Supplementary Material for Grounded Image Text Matching with Mismatched Relation Reasoning

Yu Wu[1,*] Yana Wei[1,*] Haozhe Wang[1] Yongfei Liu[1] Sibei Yang[1,2] Xuming He[1,2]
[1]ShanghaiTech University, [2]Shanghai Engineering Research Center of Intelligent Vision and Imaging
{wuyu1, weiyn1, yangsb, hexm}@shanghaitech.edu.cn jasper.whz@outlook.com liuyongfei314@gmail.com

A. RCRN

In this part, we supplement the explanation for the design of features and propagation process in Sec. 4. Below we first introduce details about our candidate representations and then further explain functions used in the propagation process.

A.1. Candidate Representation

We firstly generate an initial representation for the word tokens and visual objects based on a pre-trained vision-language model, then encode them into candidate representations. Specifically, we use the image embedder in UNITER to encode the convolutional and location features of the object regions in $\mathcal{O} = \{o_i\}_{i=1}^{N_o}$, and the text embedder to generate an embedding of the word tokens $\mathcal{W} = \{w_i\}_{i=1}^{N_w}$ and their positions in L. Those region and word features are then fed into the first k layers ($k = 6$) of Transformer in UNITER to compute their initial cross-modal representations, which are denoted as $\{\mathbf{o}_i\}_{i=1}^{N_o}$ and $\{\mathbf{w}_i\}_{i=1}^{N_w}$, respectively.

Given the word representations $\{\mathbf{w}_i\}_{i=1}^{N_w}$ and box features $\{\mathbf{o}_i\}_{i=1}^{N_o}$ generated by the transformer layers, we compute features for objects and their relations in the visual modality, and phrase features in the language modality as follows.

Visual features. For the vision modality, we encode each object by the features of its bounding box region, and each relation by the features in the union box of two objects. The visual object feature for each o_i is the corresponding box region feature $\{\mathbf{o}_i\}_{i=1}^{N_o}$. Its spatial location feature $\mathbf{l}_i^o$ is defined as $[x_i, y_i, w_i, h_i, w_i h_i]$, where (x_i, y_i), w_i and h_i are the normalized top-left coordinates, width and height of the bounding box of each object o_i respectively. The visual relation feature $\mathbf{r}_{ij}^o$ represents the direct relation between the box pair (o_i, o_j), which is computed as follows:

$$\mathbf{r}_{ij}^o = \left[\mathbf{W}_l^T \mathbf{s}_{ij}, \mathbf{W}_e^T \left[\mathbf{o}_i, \mathbf{o}_j\right]\right], \tag{S1}$$

where the relative spatial feature is represented as $\mathbf{s}_{ij} = \left[\frac{x_j - x_{c_i}}{w_i}, \frac{y_j - y_{c_i}}{h_i}, \frac{x_j + w_j - x_{c_i}}{w_i}, \frac{y_j + h_j - y_{c_i}}{h_i}, \frac{w_j h_j}{w_i h_i}\right]$. $\mathbf{W}_l$ and $\mathbf{W}_e$ are both trainable parameters, and (x_{c_i}, y_{c_i}) represents the normalized box centers for o_i.

Language features. The linguistic feature encoding is adapted from the prior work [11]. Guided by the language scene graph, both entity phrases $\mathcal{E}$ and relation phrases $\mathcal{R}$ are encoded by a LSTM and self-attention modules. In particular, the linguistic features for each entity e_i is encoded from words contained by the entity phrase itself, but the representation for each linguistic relation $r_{ij} \in \mathcal{R}$ is computed based on its corresponding subject-predicate-object (SPO) phrase, i.e. (e_i, r_{ij}, e_j). For a phrase containing N_p words, the initial word representations are $\{\mathbf{w}_i\}_{i=1}^{N_p}$. First, the initial word representations are fed into a LSTM, and we get hidden vectors of the words $\{\mathbf{h}_i\}_{i=1}^{N_p}$. Meanwhile, we represent the whole phrase using the last hidden vector, denoted as $\mathbf{h}_i^e$. Second, we input the initial word representations and the hidden vectors to the self-attention modules $\mathrm{F}_{\mathrm{attn}}^{\mathrm{app}}$, $\mathrm{F}_{\mathrm{attn}}^{\mathrm{pos}}$ and $\mathrm{F}_{\mathrm{attn}}^{\mathrm{spo}}$ respectively, and get corresponding outputs. Computation in self-attention modules is as follows:

$$\mathrm{F}_{\mathrm{attn}}^{M}\left(\{\mathbf{w}_i\}_{i=1}^{N_p}, \{\mathbf{h}_i\}_{i=1}^{N_p}\right) = \sum_{i=1}^{N_p} \frac{\exp\left(\mathbf{W}_M^T \mathbf{h}_i\right)}{\sum_{i=1}^{N_p} \exp\left(\mathbf{W}_M^T \mathbf{h}_i\right)} \mathbf{w}_i \tag{S2}$$

where $\mathbf{W}_M^T$ is the trainable parameters of the module $\mathrm{F}_{\mathrm{attn}}^{M}$, and $M \in \{\mathrm{app}, \mathrm{pos}, \mathrm{spo}\}$. For each entity phrase, we obtain an appearance feature $\mathbf{e}_i$ from $\mathrm{F}_{\mathrm{attn}}^{\mathrm{app}}$ and a location feature $\mathbf{l}_i^e$ from $\mathrm{F}_{\mathrm{attn}}^{\mathrm{pos}}$. Its linguistic feature from the LSTM is $\mathbf{h}_i^e$. For each linguistic relation $r_{ij} \in \mathcal{R}$, its feature is denoted as $\mathbf{r}_{ij}^e$, which is achieved from $\mathrm{F}_{\mathrm{attn}}^{\mathrm{spo}}$ and encodes the words from the corresponding SPO phrase.

*Both authors contributed equally to this work, which was supported by Shanghai Science and Technology Program 21010502700, Shanghai Frontiers Science Center of Human-centered Artificial Intelligence and MoE Key Lab of Intelligent Perception and Human-Machine Collaboration (ShanghaiTech University).

