tower” instead of the correct answer ”Big Ben”. Presumably the VinVL model detects clock tower but does not have the knowledge that Big Ben is the tallest clock towers in the world. Similarly, the second question ”The read vehicle in the image fights what?” requires knowledge about the usage of fire engines instead of recognizing it as bus. More correct examples are shown in Appendix C and Appendix D.



Question:
What is the tallest of this type of building in the world?
Category:
Geography, History And Culture
Answer:
KRIT: Big Ben ✓
VinVL: Clock Tower ✗

Question:
The red vehicle in the image fights what?
Category:
Vehicles And Transportation
Answer:
KRIT: Fire ✓
VinVL: Bus ✗

Figure 6: Two examples from OK-VQA dataset that KRIT generates correct answer but VinVL fails.

6 Conclusion

This paper proposes a new VLP method, KRIT, which in addition to text-image pairs, uses the text and image tags as queries to extract external knowledge from Wikipedia. KRIT takes the extracted knowledge as additional inputs. We propose two novel pretraining tasks using this external knowledge designed to enhance semantic alignment and generate representations with stronger knowledge-awareness. We pretrained KRIT on a public corpus of ∼9M image-text pairs and finetuned it on vision-language downstream tasks. Experiments on two datasets demonstrate that KRIT has better performance compared to the baselines and in particular is successful at answering questions that require external knowledge. Future work should explore the design of structure learning pretraining tasks and the use of commonsense KBs for vision-language.

References

Agarwal, O.; Ge, H.; Shakeri, S.; and Al-Rfou, R. 2021. Knowledge Graph Based Synthetic Corpus Generation for Knowledge-Enhanced Language Model Pre-training. *arXiv:2010.12688*.

Alberti, C.; Ling, J.; Collins, M.; and Reitter, D. 2019. Fusion of Detected Objects in Text for Visual Question Answering. *Proceedings of EMNLP*.

Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Zitnick, C. L.; and Parikh, D. 2015. VQA: Visual question answering. In *Proceedings of International Conference on Computer Vision (ICCV)*.

Bisk, Y.; Holtzman, A.; Thomason, J.; Andreas, J.; Bengio, Y.; Chai, J.; Lapata, M.; Lazaridou, A.; May, J.; Nisnevich, A.; Pinto, N.; and Turian, J. 2020. Experience Grounds Language. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. *Proceedings of NeurIPS*.

Chen, Y.-C.; Li, L.; Yu, L.; Kholy, A. E.; Ahmed, F.; Gan, Z.; Cheng, Y.; and Liu, J. 2019. UNITER: UNiversal Image-TExt Representation Learning. *Proceedings of ECCV*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. In *Proceedings of CVPR*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL*.

Gardères, F.; Ziaeefard, M.; Abeloos, B.; and Lecue, F. 2020. ConceptBert: Concept-Aware Representation for Visual Question Answering. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

He, B.; Jiang, X.; Xiao, J.; and Liu, Q. 2020a. KgPLM: Knowledge-guided Language Model Pre-training via Generative and Discriminative Learning. *arXiv:2012.03551*.

He, B.; Zhou, D.; Xiao, J.; Jiang, X.; Liu, Q.; Yuan, N. J.; and Xu, T. 2020b. BERT-MK: Integrating Graph Contextualized Knowledge into Pre-trained Language Models. *Proceedings of ACL*.

Hudson, D. A.; and Manning, C. D. 2019. GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering. In *CVPR*.

Kim, W.; Son, B.; and Kim, I. 2021. ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision. *arXiv:2102.03334*.

Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.-J.; Shamma, D. A.; Bernstein, M.; and Fei-Fei, L. 2016. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. In *arXiv:1602.07332*.

Kuznetsova, A.; Rom, H.; Alldrin, N.; Uijlings, J.; Krasin, I.; Pont-Tuset, J.; Kamali, S.; Popov, S.; Malloci, M.; Kolesnikov, A.; Duerig, T.; and Ferrari, V. 2020. The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. In *Proceedings of IJCV*.

