Language-Conditioned Feature Pyramids for Visual Selection Tasks

Taichi Iki[1,2] and **Akiko Aizawa**[1,2]

[1]National Institute of Informatics, Chiyoda-ku, Tokyo, Japan
[2]Graduate University for Advanced Studies, Hayama, Kanagawa, Japan
{iki,aizawa}@nii.ac.jp

Abstract

Referring expression comprehension, which is the ability to locate language to an object in an image, plays an important role in creating common ground. Many models that fuse visual and linguistic features have been proposed. However, few models consider the fusion of linguistic features with multiple visual features with different sizes of receptive fields, though the proper size of the receptive field of visual features intuitively varies depending on expressions. In this paper, we introduce a neural network architecture that modulates visual features with varying sizes of receptive field by linguistic features. We evaluate our architecture on tasks related to referring expression comprehension in two visual dialogue games. The results show the advantages and broad applicability of our architecture. Source code is available at https://github.com/Alab-NII/lcfp.

1 Introduction

Referring expressions are a ubiquitous part of human communication (Krahmer and Van Deemter, 2012) that must be studied in order to create machines that work smoothly with humans. Much effort has been taken to improve methods of creating visual common ground between machines, which have limited means of expression and knowledge about the real world, and humans, from the perspectives of both referring expression comprehension and generation (Moratz et al., 2002; Tenbrink and Moratz, 2003; Funakoshi et al., 2004, 2005, 2006; Fang et al., 2013). Even now, researchers are exploring possible methods of designing more realistic scenarios for applications, such as in visual dialogue games (De Vries et al., 2017; Haber et al., 2019; Udagawa and Aizawa, 2019).

Many models have been proposed for referring expression comprehension so far. As image recognition matured, Guadarrama et al. (2014) studied object retrieval methods based on category labels predicted by the recognition models. Hu et al. (2016b) extended this approach to broader natural language expression including categories of objects, their attributes, positional configurations, and interactions. In recent years, models that fuse linguistic features with visual features using deep learning have been studied (Hu et al., 2016b,a; Anderson et al., 2018; Deng et al., 2018; Misra et al., 2018; Li et al., 2018; Yang et al., 2019a,b; Liu et al., 2019; Can et al., 2020).

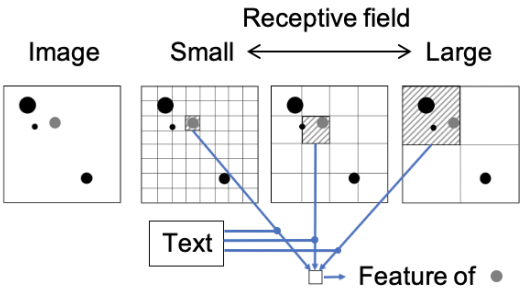

Figure 1: Illustration of visual features with different sizes of the receptive fields. Dots represent objects that have color and size as their attributes. Grids in the right three images represent the receptive fields of their visual features. Our architecture fuses linguistic features with each visual feature.

When fusing the linguistic features of a spatial referring expression with visual features, the size of the receptive field of visual features [1] is important. Let us take Figure 1 as an example. We can refer to the gray dot in the figure in various ways:

- a gray dot
- a dot next to the small dot
- a dot below and to the right of the large dot

[1]In this paper, we picture the size of the receptive field of visual features as the grid size in the input image. Note that the size of the receptive field in a real model is wider than the grid size in general because of multiple convolutional layers.

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 4687–4697

- the rightmost dot in a triangle consisting of three dots
- the third largest dot of four dots

As shown in the figure, there is an optimum size of receptive field when fusing the features of these expressions with the visual features. Although the small receptive field (in the second panel to the left) matches the expression *a gray dot*, it does not capture information about the triangle consisting of three dots to the upper left. Conversely, the largest receptive field (in the panel to the right) includes the triangle, but contains too much information to determine the color of the gray dot. Thus, linguistic and visual features have an optimum size of receptive field for fusion.

Few existing models, however, use fusion of linguistic features with visual features with different receptive field sizes. This is possibly because major datasets for referring expression comprehension, for example, Kazemzadeh et al. (2014); Plummer et al. (2015); Mao et al. (2016); Yu et al. (2016), use photographs and weigh expressions related to object category more often than positional relationships. Tenbrink and Moratz (2003); Tanaka et al. (2004); Liu et al. (2012, 2013) reveal that people often use group-based expressions (relative positional relationships of multiple objects) when there is no clear difference between objects; therefore, these expressions are not so unusual. Further investigation should be done on methods that handle referring expressions based on positional relationships.

For this reason, we focus on the OneCommon corpus (Udagawa and Aizawa, 2019), a recently proposed corpus on a visual dialogue game using composite images of simple figures. It captures various expressions based on positional relationships, such as group-based expressions, as shown in Figure 2.