A.2. Details for Propagation

We use some functions to build the correspondences between multi-modal entities and relations in Sec. 4.2.1. Their detailed implementations and explanation are shown as follows.

Similarity computation. The similarity functions $\mathrm{F}_{\mathrm{sim}}$ for entities and relations are implemented differently ($\mathrm{F}_{\mathrm{sim}}^{\mathrm{ent}}$ for entities and $\mathrm{F}_{\mathrm{sim}}^{\mathrm{rel}}$ for relations, $\mathrm{F}_{\mathrm{sim}} \in \{\mathrm{F}_{\mathrm{sim}}^{\mathrm{ent}}, \mathrm{F}_{\mathrm{sim}}^{\mathrm{rel}}\}$). Inspired by SGRAF [1], we define similarity function $\mathrm{F}_{\mathrm{sim}}^{\mathrm{ent}}$ (in Eq. 1) of two vectors $\mathbf{x}$ and $\mathbf{y}$ as follows:

$$\mathrm{F}_{\mathrm{trf}}(\mathbf{x}) = \mathrm{L2Norm}\left(\mathrm{MLP}\left(\mathbf{x}\right)\right) \tag{S3}$$

$$\mathrm{F}_{\mathrm{sim}}^{\mathrm{ent}}(\mathbf{x},\mathbf{y}) = \tanh\left(\mathbf{W}_{\mathrm{eval}}^{T}\sigma\left(\frac{\mathbf{W}_{\mathrm{sim}}^{T}|\,\mathrm{F}_{\mathrm{trf}}(\mathbf{x})-\mathrm{F}_{\mathrm{trf}}(\mathbf{y})|^{2}}{\left\|\mathbf{W}_{\mathrm{sim}}^{T}|\,\mathrm{F}_{\mathrm{trf}}(\mathbf{x})-\mathrm{F}_{\mathrm{trf}}(\mathbf{y})|^{2}\right\|_{2}}\right)\right), \tag{S4}$$

where $\mathrm{F}_{\mathrm{trf}}$ represents the feature transformation, $\mathrm{L2Norm}$ is the L2 normalization, and all the MLPs are multi-layer perceptrons with ReLU activation. σ represents ReLU function, $\mathbf{W}_{\mathrm{eval}}$ projects the vector similarity to a scalar value, and $\tanh$ is for normalizing the output scalar to the range of $[-1, 1]$. The similarities b_{ij}^{app} and b_{ij}^{pos} for the appearence and spatial location space respectively are obtained from $\mathrm{F}_{\mathrm{sim}}^{\mathrm{ent}}$. The vector similarity has been shown to be more expressive than cosine similarity function.

The similarity function $\mathrm{F}_{\mathrm{sim}}^{\mathrm{rel}}$ (in Eq. 4) focusing on relations is implemented with the conventional cosine similarity in order to reduce the computational complexity:

$$[\mathbf{A}_{ij}]_{kl} = \sigma\left(\left\langle \mathrm{F}_{\mathrm{trf}}\left(\mathbf{r}_{ij}^{e}\right), \mathrm{F}_{\mathrm{trf}}(\mathbf{r}_{kl}^{o})\right\rangle\right). \tag{S5}$$

Normalization. Another function for normalization $\mathrm{F}_{\mathrm{norm}}$ divides all the values with the maximum absolute value if the maximum absolute value is larger than 1. If the final range is $[-1, 1]$ (in local correspondences), it will then linearly maps the range to $[0, 1]$.

Independent parameters. Bottom-up and top-down propagations have different β^{rel} with independent parameters $\mathbf{W}_{\mathrm{rel}}$ in Eq. 5. The reasoning also learns different parameters $\mathbf{W}_{\mathrm{rel}}$ for grounding and matching, but shares the same local correspondences, i.e. the same parameters in $\mathrm{F}_{\mathrm{sim}}$. Empirical results show this necessity because they focus on different aspects of the reasoning results. To be specific, the matching task emphasizes the global alignment representation, but the grounding task focuses more on the local alignment variance.

Belief selection in the propagation for matching. Compared to the grounding, the matching task typically requires reasoning on the global representations of correspondences, rather than local variances on the belief. We apply a pruning strategy that only chooses top K visual objects with the highest similarity scores to be the assignment space for each linguistic entity, getting $\mathbf{b}_i \in \mathbb{R}^K$ for all $i \in [1, N_e]$ in the matching propagation. Additionally, the belief vectors are sorted for computing the confidence score in Sec. 4.2.1. The pruning selects the most representative similarities in each correspondence feature, and sorting may make the confidence computing more sensitive to the sharpness of the belief vector. The relation correspondences are also pruned according to the node pruning results.

B. Box Regression

We adopt detected object proposals for the generalized grounding, which could be a performance bottleneck due to their inaccurate localization. Consequently, we append an additional regression head to RCRN, UNITER and FGVE in order to refine the proposal locations.

We fuse visual and language features and apply an MLP to compute offsets for refining the proposal coordinates:

$$\delta = \mathrm{MLP}_{\mathrm{regress}}\left(\left[\mathbf{f}_i^{O}, \mathbf{f}_i^{\mathrm{pos}}, \mathbf{f}_{\mathrm{global}}^{L}\right]\right) \tag{S6}$$

where $\mathbf{f}_i^{O}$, $\mathbf{f}_i^{\mathrm{pos}}$ and $\mathbf{f}_{\mathrm{global}}^{L}$ represents the predicted visual region representation, location feature of the predicted region and the global representation of language expression respectively. $\delta \in \mathbb{R}^4$ corresponds to the offsets for refining the region proposal.

For instance, in RCRN, $\mathbf{f}_i^{O}$ is the feature $\mathbf{o}_i$ of the predicted box, $\mathbf{f}_i^{\mathrm{pos}}$ is the 5-dimensional location feature $\mathbf{l}_i^{O}$ for the predicted region, and $\mathbf{f}_{\mathrm{global}}^{L}$ is the the mean of all SPO representations.

