

Learning Local Representations of Images and Text

A Dissertation

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

in partial fulfillment

of the requirements for the degree

Doctor of Philosophy

by

Fuwen Tan

May

2021

APPROVAL SHEET

The dissertation
is submitted in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

Fuwen Tan

AUTHOR

The dissertation has been read and approved by the examining committee:

Vicente Ordóñez Román (Advisor)

Yanjun Qi

Yangfeng Ji

Mona Kasra

Ming-Hsuan Yang

Connelly Barnes

Accepted for the School of Engineering and Applied Science:

A handwritten signature in black ink, appearing to read 'CHB', with a stylized, flowing script.

Craig H. Benson, Dean, School of Engineering and Applied Science

May
2021

Abstract

Images and text inherently exhibit hierarchical structures, e.g. scenes built from objects, sentences built from words. In many computer vision and natural language processing tasks, learning accurate prediction models requires analyzing the correlation of the local primitives of both the input and output data. In this thesis, we develop techniques for learning local representations of images and text and demonstrate their effectiveness on visual recognition, retrieval, and synthesis. In particular, the thesis includes three primary research projects:

In the first project, we explore the benefits of learning compositional image representation for text-to-image generation. The latest text-to-image generation research is dominated by Generative Adversarial Network (GAN) based methods, which predicts pixel-wise intensity values. While demonstrating remarkable results, these methods still have difficulties generating complex scenes with multiple interacting objects. In this work, we propose to model the local structures instead of the raw pixel values of the images. We develop a sequence-to-sequence image synthesis framework that produces a scene depicted in a textual description by sequentially predicting objects, their locations, and their attributes such as sizes, aspect ratios. Compared to previous GAN-based approaches, our method achieves competitive or superior performance while producing more interpretable results.

In the second project, we show the advantage of learning compositional text representations for interactive image search using multiple rounds of text queries. Cross-modal image search is a well-studied research topic where most of the recent approaches focus on learning a linear embedding space of the visual and textual data. We observe that this global representation cannot distinguish object instances that share the same feature space. Thus we propose an effective framework that encodes multiple rounds of natural language queries with a region-aware state representation

and show that it outperforms existing sequential encoding and embedding models on both the simulated and real user queries.

In the third project, we focus on learning the visual relation of an image-pair in the context of reranking image search results for instance image recognition. We propose a lightweight and straightforward pipeline that learns to predict the similarity of an image-pair directly. The key ingredient of this work is a transformer-based architecture that models the interactions between the global/local descriptors within the individual image and across the image pair. Our experiments show that the proposed method outperforms previous approaches while using much fewer local descriptors. It can also be jointly optimized with the feature extractor, leading to further accuracy improvement.

Acknowledgements

Preparation of this thesis would not have been possible without the support and help from so many people.

I would like to express my sincere thanks to my advisor, Dr. Vicente Ordóñez Román, for his continuous guidance and support. Three years ago I started working on my thesis research while being a novice in Vision and Language. Vicente has spent great effort guiding me throughout my doctoral thesis, from defining projects, designing experiments, to presenting the research findings. He always encourages me to pursue research with high impact and conduct research with high quality. I have been extremely fortunate to have him as my mentor in my research and career over the past years.

My gratitude also goes to my other thesis committee members, Yanjun Qi, Yangfeng Ji, Mona Kasra, Ming-Hsuan Yang, and Connelly Barnes for their time and valuable feedback. Particularly, I would like to acknowledge Connelly Barnes for supporting me in the early stage of my Ph.D. process, equipping me with solid research skills, and Yanjun Qi for her remarkable class/seminar that enhances my understanding of Machine Learning, and for chairing my Ph.D. committee.

I also want to thank my industrial collaborators: Song Feng, Xiaoxiao Guo, Hui Wu in IBM Research, and Jiangbo Yuan in eBay Research. Most, if not all, of this thesis, was completed through these fruitful collaborations. Special thanks go to Song Feng who provided tremendous help when I published my first top-tier paper, e.g. creating time for weekly technical discussions, performing evaluation experiments, reading/revising my last-minute draft. I am also thankful to the hosts of my three summer internships, Bernd Heisele in Honda Research Institute, Tony Hwang and Sundar Vedula in Amazon A9, Vlad Morariu and Tong Sun in Adobe, for offering me these memorable experiences on the east/west coasts.

Many thanks to all my lab-mates and collaborators at the University of Virginia: Paola Cascante-Bonilla, Tianlu Wang, Ziyang Yang, Leticia Pinto-Alva, Aman Shrivastava, and many more that I cannot list all here. It has been a great pleasure working with you in this beautiful town.

Finally, I want to thank my parents for their unconditional support in everything. It would be impossible for me to finish my Ph.D. study without your encouragement and love.

This work was supported in part by generous gifts from IBM, SAP, and eBay.

To my family

Contents

List of Figures	ix
List of Tables	xi
1 Introduction	1
1.1 Motivation	1
1.2 Local Representations in Vision and Language	4
1.3 Outline of This Thesis	5
2 Learning Compositional Image Representations for Text-to-image	
Synthesis	8
2.1 Introduction	8
2.2 Related Work	12
2.3 Text2Scene	14
2.3.1 Text Encoder	15
2.3.2 Object and Attribute Decoders	15
2.3.3 Foreground Patch Embedding	17
2.3.4 Objective	18
2.4 Experiments	19
2.4.1 Network Architecture	20
2.4.2 Optimization	26

2.4.3	Evaluation	26
2.4.4	Results and Discussion	30
2.5	Summary	35
3	Learning Compositional Text Representations for Interactive Image Retrieval	36
3.1	Introduction	36
3.2	Related Work	39
3.3	Method	41
3.3.1	Image Representation	42
3.3.2	Query Representation	42
3.3.3	Cross Modal Similarity	43
3.3.4	Query Encoding	44
3.3.5	End-to-end Training	45
3.4	Experiments	47
3.4.1	Results on Simulated User Queries	50
3.4.2	Results on Real User Queries	52
3.5	Summary	54
4	Learning Visual Similarity of Images using Reranking Transformers	55
4.1	Introduction	55
4.2	Related Work	59
4.3	Method	61
4.3.1	Background	61
4.3.2	Attention Modules in Transformers	62
4.3.3	Model	63
4.4	Experiments	65
4.4.1	Datasets	65

4.4.2	Implementation	66
4.5	Results	69
4.5.1	Baselines	69
4.5.2	Comparison with Geometric Verification	70
4.5.3	Ablation on the Number of Local Descriptors	72
4.5.4	Comparison with Query Expansion	73
4.5.5	Comparison with Aggregated Selective Match Kernel (ASMK)	75
4.5.6	Feature Learning & RRT: Joint Optimization	76
4.5.7	Comparison with the State-of-the-Art	78
4.5.8	Limitation	78
4.5.9	Qualitative Examples	80
4.6	Summary	80
5	Conclusion	83

List of Figures

1.1	Local primitives of images and text	3
1.2	Overview of the Text2Scene model	5
1.3	The interactive image retrieval task	6
1.4	The instance image retrieval pipeline	7
2.1	Example inputs and outputs of the Text2Scene model and the ground-truth	9
2.2	A challenging case for the GAN-based text-to-image synthesis methods	10
2.3	Step-by-step generation of an abstract scene by Text2Scene	13
2.4	An illustration of the Text2Scene model	14
2.5	Evaluation metrics used by Text2Scene for the abstract scene generation task	26
2.6	User interfaces of the human subject studies for Text2Scene	28
2.7	Examples of abstract scenes generated by Text2Scene	30
2.8	Examples of object layouts generated by Text2Scene	32
2.9	Examples of synthetic images generated by Text2Scene	33
2.10	Examples of synthetic images generated by Text2Scene, and the source images from which the patch segments are retrieved for compositing .	34
2.11	Comparing Text2Scene with state-of-the-art approaches on an uncommon example	35

3.1	An example of the interactive image retrieval with the Drill-down model	37
3.2	Overview of the Drill-down model	41
3.3	Quantitative evaluation of the Drill-down model	49
3.4	Qualitative examples of Drill-down _{3×128}	52
3.5	Examples of real user queries and the top-1 images from Drill-down _{3×256} .	53
3.6	Human subject evaluation of Drill-down	54
4.1	Reranking for instance image recognition	58
4.2	Illustration of the Reranking Transformer model	62
4.3	An example showcasing the situation where the global descriptor + cosine similarity retrieval paradigm is not ideal	69
4.4	Qualitative examples of the Reranking Transformer on Revisited Ox- ford/Paris	79
4.5	Qualitative examples of the Reranking Transformer on Stanford Online Products	81
4.6	Qualitative examples comparing the Reranking Transformer with ge- ometry verification on Revisited Oxford/Paris	82

List of Tables

2.1	Architecture of our scene encoder in Text2Scene for layout generation	22
2.2	Architecture of our scene encoder in Text2Scene for synthetic image generation	23
2.3	Architectures for the object and attribute decoders in Text2Scene . .	24
2.4	Architecture of our foreground patch embedding network for synthetic image generation	25
2.5	Quantitative evaluation of Text2Scene on Abstract Scenes	29
2.6	Human evaluation of Text2Scene on Abstract Scenes	29
2.7	Quantitative evaluation of Text2Scene on layout generation	31
2.8	Quantitative evaluation of Text2Scene on synthetic image generation	31
2.9	Human evaluation of Text2Scene on synthetic image generation . . .	32
3.1	Sizes of the query/image representations and the parameters in the Drill-down model and the baselines	50
4.1	Comparing the Reranking Transformer with geometric verification on Revisited Oxford/Paris	71
4.2	Comparing the Reranking Transformer with geometric verification on GLDv2	71
4.3	Ablation on the number of local descriptors used for the Reranking Transformer and geometry verification	73

4.4	Comparing the Reranking Transformer with α QE on Revisited Oxford/Paris	74
4.5	Comparing the Reranking Transformer with ASMK on Revisited Oxford/Paris	75
4.6	Jointly optimizing the feature extractor and the Reranking Transformer on SOP	76
4.7	Comparing the Reranking Transformer with the state-of-the-art methods on Revisited Oxford/Paris	78

Chapter 1

Introduction

1.1 Motivation

Representation learning is a core problem in computer vision and natural language processing. The goal is to learn an expressive representation of the input data that can facilitate the inference stage of the downstream task. As can be interpreted from this definition, the right representation is highly correlated with the input data and the inference model. Pioneering recognition systems in computer vision (CV) and natural language processing (NLP) do not make full use of such correlations but rely on combining hand-crafted features (e.g. SIFT in CV, TF-IDF in NLP) with separably optimized prediction models. With the advent of deep learning, joint optimizing the feature representation and the inference model has become a dominant solution. It is shown that, with sufficient annotated data and computing resources, these end-to-end learnable representations significantly outperform the early hand-crafted features, especially when the inference involves only simple (e.g. linear) classifications. As evidence, the first deep learning-based visual recognition model, AlexNet [66], achieved a top-1 error of 37.5% on the ILSVRC challenge [30], a 20% error reduction from the previous state-of-the-art (with a top-1 error of 47.1%).