In this paper, we introduce a neural network architecture for referring expression comprehension considering visual features with different sizes of the receptive fields, and evaluate it on the OneCommon task. Our structure combines feature pyramid networks (FPN) (Lin et al., 2017) and feature-wise linear modulation (FiLM) (Perez et al., 2018) and modulates visual features with different sizes of the receptive fields with linguistic features of referring expressions. FPN is an architecture that uses each layer of the hierarchical convolutional neural network (CNN) feature extractor for object detection;

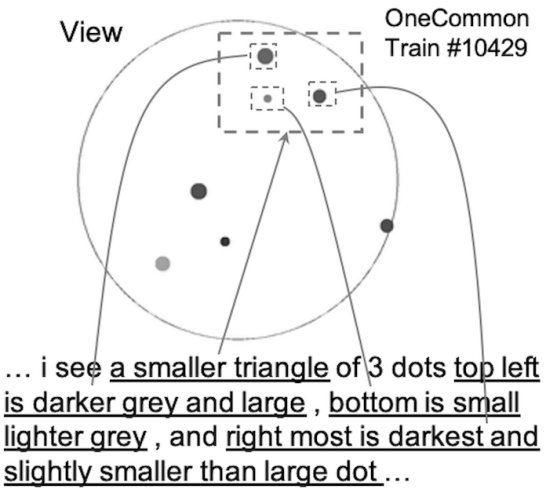

Figure 2: Example of OneCommon view and dialogue. In the OneCommon framework, two players observe slightly different views due to parallel shift. The game requires them to create common ground about the views through free conversation and identify the same dot. We show part of an utterance and underline some expressions that refer to an object or a group.

whereas, FiLM is a structure that robustly fuses linguistic features with visual features.

To confirm the broad applicability of our architecture, we further evaluate it on another task, which is expected to require the ability of object category recognition more than OneCommon does because it uses photographs. We find that our architecture achieves better accuracy in these tasks than some existing models, suggesting the advantage of fusion of linguistic features with multiple visual features that have different receptive fields.

The contributions of this paper are as follows:

1. We propose the language-conditioned feature pyramid (LCFP) architecture, which modulates visual features with multiple sizes of receptive fields using language features.
2. We apply LCFP to dialogue history object retrieval; our evaluation demonstrates the advantage of our architecture on referring expression comprehension in visual dialogue.

2 Dialogue History Object Retrieval

The main focus of this paper is the task of predicting the final object selected by the speaker given a dialogue history, a scene image, and candidate objects in the image. A dialogue history consists of a list of speaker and utterance pairs. We consider dialogues where speakers switch every turn. Candidate objects are indicated by bounding boxes in

the image. Some task instances provide additional information, such as object categories. Here, we call this task dialogue history object retrieval.

OneCommon Target Selection Task OneCommon is a dialogue corpus for common grounding. It contains 6,760 dialogues from a collaborative referring game where two players are given a view that contains 7 dots, as shown in Figure 2. Dots have four attributes: x/y coordinates on a plane, size, and color. Only some dots are seen in common because the centers of the players' views are different. The goal of the game is to select the same dot after talking. Target selection is a subtask of the game, requiring prediction of the dot that a player chose based on a given player's view and dialogue history.

GuessWhat?! Guessor Subtask GuessWhat?! (De Vries et al., 2017) is a game related to multimodal dialogue. Two players play the roles of oracle and questioner. They are given a photo and the oracle mentally selects an object. Then, the questioner asks the oracle yes-or-no questions to guess the object. The goal of the game is to select the object at the end of a question sequence. A published collection of game records consists of 150,000 games with human players, with a total of 800,000 visual question–answer pairs on 66,000 images extracted from the MS COCO dataset (Lin et al., 2014). The guesser subtask is to predict the correct object from 3–20 candidate objects based on a given photo and set of question–answer pairs. Candidate information includes bounding boxes and object category.

In addition to dialogue history object retrieval, there is an increasing amount of research on task design for visual dialogue games that require unique common understanding. For example, in the PhotoBook dataset (Haber et al., 2019), two participants are presented with multiple images, and they predict whether an image is presented only to them or also to the other person through conversation.

3 Related Work

This section first describes an overview of the models for referring expression comprehension and then gives some details about models related to the OneCommon Corpus and GuessWhat?!.

3.1 Models for Referring Expression Comprehension

Models for extracting objects from an image are often based on object detection (Ren et al., 2015; Liu et al., 2016; Lin et al., 2017; Redmon and Farhadi, 2018) or image segmentation (Ronneberger et al., 2015). Object detection considers only the bounding boxes of the objects. Image segmentation extracts the areas indicated by the outlines of the objects. Referring expression comprehension also includes reference detection (Hu et al., 2016b; Anderson et al., 2018; Deng et al., 2018; Yang et al., 2019a,b) and segmentation (Hu et al., 2016a; Li et al., 2018; Misra et al., 2018; Liu et al., 2019; Can et al., 2020) correspondingly.

The standard reference detection consists of two stages: detecting candidate objects and selecting objects that match the expression from the candidates. Essentially, they do not fuse visual feature maps with language when detecting candidates. Yang et al. (2019b) proposes a one-stage model that combines the feature map of the object detector with language to directly select the referred object. Whereas their model fuses linguistic and visual features after reducing visual features of the different receptive field sizes, ours fuses them before the reduction. Zhao et al. (2018) also proposes a model with a structure that fuses multiple scales and languages for weakly supervised learning. However, they use concatenation as the method of fusion, whereas we use FiLM.

For reference segmentation, Li et al. (2018) point out a lack of multi-scale semantics and propose a method that recursively fuses feature maps of different scales using a recurrent neural network (RNN). However, this method concatenates linguistic features with only the first input of the RNN; hence, the feature map in each scale and the linguistic features may be poorly fused. U-Net-based models (Misra et al., 2018; Can et al., 2020) have the most similar structure to ours. They produce hierarchical feature maps with CNNs, modulate those maps with language, and unify them into a single map through consecutive deconvolution operations.