During training, we use a smoothed L_1 regression loss to penalize the difference between δ and the ground-truth offsets δ^*.

$$\mathcal{L}_{\mathrm{reg}} = \mathrm{smooth}_{L_1}\left(\delta^*, \delta\right) \tag{S7}$$

where δ^* is the difference between the coordinates of a ground-truth bounding box and that of a predicted box.

C. Dataset construction

C.1. Data Generation

We construct our dataset GITM-MR by a generation program to create mismatch expressions from the corresponding original expressions by partial replacement. First, we identify all the relation phrases in the expressions of the Ref-Reasoning dataset by using an off-the-shelf parser [8]. After that, we manually select a subset of 27 commonly-occurred relations, and assign some relations acceptable to similar contexts but with different semantics, for each relation in the subset, as their replacement candidates. For example, we assign "carry" as an candidate for "wear", and

Table S1. The full substitution candidate list of the selected relations.

Relation	Substitutions
to the left of	to the right of, in front of, behind
to the right of	to the left of, in front of, behind
wearing	holding, carrying, looking at
in front of	behind, to the right of, to the left of
behind	in front of, to the right of, to the left of
holding	looking at, behind
on top of	on the side of, near, next to
above	near, next to, behind, in front of
below	near, next to, in front of, behind
sitting on	near, next to, behind, in front of, to the left of, to the right of
next to	in front of, behind
carrying	looking at, behind
inside	near, next to, behind, in front of, to the left of, to the right of
under	above, on top of, next to, in front of, behind
standing on	walking on
walking on	standing on
eating	holding, looking at, behind
standing in	walking in
playing with	standing by, behind
walking in	standing in
riding	behind, in front of, to the left of, to the right of
riding on	behind, in front of, to the left of, to the right of
playing	standing by, behind
walking down	standing on
throwing	holding, behind
standing behind	standing in front of
standing in front of	standing behind

"to the left of" as an candidate for "to the right of". Utilizing human annotations as such significantly reduces the linguistic bias and false negative cases on the generated expressions. Finally, we replace one relation in each expression by a candidate to construct a mismatch expression. The program keeps the relation phrase set and the replaced relation for each expression as the labels for MRR evaluation.

The following aspects are considered additionally to control the overall quality of the generated mismatched expressions.

Mismatched relation diversity. In the real-world scenario, the mismatched expressions should be various as the matched expressions, rather than showing certain patterns which are easy to identify. To keep the diversity, we assign multiple substitutions for each relation. The full replacement candidate list is shown in Tab. S1 for the reference. We randomly select the substitution when replacing a relation.

Linguistic bias. We reject the substitutions that violate the following rules to further control the linguistic bias:

1. The generated mismatched expression should have close perplexity in the pre-trained language model BERT [4] with the original matched expression.

2. After the substitution, the relation phrase occurrence distribution in matched and mismatched expressions should be close.

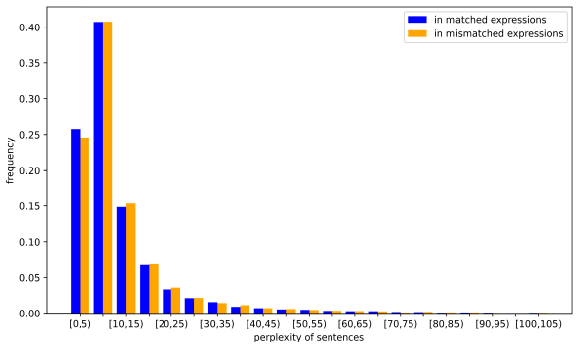


Figure S1. Sentence perplexity distributions on the validation set. Only the perplexities ranging from 0 to 105 are shown for the ease of reading.

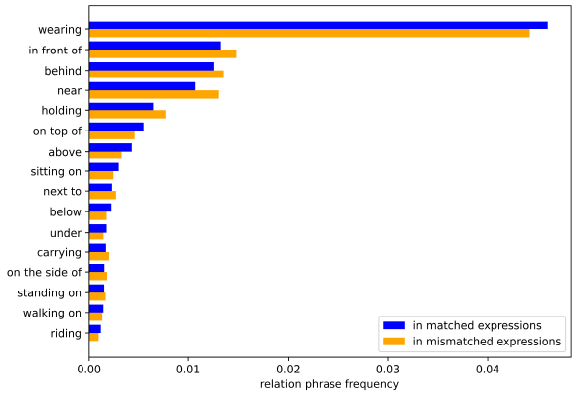


Figure S2. Relation frequencies on the validation set. Only the body part of the relations are shown for the ease of reading.

We count the distribution of sentence perplexity in the range of 0 to 105 in Fig. S1, and the frequencies from the main body of the relation phrases belonging to the substitution list in Fig. S2. The statistics of matched and mismatched expressions in the validation set are shown by the bars with different colors. The results demonstrate the high similarity between matched and mismatched expressions.

Furthermore, we generate several candidate datasets, and then train language-only transformers to do binary classification on each dataset. The lower accuracy on the task implies the lower linguistic bias on the dataset. The candidate dataset with the lowest classification accuracy is chosen as the final GITM-MR dataset.

False mismatch. There still exist image-text pairs where a relational phrase in an expression is replaced, but the entire expression still perfectly describes an object in the image. We call these false mismatched cases. To avoid those unwanted cases from hurting the quality of testing, we hired annotators from Stardust* to label the mismatch cases in the

*https://stardust.ai/

test set, indicating whether the constructed mismatched sentence description in each case can match an object in the image. In the end, we only keep the cases that the annotators identified as mismatch in the test set. The final test set contains only the reserved mismatch cases and their corresponding matched cases. We provide more details of the manual curation in Sec. C.2

C.2. More Details on Manual Data Curation

User interface. We provide instructions for labeling given to workers in Fig. S3. The user interface of data curation on the StarDust platform is shown in Fig. S4. The langauge expression is shown in the top part and the image is shown in the left. In the right part, the tester is asked with two question. In the first step, the worker needs to examine whether the sentence is reasonable. If the answer at first step is positive, then the worker should identify if the sentence describe an object in the image at the second step.