The success of deep learned features has further been demonstrated on various visual and linguistic recognition tasks, e.g. object detection [102], natural language inference [31], visual question answering [4], etc, although conceptually the solution is straightforward: encoding the raw images/text into vectorized representations using neural networks (e.g. convolutional neural network [66], recurrent network [24], transformer [130]), and making predictions on top of these representations. This solution is usually implemented as a black-box framework and deployed widely on diverse tasks. The key magic is a large amount of annotations and computational resources. It is widely believed that [119, 147] given sufficient labeled data, a large (usually overparameterized) neural network could learn to “memorize” the distribution of the supervised data, with no bells and whistles. This is especially true when the inference is a simple linear classification. For example, the state-of-the-art visual classification model, Vision Transformer [33], has 632 million parameters, and was trained on a large dataset of over 300 million image-label pairs [119]. The training of this model requires 2500 TPUv3-core-days. The state-of-the-art language model, GPT-3 [14], has 175 billion parameters, which is even more computationally prohibited. The need for large labeled datasets and large models hinders the deployment of these supervised representations on domains where annotations/demonstrations are sparse and not easy to collect (e.g. medical applications).

On the other hand, when the desired output of the model is beyond the class label but exhibits complex structures, e.g. sentences or images, the benefit of these general feature representations becomes unclear. On the vision-language tasks, it is shown in [44] that the performance of random vectors is surprisingly closed to the learned embeddings [93] when enough labeled data are present. On the image matching task, [55] demonstrates that, with proper hyperparameter settings, classical solutions such as SIFT [77] can still outperform the learned features. For these “rich” prediction problems, we believe the generally learned features are sub-optimal as they do not

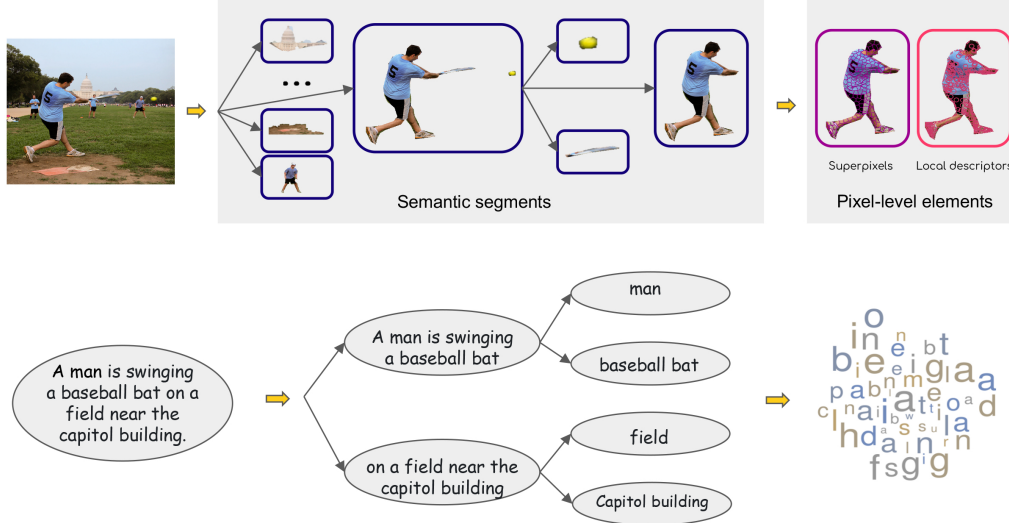


Figure 1.1: Local primitives of images and text. Complex visual scenes can be modeled as the composition of semantic segments or pixel-level elements. Complex linguistic expressions can be modeled as the composition of phrases, words or characters.

explicitly model the interaction between the local primitives of the structured input and output. This also leads to another drawback: with the black-box pipeline, the inference is less interpretable.

In this thesis, we aim to address the problem of learning image and text representations for various vision and language tasks. As we observe, one feature shared by the visual and textual data is the capacity of representing complex concepts using a hierarchy of local primitives. As shown in Fig. 1.1, both images and text can be represented by different forms of constituent elements, e.g. characters/words for text, points/segments for images. We believe that explicitly learning the representations of these local concepts can ease the inference stage, especially for “rich” visual predictions where both the input and output data comprises complex structures. Furthermore, it may also lead to more compact model architectures [122] and feature representations [121] that require less labeled data to train. Compared to the traditional approaches, modeling the local concepts of the input and output data and their interactions could also lead to more interpretable predictions. My PhD research

demonstrates that the efficiency and effectiveness of the local representations on various “rich” vision and language tasks, e.g. text-to-image synthesis [122], interactive image retrieval [121], instance-level image recognition [123].

1.2 Local Representations in Vision and Language

Local features have been one of the most common representations in computer vision. Much of the progress for visual recognition before the “deep learning revolution” has built on local, or keypoint-based descriptors such as SIFT [77] and HOG [27]. Compared to global signatures [89], these descriptors are believed to be more invariant to image changes such as illumination, translation, occlusion, and truncation. They were used in a wide variety of visual prediction tasks such as texture recognition [70], scene recognition [68], image matching [55], 3D reconstruction [1], etc. Part-based features [107, 146] were later introduced to model semantic visual concepts, e.g. classes, attributes, and relations. They were typically used in combination with statistical models for both pure visual recognition tasks and vision-and-language tasks. Famous examples include the Deformable Part Model [36] for object detection and the BabyTalk [67] system for image captioning. With the popularity of deep learning, local representations extracted from Convolutional Neural Networks play an important role in building high-capacity visual prediction models. Both grid-based [54] and region-based [3] features have shown promising performance on many vision-and-language tasks, e.g. image captioning [141], visual question answering [4], referring expression grounding [58], etc.

Compositional (part-based) representations have also been widely studied in linguistics and adjacent fields. Early systems [84, 114] explore incorporating explicit composition operations into vector-based systems. More recent approaches focus on learning distributed representations of natural language from large text corpora.

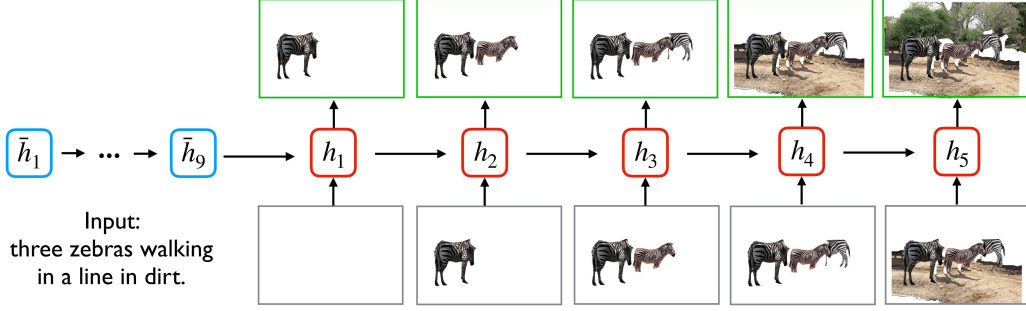


Figure 1.2: Overview of the Text2Scene model that sequentially produces a composite image from an input sentence.

Word2Vec [83] and GloVe [93] proposed to model the co-occurrence of words in both local and global contexts, while Socher et al [115] developed a unified framework to learn the hierarchy of words, phrases, and sentences using recursive neural networks. As a recent breakthrough in natural language processing, the Transformer [80] model learned contextualized text representations using a novel attention-based sequence encoder.

1.3 Outline of This Thesis

In this thesis, we explore the local representations of images and text and their applications in three research projects, one chapter for each.