The major difference between those U-Net-based models and ours is fusion architecture. The U-Net-based models generate kernels from linguistic features to convolve visual features. Our model operates an affine transformation on visual features using coefficients made from linguistic features in FiLM blocks. Suppose the dimensions of the

source and modulated visual features are D_s and D_m, respectively. Then, the size of the kernel for convolution is $D_s D_m$ and the size of the coefficients for affine transformation is $2D_m$. Because of this independency of D_s, our model has the advantage of being able to handle visual features with large dimensions, such as the last layer of ResNet50 (He et al., 2016) typically with 2048 dimensions.

3.2 Models for Dialogue History Object Retrieval

OneCommon Target Selection Udagawa and Aizawa (2019) proposed the baseline model TSEL, which creates the features of a candidate taking into account its attributes (size, color and position) and the average of the differences between its attributes and attributes of the other candidates. This model does not use visual features directly.

Udagawa and Aizawa (2020) extended the baseline model from the perspective of learning tasks and introduced TSEL-REF and TSEL-REF-DIAL. TSEL-REF has a similar structure to TSEL and learns in a multi-task setting. It resolves referring expressions in utterances, as well as the final prediction. Additional data consisting of manual annotations of reference resolution are used for the training. TSEL-REF-DIAL also learns on self-play of dialogue in addition to the TSEL-REF training.

GuessWhat?! Guesser Subtask The GuessWhat?! paper proposes baseline models that use object category and position to create candidate features. Although the paper reports that the extension of their baseline model to visual features from object recognition does not have any advantages, some models that use visual features, for example, A-ATT (Deng et al., 2018) and HACAN (Yang et al., 2019a) have recently improved the performance on GuessWhat?!. Their approach, based on reference detection and attention mechanism, fuses linguistic features with visual features that have a single size of the receptive fields.

4 Preliminary

We introduce two prerequisite architectures to describe our proposal.

4.1 Feature-wise Linear Modulation

A feature-wise linear modulation (Perez et al., 2018) block fuses a given language vector and feature map to make a new feature map. Let the output feature map dimension be d_{out}, the language vector v_{lang} with dimension d_{lang}, and the feature map f_{in} with dimension d_{in} and shape (h, w).

The Trainable parts of the block are two linear transformations B, G, two convolutional layers $\text{CNV}^{(1)}$, $\text{CNV}^{(2)}$ and a batch normalization (BN) (Ioffe and Szegedy, 2015) layer.

First, it performs a linear transformation on v_{lang} to obtain the coefficients of the affine transformation,

$$\beta = \text{B} v_{\text{lang}}; \quad \text{B} \in \text{R}^{d_{\text{lang}} d_{\text{out}}},$$
$$\gamma = \text{G} v_{\text{lang}}; \quad \text{G} \in \text{R}^{d_{\text{lang}} d_{\text{out}}}.$$

Second, it applies $\text{CNV}^{(1)}$ to f_{in} after concatenating a positional encode (PE),

$$f_{\text{vis}} = \text{F}\left(\text{CNV}^{(1)}\left(\text{PE}(f_{\text{in}})\right)\right),$$

where F is an activation function, typically a rectified linear unit (ReLU) (Nair and Hinton, 2010), $\text{PE}(f_{\text{in}})$ denotes the concatenation of the two-dimensional position of each pixel in f_{in} normalized in a range of $[-1, 1]$ on each axis.

Last, the second convolutional layer $\text{CNV}^{(2)}$ with BN and affine transformation is applied to f_{vis}.

$$f_{\text{fuse}} = \text{F}\left(\beta \odot \text{BN}(\text{CNV}^{(2)}(f_{\text{vis}})) + \gamma\right),$$
$$f_{\text{film}} = f_{\text{vis}} + f_{\text{fuse}} \qquad (1),$$

where $\odot$ denotes the element-wise product. Language and vision are fused in this equation. f_{film} is the FiLMed feature map. Note that f_{film} can be divided into language-independent f_{vis} and language-dependent f_{fuse} parts. We analyze the effect of the terms in Section 6.3

4.2 Feature Pyramid Networks

Feature Pyramid Networks (FPN) (Lin et al., 2017) use an object recognition model as a backbone and reconstruct semantically rich feature maps from the feature extraction results. Here, we suppose that the backbone is ResNet.

ResNet and Stages of Feature Map The ResNet family has a common structure for reducing the size of the input images. First, it converts an input image into a feature map with half the resolution of the image with a convolutional layer. Next, it reduces the map by a factor of two with the pooling operation. Subsequently, it applies some residual blocks, gradually reducing the resolution by half. This task is repeated until the size becomes 1/32 of the original image. We define the final layer of each resolution as the feature map of the stage;