Li, G.; Duan, N.; Fang, Y.; Jiang, D.; and Zhou, M. 2020a. Unicoder-VL: A Universal Encoder for Vision and Language by Cross-modal Pre-training. *Proceedings of AAAI*.

Li, L. H.; Yatskar, M.; Yin, D.; Hsieh, C.-J.; and Chang, K.-W. 2019. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.

Li, X.; Yin, X.; Li, C.; Zhang, P.; Hu, X.; Zhang, L.; Wang, L.; Hu, H.; Dong, L.; Wei, F.; Choi, Y.; and Gao, J. 2020b. Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks. *arXiv:2004.06165*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C. L.; and Dollár, P. 2014. Microsoft COCO: Common Objects in Context. *Proceedings of ECCV*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Lu, J.; Batra, D.; Parikh, D.; and Lee, S. 2019a. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. *Proceedings of NeurIPS*.

Lu, J.; Goswami, V.; Rohrbach, M.; Parikh, D.; and Lee, S. 2019b. 12-in-1: Multi-Task Vision and Language Representation Learning. In *arXiv:1912.02315*.

Marino, K.; Rastegari, M.; Farhadi, A.; and Mottaghi, R. 2019. OK-VQA: A Visual Question Answering Benchmark Requiring External Knowledge. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ordonez, V.; Kulkarni, G.; and Berg, T. L. 2011. Im2text: Describing images using 1 million captioned photographs. In *Proceedings of NeurIPS*.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. *Proceedings of NAACL*.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.

Rosset, C.; Xiong, C.; Phan, M.; Song, X.; Bennett, P.; and Tiwary, S. 2021. Knowledge-Aware Language Model Pretraining. *arXiv:2007.00655*.

Shao, S.; Li, Z.; Zhang, T.; Peng, C.; Yu, G.; Li, J.; Zhang, X.; and Sun, J. 2019. Objects365: A Large-scale, High-quality Dataset for Object Detection. In *Proceedings of ICCV*.

Sharma, P.; Ding, N.; Goodman, S.; and Soricut, R. 2018. Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning. In *Proceedings of ACL*.

Suhr, A.; Zhou, S.; Zhang, A.; Zhang, I.; Bai, H.; and Artzi, Y. 2019. A Corpus for Reasoning About Natural Language Grounded in Photographs. In *Proceedings of ACL*.

Tan, H.; and Bansal, M. 2019. LXMERT: Learning Cross-Modality Encoder Representations from Transformers. *Proceedings of EMNLP*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *NIPS*.

Vrandecic, D.; and Krotzsch, M. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*.

Wang, X.; Gao, T.; Zhu, Z.; Liu, Z.; Li, J.; and Tang, J. 2020. KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation. In *TACL*.

Xiong, W.; Du, J.; Wang, W. Y.; and Stoyanov, V. 2019. Pretrained Encyclopedia: Weakly Supervised Knowledge-Pretrained Language Model. *arXiv:1912.09637*.

Xu, S.; Li, H.; Yuan, P.; Wang, Y.; Wu, Y.; He, X.; Liu, Y.; and Zhou, B. 2021. K-PLUG: Knowledge-injected Pre-trained Language Model for Natural Language Understanding and Generation in E-Commerce. *arXiv:2104.06960*.

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.; and Le, Q. V. 2019. XLNet: Generalized autoregressive pretraining for language understanding. *NeurIPS*.

Young, P.; Lai, A.; Hodosh, M.; and Hockenmaier, J. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Yu, D.; Zhu, C.; Yang, Y.; and Zeng, M. 2020. JAKET: Joint Pre-training of Knowledge Graph and Language Understanding. *arXiv:2010.00796*.

Zhang, P.; Li, X.; Hu, X.; Yang, J.; Zhang, L.; Wang, L.; Choi, Y.; and Gao, J. 2021. VinVL: Revisiting Visual Representations in Vision-Language Models. *arXiv:2101.00529*.