Since the dataset involves relatively complex image scenes and sentence structures, we annotate one box in each image to help workers make a decision at the second step. The annotated box is the ground truth box in the original grounding dataset, i.e. Ref-Reasoning [11]. Because the sentence in the mismatched case is obtained by replacing one relation in the sentence from the original dataset, it is possible that the sentence constructed with substitution still points to the object corresponding to the original sentence. We also color the replacement in the sentence to help workers focus more on the important parts of sentences. To avoid the workers from biased labeling, we didn't tell them the construction of mismatched cases.

Job setting. Each StarDust annotator maintains a job approval rate based on their performance on previous jobs. We invite only experienced annotators whose job approval rate is equal to or greater than 98%. Also, we hire three independent annotators for each job and aggregate their annotations for final decision.

Quality control. In the labeling period, Stardust submitted the data labeled everyday to their platform. We checked the data after their each submission. For the manual labels that we disagreed with, we gave the reason for judgment and rejected the samples for workers to re-label. Each day, annotators also maintained a document that stores data that they considered ambiguous, and asked questions about the labeling process. We answered their questions every day in the document, and unified the labeling standards for some details.

Result. Finally, we only keep the image-text pairs where at least two testers can't find a relative object in the image according to the text. The reserved mismatch subset contains 2046 images and 5616 expressions.

Table S2. Comparsion with other datasets from the aspects of sentence length, number of entities in each sentence and perplexity gap between sentences from positive and negative cases.

Dataset	Sentence Length (mean/median)	Average Number of Entities	Perplexity (pos/neg)
RefCOCO	3.50/3	1.21	-
RefCOCO+	3.53/3	1.10	-
RefCOCOg	8.46/8	2.29	-
SVO-Probes (verb)	6.21/6	2.01	-
VALSE (relations)	10.36/10	2.73	25.03/34.45
GITM-MR	22.22/23	3.55	12.85/13.20

C.3. Comparison with Other Datasets

In this section, we compare our proposed benchmark GITM-MR with existing datasets of grounding or matching task, showing the advantage of our GITM-MR on sentence complexity and sentence rationality. In Tab. S2, RefCOCO [6], RefCOCO+ [6] and RefCOCOg [5] are three datasets for the visual grounding task. SVO-Probe [2] is a dataset designed to test pre-trained VL models' understanding of verbs, subjects and objects. We make statistics on its subset involving negative verbs. VALSE [7] is a benchmark aimed at gauging the sensitivity of pre-trained VL models to *foiled* instances, the statistics in this table refers to one VALSE subset involving spatial relations.

Sentence complexity. We compare the sentence complexity from two aspects: sentence length and number of entities in each sentence. Firstly, Tab. S2 and Fig. S5 shows that the average sentence length of GITM-MR is much longer than other datasets. Secondly, in Tab. S2, we compared the number of parsed entities in each dataset, and GITM has the highest average number of entities in each sentence. More entities in one sentence means more relations exist, which makes the reasoning process involved in grounding and matching tasks more complicated or leads to longer reasoning paths.

Sentence rationality. In Tab. S2, we measure the rationality of the constructed negative sentences by comparing the perplexity difference between the positive and negative cases in each dataset, following VALSE. Compared with VALSE (split on relation substitution), the perplexity gap between the positive and negative cases in our GITM-MR is much smaller, which means the constructed negative examples have a smaller plausibility bias, as described in [7].

D. Experiments

Overview

Help us identify whether the given linguistic expression matches the image.

Steps

1. Examine whether the given image-text pair is reasonable.
2. If it is, then identify if the entity described by the text can be found in the image (match or not).

Rules & Tips

Rules:

- Unreasonable cases:
  - The sentence **cannot** be understood and is just like a string of unrelated words.
  - The sentence **does not** conform to common sense, e.g., a man eating a chair.
- Mismatched:
  - Interpreting the sentence in any way, you **cannot** find the corresponding entity in the picture.
- Match:
  - If the sentence can be used to describe one object in the image, the image and sentence are matched.
  - If the sentence can correspond to multiple objects in the image, this pair is also matched.

Tips:

1. The sentence **does not** have to use perfect grammar。

Figure S3. A screen-shot of instructions provided to the annotators to filter the false mismatched cases.

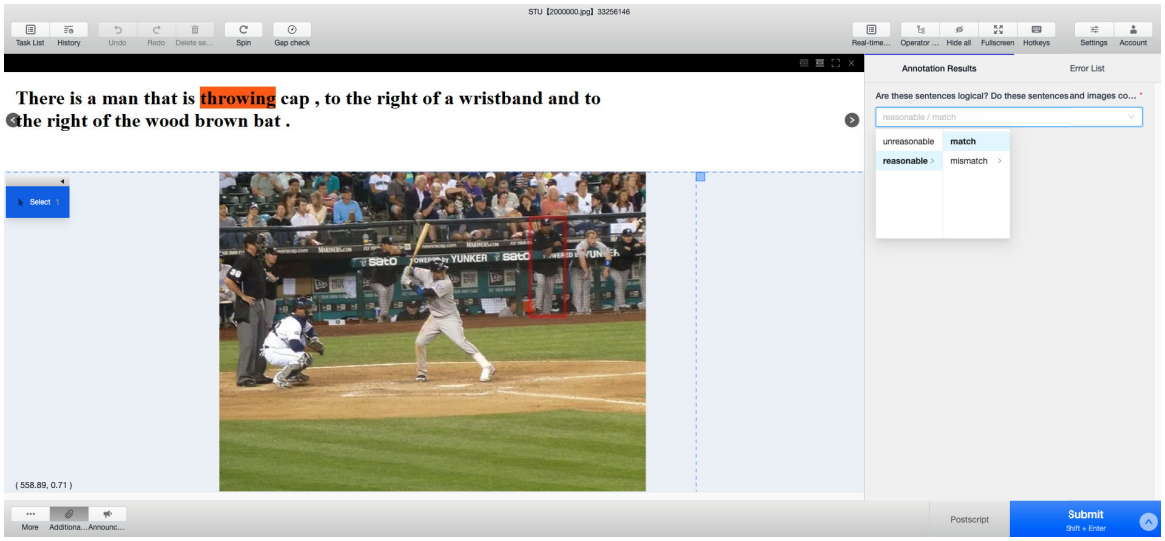


Figure S4. A screen-shot of user interface for the manual data curation.