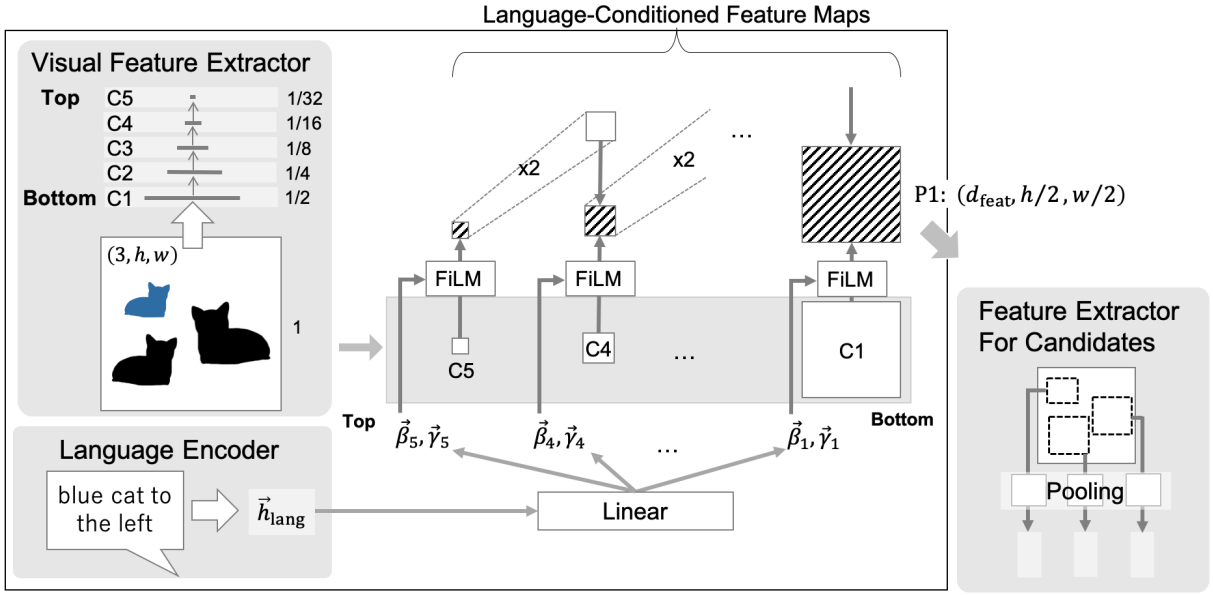

Figure 3: Overview of our architecture, consisting of a visual feature extractor and a language encoder. The feature maps (C1, ..., C5) from the extractor are fused in feature-wise linear modulation blocks with the language embedded and summed recursively. Striped boxes denote language-conditioned feature maps. For dialogue history object retrieval, the finest map (P1) is fed into the subsequent pooling layer.

namely, C1 is the final layer of the 1/2 resolution map, C2 is of 1/4, ..., C5 is of 1/32.

Top-down Reconstruction FPN makes feature pyramids from the stages of a backbone in a top-down manner. Suppose that $\mathrm{CNV}^{(2)}$, ..., $\mathrm{CNV}^{(5)}$ are trainable convolutional layers and P2, ..., P5 (P stands for pyramid) are the reconstructed feature maps on each stage[2]. Then Pi can be represented as follows:

$$\mathrm{P}i = \mathrm{CNV}^{(i)}(\mathrm{C}i) + \mathrm{Resize}_2(\mathrm{P}(i+1)) \quad (2).$$

where $\mathrm{P6} = 0$ and Resize_2 denotes the operation to enlarge the image twice. This means that Pi contains information about higher and coarser stages, which hold more complex semantics in general because of their wider receptive fields.

5 Proposed Method

Our architecture consists of language-conditioned feature pyramids (LCFP) for general feature extraction and a feature extractor for specific tasks, as shown in Figure 3. In this section, we describe LCFP and the following structure for dialogue history object retrieval.

[2]The reason we do not mention P1 is that the original paper does not use C1 and P1 owing to their large memory footprint.

5.1 Language-Conditioned Feature Pyramids

Language Encoder LCFP requires a fixed-length vector of language information to generate input for FiLM blocks. We can use any fixed vector, such as the last hidden layers of RNNs or transformer-based language models such as Devlin et al. (2019). Our proposal adopts gated recurrent unit (GRU) (Cho et al., 2014) in accordance with the FiLM paper (Perez et al., 2018). Suppose that d_{lang} is the dimension of hidden layer,

$$h_{\mathrm{lang}} = \mathrm{GRU}(\mathrm{text}) \in \mathrm{R}^{d_{\mathrm{lang}}}.$$

Visual Feature Extractor We use ResNet as our backbone. In addition to the C2-C5 described in Section 4.2, we use C1 because our goal is to incorporate information in the low stages, i.e., visual features with small receptive fields.

$$\{\mathrm{C}i; i = 1, ..., 5\} = \mathrm{ResNet}(\mathrm{image}).$$

Fusing Language and Vision The key idea to combine aforementioned two architectures is to replace convolutional layers of FPN in Equation 2 with FiLM blocks.

We represent the block as a function $\mathrm{FiLM}(v_{\mathrm{lang}}, f_{\mathrm{in}})$. Then, our feature reconstruction can be expressed as follows:

$$\mathrm{P}i = \mathrm{FiLM}^{(i)}(h_{\mathrm{lang}}, \mathrm{C}i) + \mathrm{Resize}_2(\mathrm{P}(i+1)) \quad (3),$$

where the weights of the FiLM block in each stage

are different from each other. We set kernel sizes for $\mathrm{CNV}_i^{(1),\,(2)}$ in each FiLM block 1×1 and 3×3, respectively, according to Perez et al. (2018). $\{\mathrm{P}i; i = 1, ..., 5\}$ is the output of LCFP.