Zhou, L.; Palangi, H.; Zhang, L.; Hu, H.; Corso, J. J.; and Gao, J. 2019. Unified Vision-Language Pre-Training for Image Captioning and VQA. *Proceedings of AAAI*.

Zhou, W.; Lee, D.-H.; Selvam, R. K.; Lee, S.; Lin, B. Y.; and Ren, X. 2020. Pre-training Text-to-Text Transformers for Concept-centric Common Sense. *arXiv:2011.07956*.

Appendix

A Fine-tuning Settings

VQA. During training, we randomly sample a set of 2k images from the validation set as our validation set, the rest of images in training and validation sets are used in the VQA fine-tuning. We finetune the model for 30 epochs with a learning rate of $5e^{-5}$ and a batch size of 192.

OK-VQA. After filtering the question-answer pairs with high-confidence, the training set contains 4,690 questions and the testing set contains 2710 questions. We finetune KRIT 200 epochs on the filtered dataset with batch size 128. We use the learning rate $2e^{-5}$ and linearly decreases. We finetune the baseline model with the same parameter setting.

B Rule Selection in Ablation Studies

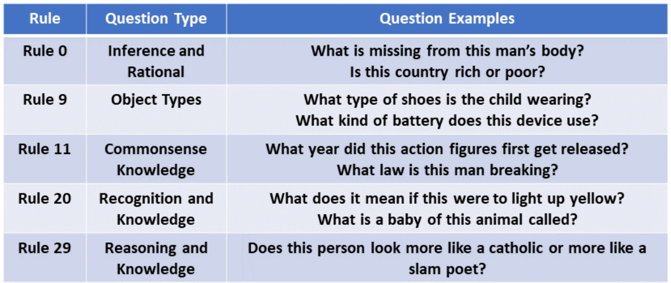
To better understand the ablation results, we randomly picked 2500 samples from OK-VQA and perform clustering on the top 5 rules each sample selected. Figure 8 visualizes the distributions. In Figure 8, x-axis represents the rule index, and the color indicates the percent of samples of selection. From the visualization, most of the samples select 5 rules in the reasoner with 30 rule-blocks. In the reasoner with 50 rule-blocks, the pattern of rule selection is less obvious than the reasoner with 30 rule-blocks. The model with 30 rule blocks seems enough to handle various tasks in our pretraining corpus. With clear rule selection, reasoner module provides explainability. We can analyze the selected rule for a given question to examine the insights of the answer. For example, if our model does not generate correct answer, we can explore the selected rules to check whether the model applies the correct rule.

Figure 7 shows our interpretation of selected rules in the samples from OK-VQA and the rough estimation of the question types. The selected five rules are the top-5 rules selected in the model with 30 rule blocks. Based on the analysis, samples in each rule requires a different type of ability to answer the question. For example, most of the samples in rule 9 are still recognition problems but models are required to generate more specific categories of an object with external knowledge. Samples in rule 0 requires more inference and rationale to generate the answer.

Based on the analysis of learned rules in our KRIT model, we integrated reasoner module and deep learning. The reasoner module provides a tool to interpret the neural models and the MLPs in reasoner boost learning and reasoning inference using the structures of rule blocks. This novel module increases the explainability and is a promising direction for future research.

C Correct Examples from KRIT

Existing VLP models are not able to learn much additional knowledge about these categories from general image captioning or visual question-answering datasets. The knowledge embeddings used in KRIT provide extra knowledge among entities that cannot be reflected from image-text pairs. Figure 9 has six examples from different categories comparing the answers generated by KRIT and VinVL. From examples, we find that the VinVL model is limited to visual detection and KRIT has stronger visual understanding and reasoning ability. These examples are from several different categories and external knowledge is important to answer them. For example, the first question at the top requires the model to know that fireplace is used for keeping warm before central air. The second question in the top row asks the model to answer the information of a famous person, which can only be extracted from external knowledge. Similarly, in the last example at bottom, VinVL thinks the camera is fisheye instead of Go Pro for sport activities. Without external knowledge, VinVL may not be able to learn the knowledge that Go Pro is more like the camera the guy use for skiing.