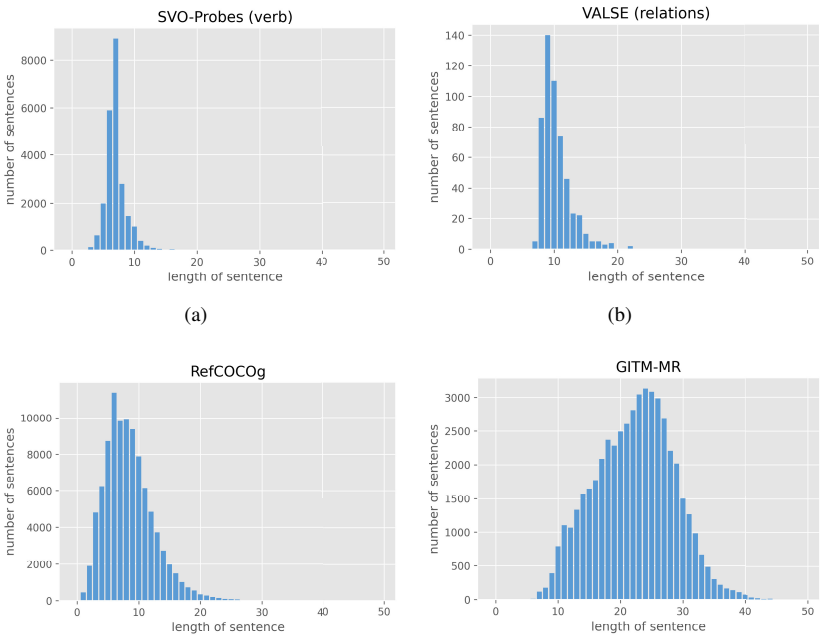


Figure S5. The distribution of sentence length in different datasets. Overall, our dataset has longer sentences than other datasets. In terms of details, the length of most sentences in other datasets is within 16, which corresponds to the standard for defining our OOD setting in Sec. D.1.3.

D.1. Setup

D.1.1 Training Details on RCRN

To prevent numerical instability, we apply several techniques when implementing and training RCRN. (1) We implement the multiplication in Eq. 4 and 6 using log space transformation. We firstly substitute production with summation in log space, and then transform the value back by the exponential function. (2) We initialize the learning by training only on grounding task from scratch to learn transformation layers in similarity functions first. Empirical results show that these learned layers are helpful for later multi-task joint learning. (3) We cut off the gradients on all the input of β_i's for stable training.

D.1.2 VL Pre-trained Models for GITM-MR

As referred in Sec. 5.1, we modify UNITER, TCL, FIBER and FGVE to complete the three substasks of GITM-MR. In terms of the modification for them, we use the pretrained components from original models, and add two subtask heads for grounding and matching.

UNITER For the matching task, we extract the representation of [CLS] token as the joint representation of the input image-text pair, and then feed it into an FC layer and a sigmoid function to predict a score between 0 and 1. We apply the binary cross-entropy loss on the matching scores for optimization. In order to complete the grounding task, we add a MLP layer on top of the region token outputs from transformer layers to compute the alignment score between the expression and each visual region. Then we apply cross-entropy loss on the normalized alignment scores.

UNITER+MIL To allow the pre-trained models to simultaneously complete the MRR task, we conduct a multi-instance learning strategy following [9], creating the baseline UNITER+MIL. Concretely, for the matching task, we firstly take the textual token outputs from Transformer, and feed it into the FC layer (matching head). Thus we get a local matching score for each textual token. Then we get minimum local matching score instead of the score for [CLS] token as the global prediction to compute the matching loss. For the MRR task, we compute a matching score for each relation in the language expression assisted by the language parser. The score for a relation is the average of local matching scores for words in this relation. Finally, the relation earning the lowest score is the predicted mismatched relation.

TCL TCL is modified for the GITM-MR task similarly. Different from UNITER which using image token represen-

tations for grounding, TCL is a one-stage pretrained model without output representation for each proposal in the image. So TCL can only complete the matching task in GITM-MR.

FIBER We use the fine-grained pre-trained FIBER to complete GITM-MR. For the grounding task, the OD-head from FIBER is preserved. For the matching task, following the coarse-grained pre-trained FIBER, we concatenate the two global representations from both the visual and language backbone, and feed them into a MLP layer to get the matching score. We also tried to use the coarse-grained pre-trained FIBER to complete the matching task, but it struggles to get an accuracy of just over 50%.

FGVE To modify FGVE for the MRR task, we use Fast Align Algorithm Aligner[†] tool to align the knowledge elements (KE) with tokens in the sentence, and then maximally pool the mismatch score predicted on each KE to aligned tokens. Once the score on each token is obtained, the mismatch score on each relation phrases are computed as the maximum of their token scores, and the one with the highest mismatch score will be MRR prediction. The multi-instance learning losses are preserved and modified for our binary classification matching task.

D.1.3 Data Partition Criterion for OOD setting

As mentioned in Sec. 3, we construct training sets including only simple sentences. The models are trained on the simple training set, and evaluated on test set containing both simple and complex sentences. Next, we describe the criterion of selecting simple image-sentence pairs for training from two aspects: number of entities and sentence length.

Number of entities. The complexity of a sentence is proportional to the number of entities contained in it. Smaller number of entities represents lower sentence complexity. However, the number of entities obtained from the off-the-shelf parser is not accurate. We only get a part of ground-truth numbers of entities in val set. Due to the fact that the length of a sentence is also positively correlated with the number of entities, we use the ground-truth numbers of entities in val set to infer appropriate sentence lengths as dividing lines between the simple and the complex.