5.2 LCFP-Based Dialogue History Object Retrieval

We formulate dialogue history object retrieval as a classification that predicts a selected object based on a dialogue history, scene image, and set of candidate information. The candidate information consists of a bounding box (x_1, y_1, x_2, y_2) in an image and a fixed-length vector v that represents the additional information.

Candidate Features We extract a region corresponding to a bounding box of each candidate from the feature map P1 obtained via LCFP. For candidate i, the features in the region are averaged to be converted into a fixed-length vector:

$$f'_i = \textstyle\sum_{k \in \mathrm{region}_i} \mathrm{P1}_k / \sum_{k \in \mathrm{region}_i} 1,$$

where region_i and $\mathrm{P1}_k$ indicate the region of candidate i and the vector at position k in feature map P1, respectively. We concatenate f'_i with v_i additional information vector for candidate i to make a full feature vector:

$$f_i = [f'_i; v_i].$$

Probability Calculation We apply a linear layer with ReLU activation to each feature and another linear layer with a one-dimensional output to obtain a logit for each candidate:

$$\mathrm{logit}_i = W_2 \mathrm{ReLU}(W_1 f_i + b).$$

We apply softmax over all logits of the candidates when we need probability of the selected candidate.

6 Experiments

We first validate the advantage of our architecture on two tasks in dialogue history object retrieval described in Section 2. We then investigate the cause of the advantage through ablation studies.

Common Text Processing We consider dialogue history as a text that starts with task name followed by a `<text>` token, with a sequence of utterances and a `<selection>` token at the end. Each utterance is interposed between a speaker token, `<you>` or `<them>`, and an end-of-sequence token `<eos>`. Tokenization of utterances is different for each task.

Model	Accuracy Valid.	Test (Full)	Test (SO)
TSEL	-	-	67.79 ±1.53
TSEL-REF	-	-	69.01 ±1.58
TSEL-REF-DIAL	-	-	69.09 ±1.12
LCFP	**72.99 ±1.37**	**73.47 ±1.09**	**78.26 ±1.21**
Human	-	-	90.79

Table 1: Accuracy on OneCommon Target Selection. SO indicates successful games only. The average results of 10 trials are shown. The values of TSEL, TSEL-REF, TSEL-REF-DIAL, and Human are from Udagawa and Aizawa (2020).

Common Implementation We implemented our model with the PyTorch framework (Paszke et al., 2019). We used ResNet50 provided from the PyTorch vision package, which is pretrained on object recognition tasks with the ImageNet dataset (Deng et al., 2009) as a backbone. All weights of the backbone, including those of statistics for batch normalization, are fixed. The dimensions of token embeddings, GRU hidden states, feature maps, additional information, and the last linear layer are 256, 1024, 256, 256 and 1024 respectively. For optimization, we used ADAM (Kingma and Ba, 2014) with alpha 5e-4, eps 1e-9, and mini-batch size 32. No regularization was used except for BN. We ran 5 epochs in a trial and chose the weight set with the lowest validation loss.

6.1 OneCommon Target Selection Task

Model Detail Tokenization was performed by splitting using white spaces; all tokens are uncased. Tokens that appear fewer than five times in the training dataset were replaced with an `<unk>` token. We drew the game views based on candidate dot data in a 224px square image. The additional information vector is disabled by inputting a vector that denotes that information is not provided.

Results Table 1 compares accuracy between the existing models and ours. Our model achieves better accuracy than the three models described in Section 3.2, although the accuracy is lower than with human performance. In particular, our model outperforms TSEL-REF and TSEL-REF-DIAL, which use additional learning, with learning only from standard training data. This result demonstrates the advantages and the high learning efficiency of our architecture.

Model		Error Train	Valid.	Test
LSTM[1]	SL	27.9	37.9	38.7
HRED[1]	SL	32.6	38.2	39.0
LSTM+VGG[1]	SL	26.1	38.5	39.2
HRED+VGG[1]	SL	27.4	38.4	39.6
A-ATT[2]	SL	26.7	33.7	34.2
HACAN w/o HAST[3]	SL	26.9	33.6	34.1
GST (SL)[4]	SL	24.7	33.7	34.3
LCFP (ours)	SL	**20.1 ±1.6**	**32.2 ±0.2**	**33.1 ±0.5**
HACAN[3]	HAST	26.1	32.3	33.2
GST (RL, Max.Q's=8)[4]	RL	**16.7**	**16.9**	**18.4**
Human[a]	-	9.0	9.0	9.2

Table 2: Error rate on GuessWhat?! Guesser Subtask. SL: Supervised learning, RL: Reinforcement learning, HAST: History-Advantaged Sequence Training (Yang et al., 2019a). The average result of 5 trials for LCFP. [1] (De Vries et al., 2017), [2] (Deng et al., 2018), [3] (Yang et al., 2019a) and [4] (Pang and Wang, 2020).