Rule	Question Type	Question Examples
Rule 0	Inference and Rational	What is missing from this man's body? Is this country rich or poor?
Rule 9	Object Types	What type of shoes is the child wearing? What kind of battery does this device use?
Rule 11	Commonsense Knowledge	What year did this action figures first get released? What law is this man breaking?
Rule 20	Recognition and Knowledge	What does it mean if this were to light up yellow? What is a baby of this animal called?
Rule 29	Reasoning and Knowledge	Does this person look more like a catholic or more like a slam poet?

Figure 7: Selected rule analysis on OK-VQA dataset and question types

D Incorrect Examples from KRIT

VinVL only outperforms KRIT on questions in categories "Cooking And Food" and"Objects, Material and Clothing". One potential reason for this is that these questions require reliable object detection and uses less knowledge reasoning. By adding external knowledge in pretraining, our model might generate related but not accurate answers. Figure 10 presents four examples that KRIT fails to generate correct answers. These examples reflect that rich external knowledge vectors in KRIT may increase the complexity of visual understanding. For instance, in the first image, KRIT generates the answer "Minneapolis" instead of the correct simple answer "Street", because the model might learn high correlations between Minneapolis, city and street signs from external knowledge but fail to ground the knowledge to the question. Similarly, KRIT generates the "Loafer" instead of the correct answer "Sandal" and "Red Relvet" instead of "Birthday". One interesting sample is the third image. Based on the image, people might answer "Down" naively. However, if you check the location of the sun, "North" might be a correct answer as well and requires more complicated reasoning ability. Although KRIT fails to generate correct answers on these examples, the analysis on these failed samples demonstrates that external knowledge can enhance the knowledge-awareness of existing VLP models.

E Experiments on NLVR^2

We also perform experiments on NLVR^2 (Suhr et al. 2019). The Natural Language Visual Reasoning for Real (NLVR^2) dataset (Suhr et al. 2019) asks a model to determine if a natural language statement is true or not of a pair of images.