Following this idea, we first draw a histogram (Fig. 2 in the main paper) to observe the distribution of sentence length and number of entities in the validation set. Second, we choose appropriate sentence lengths for division according to the histogram. Finally, the selected sentence lengths are applied to select training set containing simple sentences.

The reason for choosing sentence lengths 11 and 16 as dividing lines is shown as follows. In Fig. 2 in the main paper, when sentence length is shorter than 11, most sentences contain only two entities, and a few sentences contain three entities. In sentences which have length less than 16, two or three entities are mainly included, and there still exists small number of sentences with four entities.

Sentence length. As shown in Fig. S5, in the common datasets of matching and grounding tasks, most sentences are within 16 in length. This shows that sentences shorter than 16 are easy to collect for training, but longer sentences are more difficult to obtain. So it is reasonable to choose Train-Len16 as an OOD scenario. Moreover, Train-Len11 is a more challenging evaluation setting, because the simpler (shorter) the sentences in the training set, the higher the generalization ability of the model is required.

D.1.4 Discussion on Length Generalization

The length generalization of models serves as a crucial test of their ability to understand relations. In the GITM-MR task, length generalization can be viewed as a proxy for relation number generalization. As longer sentences typically contain more relations, they require more complex reasoning paths to identify the referents, much like the concept of reasoning hops in VQA tasks, where the number of reasoning steps required to answer a question reflects the level of reasoning needed. Thus, the ability of models to generalize to longer sentences with more relations demonstrates their capability to handle complex relation understanding tasks and to perform reasoning across multiple entities and relations.

D.2. Results

D.2.1 Oracle Results

As mentioned in the dicussion in Sec. 5.2, we investigate models' performance on grounding and mismatch relation reasoning by evaluting for results of these two tasks independently, without considering the matching prediction. The results in Tab. S3 shows that our RCRN achieves high accuracies on both subtasks in the different training setups. Particularly, RCRN outperforms other models on the MRR task, and the gap is especially notable in the OOD test. This demonstrates our models' strong ability to complete the grounding and MRR task, which requires models to learn fine-level cross-modal alignments.

[†]https://amrlib.readthedocs.io/en/latest/faa_aligner/

Table S3. Oracle experiment results on the GITM-MR dataset. In this experiment, the computation of grounding and MRR accuracies are independent from the matching results.

Training Set	Method	Full Test		In-Distribution		Out-of-Distribution	
		Grounding%	MRR%	Grounding%	MRR%	Grounding%	MRR%
Train-Len16	UNITER	40.21	-	52.97	-	37.50	-
	UNITER+MIL	41.62	60.77	54.55	89.96	38.88	54.53
	FGVE+MAX	-	49.57	-	79.24	-	43.22
	FIBER	28.51	-	**57.24**	-	22.41	-
	RCRN(Ours)	**42.06**	**72.08**	54.43	**90.74**	**39.43**	**68.09**
Train-Len11	UNITER	35.04	-	45.81	-	34.59	-
	UNITER+MIL	32.27	52.67	41.87	96.60	31.87	50.82
	FGVE+MAX	-	49.71	-	97.57	-	47.68
	FIBER	25.72	-	**51.23**	-	24.66	-
	RCRN(Ours)	**38.16**	**65.55**	45.81	**98.54**	**37.85**	**64.16**

Table S4. Additional results trained on a subset of 5% in-distribution training data. **Add** means "additional training data volume to achieve ours performance" metric. The '-' denotes that the model is not applicable in corresponding subtask or metric. The results show that the proposed method outperform prior methods in limited data setting.

Method	Matching%	Grounding%	MRR%	Add
TCL	52.14	-	-	+>250K
UNITER	56.05	21.79	-	+70K
UNITER+MIL	56.14	20.37	**38.83**	+50K
FIBER	54.01	23.95	-	+50K
RCRN(ours)	**60.97**	**29.96**	38.07	-

D.2.2 Additional Results on Limited Training Data

To further demonstrate the data efficiency of our RCRN in the limited data setting, we evaluate our result on an additional training setting, along with a new metric. We uniformly sample 5% training data from the original training set, and evaluate on the original test set. We also adopt a new evaluation metric, which uses the performance of our model on three subtasks with those training data as a reference point, and compare the amount of additional training data each baseline requires to achieve equal or higher performance on all the subtasks they can solve. The result shown in Tab. S4 verifies the high performance of our method under limited training data setting again, and this superiority is independent of whether the maximum expression length is limited. Moreover, most baselines double or triple the amount of training data to achieve our model's performance, requiring more than 50K additional training samples. Notably, even with additional 250K samples, TCL still struggles to reach our performance level.

In order to assess the generalization capability of our approach, we conduct zero-shot testing on a subset of ARO [12] dataset named Visual Genome Relation, which focuses on evaluating the relation understanding capabilities of models. The zero-shot match accuracy results are as Tab. S5, which demonstrates the generalization capability of our proposed method.

Table S5. The zero-shot test result on Visual Genome Relation subset in ARO dataset.

Method	TCL	UNITER	UNIETR+MIL	FIBER	RCRN(ours)
Accuracy%	50.57	57.41	57.89	54.21	**62.88**

Table S6. Comparison with state-of-the-art interpretable models for REG and image-text matching tasks. The '-' denotes that the model is not applicable in corresponding subtask. Models are evaluated on ground-truth objects by following the setting in SGMN.

Method	GITM-MR			Ref-Reasoning	#Param
	Mat	Grd	MRR	Grd	
DGA[10]	-	36.49	-	45.87	50.99M
SGMN[11]	-	42.79	-	51.39	38.71M
SGR[1]	56.30	-	-	-	37.38M
Ours(RCRN-T)	**67.12**	**55.64**	**71.72**	**58.92**	**35.52M**

D.2.3 Comparsion with Interpretable Models without VL Pre-training

The proposed RCRN without the transformer layers (denoted as RCRN-T) can independently handle these three subtasks as a single unified model with interpretability. In Tab. S6, we compare our RCRN-T with several existing state-of-the-art interpretable models without vision-language pre-training for REG or image-text matching tasks on both GITM-MR and Ref-Reasoning [11] dataset. The DGA and SGMN are grounding methods, and SGR is a ITM model. All these state-of-art models don't embrace appropriate design to accomplish the MRR task. For a fair comparison, all the models use the same visual object features and the same setting in word embedding.

