

**Towards
Diverse and Natural
Descriptions
for Image Captioning**

DAI, Bo

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy
in
Information Engineering

The Chinese University of Hong Kong
August 2018

Abstract

Generating descriptions of images has been an important task in computer vision. Despite the substantial progress in recent years, captions produced by existing methods are far from being perfect. Specifically, they resemble training captions in n-gram statistics, leading to rigidity, lack of variability, as well as insufficient coverage of images' important semantics. To generate natural and diverse captions, in this thesis we study the limitations of existing approaches from different aspects, including evaluation metrics, training strategies as well as model structures, and propose improved approaches accordingly.

At first, we point out issues of the commonly used training strategy and evaluation metrics, and propose an alternative training strategy based on Conditional Generative Adversarial Networks (CGAN), which jointly learns captioning models with a parametric evaluator assessing how well a caption fits the visual content. While the resulting evaluator serves as a better evaluation metric, the resulting captioning model is able to produce more diverse captions. This work has been accepted to the International Conference on Computer Vision 2017 (ICCV 2017).

In the second part, we follow the direction in the first part and afford one more training strategy, Contrastive Learning, that explicitly takes into

account the distinctiveness of captions, which introduces an independent reference model, and formulates two constraints about distinctiveness upon comparisons between the reference model and the target model. The proposed training strategy encourages captioning models to focus on the unique semantics of images, resulting in semantically well-matched captions. This work has been accepted to the 31 Annual Conference on Neural Information Processing Systems (NIPS2017).

In the third part, we rethink the form of latent states in existing approaches, where they are represented as 1D vectors. Alternatively, we propose to represent them as multi-channel 2D maps, to preserve important properties such as spatial locality more effectively. Compared to a captioning model with 1D states, the same model with 2D states consistently achieves higher performance with comparable parameter sizes. In addition, on top of 2D states, we are able to visually reveal the internal dynamics in the process of caption generation, as well as the connections between input visual domain and output linguistic domain. This work has been accepted to the European Conference on Computer Vision 2018 (ECCV2018).

In the last two parts, we take one step further to rethink the popular pipeline for image captioning, which represents images as feature vectors, then generates captions sequentially based on them. Instead of implicit feature vectors, we propose to represent visual semantics explicitly using scene-graphs, consisting of visual concepts, their attributes, as well as their pair-wise relationships, and generate captions upon such representations. To obtain scene-graphs from images, in the fourth part we propose a novel framework to detect visual relationships between pairs of objects, which correspond

to edges in the scene-graphs. As for caption generation, in the fifth part we propose to replace the sequential structure well adopted in existing captioning models with a recursive compositional one, which captures the hierarchical dependencies among words in a sentence, fitting the properties of natural language. Besides, it generalizes better and yields more diverse captions. The fourth work has been accepted to the Conference on Computer Vision and Pattern Recognition (CVPR2017), and the fifth work has been submitted to the 32 Annual Conference on Neural Information Processing Systems (NIPS2018).

摘要

圖像產生自然語言描述一直是計算機視覺的一個重要課題。儘管這個課題在今年有顯著的發展,現有方法產生的描述仍然有很大的提升空間。具體來說,他們在 n 元語法上(n -gram)同訓練用的描述有很大的相似性。這樣的相似性使得他們缺少變化,給人一種死板的感覺,同時對圖像的重要語義也缺乏足夠的涉及。為了產生更自然更多變的圖像描述,我們在本論文中從不同的角度研究了現有方法的侷限性,並提出相應的改進辦法。

在本論文的第一部分,我們指出了常用訓練策略和評價指標中的不足,並提出了另一種基於條件對抗生成模型(CGAN)的訓練策略。這個策略在訓練圖像描述模型的同時,同時訓練一個帶參數的評價模型。該評價模型可當作更好的評價指標,用來評價生成的描述的好壞以及其與對應圖像的契合度。同時這樣得到的描述模型也能夠產生更多變的描述。這部分工作已經被國際計算機視覺大會(ICCV 2017)接收。

在本論文的第二部分,我們沿第一部分的方向提出了又一種訓練策略,名對比學習。對比學習顯式的考慮了描述的獨特性(distinctiveness),通過引入一個獨立的參考模型,在參考模型和目標模型的對比上關於獨特性的兩個約束。這樣的訓練策略鼓勵目標模型關注圖像獨特的語義並把它加入到描述中,使

得產生的描述很好的和圖像聯繫在了一起。這部分工作已經被神經信息處理系統會議（NIPS 2017）接收。

在本論文的第三部分，我們審視了現有方法中對隱狀態的表示方法。在現有方法中他們被表示成了一維向量，對此我們提出了一種更好的表述方法，即二維多通道圖（multi-channel 2D map），用來更好的保留視覺信息的特性，比如空間上的局部性。同一模型採用二維隱狀態能比採用一維隱狀態在效果上有穩定的提升。更重要的是，利用二維隱狀態，我們可以從視覺上揭露描述生成過程中的內部動態，以及作輸入的視覺和作輸出的語言之間的聯繫。這部分工作已被歐洲計算機視覺大會（ECCV 2018）接收。

在本論文的最後兩個部分，我們沿著第三部分的方向進一步審視了產生圖像描述的主要方法。目前流行的產生圖像描述的方法將圖片表示成特征向量，並基於特征向量順序得產生整個描述。不同於隱式的特征向量，我們提出將圖像的語義顯式得用場景圖（scene-graph）來表示，並在該顯式表示的基礎上產生圖像的描述。場景圖中包含圖像的視覺實體和他們的性質，以及他們兩兩之間的關係。為了得到圖像的場景圖表示，在第四部分我們提出了一個全新的檢測物體兩兩之間的視覺關係的框架，可以用來構建場景圖的邊。至於描述的產生，在第五部分我們提出將現有方法中常用的順序結構替換成一種遞歸式的合成結構。相比順序結構，這種新型的結構能夠抓住同一個句子中單詞之間的多層次的依賴關係，更符合自然語言的特性。除此之外，它具有更强的泛化能力，還能夠產生更多變的描述。第四部分的工作已被計算機視覺和模式識別大會（CVPR2017）

接收，同時第五部分的工作已投稿至神經信息處理系統會議（NIPS 2018）。

Contents

Abstract	i
Chinese Abstract	iv
List of Figures	xii
List of Tables	xiv
1 Introduction	1
1.1 Image Captioning	1
1.2 Adversarial Learning for Captioning	3
1.3 Contrastive Learning for Captioning	3
1.4 Captioning Models with 2D States	4
1.5 Images as Scene-graphs	5
1.6 A Neural Compositional Captioning Model	5
2 Adversarial Learning for Captioning	7
2.1 Introduction	7
2.2 Related Work	10
2.3 Adversarial Learning for Captioning	14
2.3.1 Overall Formulation	14

2.3.2	Training G : Policy Gradient & Early Feedback	16
2.3.3	Training E : Naturalness & Relevance	18
2.4	Experiment	19
3	Contrastive Learning for Captioning	26
3.1	Introduction	26
3.2	Related Work	28
3.3	Background	30
3.4	Contrastive Learning for Captioning	32
3.4.1	Empirical Study: Self Retrieval	33
3.4.2	Contrastive Learning	34
3.4.3	Discussion	37
3.5	Experiment	38
3.5.1	Datasets	38
3.5.2	Settings	39
3.5.3	Results	40
4	Captioning Models with 2D States	45
4.1	Introduction	45
4.2	Related Work	47
4.3	Formulations	50
4.3.1	Encoder-Decoder for Image Captioning	50
4.3.2	From 1D to 2D	51
4.4	Qualitative Studies on 2D States	54
4.4.1	State Manipulation	54
4.4.2	Revealing Decoding Dynamics	56

4.4.3	Connecting Visual and Linguistic Domains	58
4.5	Comparison on Captioning Performance	61
4.5.1	Settings	62
4.5.2	Comparative Results	63
4.5.3	Ablation Study	65
4.6	Additional Materials	66
4.6.1	Activated Regions	66
4.6.2	Word-Channel Association	67

5 Images as Scene-graphs: Detecting Visual Relationships with

DR-Net		69
5.1	Introduction	69
5.2	Related Work	72
5.3	Visual Relationship Detection	74
5.3.1	Overall Pipeline	75
5.3.2	Joint Recognition	76
5.4	Deep Relational Network	79
5.4.1	Revisit of CRF	79
5.4.2	From CRF to DR-Net	80
5.4.3	Comparison with Other Formulations	83
5.5	Experiments	85
5.5.1	Experiment Settings	85
5.5.2	Comparative Results	86
5.5.3	Scene Graph Generation	90
5.6	Additional Materials	92
5.6.1	Proof of Eq.(5.4.2) in Section 5.4	92

5.6.2	Pair Filter	93
6	A Neural Compositional Captioning Model	95
6.1	Introduction	95
6.2	Related Work	97
6.3	Compositional Captioning	99
6.3.1	Explicit Representation of Semantics	100
6.3.2	Recursive Composition of Captions	101
6.4	Experiments	106
6.4.1	Experiment Settings	106
6.4.2	Experiment Results	107
6.5	Additional Materials	112
6.5.1	Semantic Non-Maximum Suppression for Noun Phrases	112
6.5.2	Encoders in the C-Module	113
6.5.3	Hyperparameters	115
7	Conclusion	116
	Bibliography	133

List of Figures

1.1	Multiple sample captions on each image.	2
1.2	Multiple sample captions on multiple similar images.	2
2.1	Illustration of image caption generation and evaluation.	11
2.2	Images with similar captions.	12
2.3	The structures of the generator G and the evaluator E	14
2.4	Human comparison between pairs of generators.	22
2.5	Descriptions generated using different \mathbf{z}	24
2.6	Comparison between captions generated by G -GAN and G - MLE	25
3.1	Comparison between nondistinctive and distinctive captions.	32
3.2	Comparison between captions generated by models trained with and without contrastive learning.	41
4.1	Structure of the encoder-decoder framework with RNN-2DS.	52
4.2	Results of the study of state manipulation.	55
4.3	Procedure of finding the activated region.	57
4.4	Activated regions of channels during the decoding processes.	58
4.5	Associations between words and channels.	59

4.6	Results of the study of channel ablation.	61
4.7	Comparison between RNN-2DS and LSTM-1DS	63
4.8	Sample captions generated by different decoders.	66
5.1	Sample visual relationships in images.	70
5.2	The proposed framework for visual relationship detection.	74
5.3	Structure of the spatial module.	77
5.4	Performances with different IoU thresholds.	88
5.5	Performances with different numbers of inference units.	89
5.6	Sample images and their generated scene-graphs.	91
5.7	The network for pair filtering.	93
6.1	Sample images with captions containing frequent but incorrect n-grams.	96
6.2	Overview of the compositional paradigm.	99
6.3	Structure of the phrase encoder.	104
6.4	Performances with different amounts of training data.	109
6.5	Comparison on the generalization ability of different methods.	110
6.6	Captions generated by changing composing orders or the set of noun-phrases.	112
6.7	Performances with different hyperparameters.	114

List of Tables

2.1	Quantitative comparison between different generators.	21
2.2	Image rankings for different generators.	24
3.1	Results of the study of self retrieval.	32
3.2	Quantitative comparison between captioning models on the online COCO testing server.	39
3.3	Quantitative comparison between captioning models on InstaPIC- 1.1M.	40
3.4	Quantitative comparison between models trained using differ- ent strategies.	42
3.5	Comparison between different model choices.	43
3.6	Results of periodical replacement of the reference model. . . .	43
4.1	Quantitative comparison between different decoders on COCO and Flickr30k.	64
4.2	Results of studies on vocabulary usage.	65
4.3	Quantitative comparison between RNN-2DS with different model choices.	67

5.1	Quantitative comparison between different models for visual relationship detection.	84
5.2	Quantitative comparison between different model choices.	84
5.3	Comparison between different relationship detectors on sample images.	85
5.4	Quantitative comparison between different methods for scene-graph generation.	90
6.1	Quantitative comparison between different captioning models on COCO and Flickr30k.	107
6.2	Quantitative comparison between different captioning models on diversity.	111
6.3	Quantitative comparison between different phrase encoders.	113

Chapter 1

Introduction

1.1 Image Captioning

Generating captions of images has been an important task in computer vision. Compared to other forms of semantic summary, *e.g.* object tagging, linguistic descriptions are often richer, more comprehensive, and a more natural way to convey image content. Along with the recent surge of deep learning technologies, there has been remarkable progress in image captioning over the past few years [115, 122, 127, 123, 57]. Latest studies on this topic often adopt a combination of an LSTM or its variant and a CNN. The former is to produce the word sequence while the latter is to capture the visual features of the images.

The advance in image captioning has been marked as a prominent success of AI¹. It has been reported [115, 122] that with certain metrics, like BLEU [84] or CIDEr [113], state-of-the-art techniques have already surpassed hu-

¹ARTIFICIAL INTELLIGENCE AND LIFE IN 2030, <https://ai1000.stanford.edu/2016-report>



	A cow standing in a field next to houses
	A cow standing in a field with houses
	A cow standing in a field of grass
	A train that is pulling into a station
	A train that is going into a train station
	A train that is parked in a train station

Figure 1.1: Captions generated by the state-of-the-art captioning model, where multiple captions are generated for one image.





			
a man flying through the air while riding a snowboard	a man flying through the air while riding a snowboard	a man flying through the air while riding a snowboard	a man flying through the air while riding a skateboard

Figure 1.2: Captions generated by the state-of-the-art captioning model, where a single caption is respectively generated for multiple similar images.

man’s performance. A natural question to ask is then: *has the problem of generating image captions been solved?* Let us take a step back, and look at samples of the current results, where multiple sentences are generated for one image in Figure 1.1, by the Encoder-and-Decoder model [115], a state-of-the-art caption generator. And in Figure 1.2, the same model generates a single sentence respectively for multiple similar images. Though faithfully describing the content of the images, these sentences are rigid, lacking in vitality and following a “safe” but “restrictive” way, resulting in almost identical sentences for similar images. In this thesis, we study these issues in differ-

ent angles, namely evaluation metrics, training strategies as well as model structures, and explore possible improvements accordingly.

1.2 Adversarial Learning for Captioning

The issues of existing image captioning techniques are related to a learning principle widely used in practice, that is, to maximize the likelihood of training samples. This principle encourages high resemblance to the “ground-truth” captions, while suppressing other reasonable descriptions. Conventional evaluation metrics, *e.g.* BLEU and METEOR, also favor such restrictive methods.

In this part of work, we explore an alternative approach, with the aim to improve the naturalness and diversity – two essential properties of human expression. Specifically, we propose a new framework based on Conditional Generative Adversarial Networks (CGAN), which jointly learns a generator to produce descriptions conditioned on images and an evaluator to assess how well a description fits the visual content. It is noteworthy that training a sequence generator is nontrivial. We overcome the difficulty by Policy Gradient, a strategy stemming from Reinforcement Learning, which allows the generator to receive early feedback along the way.

1.3 Contrastive Learning for Captioning

We argue the issues of existing image captioning techniques also associated to *distinctiveness*, an overlooked property of natural descriptions, which is closely related to the quality of captions, as distinctive captions are more

likely to describe images with their unique aspects.

In this part of work, we propose a new learning method, Contrastive Learning (CL), for image captioning. Specifically, via two constraints formulated on top of a reference model, the proposed method can encourage distinctiveness, while maintaining the overall quality of the generated captions. The proposed method is generic and can be used for models with various structures.

1.4 Captioning Models with 2D States

RNNs and their variants have been widely adopted for image captioning. In RNNs, the production of a caption is driven by a sequence of latent states. Existing captioning models usually represent latent states as vectors, taking this practice for granted. In this part of work, we rethink this choice and study an alternative formulation, namely using two-dimensional maps to encode latent states. This is motivated by the curiosity about a question: *how the spatial structures in the latent states affect the resultant captions?*

Our study leads to two significant observations. First, the formulation with 2D states is generally more effective in captioning, consistently achieving higher performance with comparable parameter sizes. The resultant captions also better fit the visual content. Second, 2D states preserve spatial locality. Taking advantage of this, we *visually* reveal the internal dynamics in the process of caption generation, as well as the connections between input visual domain and output linguistic domain.

1.5 Images as Scene-graphs

Mainstream captioning models often encode images using feature vectors, which is an implicit representation for visual semantics. Although effective, it is hard to interpret and evaluate. To overcome these issues, we propose to represent images as scene-graphs, which are structured representations, consisting of visual concepts, their attributes and their pair-wise visual relationships. While great success has been made in recognizing individual objects, reasoning about their relationships remains a challenging task.

In this part of work, we propose an integrated framework to tackle this task. At the heart of the framework is the Deep Relational Network (DR-Net), a novel formulation designed specifically for exploiting the statistical dependencies between objects and their relationships. The framework is able to detect visual relationships efficiently, facing the high diversity of visual appearance for each kind of relationships and the large number of distinct visual phrases.

1.6 A Neural Compositional Captioning Model

Mainstream captioning models often follow a sequential structure to generate captions, and inadequate generalization performance. In this part of work, we along the direction of encoding images using explicit representations and present an alternative paradigm, which factorizes the captioning procedure into two stages: (1) extracting an *explicit* semantic representation from the given image; and (2) constructing the caption based on a recursive *compositional* procedure in a bottom-up manner.

Compared to conventional ones, our paradigm better preserves the semantic content through an explicit factorization of semantics and syntax. By using the compositional generation procedure, caption construction follows a recursive structure, which naturally fits the properties of human language. Moreover, the proposed compositional procedure requires less data to train, generalizes better, and yields more diverse captions.

Chapter 2

Adversarial Learning for Captioning

2.1 Introduction

Being an important task in computer vision, image captioning has made remarkable progress in recent years. However, our brief survey (see Section 2.2) shows that existing efforts primarily focus on *fidelity*, while other essential qualities of human languages, *e.g.* *naturalness* and *diversity*, have received less attention. More specifically, mainstream captioning models, including those based on LSTMs [42], are mostly trained with the (conditional) maximum likelihood objective. This objective encourages the use of the *n-grams* that appeared in the training samples. Consequently, the generated sentences will bear high resemblance to training sentences in *detailed wording*, with very limited variability in expression [20]. Moreover, conventional evaluation metrics, such as BLEU [84], METEOR [66], ROUGE [71], and CIDEr [113], tend to favor this “*safe*” but restricted way. Under these met-

rics, sentences that contain matched n-grams would get substantially higher scores than those using variant expressions [2]. This issue is manifested by the fact that human descriptions get considerably lower scores.

Motivated to move beyond these limitations, we explore an alternative approach in this work. We wish to produce sentences that possess three properties: (1) **Fidelity**: the generated descriptions should reflect the visual content faithfully. Note that we desire the fidelity in *semantics* instead of *wording*. (2) **Naturalness**: the sentences should *feel* like what real people would say when presented with the image. In other words, when these sentences are shown to a real person, she/he would ideally not be able to tell that they are machine-generated. (3) **Diversity**: the generator should be able to produce notably different expressions given an image – just like human beings, different people would describe an image in different ways.

Towards this goal, we develop a new framework on top of the Conditional GAN [83]. GAN has been successfully used in image generation. As reported in previous works [92, 45], they can produce *natural* images nearly indistinguishable from real photos, freely or constrained by conditions. This work studies a different task for the GAN method, namely, generating *natural* descriptions conditioned on a given image. To our best knowledge, this is the first time the GAN method is used for image description.

Applying GANs to text generation is nontrivial. It comes with two significant challenges due to the special nature of linguistic representation. First, in contrast to image generation, where the transformation from the input random vector to the produced image is a deterministic continuous mapping, the process of generating a linguistic description is a *sequential sampling*

procedure, which samples a *discrete* token at each step. Such operations are *non-differentiable*, making it difficult to apply back-propagation directly. We tackle this issue via *Policy Gradient*, a classical method originating from reinforcement learning [109]. The basic idea is to consider the production of each word as an *action*, for which the reward comes from the evaluator. By approximating the stochastic policy with a parametric function approximator, we allow gradients to be back-propagated.

Second, in the conventional GAN setting, the generator would receive feedback from the evaluator when an entire sample is produced. For sequence generation, this would lead to several difficulties in training, including *vanishing gradients* and *error propagation*. To mitigate such difficulties, we devise a mechanism that allows the generator to get early feedback. Particularly, when a description is *partly* generated, our framework would calculate an approximated *expected future reward* through Monte Carlo rollouts [129]. Empirically, we found that this significantly improves the efficiency and stability of the training process.

Overall, our contributions can be briefly summarized as follows: (1) We explore an alternative approach to generate image descriptions, which, unlike most of the previous work, encourages not only *fidelity* but also *naturalness* and *diversity*. (2) From a technical standpoint, our approach relies on the conditional GAN method to learn the generator, instead of using MLE, a paradigm widely adopted in state-of-the-art methods. (3) Our framework not only results in a generator that can produce natural and diverse expressions, but also yields a description evaluator at the same time, which, as we will show in our experiments, is substantially more consistent with human

evaluation.

2.2 Related Work

Generation. Generating descriptions for images has been a long standing topic in computer vision. Early studies mostly adopted *detection-based* approaches. Such methods first detect visual concepts (*e.g.* object categories, relationships, and attributes) using CRFs [25, 60, 17], SVMs [68], or CNNs [24, 69], then generate descriptions thereon using simple methods, such as sentence templates [60, 68], or by retrieving relevant sentences from existing data [25, 24, 67, 62].

In recent years, the Encoder-and-Decoder paradigm proposed in [115] became increasingly popular. Many state-of-the-art frameworks [134, 127, 123, 78, 122, 115] for this task adopt the *maximum likelihood* principle for learning. Such a framework usually works as follows. Given an image I , it first derives a feature representation $\mathbf{f}(I)$, and then generates the words w_1, \dots, w_T sequentially, following a Markov process conditioned on $\mathbf{f}(I)$. The model parameters are learned via maximum likelihood estimation (MLE), *i.e.* maximizing the conditional log-likelihood of the training samples, as:

$$\sum_{(I_i, S_i) \sim \mathcal{D}} \sum_{t=0}^{T_i} \log p \left(w_i^{(t)} | \mathbf{f}(I), w_i^{(t-1)}, \dots, w_i^{(t-n)} \right) \quad (2.1)$$

Here, I_i and $S_i = (w_i^{(0)}, \dots, w_i^{(T_i)})$ are the image and the corresponding descriptive sentence of the i -th sample, and n is the order of the Markov chain – the distribution of the current word depends on n preceding words. Along with the popularity of deep neural networks, latest studies often adopt neural

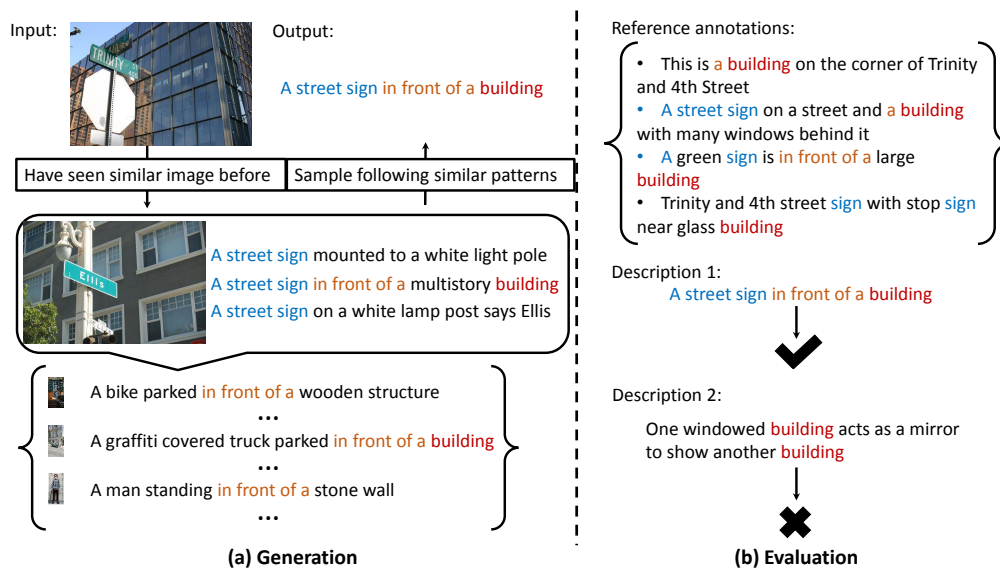


Figure 2.1: We illustrate the procedures of image caption generation and evaluation for state-of-the-art approaches. While the generation procedure tends to follow observed patterns, the evaluation procedure also favors this point. Best viewed in color.

networks for both image representation and language modeling. For example, [122] uses a CNN for deriving the visual features $\mathbf{f}(I)$, and an LSTM [42] net to express the sequential relations among words. Despite the evolution of the modeling choices, the maximum likelihood principle remains the predominant learning principle.

As illustrated in Figure 2.1, when similar images are presented, the sentences generated by such a model often contain *repeated* patterns [19]. This is not surprising – under the MLE principle, the joint probability of a sentence is, to a large extent, determined by whether it contains the frequent n-grams from the training set. Therefore, the model trained in this way will tend to produce such n-grams. In particular, when the generator yields a few of words that match the prefix of a frequent n-gram, the remaining words of that n-gram will likely be produced following the Markov chain.





	a woman holding a skateboard on a street						
	0.71	0.61	0.75	0.36	1.49	0.28	0.05
	B3	B4	ROUGE	METEOR	CIDEr	SPICE	E-GAN
	0.25	0.01	0.48	0.19	0.36	0.14	0.37
	three women one with a skateboard outside a store						
	a baseball player swinging a bat at a ball						
	0.71	0.65	0.78	0.39	2.21	0.28	0.48
	B3	B4	ROUGE	METEOR	CIDEr	SPICE	E-GAN
	0.01	0.01	0.31	0.23	0.82	0.25	0.82
	the umpire stands over a catcher as the batter swings						
	a man holding a tennis racquet on a tennis court						
	0.99	0.99	1.0	1.0	3.53	0.58	0.69
	B3	B4	ROUGE	METEOR	CIDEr	SPICE	E-GAN
	0.01	0.01	0.48	0.28	1.03	0.2	0.67
	a man getting ready to serve a tennis ball						

Figure 2.2: Examples of images with two semantically similar captions, selected from ground-truth annotations. While existing metrics assign higher scores to those with more matched n-grams, *E-GAN* gives scores consistent with human evaluation.

Evaluation. Along with the development of the generation methods, various evaluation metrics have been proposed to assess the quality of the generated sentences. Classical metrics include BLEU [84] and ROUGE [71], which respectively focuses on the precision and recall of n-grams. Beyond them, METEOR [66] uses a combination of both the precision and the recall of n-grams. CIDEr[113] uses weighted statistics over n-grams. As we can see, such metrics mostly rely on matching n-grams with the “*ground-truths*”. As a result, sentences that contain frequent n-grams will get higher scores as compared to those using variant expressions, as shown in Figure 2.2. Recently, a new metric SPICE [2] was proposed. Instead of matching between n-grams, it focuses on those linguistic entities that reflect visual concepts (*e.g.* objects and relationships). However, other qualities, *e.g.* the naturalness of the

expressions, are not considered in this metric.

Our Alternative Way. Previous approaches, including both generation methods and evaluation metrics, primarily focus on the *resemblance* to the training samples. While this is a *safe* way to generate plausible descriptions, it is *limited*. For example, when presented an image, different people would probably give different descriptions that do not overlap much in the wording patterns. This diversity in expression is an essential property of human languages, which, however, is often overlooked in previous works (both generation and evaluation). In this work, we explore an alternative approach – instead of emphasizing n-gram matching, we aim to improve the *naturalness* and *diversity*, *i.e.* generating sentences that feel like what real people would say, rather than focusing on word-by-word matching. Specifically, our approach jointly trains a generator G and an evaluator E in an adversarial way, where G is to produce natural descriptions, while E is to distinguish irrelevant or artificial descriptions from natural ones.