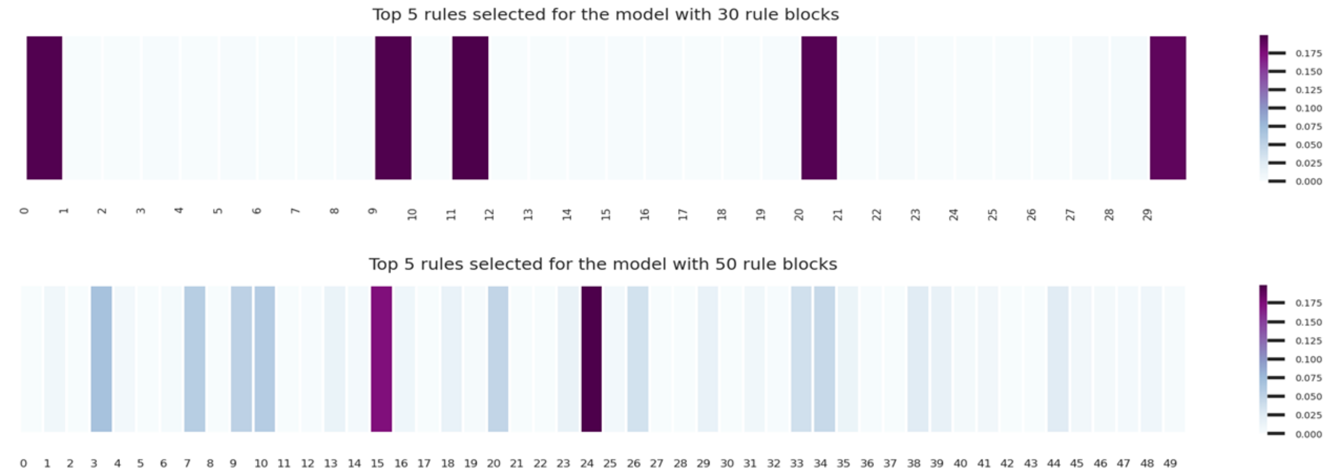


Figure 8: The visualizations of top-5 selected rules in Reasoner Module

	NLVR^2	
Model	**Dev**	**Test-P**
VisualBERT (Li et al. 2019)	67.40	67.00
LXMERT (Tan and Bansal 2019)	74.90	74.50
12-in-1 (Lu et al. 2019b)	–	78.87
UNITER-B (Chen et al. 2019)	77.14	77.87
ViLT (Kim, Son, and Kim 2021)	75.70	76.13
Oscar-B (Li et al. 2020b)	78.07	78.36
VinVL (Zhang et al. 2021)	**82.05**	**83.08**
KR-VLT (ours)	79.38	79.58

Table 4: Results of KRIT on NLVR^2

When fine-tuning, we construct two input sequences, each containing the concatenation of the text, an image and the extracted knowledge from text and the image. Then, the [CLS] tokens for the two sequences are concatenated as the joint input for a binary classifier to predict whether the statement is true. Table 4 presents the result.

Broader Impacts

Multi-modal language and vision understanding has many applications. Examples include: information retrieval and tagging and designing accessible interfaces (i.e., image descriptions and closed captioning). However, we need to carefully understand the limitations and problems presented by the data that these methods are typically trained on. Datasets are often not representative of all people and demographic groups. A dataset crawled from the Internet is more likely to capture affluent western concepts and examples. While it is very challenging to create truly representative data, we can characterize datasets to help avoid models trained on them being applied in ways that are inappropriate. The fact that these datasets are not representative of all groups is one limitation of our work. Before a system such as the one presented here is deployed more work would need to be done to understand how such a model, in the context of an application, may disadvantage or advantage certain people. Training large models often consumes a lot of power and we must not neglect the environment impact of this process. During our experiments we made every effort to use the computational resources efficiently.