Model		Stage 5	4	3	2	1	Valid. err. OC	GW
Setting 1: Stages Ablation								
A5	f_{vis}	✓					45.8	38.4
	f_{fuse}	✓						
A3	f_{vis}	✓	✓	✓			28.5	33.1
	f_{fuse}	✓	✓	✓				
Full	f_{vis}	✓	✓	✓	✓	✓	**27.0**	**32.2**
	f_{fuse}	✓	✓	✓	✓	✓		
Setting 2: Language-Conditioned Parts Ablation								
A5'	f_{vis}	✓	✓	✓	✓	✓	38.8	37.8
	f_{fuse}	✓						
A3'	f_{vis}	✓	✓	✓	✓	✓	27.4	32.9
	f_{fuse}	✓	✓	✓				
Full	f_{vis}	✓	✓	✓	✓	✓	**27.0**	**32.2**
	f_{fuse}	✓	✓	✓	✓	✓		

Table 3: Ablation study on the OneCommon Target Selection Task (OC) and GuessWhat?! Guesser Subtask (GW). Error is shown. We ablate some of f_{vis} and f_{fuse} in the FiLM block at each stage. f_{vis} and f_{fuse} rows in each model show the condition where ✓ **indicates that the model uses the corresponding information.**

6.2 GuessWhat?! Guesser Subtask

Although it contains many referring expressions related to positional relationships, OneCommon uses a view with simple figures. We next evaluated our architecture on the Guesser subtask of Guess What?!, which uses photographs, to verify whether our structure can be applied to more complex visual information.

Model Detail We tokenized utterences by NLTK's TweetTokenizer under case-insensitive conditions and omitted tokens appearing fewer than five times in the training dataset. We resized the photos to 224px square, regardless of their aspect ratio. As additional information, we input object categories provided by the dataset by converting them into one-hot embedding vectors.

Results Table 2 shows the error rate of the task. The table also shows the learning methods of the models. Our model achieves the lowest error rate of models of supervised learning, including models that use visual features (LSTM+VGG, HRED+VGG, A-ATT and HACAN w/o HAST). This demonstrates that our architecture can be applied to visual input of natural objects as well as simple figures. Our method alone does not match the results of the method using reinforcement learning; however, our method can be combined with those more sophisticated learning methods. Examining such combinations will be an interesting topic for the future.

6.3 Ablation

To confirm the importance of fusing multiple visual features that have different receptive field sizes with linguistic features, we performed ablation in two settings: *Stage ablation* and *Language-conditioned parts ablation*. The former examines the effect of applying FiLM to small receptive fields by removing FiLM for some stages. The latter examines the effect of language modulation by leaving only the language-independent parts of FiLM.

Stage Ablation Stage ablation in Table 3 compares A5, A3 and Full models. A5 uses only the last stage of the image extractor and Full uses all stages. A3 is in the middle. The same trend exists for both OneCommon and GuessWhat?!; The Full model outperforms A5 and achieves a slightly better result than A3. This shows that considering visual features with a small receptive field size improves performance.

Language-Conditioned Parts Ablation This ablation introduces A5' and A3' models that use the language-independent f_{vis} part in all stages but do not use the language-dependent f_{fuse} part in some stages (see Equation. 1 in Section 4.1 for the definition of f_{vis} and f_{fuse}). Comparing A5 and A5' and A3 and A3' shows that the models consistently achieve better results when using the language-dependent part, suggesting that the language fusion has a positive impact. Although the

Token	N	TSEL [%]	LCFP [%]
(overall)	2702	66	74
triangle	304	60 (-6)	71 (-3)
group	100	55 (-11)	72 (-2)
pair	72	56 (-10)	72 (-2)
square	10	47 (-19)	80 (+6)
diamond	6	72 (+6)	100 (+26)
trapezoid	4	42 (-24)	75 (+1)

Table 4: Accuracy of example sets containing group-related tokens on OneCommon Target Selection. N represents the number of examples that contain group-related tokens in their dialogue. We show the differences between the accuracy of the overall and example sets in parentheses. We merged the validation and test splits for this table. The average results of three trials are shown.

impacts of the language fusion in stages 2 and 1 were expected to be relatively small owing to the small difference between Full and A3' model, they still have some impact on the performance.

Combining these, we conclude that the advantage evaluated in the previous subsection is a result of the fusion of linguistic features with multiple visual features with different receptive field sizes.

7 Discussion

Finally, this section focuses on linguistic expressions. We discuss the effect of our architecture on group-based referring expressions and our first intuition regarding the relationship between expression and receptive fields using OneCommon.

7.1 Effect on Group-Based Expression Comprehension

To obtain an insight into the performance of group-based referring expression, we performed an aggregation over examples in which dialogue includes tokens related to groups. We took the six tokens shown in Table 4 as a marker that indicates that the dialogue contains a group-based referring expression. If the model struggles to handle group-based referring expressions, the accuracy should be lower than the overall accuracy.

Table 4 shows the results. The baseline model TSEL yields low accuracy on *triangle*, *group*, *pair*, *square*, and *trapezoid* with large drops ranging from 6% to 24% compared to the overall accuracy. Conversely, our architecture reduces the drop. In the worst case *triangle*, accuracy drops by 3% . This supports the idea that our architecture improves the understanding of group-based referring expressions.

Note that dialogue history object retrieval resolves the final reference of the dialogue. The existance of a group-based referring expression does not necessarily mean that it relates to the answer; hence, this is indirect support.

7.2 Expressions and the Size of Receptive Fields

We visualized the activation pattern of the modulated features in our architecture to verify our first intuition that linguistic and visual features have an optimum size of receptive field for fusion.