From a technical standpoint, our approach is based on the conditional GAN approach. GANs [33] and conditional GANs [83] are popular formulations for learning generators. For computer vision, GAN was originally introduced to generate images [92]. In a recent work [129], a text generator based on the GAN method was proposed. Note that this is an unconstrained generator that does not take into account any conditions. Hence, it can not be directly used for generating descriptions for images – in this task, the relevance of the generated text to the given image is essential. To our best knowledge, this is the first study that explores the use of *conditional* GAN in generating image descriptions.

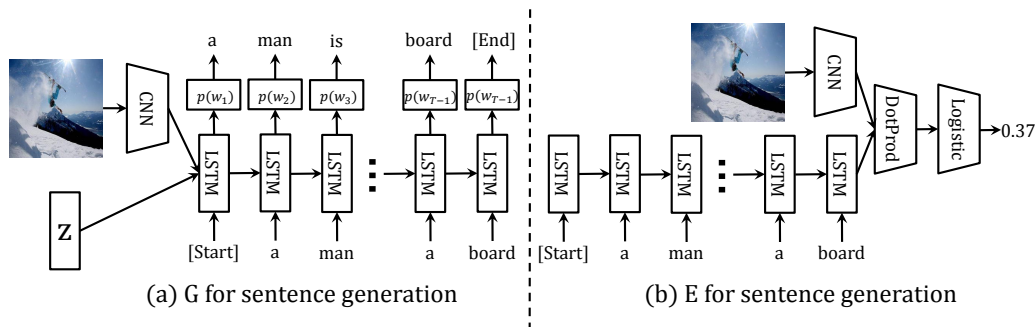


Figure 2.3: The structures of the generator G and the evaluator E .

2.3 Adversarial Learning for Captioning

We propose a new framework for generating image descriptions based on the conditional GAN [83] method, which consists of a generator G , and an evaluator E . Given an image I , the former is for generating *natural* and *semantically relevant* descriptions; while the latter is for evaluating how well a sentence describes I .

2.3.1 Overall Formulation

Our framework contains a *generator* G and a *evaluator* E , whose structures are respectively shown in Figure 2.3 (a) and (b). It is worth noting that our framework is orthogonal to works that focus on architectural designs of the G and the E . Their structures are not restricted to the ones introduced in this paper. In our framework, given an image I , the generator G takes two inputs: an image feature $\mathbf{f}(I)$ derived from a convolutional neural network (CNN) and a random vector \mathbf{z} . In particular, we follow the setting in NeuralTalk2¹, adopting *VGG16* [106] as the CNN architecture. The random vector \mathbf{z} allows the generator to produce different descriptions given an im-

¹<https://github.com/karpathy/neuraltalk2>

age. One can control the *diversity* by tuning the variance of \mathbf{z} . With both $\mathbf{f}(I)$ and \mathbf{z} as the initial conditions, the generator relies on an LSTM [42] net as a decoder, which generates a sentence, word by word. Particularly, the LSTM net assumes a sequence of latent states (s_0, s_1, \dots) . At each step t , a word w_t is drawn from the conditional distribution $p(w|s_t)$.

The evaluator E is also a neural network, with an architecture similar to G but operating in a different way. Given an image I and a descriptive sentence $S = (w_0, w_1, \dots)$, it embeds them into vectors $\mathbf{f}(I)$ and $\mathbf{h}(S)$ of the same dimension, respectively via a CNN and an LSTM net. Then the *quality* of the description, *i.e.* how well it describes I , is measured by the dot product of the embedded vectors, as

$$r_{\boldsymbol{\eta}}(I, S) = \sigma(\langle \mathbf{f}(I, \boldsymbol{\eta}_I), \mathbf{h}(S, \boldsymbol{\eta}_S) \rangle). \quad (2.2)$$

Here, $\boldsymbol{\eta} = (\boldsymbol{\eta}_I, \boldsymbol{\eta}_S)$ denotes the evaluator parameters, and σ is a logistic function that turns the dot product into a probability value in $[0, 1]$. Note that while the CNN and the LSTM net in E have the same structure as those in G , their parameters are not tied with each other.

For this framework, the learning objective of G is to generate descriptions that are *natural*, *i.e.* indistinguishable from what humans would say when presented with the same image; while the objective of E is to distinguish between artificial descriptions (*i.e.* those from G) and the real ones (*i.e.* those from the training set). This can be formalized into a minimax problem as follows:

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\eta}} \mathcal{L}(G_{\boldsymbol{\theta}}, E_{\boldsymbol{\eta}}). \quad (2.3)$$

Here, G_{θ} and E_{η} are a generator with parameter θ and an evaluator with parameter η . The objective function \mathcal{L} is:

$$\mathbb{E}_{S \sim \mathcal{P}_I} [\log r_{\eta}(I, S)] + \mathbb{E}_{\mathbf{z} \sim \mathcal{N}_0} [\log(1 - r_{\eta}(I, G_{\theta}(I, \mathbf{z})))] . \quad (2.4)$$

Here, \mathcal{P}_I denotes the descriptive sentences for I provided in the training set, \mathcal{N}_0 denotes a standard normal distribution, and $G_{\theta}(I, \mathbf{z})$ denotes the sentence generated with I and \mathbf{z} . The overall learning procedure alternates between the updating of G and E , until they reach an equilibrium.

This formulation reflects an essentially different philosophy in *how to train a description generator* as opposed to those based on MLE. As mentioned, our approach aims at the *semantical relevance* and *naturalness*, *i.e.* whether the generated descriptions feel like what human would say, while the latter focuses more on word-by-word patterns.

2.3.2 Training G : Policy Gradient & Early Feedback

As mentioned, unlike in conventional GAN settings, the production of sentences is a discrete sampling process, which is *nondifferentiable*. A question thus naturally arises - how can we *back-propagate the feedback* from E under such a formulation? We tackle this issue via *Policy Gradient* [109], a technique originating from reinforcement learning. The basic idea is to consider a sentence as a sequence of *actions*, where each word w_t is an action. The choices of such “actions” are governed by a *policy* π_{θ} .

With this interpretation, the generative procedure works as follows. It begins with an empty sentence, denoted by $S_{1:0}$, as the initial state. At each step t , the *policy* π_{θ} takes the conditions $\mathbf{f}(I)$, \mathbf{z} , and the preceding words

$S_{1:t-1}$ as inputs, and yields a conditional distribution $\pi_{\theta}(w_t|\mathbf{f}(I), \mathbf{z}, S_{1:t-1})$ over the extended vocabulary, namely all words plus an indicator of sentence end, denoted by e . This computation is done by moving forward along the LSTM net by one step. From this conditional distribution, an action w_t will be sampled. If $w_t = e$, the sentence will be terminated, otherwise w_t will be appended to the end. The *reward* of this sequence of actions S is $r_{\eta}(I, S)$, the score given by the evaluator E .

Now, we have defined an action space, a policy, and a reward function, and it seems that we are ready to apply the reinforcement learning method. However, there is a serious issue here – a sentence can only be evaluated when it is *completely* generated. In other words, we can only see the reward at the end. We found empirically that this would lead to a number of practical difficulties, *e.g.* gradients vanishing along a long chain and overly slow convergence in training.

We address this issue through *early feedback*. To be more specific, we evaluate an *expected future reward* as defined below when the sentence is *partially* generated:

$$V_{\theta, \eta}(I, \mathbf{z}, S_{1:t}) = \mathbb{E}_{S_{t+1:T} \sim G_{\theta}(I, \mathbf{z})}[r_{\eta}(I, S_{1:t} \oplus S_{t+1:T})]. \quad (2.5)$$

where \oplus represents the concatenation operation. Here, the expectation can be approximated using Monte Carlo rollouts [129]. Particularly, when we have a part of the sentence $S_{1:t}$, we can continue to sample the remaining words by simulating the LSTM net until it sees an end indicator e . Conducting this conditional simulation for n times would result in n sentences. We can use the evaluation score averaged over these simulated sentences to

approximate the *expected future reward*. To learn the generator G_{θ} , we use maximizing this expected reward $V_{\theta,\eta}$ as the learning objective. Following the argument in [109], we can derive the gradient of this objective *w.r.t.* θ as:

$$\tilde{\mathbb{E}} \left[\sum_{t=1}^{T_{\max}} \sum_{w_t \in \mathcal{V}} \nabla_{\theta} \pi_{\theta}(w_t | I, \mathbf{z}, S_{1:t-1}) \cdot V_{\theta',\psi}(I, \mathbf{z}, S_{1:t} \oplus w_t) \right]. \quad (2.6)$$

Here, \mathcal{V} is the vocabulary, T_{\max} is the max length of a description, and $\tilde{\mathbb{E}}$ is the mean over all simulated sentences within a mini-batch. θ' is a copy of the generator parameter θ at the beginning of the update procedure of the generator. During the procedure, the generator will be updated multiple times, and each update will use the same set of parameters (θ') to compute Eq.(2.5).

Overall, using policy gradients, we make the generator trainable with gradient descent. Using expected future reward, we can provide early feedback to the generator along the way, thus substantially improving the effectiveness of the training process. Note that policy gradients have also been used in image description generation in [95, 76]. These works, however, adopt conventional metrics, *e.g.* BLEU and CIDEr as rewards, instead of relying on GAN. Hence, their technical frameworks are fundamentally different.

2.3.3 Training E : Naturalness & Relevance

The primary purpose of E is to determine how well a description S describes a given image I . A good description needs to satisfy two criteria: *natural* and *semantically relevant*. To enforce both criteria, inspired by [92] we extend Eq.(2.4) to consider three types of descriptions for each training image I : (1) \mathcal{S}_I : the set of descriptions for I provided by human, (2) \mathcal{S}_G :

those from the generator G_{θ} , and (3) $\mathcal{S}_{\setminus I}$: the human descriptions for different images, which is uniformly sampled from all descriptions that are not associated with the given image I . To increase the scores for the descriptions in \mathcal{S}_I while suppressing those in the others, we use a joint objective formulated as:

$$\max_{\boldsymbol{\eta}} \mathcal{L}_E(\boldsymbol{\eta}) = \frac{1}{N} \sum_{i=1}^N \mathcal{L}_E(I_i; \boldsymbol{\eta}). \quad (2.7)$$

Here, N is the number of training images. The term for each image I_i is given by:

$$\begin{aligned} \mathcal{L}_E(I; \boldsymbol{\eta}) &= \mathbb{E}_{S \in \mathcal{S}_I} \log r_{\boldsymbol{\eta}}(I, S) \\ &\quad + \alpha \cdot \mathbb{E}_{S \in \mathcal{S}_G} \log(1 - r_{\boldsymbol{\eta}}(I, S)) \\ &\quad + \beta \cdot \mathbb{E}_{S \in \mathcal{S}_{\setminus I}} \log(1 - r_{\boldsymbol{\eta}}(I, S)). \end{aligned} \quad (2.8)$$

The second term forces the evaluator to distinguish between the human descriptions and the generated ones, which would in turn provide useful feedbacks to G_{θ} , pushing it to generate more *natural* descriptions. The third term, on the other hand, ensures the *semantic relevance*, by explicitly suppressing mismatched descriptions. The coefficients α and β are to balance the contributions of these terms, whose values are empirically determined on the validation set.

2.4 Experiment

Datasets We conducted experiments to test the proposed framework on two datasets: (1) *MSCOCO* [73], which contains 82,081 training images

and 40,137 validation images. (2) *Flickr30k* [128], which contains 31,783 images in total. We followed the split in [50], which has 1,000 images for validation, 1,000 for testing, and the rest for training. In both datasets, each image has at least 5 ground truth sentences. Note that our experiments involve comparison between human descriptions and model-generated ones. As we have no access to the ground-truth annotations of the testing images in MSCOCO, for this dataset, we use the training set for both training and validation, and the validation set for testing the performance.

Experimental settings To process the annotations in each dataset, we follow [50] to remove non-alphabet characters, convert all remaining characters to lower-case, and replace all the words that appeared less than 5 times with a special word *UNK*. As a result, we get a vocabulary of size 9,567 on MSCOCO, and a vocabulary of size 7,000 on Flickr30k. All sentences are truncated to contain at most 16 words during training. We respectively pretrain G using standard MLE [115], for 20 epoches, and E with supervised training based on Eq.(2.8), for 5 epoches. Subsequently, G and E are jointly trained, where each iteration consists of one step of G-update followed by one step of E-update. We set the mini-batch size to 64, the learning rate to 0.0001, and $n = 16$ in Monte Carlo rollouts. When testing, we use beam search based on the expected rewards from E-GAN, instead of the log-likelihoods, which we found empirically leads to better results.

Models We compare three methods for sentence generation: (1)**Human**: a sentence randomly sampled from ground-truth annotations of each image is used as the output of this method. Other human-provided sentences

		B3	B4	MT	RG	CD	SP	E-NGAN	E-GAN
COCO	human	0.290	0.192	0.240	0.465	0.849	0.211	0.527	0.626
	G-MLE	0.393	0.299	0.248	0.527	1.020	0.199	0.464	0.427
	G-GAN	0.305	0.207	0.224	0.475	0.795	0.182	0.528	0.602
Flickr	human	0.269	0.185	0.194	0.423	0.627	0.159	0.482	0.464
	G-MLE	0.372	0.305	0.215	0.479	0.767	0.168	0.465	0.439
	G-GAN	0.153	0.088	0.132	0.330	0.202	0.087	0.582	0.456

Table 2.1: This table lists the performances of different generators on MSCOCO and Flickr30k. On BLEU- $\{3,4\}$ (B3, B4), METEOR (MT), ROUGE_L (RG), CIDEr (CD), and SPICE (SP), *G-MLE* is shown to be the best among all generators, surpassing human by a significant margin. While *E-NGAN* regard *G-GAN* as the best generator, *E-GAN* regard *human* as the best one.

will be used as the references for metric evaluation. This baseline is tested for the purpose of comparing human-provided and model-generated descriptions. (2)**G-MLE**: a generator trained based on MLE [115] is used to produce the descriptions. This baseline represents the state-of-the-art of mainstream methods. (3)**G-GAN**: the same generator trained by our framework proposed in this paper, which is based on the conditional GAN formulations.

For both *G-MLE* and *G-GAN*, *VGG16* [106] is used as the image encoders. Activations at the `fc7` layer, which are of dimension 4096, are used as the image features and fed to the description generators. Note that *G-GAN* also takes a random vector \mathbf{z} as input. Here, \mathbf{z} is a 1024-dimensional vector, whose entries are sampled from a standard normal distribution.

Evaluation metrics We consider multiple evaluation metrics, including six conventional metrics BLEU-3 and BLEU-4[84], METEOR[66], ROUGE_L[71], CIDEr[113], SPICE[2], and two additional metrics relevant to our formulation: *E-NGAN* and *E-GAN*. Particularly, *E-GAN* refers to the evaluator

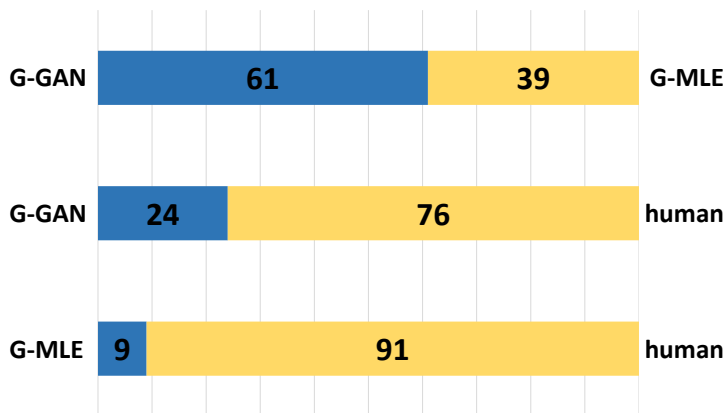


Figure 2.4: The figure shows the human comparison results between each pair of generators. With names of the generators placed at each side of the comparison, the blue and orange areas respectively indicate percentages of the generator in the left and right being the better one.

trained using our framework, *E-NGAN* refers to the evaluator trained according to Eq.(2.8) without updating the generator alternatively. In other words, it is trained to distinguish between human-provided sentences and those generated by an MLE-based model.

Table 2.1 lists the performances of different generators under these metrics. On both datasets, the sentences produced by *G-MLE* receive considerably higher scores than those provided by human, on nearly all conventional metrics. This is not surprising. As discussed earlier, such metrics primarily focus on n-gram matching *w.r.t.* the references, while ignoring other important properties, *e.g.* naturalness and diversity. These results also clearly suggest that these metrics may not be particularly suited when evaluating the overall quality of the generated sentences. On the contrary, *E-GAN* regards *Human* as the best generator, while *E-NGAN* regards *G-GAN* as the best one. These two metrics obviously take into account more than just n-gram matching.

User study & qualitative comparison To fairly evaluate the quality of the generated sentences as well as how *consistent* the metrics are with human’s perspective, we conducted a user study. Specifically, we invited 30 human evaluators to compare the outputs of different generators. Each time, a human evaluator would be presented an image with two sentences from different methods and asked to choose the better one. Totally, we collected about 3,000 responses.

The comparative results are shown in Figure 2.4: From human’s views, *G-GAN* is better than *G-MLE* in 61% of all cases. In the comparison between human and models, *G-MLE* only won in 9% of the cases, while *G-GAN* won in over 24%. These results clearly suggest that the sentences produced by *G-GAN* are of considerably higher quality, *i.e.* being more natural and semantically relevant. The examples in Figure 2.6 also confirm this assessment. Particularly, we can see when *G-MLE* is presented with similar images, it tends to generate descriptions that are almost the same. On the contrary, *G-GAN* describes them with more distinctive and diverse ones. We also varied \mathbf{z} to study the capability of *G-GAN* in giving *diverse* descriptions while maintaining the semantical relatedness. The qualitative results are listed in Figure 2.5.

For the evaluation metrics, the assessments provided by *E-GAN* are the most consistent with human’s evaluation, where the Kendall’s rank correlation coefficient between *E-GAN* and *HE* is 0.14, while that for CIDEr and SPICE are -0.30 and -0.25. Also note that *E-GAN* yields a larger numerical gap between scores of human and those of other generators as compared to *E-NGAN*, which suggests that adversarial training can improve the discrim-

		R@1	R@3	R@5	R@10
S	G-MLE	5.06	12.28	18.24	29.30
	G-GAN	14.30	30.88	40.06	55.82
P	G-MLE	9.88	20.12	27.30	39.94
	G-GAN	12.04	23.88	30.70	41.78

Table 2.2: The recalls of image rankings for different generators. Here recalls is the ratio of the original image being in the top- k in the ranked lists. The ranks are based on the similarities (S) between a image and a description, estimated by E -GAN, as well as the log-likelihoods (P), computed by different generators.





			
Z₁	a baseball player holds a bat up to hit the ball	a man riding a snowboard down a slope	a group of people sitting around a table having a meal in a restaurant
Z₂	a baseball player holding white bat and wear blue baseball uniform	a person standing on a snowboard sliding down a hill	a young man sitting at a table with coffee and a lot of food
Z₃	a professional baseball player holds up his bat as he watches	a man is jumping over a snow covered hill	a pretty young man sitting next to two men in lots of people

Figure 2.5: This figure shows example images with descriptions generated by G -GAN with different \mathbf{z} .

inative power of the evaluator.

Evaluation by retrieval To compare the *semantic relevance*, we conducted an experiment using generated descriptions for retrieval. Specifically, we randomly select 5,000 images from the MSCOCO validation set; and for each image, we use the generated description as a query, ranking all 5,000 images according to the similarities between the images and the descriptions, computed by E -GAN, as well as the log-likelihoods. Finally, we compute the recall of the original image that appeared in the top- k ranks. The results for $k = 1, 3, 5, 10$ are listed in Table 2.2, where G -GAN is shown to provide

				
G-MLE	a man flying through the air while riding a snowboard	a man flying through the air while riding a snowboard	a man flying through the air while riding a snowboard	a man flying through the air while riding a skateboard
G-GAN	a man on a skateboard in a snowy park	a man skiing down the slope near a mountain	a man performing a grind trick on a skateboard ramp	a man with stunts on his skis in the snow
				
G-MLE	a group of people standing around a boat	a group of people sitting around a table	a group of people sitting at a table	a group of people sitting around a living room
G-GAN	the bench is sitting on the ground by the water	a group of people watching each other	a table with a lot of stuff on it	furnished living room with furniture and built area

Figure 2.6: This figure lists some images and corresponding captions generated by *G-GAN* and *G-MLE*. *G-MLE* tends to generate similar descriptions for similar images, while *G-GAN* generates better distinguishable descriptions for them.

more discriminative descriptions, outperforming *G-MLE* by a large margin across all cases.

Failure Analysis We analyzed failure cases and found that a major kind of errors is the inclusion of incorrect details. *e.g.* colors (red/yellow hat), and counts (three/four people). A possible cause is that there are only a few samples for each particular detail, and they are not enough to make the generator capture these details reliably. Also, the focus on diversity and overall quality may also encourage the generator to include more details, with the risk of some details being incorrect.

Chapter 3

Contrastive Learning for Captioning

3.1 Introduction

Image captioning, a task to generate natural descriptions of images, has been an active research topic in computer vision and machine learning. Thanks to the advances in deep neural networks, especially the wide adoption of RNN and LSTM, there has been substantial progress on this topic in recent years [115, 122, 78, 95]. However, studies [16, 24, 20, 63] have shown that even the captions generated by state-of-the-art models still leave a lot to be desired. Compared to human descriptions, machine-generated captions are often quite rigid and tend to favor a “*safe*” (*i.e.* matching parts of the training captions in a word-by-word manner) but *restrictive* way. As a consequence, captions generated for different images, especially those that contain objects of the same categories, are sometimes very similar [16], despite their differences in other aspects.

We argue that **distinctiveness**, a property often overlooked in previous work, is significant in natural language descriptions. To be more specific, when people describe an image, they often mention or even emphasize the *distinctive* aspects of an image that distinguish it from others. With a distinctive description, someone can easily identify the image it is referring to, among a number of similar images. In this work, we performed a *self-retrieval* study (see Section 3.4.1), which reveals the lack of distinctiveness affects the quality of descriptions.

From a technical standpoint, the lack of *distinctiveness* is partly related to the way that the captioning model was learned. A majority of image captioning models are learned by Maximum Likelihood Estimation (MLE), where the probabilities of training captions conditioned on corresponding images are maximized. While well grounded in statistics, this approach does not explicitly promote distinctiveness. Specifically, the differences among the captions of different images are not explicitly taken into account. We found empirically that the resultant captions highly resemble the training set in a word-by-word manner, but are not *distinctive*.

In this paper, we propose **Contrastive Learning (CL)**, a new learning method for image captioning, which explicitly encourages *distinctiveness*, while maintaining the overall quality of the generated captions. Specifically, it employs a baseline, *e.g.* a state-of-the-art model, as a *reference*. During learning, in addition to true image-caption pairs, denoted as (I, c) , this method also takes as input *mismatched pairs*, denoted as (I, c') , where c' is a caption describing another image. Then, the target model is learned to meet two goals, namely (1) giving higher probabilities $p(c|I)$ to positive pairs, and

(2) lower probabilities $p(c_j|I)$ to negative pairs, compared to the reference model. The former ensures that the overall performance of the target model is not inferior to the reference; while the latter encourages distinctiveness.

It is noteworthy that the proposed learning method (CL) is generic. While in this paper, we focused on models based on recurrent neural networks [115, 78], the proposed method can also generalize well to models based on other formulations, *e.g.* probabilistic graphical models [25, 60]. Also, by choosing the state-of-the-art model as the reference model in CL, one can build on top of the latest advancement in image captioning to obtain improved performances.

3.2 Related Work

Models for Image Captioning The history of image captioning can date back to decades ago. Early attempts are mostly based on detections, which first detect visual concepts (*e.g.* objects and their attributes) [60, 25] followed by template filling [60] or nearest neighbor retrieving for caption generation [20, 25]. With the development of neural networks, a more powerful paradigm, *encoder-and-decoder*, was proposed by [115], which then becomes the core of most state-of-the-art image captioning models. It uses a CNN [106] to represent the input image with a feature vector, and applies a LSTM net [42] upon the feature to generate words one by one.

Based on the encoder-and-decoder, many variants are proposed, where *attention mechanism* [122] appears to be the most effective add-on. Specifically, attention mechanism replaces the feature vector with a set of feature vectors, such as the features from different regions [122], and those under

different conditions [134]. It also uses the LSTM net to generate words one by one, where the difference is that at each step, a mixed guiding feature over the whole feature set, will be *dynamically* computed. In recent years, there are also approaches combining attention mechanism and detection. Instead of doing attention on features, they consider the attention on a set of detected visual concepts, such as attributes [126] and objects [127].

Despite of the specific structure of any image captioning model, it is able to give $p(c|I)$, the probability of a caption conditioned on an image. Therefore, all image captioning models can be used as the target or the reference in CL method.

Learning Methods for Image Captioning Many state-of-the-art image captioning models adopt *Maximum Likelihood Estimation (MLE)* as their learning method, which maximizes the conditional log-likelihood of the training samples, as:

$$\sum_{(c_i, I_i) \in \mathcal{D}} \sum_{t=1}^{T_i} \ln p(w_i^{(t)} | I_i, w_i^{(t-1)}, \dots, w_i^{(1)}, \boldsymbol{\theta}), \quad (3.1)$$

where $\boldsymbol{\theta}$ is the parameter vector, I_i and $c_i = (w_i^{(1)}, w_i^{(2)}, \dots, w_i^{(T_i)})$ are a training image and its caption. Although effective, some issues, including high resemblance in model-generated captions, are observed [16] on models learned by MLE.

Facing these issues, alternative learning methods are proposed in recent years. Techniques of reinforcement learning (RL) have been introduced in image captioning by [95] and [76]. RL sees the procedure of caption generation as a procedure of sequentially sampling actions (words) in a policy

space (vocabulary). The rewards in RL are defined to be evaluation scores of sampled captions. Note that distinctiveness has not been considered in both approaches, RL and MLE.

Prior to this work, some relevant ideas have been explored [112, 81, 16]. Specifically, [112, 81] proposed an introspective learning (IL) approach that learns the target model by comparing its outputs on (I, c) and (I_j, c) . Note that IL uses the target model itself as a reference. On the contrary, the reference model in CL provides more *independent* and *stable* indications about distinctiveness. In addition, (I_j, c) in IL is pre-defined and fixed across the learning procedure, while the negative sample in CL, *i.e.* (I, c_j) , is *dynamically* sampled, making it more diverse and random. Recently, Generative Adversarial Networks (GAN) was also adopted for image captioning [16], which involves an evaluator that may help promote the distinctiveness. However, this evaluator is *learned* to *directly* measure the distinctiveness as a parameterized approximation, and the approximation accuracy is not ensured in GAN. In CL, the *fixed* reference provides stable *bounds* about the distinctiveness, and the bounds are supported by the model’s performance on image captioning. Besides that, [16] is specifically designed for models that generate captions word-by-word, while CL is more generic.