In Chapter 2, we introduce Text2Scene, a sequence-to-sequence model that generates various forms of images from natural language descriptions. Unlike recent works that rely on Generative Adversarial Networks (GANs) [40] to generate pixel-wise intensity values, we propose to learn a compositional (part-based) representation of the image. Text2Scene sequentially generates objects and their attributes (location, size, appearance, etc) by attending to different parts of the input text and the current status of the generated scene. Fig. 1.2 provides an overview of the Text2Scene model. Compared to the state-of-the-art approaches [100, 148], the pro-

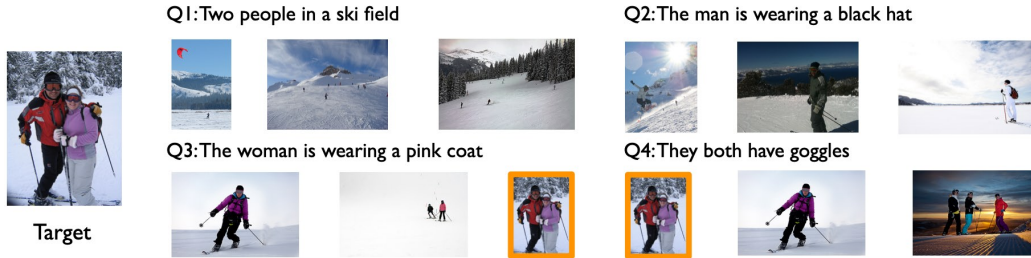


Figure 1.3: An example of retrieving a target image using our Drill-down model [121]. The user progressively provides four rounds of text queries to refine the retrieval results.

posed generation pipeline is more interpretable and data-efficient. We demonstrate that Text2Scene can handle the generation of different forms of scene representations, including cartoon-like scenes, bounding-box based scene layouts, and composite images with superior performance on both automatic and human evaluations. This work was published in CVPR 2019 [122];

In Chapter 3, we focus on learning compositional representations of text. Particularly, we study the task of interactive image retrieval using natural language queries, where a user progressively provides input queries to refine a set of retrieval results. The key challenge of this task is how to integrate multiple rounds of text queries. While most of the previous approaches leverage task-agnostic text representations, e.g. the hidden state from a recurrent neural network, we propose Drill-down, an effective framework that learns a region-aware text representation that significantly extends previous methods. The proposed representation has an extra dimension that can help distinguish object instances from the same category, while still maintaining small memory/computational budgets. Fig. 1.3 shows an example of retrieving a target image using our Drill-down model. We compare our method with existing sequential encoding and embedding networks, demonstrating superior performance on two proposed benchmarks: automatic image retrieval on a simulated scenario that uses region captions as queries, and interactive image retrieval using real queries from human evaluators. This work was published in NeurIPS 2019 [121].

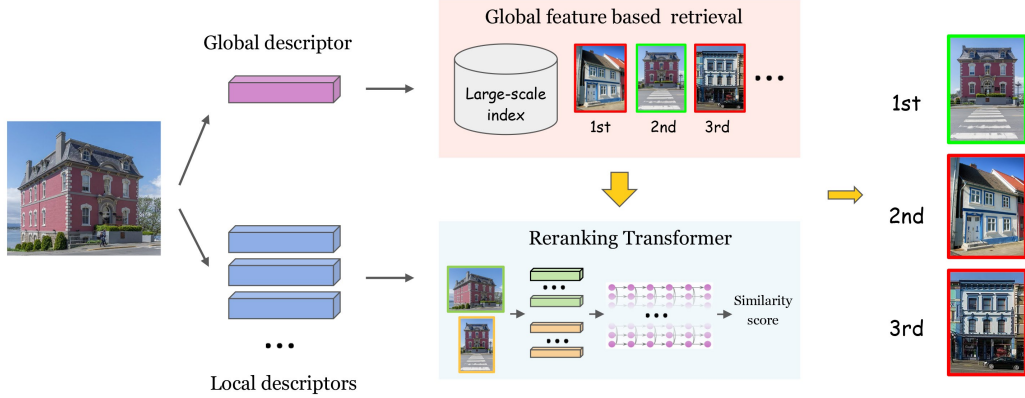


Figure 1.4: Overview of the instance retrieval pipeline using our Reranking Transformers [123]: we first perform global feature based retrieval to obtain the initial ranks of the candidate images, then use the proposed Reranking Transformer to refine the top-ranked images.

In Chapter 4, we explore learning the visual similarity of an image pair for instance-level image recognition. The goal of the task is to search in a large database for images that match a specific object/scene instance in a query image. To address this task, early systems typically perform a global retrieval step to reduce the search space, and a local refinement step that performs domain-specific reranking by leveraging operations such as geometric verification. While these works have managed to match images that are sufficiently similar, they still have difficulty handling challenging cases, such as large viewpoint variations. In this work, we propose Reranking Transformers (RRTs) [123] as a lightweight model to learn the matching images (Fig. 1.4). The key component of our approach is a transformer based architecture that models the correlations between the local structures of the image pair. We perform extensive experiments, demonstrating that the proposed approach outperforms prior reranking approaches while using much fewer descriptors. We also show that, unlike existing approaches, RRTs can be optimized jointly with the feature extractor, which can lead to feature representations tailored to downstream tasks and further accuracy improvements.

Chapter 2

Learning Compositional Image Representations for Text-to-image Synthesis

2.1 Introduction

As the first work in this thesis, we introduce Text2Scene [122], a model to interpret visually descriptive language in order to generate compositional scene representations. We specifically focus on generating a scene representation consisting of a list of objects, along with their attributes (e.g. location, size, aspect ratio, pose, appearance). We propose a unified framework to generate three types of scenes as shown in Figure 2.1, (1) Cartoon-like scenes as depicted in the Abstract Scenes dataset [154], (2) Object layouts corresponding to image scenes from the COCO dataset [74], and (3) Synthetic scenes corresponding to images in the COCO dataset [74].

Generating rich textual representations for scene generation is a challenging task. For instance, input textual descriptions could hint only indirectly at the presence of attributes – e.g. in the first example in Fig. 2.1 the input text “Mike is surprised”

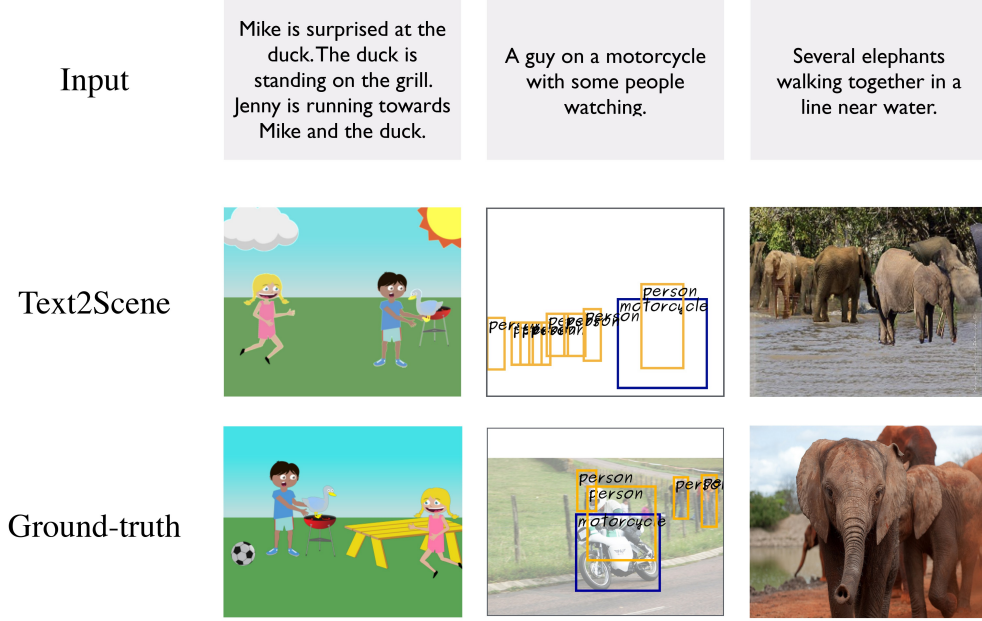


Figure 2.1: Sample inputs (top) and outputs of our Text2Scene model (middle), along with *ground truth* reference scenes (bottom) for the generation of abstract scenes (left), object layouts (middle), and synthetic image composites (right).

should change the facial attribute on the generated object “Mike”. Textual descriptions often have complex information about relative spatial configurations – e.g. in the first example in Fig. 2.1 the input text “Jenny is running towards Mike and the duck” makes the orientation of “Jenny” dependent on the positions of both “Mike”, and “duck”. In the last example in Fig. 2.1 the text “elephants walking together in a line” also implies a certain overall spatial configuration of the objects in the scene.

To address this problem, state-of-the-art approaches [51, 56, 100, 143, 148, 150] typically rely on Generative Adversarial Networks (GANs) [40], which have demonstrated impressive results on a number of image synthesis tasks, such as generating flowers [85] or birds [136]. However, our experiments show that, these GAN-based methods still struggle with complex scenes of multiple interacting objects. As shown in Fig. 2.2(A), the state-of-the-art approach AttnGAN [143] has difficulty in handling a caption like “a cat curled up on a skateboard in a living room”. We believe that it

A **cat** curled up on
a **skateboard** in a
living room.



(A)



Cat on **skateboard** ~3 examples in COCO

(B)

Figure 2.2: GAN-based text-to-image synthesis methods struggle with few-shot cases. (A) the state-of-the-art approach AttnGAN [143] has difficulty in handling the caption “a cat curled up on a skateboard in a living room”; (B) in the COCO [74] dataset, there are only three examples that capture a scene with a “cat” on a “skateboard”.

is because training a generative adversarial network or pixel-wise synthesis model in general requires a large amount of labeled data covering all the scenarios, which are not easy or even impossible to collect. For example, most of the GAN-based methods leverage the COCO dataset [74] as one of the main benchmarks. While the training split of COCO [74] has over 800 thousand images, there are only three examples that capture a scene with a “cat” on a “skateboard” (Fig. 2.2 (B)).

Our method, unlike recent approaches, does not rely on Generative Adversarial Networks (GANs) [40]. Instead, we produce an interpretable model that iteratively generates a scene by predicting and adding new objects at each time step. In particular, we leverage a sequence-to-sequence approach where objects are placed sequentially on an initially empty canvas (see an overview in Fig 2.4). Generally, Text2Scene, consists of a text encoder (Fig 2.4 (A)) that maps the input sentence to a set of latent representations, an image encoder (Fig 2.4 (B)) which encodes the current generated canvas, a convolutional recurrent module (Fig 2.4 (C)), which passes the current state

to the next step, attention modules (Fig 2.4 (D)) which focus on different parts of the input text, an object decoder (Fig 2.4 (E)) that predicts the next object conditioned on the current scene state and attended input text, and an attribute decoder (Fig 2.4 (F)) that assigns attributes to the predicted object. To the best of our knowledge, Text2Scene is the first model demonstrating its capacities on both abstract and real images, thus opening the possibility for future work on transfer learning across domains. Text2Scene is superior to the best result reported in Abstract Scenes [154], and provides near state-of-the-art performance on COCO [74] under automatic evaluation metrics, and state-of-the-art results when evaluated by humans. Compared to the GAN based approaches, Text2Scene is shown to be able to generalize to uncommon situations illustrated in Fig. 2.2 and has a more interpretable generation pipeline (Fig. 2.3).