Table S7. Ablation study for VLP layers on GITM-MR validation set.

VLP Layers	Mat	Grd	MRR
3	62.28	26.88	52.97
5	62.25	28.05	50.40
6	**62.52**	26.71	**53.69**
7	62.18	**29.08**	47.83
9	62.07	26.43	52.99
12	62.12	27.13	50.71

Table S8. MRR comparisons with simple baselines.

Method	Formula	Acc
Baseline 1	$\hat{r} = \arg\min_{r_{ij} \in \mathcal{R}}(\{(p_i^{\text{bp}} + p_j^{\text{bp}}) + (p_j^{\text{td}} + p_i^{\text{td}})\})$	51.77
Baseline 2	$\hat{r} = \arg\min_{r_{ij} \in \mathcal{R}}(\min\{(p_i^{\text{bp}} - p_j^{\text{bp}}), (p_j^{\text{td}} - p_i^{\text{td}})\})$	46.81
Baseline 3	$\hat{r} = \arg\min_{r_{ij} \in \mathcal{R}}(\min\{(p_i^{\text{bp}} + p_j^{\text{bp}}), (p_j^{\text{td}} + p_i^{\text{td}})\})$	38.01
Ours	$\hat{r} = \arg\min_{r_{ij} \in \mathcal{R}}(\{(p_i^{\text{bp}} - p_j^{\text{bp}}) + (p_j^{\text{td}} - p_i^{\text{td}})\})$	**55.72**

Table S9. Results with proposals from different detectors.

	Box	Full Test			In-Distribution			Out-of-Distribution		
Proposal	Recall	Mat	Grd	MRR	Mat	Grd	MRR	Mat	Grd	MRR
Ground Truth	100	64.32	52.40	68.21	70.66	67.00	98.54	64.06	51.79	66.93
VinVL (Ours)	74.86	63.41	38.16	65.55	68.7	45.81	98.54	63.19	37.85	64.16
Faster R-CNN	61.83	61.26	36.73	65.24	72.62	47.78	99.51	60.78	36.27	63.79

Tab. S6 shows that the proposed RCRN-T outperforms all existing state-of-the-art interpretable models under the same backbone setting and earns minimum number of parameters . SGR achieves a relatively low accuracy 56.30% on matching task, which shows that slight difference on the relations in sentences is hard to distinguish without special fine-level relation semantic learning design. DGA only learns a low-order language guided contextual representation for objects, and relatively fixed context modeling design limits SGMN's learning process. Compared with Ref-Reasoning, GITM-MR is more challenging on grounding because it has at least 2 entities in each expression, and it includes OOD scenarios in testing.

D.2.4 Ablation on VLP Layers

Tab. S7 shows the ablation study on the number of VLP layers used for generating the candidate representation. The results on the full test set show that our RCRN gets the best performance with 6 pre-trained VLP layers. Features from shallower layers may not have learned accurate enough cross-modal correspondence. Features from the next few layers may have overfitted to the finetuning tasks, and can't be refined through the propagation process easily.

D.2.5 Ablation on MRR Methods

As shown in Tab. S8, we compare our method with three simple baselines for mismatched relation prediction, under the same training setting. For each relation, **Baseline 1** calculates the MRR score as the summation of confidence scores from two related nodes. **Baseline 2** takes the minimum of the confidence gap as the MRR score. **Baseline 3** computes the MRR score by selecting the minimum of the sum of the confidence scores. The ablation results demonstrate the superiority of our MRR method.

D.2.6 Ablation on Object Detectors

Object detector is an important external module of our RCRN, stated as Sec. 4.1.1. To investigate the impact of the object detector, we conduct experiments to compare the results of RCRN trained with proposals from different detectors, while using the same feature map from VinVL, as reported in Tab. S9 (with the same setting as Tab. S3). We consider the ground truth boxes and proposals from two most popular detectors used in VL tasks, namely Faster R-CNN and VinVL. The results show that, for the matching and MRR task, though the quality of detected proposals affects the model's performance, the performance gaps are relatively small. This observation suggests that our model is robust to the variations introduced by external modules.

D.3. Visualization and Interpretability

D.3.1 More Visualizations

Fig. S6 and Fig. S7 show some additional visualizations on RCRN. We highlight some inspiring evidences here. In Fig. S6, the tiny relation mismatch causes notably drop on the matching probabilities of the attached nodes. In Fig. S7, the ambiguity of "fan" is significantly reduced by the message propagation, mainly relied on the message from the child branch "to the left of cord".

D.3.2 Generation of GT Correspondence

We propose an algorithm mentioned above to find the ground-truth scene graph of the expression from the ground-truth image scene graph. The basic idea is that as the expressions from Ref-Reasoning are generated from some certain forms of the subgraphs of the image scene graphs in GQA [3], the original subgraphs must exist in the complete scene graphs. We can achieve the golden subgraph by searching for the subgraph that is most close to the parsed language graph. The main work flow of the algorithm is shown as Alg. S1.

Here we still suppose the language scene graph has a tree structure. The function searchSuccessors expands the current subgraph g from their leaves, and only accepts the successors that matches the corresponding part in the parsed graph $(\mathcal{V}, \mathcal{E})$, which is easy to verify by traversal. Note that the predicted subgraph $\hat{\mathcal{G}}$ may not exist, or not be unique, since the parsed language graph may have error. We only

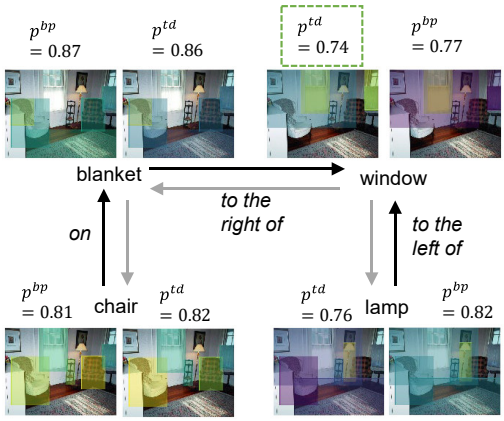
Algorithm S1 The graph matching algorithm.