3.3 Background

Our formulation is partly inspired by *Noise Contrastive Estimation (NCE)* [39]. NCE is originally introduced for estimating probability distributions, where the partition functions can be difficult or even infeasible to compute. To estimate a parametric distribution $p_m(\cdot; \theta)$, which we refer to as the *target* dis-

tribution, NCE employs not only the observed samples $X = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{T_m})$, but also the samples drawn from a *reference* distribution p_n , denoted as $Y = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{T_n})$. Instead of estimating $p_m(\cdot; \boldsymbol{\theta})$ directly, NCE estimates the density ratio p_m/p_n by training a classifier based on logistic regression.

Specifically, let $U = (\mathbf{u}_1, \dots, \mathbf{u}_{T_m+T_n})$ be the union of X and Y . A binary class label C_t is assigned to each u_t , where $C_t = 1$ if $u_t \in X$ and $C_t = 0$ if $u_t \in Y$. The posterior probabilities for the class labels are therefore

$$P(C = 1|\mathbf{u}, \boldsymbol{\theta}) = \frac{p_m(\mathbf{u}; \boldsymbol{\theta})}{p_m(\mathbf{u}; \boldsymbol{\theta}) + \nu p_n(\mathbf{u})}, \quad P(C = 0|\mathbf{u}, \boldsymbol{\theta}) = \frac{\nu p_n(\mathbf{u})}{p_m(\mathbf{u}; \boldsymbol{\theta}) + \nu p_n(\mathbf{u})}, \quad (3.2)$$

where $\nu = T_n/T_m$. Let $G(\mathbf{u}; \boldsymbol{\theta}) = \ln p_m(\mathbf{u}; \boldsymbol{\theta}) - \ln p_n(\mathbf{u})$ and $h(\mathbf{u}, \boldsymbol{\theta}) = P(C = 1|\mathbf{u}, \boldsymbol{\theta})$, then we can write

$$h(\mathbf{u}; \boldsymbol{\theta}) = r_\nu(G(\mathbf{u}; \boldsymbol{\theta})), \quad \text{with} \quad r_\nu(z) = \frac{1}{1 + \nu \exp(-z)}. \quad (3.3)$$

The objective function of NCE is the joint conditional log-probabilities of C_t given the samples U , which can be written as

$$\mathcal{L}(\boldsymbol{\theta}; X, Y) = \sum_{t=1}^{T_m} \ln[h(\mathbf{x}_t; \boldsymbol{\theta})] + \sum_{t=1}^{T_n} \ln[1 - h(\mathbf{y}_t; \boldsymbol{\theta})]. \quad (3.4)$$

Maximizing this objective with respect to $\boldsymbol{\theta}$ leads to an estimation of $G(\cdot; \boldsymbol{\theta})$, the logarithm of the density ratio p_m/p_n . As p_n is a known distribution, $p_m(\cdot; \boldsymbol{\theta})$ can be readily derived.

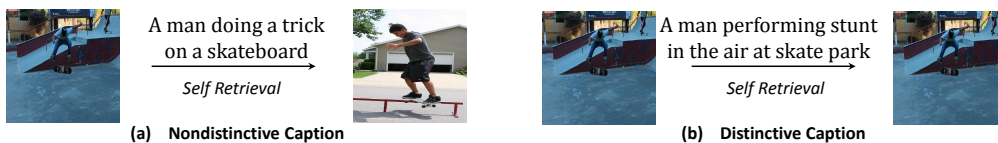


Figure 3.1: This figure illustrates respectively a nondistinctive and distinctive captions of an image, where the nondistinctive one fails to retrieve back the original image in *self retrieval* task.

Method	Self Retrieval Top-K Recall				Captioning	
	1	5	50	500	ROUGE.L	CIDEr
Neuraltalk2 [50]	0.02	0.32	3.02	27.50	0.652	0.827
AdaptiveAttention [78]	0.10	0.96	11.76	78.46	0.689	1.004
AdaptiveAttention + CL	0.32	1.18	11.84	80.96	0.695	1.029

Table 3.1: This table lists results of self retrieval and captioning of different models. The results are reported on standard MSCOCO test set. See Section 3.4.1 for more details.

3.4 Contrastive Learning for Captioning

Learning a model by characterizing desired properties relative to a strong baseline is a convenient and often quite effective way in situations where it is hard to describe these properties directly. Specifically, in image captioning, it is difficult to characterize the distinctiveness of natural image descriptions via a set of rules, without running into the risk that some subtle but significant points are missed. Our idea in this work is to introduce a baseline model as a reference, and try to enhance the distinctiveness on top, while maintaining the overall quality of the generated captions.

In the following we will first present an empirical study on the correlation between *distinctiveness* of its generated captions and the *overall performance* of a captioning model. Subsequently, we introduce the main framework of *Contrastive Learning* in detail.

3.4.1 Empirical Study: Self Retrieval

In most of the existing learning methods of image captioning, models are asked to generate a caption that best describes the semantics of a given image. In the meantime, **distinctiveness** of the caption, which, on the other hand, requires the image to be the best matching *among all images* for the caption, has not been explored. However, distinctiveness is crucial for high-quality captions. A study by Jas [47] showed that *specificity* is common in human descriptions, which implies that image descriptions often involve distinctive aspects. Intuitively, a caption satisfying this property is very likely to contain key and unique content of the image, so that the original image could easily be retrieved when the caption is presented.

To verify this intuition, we conducted an empirical study which we refer to as *self retrieval*. In this experiment, we try to retrieve the original image given its model-generated caption and investigate top- k recalls, as illustrated in Figure 3.1. Specifically, we randomly sampled 5,000 images ($I_1, I_2, \dots, I_{5000}$) from standard MSCOCO [73] test set as the experiment benchmark. For an image captioning model $p_m(:, \theta)$, we first ran it on the benchmark to get corresponding captions ($c_1, c_2, \dots, c_{5000}$) for the images. After that, using each caption c_t as a query, we computed the conditional probabilities ($p_m(c_t|I_1), p_m(c_t|I_2), \dots, p_m(c_t|I_{5000})$), which were used to get a ranked list of images, denoted by \mathbf{r}_t . Based on all ranked lists, we can compute top- k recalls, which is the fraction of images within top- k positions of their corresponding ranked lists. The top- k recalls are good indicators of how well a model captures the distinctiveness of descriptions.

In this experiment, we compared three different models, including *Neu-*

raltalk2 [50] and *AdaptiveAttention* [78] that are learned by MLE, as well as *AdaptiveAttention* learned by our method. The top- k recalls are listed in Table 3.1, along with overall performances of these models in terms of *Rouge* [71] and *Cider* [113]. These results clearly show that the recalls of self retrieval are positively correlated to the performances of image captioning models in classical captioning metrics. Although most of the models are not explicitly learned to promote distinctiveness, the one with better recalls of self retrieval, which means the generated-captions are more distinctive, performs better in the image captioning evaluation. Such positive correlation clearly demonstrates the significance of *distinctiveness* to captioning performance.

3.4.2 Contrastive Learning

In Contrastive Learning (CL), we learn a target image captioning model $p_m(\cdot; \theta)$ with parameter θ by constraining its behaviors relative to a reference model $p_n(\cdot; \phi)$ with parameter ϕ . The learning procedure requires two sets of data: (1) the observed data X , which is a set of ground-truth image-caption pairs $((c_1, I_1), (c_2, I_2), \dots, (c_{T_m}, I_{T_m}))$, and is readily available in any image captioning dataset, (2) the noise set Y , which contains mismatched pairs $((c_{/1}, I_1), (c_{/2}, I_2), \dots, (c_{/T_n}, I_{T_n}))$, and can be generated by randomly sampling $c_{/t} \in \mathcal{C}_{/I_t}$ for each image I_t , where $\mathcal{C}_{/I_t}$ is the set of all ground-truth captions except captions of image I_t . We refer to X as *positive pairs* while Y as *negative pairs*.

For any pair (c, I) , the target model and the reference model will respectively give their estimated conditional probabilities $p_m(c|I, \theta)$ and $p_n(c|I, \phi)$. We wish that $p_m(c_t|I_t, \theta)$ is greater than $p_n(c_t|I_t, \phi)$ for any positive pair

(c_t, I_t) , and vice versa for any negative pair $(c_{/t}, I_t)$. Following this intuition, our initial attempt was to define $D((c, I); \boldsymbol{\theta}, \boldsymbol{\phi})$, the difference between $p_m(c|I, \boldsymbol{\theta})$ and $p_n(c|I, \boldsymbol{\phi})$, as

$$D((c, I); \boldsymbol{\theta}, \boldsymbol{\phi}) = p_m(c|I, \boldsymbol{\theta}) - p_n(c|I, \boldsymbol{\phi}), \quad (3.5)$$

and set the loss function to be:

$$\mathcal{L}'(\boldsymbol{\theta}; X, Y, \boldsymbol{\phi}) = \sum_{t=1}^{T_m} D((c_t, I_t); \boldsymbol{\theta}, \boldsymbol{\phi}) - \sum_{t=1}^{T_n} D((c_{/t}, I_t); \boldsymbol{\theta}, \boldsymbol{\phi}). \quad (3.6)$$

In practice, this formulation would meet with several difficulties. First, $p_m(c|I, \boldsymbol{\theta})$ and $p_n(c|I, \boldsymbol{\phi})$ are very small ($\sim 1e-8$), which may result in numerical problems. Second, Eq.(3.6) treats easy samples, hard samples, and mistaken samples equally. This, however, is not the most effective way. For example, when $D((c_t, I_t); \boldsymbol{\theta}, \boldsymbol{\phi}) \gg 0$ for some positive pair, further increasing $D((c_t, I_t); \boldsymbol{\theta}, \boldsymbol{\phi})$ is probably not as effective as updating $D((c_{t'}, I_{t'}); \boldsymbol{\theta}, \boldsymbol{\phi})$ for another positive pair, for which $D((c_{t'}, I_{t'}); \boldsymbol{\theta}, \boldsymbol{\phi})$ is much smaller.

To resolve these issues, we adopted an alternative formulation inspired by NCE (Section 3.3), where we replace the difference function $D((c, I); \boldsymbol{\theta}, \boldsymbol{\phi})$ with a log-ratio function $G((c, I); \boldsymbol{\theta}, \boldsymbol{\phi})$:

$$G((c, I); \boldsymbol{\theta}, \boldsymbol{\phi}) = \ln p_m(c|I, \boldsymbol{\theta}) - \ln p_n(c|I, \boldsymbol{\phi}), \quad (3.7)$$

and further use a logistic function r_ν (Eq.(3.3)) after $G((c, I); \boldsymbol{\theta}, \boldsymbol{\phi})$ to saturate the influence of easy samples. Following the notations in NCE, we

let $\nu = T_n/T_m$, and turn $D((c, I); \boldsymbol{\theta}, \boldsymbol{\phi})$ into:

$$h((c, I); \boldsymbol{\theta}, \boldsymbol{\phi}) = r_\nu(G((c, I); \boldsymbol{\theta}, \boldsymbol{\phi})). \quad (3.8)$$

Note that $h((c, I); \boldsymbol{\theta}, \boldsymbol{\phi}) \in (0, 1)$. Then, we define our updated loss function as:

$$\mathcal{L}(\boldsymbol{\theta}; X, Y, \boldsymbol{\phi}) = \sum_{t=1}^{T_m} \ln[h((c_t, I_t); \boldsymbol{\theta}, \boldsymbol{\phi})] + \sum_{t=1}^{T_n} \ln[1 - h((c/t, I_t); \boldsymbol{\theta}, \boldsymbol{\phi})]. \quad (3.9)$$

For the setting of $\nu = T_n/T_m$, we choose $\nu = 1$, *i.e.* $T_n = T_m$, to ensure balanced influences from both positive and negative pairs. This setting consistently yields good performance in our experiments. Furthermore, we copy X for K times and sample K different Y s, in order to involve more diverse negative pairs without overfitted to them. In practice we found $K = 5$ is sufficient to make the learning stable. Finally, our objective function is defined to be

$$J(\boldsymbol{\theta}) = \frac{1}{K} \frac{1}{T_m} \sum_{k=1}^K \mathcal{L}(\boldsymbol{\theta}; X, Y_k, \boldsymbol{\phi}). \quad (3.10)$$

Note that $J(\boldsymbol{\theta})$ attains its upper bound 0 if positive and negative pairs can be perfectly distinguished, namely, for all t , $h((c_t, I_t); \boldsymbol{\theta}, \boldsymbol{\phi}) = 1$ and $h((c/t, I_t); \boldsymbol{\theta}, \boldsymbol{\phi}) = 0$. In this case, $G((c_t, I_t); \boldsymbol{\theta}, \boldsymbol{\phi}) \rightarrow \infty$ and $G((c/t, I_t); \boldsymbol{\theta}, \boldsymbol{\phi}) \rightarrow -\infty$, which indicates the target model will give higher probability $p(c_t|I_t)$ and lower probability $p(c/t|I_t)$, compared to the reference model. Towards this goal, the learning process would encourage *distinctiveness* by suppressing negative pairs, while maintaining the overall performance by maximizing the

probability values on positive pairs.

3.4.3 Discussion

Maximum Likelihood Estimation (MLE) is a popular learning method in the area of image captioning [115, 122, 78]. The objective of MLE is to maximize *only* the probabilities of ground-truth image-caption pairs, which may lead to some issues [16], including high resemblance in generated captions. While in CL, the probabilities of ground-truth pairs are *indirectly* ensured by the positive constraint (the first term in Eq.(3.9)), and the negative constraint (the second term in Eq.(3.9)) suppresses the probabilities of mismatched pairs, forcing the target model to also learn from distinctiveness.

Generative Adversarial Network (GAN) [16] is a similar learning method that involves an auxiliary model. However, in GAN the auxiliary model and the target model follow two *opposite* goals, while in CL the auxiliary model and the target model are models in the same track. Moreover, in CL the auxiliary model is stable across the learning procedure, while itself needs careful learning in GAN.

It's worth noting that although our CL method bears certain level of resemblance with Noise Contrastive Estimation (NCE) [39]. The motivation and the actual technical formulation of CL and NCE are essentially different. For example, in NCE the logistic function is a result of computing posterior probabilities, while in CL it is explicitly introduced to saturate the influence of easy samples.

As CL requires only $p_m(c|I)$ and $p_n(c|I)$, the choices of the target model and the reference model can range from models based on LSTMs [42] to

models in other formats, such as MRFs [25] and memory-networks [85]. On the other hand, although in CL, the reference model is usually fixed across the learning procedure, one can replace the reference model with the latest target model periodically. The reasons are (1) $\nabla J(\boldsymbol{\theta}) \neq \mathbf{0}$ when the target model and the reference model are identical, (2) latest target model is usually stronger than the reference model, (3) and a stronger reference model can provide stronger bounds and lead to a stronger target model.

3.5 Experiment

3.5.1 Datasets

We use two large scale datasets to test our contrastive learning method. The first dataset is MSCOCO [73], which contains 122,585 images for training and validation. Each image in MSCOCO has 5 human annotated captions. Following splits in [78], we reserved 2,000 images for validation. A more challenging dataset, InstaPIC-1.1M [85], is used as the second dataset, which contains 648,761 images for training, and 5,000 images for testing. The images and their ground-truth captions are acquired from Instagram, where people post images with related descriptions. Each image in InstaPIC-1.1M is paired with 1 caption. This dataset is challenging, as its captions are natural posts with varying formats. In practice, we reserved 2,000 images from the training set for validation.

On both datasets, non-alphabet characters except emojis are removed, and alphabet characters are converted to lowercases. Words and emojis that appeared less than 5 times are replaced with *UNK*. And all captions are

COCO Online Testing Server C5					
Method	BLEU-3	BLEU-4	METEOR	ROUGE.L	CIDEr
Google NIC [115]	0.407	0.309	0.254	0.530	0.943
Hard-Attention[122]	0.383	0.277	0.241	0.516	0.865
AdaptiveAttention [78]	0.429	0.323	0.258	0.541	1.001
AdaptiveAttention + CL	0.436	0.326	0.260	0.544	1.010
PG-BCMR [76]	0.445	0.332	0.257	0.550	1.013
ATT-FCN [†] [127]	0.424	0.316	0.250	0.535	0.943
MSM [†] [126]	0.436	0.330	0.256	0.542	0.984
AdaptiveAttention [†] [78]	0.443	0.335	0.264	0.550	1.037
Att2in [†] [95]	-	0.344	0.268	0.559	1.123
COCO Online Testing Server C40					
Method	BLEU-3	BLEU-4	METEOR	ROUGE.L	CIDEr
Google NIC [115]	0.694	0.587	0.346	0.682	0.946
Hard-Attention [122]	0.658	0.537	0.322	0.654	0.893
AdaptiveAttention [78]	0.717	0.607	0.347	0.689	1.004
AdaptiveAttention + CL	0.728	0.617	0.350	0.695	1.029
PG-BCMR [76]	-	-	-	-	-
ATT-FCN [†] [127]	0.709	0.599	0.335	0.682	0.958
MSM [†] [126]	0.740	0.632	0.350	0.700	1.003
AdaptiveAttention [†] [78]	0.740	0.633	0.359	0.706	1.051
Att2in [†] [95]	-	-	-	-	-

Table 3.2: This table lists published results of state-of-the-art image captioning models on the online COCO testing server. † indicates ensemble model. ”-” indicates not reported. In this table, CL improves the base model (AdaptiveAttention [78]) to gain the best results among all single models on C40.

truncated to have at most 18 words and emojis. As a result, we obtained a vocabulary of size 9,567 on MSCOCO, and a vocabulary of size 22,886 on InstaPIC-1.1M.

3.5.2 Settings

To study the generalization ability of proposed CL method, we tested it on two different image captioning models, namely **Neuraltalk2** [50] and **AdaptiveAttention** [78]. Both models are based on *encoder-and-decoder*

Method	BLEU-3	BLEU-4	METEOR	ROUGE.L	CIDEr
Google NIC [115]	0.007	0.003	0.038	0.081	0.004
Hard-Attention [122]	0.000	0.000	0.026	0.140	0.049
CSMN [85]	0.015	0.008	0.037	0.120	0.133
AdaptiveAttention [78]	0.011	0.005	0.029	0.093	0.126
AdaptiveAttention + CL	0.013	0.006	0.032	0.101	0.144

Table 3.3: This table lists results of different models on the test split of InstaPIC-1.1M [85], where CL improves the base model (AdaptiveAttention [78]) by significant margins, achieving the best result on Cider.

[115], where no attention mechanism is used in the former, and an adaptive attention component is used in the latter.

For both models, we have pretrained them by MLE, and use the pretrain checkpoints as initializations. In all experiments except for the experiment on model choices, we choose the same model and use the same initialization for target model and reference model. In all our experiments, we fixed the learning rate to be $1e-6$ for all components, and used Adam optimizer. Seven evaluation metrics have been selected to compare the performances of different models, including Bleu-1,2,3,4 [84], Meteor [66], Rouge [71] and Cider [113]. All experiments for ablation studies are conducted on the validation set of MSCOCO.

3.5.3 Results

Overall Results We compared our best model (*AdaptiveAttention* [78] learned by CL) with state-of-the-art models on two datasets. On MSCOCO, we submitted the results to the online COCO testing server. The results along with other published results are listed in Table 3.2. Compared to MLE-learned *AdaptiveAttention*, CL improves the performance of it by signif-

				
AA	Two people on a tennis court playing tennis	A fighter jet flying through a blue sky	A row of boats on a river near a river	A bathroom with a toilet and a sink
AA + CL	Two tennis players shaking hands on a tennis court	A fighter jet flying over a lush green field	A row of boats docked in a river	A bathroom with a red toilet and red walls
				
AA	Three clocks are mounted to the side of a building	Two people on a yellow yellow and yellow motorcycle	A baseball player pitching a ball on top of a field	A bunch of lights hanging from a ceiling
AA + CL	Three three clocks with three different time zones	Two people riding a yellow motorcycle in a forest	A baseball game in progress with pitcher throwing the ball	A bunch of baseballs bats hanging from a ceiling

Figure 3.2: This figure illustrates several images with captions generated by different models, where *AA* represents AdaptiveAttention [78] learned by MLE, and *AA + CL* represents the same model learned by CL. Compared to *AA*, *AA + CL* generated more distinctive captions for these images.

icant margins across all metrics. While most of state-of-the-art results are achieved by ensembling multiple models, our improved *AdaptiveAttention* gains competitive results as a *single* model. Specifically, on Cider, CL improves *AdaptiveAttention* from 1.003 to 1.029, which is the best single-model result on C40 among all published ones. In terms of Cider, if we use MLE, we need to combine 5 models to get 4.5% boost on C40 for *AdaptiveAttention*. Using CL, we improve the performance by 2.5% with just a single model. On InstaPIC-1.1M, CL improves the performance of *AdaptiveAttention* by 14% in terms of Cider, which is the state-of-the-art. Some qualitative results are shown in Figure 3.2. It’s worth noting that the proposed learning method can be used with stronger base models to obtain better results without any modification.

Compare Learning Methods Using *AdaptiveAttention* learned by MLE as base model and initialization, we compared our CL with similar learn-

Method	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr
AdaptiveAttention [78] (Base)	0.433	0.327	0.260	0.540	1.042
Base + IL [112]	0.408	0.307	0.253	0.530	1.004
Base + GAN [16]	0.290	0.190	0.212	0.458	0.700
Base + CL(P)	0.437	0.334	0.262	0.545	1.059
Base + CL(N)	0.299	0.212	0.246	0.479	0.603
Base + CL(Full)	0.460	0.353	0.271	0.559	1.142

Table 3.4: This table lists results of a model learned by different methods. The best result is obtained by the one learned with full CL, containing both the positive constraint and negative constraint.

ing methods, including **CL(P)** and **CL(N)** that respectively contains only the positive constraint and the negative constraint in CL. We also compared with **IL** [112], and **GAN** [16]. The results on MSCOCO are listed in Table 3.4, where (1) among IL, CL and GAN, CL improves performance of the base model, while both IL and GAN decrease the results. This indicates the trade-off between learning distinctiveness and maintaining overall performance is not well settled in IL and GAN. (2) comparing models learned by CL(P), CL(N) and CL, we found using the positive constraint or the negative constraint alone is not sufficient, as only one source of guidance is provided. While CL(P) gives the base model lower improvement than full CL, CL(N) downgrades the base model, indicating overfits on distinctiveness. Combining CL(P) and CL(N), CL is able to encourage distinctiveness while also emphasizing on overall performance, resulting in largest improvements on all metrics.

Compare Model Choices To study the generalization ability of CL, *AdaptiveAttention* and *Neuraltalk2* are respectively chosen as both the target and the reference in CL. In addition, *AdaptiveAttention* learned by MLE, as a better model, is chosen to be the reference, for *Neuraltalk2*. The results

Target Model	Reference	BLEU-3	BLEU-4	METEOR	ROUGE_L	CIDEr
NT	-	0.389	0.291	0.238	0.516	0.882
NT	NT	0.399	0.300	0.242	0.524	0.905
NT	AA	0.411	0.311	0.249	0.533	0.956
AA	-	0.433	0.327	0.260	0.540	1.042
AA	AA	0.460	0.353	0.271	0.559	1.142

Table 3.5: This table lists results of different model choices on MSCOCO. In this table, NT represents *Neuraltalk2* [50], and AA represents *AdaptiveAttention* [78]. ”-” indicates the target model is learned using MLE.

Run	BLEU-3	BLEU-4	METEOR	ROUGE_L	CIDEr
0	0.433	0.327	0.260	0.540	1.042
1	0.460	0.353	0.271	0.559	1.142
2	0.460	0.353	0.272	0.559	1.142

Table 3.6: This table lists results of periodical replacement of the reference in CL. In run 0, the model is learned by MLE, which are used as both the target and the reference in run 1. In run 2, the reference is replaced with the best target in run 1.

are listed in Table 3.5, where compared to models learned by MLE, both *AdaptiveAttention* and *Neuraltalk2* are improved after learning using CL. For example, on Cider, *AdaptiveAttention* improves from 1.042 to 1.142, and *Neuraltalk2* improves from 0.882 to 0.905. Moreover, by using a stronger model, *AdaptiveAttention*, as the reference, *Neuraltalk2* improves further from 0.905 to 0.956, which indicates stronger references empirically provide tighter bounds on both the positive constraint and the negative constraint.

Reference Replacement As discussed in Section 3.4.3, one can periodically replace the reference with latest best target model, to further improve the performance. In our study, using *AdaptiveAttention* learned by MLE as a start, each run we fix the reference model until the target saturates its performance on the validation set, then we replace the reference with latest best

target model and rerun the learning. As listed in Table 3.6, in second run, the relative improvements of the target model is incremental, compared to its improvement in the first run. Therefore, when learning a model using CL, with a sufficiently strong reference, the improvement is usually saturated in the first run, and there is no need, in terms of overall performance, to replace the reference multiple times.

Chapter 4

Captioning Models with 2D States

4.1 Introduction

Image captioning, a task of generating short descriptions for given images, has received increasing attention in recent years. Latest works on this task [115, 122, 95, 78] mostly adopt the encoder-decoder paradigm, where a recurrent neural network (RNN) or one of its variants, *e.g.* GRU [13] and LSTM [42], is used for generating the captions. Specifically, the RNN maintains a series of *latent states*. At each step, it takes the visual features together with the preceding word as input, updates the latent state, then estimates the conditional probability of the next word. Here, the latent states serve as pivots that connect between the visual and the linguistic domains.

Following the standard practice in language models [13, 35], existing captioning models usually formulate the latent states as *vectors* and the connections between them as fully-connected transforms. Whereas this is a natural

choice for purely linguistic tasks, it becomes a question when the visual domain comes into play, *e.g.* in the task of image captioning.

Along with the rise of deep learning, convolutional neural networks (CNN) have become the dominant models for many computer vision tasks [40, 100]. *Convolution* has a distinctive property, namely *spatial locality*, *i.e.* each output element corresponds to a local region in the input. This property allows the spatial structures to be maintained by the feature maps across layers. The significance of spatial locality for vision tasks have been repeatedly demonstrated in previous work [40, 5, 44, 98, 70].

Image captioning is a task that needs to bridge both the linguistic and the visual domains. Thus for this task, it is important to capture and preserve properties of the visual content in the latent states. This motivates us to explore an alternative formulation for image captioning, namely representing the latent states with 2D maps and connecting them via convolutions. As opposed to the standard formulation, this variant is capable of preserving spatial locality, and therefore it may strengthen the role of visual structures in the process of caption generation.

We compared both formulations, namely the standard one with vector states and the alternative one that uses 2D states, which we refer to as *RNN-2DS*. Our study shows: (1) The spatial structures significantly impact the captioning process. Editing the latent states, *e.g.* suppressing certain regions in the states, can lead to substantially different captions. (2) Preserving the spatial structures in the latent states is beneficial for captioning. On two public datasets, MSCOCO [73] and Flickr30k [128], RNN-2DS achieves notable performance gain consistently across different settings. In

particular, a simple RNN-2DS without gating functions already outperforms more sophisticated networks with vector states, *e.g.* LSTM. Using 2D states in combination with more advanced cells, *e.g.* GRU, can further boost the performance. (3) Using 2D states makes the captioning process amenable to visual interpretation. Specifically, we take advantage of the spatial locality and develop a simple yet effective way to identify the connections between latent states and visual regions. This enables us to visualize the dynamics of the states as a caption is being generated, as well as the connections between the visual domain and the linguistic domain.

In summary, our contributions mainly lie in three aspects. First, we rethink the form of latent states in image captioning models, for which existing work simply follows the standard practice and adopts the vectorized representations. To our best knowledge, this is the first study that systematically explores two dimensional states in the context of image captioning. Second, our study challenges the prevalent practice, which reveals the significance of spatial locality in image captioning and suggests that the formulation with 2D states and convolution is more effective. Third, leveraging the spatial locality of the alternative formulation, we develop a simple method that can visualize the dynamics of the latent states in the decoding process.