Our main contributions can be summarized as follows:

- We propose Text2Scene, a framework to generate compositional scene representations from natural language descriptions.
- We show that Text2Scene can be used to generate, under minor modifications, different forms of scene representations, including cartoon-like scenes, semantic layouts corresponding to real images, and synthetic image composites.
- We conduct extensive experiments on the tasks of abstract image generation for the Abstract Scenes dataset [154], semantic layout and synthetic image generations for the COCO dataset [74]. We show that, compared to state-of-the-art approaches, Text2Scene achieves superior performance while delivering more interpretable results.

2.2 Related Work

Most of the recent approaches in text-to-image synthesis [51, 56, 100, 101, 143, 148, 150] have leveraged conditional Generative Adversarial Networks (GANs). While these works have managed to generate results of increasing quality, there are major challenges when attempting to synthesize images for complex scenes with multiple interacting objects without explicitly defining such interactions [144]. Inspired by *the principle of compositionality* [152], our model does not use GANs but produces a scene by sequentially generating objects (e.g. in the forms of clip-arts, bounding boxes, or segmented object patches) containing the semantic elements that compose the scene.

Our work is also related to prior research on using abstract scenes to mirror and analyze complex situations in the real world [39, 132, 153, 154]. In [154], a graphical model was introduced to generate an abstract scene from textual descriptions. Unlike this previous work, our method does not use a semantic parser but is trained end-to-end from input sentences. Our work is also related to recent research on generating images from pixel-wise semantic labels [22, 53, 96], especially [96] which proposed a retrieval-based semi-parametric method for image synthesis given the spatial semantic map. Our synthetic image generation model optionally uses the cascaded refinement module in [96] as a post-processing step. Unlike these works, our method is not given the spatial layout of the objects in the scene but learns to predict a layout indirectly from text.

Most closely related to our approach are [43, 51, 56], and [60], as these works also attempt to predict explicit 2D layout representations. Johnson et al [56] proposed a graph-convolutional model to generate images from structured scene graphs. The presented objects and their relationships were provided as inputs in the scene graphs, while in our work, the presence of objects is inferred from text. Hong et al [51] tar-

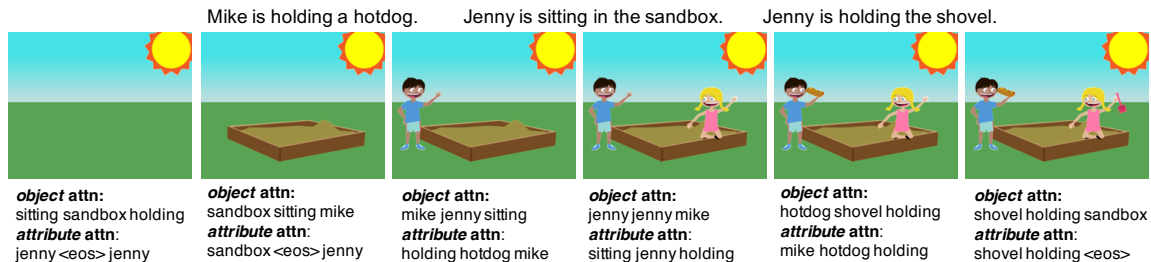


Figure 2.3: Step-by-step generation of an abstract scene, showing the top-3 attended words for the object prediction and attribute prediction at each time step. Notice how except for predicting the *sun* at the first time step, the top-1 attended words in the object decoder are almost one-to-one mappings with the predicted objects. The attended words by the attribute decoder also correspond semantically to useful information for predicting either pose or location, e.g. to predict the location of the *hotdog* at the fifth time step, the model attends to *mike* and *holding*.

geted image synthesis using conditional GANs but unlike prior works, it generated layouts as intermediate representations in a separably trained module. Our work also attempts to predict layouts for photographic image synthesis but unlike [51], we generate pixel-level outputs using a semi-parametric retrieval module without adversarial training and demonstrate superior results. Kim et al [60] performed pictorial generation from chat logs, while our work uses text which is considerably more under-specified. Gupta et al [43] proposed a semi-parametric method to generate cartoon-like pictures. However the presented objects were also provided as inputs to the model, and the predictions of layouts, foregrounds and backgrounds were performed by separably trained modules. Our method is trained end-to-end and goes beyond cartoon-like scenes. To the best of our knowledge, our model is the first work targeting various types of scenes (e.g. abstract scenes, semantic layouts and composite images) under a unified framework.

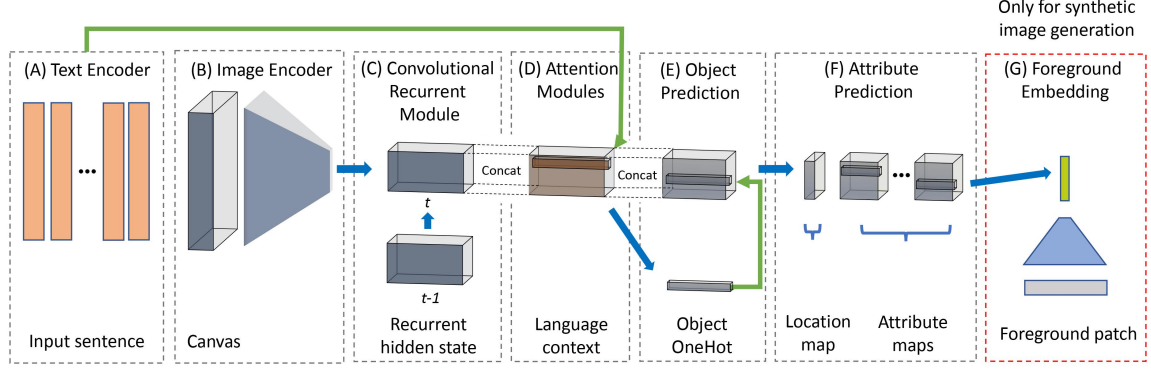


Figure 2.4: Overview of Text2Scene. Our general framework consists of (A) a Text Encoder that produces a sequential representation of the input, (B) an Image Encoder that encodes the current state of the generated scene, (C) a Convolutional Recurrent Module that tracks, for each spatial location, the history of what have been generated so far, (D-F) two attention-based predictors that sequentially focus on different parts of the input text, first to decide what object to place, then to decide what attributes to be assigned to the object, and (G) an optional foreground embedding step that learns an appearance vector for patch retrieval in the synthetic image generation task.

2.3 Text2Scene

Text2Scene adopts a Seq-to-Seq framework [120] and introduces key designs for spatial and sequential reasoning. Specifically, at each time step, the model modifies a background canvas in three steps: (1) the model attends to the input text to decide what is the next object to add, or decide whether the generation should end; (2) if the decision is to add a new object, the model *zooms in* the language context of the object to *decide* its attributes (e.g. pose, size) and relations with its surroundings (e.g. location, interactions with other objects); (3) the model refers back to the canvas and *grounds* (places) the extracted textual attributes into their corresponding visual representations.

To model this procedure, Text2Scene consists of a text encoder, which takes as input a sequence of M words w_i (section 2.3.1), an object decoder, which predicts sequentially T objects o_t , and an attribute decoder that predicts for each o_t their locations l_t and a set of k attributes $\{R_t^k\}$ (section 2.3.2). The scene generation starts

from an initially empty canvas B_0 that is updated at each time step. In the synthetic image generation task, we also jointly train a foreground patch embedding network (section 2.3.3) and treat the embedded vector as a target attribute. Figure 2.3 shows a step-by-step generation of an abstract scene.

2.3.1 Text Encoder

Our text encoder is a bidirectional recurrent network with Gated Recurrent Units (GRUs) [24]. For a given sentence, we compute for each word w_i :

$$h_i^E = \text{BiGRU}(x_i, h_{i-1}^E, h_{i+1}^E), \quad (2.1)$$

Here BiGRU is a bidirectional GRU cell, x_i is a word embedding vector corresponding to the i -th word w_i , and h_i^E is a hidden vector encoding the current word and its context. We use the pairs $[h_i^E; x_i]$, the concatenation of h_i^E and x_i , as the encoded text feature.

2.3.2 Object and Attribute Decoders

At each step t , our model predicts the next object o_t from an object vocabulary \mathcal{V} and its k attributes $\{R_t^k\}$, using text feature $\{[h_i^E; x_i]\}$ and the current canvas B_t as input. For this part, we use a convolutional network (CNN) Ω to encode B_t into a $\mathcal{C} \times H \times W$ feature map, representing the current scene state. We model the history of the scene states $\{h_t^D\}$ with a convolutional GRU (ConvGRU):

$$h_t^D = \text{ConvGRU}(\Omega(B_t), h_{t-1}^D), \quad (2.2)$$

The initial hidden state is created by spatially replicating the last hidden state of the text encoder. Here h_t^D provides an informative representation of the temporal

dynamics of each spatial (grid) location in the scene. Since this representation might fail to capture small objects, a one-hot vector of the object predicted at the previous step o_{t-1} is also provided as input to the downstream decoders. The initial object is set as a special start-of-scene token.