Input: The ground-truth coordinate of the referent b; The parsed entity phrases $\mathcal{V}$; The parsed relation phrases $\mathcal{E}$; The ground-truth scene graph objects $\mathcal{V}^*$ with phrase annotations; The ground-truth scene graph relations $\mathcal{E}^*$ with phrase annotations;

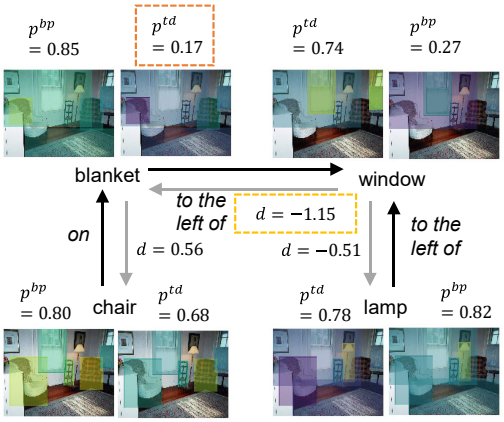
Output: The predicted language scene graph $\hat{\mathcal{G}}$;

stack $\leftarrow$ [$(\{v\}, \{\})$ for v in $\mathcal{V}^*$ if $\mathrm{coordinateOf}(v) == b$]
while len(stack) > 0 **do**
    $g \leftarrow$ stack.pop()
    if $g == (\mathcal{V}, \mathcal{E})$ **then**
        $\hat{\mathcal{G}} \leftarrow g$
        return $\hat{\mathcal{G}}$
    else
        stack.extend($\mathrm{searchSuccessors}(g, \mathcal{V}, \mathcal{E}, \mathcal{V}^*, \mathcal{E}^*)$)
    end if
end while

use the cases with a unique result for the subsequent procedures to find golden correspondences. Finally, we obtain the correspondences from the parsed phrases to visual components by those subgraphs, using the ground-truth correspondences annotated in the complete image scene graph.

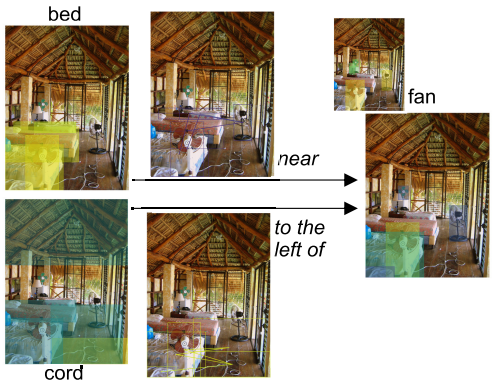


(a) A window to the right of the blanket which is on the chair and the window to the left of the lamp.

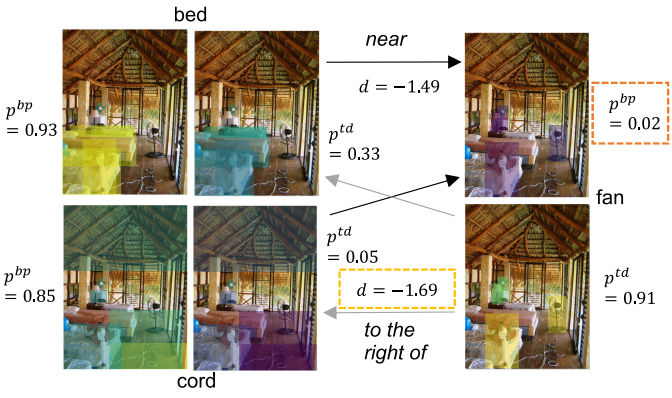


(b) A window to the left of the blanket which is on the chair and the window to the left of the lamp.

Figure S6. Two additional visualization showcases of RCRN. (a) is a matched case and (b) is the corresponding mismatched case. Both of them show the matching propagation results.



(a) The fan which is near bed and is to the left of cord.



(b) The fan which is near bed and is to the right of cord.

Figure S7. Another two cases of RCRN. (a) is the grounding process in a matched case and (b) is the matching propagation result on the corresponding mismatched case.

References

[1] Haiwen Diao, Ying Zhang, Lin Ma, and Huchuan Lu. Similarity reasoning and filtration for image-text matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1218–1226, 2021. 2, 8

[2] Lisa Anne Hendricks and Aida Nematzadeh. Probing image-language transformers for verb understanding. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3635–3644, 2021. 4

[3] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019. 9

[4] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019. 3

[5] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016. 4

[6] Varun K Nagaraja, Vlad I Morariu, and Larry S Davis. Modeling context between objects for referring expression understanding. In *European Conference on Computer Vision*, pages 792–807. Springer, 2016. 4

[7] Letitia Parcalabescu, Michele Cafagna, Lilitta Muradjan, Anette Frank, Iacer Calixto, and Albert Gatt. Valse: A task-independent benchmark for vision and language models centered on linguistic phenomena. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8253–8280, 2022. 4

[8] Sebastian Schuster, Ranjay Krishna, Angel Chang, Li Fei-Fei, and Christopher D Manning. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the fourth workshop on vision and language*, pages 70–80, 2015. 2

[9] Yun Wang, Juncheng Li, and Florian Metze. A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 31–35. IEEE, 2019. 6

[10] Sibei Yang, Guanbin Li, and Yizhou Yu. Dynamic graph attention for referring expression comprehension. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4644–4653, 2019. 8

[11] Sibei Yang, Guanbin Li, and Yizhou Yu. Graph-structured referring expression reasoning in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9952–9961, 2020. 1, 4, 8

[12] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *The Eleventh International Conference on Learning Representations*, 2022. 8