4.2 Related Work

Image Captioning. Image captioning has been an active research topic in computer vision. Early techniques mainly rely on detection results. Kulkarni *et al* [60] proposed to first detect visual concepts, and then generate captions by filling sentence templates. Farhadi *et al* [25] proposed to generate captions

for a given image by retrieving from training captions based on detected concepts.

In recent years, the methods based on neural networks are gaining ground. Particularly, the encoder-decoder paradigm [115], which uses a CNN [106] to encode visual features and then uses an LSTM net [42] to decode them into a caption, was shown to outperform classical techniques and has been widely adopted. Along with this direction, Xu *et al* [122] proposed to use a dynamic attention map to guide the decoding process. Yao *et al* [126] additionally incorporate visual attributes detected from the images, obtaining further improvement. While achieving significant progress, all these methods rely on *vectors* to encode visual features and to represent latent states.

Multi-dimensional RNN. Existing works that aim at extending RNN to more dimensions roughly fall into three categories:

(1) RNNs are applied on *multi-dimensional grids*, *e.g.* the 2D grid of pixels, via recurrent connections along different dimensions [36, 136]. Such extensions have been used in image generation [119] and CAPTCHA recognition [99].

(2) Latent states of RNN cells are stacked across multiple steps to form feature maps. This formulation is usually used to capture temporal statistics, *e.g.* those in language processing [116, 28] and audio processing [53]. For both categories above, the latent states are still represented by *1D vectors*. Hence, they are essentially different from this work.

(3) Latent states themselves are represented as multi-dimensional arrays. The RNN-2DS studied in this paper belongs to the third category, where latent states are represented as 2D feature maps. The idea of extending

RNN with 2D states has been explored in various vision problems, such as rainfall prediction [120], super-resolution [44], instance segmentation [98], and action recognition [70]. It is worth noting that all these works focused on tackling visual tasks, where both the inputs and the outputs are in 2D forms. To our best knowledge, this is the first work that studies recurrent networks with 2D states in image captioning. A key contribution of this work is that it reveals the significance of 2D states in connecting the visual and the linguistic domains.

Interpretation. There are studies to analyze recurrent networks. Karpathy *et al* [51] try to interpret the latent states of conventional LSTM models for natural language understanding. Similar studies have been conducted by Ding *et al* [21] for neural machine translation. However, these studies focused on linguistic analysis, while our study tries to identify the connections between linguistic and visual domains by leveraging the spatial locality of the 2D states.

Our visualization method on 2D latent states also differs from the attention module [122] fundamentally, in both theory and implementation. (1) Attention is a *mechanism* specifically designed to guide the focus of a model, while the 2D states are a form of *representation*. (2) Attention is usually implemented as a sub-network. In our work, the 2D states by themselves do not introduce any attention mechanism. The visualization method is mainly for the purpose of interpretation, which helps us better understand the internal dynamics of the decoding process. To our best knowledge, this is accomplished for the first time for image captioning.

4.3 Formulations

To begin with, we review the encoder-decoder framework [115] which represents latent states as 1D vectors. Subsequently, we reformulate the latent states as multi-channel 2D feature maps for this framework. These formulations are the basis for our comparative study.

4.3.1 Encoder-Decoder for Image Captioning

The encoder-decoder framework generates a caption for a given image in two stages, namely *encoding* and *decoding*. Specifically, given an image I , it first encodes the image into a feature vector \mathbf{v} , with a *Convolutional Neural Network (CNN)*, such as VGGNet [106] or ResNet [40]. The feature vector \mathbf{v} is then fed to a *Recurrent Neural Network (RNN)* and decoded into a sequence of words (w_1, \dots, w_T) . For decoding, the RNN implements a recurrent process driven by latent states, which generates the caption through multiple steps, each yielding a word. Specifically, it maintains a set of latent states, represented by a vector \mathbf{h}_t that would be updated along the way. The computational procedure can be expressed by the formulas below:

$$\mathbf{h}_0 = \mathbf{0}, \quad \mathbf{h}_t = g(\mathbf{h}_{t-1}, \mathbf{x}_t, \mathbf{I}), \quad (4.1)$$

$$\mathbf{p}_{t|1:t-1} = \text{Softmax}(\mathbf{W}_p \mathbf{h}_t), \quad (4.2)$$

$$w_t \sim \mathbf{p}_{t|1:t-1}. \quad (4.3)$$

The procedure can be explained as follows. First, the latent state \mathbf{h}_0 is initialized to be zeros. At the t -th step, \mathbf{h}_t is updated by an RNN cell g , which takes three inputs: the previous state \mathbf{h}_{t-1} , the word produced at

the preceding step (represented by an embedded vector \mathbf{x}_t), and the visual feature \mathbf{v} . Here, the cell function g can take a simple form:

$$g(\mathbf{h}, \mathbf{x}, \mathbf{v}) = \tanh(\mathbf{W}_h \mathbf{h} + \mathbf{W}_x \mathbf{x} + \mathbf{W}_v \mathbf{v}). \quad (4.4)$$

More sophisticated cells, such as GRU [13] and LSTM [42], are also increasingly adopted in practice. To produce the word w_t , the latent state \mathbf{h}_t will be transformed into a probability vector $\mathbf{p}_{t|1:t-1}$ via a fully-connected linear transform $\mathbf{W}_p \mathbf{h}_t$ followed by a softmax function. Here, $\mathbf{p}_{t|1:t-1}$ can be considered as the probabilities of w_t conditioned on previous states.

Despite the differences in their architectures, all existing RNN-based captioning models represent latent states as *vectors* without explicitly preserving the spatial structures. In what follows, we will discuss the alternative choice that represents latent states as 2D multi-channel feature maps.

4.3.2 From 1D to 2D

From a technical standpoint, a natural way to maintain spatial structures in latent states is to formulate them as 2D maps and employ convolutions for state transitions, which we refer to as RNN-2DS.

Specifically, as shown in Figure 4.1, the visual feature \mathbf{V} , the latent state \mathbf{H}_t , and the word embedding \mathbf{X}_t are all represented as 3D tensors of size $C \times H \times W$. Such a tensor can be considered as a multi-channel map, which comprises C channels, each of size $H \times W$. Unlike the normal setting where the visual feature is derived from the activation of a fully-connected layer, \mathbf{V} here is derived from the activation of a convolutional layer that preserves spatial structures. And \mathbf{X}_t is the 2D word embedding for w_{t-1} ,

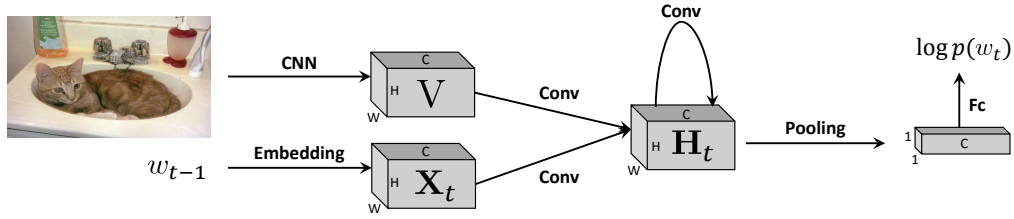


Figure 4.1: The overall structure of the encoder-decoder framework with RNN-2DS. Given an image I , a CNN first turns it into a multi-channel feature map \mathbf{V} that preserves high-level spatial structures. \mathbf{V} will then be fed to an RNN-2DS, where the latent state \mathbf{H}_t is also represented by multi-channel maps and the state transition is via convolution. At each step, the 2D states are transformed into a 1D vectors and then decoded into conditional probabilities of words.

of size $C \times H \times W$. To reduce the number of parameters, we use a lookup table of smaller size $C_x \times H_x \times W_x$ to fetch the raw word embedding, which will be enlarged to $C \times H \times W$ by two convolutional layers¹. With these representations, state updating can then be formulated using *convolutions*. For example, Eq.(4.4) can be converted into the following form:

$$\mathbf{H}_t = \text{relu}(\mathbf{K}_h \circledast \mathbf{H}_{t-1} + \mathbf{K}_x \circledast \mathbf{X}_t + \mathbf{K}_v \circledast \mathbf{V}). \quad (4.5)$$

Here, \circledast denotes the convolution operator, and \mathbf{K}_h , \mathbf{K}_x , and \mathbf{K}_v are convolution kernels of size $C \times C \times H_k \times W_k$. It is worth stressing that the modification presented above is very flexible and can easily incorporate more

¹In our experiments, the raw word embedding is of size $4 \times 15 \times 15$, and is scaled up to match the size of latent states via two convolutional layers respectively with kernel sizes $32 \times 4 \times 5 \times 5$ and $C \times 32 \times 5 \times 5$.

sophisticated cells. For example, the original updating formulas of GRU are

$$\begin{aligned}
\mathbf{r}_t &= \sigma(\mathbf{W}_{rh}\mathbf{h}_{t-1} + \mathbf{W}_{rx}\mathbf{x}_t + \mathbf{W}_{rv}\mathbf{v}), \\
\mathbf{z}_t &= \sigma(\mathbf{W}_{zh}\mathbf{h}_{t-1} + \mathbf{W}_{zx}\mathbf{x}_t + \mathbf{W}_{zv}\mathbf{v}), \\
\tilde{\mathbf{h}}_t &= \tanh(\mathbf{r}_t \circ (\mathbf{W}_{hh}\mathbf{h}_{t-1}) + \mathbf{W}_{hx}\mathbf{x}_t + \mathbf{W}_{hv}\mathbf{v}), \\
\mathbf{h}_t &= \mathbf{z}_t \circ \mathbf{h}_{t-1} + (1 - \mathbf{z}_t) \circ \tilde{\mathbf{h}}_t,
\end{aligned} \tag{4.6}$$

where σ is the sigmoid function, and \circ is the element-wise multiplication operator. In a similar way, we can convert them to the 2D form as

$$\begin{aligned}
\mathbf{R}_t &= \sigma(\mathbf{K}_{rh} \otimes \mathbf{H}_{t-1} + \mathbf{K}_{rx} \otimes \mathbf{X}_t + \mathbf{K}_{rv} \otimes \mathbf{V}), \\
\mathbf{Z}_t &= \sigma(\mathbf{K}_{zh} \otimes \mathbf{H}_{t-1} + \mathbf{K}_{zx} \otimes \mathbf{X}_t + \mathbf{K}_{zv} \otimes \mathbf{V}), \\
\tilde{\mathbf{H}}_t &= \text{relu}(\mathbf{R}_t \circ (\mathbf{K}_{hh} \otimes \mathbf{H}_{t-1}) + \mathbf{K}_{hx} \otimes \mathbf{X}_t + \mathbf{K}_{hv} \otimes \mathbf{V}), \\
\mathbf{H}_t &= \mathbf{Z}_t \circ \mathbf{H}_{t-1} + (1 - \mathbf{Z}_t) \circ \tilde{\mathbf{H}}_t.
\end{aligned} \tag{4.7}$$

Given the latent states \mathbf{H}_t , the word w_t can be generated as follows. First, we compress \mathbf{H}_t (of size $C \times H \times W$) into a C -dimensional vector \mathbf{h}_t by mean pooling across spatial dimensions. Then, we transform \mathbf{h}_t into a probability vector $\mathbf{p}_{t|1:t-1}$ and draw w_t therefrom, following Eq.(4.2) and Eq.(4.3). Note that the pooling operation could be replaced with more sophisticated modules, such as an attention module, to summarize the information from all locations for word prediction. We choose the pooling operation as it adds zero extra parameters, which makes the comparison between 1D and 2D states fair.

Since this reformulation is generic, besides the encoder-decoder frame-

work, it can be readily extended to other captioning models that adopt RNNs as the language module, *e.g.* Att2in [95] and Review Net [123].

4.4 Qualitative Studies on 2D States

Thanks to the preserved spatial locality, the use of 2D states makes the framework amenable to some qualitative analysis. Taking advantage of this, we present three studies in this section: (1) We manipulate the 2D states and investigate how it impacts the generated captions. The results of this study would corroborate the statement that 2D states help to preserve spatial structures. (2) Leveraging the spatial locality, we identify the associations between the activations of latent states and certain subregions of the input image. Based on the dynamic associations between state activations and the corresponding subregions, we can visually reveal the internal dynamics of the decoding process. (3) Through latent states we also interpret the connections between the visual and the linguistic domains.

4.4.1 State Manipulation

We study how the spatial structures of the 2D latent states influence the resultant captions by controlling the accessible parts of the latent states.

As discussed in Section 4.3.2, the prediction at t -th step is based on \mathbf{h}_t , which is pooled from \mathbf{H}_t across H and W . In other words, \mathbf{h}_t summarizes the information from the entire area of \mathbf{H}_t . In this experiment, we replace the original region $(1, 1, H, W)$ with a subregion between the corners (x_1, y_1)

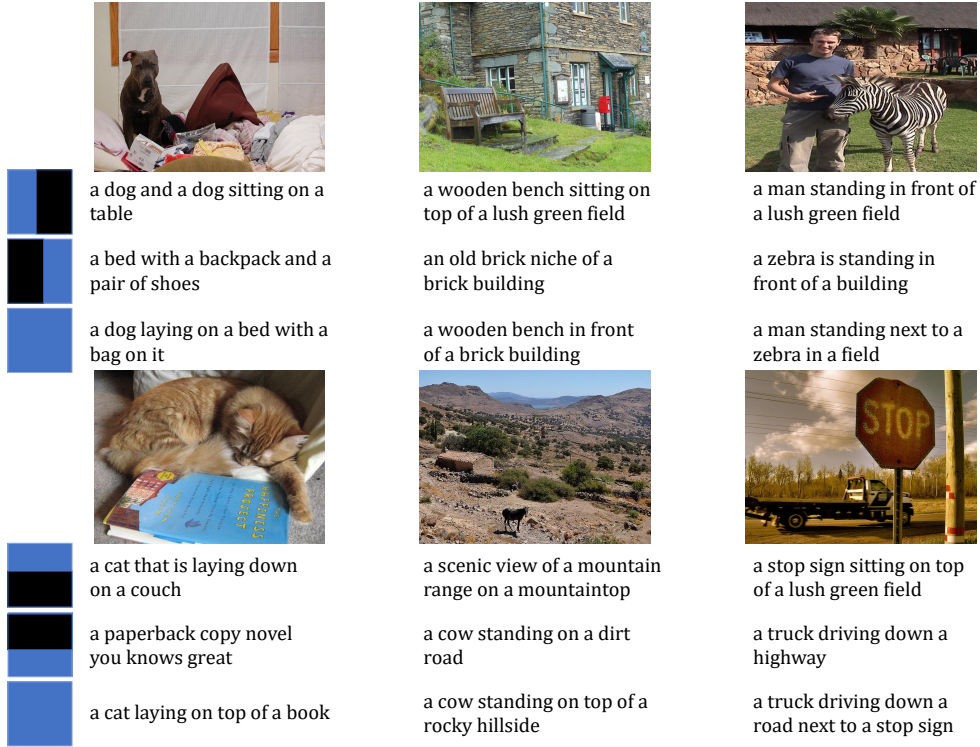


Figure 4.2: This figure lists several images with generated captions relying on various parts of RNN-2DS’s states. The accessible part is marked with blue color in each case.

and (x_2, y_2) to get a modified summarizing vector \mathbf{h}'_t as

$$\mathbf{h}'_t = \frac{1}{(y_2 - y_1 + 1)(x_2 - x_1 + 1)} \sum_{i=y_1}^{y_2} \sum_{j=x_1}^{x_2} \mathbf{H}_t|_{(i,j)}. \quad (4.8)$$

Here, \mathbf{h}'_t only captures a subregion of the image, on which the probabilities for the word w_t is computed. We expect that this caption only partially reflects the visual semantics.

Figure 4.2 shows several images together with the captions generated using different subregions of the 2D states. Take the bottom-left image in Figure 4.2 for an instance, when using only the upper half of the latent states,

the decoder generates a caption focusing on the cat, which indeed appears in the upper half of the image. Similarly, using only the lower half of the latent states results in a caption that talks about the book located in the lower half of the image. In other words, depending on a specific subregion of the latent states, a decoder with 2D states tends to generate a caption that conveys the visual content of the corresponding area in the input image. This observation suggests that the 2D latent states do preserve the spatial structures of the input image.

Manipulating latent states differs essentially from the passive data-driven attention module [122] commonly adopted in captioning models. It is a controllable operation, and does not require a specific module to achieve such functionality. With this operation, we can extend a captioning model with 2D states to allow *active* management of the focus, which, for example, can be used to generate multiple complementary sentences for an image. While the attention module can be considered as an automatic manipulation on latent states, the combination of 2D states and the attention mechanism worths exploring in the future work.

4.4.2 Revealing Decoding Dynamics

This study intends to analyze internal dynamics of the decoding process, *i.e.* how the latent states evolve in a series of decoding steps. We believe that it can help us better understand how a caption is generated based on the visual content. The spatial locality of the 2D states allows us to study this in an efficient and effective way.

We use *activated regions* to align the activations of the latent states at

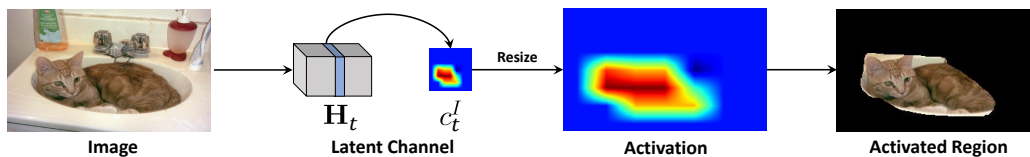


Figure 4.3: This figure shows our procedure of finding the activated region of a latent channel at the t -th step.

different decoding steps with the subregions in the input image. Specifically, we treat the channels of 2D states as the basic units in our study, which are 2D maps of activation values. Given a state channel c at the t -th decoding step, we resize it to the size of the input image I via bicubic interpolation. The pixel locations in I whose corresponding interpolated activations are above a certain threshold² are considered to be *activated*. The collection of all such pixel locations is referred to as the *activated region* for the state channel c at the t -th decoding step, as shown in Figure 4.3.

With activated regions computed respectively at different decoding steps for one state channel, we may visually reveal the internal dynamics of the decoding process at that channel. Figure 4.4 shows several images and their generated captions, along with the activated regions of some channels following the decoding processes. These channels are selected as they are associated with nouns in the generated captions, which we will introduce in the next section. Via this study we found that (1) The activated regions of channels often capture salient visual entities in the image, and also reflect the surrounding context occasionally. (2) During a decoding process, different channels have different dynamics. For a channel associated with a noun, the activated regions of its associated channel become significant as the decoding process approaches the point where the noun is produced, and the channel

²See Section 4.6 for the complete algorithm.

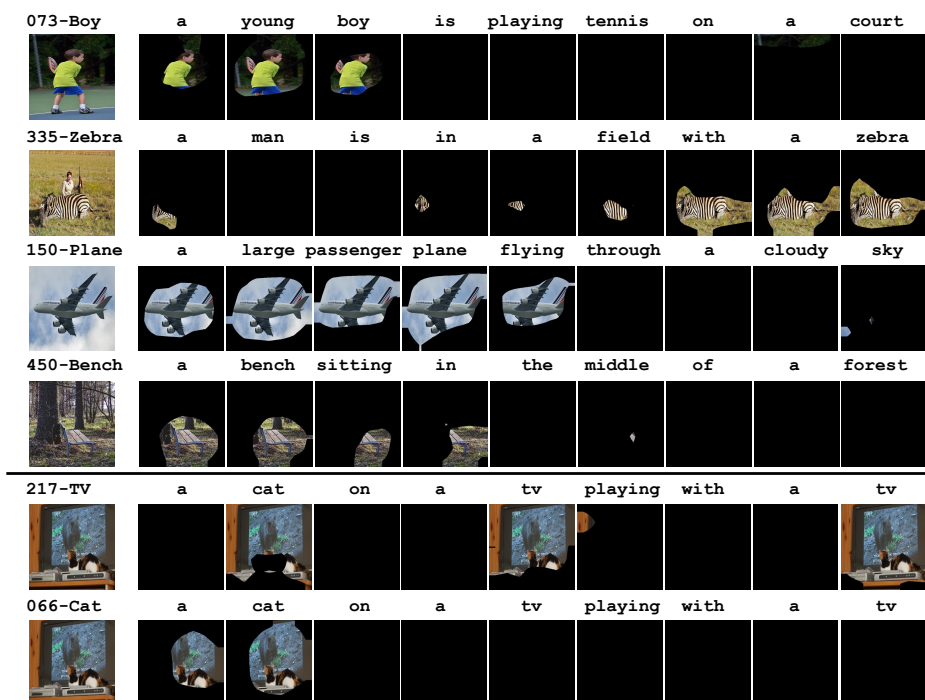


Figure 4.4: This figure shows the changes of several channels, in terms of the activated regions, during the decoding processes. On the last two cases, changes of two channels in the same decoding process are shown and compared. (Best viewed in high resolution)

becomes deactivated afterwards.

The revealed dynamics can help us better understand the decoding process, which also point out some directions for future study. For instance, in Figure 4.4, the visual semantics are distributed to different channels, and the decoder moves its focus from one channel to another. The mechanism that triggers such movements remains needed to be explored.

4.4.3 Connecting Visual and Linguistic Domains

Here we investigate how the visual domain is connected to the linguistic domain. As the latent states serve as pivots that connect both domains,



Figure 4.5: Sample words and their associated channels in $RNN-2DS-(512, 7, 7)$. For each word, 5 activated regions of its associated channel on images that contain this word in the generated captions are shown. The activated regions are chosen at the steps where the words are produced. (Best viewed in high resolution)

we try to use the activations of the latent states to identify the detailed connections.

First, we find the associations between the latent states and the words. Similar to Section 4.4.2, we use state channels as the basic units here, so that we can use the activated regions which connect the latent states to the input image. In Section 4.4.2, we have observed that a channel associated with a certain word is likely to remain active until the word is produced, and its activation level will drop significantly afterwards thus preventing that word from being generated again. Hence, one way to judge whether a channel is associated with a word is to estimate the difference in its level of activations before and after the word is generated. The channel that yields *maximum difference* can be considered as the one associated with the word³.

Words and Associated Channels. For each word in the vocabulary, we could find its associated channel as described above, and study the corresponding activated regions, as shown in Figure 4.5. We found that (1) Only nouns have strong associations with the state channels, which is consistent

³See Section 4.6 for the complete algorithm.

with the fact that spatial locality is highly-related with the visual entities described as nouns. (2) Some channels have multiple associated nouns. For example, *Channel-066* is associated with “*cat*”, “*dog*”, and “*cow*”. This is not surprising – since there are more nouns in the vocabulary than the number of channels, some nouns have to share channels. Here, it is worth noting that the nouns that share a channel tend to be visually relevant. This shows that the latent channels can capture meaningful visual structures. (3) Not all channels have associated words. Some channels may capture abstract notions instead of visual elements. The study of such channels is an interesting direction in the future.

Match of Words and Associated Channels. On top of the activated regions, we could also estimate the match between a word and its associated channel. Specifically, noticing the activated regions visually look like the attention maps in [75], we borrow the measurement of attention correctness from [75], to estimate the match. *Attention correctness* computes the similarity between a human-annotated segmentation mask of a word, and the activated region of its associated channel, at the step the word is produced. The computation is done by summing up the normalized activations within that mask. On MSCOCO [73], we evaluated the attention correctness on 80 nouns that have human-annotated masks. As a result, the averaged attention correctness is 0.316. For reference, following the same setting except for replacing the activated regions with the attention maps, AdaptiveAttention [78], a state-of-the-art captioning model, got a result of 0.213.

Deactivation of Word-Associated Channels. We also verify the match of the found associations between the state channels and the words



Original	a red and red bird perched on a branch	a man getting ready to board a plane	a man standing in front of a fence with a bird	a vase filled with pink and yellow flowers
Deactivate word-associated channel	a red and green leaf filled with lots of fruit	a man standing next to a boarding gate	a man holding a baseball bat over his shoulder	a bouquet of red flowers sitting on a table

Figure 4.6: This figure lists some images with generated captions before and after some word-associated channel being deactivated. The word that associates with the deactivated channel is marked in **red**.

alternatively via an ablation study, where we compare the generated captions with and without the involvement of a certain channel. Specifically, on images that contain the target word w in the generated captions, we re-run the decoding process, in which we deactivate the associated channel of w by clipping its value to zero at all steps, then compare the generated captions with previous ones. As shown in Figure 4.6, deactivating a word-associated channel leads to the miss of the corresponding words in the generated captions, even though the input still contains the visual semantics for those words. This ablation study corroborates the validity of our found associations.

4.5 Comparison on Captioning Performance

In this section, we compare the encoder-decoder framework with 1D states and 2D states. Specifically, we run our studies on MSCOCO [73] and Flickr30k [128], where we at first introduce the settings, followed by the results.

4.5.1 Settings

MSCOCO [73] contains 122,585 images. We follow the splits in [50], using 112,585 images for training, 5,000 for validation, and the remaining 5,000 for testing. Flickr30K [128] contains 31,783 images in total, and we follow splits in [50], which has 1,000 images respectively for validation and testing, and the rest for training. In both datasets, each image comes with 5 ground-truth captions. To obtain a vocabulary, we turn words to lowercase and remove those with non-alphabet characters. Then we replace words that appear less than 6 times with a special token *UNK*, resulting in a vocabulary of size 9,487 for MSCOCO, and 7,000 for Flickr30k. Following the common convention [50], we truncated all ground-truth captions to have at most 18 words.

All captioning methods in our experiments are based on the encoder-decoder paradigm [115]. We use ResNet-152 [40] pretrained on ImageNet [100] as the encoder in all methods. In particular, we take the output of the layer `res5c` as the visual feature \mathbf{V} . We use the combination of the cell type and the state shape to refer to each type of the decoder. *e.g.* *LSTM-1DS-(L)* refers to a standard LSTM-based decoder with latent states of size L , and *GRU-2DS-(C, H, W)* refers to an RNN-2DS decoder with GRU cells as in Eq.(4.7), whose latent states are of size $C \times H \times W$. Moreover, all RNN-2DS models adopt a raw word-embedding of size $4 \times 15 \times 15$, except when a different size is explicitly specified. The convolution kernels \mathbf{K}_h , \mathbf{K}_x , and \mathbf{K}_v share the same size $C \times C \times 3 \times 3$.

The focus of this paper is the representations of latent states. To ensure fair comparison, no additional modules including the attention module

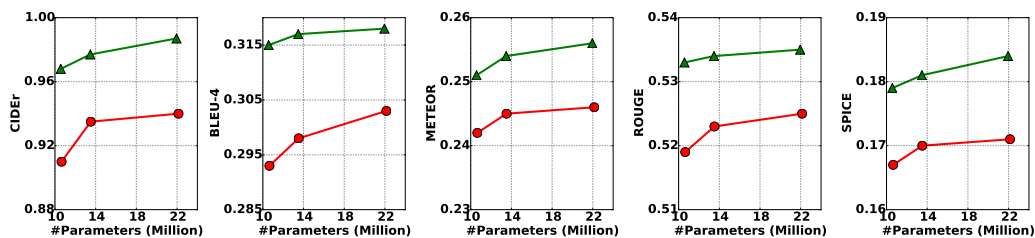


Figure 4.7: The results, in terms of different metrics, obtained using RNN-2DS (green) and LSTM-1DS (red) on the MSCOCO offline test set with similar parameter sizes. Specifically, RNN-2DS of sizes 10.31M, 12.43M and 17.75M have compared to LSTM-1DS of sizes 11.59M, 13.93M and 18.39M.