Attention-based Object Decoder: Our object decoder is an attention-based model that outputs the likelihood scores of all possible objects in an object vocabulary \mathcal{V} . It takes as input the recurrent scene state h_t^D , text features $\{[h_i^E; x_i]\}$ and the previously predicted object o_{t-1} :

$$u_t^o = \text{AvgPooling}(\Psi^o(h_t^D)), \quad (2.3)$$

$$c_t^o = \Phi^o([u_t^o; o_{t-1}], \{[h_i^E; x_i]\}), \quad (2.4)$$

$$p(o_t) \propto \Theta^o([u_t^o; o_{t-1}; c_t^o]), \quad (2.5)$$

here Ψ^o is a convolutional network with spatial attention on h_t^D , similar as in [141]. The goal of Ψ^o is to collect the spatial contexts necessary for the object prediction, e.g. what objects have already been added. The attended spatial features are then fused into a vector u_t^o by average pooling. Φ^o is the text-based attention module, similar as in [80], which uses $[u_t^o; o_{t-1}]$ to attend to the language context $\{[h_i^E; x_i]\}$ and collect the context vector c_t^o . Ideally, c_t^o encodes information about all the described objects that have not been added to the scene thus far. Θ^o is a two-layer perceptron predicting the likelihood of the next object $p(o_t)$ from the concatenation of u_t^o , o_{t-1} , and c_t^o , using a softmax function.

Attention-based Attribute Decoder The attribute set corresponding to the object o_t can be predicted similarly. We use another attention module Φ^a to “zoom in” the language context of o_t , extracting a new context vector c_t^a . Compared with c_t^o which may contain information of all the objects that have not been added yet, c_t^a focuses more specifically on contents related to the current object o_t . For each spatial

location in h_t^D , the model predicts a location likelihood l_t , and a set of attribute likelihoods $\{R_t^k\}$. Here, possible locations are discretized into the same spatial resolution of h_t^D . In summary, we have:

$$c_t^a = \Phi^a(o_t, \{[h_i^E; x_i]\}) \quad (2.6)$$

$$u_t^a = \Psi^a([h_t^D; c_t^a]) \quad (2.7)$$

$$p(l_t, \{R_t^k\}) = \Theta^a([u_t^a; o_t; c_t^a]), \quad (2.8)$$

Φ^a is a text-based attention module aligning o_t with the language context $\{[h_i^E; x_i]\}$. Ψ^a is an image-based attention module aiming to find an affordable location to add o_t . Here c_t^a is spatially replicated before concatenating with h_t^D . The final likelihood map $p(l_t, \{R_t^k\})$ is predicted by a convolutional network Θ^a , followed by softmax classifiers for l_t and discrete $\{R_t^k\}$. For continuous attributes $\{R_t^k\}$ such as the appearance vector Q_t for patch retrieval (next section), we normalize the output using an ℓ_2 -norm.

2.3.3 Foreground Patch Embedding

We predict a particular attribute: an appearance vector Q_t , only for the model trained to generate synthetic image composites (i.e. images composed of patches retrieved from other images). As with other attributes, Q_t is predicted for every location in the output feature map which is used at test time to retrieve similar patches from a pre-computed collection of object segments from other images. We train a patch embedding network using a CNN which reduces the foreground patch in the target image into a 1D vector F_t . The goal is to minimize the ℓ_2 -distance between Q_t and F_t using a triplet embedding loss [38] that encourages a small distance $\|Q_t, F_t\|_2$ but a larger distance with other patches $\|Q_t, F_k\|_2$. Here F_k is the feature of a "negative"

patch, which is randomly selected from the same category of F_t :

$$L_{triplet}(Q_t, F_t) = \max\{\|Q_t, F_t\|_2 - \|Q_t, F_k\|_2 + \alpha, 0\} \quad (2.9)$$

α is a margin hyper-parameter.

2.3.4 Objective

The loss function for a given example in our model with reference values $(o_t, l_t, \{R_t^k\}, F_t)$ is:

$$\begin{aligned} L = & -w_o \sum_t \log p(o_t) - w_l \sum_t \log p(l_t) \\ & - \sum_k w_k \sum_t \log p(R_t^k) + w_e \sum_t L_{triplet}(Q_t, F_t) \\ & + w_a^O L_{attn}^O + w_a^A L_{attn}^A, \end{aligned}$$

where the first three terms are negative log-likelihood losses corresponding to the object, location, and discrete attribute softmax classifiers. $L_{triplet}$ is a triplet embedding loss optionally used for the synthetic image generation task. L_{attn}^* are regularization terms inspired by the doubly stochastic attention module proposed in [141]. Here $L_{attn}^* = \sum_i [1 - \sum_t \alpha_{ti}^*]^2$ where $\{\alpha_{ti}^o\}$ and $\{\alpha_{ti}^a\}$ are the attention weights from Φ^o and Φ^a respectively. These regularization terms encourage the model to distribute the attention across all the words in the input sentence so that it will not miss any described objects. Finally, w_o , w_l , $\{w_k\}$, w_e , w_a^O , and w_a^A are hyperparameters controlling the relative contribution of each loss.

Details for different scene generation tasks In the Abstract Scenes generation task, B_t is represented directly as an RGB image. In the layout generation task, B_t is a 3D tensor with a shape of (V, H, W) , where each spatial location contains a one-hot

vector indicating the semantic label of the object at that location. Similarly, in the synthetic image generation task, B_t is a 3D tensor with a shape of $(3\mathcal{V}, H, W)$, where every three channels of this tensor encode the color patches of a specific category from the background canvas image. For the foreground embedding module, we adopt the canvas representation in [96] to encode the foreground patch for simplicity. As the composite images may exhibit gaps between patches, we also leverage the stitching network in [96] for post-processing. Note that the missing region may also be filled by any other inpainting approaches.

2.4 Experiments

We conduct experiments on three text-to-scene tasks: (I) constructing abstract scenes of clip-arts in the Abstract Scenes [154] dataset; (II) predicting semantic object layouts of real images in the COCO [74] dataset; and (III) generating synthetic image composites in the COCO [74] dataset.

Task (I): Clip-art Generation on Abstract Scenes. We use the dataset introduced by [154], which contains over 1,000 sets of 10 semantically similar scenes of children playing outside. The scenes are composed of 58 clip-art objects. The attributes we consider for each clip-art object are the location, size ($|R^{size}| = 3$), and the direction the object is facing ($|R^{direction}| = 2$). For the person objects, we also explicitly model the pose ($|R^{pose}| = 7$) and expression ($|R^{expression}| = 5$). There are three sentences describing different aspects of a scene. After filtering empty scenes, we obtain 9997 samples. Following [154], we reserve 1000 samples as the test set and 497 samples for validation.

Task (II): Semantic Layout Generation on COCO. The semantic layouts contain bounding boxes of the objects from 80 object categories defined in the COCO [74] dataset. We use the val2017 split as our test set and use 5000 samples from the

train2017 split for validation. We normalize the bounding boxes and order the objects from bottom to top as the y-coordinates typically indicate the distances between the objects and the camera. We further order the objects with the same y-coordinate based on their x-coordinates (from left to right) and categorical indices. The attributes we consider are location, size ($|R^{size}| = 17$), and aspect ratio ($|R^{aspect.ratio}| = 17$). For the size attribute, we discretize the normalized size range evenly into 17 scales. We also use 17 aspect ratio scales, which are $\{\frac{1}{i+1}\}_{i=1}^8$ and $\{i+1\}_{i=0}^8$.

Task (III): Synthetic Image Generation on COCO. We demonstrate our approach by generating synthetic image composites given input captions in COCO [74]. For fair comparisons with alternative approaches, we use the val2014 split as our test set and use 5000 samples from the train2014 split for validation. We collect segmented object and stuff patches from the training split. The stuff segments are extracted from the training images by taking connected components in corresponding semantic label maps from the COCO-Stuff annotations [50]. For object segments, we use all 80 categories defined in COCO. For stuff segments, we use the 15 super-categories defined in [50] as the class labels, which results in 95 categories in total. We order the patches as in the layout generation task but when composing the patches, we always render the object patches in front of the stuff patches. In our experiment, Q_t and F_t have a dimension of 128.

2.4.1 Network Architecture

Text Encoder

We use the same network architecture for the text encoders in all our experiments, which consists of a single layer bidirectional recurrent network with Gated Recurrent Units (GRUs). It takes a linear embedding of each word as input and has a hidden

dimension of 256 for each direction. We initialize the word embedding network with the pre-trained parameters from GloVe [93]. The word embedding vectors are kept fixed for abstract scene and semantic layout generations but finetuned for synthetic image generation.

Scene Encoder

The scene encoder Ω for abstract scene generation is an Imagenet (ILSVRC) pre-trained ResNet-34 [47]. Its parameters are fixed in all the experiments on Abstract Scene [154]. For layout and synthetic image generations, we develop our own scene encoders as the inputs for these tasks are not RGB images.

Table 2.1 and 2.2 show the architecture details. Here $|\mathcal{V}|$ is the size of the categorical vocabulary. In the layout generation task, $|\mathcal{V}|$ is 83, including 80 object categories in COCO [74] and three special categorical tokens: *sos*, *eos*, *pad*, representing the start and end points for sequence generation and the padding token. For synthetic image generation, $|\mathcal{V}|$ is 98, including 80 object categories in COCO [74], 15 super-categories for stuffs in COCO-stuff [50] and the special categorical tokens: *sos*, *eos*, *pad*.

The input for synthetic image generation has a layer-wise structure where every three channels contain the color patches of a specific category from the background canvas image. In this case, the categorical information of the color patches can be easily learned. On the other hand, since the input is a large but sparse volume with very few non-zero values, to reduce the number of parameters and memory usage, we use a depth-wise separable convolution as the first layer of Ω (index (2)), where each group of three channels (g3) is convolved to one single channel in the output feature map.