Figure 4 shows the results. For visualization, we input simple expressions related to single attributes such as *select the largest dot* (size) or *select the darkest dot* (color). The stage with the most activated pattern varies depending on attributes in the expressions. We observed this phenomenon on different view inputs from the view in Figure 4. The model pays the most attention to stage 1, which has the smallest receptive field, when it receives an input expression related to color. Then, it moves to the stages with the larger receptive fields as the input changes to size and position. That is likely to correspond to the typical magnitude of localization.

These results suggest that the model selects visual features by the size of the receptive field according to the referring expression, supporting our first intuition.

Failure Cases Although the model makes a good predictions regarding size and color, it does not handle position well. Thus, there is still room to improve expression related to positional relationships, although the model improves this ability.

Through this visualization, we observed that our model tends to set the wrong range. For example, for four position-related expressions in Figure 4, the model predicts answers only from dots in the salient triangle formed by dots c, d and e.

A possible explanation of this observation is data bias. Because the OneCommon game framework rewards players if they successfully create common ground with each other, players may think to mention to more salient dots to increase the success rate. As a result, the variation of expressions could be restricted. In fact, Udagawa and Aizawa (2019) reports these trends on color and size attributes. This suggests the importance of exploring task design for data collection from the viewpoint of collecting a wide range of general reference expressions.

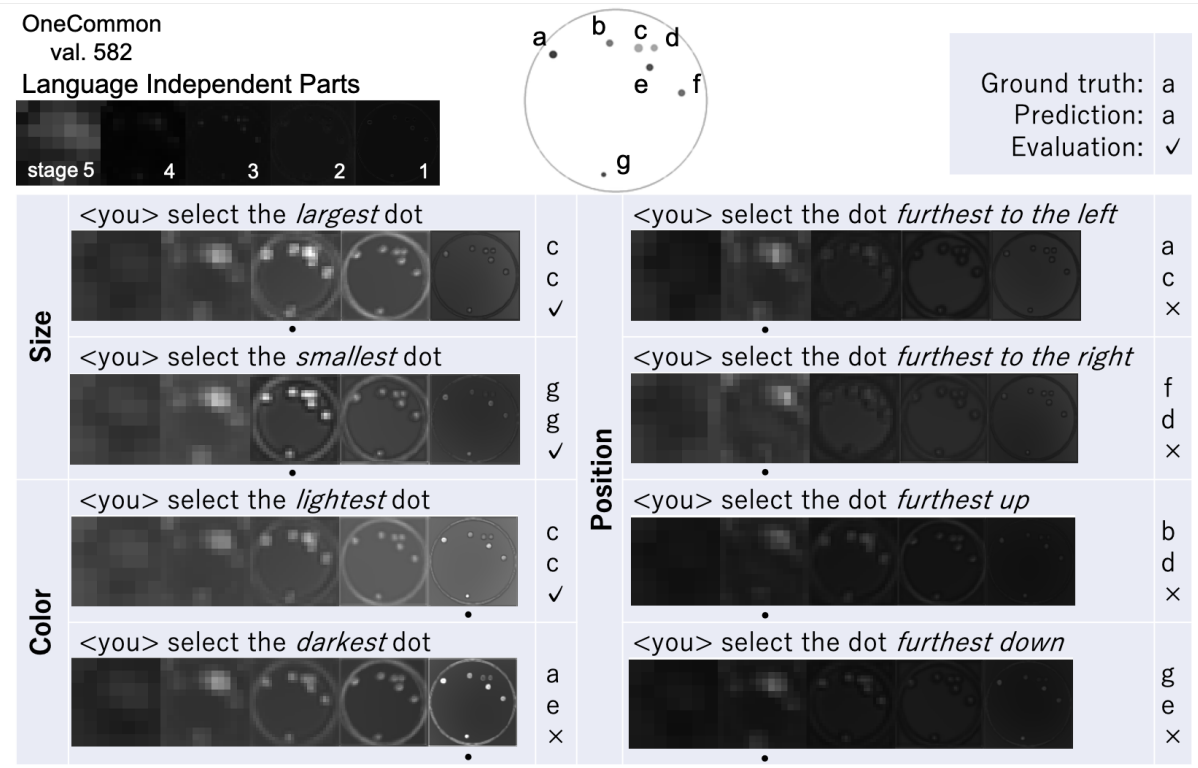

Figure 4: Single-attribute referring expressions and averaged activation pattern in feature-wise linear modulation blocks. All patterns are normalized with the same factor. The input view is shown in the top center (characters are a guide to identify the dots, not inputs). Each band of patterns has five maps corresponding to the stages of the model. The language-independent parts (f_{vis}) to the upper left are common to all expressions. The remaining parts (f_{fuse}) are responses to the expressions. Black dots under the maps indicate the stage with the largest activation.

8 Conclusion

To improve referring expression comprehension, this paper proposes a neural network architecture that modulates visual features; the visual features have different sizes of receptive fields in each hierarchy extracted by CNNs with linguistic features. As our architecture affine transforms visual features with linguistic features, it requires a lower calculation cost than methods that generate convolution kernels.

Our evaluation of referring expression comprehension tasks on two visual dialogue games demonstrates the model's advantage in the understanding of referring expressions and the broad applicability of our architecture. Ablation studies support the importance of multiple fusion.

We expect that hierarchical visual information is also important to generation. However, our architecture is difficult to directly apply to referring expression generation because it outputs modulated feature maps. Therefore, the future direction is to extend our architecture to language generation.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments. This work was supported by NEDO SIP-2 "Big-data and AI-enabled Cyberspace Technologies."