[122] are added to the methods. Moreover, no other training strategies are utilized, such as the scheduled sampling [7], except for the maximum likelihood objective, where we use the ADAM optimizer [54]. During training, we first fix the CNN encoder and optimize the decoder with learning rate 0.0004 in the first 20 epochs, and then jointly optimize both the encoder and decoder, until the performance on the validation set saturates.

For evaluation, we report the results using metrics including BLEU-4 (B4) [84], METEOR (MT) [66], ROUGE (RG) [71], CIDER (CD) [113], and SPICE (SP) [2].

4.5.2 Comparative Results

First, we compared *RNN-2DS* with *LSTM-1DS*. The former has 2D states with the simplest type of cells while the latter has 1D states with sophisticated LSTM cells. As the capacity of a model is closely related to the number of parameters, to ensure a fair comparison, each config of *RNN-2DS* is compared to an *LSTM-1DS* config *with a similar number of parameters*. In this way, the comparative results will signify the differences in the inherent expressive power of both formulations.

Model		RNN-1DS	GRU-1DS	LSTM-1DS	RNN-2DS	GRU-2DS	LSTM-2DS
#Param		13.58M	13.53M	13.52M	13.48M	17.02M	18.79M
COCO-offline	CD	0.914	0.920	0.935	0.977	1.001	0.994
	B4	0.293	0.295	0.298	0.317	0.323	0.319
	RG	0.520	0.520	0.523	0.534	0.539	0.538
	SP	0.168	0.169	0.170	0.181	0.186	0.187
COCO-online	CD	0.868	0.889	0.904	0.930	0.962	0.958
	B4	0.286	0.291	0.295	0.305	0.316	0.313
	RG	0.515	0.518	0.523	0.527	0.535	0.531
	SP	-	-	-	-	-	-
Flickr30k	CD	0.353	0.360	0.381	0.420	0.438	0.427
	B4	0.195	0.195	0.202	0.217	0.218	0.220
	RG	0.427	0.428	0.437	0.442	0.445	0.444
	SP	0.117	0.117	0.120	0.125	0.131	0.132

Table 4.1: The results obtained using different decoders on the offline and online test sets of MSCOCO, and the test set of Flickr30k.

The resulting curves in terms of different metrics are shown in Figure 4.7, in which we can see that *RNN-2DS* outperforms *LSTM-1DS* consistently, across different parameter sizes and under different metrics. These results show that *RNN-2DS*, with the states that preserve spatial locality, can capture both visual and linguistic information more efficiently.

We also compared different types of decoders with similar numbers of parameters, namely *RNN-1DS*, *GRU-1DS*, *LSTM-1DS*, *RNN-2DS*, *GRU-2DS*, and *LSTM-2DS*. Table 4.1 shows the results of these decoders on both datasets, from which we observe: (1) *RNN-2DS* outperforms *RNN-1DS*, *GRU-1DS*, and *LSTM-1DS*, indicating that embedding latent states in 2D forms is more effective. (2) *GRU-2DS*, which is also based on the proposed formulation but adds several gate functions, surpasses other decoders and yields the best result. This suggests that the techniques developed for conventional RNNs including gate functions and attention modules [122] are

	RNN-1DS	GRU-1DS	LSTM-1DS	RNN-2DS	GRU-2DS
#used_words	485	500	502	951	951
vocabulary	5.11%	5.27%	5.29%	10.02%	10.02%
training data	80.34%	80.47%	80.75%	86.59%	86.58%

Table 4.2: This table lists the number of words used by different methods when generating captions for testing images of MSCOCO [73], with the ratio of them in the vocabulary, as well as the ratio of their samples in the training set.

very likely to benefit RNNs with 2D states as well.

We also compared the usage of vocabulary for different decoders. As shown in Table 4.2, on all 5,000 testing images of MSCOCO, the encoder-decoder framework with 1D states uses only 5% of the words in vocabulary to generate their captions, which accounts for over 80% of the training words. On the contrary, by replacing 1D states with 2D states, the encoder-decoder framework now uses 10% of the words in vocabulary, which is twice the original ratio. Figure 4.8 includes some qualitative samples, in which we can see the captions generated by *LSTM-1DS* rely heavily on the language priors, which sometimes contain the phrases that are not consistent with the visual content but appear frequently in training captions. On the contrary, the sentences from *RNN-2DS* and *GRU-2DS* are more relevant to the visual content.

4.5.3 Ablation Study

Table 4.3 compares the performances obtained with different design choices in *RNN-2DS*, including pooling methods, activation functions, and sizes of word embeddings, kernels and latent states. The results show that mean pooling outperforms max pooling by a significant margin, indicating that information from all locations is significant. The table also shows the best com-









				
LSTM-1DS	a small bird sitting on a tree branch	a person walking down a street with an umbrella	a giraffe standing next to a wooden fence	a cat sitting on a chair in a room
RNN-2DS	a bird perched on a bird feeder	a fire hydrant in front of a building	a giraffe laying down on a dirt ground	a cat sitting on top of a wooden table
GRU-2DS	a bird is sitting on a bird feeder	a fire hydrant is covered in snow in the snow	a giraffe laying on the ground in front of a building	a cat sitting in a bowl on a table
				
LSTM-1DS	a man laying on a bed with a laptop	a cat laying on top of a pair of shoes	two hot dogs with ketchup on a plate	a large elephant standing next to a baby elephant
RNN-2DS	a man laying on a bed with a book	a black cat laying on top of a piece of luggage	a hot dog and french fries on a plate	an elephant standing in a field of grass
GRU-2DS	a man laying in bed reading a book	a black cat laying on top of a black suitcase	a hot dog and french fries are on a plate	an elephant standing in a field of grass

Figure 4.8: This figure shows some qualitative samples of captions generated by different decoders, where words in red indicate they are inconsistent with the image.

bination of modeling choices for RNN-2DS: mean pooling, ReLU, the word embeddings of size $4 \times 15 \times 15$, the kernel of size 3×3 , and the latent states of size $256 \times 7 \times 7$.

4.6 Additional Materials

4.6.1 Activated Regions

For a given image I , the channel c of \mathbf{H}_t , denoted by c_t^I , is a map of size $H \times W$. To obtain the activated region, we first resize c_t^I to the size of I with bicubic interpolation, and then identify the activated pixels by thresholding. In particular, those pixels whose corresponding values in c_t^I are above the

Pooling	Activation	Word-Embedding	Kernel	Latent-State	CD	SP
Mean	ReLU	$4 \times 15 \times 15$	3×3	$256 \times 7 \times 7$	0.977	0.181
-	tanh	-	-	-	0.924	0.174
Max	-	-	-	-	0.850	0.166
-	-	$1 \times 15 \times 15$	-	-	0.965	0.180
-	-	$7 \times 15 \times 15$	-	-	0.951	0.179
-	-	-	1×1	-	0.927	0.177
-	-	-	5×5	-	0.951	0.177
-	-	-	-	$256 \times 5 \times 5$	0.934	0.173
-	-	-	-	$256 \times 11 \times 11$	0.927	0.176

Table 4.3: The results obtained on the MSCOCO offline test set using RNN-2DS with different choices on pooling functions, activation functions, word-embeddings, kernels and latent states. Except for the first row, each row only lists the choice that is different from the first row. ”-” means the same.

threshold $\lambda \cdot v^*$ are considered as *activated*. Here, v^* is the maximum value in the corresponding channel c_t^I over all decoding steps, and λ is a coefficient in $[0, 1]$ that controls the range of the activated regions. In practice, we set $\lambda = 0.2$.

4.6.2 Word-Channel Association

To identify the connections between latent states and words, we devise a metric to measure the degree of *association* between a word w and a channel c , denoted by $s(w, c)$.

The metric is designed following this observation: a channel associated with a certain word is likely to remain active until the word is produced, and its activation level will drop significantly afterwards thus preventing that word from being generated again.

First, we measure the *activation level* of a channel c at the t -th step on a

given image I as the sum of its entries:

$$\eta(c_t^I) = \sum_{i=1}^H \sum_{j=1}^W c_t^I(i, j). \quad (4.9)$$

Then we use A_{t_1, t_2} to denote the activation level averaged over a certain period $[t_1, t_2]$, as

$$A_{t_1, t_2}(c^I) = \frac{1}{t_2 - t_1 + 1} \sum_{j=t_1}^{t_2} \eta(c_j^I). \quad (4.10)$$

Finally, the *association score* $s(w, c)$ is defined to be the difference between the average activation level up to the step where w is produced and the average level afterwards. Such a difference is averaged over all samples that contain the word w . Formally, this can be expressed as:

$$s(w, c) = \frac{1}{|\mathcal{I}(w)|} \sum_{I \in \mathcal{I}(w)} A_{1, t_I^w}(c^I) - A_{t_I^w + 1, T_I}(c^I). \quad (4.11)$$

Here, $\mathcal{I}(w)$ is the set of all images that contain w in their *generated* captions. T_I is the length of the caption for I . t_I^w is the step at which w is produced for I . $A_{1, t_I^w}(c^I)$ and $A_{t_I^w + 1, T_I}(c^I)$ are respectively the average activations *before* and *after* w is produced. Based on the *association score*, for each word w , we could find the *most relevant* channel as $c^* = \operatorname{argmax}_c s(w, c)$.

Chapter 5

Images as Scene-graphs: Detecting Visual Relationships with DR-Net

5.1 Introduction

Images in the real world often involve multiple objects that interact with each other. To understand such images, being able to recognize individual objects is generally not sufficient. The *relationships* among them also contain crucial messages. For example, image captioning, a popular application in computer vision, can generate richer captions based on relationships in addition to objects in the images. Thanks to the advances in deep learning, the past several years witness remarkable progress in several key tasks in computer vision, such as *object recognition* [94], *scene classification* [133], and *attribute detection* [131]. However, visual relationship detection remains

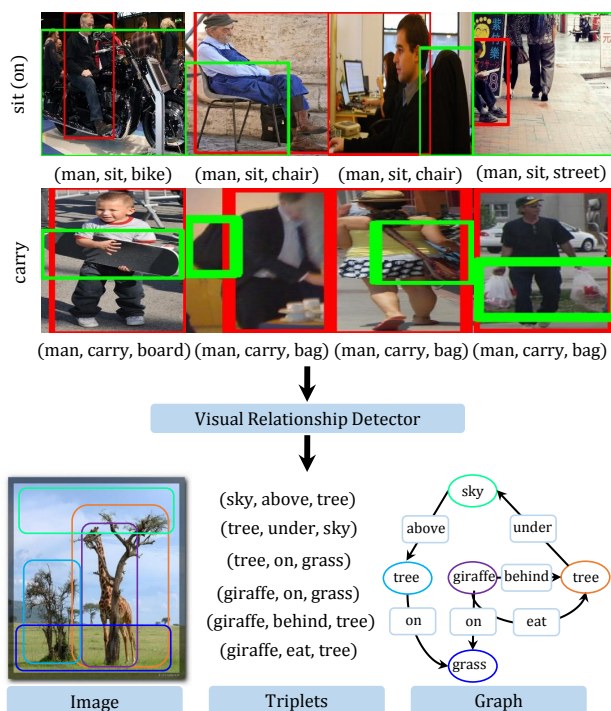


Figure 5.1: Visual relationships widely exist in real-world images. Here are some examples from the VRD [77] dataset, with relationship predicates “*sit*” and “*carry*”. We develop a method that can effectively detect such relationships from a given image. On top of that, a scene graph can be constructed.

a *very difficult* task. On Visual Genome [58], a large dataset designed for structural image understanding, the state-of-the-art can only obtain 11.79% of Recall@50 [77]. This performance is clearly far from being satisfactory.

A natural approach to this problem is to treat it as a classification task. Early attempts [102] used to consider different combinations of objects and relationship predicates (known as *visual phrases*) as different classes. While it may work in a restricted context where the number of possible combinations is moderate, such strategy would be met with a fundamental difficulty in general – an extremely large number of imbalanced classes. As a case in point, Visual Genome [58] contains over 75K distinct visual phrases, and the

number of samples for each phrase ranges from just a handful to over $10K$. Even the most sophisticated classifier would suffer facing such a large and highly imbalanced class space.

An alternative strategy is to consider each type of relationship predicates as a class. Whereas the number of classes is drastically smaller, along with this change also comes with an undesirable implication, namely the substantially increased diversity within each class. To be more specific, phrases with different object categories are considered to be in the same class, as long as they have the same type of relationship predicates. Consequently, the images in each class are highly diverse – some images in the same class may even share nothing in common, *e.g.* “*mountain-near-river*” and “*person-near-dog*”. See Figure 5.1 for an illustration. Our experiments suggest that even with the model capacity of deep networks, handling the intra-class diversity at this level remains very difficult.

In this work, we develop a new framework to tackle the problem of *visual relationship detection*. This framework formulates the prediction output as a triplet in the form of $(subject, predicate, object)$, and jointly infers their class labels by exploiting two kinds of relations among them, namely *spatial configuration* and *statistical dependency*. Such relations are ubiquitous, informative, and more importantly they are often more reliable than visual appearance.

It is worth emphasizing that the formulation of the proposed model is significantly different from previous relational models such as conditional random fields (CRFs) [65]. Particularly, in our formulation, the statistical inference procedure is embedded into a deep neural network called *Deep Re-*

lational Network (DR-Net) via iteration unrolling. The formulation of DR-Net moves beyond the conventional scope, extending the expressive power of Deep Neural Networks (DNNs) to relational modeling. This new way of formulation also allows the model parameters to be learned in a discriminative fashion, using the latest techniques in deep learning. On two large datasets, the proposed framework outperforms not only the classification-based methods but also the CRFs based on deep potentials.

To sum up, the major contributions of this work consist in two aspects: (1) DR-Net, a novel formulation that combines the strengths of statistical models and deep learning; and (2) an effective framework for visual relationship detection which brings the state-of-the-art to a new level.

5.2 Related Work

Over the past decade, there have been a number of studies that explore the use of *visual relationships*. Earlier efforts often focus on *specific* types of relationships, such as positional relations [38, 49, 30, 15, 59] and actions (*i.e.* interactions between objects) [124, 31, 93, 111, 91, 97, 37, 4, 23, 25, 121]. In most of these studies, relationships are usually extracted using simple heuristics or hand-crafted features, and used as an auxiliary components to facilitate other tasks, such as object recognition [29, 107, 61, 14, 64, 103, 90, 27, 101], image classification and retrieval [82, 32], scene understanding and generation [135, 43, 11, 125, 46, 34, 8], as well as text grounding [88, 52, 96]. They are essentially different from our work, which aims to provide a method dedicated to *generic* visual relationship detection. On a unified framework, our method can recognize a wide variety of relationships, such

as relative positions (“*behind*”), actions (“*eat*”), functionals (“*part of*”), and comparisons (“*taller than*”).

Recent years have seen new methods developed specifically for detecting visual relationships. An important family of methods [18, 22, 102] consider each distinct combination of object categories and relationship predicates as a distinct class (often referred to as a *visual phrase*). Such methods would face difficulties in a general context, where the number of such combinations can be very large. An alternative paradigm that considers relationship predicates and object categories separately becomes more popular in recent efforts. Vedantam *et al* [114] presented a study along this line using synthetic clip-arts. This work, however, relies on multiple synthetic attributes that are difficult to obtain from natural images. Fang *et al* [24] proposed to incorporate relationships in an image captioning framework. This work treats object categories and relationship predicates uniformly as words, and does not discuss how to tackle the various challenges in relationship detection.

The method proposed recently by Lu *et al* [77] is the most related. In this method, pairs of detected objects are fed to a classifier, which combines appearance features and a language prior for relationship recognition. Our method differs in two aspects: (1) We exploit both spatial configurations and statistical dependencies among *relationship predicates*, *subjects*, and *objects*, via a Deep Relational Network, instead of simply fusing them as different features. (2) Our framework, from representation learning to relational modeling, is integrated into a single network that is learned in an end-to-end fashion. Experiments show that the proposed framework performs substantially better in all different task settings. For example, on two large datasets, the

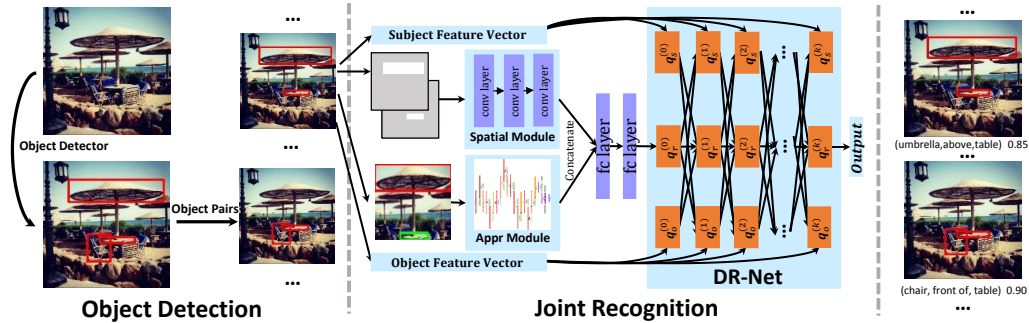


Figure 5.2: The proposed framework for visual relationship detection. Given an image, it first employs an object detector to locate individual objects. Each object also comes with an appearance feature. For each pair of objects, the corresponding local regions and the spatial masks will be extracted, which, together with the appearance features of individual objects, will be fed to the DR-Net. The DR-Net will jointly analyze all aspects and output \mathbf{q}_s , \mathbf{q}_r , and \mathbf{q}_o , the predicted category probabilities for each component of the triplet. Finally, the triplet (s, r, o) will be derived by choosing the most probable categories for each component.

Recall@50 of relationship predicate recognition are respectively raised from 47.9% to 80.8% and from 53.5% to 88.3%.

5.3 Visual Relationship Detection

Visual relationships play a crucial role in image understanding. Whereas a relationship may involve multiple parties in general, many important relationships, including *relative positions* (e.g. “above”) and *actions* (e.g. “ride”) occur between exactly two objects. In this paper, we focus on such relationships. In particular, we follow a widely adopted convention [102, 77] and characterize each visual relationship by a *triplet* in the form of (s, r, o) , e.g. $(girl, on, horse)$ and $(man, eat, apple)$. Here, s , r , and o respectively denote the *subject category*, the *relationship predicate*, and the *object category*. The task is to locate all visual relationships from a given image, and

infer the triplets.

5.3.1 Overall Pipeline

As mentioned, there are two different paradigms for relationship detection: one is to consider each distinct triplet as a different category (also known as *visual phrases* [102]), the other is to recognize each component individually. The former is not particularly suitable for generic applications, due to difficulties like the excessively large number of classes and the imbalance among them. In this work, we adopt the latter paradigm and aim to take its performance to a next level. Particularly, we focus on developing a new method that can effectively capture the rich relations (both *spatial* and *semantic*) among the three components in a triplet and exploit them to improve the prediction accuracy.

As shown in Figure 5.2, the overall pipeline of our framework comprises three stages, as described below.

(1) Object detection. Given an image, we use an object detector to locate a set of candidate objects. In this work, we use Faster RCNN [94] for this purpose. Each candidate object comes with a bounding box and an appearance feature, which will be used in the joint recognition stage for predicting the object category.

(2) Pair filtering. The next step is to produce a set of *object pairs* from the detected objects. With n detected objects, we can form $n(n - 1)$ pairs. We found that a considerable portion of these pairs are *obviously* meaningless and it is unlikely to recognize important relationships therefrom. Hence, we introduce a low-cost neural network to filter out such pairs, so as to reduce

the computational cost of the next stage. This filter takes into account both the spatial configurations (*e.g.* objects too far away are unlikely to be related) and object categories (*e.g.* certain objects are unlikely to form a meaningful relationship).

(3) Joint recognition. Each retained pair of objects will be fed to the *joint recognition* module. Taking into account multiple factors and their relations, this module will produce a triplet as the output.

5.3.2 Joint Recognition

In joint recognition, multiple factors are taken into consideration. These factors are presented in detail below.

(1) Appearance. As mentioned, each detected object comes with an appearance feature, which can be used to infer its category. In addition, the type of the relationship may also be reflected in an image visually. To utilize this information, we extract an appearance feature for each *candidate pair* of objects, by applying a CNN [105, 41] to an *enclosing box*, *i.e.* a bounding box that encompasses both objects with a small margin. The appearance inside the enclosing box captures not only the objects themselves but also the surrounding context, which is often useful when reasoning about the relationships.

(2) Spatial Configurations. The relationship between two objects is also reflected by the spatial configurations between them, *e.g.* their relative positions and relative sizes. Such cues are complementary to the appearance of individual objects, and resilient to photometric variations, *e.g.* the changes in illumination.

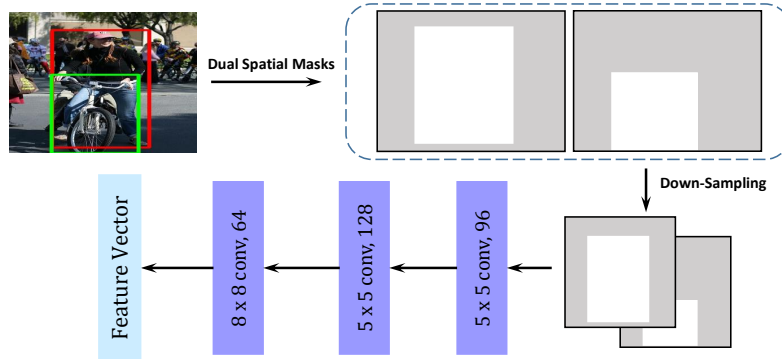


Figure 5.3: This figure illustrates the process of spatial feature vector generation. The structure of our spatial module is also presented in this figure.

To leverage the spatial configurations, we are facing a question: *how to represent it in a computer?* Previous work [49] suggests a list of geometric measurements. While simple, this way may risk missing certain aspects of the configurations. In this work, we instead use *dual spatial masks* as the representation, which comprise two binary masks, one for the subject and the other for the object. The masks are derived from the bounding boxes and may overlap with each other, as shown in Figure 5.3. The masks are down-sampled to the size 32×32 , which we found empirically is a good balance between fidelity and cost. (We have tried mask sizes of 8, 16, 32, 64 and 128, resulting top-1 recalls are 0.47, 0.48, 0.50, 0.51 and 0.51.) The dual spatial masks for each candidate pair will be compressed into a 64-dimensional vector via three convolutional layers.

(3) Statistical Relations. In a triplet (s, r, o) , there exist strong statistical dependencies between the relationship predicate r and the object categories s and o . For example, $(cat, eat, fish)$ is common, while $(fish, eat, cat)$ or $(cat, ride, fish)$ is very unlikely. On Visual Genome, the entropy of the prior distribution $p(r)$ is 2.88, while that of the conditional distribu-

tion $p(r|s, o)$ is 1.21. This difference is a clear evidence of the statistical dependency.

To exploit the statistical relations, we propose *Deep Relational Network (DR-Net)*, a novel formulation that incorporates statistical relational modeling into a deep neural network framework. In our experiments, we found that the use of such relations can effectively resolve the ambiguities caused by visual or spatial cues, thus substantially improving the prediction accuracy.

(4) Integrated Prediction. Next, we describe how these factors are actually combined. As shown in Figure 5.2, for each candidate pair, the framework extracts the appearance feature and the spatial feature, respectively via the appearance module and the spatial module. These two features are subsequently concatenated and further compressed via two fully-connected layers. This *compressed pair feature*, together with the appearance features of individual objects will be fed to the DR-Net for joint inference. Through multiple inference units, whose parameters capture the statistical relations among triplet components, the *DR-Net* will output the posterior probabilities of s , r , and o . Finally, the framework produces the prediction by choosing the most probable classes for each of these components.

In the training, all stages in our framework, namely object detection, pair filtering and joint recognition are trained respectively. As for joint recognition, different factors will be integrated into a single network and jointly fine-tuned to maximize the joint probability of the ground-truth triplets.

5.4 Deep Relational Network

As shown above, there exist strong statistical relations among the object categories s and o and the relationship predicates r . Hence, to accurately recognize visual relationships, it is important to exploit such information, especially when the visual cues are ambiguous.

5.4.1 Revisit of CRF

The *Conditional Random Field (CRF)* [65] is a classical formulation to incorporate statistical relations into a discriminative task. Specifically, for the task of recognizing visual relationships, the CRF can be formulated as

$$p(r, s, o | \mathbf{x}_r, \mathbf{x}_s, \mathbf{x}_o) = \frac{1}{Z} \exp(\Phi(r, s, o | \mathbf{x}_r, \mathbf{x}_s, \mathbf{x}_o; \mathbf{W})). \quad (5.1)$$

Here, \mathbf{x}_r is the *compressed pair feature* that combines both the appearance of the enclosing box and the spatial configurations; \mathbf{x}_s and \mathbf{x}_o are the appearance features respectively for the subject and the object; \mathbf{W} denotes the model parameters; and Z is the normalizing constant, whose value depends on the parameters \mathbf{W} . The joint potential Φ can be expressed as a sum of individual potentials as

$$\begin{aligned} \Phi = & \psi_a(s | \mathbf{x}_s; \mathbf{W}_a) + \psi_a(o | \mathbf{x}_o; \mathbf{W}_a) + \psi_r(r | \mathbf{x}_r; \mathbf{W}_r) \\ & + \varphi_{rs}(r, s | \mathbf{W}_{rs}) + \varphi_{ro}(r, o | \mathbf{W}_{ro}) + \varphi_{so}(s, o | \mathbf{W}_{so}). \end{aligned} \quad (5.2)$$

Here, the unary potential ψ_a associates individual objects with their appearance; ψ_r associates the relationship predicate with the feature \mathbf{x}_r ; while the binary potentials φ_{rs} , φ_{ro} and φ_{so} capture the statistical relations among the

relationship predicate r , the subject category s , and the object category o .

CRF formulations like this have seen wide adoption in computer vision literatures [132, 89] over the past decade, and have been shown to be a viable way to capture statistical dependencies. However, the success of CRF is limited by several issues: First, learning CRF requires computing the normalizing constant Z , which can be very expensive and even intractable, especially when cycles exist in the underlying graph, like the formulation above. Hence, approximations are often used to circumvent this problem, but they sometimes result in poor estimates. Second, when cyclic dependencies are present, variational inference schemes such as mean-field methods [56] and loopy belief propagation [86], are widely used to simplify the computation. This often leaves a gap between the objective of inference and that of training, thus leading to suboptimal results.

5.4.2 From CRF to DR-Net

Inspired by the success of deep neural networks [41, 105], we explore an alternative approach to relational modeling, that is, to *unroll* the inference into a feed-forward network.