Index	Input	Operation	Output Shape
(1)	-	Input	$ \mathcal{V} \times 64 \times 64$
(2)	(1)	Conv(7×7 , $ \mathcal{V} \rightarrow 128$, s2)	$128 \times 32 \times 32$
(3)	(2)	Residual($128 \rightarrow 128$, s1)	$128 \times 32 \times 32$
(4)	(3)	Residual($128 \rightarrow 256$, s2)	$256 \times 16 \times 16$
(5)	(4)	Bilateral upsampling	$256 \times 28 \times 28$

Table 2.1: Architecture of our scene encoder Ω for layout generation. We follow the notation format used in [56]. Here $|\mathcal{V}|$ is the size of the categorical vocabulary. The input and output of each layer have a shape of $C \times H \times W$, where C is the number of channels and H and W are the height and width. The notation $Conv(K \times K, C_{in} \rightarrow C_{out})$ represents a convolutional layer with $K \times K$ kernels, C_{in} input channels and C_{out} output channels. The notation s2 means the convolutional layer has a stride of 2. The notation $Residual(C_{in} \rightarrow C_{out})$ is a residual module consisting of two 3×3 convolutions and a skip-connection layer. In the first residual block (index (3)), the skip-connection is an identity function and the first convolution has a stride of 1 (s1). In the second residual block (index (4)), the skip-connection is a 1×1 convolution with a stride of 2 (s2) and the first convolution also has a stride of 2 to downsample the feature map. Here all the convolutional layers are followed by a ReLU activation.

Convolutional Recurrent Module

The scene recurrent module for all our experiments is a convolutional GRU network [155] with one ConvGRU cell. Each convolutional layer in this module has a 3×3 kernel with a stride of 1 and a hidden dimension of 512. We pad the input of each convolution so that the output feature map has the same spatial resolution as the input. The hidden state is initialized by spatially replicating the last hidden state from the text encoder.

Object and Attribute Decoders

Table 2.3 shows the architectures for our object and attribute decoders. Ψ^o and Ψ^a are the spatial attention modules consisting of two convolutional layers. Θ^o is a two-layer perceptron predicting the likelihood of the next object using a softmax function. Θ^a is a four-layer CNN predicting the likelihoods of the location and attributes of the object. The output of Θ^a has $1 + \sum_k |R^k|$ channels, where $|R^k|$ denotes the

Index	Input	Operation	Output Shape
(1)	-	Input	$3 \mathcal{V} \times 128 \times 128$
(2)	(1)	Conv(7×7 , $3 \mathcal{V} \rightarrow \mathcal{V} $, s2, g3)	$ \mathcal{V} \times 64 \times 64$
(3)	(2)	Residual($ \mathcal{V} \rightarrow \mathcal{V} $, s1)	$ \mathcal{V} \times 64 \times 64$
(4)	(3)	Residual($ \mathcal{V} \rightarrow 2 \mathcal{V} $, s1)	$2 \mathcal{V} \times 64 \times 64$
(5)	(4)	Residual($2 \mathcal{V} \rightarrow 2 \mathcal{V} $, s1)	$2 \mathcal{V} \times 64 \times 64$
(6)	(5)	Residual($2 \mathcal{V} \rightarrow 3 \mathcal{V} $, s2)	$3 \mathcal{V} \times 32 \times 32$
(7)	(6)	Residual($3 \mathcal{V} \rightarrow 3 \mathcal{V} $, s1)	$3 \mathcal{V} \times 32 \times 32$
(8)	(7)	Residual($3 \mathcal{V} \rightarrow 4 \mathcal{V} $, s1)	$4 \mathcal{V} \times 32 \times 32$

Table 2.2: Architecture of our scene encoder Ω for synthetic image generation. The notations are in the same format of Table 2.1. The first convolution (index (2)) is a depthwise separable convolution where each group of three channels (g3) is convolved to one single channel in the output feature map. All the convolutional layers are followed by a LeakyReLU activation with a negative slope of 0.2.

discretized range of the k -th attribute, or the dimension of the appearance vector Q_t used as the query for patch retrieval for synthetic image generation. The first channel of the output from Θ^a predicts the location likelihoods which are normalized over the spatial domain using a softmax function. The rest channels predict the attributes for every grid location. During training, the likelihoods from the ground-truth locations are used to compute the loss. At each step of the test time, the top-1 location is first sampled from the model. The attributes are then collected from this sampled location. The text-based attention modules are defined similarly as in [80]. When denoting $d_i = [h_i^E; x_i]$, $s_t^o = [u_t^o; o_{t-1}]$, and $s_t^a = o_t$, Φ^o and Φ^a are defined as:

$$\begin{aligned}
c_t^* = \Phi^*(s_t^*, \{d_i\}) &= \sum_i \frac{\exp(\text{score}(s_t^*, d_i))}{\sum_j \exp(\text{score}(s_t^*, d_j))} \cdot d_i \\
\text{score}(s_t^*, d_k) &= (s_t^*)^\top W_\Phi^* d_k, \quad * \in o, a
\end{aligned}$$

Here, W_Φ^o and W_Φ^a are trainable matrices which learn to compute the attention scores for collecting the context vectors c_t^o and c_t^a .

These architecture designs are used for all the three generation tasks. The only difference is the grid resolution (H, W). For abstract scene and layout generations,

Module	Index	Input	Operation	Output Shape
Ψ^o	(1)	-	Conv(3×3, 512→256)	$256 \times H \times W$
	(2)	(1)	Conv(3×3, 256→1)	$1 \times H \times W$
Ψ^a	(1)	-	Conv(3×3, 1324→256)	$256 \times H \times W$
	(2)	(1)	Conv(3×3, 256→1)	$1 \times H \times W$
Θ^o	(1)	-	Linear((1324 + $ \mathcal{V} $)→512)	512
	(2)	(1)	Linear(512→ $ \mathcal{V} $)	$ \mathcal{V} $
Θ^a	(1)	-	Conv(3×3, (1324+ $ \mathcal{V} $)→512)	$512 \times H \times W$
	(2)	(1)	Conv(3×3, 512→256)	$256 \times H \times W$
	(3)	(2)	Conv(3×3, 256→256)	$256 \times H \times W$
	(4)	(3)	Conv(3×3, 256→ $(1 + \sum_k R^k)$)	$(1 + \sum_k R^k) \times H \times W$

Table 2.3: Architectures for the object and attribute decoders. The notation *Linear*($C_{in} \rightarrow C_{out}$) represents a fully connected layer with C_{in} input channels and C_{out} output channels. All layers, except the last layer of each module, are followed by a ReLU activation.

(H, W) = (28, 28). For synthetic image generation, (H, W) = (32, 32). Note that, although our model uses a fixed grid resolution, the composition can be performed on canvases of different sizes.

Foreground Patch Embedding

The foreground segment representation we use is similar with the one in [96], where each segment P is represented by a tuple $(P^{color}, P^{mask}, P^{context})$. Here $P^{color} \in \mathbb{R}^{3 \times H \times W}$ is a color patch containing the segment, $P^{mask} \in \{0, 1\}^{1 \times H \times W}$ is a binary mask indicating the foreground region of P^{color} , $P^{context} \in \{0, 1\}^{|\mathcal{V}| \times H \times W}$ is a semantic map representing the semantic context around P . The context region of P is obtained by computing the bounding box of the segment and enlarging it by 50% in each direction.

Table 2.4 shows the architecture of our foreground patch embedding network. Here, the concatenation of $(P^{color}, P^{mask}, P^{context})$ is fed into a five-layer convolutional network which reduces the input into a 1D feature vector F_s (index (7)). As this convolutional backbone is relatively shallow, F_s is expected to encode the shape, appearance, and context, but may not capture the fine-grained semantic attributes of

Index	Input	Operation	Output Shape
(1)	-	Input layout	$(\mathcal{V} + 4) \times 64 \times 64$
(2)	(1)	Conv(2×2 , $(\mathcal{V} + 4) \rightarrow 256$, s2)	$256 \times 32 \times 32$
(3)	(2)	Conv(2×2 , $256 \rightarrow 256$, s2)	$256 \times 16 \times 16$
(4)	(3)	Conv(2×2 , $256 \rightarrow 256$, s2)	$256 \times 8 \times 8$
(5)	(4)	Conv(2×2 , $256 \rightarrow 256$, s2)	$256 \times 4 \times 4$
(6)	(5)	Conv(2×2 , $256 \rightarrow 128$, s2)	$256 \times 2 \times 2$
(7)	(6)	Global average pooling	256
(8)	-	Input patch feature	2048
(9)	(7)(8)	Linear($(256 + 2048) \rightarrow 128$)	128

Table 2.4: Architecture of our foreground patch embedding network for synthetic image generation. All the convolutional layers are followed by a LeakyReLU activation with a negative slope of 0.2.

P . In our experiments, we find that incorporating the knowledge from the pre-trained deep features of P^{color} can help retrieve segments associated with strong semantics, such as the "person" segments. Therefore, we also use the pre-trained features F_d (index (8)) of P^{color} from the mean pooling layer of ResNet152 [47], which has 2048 features. The final vector F_t is predicted from the concatenation of (F_s, F_d) by a linear regression.

Inpainting Network

Our inpainting network has the same architecture as the image synthesis module proposed in [96], except that we exclude all the layer-normalization layers. To generate the simulated canvases on COCO, we follow the procedures proposed in [96], but make minor modifications: (1) we use the trained embedding patch features to retrieve alternative segments to stencil the canvas, instead of the intersection-over-union based criterion used in [96]. (2) we do not perform boundary elision for the segments as it may remove fine grained details of the segments such as human faces.