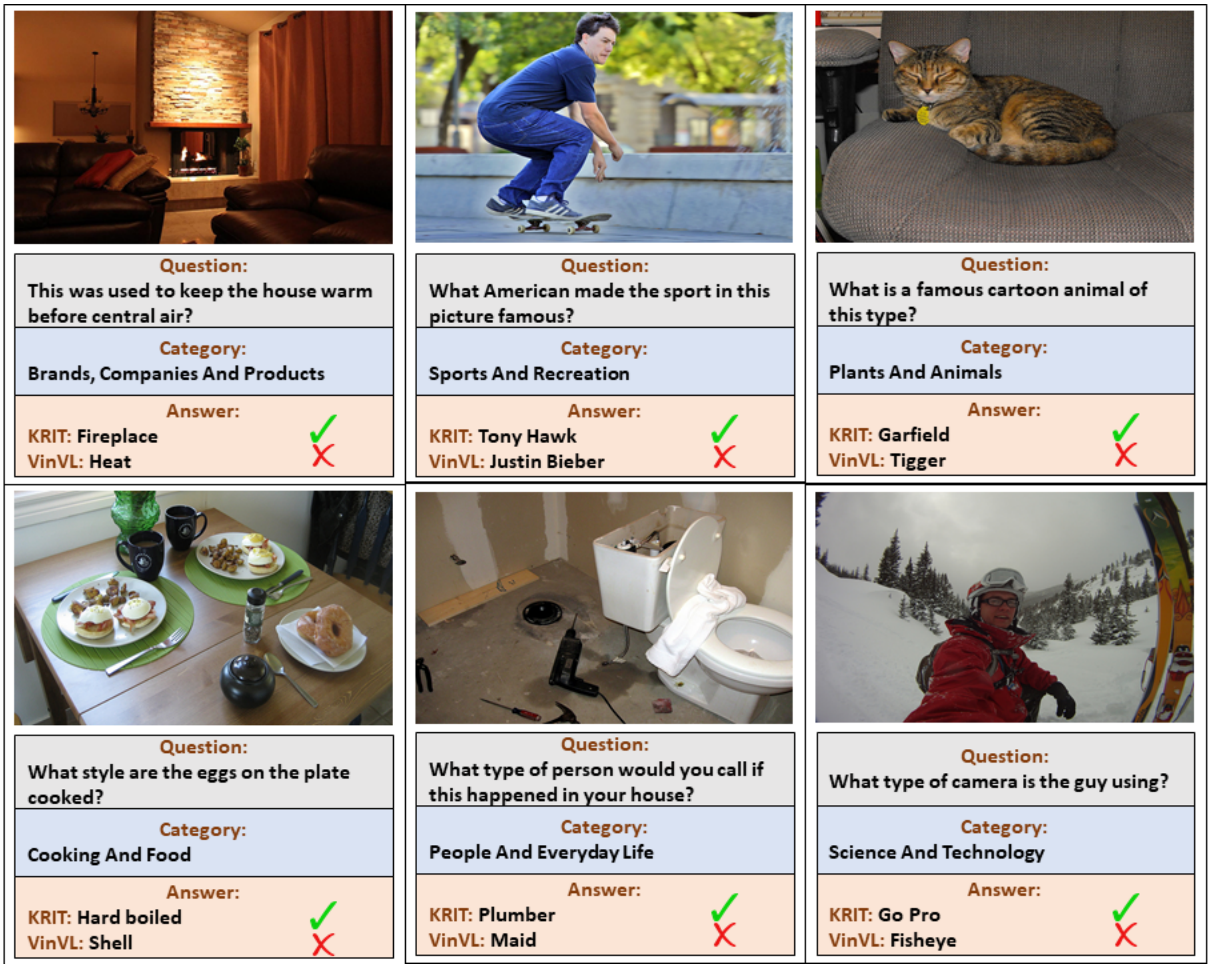


Figure 9: Eight examples from OK-VQA testing set that KRIT model generates correct answers but VinVL does not. Comparing the generated answers from KRIT and VinVL indicates that VinVL model is limited to visual detection and KRIT has stronger reasoning and understanding ability.

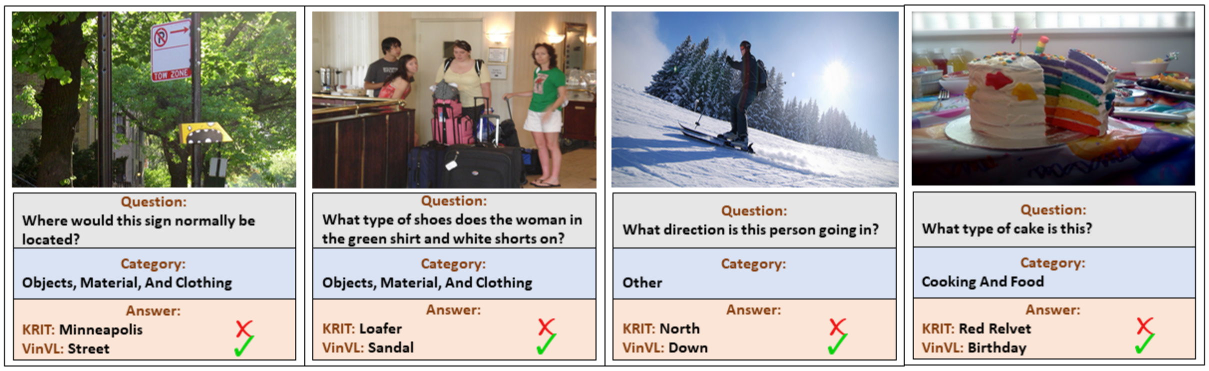


Figure 10: Four examples from OK-VQA testing set that KRIT model generates incorrect answers but VinVL gets correct answers.