Consider the CRF formulated above. Given s and o , then the posterior distribution of r is given by

$$p(r|s, o, \mathbf{x}_r; \mathbf{W}) \propto \exp(\psi_r(r|\mathbf{x}_r; \mathbf{W}_r) + \varphi_{rs}(r, s|\mathbf{W}_{rs}) + \varphi_{ro}(r, o|\mathbf{W}_{ro})). \quad (5.3)$$

In typical formulations, $\psi_r(r|\mathbf{x}_r)$ is often devised to be a linear functional of

\mathbf{x}_r for each r . Let \mathbf{W}_{rs} and \mathbf{W}_{ro} be matrices such that $\mathbf{W}_{rs}(r, s) = \varphi_{rs}(r, s)$ and $\mathbf{W}_{ro}(r, o) = \varphi_{ro}(r, o)$, and let \mathbf{q}_r be a vector of the posterior probabilities for r , then the formula above can be rewritten as¹

$$\mathbf{q}_r = \boldsymbol{\sigma}(\mathbf{W}_r \mathbf{x}_r + \mathbf{W}_{rs} \mathbf{1}_s + \mathbf{W}_{ro} \mathbf{1}_o). \quad (5.4)$$

Here, $\boldsymbol{\sigma}$ denotes the *softmax* function. $\mathbf{1}_s$ and $\mathbf{1}_o$ are one-hot indicator vectors for s and o . It can be shown that this is the optima to the optimization problem below:

$$\max_{\mathbf{q}} E_q [\psi_r(r|\mathbf{x}_r; \mathbf{W}_r) + \varphi_{rs}(r, s|\mathbf{W}_{rs}) + \varphi_{ro}(r, o|\mathbf{W}_{ro})] + H_q(\mathbf{q}). \quad (5.5)$$

Based on this optimization problem, the solution given in Eq.(5.4) can be generalized to the case where s and o are not deterministic and the knowledge of them are instead given by probabilistic vectors \mathbf{q}_s and \mathbf{q}_o , as follows:

$$\mathbf{q}_r = \boldsymbol{\sigma}(\mathbf{W}_r \mathbf{x}_r + \mathbf{W}_{rs} \mathbf{q}_s + \mathbf{W}_{ro} \mathbf{q}_o). \quad (5.6)$$

Similar derivation also applies to the inference of s and o conditioned on other components. Together, we can obtain a set of *updating formulas* as

¹A proof of this statement is provided in the additional materials.

below:

$$\begin{aligned}\mathbf{q}'_s &= \sigma(\mathbf{W}_a \mathbf{x}_s + \mathbf{W}_{sr} \mathbf{q}_r + \mathbf{W}_{so} \mathbf{q}_o), \\ \mathbf{q}'_r &= \sigma(\mathbf{W}_r \mathbf{x}_r + \mathbf{W}_{rs} \mathbf{q}_s + \mathbf{W}_{ro} \mathbf{q}_o), \\ \mathbf{q}'_o &= \sigma(\mathbf{W}_a \mathbf{x}_o + \mathbf{W}_{os} \mathbf{q}_s + \mathbf{W}_{or} \mathbf{q}_r).\end{aligned}\tag{5.7}$$

These formulas take the current probability vectors \mathbf{q}_s , \mathbf{q}_r , and \mathbf{q}_o as inputs, and output the updated versions \mathbf{q}'_s , \mathbf{q}'_r and \mathbf{q}'_o . From the perspective of neural networks, these formulas can also be viewed as a *computing layer*. In this sense, the iterative updating procedure can be *unrolled* into a network that comprises a sequence of such layers. We call this network the *Deep Relational Network (DR-Net)*, as it relates multiple variables, and refer to its building blocks, *i.e.* the computing layers mentioned above, as *inference units*.

Discussion DR-Net is for relational modeling, which is different from those methods for feature/modality combination. Specifically, *object categories* and *relationship predicates* are two distinct domains that are statistically related. The former is not an extra feature of the latter; while the latter is not a feature of the former either. DR-Net captures the relations between them via the links in the inference units, rather than combining them using a fusion layer.

The basic formulation in Eq.(5.7) comes with several symmetry constraints: $\mathbf{W}_{sr} = \mathbf{W}_{rs}^T$, $\mathbf{W}_{so} = \mathbf{W}_{os}^T$, and $\mathbf{W}_{ro} = \mathbf{W}_{or}^T$. In addition, all inference units share the same set of weights. However, from a pragmatic standpoint, one may also consider lifting these constraints, *e.g.* allowing each

inference units to have their own weights. This may potentially increase the expressive power of the network. We will compare these two settings, namely with and without weight sharing, in our experiments.

A DR-Net can also be considered as a special form of the Recurrent Neural Network (RNN) – at each step it takes in a fixed set of inputs, *i.e.* the observed features \mathbf{x}_s , \mathbf{x}_r , and \mathbf{x}_o , and refines the estimates of posterior probabilities.

5.4.3 Comparison with Other Formulations

There are previous efforts that also explore the incorporation of relational structures with deep networks [12, 132, 104, 6]. The deep structured models presented in [12, 104, 119] combine a deep network with an MRF or CRF on top to capture the relational structures among their outputs. In these works, classical message-passing methods are used in training and inference. Zheng *et al* [132] proposed a framework for image segmentation, which adopts an apparently similar idea, that is, to reformulate a structured model into a neural network by turning inference updates into neural layers. In addition to the fact that this work is in a fundamentally different domain (high-level understanding vs. low-level vision), they focused on capturing dependencies among elements in the same domain, *e.g.* those among pixel-wise labels. From a technical view, DR-Net is more flexible, *e.g.* it can handle graphs with nodes of different cardinalities and edges of different types. In [132], the message passing among pixels is *approximately* instantiated using CNN filters and this is primarily suited for grid structures; while in DR-Net, the inference steps are exactly reproduced using fully-connected layers. Hence,

		Predicate Recognition		Union Box Detection		Two Boxes Detection	
		Recall@50	Recall@100	Recall@50	Recall@100	Recall@50	Recall@100
VRD	VP [102]	0.97	1.91	0.04	0.07	-	-
	Joint-CNN[24]	1.47	2.03	0.07	0.09	0.07	0.09
	VR [77]	47.87	47.87	16.17	17.03	13.86	14.70
	DR-Net	80.78	81.90	19.02	22.85	16.94	20.20
	DR-Net + pair filter	-	-	19.93	23.45	17.73	20.88
sVG	VP [102]	0.63	0.87	0.01	0.01	-	-
	Joint-CNN[24]	3.06	3.99	1.24	1.60	1.21	1.58
	VR [77]	53.49	54.05	13.80	17.39	11.79	14.84
	DR-Net	88.26	91.26	20.28	25.74	17.51	22.23
	DR-Net + pair filter	-	-	23.95	27.57	20.79	23.76

Table 5.1: Comparison with baseline methods, using *Recall@50* and *Recall@100* as the metrics. We use “-” to indicate “*not applicable*”. For example, no results are reported for *DR-Net + pair filter* on Predicate Recognition, as in this setting, pairs are given, and thus pair filtering can not be applied. Also, no results are reported for *VP* on Two Boxes detection, as *VP* detects the entire instance as a single entity.

		A ₁	A ₂	S	A ₁ S	A ₁ SC	A ₁ SD	A ₂ SD	A ₂ SDF
VRD	Predicate Recognition	63.39	65.93	64.72	71.81	72.77	80.66	80.78	-
	Union Box Detection	12.01	12.56	13.76	16.04	16.37	18.15	19.02	19.93
	Two Boxes Detection	10.71	11.22	12.16	14.38	14.66	16.12	16.94	17.73
sVG	Predicate Recognition	72.13	72.54	75.18	79.10	79.18	88.00	88.26	-
	Union Box Detection	13.24	13.84	14.01	16.04	16.08	20.21	20.28	23.95
	Two Boxes Detection	11.35	11.98	12.07	13.77	13.81	17.42	17.51	20.79

Table 5.2: Comparison of different variants of the proposed method, using *Recall@50* as the metric.

it can be applied to capture relationships of arbitrary structures. SPENs introduced in [6] define a neural network serving as an energy function over observed features for multi-label classification. SPENs are used to measure the consistency of configurations, while DR-Net is used to find a good configuration of variables. Also, no inference unrolling is involved in SPENs learning.


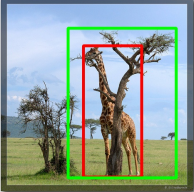
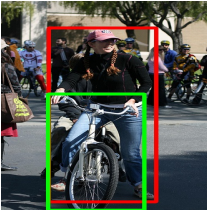
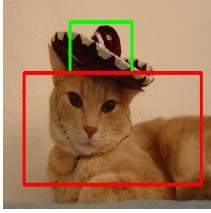
				
VR[77]	(sky, in , water)	(giraffe, have , tree)	(woman, ride , bicycle)	(cat, have , hat)
A ₁	(sky, on , water)	(giraffe, have , tree)	(woman, behind , bicycle)	(cat, on , hat)
S	(sky, above , water)	(giraffe, in , tree)	(woman, wear , bicycle)	(cat, have , hat)
A ₁ S	(sky, above , water)	(giraffe, behind , tree)	(woman, wear , bicycle)	(cat, have , hat)
A ₁ SC	(sky, above , water)	(giraffe, behind , tree)	(woman, ride , bicycle)	(cat, have , hat)
A ₁ SD	(sky, above , water)	(giraffe, behind , tree)	(woman, ride , bicycle)	(cat, wear , hat)

Table 5.3: This table lists predicate recognition results for some object pairs. Images containing these pairs are listed in the first row, where the red and green boxes respectively correspond to the subjects and the objects. The most probable predicate predicted by different methods are listed in the following rows, in which **black** indicates wrong prediction and **red** indicates correct prediction.

5.5 Experiments

We tested our model on two datasets: (1) **VRD**: the dataset used in [77], containing 5,000 images and 37,993 visual relationship instances that belong to 6,672 triplet types. We follow the train/test split in [77]. (2) **sVG**: a substantially larger subset constructed from Visual Genome [58]. *sVG* contains 108K images and 998K relationship instances that belong to 74,361 triplet types. All instances are randomly partitioned into disjoint training and testing sets, which respectively contain 799K and 199K instances.

5.5.1 Experiment Settings

Model training. In all experiments, we trained our model using Caffe[48]. The appearance module is initialized with a model pre-trained on ImageNet, while the spatial module and the DR-Net are initialized randomly. After

initialization, the entire network is jointly optimized using SGD.

Performance metrics. Following [77], we use $Recall@K$ as the major performance metric, which is the the fraction of ground-truth instances that are correctly recalled in top K predictions. Particularly, we report $Recall@100$ and $Recall@50$ in our experiments. The reason of using *recall* instead of *precision* is that the annotations are incomplete, where some true relationships might be missing.

Task settings. Like in [77], we studied three task settings: **(1) Predicate recognition:** this task focuses on the accuracy of *predicate* recognition, where the labels and the locations of both the *subject* and *object* are given. **(2) Union box detection:** this task treats the whole triplet as a union bounding box. A prediction is considered correct if all three elements in a triplet (s, r, o) are correctly recognized, and the IoU between the predicted box and the ground-truth is above 0.5. **(3) Two boxes detection:** this is similar to the one above, except that it requires the IoU metrics for the subject and the object are both above 0.5. This is relatively more challenging.

5.5.2 Comparative Results

Compare with baselines. We compared our method with the following methods under all three task settings outlined above. (1) **Visual Phrase(VP)** [102]: a representative approach that treats each distinct triplet as a different class. and employs a DPM detector [26] for each class. (2) **Joint-CNN** [24]: a neural network [105] that has $2N+K$ -way outputs, jointly predicts the class responses for subject, object, and relationship predicate. (3) **Visual Re-**

lationship (VR) [77]: This is the state-of-the-art and is the most closely related work.

Table 5.1 compares the results. On both datasets, we observed: (1) VP [102] performs very poorly, failing in most cases, as it is difficult to cope with such a huge and imbalanced class space. (2) Joint-CNN [24] also works poorly, as it’s hard for the CNN to learn a common feature representation for both relationship predicates and objects. (3) VR [77] performs substantially better than the two above. However, the performance remains unsatisfactory. (4) The proposed method outperforms the state-of-the-art method *VR* [77] by a considerable margin in all three tasks. Compared to *VR*, it improves the *Recall@100* of *predicate recognition* by over 30% on both datasets. Thanks to the remarkably improved accuracy in recognizing the relationship predicates, the performance gains on the other two tasks are also significant. (5) Despite the significant gain compared to others, the recalls on *union box detection* and *two boxes detection* remains weak. This is primarily ascribed to the limitations of the object detectors. As shown in Figure 5.4, we observe that the object detector can only obtain about 30% of object recall, measured by *Recall@50*. To improve on these tasks, a more sophisticated object detector is needed.

Compare different configs. We also compared different variants of the proposed method, in order to identify the contributions of individual components listed below: (1)**Pair (F)ilter**: the pair filter discussed in section 5.3, used to filter out object pairs with trivial relationships. (2)**(A)ppearance Module**: the appearance module, which has two versions, A_1 : based on VGG16 [105], which is also the network used in VR [77], A_2 : based on

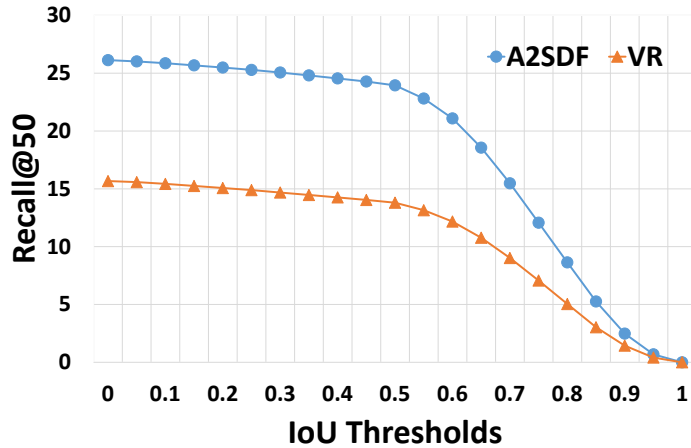


Figure 5.4: This figure shows the performance on the *union-box detection* task with different IoU thresholds.

ResNet101 [41]. (3)**(S)patial Module**: the network to capture the spatial configs, as mentioned in section 5.3. (4)**(C)RF**: a classical CRF formulation, used as a replacement of the DR-Net to capture statistical dependencies. (5)**(D)R-Net**: the DR-Net discussed in section 5.4. The name of a configuration is the concatenation of abbreviations of involved components, *e.g.*, the configuration named A_1SC contains an appearance module based on VGG16, a spatial module, and a CRF.

In Table 5.2, we compared A_1 , A_2 , S , A_1S , A_1SC , A_1SD , A_2SD and A_2SDF . The results show: (1) Using better networks (ResNet-101 vs. VGG16) can moderately improve the performance. However, even with state-of-the-art network A_2 , visual relationship detection could not be done effectively using appearance information alone. (2) The combination of appearance and spatial configs considerably outperforms each component alone, suggesting that visual appearances and spatial configurations are complementary to each other. (3) The statistical dependencies are important. However, CRF is not

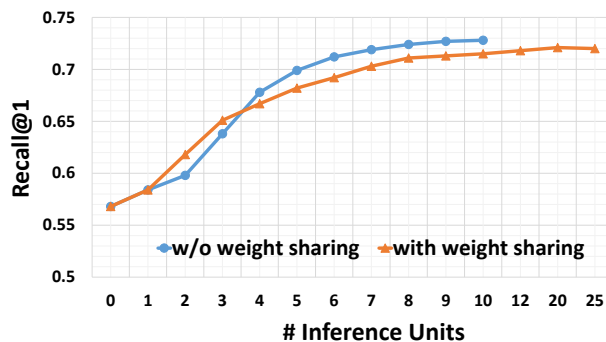


Figure 5.5: This figure shows the recall curves of two possible settings in DR-Net. In each setting, we change the number of inference units to see how the recall changes.

able to effectively exploit them. With the use of DR-Net, the performance gains are significant. We evaluated the perplexities of the predictions for our model *with* and *without* DR-Net, which are 2.64 and 3.08. These results show the benefit of exploiting statistical dependencies for joint recognition.

Table 5.3 further shows the predicted relationships on several example images. The first two columns show that the incorporation of spatial configuration can help detect positional relationships. The third column shows that the use of statistical dependencies can help to resolve the ambiguities in the relationship predicates. Finally, the fourth column shows that for subtle cases, DR-Net can identify the relationship predicate more accurately than the config that relies on CRF.

Compare architectural choices. This study is to compare the effect of different choices in the DR-Net architecture. The choices we study here include: the number of inference units and whether the relational weights are shared across these units. The comparison is conducted on *sVG*.

Figure 5.5 shows the resultant curves. From the results we can see: (1) On

Average Similarity				
VR [77]	A_1	S	A_1S	A_1SD
0.2076	0.2081	0.2114	0.2170	0.2271

Table 5.4: This table lists the average similarities between generated scene graphs and the ground truth. All methods are named after their visual relationship detectors.

both settings, the recall increases as the number of inference units increases. The best model can improve the recall from 56% to 73%, as the number of inference units increases. With weight sharing, the recall saturates with 12 inference units; while without sharing, the recall increases more rapidly, and saturates when it has 8 inference units. (2) Generally, with same number of inference units, the network without weight sharing performs relatively better, due to the greater expressive power.

5.5.3 Scene Graph Generation

Our model for visual relationship detection can be used for scene graph generation, which can serve as the basis for many tasks, *e.g.* image captioning[2, 1], visual question answering[118] and image retrieval[49].

The task here is to generate a directed graph for each image that captures objects, object attributes, and the relationships between them [49]. See Figure 5.6 for an illustration. We compared several configs of our method, including A_1 , S , A_1S and A_1SD , with VR [77] on this task, on a dataset $sVG-a$, which extends sVG with attribute annotations. All methods are augmented with an attribute recognizer.

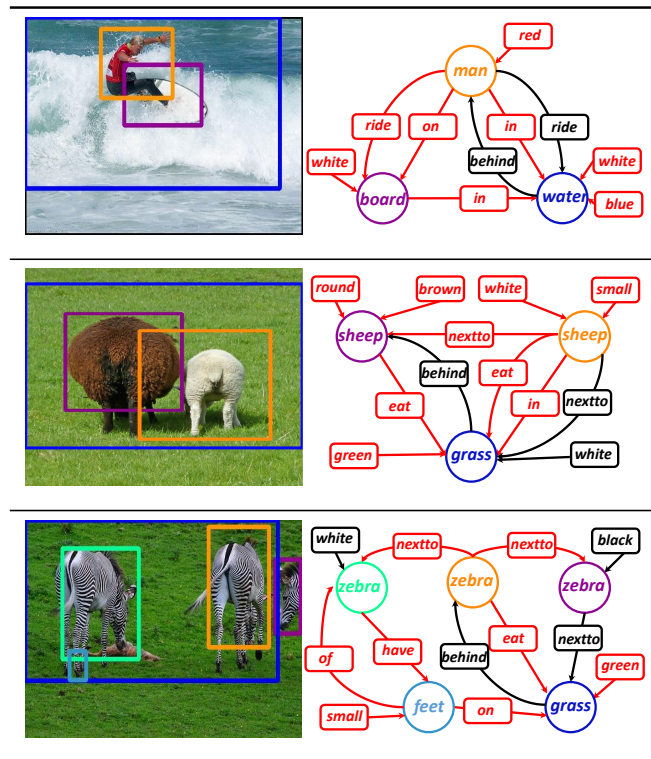


Figure 5.6: This figure illustrates some images and their corresponding scene graphs. The scene graphs are generated according to section 5.5.3. In the scene graphs, the **black** edges indicate wrong prediction, and the **red** edges indicate correct prediction.

For each test image, we measure the similarity [10] between the generated scene graph and the ground truth. We report average similarity over all test images as our metric. Table 5.4 compares the results of these approaches, where A_1SD achieves the best result. This comparison indicates that with better relationship detection, one can obtain better scene graphs.

5.6 Additional Materials

5.6.1 Proof of Eq.(5.4.2) in Section 5.4

The posterior distribution of r is given by

$$p(r|s, o, \mathbf{x}_r; \mathbf{W}) \propto \exp(\psi_r(r|\mathbf{x}_r; \mathbf{W}_r) + \phi_{rs}(r, s|\mathbf{W}_{rs}) + \phi_{ro}(r, o|\mathbf{W}_{ro})). \quad (5.8)$$

Here, we have

1. The unary potential $\psi_r(r|\mathbf{x}_r)$ is assumed to be a linear functional of \mathbf{x}_r for each predicate r , then we can write $\psi_r(r|\mathbf{x}_r) := \mathbf{a}_r^T \mathbf{x}_r$. Combining the linear functionals for all categories, we can form a coefficient matrix $\mathbf{W}_r = [\mathbf{a}_{r_1}^T, \mathbf{a}_{r_2}^T, \dots, \mathbf{a}_{r_{|\mathcal{R}|}}^T]$. Thus, $\psi_r(r|\mathbf{x}_r; \mathbf{W}_r) = \mathbf{1}_r^T \mathbf{W}_r \mathbf{x}_r$.
2. Both r and s be categorical variables. Hence, the potential ϕ_{rs} can be represented by a matrix of size $|\mathcal{R}| \times |\mathcal{O}|$, where \mathcal{R} is the set of all relationship predicates while \mathcal{O} is the set of all object categories. Particularly, let $\mathbf{1}_r$ and $\mathbf{1}_s$ be indicator vectors for r and s , then we have $\phi_{rs}(r, s|\mathbf{W}_{rs}) = \mathbf{1}_r^T \mathbf{W}_{rs} \mathbf{1}_s$.
3. Likewise, the potential ϕ_{ro} can also be characterized by a matrix \mathbf{W}_{ro} , such that $\phi_{ro}(r, o|\mathbf{W}_{ro}) = \mathbf{1}_r^T \mathbf{W}_{ro} \mathbf{1}_o$.

Let $\mathbf{q}_r(r) = p(r|s, o, \mathbf{x}_r; \mathbf{W})$, then Eq.(5.8) can be rewritten:

$$\mathbf{q}_r(r) \propto \exp(\mathbf{1}_r^T \mathbf{W}_r \mathbf{x}_r + \mathbf{1}_r^T \mathbf{W}_{rs} \mathbf{1}_s + \mathbf{1}_r^T \mathbf{W}_{ro} \mathbf{1}_o) \quad (5.9)$$

$$= \exp(\mathbf{1}_r^T (\mathbf{W}_r \mathbf{x}_r + \mathbf{W}_{rs} \mathbf{1}_s + \mathbf{W}_{ro} \mathbf{1}_o)). \quad (5.10)$$

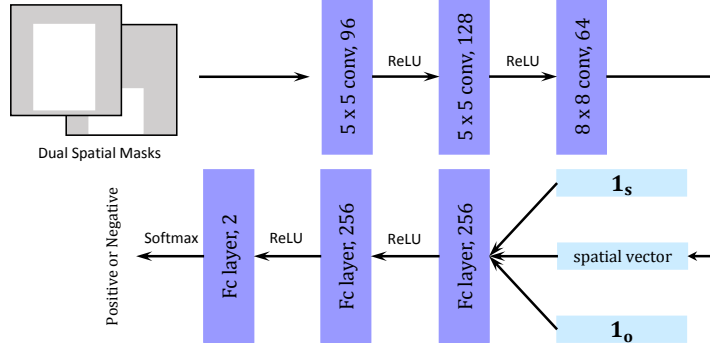


Figure 5.7: The network for pair filtering.

This equation can be interpreted as follows. The expression $\mathbf{e} = \mathbf{W}_r \mathbf{x}_r + \mathbf{W}_{rs} \mathbf{1}_s + \mathbf{W}_{ro} \mathbf{1}_o$ is a vector of length $|\mathcal{R}|$, and the operator $\mathbf{1}_r^T \mathbf{e}$ takes the r -th entry. \mathbf{q}_r is comprised of the normalized exponents of these entries, and thus can be written as

$$\mathbf{q}_r = \sigma(\mathbf{W}_r \mathbf{x}_r + \mathbf{W}_{rs} \mathbf{1}_s + \mathbf{W}_{ro} \mathbf{1}_o) \quad (5.11)$$

Here, σ is the softmax function that produces a vector of normalized exponents. This completes the proof.

5.6.2 Pair Filter

As mentioned in the paper, we use a simple network to filter out part of the pairs before feeding them to the main DR-Net for further analysis. Here are some technical details about the network. Figure 5.7 shows the architecture of this network. The network comprises three convolutional layers followed by three fully-connected layers. These layers are interleaved with *ReLU* activations. It is designed to be relatively shallow, so that it can perform the filtering with low cost. To train this network, we randomly

sample pairs of bounding boxes from each training image, treating those with 0.5 IoU (or above) with any ground-truth pairs as positive samples, and the rest as negative samples.

In testing, from n detected objects, we can form $n(n - 1)$ pairs. We use this filter to remove 40% of them, retaining 60%. This filtering rate was chosen empirically based on the overall empirical performance on a validation set.

Chapter 6

A Neural Compositional Captioning Model

6.1 Introduction

Image captioning, the task to generate short descriptions for given images, has received increasing attention in recent years. State-of-the-art models [78, 3, 115, 122] mostly adopt the encoder-decoder paradigm [115], where the content of the given image is first encoded via a convolutional network into a feature vector, which is then decoded into a caption via a recurrent network. In particular, the words in the caption are produced in a *sequential* manner – the choice of each word depends on both the preceding word and the image feature. Despite its simplicity and the effectiveness shown on various benchmarks [73, 128], the sequential model has a fundamental problem, namely, it does not reflect the *inherent* hierarchical structure of natural languages [80, 9].

As a result, sequential models have several significant drawbacks. First,

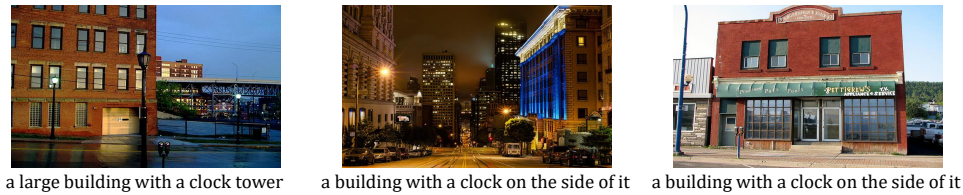


Figure 6.1: This figure shows three test images in MS-COCO [73] with captions generated by the neural image captioner [115], which contain n-gram *building with a clock* that appeared frequently in the training set but is not semantically correct for these images.

they rely excessively on n-gram statistics rather than hierarchical dependencies among words in a caption. Second, such models usually favor the frequent n-grams [16] in the training set, which, as shown in Figure 6.1, may lead to captions that are only correct *syntactically* but not *semantically*, containing semantic concepts that are irrelevant to the conditioned image. Third, the entanglement of syntactic rules and semantics obscures the dependency structure and makes the model difficult to generalize.

To tackle these issues, we propose a new paradigm for image captioning, where the extraction of semantics (*i.e. what to say*) and the construction of syntactically correct captions (*i.e. how to say*) are decomposed into two stages. Specifically, it derives an *explicit* representation of the semantic content of the given image, which comprises a set of noun-phrases, *e.g. a white cat, a cloudy sky* or *two men*. With these noun-phrases as the basis, it then proceeds to construct the caption through *recursive composition* until a complete caption is obtained. In particular, at each step of the composition, a higher-level phrase is formed by joining two selected sub-phrases via a connecting phrase. It is noteworthy that the compositional procedure described above is not a hand-crafted algorithm. Instead, it consists of two parametric modular nets, a *connecting module* for phrase composition and an *evaluation*

module for deciding the completeness of phrases.

The proposed paradigm has several key advantages compared to conventional captioning models: (1) The factorization of *semantics* and *syntax* not only better preserves the semantic content of the given image but also makes caption generation easy to interpret and control. (2) The recursive composition procedure naturally reflects the inherent structures of natural language and allows the hierarchical dependencies among words and phrases to be captured. Through a series of ablative studies, we show that the proposed paradigm can effectively increase the diversity of the generated captions while preserving semantic correctness. It also generalizes better to new data and can maintain reasonably good performance when the number of available training data is small.