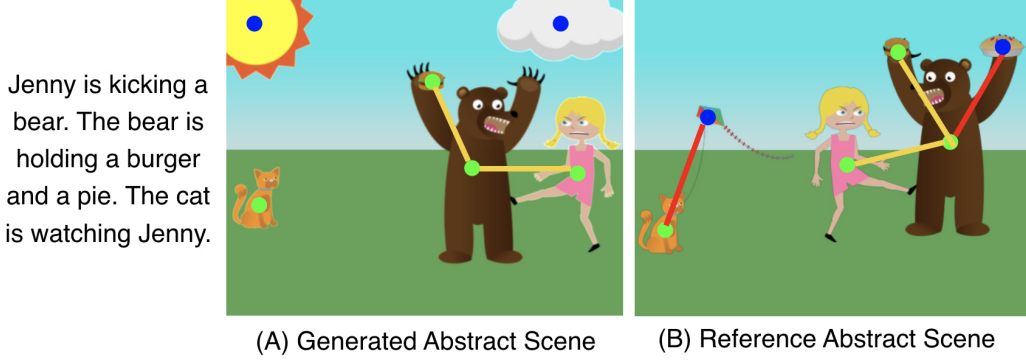


Figure 2.5: Evaluation metrics for the abstract scene generation task: the green dots show the common $U\text{-obj}$ between the reference (B) and the generated abstract scene (A), the blue dots show the missing and mispredicted objects. Similarly, the yellow lines show the common $B\text{-obj}$ and the red lines show the missing and mispredicted $B\text{-obj}$. The $U\text{-obj}$ precision/recall for this example is 0.667/0.667, the $B\text{-obj}$ precision/recall is 1.0/0.5.

2.4.2 Optimization

For optimization we use Adam [61] with an initial learning rate of $5e-5$. The learning rate is decayed by 0.8 every 3 epochs. We clip the gradients in the back-propagation such that the norm of the gradients is not larger than 10. Models are trained until validation errors stop decreasing. For abstract scene generation, we set the hyperparameters $(w_o, w_l, w_{pose}, w_{expression}, w_{size}, w_{direction}, w_a^O, w_a^A)$ to $(8, 2, 2, 2, 1, 1, 1, 1)$. For semantic layout generation, we set the hyperparameters $(w_o, w_l, w_{size}, w_{aratio}, w_a^O, w_a^A)$ to $(5, 2, 2, 2, 1, 0)$. For synthetic image generation, we set the hyperparameters $(w_o, w_l, w_{size}, w_{aratio}, w_a^O, w_a^A, w_e, \alpha)$ to $(5, 2, 2, 2, 1, 0, 10, 0.5)$. The hyperparameters are chosen to make the losses of different components comparable. Exploration of the best hyperparameters is left for future work.

2.4.3 Evaluation

Automatic Metrics. Our tasks pose new challenges in evaluating the models as (1) the three types of scene representations are quite different from each other; and

(2) there is no absolute one-to-one correspondence between a sentence and a scene. For the abstract scene generation task, we draw inspiration from the evaluation metrics applied in machine translation [11] but we aim at aligning multimodal visual-linguistic data instead. To this end, we propose to compute the following metrics: precision/recall on single objects (U-obj), “bigram” object pairs (B-obj); classification accuracies for poses, expressions; Euclidean distances (defined as a Gaussian function with a kernel size of 0.2) for normalized coordinates of U-obj and B-obj. A “bigram” object pair is defined as a pair of objects with overlapping bounding boxes as illustrated in Figure 2.5.





In the layout generation experiment, it is harder to define evaluation metrics given the complexity of real world object layouts. Inspired by [51], we employ caption generation as an extrinsic evaluation. We generate captions from the semantic layouts using [145] and compare them back to the original captions used to generate the scenes. We use commonly used metrics for captioning such as BLEU [90], METEOR [11], ROUGE-L [73], CIDEr [131] and SPICE [2].

For synthetic image generation, we adopt the Inception Score (IS) metric [108] which is commonly used in recent text to image generation methods. However, as IS does not evaluate the correspondence between images and captions, we also employ an extrinsic evaluation using image captioning using the Show-and-Tell caption generator [133] as in [51].

Baselines. For abstract scene generation, we run comparisons with [154]. We also consider variants of our full model: (1) Text2Scene (w/o attention): a model without any attention module. In particular, we replace Eq. 2.3 with a pure average pooling operation on h_t^D , discard c_t^o in Eq. 2.5, discard c_t^a and replace u_t^a with h_t^D in Eq. 2.8. (2) Text2Scene (w object attention): a model with attention modules for object prediction but no dedicated attention for attribute prediction. Specifically, we replace

Image Tagging Instructions (Click to collapse)

Determine if the sentences describe the clip art images.

Jenny wants the baseball.

☐ True ☐ False ☐ Unknown

Mike wears a blue cap.

☐ True ☐ False ☐ Unknown

Mike does not want to share his ball.

☐ True ☐ False ☐ Unknown

Jenny wants the baseball.

☐ True ☐ False ☐ Unknown

Mike wears a blue cap.

☐ True ☐ False ☐ Unknown

Mike does not want to share his ball.

☐ True ☐ False ☐ Unknown

Jenny wants the baseball.

☐ True ☐ False ☐ Unknown

Mike wears a blue cap.

☐ True ☐ False ☐ Unknown

Mike does not want to share his ball.

☐ True ☐ False ☐ Unknown

Jenny wants the baseball.

☐ True ☐ False ☐ Unknown

Mike wears a blue cap.

☐ True ☐ False ☐ Unknown

Mike does not want to share his ball.

☐ True ☐ False ☐ Unknown


Next

(A)

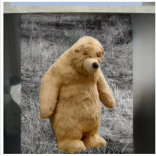
A/B test Instructions (Click to collapse)

Which image matches the caption better.

a teddy bear sitting in a very unusual spot high up



☐ Option A



☐ Option B

Next

(B)

Figure 2.6: Screen shots of the user interfaces for our human subject studies on Amazon Mechanical Turk. (A) User interface for the evaluation study of the abstract scene generation experiment; (B) User interface for the evaluation study of the synthetic image generation experiment.

(u_t^a, c_t^a) with (h_t^D, c_t^o) in Eq. 2.8. (3) Text2Scene (w both attentions): a model with dedicated attention modules for both object and attribute predictions but no regularization. For synthetic image generation, we compare our approach with a broad range of state-of-the-art methods, including [51, 56, 101, 143, 148, 150].

Human Evaluations. We also conduct human evaluations using crowdsourcing on 100 groups of clip-art scenes generated for the Abstract Scene dataset using random captions from the test split. Human annotators are asked to determine whether an input text is a true statement given the generated scene (entailment). Each scene in this dataset is associated with three sentences that are used as the statements. Each sentence-scene pair is reviewed by three annotators to determine if the entailment is true, false or uncertain. Ignoring uncertain responses, we use the ratio of

Methods	U-obj		B-obj		Pose	Expr	U-obj	B-obj
	Prec	Recall	Prec	Recall			Coord	Coord
Zitnick et al. [154]	0.722	0.655	0.280	0.265	0.407	0.370	0.449	0.416
Text2Scene (w/o attention)	0.665	0.605	0.228	0.186	0.305	0.323	0.395	0.338
Text2Scene (w object attention)	0.731	0.671	0.312	0.261	0.365	0.368	0.406	0.427
Text2Scene (w both attentions)	0.749	0.685	0.327	0.272	0.408	0.374	0.402	0.467
Text2Scene (full)	0.760	0.698	0.348	0.301	0.418	0.375	0.409	0.483

Table 2.5: Quantitative evaluation on the Abstract Scenes dataset. Our full model performs better in all metrics except U-obj Coord which measures exact object coordinates. It also shows that our sequential attention approach is effective.

Methods	Scores	≥ 1 ≥ 2		Obj-Single	Obj-Pair	Location	Expression
				sub-pred	sub-pred-obj	pred:loc	pred:expr
Reference	0.919	1.0	0.97	0.905	0.88	0.933	0.875
Zitnick et al. [154]	0.555	0.92	0.49	0.53	0.44	0.667	0.625
Text2Scene (w/o attention)	0.455	0.75	0.42	0.431	0.36	0.6	0.583
Text2Scene (full)	0.644	0.94	0.62	0.628	0.48	0.667	0.708

Table 2.6: Human evaluation on Abstract Scenes. Each scene is generated from three textual statements. Users are asked to rate if the generated scene validates each input statement. Our method generates scenes that abide by at least one of the statements 94% of the times, and by at least two statements 64%, and is superior in all types of statements except Location.

the sentence-scene pairs marked as `true` for evaluation. Figure 2.6 (A) shows the user interface of this study.

We also perform predicate-argument semantic frame analysis [18] on our results. Using the semantic parser from [154], we subdivide the sentences as: `sub-pred` corresponding to sentences referring to only one object, `sub-pred-obj` corresponding to sentences referring to object pairs with semantic relations, `pred:loc` corresponding to sentences referring to locations, and `pred:pa` corresponding to sentences mentioning facial expressions.

For synthetic image generation we use a similar human evaluation as in [96]. We compare our method against SG2IM [56], HDGAN [150] and AttnGAN [143]. We resize our generated images to the same resolutions as in these alternative methods, 64×64 for SG2IM [56], 256×256 for HDGAN [150] and AttnGAN [143]. For each

Input	Zitnick et al. 2013	Text2Scene (w/o Attention)	Text2Scene	Reference
Jenny is wearing sunglasses. Mike is holding the red shovel. Mike is wearing a viking head.				
Mike went down the slide fast. Jenny is worried that Mike is hurt. Jenny is wearing a chef hat.				
Mike is angry at Jenny. Jenny is sad that Mike took the frisbee. The pizza is on the table.				
Jenny is holding a bucket and shovel. Mike fell off the swingset. There is rain and lightning in the sky				

Figure 2.7: Examples of generated abstract scenes. The first column shows the input text, and the last column shows the reference *true* scene from the dataset.

sentence randomly selected from the test set, we present images generated by our method and a competing method and allow the user to choose the one which better represents the text. We collect results for 500 sentences. For each sentence, we collect responses from 5 different annotators. Figure 2.6 (B) shows the user interface of this study.