References

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086.

Ozan Arkan Can, İlker Kesen, and Deniz Yuret. 2020. Bilingunet: Image segmentation by modulating top-down and bottom-up visual processing with referring expressions. *arXiv preprint arXiv:2003.12739*.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

Harm De Vries, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron Courville. 2017. Guesswhat?! visual object discovery through multi-modal dialogue. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5503–5512.

Chaorui Deng, Qi Wu, Qingyao Wu, Fuyuan Hu, Fan Lyu, and Mingkui Tan. 2018. Visual grounding via accumulated attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7746–7755.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. IEEE.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Rui Fang, Changsong Liu, Lanbo She, and Joyce Chai. 2013. Towards situated dialogue: Revisiting referring expression generation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 392–402.

Kotaro Funakoshi, Satoru Watanabe, Naoko Kuriyama, and Takenobu Tokunaga. 2004. Generation of relative referring expressions based on perceptual grouping. In *Proceedings of the 20th international conference on Computational Linguistics*, pages 666–672. Association for Computational Linguistics.

Kotaro Funakoshi, Satoru Watanabe, and Takenobu Tokunaga. 2006. Group-based generation of referring expressions. In *Proceedings of the Fourth International Natural Language Generation Conference*, pages 73–80.

Kotaro Funakoshi, Satoru Watanabe, Takenobu Tokunaga, and Naoko Kuriyama. 2005. Understanding referring expressions involving perceptual grouping. In *2005 International Conference on Cyberworlds (CW'05)*, pages 413–420. IEEE.

Sergio Guadarrama, Erik Rodner, Kate Saenko, Ning Zhang, Ryan Farrell, Jeff Donahue, and Trevor Darrell. 2014. Open-vocabulary object retrieval. In *Robotics: science and systems*, volume 2, page 6.

Janosch Haber, Tim Baumgärtner, Ece Takmaz, Lieke Gelderloos, Elia Bruni, and Raquel Fernández. 2019. The photobook dataset: Building common ground through visually-grounded dialogue. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1895–1910.

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

Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. 2016a. Segmentation from natural language expressions. In *European Conference on Computer Vision*, pages 108–124. Springer.

Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. 2016b. Natural language object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4555–4564.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. 2014. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Emiel Krahmer and Kees Van Deemter. 2012. Computational generation of referring expressions: A survey. *Computational Linguistics*, 38(1):173–218.

Ruiyu Li, Kaican Li, Yi-Chun Kuo, Michelle Shu, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. 2018. Referring image segmentation via recurrent refinement networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5745–5753.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Changsong Liu, Rui Fang, and Joyce Y Chai. 2012. Towards mediating shared perceptual basis in situated dialogue. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 140–149. Association for Computational Linguistics.

Changsong Liu, Rui Fang, Lanbo She, and Joyce Chai. 2013. Modeling collaborative referring for situated referential grounding. In *Proceedings of the SIGDIAL 2013 Conference*, pages 78–86.

Runtao Liu, Chenxi Liu, Yutong Bai, and Alan L Yuille. 2019. Clevr-ref+: Diagnosing visual reasoning with referring expressions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4185–4194.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. 2016. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer.

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

Dipendra Misra, Andrew Bennett, Valts Blukis, Eyvind Niklasson, Max Shatkhin, and Yoav Artzi. 2018. Mapping instructions to actions in 3d environments with visual goal prediction. *arXiv preprint arXiv:1809.00786*.

Reinhard Moratz, Thora Tenbrink, John Bateman, and Kerstin Fischer. 2002. Spatial knowledge representation for human-robot interaction. In *International Conference on Spatial Cognition*, pages 263–286. Springer.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 807–814.

Wei Pang and Xiaojie Wang. 2020. Guessing state tracking for visual dialogue. In *The European Conference on Computer Vision (ECCV)*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. 2018. Film: Visual reasoning with a general conditioning layer. In *Thirty-Second AAAI Conference on Artificial Intelligence*, pages 3942–3951.

Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649.

Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

Hozumi Tanaka, Takenobu Tokunaga, and Yusuke Shinyama. 2004. Animated agents capable of understanding natural language and performing actions. In *Life-Like Characters*, pages 429–443. Springer.

Thora Tenbrink and Reinhard Moratz. 2003. Group-based spatial reference in linguistic human-robot interaction. In *Proceedings of EuroCogSci*, volume 3, pages 325–330.

Takuma Udagawa and Akiko Aizawa. 2019. A natural language corpus of common grounding under continuous and partially-observable context. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7120–7127.

Takuma Udagawa and Akiko Aizawa. 2020. An annotated corpus of reference resolution for interpreting common grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9081–9089.

Tianhao Yang, Zheng-Jun Zha, and Hanwang Zhang. 2019a. Making history matter: History-advantage sequence training for visual dialog. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2561–2569.

Zhengyuan Yang, Boqing Gong, Liwei Wang, Wenbing Huang, Dong Yu, and Jiebo Luo. 2019b. A fast and accurate one-stage approach to visual grounding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4683–4693.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. 2016. Modeling context in referring expressions. In *European Conference on Computer Vision*, pages 69–85. Springer.

Fang Zhao, Jianshu Li, Jian Zhao, and Jiashi Feng. 2018. Weakly supervised phrase localization with multi-scale anchored transformer network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5696–5705.