6.2 Related Work

Literature in image captioning is vast, with the increased interest received in the neural network era. The early approaches were bottom-up and detection based, where a set of visual concepts such as objects and attributes were extracted from images [25, 60]. These concepts were then assembled into captions by filling the blanks in pre-defined templates [60, 68], learned templates [72], or served as anchors to retrieve the most similar captions from the training set [20, 25].

Recent works on image captioning adopt an alternative paradigm, which applies convolutional neural networks [40] as image representation, followed by recurrent neural networks [42] for caption generation. Specifically, Vinyals *et al* [115] proposed the *neural image captioner*, which represents the input

image with a single feature vector, and uses an LSTM [42] conditioned on this vector to generate words one by one. Xu *et al* [122] extended their work by representing the input image with a set of feature vectors, and applied an attention mechanism to these vectors at every time step of the recurrent decoder in order to extract the most relevant image information. Lu *et al* [78] adjusted the attention computation to also attend to the already generated text. Anderson *et al* [3] added an additional LSTM to better control the attention computation. Some of the recent approaches directly extract phrases or semantic words from the input image. Yao *et al* [126] predicted the occurrences of frequent training words, where the prediction is fed into the LSTM as an additional feature vector. Tan *et al* [110] treated noun-phrases as hyper-words and added them into the vocabulary, such that the decoder was able to produce a full noun-phrase in one time step instead of a single word. In [74], the authors proposed a hierarchical approach where one LSTM decides on the phrases to produce, while the second-level LSTM produced words for each phrase.

Despite the improvement over the model architectures, all these approaches generate captions *sequentially*. This tends to favor frequent n-grams [16], leading to issues such as incorrect semantic coverage, and lack of diversity. On the contrary, our proposed paradigm proceeds in a bottom-up manner, by representing the input image with a set of noun-phrases, and then constructs captions according to a recursive composition procedure. With such explicit disentanglement between semantics and syntax, the recursive composition procedure preserves semantics more effectively, requires less data to learn, and also leads to more diverse captions.

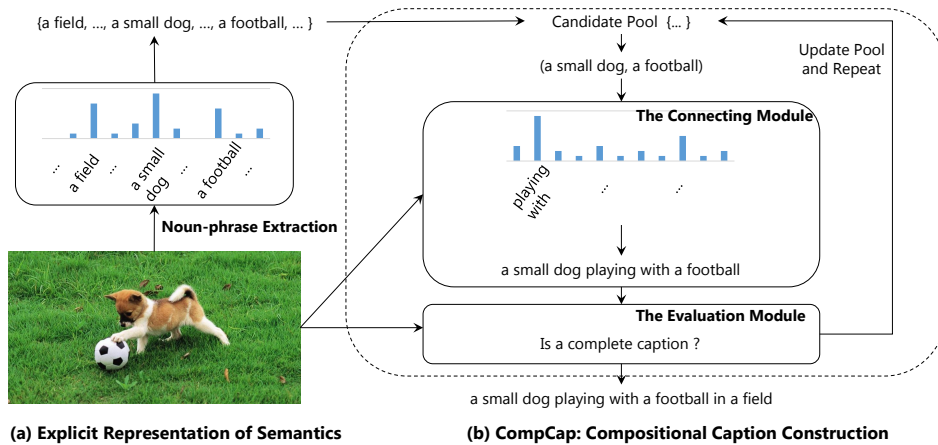


Figure 6.2: An overview of the proposed compositional paradigm. A set of noun-phrases is extracted from the input image first, serving as the initial pool of phrases for the compositional generation procedure. The procedure then recursively uses a connecting module to compose two phrases from the pool into a longer phrase, until an evaluation module determines that a complete caption is obtained.

Work conceptually related to ours is by Kuznetsova *et al* [63], which mines four types of phrases including noun-phrases from the training captions, and generates captions by selecting one phrase from each category and composes them via dynamic programming. Since the composition procedure is not recursive, it can only generate captions containing a single object, thus limiting the versatile nature of image description. In our work, any number of phrases can be composed, and we exploit powerful neural networks to learn plausible compositions.

6.3 Compositional Captioning

The structure of natural language is inherently *hierarchical* [9, 80], where the typical parsing of sentence takes the form of trees [55, 87, 108]. Hence, it seems natural to also produce captions following such hierarchical struc-

ture. Specifically, we propose a two-stage framework for image captioning, as shown in Figure 6.2. Given an image, we first derive a set of noun-phrases as an explicit semantic representation. We then construct the caption in a bottom-up manner, via a recursive compositional procedure which we refer to as **CompCap**. This procedure can be considered as an *inverse* of the sentence parsing process. Unlike mainstream captioning models that primarily rely on the n-gram statistics among consecutive words, CompCap can take into account the nonsequential dependencies among words and phrases of a sentence. In what follows, we will present these two stages in more detail.

6.3.1 Explicit Representation of Semantics

Conventional captioning methods usually encode the content of the given image into feature vectors, which are often difficult to interpret. In our framework, we represent the image semantics *explicitly* by a set of *noun-phrases*, e.g. “a black cat”, “a cloudy sky” and “two boys”. These noun-phrases can capture not only the object categories but also the associated attributes.

Next, we briefly introduce how we extract such noun-phrases from the input image. In our study, we found that the number of distinct noun-phrases in a dataset is significantly smaller than the number of images. For example, MS-COCO [73] contains 120K images but only about 3K distinct noun-phrases in the associated captions. Given this observation, it is reasonable to formalize the task of noun-phrase extraction as a multi-label classification problem.

Specifically, we derive a list of distinct noun-phrases $\{NP_1, NP_2, \dots, NP_K\}$

from the training captions by parsing the captions and selecting those noun-phrases that occur for more than 50 times. We treat each selected noun-phrase as a *class*. Given an image I , we first extract the visual feature \mathbf{v} via a Convolutional Neural Network as $\mathbf{v} = \text{CNN}(I)$, and further encode it via two fully-connected layers as $\mathbf{x} = F(\mathbf{v})$. We then perform binary classification for each noun-phrase NP_k as $S_C(NP_k|I) = \sigma(\mathbf{w}_k^T \mathbf{x})$, where \mathbf{w}_k is the weight vector corresponding to the class NP_k and σ denotes the sigmoid function.

Given $\{S_C(NP_k|I)\}_k$, the scores for individual noun-phrases, we choose to represent the input image using n of them with top scores. While the selected noun-phrases may contain semantically similar concepts, we further prune this set through *Semantic Non-Maximum Suppression*, where only those noun-phrases whose scores are the maximum among similar phrases are retained ¹.

6.3.2 Recursive Composition of Captions

Starting with a set of noun-phrases, we construct the caption through a recursive compositional procedure called **CompCap**. We first provide an overview, and describe details of all the components in the following paragraphs.

At each step, CompCap maintains a phrase pool \mathcal{P} , and scans all *ordered* pairs of phrases from \mathcal{P} . For each ordered pair $P^{(l)}$ and $P^{(r)}$, a *Connecting Module (C-Module)* is applied to generate a sequence of words, denoted as $P^{(m)}$, to connect the two phrases in a plausible way. This yields a longer phrase in the form of $P^{(l)} \oplus P^{(m)} \oplus P^{(r)}$, where \oplus denotes the operation of

¹ The details of this procedure are provided in the Section 6.5.

sequence concatenation. The C-Module also computes a connecting score for $P^{(l)} \oplus P^{(m)} \oplus P^{(r)}$. Among all phrases that can be composed from scanned pairs, we choose the one with the maximum connecting score as the new phrase P_{new} . A parametric module could also be used to determine P_{new} .

Subsequently, we apply an *Evaluation Module (E-Module)* to assess whether P_{new} is a *complete* caption. If P_{new} is determined to be complete, we take it as the resulting caption; otherwise, we update the pool \mathcal{P} by replacing the corresponding constituents $P^{(l)}$ and $P^{(r)}$ with P_{new} , and invoke the pair selection and connection process again based on the updated pool. The procedure continues until a complete caption is obtained or only a single phrase remains in \mathcal{P} .

We next introduce the connecting and the evaluation module, respectively.

The Connecting Module. The *Connecting Module (C-Module)* aims to select a *connecting phrase* $P^{(m)}$ given both the left and right phrases $P^{(l)}$ and $P^{(r)}$, and to evaluate the *connecting score* $S(P^{(m)} | P^{(l)}, P^{(r)}, I)$. While this task is closely related to the task of filling in the blanks of captions [130], we empirically found that the conventional way of using an LSTM to decode the intermediate words fails. This may be due to various reasons, *e.g.* we have to deal with not only complete captions but also parts thereof, which constitute a significantly larger space. In this work, we adopt an alternative strategy, namely, to treat the generation of connecting phrases as a classification problem. This is motivated by the observation that the number of distinct connecting phrases is actually limited in the proposed paradigm, since semantic words such as nouns and adjectives are not involved in the

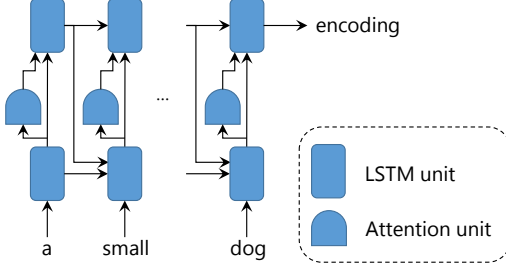
connecting phrases. For example, in MS-COCO [73], there are over 1 million samples collected for the connecting module, which contain only about 1,000 distinct connecting phrases.

Specifically, we mine a set of distinct connecting sequences from the training captions, denoted as $\{P_1^{(m)}, \dots, P_L^{(m)}\}$, and treat them as different classes. This can be done by walking along the parsing trees of the captions and extracting the intermediate sequences between noun-phrases or the longer connected phrases derived thereon. We then define the connecting module as a classifier, which takes the left and right phrases $P^{(l)}$ and $P^{(r)}$ as input and outputs a normalized score $S(P_j^{(m)} | P^{(l)}, P^{(r)}, I)$ for each $j \in \{1, \dots, L\}$.

In particular, we adopt a two-level LSTM model [3] to encode $P^{(l)}$ and $P^{(r)}$ respectively, as shown in Fig. 6.3. Here, \mathbf{x}_t is the word embedding for the word at t -th step, and \mathbf{v} and $\{\mathbf{u}_1, \dots, \mathbf{u}_M\}$ are, respectively, global and regional image features extracted from a Convolutional Neural Network. In this model, the low-level LSTM controls the attention while interacting with the visual features, and the high-level LSTM drives the evolution of the encoded state. The encoders for $P^{(l)}$ and $P^{(r)}$ share the same structure but have different parameters, as one phrase should be encoded differently based on its place in the ordered pair. Their encodings, denoted by $\mathbf{z}^{(l)}$ and $\mathbf{z}^{(r)}$, go through two fully-connected layers followed by a softmax layer, as

$$S(P_j^{(m)} | P^{(l)}, P^{(r)}, I) = \text{Softmax}(\mathbf{W}_{combine} \cdot (\mathbf{W}_l \cdot \mathbf{z}^{(l)} + \mathbf{W}_r \cdot \mathbf{z}^{(r)}))|_j, \quad \forall j = 1, \dots, L. \quad (6.1)$$

The values of the softmax output, *i.e.* $S(P_j^{(m)} | P^{(l)}, P^{(r)}, I)$, are then used as the *connecting scores*, and the connecting phrase that yields the highest connecting score is chosen to connect $P^{(l)}$ and $P^{(r)}$.



(a) Structure of the Phrase Encoder

$$\begin{aligned}
 \mathbf{h}_0^{(att)} &= \mathbf{h}_0^{(lan)} = \mathbf{0} \\
 \mathbf{h}_t^{(att)} &= \text{LSTM}(\mathbf{x}_t, \mathbf{v}, \mathbf{h}_{t-1}^{(lan)}, \mathbf{h}_{t-1}^{(att)}) \\
 \mathbf{a}_t &= \text{Attention}(\mathbf{h}_t^{(att)}, \mathbf{u}_1, \dots, \mathbf{u}_M) \\
 \mathbf{h}_t^{(lan)} &= \text{LSTM}(\mathbf{a}_t, \mathbf{h}_t^{(att)}, \mathbf{h}_{t-1}^{(lan)}) \\
 \mathbf{z} &= \mathbf{h}_T^{(lan)}
 \end{aligned}$$

(b) Computation of the Phrase Encoder

Figure 6.3: This figure shows the two-level LSTM used to encode phrases in the connecting and evaluation modules. **Left:** the structure of the phrase encoder, **right:** its updating formulas.

While not all pairs of $P^{(l)}$ and $P^{(r)}$ can be connected into a longer phrase, in practice a virtual connecting phrase $P_{\text{neg}}^{(m)}$ is added to serve as a negative class.

Based on the C-Module, we compute the connecting score for a phrase as follow. For each noun-phrase P in the initial set, which is derived in the phrase-from-image extraction stage, we set its score to be the binary classification score $S_C(P|I)$. For each longer phrase produced via the C-Module, its score is computed as

$$S(P^{(l)} \oplus P^{(m)} \oplus P^{(r)} | I) = S(P^{(l)} | I) + S(P^{(r)} | I) + S(P^{(m)} | P^{(l)}, P^{(r)}, I). \quad (6.2)$$

The Evaluation Module. The *Evaluation Module (E-Module)* is used to determine whether a phrase is a complete caption. Specifically, given an input phrase P , the E-Module encodes it into a vector \mathbf{z}_e , using a two-level LSTM model as described above, and then evaluates the probability of P

being a complete caption as

$$\Pr(P \text{ is complete}) = \sigma(\mathbf{w}_{cp}^T \mathbf{z}_e). \quad (6.3)$$

Extensions. Instead of following the greedy search strategy described above, we can extend the framework for generating diverse captions for a given image, via beam search or probabilistic sampling. Particularly, we can retain multiple ordered pairs at each step and multiple connecting sequences for each retained pair. In this way, we can form multiple beams for beam search, and thus avoid being stuck in local minima. Another possibility is to generate diverse captions via probabilistic sampling, *e.g.* sampling a part of the ordered pairs for pair selection instead of using all of them, or sampling the connecting sequences based on the softmax probabilities instead of choosing the one that yields the highest score.

The framework can also be extended to incorporate user preferences or other conditions, as it consists of operations that are interpretable and controllable. For example, one can influence the resultant captions by filtering the initial noun phrases or modulating their scores. Such control is much easier to implement on an explicit representation, *i.e.* a set of noun phrases, than on an encoded feature vector. We show examples in the Experimental section.

6.4 Experiments

6.4.1 Experiment Settings

All experiments are conducted on MS-COCO [73] and Flickr30k [128]. There are 123,287 images and 31,783 images respectively in MS-COCO and Flickr30k, each of which has 5 ground-truth captions. We follow the splits in [50] for both datasets. In both datasets, the vocabulary is obtained by turning words to lowercase and removing words that have non-alphabet characters and appear less than 5 times. The removed words are replaced with a special token *UNK*, resulting in a vocabulary of size 9,487 for MS-COCO, and 7,000 for Flickr30k. In addition, training captions are truncated to have at most 18 words. To collect training data for the connecting module and the evaluation module, we further parse ground-truth captions into trees using NLPtoolkit [79].

Several representative methods are compared with CompCap. They are 1) *Neural Image Captioner (NIC)* [115], which is the backbone network for state-of-the-art captioning models. 2) *AdapAtt* [78] and 3) *TopDown* [3] are methods that apply the attention mechanism and obtain state-of-the-art performances. While all of these baselines encode images as semantical feature vectors, we also compare CompCap with 4) *LSTM-A5* [126], which predicts the occurrence of semantical concepts as additional visual features. Subsequently, besides being used to extract noun-phrases that fed into CompCap, predictions of the noun-phrase classifiers also serve as additional features for *LSTM-A5*.

To ensure a fair comparison, we have re-implemented all methods, and

	COCO-offline					Flickr30k				
	SP	CD	B4	RG	MT	SP	CD	B4	RG	MT
NIC [115]	17.4	92.6	30.2	52.3	24.3	12.0	40.7	19.9	42.9	18.0
AdapAtt [78]	18.1	97.0	31.2	53.0	25.0	13.4	48.2	23.3	45.5	19.3
TopDown [3]	18.7	101.132.4	53.8	25.7		13.8	49.8	23.7	45.6	19.7
LSTM-A5 [126]	18.0	96.6	31.2	53.0	24.9	12.2	43.7	20.4	43.8	18.2
CompCap + Pred _{np}	19.9	86.2	25.1	47.8	24.3	14.9	42.0	16.4	39.4	19.0
CompCap + GT _{np}	36.8	122.2	42.8	55.3	33.6	31.9	89.7	37.8	50.5	28.7
CompCap + GT _{np} + GT _{order}	33.8	182.6	64.1	82.4	45.1	29.8	132.8	54.9	77.1	39.6

Table 6.1: This table lists results of different methods on MS-COCO [73] and Flickr30k [128]. Results of CompCap using ground-truth noun-phrases and composing orders are shown in the last two rows.

train all methods using the same hyperparameters. Specifically, we use ResNet-152 [40] pretrained on ImageNet [100] to extract image features, where activations of the last convolutional and fully-connected layer are used respectively as the regional and global feature vectors. During training, we fix ResNet-152 without finetuning, and set the learning rate to be 0.0001 for all methods. When testing, for all methods we select parameters that obtain best performance on the validation set to generate captions. Beam-search of size 3 is used for baselines. As for CompCap, $n = 7$ noun-phrases with top scores are selected to represent the input image, and beam-search of size 3 is used for pair selection, while no beam-search is used for connecting phrase selection.

6.4.2 Experiment Results

General Comparison. We compare the quality of the generated captions on the offline test set of MS-COCO and the test set of Flickr30k, in terms of SPICE (SP) [2], CIDEr (CD) [113], BLEU-4 (B4) [84], ROUGE

(RG) [71], and METEOR (MT) [66]. As shown in Table 6.1, among all methods, CompCap with predicted noun-phrases obtains the best results under the SPICE metric, which has higher correlation with human judgments [2], but is inferior to baselines in terms of CIDEr, BLEU-4, ROUGE and METEOR. These results well reflect the properties of methods that generate captions sequentially and compositionally. Specifically, while SPICE focuses on semantical analysis, metrics including CIDEr, BLEU-4, ROUGE and METEOR are known to favor frequent training n-grams [16], which are more likely to appear when following a sequential generation procedure. On the contrary, the compositional generation procedure preserves semantic content more effectively, but may contain more n-grams that are not observed in the training set.

An ablation study is also conducted on components of the proposed compositional paradigm, as shown in the last three rows of Table 6.1. In particular, we represented the input image with ground-truth noun-phrases collected from 5 associated captions, leading to a significant boost in terms of all metrics. This indicates that CompCap effectively preserves the semantic content, and the better the semantic understanding we have for the input image, CompCap is able to generate better captions for us. Moreover, we also randomly picked one ground-truth caption, and followed its composing order to integrate its noun-phrases into a complete caption, so that CompCap only accounts for connecting phrase selection. As a result, metrics except for SPICE obtain further boost, which is reasonable as we only use a part of all ground-truth noun-phrases, and frequent training n-grams are more likely to appear following some ground-truth composing order.

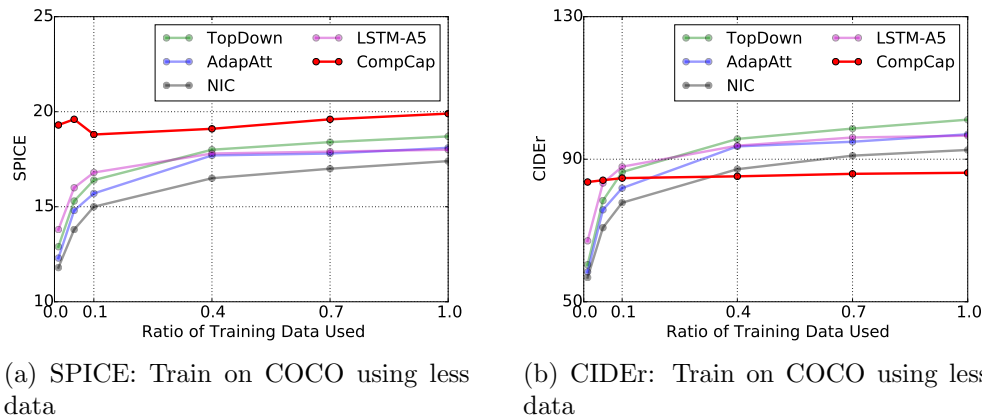


Figure 6.4: This figure shows the performance curves of different methods when less data is used for training. Unlike baselines, CompCap obtains stable results as the ratio of used data decreases.

Generalization Analysis. As the proposed compositional paradigm disentangles semantics and syntax into two stages, and CompCap mainly accounts for composing semantics into a syntactically correct caption, CompCap is good at handling out-of-domain semantic content, and requires less data to learn. To verify this hypothesis, we conducted two studies. In the first experiment, we controlled the ratio of data used to train the baselines and modules of CompCap, while leaving the noun-phrase classifiers being trained on full data. The resulting curves in terms of SPICE and CIDEr are shown in Figure 6.4, while other metrics follow similar trends. Compared to baselines, CompCap is steady and learns how to compose captions even only 1% of the data is used.

In the second study, we trained baselines and CompCap on MS-COCO/Flickr30k, and tested them on Flickr30k/MS-COCO. Again, the noun-phrase classifiers are trained with in-domain data. The results in terms of SPICE and CIDEr are shown in Figure 6.5, where significant drops are observed for the base-

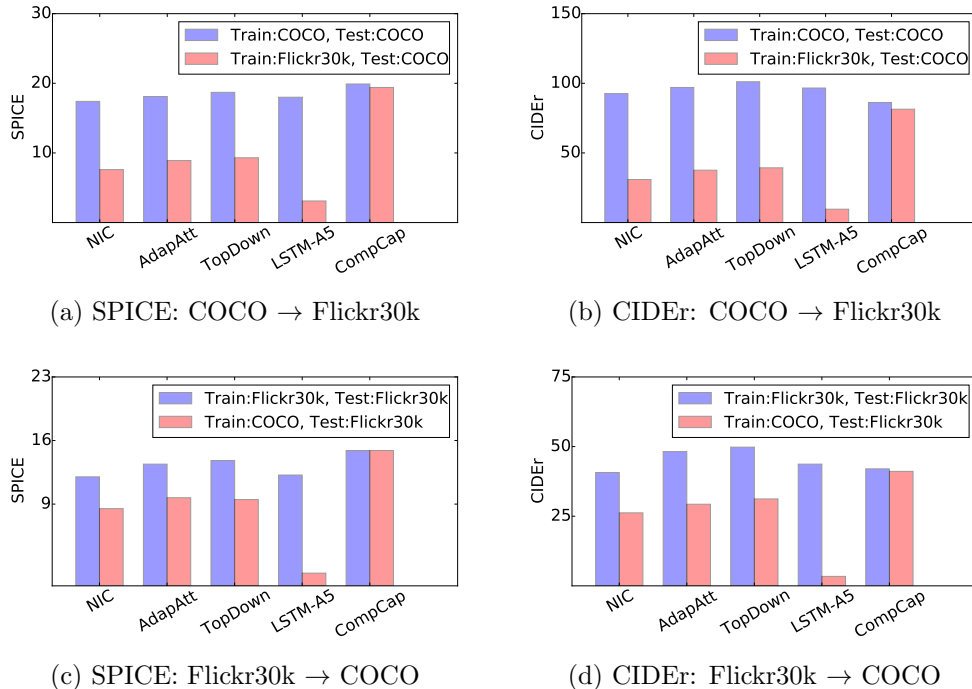


Figure 6.5: This figure compares the generalization ability of different methods, where they are trained on one dataset, and tested on the other. Compared to baselines, CompCap is shown to generalize better across datasets.

lines. On the contrary, competitive results are obtained for CompCap trained using in-domain and out-of-domain data, which suggests the benefit of disentangling semantics and syntax, as the distribution of semantics often varies from dataset to dataset, but the distribution of syntax is relatively stable across datasets.

Diversity Analysis. One important property of CompCap is the ability to generate diverse captions, as these can be obtained by varying the involved noun-phrases or the composing order. To analyze the diversity of captions, we computed five metrics that evaluate the degree of diversity from various aspects. As shown in Table 6.2, we computed the ratio of novel captions and unique captions [117], which respectively account for the percentage of

	NIC [115]	AdapAtt [78]	TopDown [3]	LSTM-A5 [126]	CompCap
Novel Caption Ratio	44.53%	49.34%	45.05%	50.06%	90.48%
Unique Caption Ratio	55.05%	59.14%	61.58%	62.61%	83.86%
Diversity (Dataset)	7.69	7.86	7.99	7.77	9.85
Diversity (Image)	2.25	3.61	2.30	3.70	5.57
Vocabulary Usage	6.75%	7.22%	7.97%	8.14%	9.18%

Table 6.2: This table measures the diversity, on MSCOCO, of generated captions from various aspects, which suggests CompCap is able to generate more diverse captions.

captions that are not observed in the training set, and the percentage of distinct captions among all generated captions. We further computed the percentage of words in the vocabulary that are used to generate captions, referred to as the vocabulary usage.

Finally, we quantify the diversity of a set of captions by averaging their pair-wise editing distances, which leads to two additional metrics. Specifically, when only a *single* caption is generated for each image, we report the average distance over captions of different images, which is defined as the diversity at the dataset level. If *multiple* captions are generated for each image, we then compute the average distance over captions of the same image, followed by another average over all images. The final average is reported as the diversity at the image level. The former measures how diverse the captions are for different images, and the latter measures how diverse the captions are for a single image. In practice, we use 5 captions with top scores in the beam search to compute the diversity at the image level, for each method.

CompCap obtained the best results in all metrics, which suggests that captions generated by CompCap are diverse and novel. We further show qualitative samples in Figure 6.6, where captions are generated following

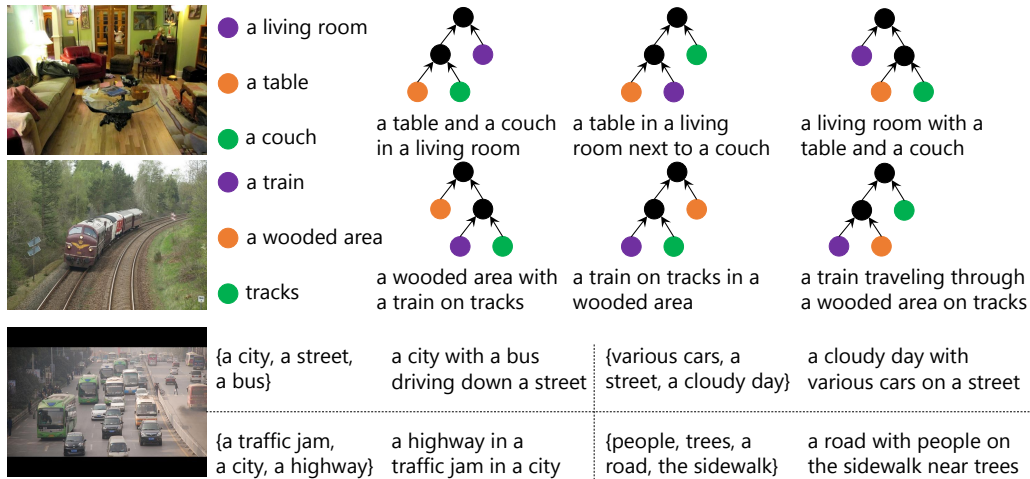


Figure 6.6: This figure shows images with diverse captions generated by Comp-Cap. In first two rows, captions are generated with same noun-phrases but different composing orders. And in the last row, captions are generated with different sets of noun-phrases.

different composing orders, or using different noun-phrases.

6.5 Additional Materials

6.5.1 Semantic Non-Maximum Suppression for Noun Phrases

The key for suppression is to find semantically similar noun-phrases. To do that, we first compare the central nouns in noun-phrases, where if two central nouns are synonyms, or plurals of synonyms, we then regard their corresponding noun-phrases as semantically similar. On the other hand, two noun-phrases that do not have synonymic central nouns are also likely to be semantically similar, conditioned on the input image. *e.g.* *a man* and *a cook* conditioned on an image of somebody in a kitchen. To suppressing noun-

phrases in such cases, we use encoders in the C-Module (See sec 3.2 of the main content) to get two encodings $\mathbf{z}^{(l)}$ and $\mathbf{z}^{(r)}$ for each noun-phrase. Intuitively, if two noun-phrases are semantically similar conditioned on the input image, the normalized euclidean distance between their encodings should be small. As a result, we compute the normalized euclidean distances respectively for $\mathbf{z}^{(l)}$ and $\mathbf{z}^{(r)}$ of two noun-phrases, and take the sum of two distances as the measurement, which is more robust than using a single encoding. Finally, if the sum of distances is less than ϵ we then regard the corresponding noun-phrases as semantically similar, conditioned on the input image. In practice, we use $\epsilon = 0.002$, which is obtained by grid search on the evaluation set.

6.5.2 Encoders in the C-Module

	SP	CD	B4	RG	MT
Encoders with shared parameters	18.9	84.9	24.3	46.8	23.2
Encoders with independent parameters	19.9	86.2	25.1	47.8	24.3

Table 6.3: This table lists results of CompCap using C-Modules that have encoders with shared parameters or not. Results are reported on MS-COCO [73].

As mentioned in the main content, the C-Module contains two encoders, respectively for $P^{(l)}$ and $P^{(r)}$ of an ordered pair. While these encoders share the same structure, we let them have independent parameters as the same phrase should have different encodings according to its position in the ordered pair. To show that, we compared C-Modules that have encoders with shared parameters or not, as shown in Table 6.3. The results support our hypothesis, where the C-Module that has encoders with independent parameters leads to better performance.

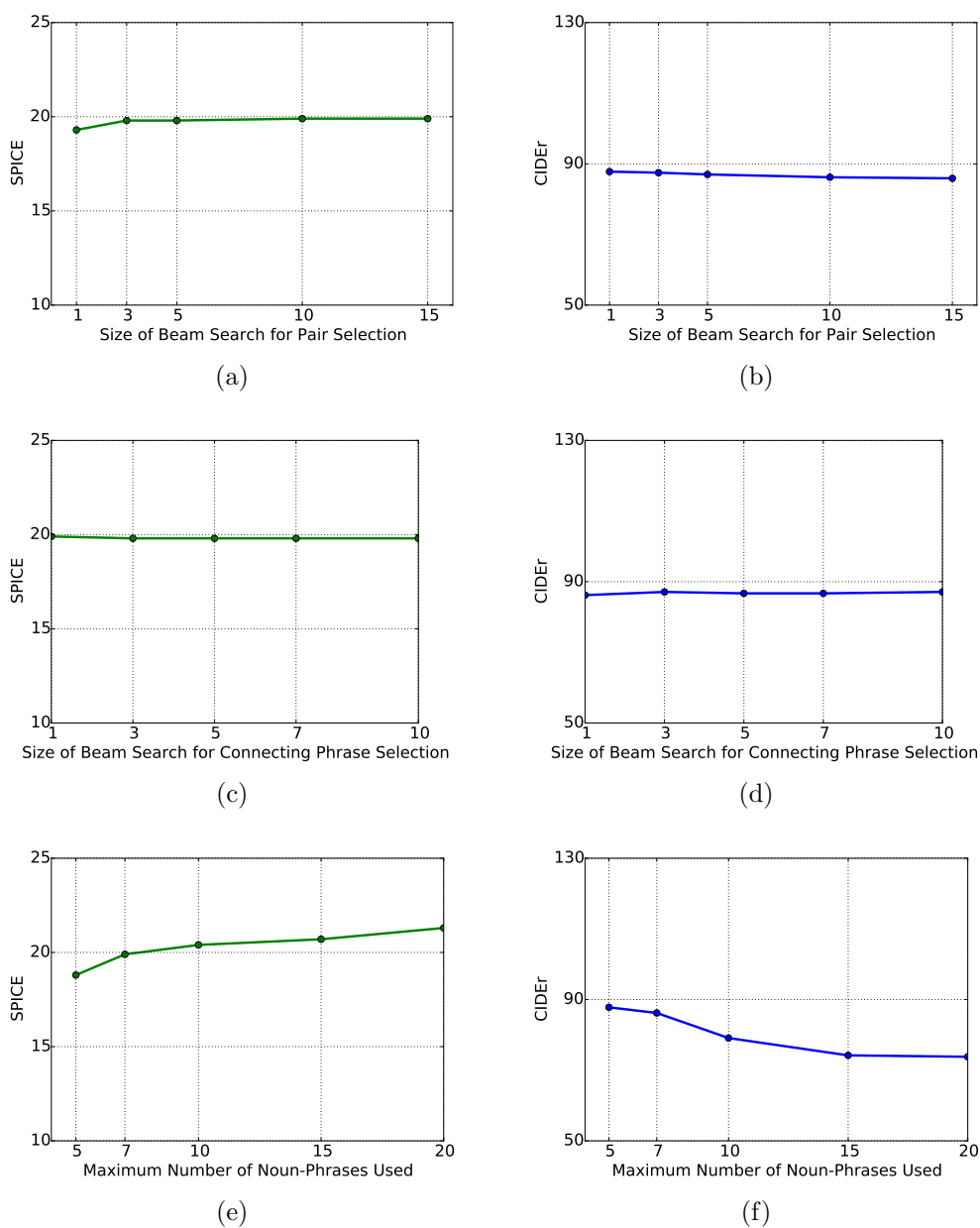


Figure 6.7: This figure shows the performance curves of CompCap by tuning different hyperparameters. Specifically, (a) and (b) are results of tuning the size of beam search for pair selection, in terms of SPICE and CIDEr. Similarly, (c) and (d), (e) and (f) are respectively results of tuning the size of beam search for connecting phrase selection, and the maximum number of noun-phrases used to generate captions.

6.5.3 Hyperparameters

Several hyperparameters can be tuned for CompCap, namely the maximum number of noun-phrases used to generate captions, the size of beam search for pair selection and the size of beam search for connecting phrase selection. While we respectively set them to be 7, 10 and 1 for experiments in the main content, here we show the curves of adjusting these hyperparameters, one at a time. The curves are shown in Figure 6.7, where the size of beam search for pair selection and connecting phrase selection have minor influence on the performance of CompCap. Moreover, as the maximum number of used noun-phrases increases, SPICE improves but CIDEr decreases, which is reasonable as too much semantics in a caption may risk the syntactic correctness of that caption. We use 7 in our experiments as it is a good trade-off.

Chapter 7

Conclusion

In this thesis, we studied the problem of generating more natural and diverse image descriptions by respectively taking limitations in terms of evaluation metrics, learning strategies and model structures, of existing captioning pipelines into consideration, and proposing corresponding components that avoid such limitations.

In the first part, we proposed an alternative learning strategy for image captioning, which aims to improve the overall quality of captions, including *semantic relevance*, *naturalness*, and *diversity*. The proposed approach is based on conditional GAN, and jointly trains a generator G and an evaluator E in an end-to-end manner, with the help of Policy Gradient. The caption generator trained in this way produces captions that are more natural, diverse and semantically relevant as compared to a state-of-the-art MLE-based model. Besides, the evaluator also provides caption quality assessment that is more consistent with human's evaluation.

In the second part, we proposed Contrastive Learning, a new learning strategy that employs a state-of-the-art model as reference, by which it is

able to maintain the optimality of the target model, while encouraging the target model to learn from distinctiveness, which is an important property of high quality captions. It not only leads more accurate captions being generated by the target model, but also extends well to models with different structures, which clearly shows its generalization ability.

In the third part, we studied the impact of embedding latent states as 2D multi-channel feature maps in the context of image captioning. Compared to the standard practice that embeds latent states as 1D vectors, 2D states consistently achieve higher captioning performances across different settings. Such representations also preserve the spatial locality of the latent states, which helps reveal the internal dynamics of the decoding process, and interpret the connections between visual and linguistic domains.

In the fourth part, we present a new framework for visual relationship detection, which integrates a variety of cues: appearance, spatial configurations, as well as the statistical relations between objects and relationship predicates. At the heart of this framework is the *Deep Relational Network (DR-Net)*, a novel formulation that extends the expressive power of deep neural networks to relational modeling. The proposed method not only outperforms the state of the art by a remarkable margin, but also yields promising results in scene graph generation, a task that represents higher level of image understanding.

In the final part, we propose a novel paradigm for image captioning. While the typical existing approaches encode images using feature vectors and generate captions sequentially, the proposed method generates captions in a compositional manner. In particular, our approach factorizes the captioning procedure into two stages. In the first stage, an explicit representation

of the input image, consisting of noun-phrases, is extracted. In the second stage, a recursive compositional procedure is applied to assemble extracted noun-phrases into a caption. As a result, caption generation follows a hierarchical structure, which naturally fits the properties of human language. The proposed compositional procedure is shown to preserve semantics more effectively, require less data to train, generalize better across datasets, and yield more diverse captions.

While image captioning is an important task in the community of computer vision, it lies at the intersection of observation and communication, making it a must-solve problem towards artificial intelligence. To this end, we hope our work could attract more attention to this problem and provide insights to interested researchers.

Bibliography

- [1] Somak Aditya, Yezhou Yang, Chitta Baral, Cornelia Fermuller, and Yiannis Aloimonos. From images to sentences through scene description graphs using common-sense reasoning and knowledge. *arXiv preprint arXiv:1511.03292*, 2015.
- [2] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*, pages 382–398. Springer, 2016.
- [3] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and vqa. *arXiv preprint arXiv:1707.07998*, 2017.
- [4] Stanislaw Antol, C Lawrence Zitnick, and Devi Parikh. Zero-shot learning via visual abstraction. In *European Conference on Computer Vision*, pages 401–416. Springer, 2014.
- [5] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. *arXiv preprint arXiv:1704.05796*, 2017.
- [6] David Belanger and Andrew McCallum. Structured prediction energy networks. *arXiv preprint arXiv:1511.06350*, 2015.
- [7] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1171–1179, 2015.

-
- [8] Alexander C Berg, Tamara L Berg, Hal Daume, Jesse Dodge, Amit Goyal, Xufeng Han, Alyssa Mensch, Margaret Mitchell, Aneesh Sood, Karl Stratos, et al. Understanding and predicting importance in images. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3562–3569. IEEE, 2012.
- [9] Andrew Carnie. *Syntax: A generative introduction*. John Wiley & Sons, 2013.
- [10] Pierre-Antoine Champin and Christine Solnon. Measuring the similarity of labeled graphs. In *International Conference on Case-Based Reasoning*, pages 80–95. Springer, 2003.
- [11] Angel X Chang, Manolis Savva, and Christopher D Manning. Semantic parsing for text to 3d scene generation. *ACL 2014*, page 17, 2014.
- [12] Liang-Chieh Chen, Alexander G Schwing, Alan L Yuille, and Raquel Urtasun. Learning deep structured models. In *Proc. ICML, 2015*.
- [13] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
- [14] Myung Jin Choi, Joseph J Lim, Antonio Torralba, and Alan S Willsky. Exploiting hierarchical context on a large database of object categories. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 129–136. IEEE, 2010.
- [15] Wongun Choi, Yu-Wei Chao, Caroline Pantofaru, and Silvio Savarese. Understanding indoor scenes using 3d geometric phrases. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 33–40, 2013.
- [16] Bo Dai, Sanja Fidler, Raquel Urtasun, and Dahua Lin. Towards diverse and natural image descriptions via a conditional gan. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.
- [17] Bo Dai, Yuqi Zhang, and Dahua Lin. Detecting visual relationships with deep relational networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

-
- [18] Pradipto Das, Chenliang Xu, Richard F Doell, and Jason J Corso. A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2634–2641, 2013.
- [19] Jacob Devlin, Hao Cheng, Hao Fang, Saurabh Gupta, Li Deng, Xiaodong He, Geoffrey Zweig, and Margaret Mitchell. Language models for image captioning: The quirks and what works. *arXiv preprint arXiv:1505.01809*, 2015.
- [20] Jacob Devlin, Saurabh Gupta, Ross Girshick, Margaret Mitchell, and C Lawrence Zitnick. Exploring nearest neighbor approaches for image captioning. *arXiv preprint arXiv:1505.04467*, 2015.
- [21] Yanzhuo Ding, Yang Liu, Huanbo Luan, and Maosong Sun. Visualizing and understanding neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1150–1159, 2017.
- [22] Santosh K Divvala, Ali Farhadi, and Carlos Guestrin. Learning everything about anything: Webly-supervised visual concept learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3270–3277, 2014.
- [23] Mohamed Elhoseiny, Scott Cohen, Walter Chang, Brian Price, and Ahmed Elgammal. Sherlock: Scalable fact learning in images. *arXiv preprint arXiv:1511.04891*, 2015.
- [24] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al. From captions to visual concepts and back. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1473–1482, 2015.
- [25] Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. Every picture tells a story: Generating sentences from images. In *European conference on computer vision*, pages 15–29. Springer, 2010.

-
- [26] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.
- [27] Sanja Fidler and Ales Leonardis. Towards scalable representations of object categories: Learning a hierarchy of parts. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.
- [28] Xinyu Fu, Eugene Ch’ng, Uwe Aickelin, and Simon See. Crnn: a joint neural network for redundancy detection. In *Smart Computing (SMARTCOMP), 2017 IEEE International Conference on*, pages 1–8. IEEE, 2017.
- [29] Carolina Galleguillos and Serge Belongie. Context based object categorization: A critical survey. *Computer Vision and Image Understanding*, 114(6):712–722, 2010.
- [30] Carolina Galleguillos, Andrew Rabinovich, and Serge Belongie. Object categorization using co-occurrence, location and appearance. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.
- [31] Georgia Gkioxari, Ross Girshick, and Jitendra Malik. Contextual action recognition with r* cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1080–1088, 2015.
- [32] Yunchao Gong, Qifa Ke, Michael Isard, and Svetlana Lazebnik. A multi-view embedding space for modeling internet images, tags, and their semantics. *International journal of computer vision*, 106(2):210–233, 2014.
- [33] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, page 2672–2680, 2014.
- [34] Stephen Gould, Jim Rodgers, David Cohen, Gal Elidan, and Daphne Koller. Multi-class segmentation with relative location prior. *International Journal of Computer Vision*, 80(3):300–316, 2008.
- [35] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

-
- [36] Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. Multi-dimensional recurrent neural networks. In *ICANN (1)*, pages 549–558, 2007.
- [37] Sergio Guadarrama, Niveda Krishnamoorthy, Girish Malkarnenkar, Subhashini Venugopalan, Raymond Mooney, Trevor Darrell, and Kate Saenko. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2712–2719, 2013.
- [38] Abhinav Gupta and Larry S Davis. Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In *European conference on computer vision*, pages 16–29. Springer, 2008.
- [39] Michael U Gutmann and Aapo Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13(Feb):307–361, 2012.
- [40] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.
- [41] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.
- [42] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [43] Derek Hoiem, Alexei A Efros, and Martial Hebert. Putting objects in perspective. *International Journal of Computer Vision*, 80(1):3–15, 2008.
- [44] Yan Huang, Wei Wang, and Liang Wang. Bidirectional recurrent convolutional networks for multi-frame super-resolution. In *Advances in Neural Information Processing Systems*, pages 235–243, 2015.
- [45] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016.

-
- [46] Hamid Izadinia, Fereshteh Sadeghi, and Ali Farhadi. Incorporating scene context and object layout into appearance modeling. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 232–239. IEEE, 2014.
- [47] Mainak Jas and Devi Parikh. Image specificity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2727–2736, 2015.
- [48] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.
- [49] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David A Shamma, Michael S Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3668–3678. IEEE, 2015.
- [50] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015.
- [51] Andrej Karpathy, Justin Johnson, and Li Fei-Fei. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*, 2015.
- [52] Andrej Karpathy, Armand Joulin, and Fei Fei F Li. Deep fragment embeddings for bidirectional image sentence mapping. In *Advances in neural information processing systems*, page 1889–1897, 2014.
- [53] Gil Keren and Björn Schuller. Convolutional rnn: an enhanced model for extracting features from sequential data. In *Neural Networks (IJCNN), 2016 International Joint Conference on*, pages 3412–3419. IEEE, 2016.
- [54] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [55] Dan Klein and Christopher D Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st annual meeting of the association for computational linguistics*, 2003.

-
- [56] Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *Adv. Neural Inf. Process. Syst*, 2011.
- [57] Jonathan Krause, Justin Johnson, Ranjay Krishna, and Li Fei-Fei. A hierarchical approach for generating descriptive image paragraphs. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [58] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016.
- [59] Girish Kulkarni, Visruth Premraj, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Baby talk: Understanding and generating image descriptions. In *Proceedings of the 24th CVPR*. Citeseer, 2011.
- [60] Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Babytalk: Understanding and generating simple image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2891–2903, 2013.
- [61] M Pawan Kumar and Daphne Koller. Efficiently selecting regions for scene understanding. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3217–3224. IEEE, 2010.
- [62] Polina Kuznetsova, Vicente Ordonez, Alexander C Berg, Tamara L Berg, and Yejin Choi. Collective generation of natural image descriptions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 359–368. Association for Computational Linguistics, 2012.
- [63] Polina Kuznetsova, Vicente Ordonez, Tamara L Berg, and Yejin Choi. Treetalk: Composition and compression of trees for image descriptions. *TACL*, 2(10):351–362, 2014.
- [64] Lubor Ladicky, Chris Russell, Pushmeet Kohli, and Philip HS Torr. Graph cut based inference with co-occurrence statistics. In *European Conference on Computer Vision*, pages 239–253. Springer, 2010.

-
- [65] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.
- [66] Michael Denkowski Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. *ACL 2014*, page 376, 2014.
- [67] Rémi Lebret, Pedro O Pinheiro, and Ronan Collobert. Simple image description generator via a linear phrase-based approach. *arXiv preprint arXiv:1412.8419*, 2014.
- [68] Siming Li, Girish Kulkarni, Tamara L Berg, Alexander C Berg, and Yejin Choi. Composing simple image descriptions using web-scale n-grams. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 220–228. Association for Computational Linguistics, 2011.
- [69] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and region captions. In *ICCV*, 2017.
- [70] Zhenyang Li, Kirill Gavriluk, Efstratios Gavves, Mihir Jain, and Cees GM Snoek. Videolstm convolves, attends and flows for action recognition. *Computer Vision and Image Understanding*, 2017.
- [71] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8. Barcelona, Spain, 2004.
- [72] Dahua Lin, Sanja Fidler, Chen Kong, and Raquel Urtasun. Generating multi-sentence lingual descriptions of indoor scenes. In *BMVC*, 2015.
- [73] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.
- [74] Huan Ling and Sanja Fidler. Teaching machines to describe images via natural language feedback. In *NIPS*, 2017.
- [75] Chenxi Liu, Junhua Mao, Fei Sha, and Alan L Yuille. Attention correctness in neural image captioning. In *AAAI*, pages 4176–4182, 2017.

-
- [76] Siqu Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. Optimization of image description metrics using policy gradient methods. *arXiv preprint arXiv:1612.00370*, 2016.
- [77] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. *arXiv preprint arXiv:1608.00187*, 2016.
- [78] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. *arXiv preprint arXiv:1612.01887*, 2016.
- [79] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.
- [80] Christopher D Manning and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.
- [81] 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.
- [82] Thomas Mensink, Efstratios Gavves, and Cees GM Snoek. Costa: Co-occurrence statistics for zero-shot classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2441–2448, 2014.
- [83] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.
- [84] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

-
- [85] Cesc Chunseong Park, Byeongchang Kim, and Gunhee Kim. Attend to you: Personalized image captioning with context sequence memory networks. In *CVPR*, 2017.
- [86] Judea Pearl. Probabilistic reasoning in intelligent systems: Networks of plausible reasoning, 1988.
- [87] Slav Petrov and Dan Klein. Improved inference for unlexicalized parsing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 404–411, 2007.
- [88] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 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, 2015.
- [89] Ariadna Quattoni, Michael Collins, and Trevor Darrell. Conditional random fields for object recognition. In *Advances in neural information processing systems*, pages 1097–1104, 2004.
- [90] Andrew Rabinovich, Andrea Vedaldi, Carolina Galleguillos, Eric Wiewiora, and Serge Belongie. Objects in context. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.
- [91] Vignesh Ramanathan, Congcong Li, Jia Deng, Wei Han, Zhen Li, Kunlong Gu, Yang Song, Samy Bengio, Chuck Rossenber, and Li Fei-Fei. Learning semantic relationships for better action retrieval in images. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1100–1109. IEEE, 2015.
- [92] Scott Reed, Zeynep Akata, Xinchun Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 3, 2016.
- [93] Michaela Regneri, Marcus Rohrbach, Dominikus Wetzels, Stefan Thater, Bernt Schiele, and Manfred Pinkal. Grounding action descriptions in videos. *Transactions of the Association for Computational Linguistics*, 1:25–36, 2013.

-
- [94] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.
- [95] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. *arXiv preprint arXiv:1612.00563*, 2016.
- [96] Anna Rohrbach, Marcus Rohrbach, Ronghang Hu, Trevor Darrell, and Bernt Schiele. Grounding of textual phrases in images by reconstruction. *arXiv preprint arXiv:1511.03745*, 2015.
- [97] Marcus Rohrbach, Wei Qiu, Ivan Titov, Stefan Thater, Manfred Pinkal, and Bernt Schiele. Translating video content to natural language descriptions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 433–440, 2013.
- [98] Bernardino Romera-Paredes and Philip Hilaire Sean Torr. Recurrent instance segmentation. In *European Conference on Computer Vision*, pages 312–329. Springer, 2016.
- [99] Chen Rui, Yang Jing, Hu Rong-gui, and Huang Shu-guang. A novel lstm-rnn decoding algorithm in captcha recognition. In *Instrumentation, Measurement, Computer, Communication and Control (IMCCC), 2013 Third International Conference on*, pages 766–771. IEEE, 2013.
- [100] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.
- [101] Bryan C Russell, William T Freeman, Alexei A Efros, Josef Sivic, and Andrew Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1605–1614. IEEE, 2006.

-
- [102] Mohammad Amin Sadeghi and Ali Farhadi. Recognition using visual phrases. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1745–1752. IEEE, 2011.
- [103] Ruslan Salakhutdinov, Antonio Torralba, and Josh Tenenbaum. Learning to share visual appearance for multiclass object detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1481–1488. IEEE, 2011.
- [104] Alexander G Schwing and Raquel Urtasun. Fully connected deep structured networks. *arXiv preprint arXiv:1503.02351*, 2015.
- [105] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.
- [106] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [107] Josef Sivic, Bryan C Russell, Alexei A Efros, Andrew Zisserman, and William T Freeman. Discovering objects and their location in images. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 370–377. IEEE, 2005.
- [108] Richard Socher, John Bauer, Christopher D Manning, et al. Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 455–465, 2013.
- [109] Richard S Sutton, David A McAllester, Satinder P Singh, Yishay Mansour, et al. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*, volume 99, pages 1057–1063, 1999.
- [110] Ying Hua Tan and Chee Seng Chan. phi-lstm: a phrase-based hierarchical lstm model for image captioning. In *Asian Conference on Computer Vision*, pages 101–117. Springer, 2016.
- [111] Jesse Thomason, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Raymond J Mooney. Integrating language and vision to generate natural language descriptions of videos in the wild. In *COLING*, volume 2, page 9, 2014.

-
- [112] Ramakrishna Vedantam, Samy Bengio, Kevin Murphy, Devi Parikh, and Gal Chechik. Context-aware captions from context-agnostic supervision. *arXiv preprint arXiv:1701.02870*, 2017.
- [113] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4566–4575, 2015.
- [114] Ramakrishna Vedantam, Xiao Lin, Tanmay Batra, C Lawrence Zitnick, and Devi Parikh. Learning common sense through visual abstraction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2542–2550, 2015.
- [115] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2015.
- [116] Chenglong Wang, Feijun Jiang, and Hongxia Yang. A hybrid framework for text modeling with convolutional rnn. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2061–2069. ACM, 2017.
- [117] Yufei Wang, Zhe Lin, Xiaohui Shen, Scott Cohen, and Garrison W Cottrell. Skeleton key: Image captioning by skeleton-attribute decomposition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7272–7281, 2017.
- [118] Qi Wu, Damien Teney, Peng Wang, Chunhua Shen, Anthony Dick, and Anton van den Hengel. Visual question answering: A survey of methods and datasets. *arXiv preprint arXiv:1607.05910*, 2016.
- [119] Zhirong Wu, Dahua Lin, and Xiaoou Tang. Deep markov random field for image modeling. In *European Conference on Computer Vision*, pages 295–312. Springer, 2016.
- [120] SHI Xingjian, Zhouong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for

- precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015.
- [121] Yuanjun Xiong, Kai Zhu, Dahua Lin, and Xiaoou Tang. Recognize complex events from static images by fusing deep channels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1600–1609, 2015.
- [122] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C Courville, Ruslan Salakhutdinov, Richard S Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, volume 14, pages 77–81, 2015.
- [123] Zhilin Yang, Ye Yuan, Yuexin Wu, William W Cohen, and Ruslan R Salakhutdinov. Review networks for caption generation. In *Advances in Neural Information Processing Systems*, pages 2361–2369, 2016.
- [124] Bangpeng Yao and Li Fei-Fei. Grouplet: A structured image representation for recognizing human and object interactions. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, page 9–16. IEEE, 2010.
- [125] Jian Yao, Sanja Fidler, and Raquel Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 702–709. IEEE, 2012.
- [126] Ting Yao, Yingwei Pan, Yehao Li, Zhaofan Qiu, and Tao Mei. Boosting image captioning with attributes. *arXiv preprint arXiv:1611.01646*, 2016.
- [127] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4651–4659, 2016.
- [128] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.

-
- [129] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: sequence generative adversarial nets with policy gradient. *arXiv preprint arXiv:1609.05473*, 2016.
- [130] Licheng Yu, Eunbyung Park, Alexander C Berg, and Tamara L Berg. Visual madlibs: Fill in the blank description generation and question answering. In *Computer Vision (ICCV), 2015 IEEE International Conference on*, pages 2461–2469. IEEE, 2015.
- [131] Ning Zhang, Manohar Paluri, Marc’Aurelio Ranzato, Trevor Darrell, and Lubomir Bourdev. Panda: Pose aligned networks for deep attribute modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1637–1644, 2014.
- [132] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1529–1537, 2015.
- [133] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Antonio Torralba, and Aude Oliva. Places: An image database for deep scene understanding. *arXiv preprint arXiv:1610.02055*, 2016.
- [134] Luowei Zhou, Chenliang Xu, Parker Koch, and Jason J Corso. Image caption generation with text-conditional semantic attention. *arXiv preprint arXiv:1606.04621*, 2016.
- [135] C Lawrence Zitnick, Devi Parikh, and Lucy Vanderwende. Learning the visual interpretation of sentences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1681–1688, 2013.
- [136] Zhen Zuo, Bing Shuai, Gang Wang, Xiao Liu, Xingxing Wang, Bing Wang, and Yushi Chen. Convolutional recurrent neural networks: Learning spatial dependencies for image representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 18–26, 2015.