2.4.4 Results and Discussion

Abstract Scenes and Semantic Layouts: Table 2.5 shows quantitative results on Abstract Scenes. Text2Scene improves over [154] and our variants on all metrics except U-obj Coord. Human evaluation results on Table 2.6 confirm the quality of our outputs, where Scores are the percentage of sentence-scene pairs with a true entailment; (≥ 1) (≥ 2) indicate if our method produces scenes that entailed

Methods	B1	B2	B3	B4	METEOR	ROUGE	CIDEr	SPICE
Captioning from True Layout [145]	0.678	0.492	0.348	0.248	0.227	0.495	0.838	0.160
Text2Scene (w/o attention)	0.591	0.391	0.254	0.169	0.179	0.430	0.531	0.110
Text2Scene (w object attention)	0.591	0.391	0.256	0.171	0.179	0.430	0.524	0.110
Text2Scene (w both attentions)	0.600	0.401	0.263	0.175	0.182	0.436	0.555	0.114
Text2Scene (full)	0.615	0.415	0.275	0.185	0.189	0.446	0.601	0.123

Table 2.7: Quantitative evaluation on the layout generation task. Our full model generates more accurate captions from the generated layouts than the baselines. We also include caption generation results from ground truth layouts as an upper bound on this task.

Methods	IS	B1	B2	B3	B4	METEOR	ROUGE	CIDEr	SPICE
Real image	36.00±0.7	0.730	0.563	0.428	0.327	0.262	0.545	1.012	0.188
GAN-INT-CLS [100]	7.88±0.07	0.470	0.253	0.136	0.077	0.122	–	0.160	–
SG2IM* [56]	6.7±0.1	0.504	0.294	0.178	0.116	0.141	0.373	0.289	0.070
StackGAN [148]	10.62±0.19	0.486	0.278	0.166	0.106	0.130	0.360	0.216	0.057
HDGAN [150]	11.86±0.18	0.489	0.284	0.173	0.112	0.132	0.363	0.225	0.060
Hong et al [51]	11.46±0.09	0.541	0.332	0.199	0.122	0.154	–	0.367	–
AttnGan [143]	25.89±0.47	0.640	0.455	0.324	0.235	0.213	0.474	0.693	0.141
Text2Scene (w/o inpaint.)	22.33±1.58	0.602	0.412	0.288	0.207	0.196	0.448	0.624	0.126
Text2Scene (w inpaint.)	24.77±1.59	0.614	0.426	0.300	0.218	0.201	0.457	0.656	0.130

Table 2.8: Quantitative evaluation on the synthetic image generation task. Our model is superior on automated metrics to all competing approaches except AttnGan, even without post-processing. *The result of SG2IM is evaluated on the validation set defined in [56], which is a subset of the COCO val2014 split.

at least one (or two) out of three statements. Text2Scene also shows better results on statements with specific semantic information such as `Obj-single`, `Obj-pair`, and `Expression`, and is comparable on `Location` statements. As a sanity check, we also test reference *true* scenes provided in the Abstract Scenes dataset (first row). Results show that it is more challenging to generate the semantically related object pairs. Overall, the results also suggest that our proposed metrics correlate with human judgment on the task.

Figure 2.7 shows qualitative examples of our models on Abstract Scenes in comparison with baseline approaches and the reference scenes. These examples illustrate that Text2Scene is able to capture semantic nuances such as the spatial relation between two objects (e.g., the bucket and the shovel are correctly placed in Jenny’s hands in the last row) and object locations (e.g., Mike is on the ground near the

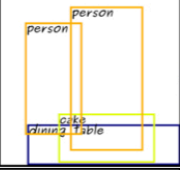
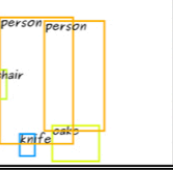

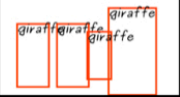


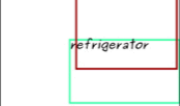


Input Caption	Predicted Layout	Reference Layout	Reference Image	Generated Caption
A happy couple is cutting a decorated cake .				A woman and a woman are cutting a cake
Four giraffes are reaching in the tree for food.				A couple of giraffes are standing in a field.
A gray cat standing on the top of a refrigerator .				A cat is sitting in a room.

Figure 2.8: Generated layouts from input captions and generated captions from the predicted layouts (best viewed in color). Our model successfully predicts the presence (purple text) and number of objects (blue text), and their spatial relations (red text).

	Ratio
Text2Scene > SG2IM [56]	0.7672
Text2Scene > HDGAN [150]	0.8692
Text2Scene > AttnGAN [143]	0.7588

Table 2.9: Two-alternative forced-choiced evaluation on the synthetic image generation task against competing methods.

swing set in the last row).

Table 2.7 shows an extrinsic evaluation on the layout generation task. We perform this evaluation by generating captions from our predicted layouts. Results show our full method generates the captions that are closest to the captions generated from true layouts. Qualitative results in Figure 2.8 also show that our model learns important visual concepts such as the presence and number of object instances, and their spatial relations.

Synthetic Image Composites: Table 2.8 shows evaluation of synthetic scenes using automatic metrics. Text2Scene without any post-processing already outper-



Figure 2.9: Qualitative examples of synthetic image generation (best viewed in color). The first column shows input captions with manually highlighted objects (purple), counts (blue) and relations (red). The second columns shows the *true* images. Columns in the middle show competing approaches. The last two columns show the outputs of our model before and after pre-processing.

forms all previous methods by large margins except AttnGAN [143]. As our model adopts a composite image generation framework without adversarial training, gaps between adjacent patches may result in unnaturally shaded areas. We observe that, after performing a regression-based inpainting [96], the composite outputs achieve consistent improvements on all automatic metrics. We posit that our model can be further improved by incorporating more robust post-processing or in combination with GAN-based methods. On the other hand, human evaluations show that our method significantly outperforms alternative approaches including AttnGAN [143], demonstrating the potential of leveraging realistic image patches for text-to-image generation. It is important to note that SG2IM [56] and Hong et al [51] also use segment and bounding box supervision – as does our method, and AttnGan [143] uses



Figure 2.10: Example synthetic images and the source images from which the patch segments are retrieved for compositing. For each synthetic image, we show three source images for clarity.

an Imagenet (ILSVRC) pretrained Inceptionv3 network. In addition, as our model contains a patch retrieval module, it is important that the model does not generate a synthetic image by simply retrieving patches from a single training image. On average, each composite image generated from our model contains 8.15 patches from 7.38 different source images, demonstrating that the model does not simply learn a global image retrieval. Fig. 2.9 shows qualitative examples of the synthetic image composites. We also present in Fig. 2.10 the generated images and the corresponding source images from which the patch segments are retrieved for compositing. For each generated image, we show three source images for clarity. The examples illustrate that our model learns not only the presence and spatial layout of objects, but also the semantic knowledge that helps retrieve segments in similar contexts. Since our

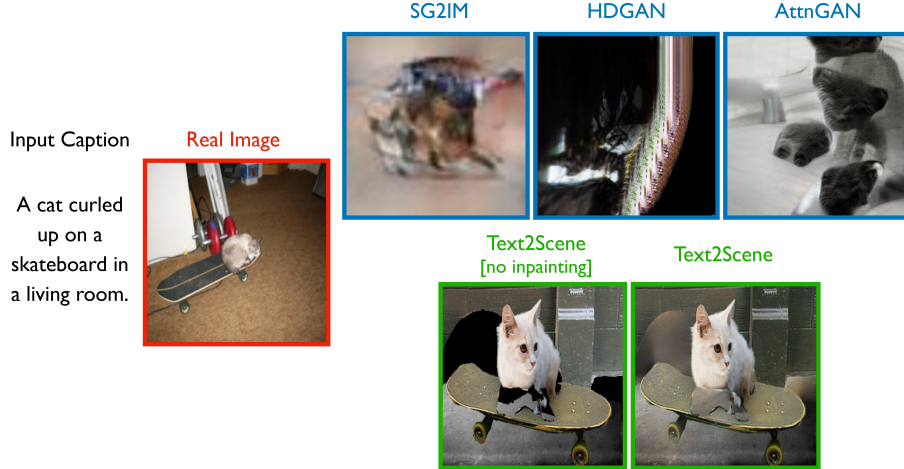


Figure 2.11: Comparing Text2Scene with state-of-the-art approaches on an uncommon example. While SG2IM [56], HDGAN [150], AttnGAN [143], all struggle with this example, Text2Scene successfully learns the presence of the objects “cat” and “skateboard” as well as their relation.

model learns about objects and relations separately, we also observed that it is often able to generalize to uncommon situations. Fig. 2.11 provides the result of generating a composite image from the caption “A cat curled up on a skateboard in a living room” using Text2Scene. This example is also discussed in Section 2.1. It is shown that the previous state-of-the-art approaches, e.g. SG2IM [56], HDGAN [150], AttnGAN [143], struggle with this example, while Text2Scene successfully learns the presence of the objects “cat” and “skateboard” as well as their relation.

2.5 Summary

In this chapter, we present a novel sequence-to-sequence model for generating compositional scene representations from visually descriptive language. We provide extensive quantitative analysis of our model for different scene generation tasks on datasets from different domains: Abstract Scenes [154] and COCO [74]. Experimental results demonstrate the capacity of our model to capture finer semantic concepts from visually descriptive text and generate complex scenes.

Chapter 3

Learning Compositional Text Representations for Interactive Image Retrieval

3.1 Introduction

In the previous chapter, we explore learning compositional representations of images. In this chapter, we focus more on learning compositional representations of text. In particular, we aim to integrate multiple rounds of text queries into an efficient and effective representation for interactive image retrieval.

Retrieving images from text-based queries has been an active area of research that requires some level of visual and textual understanding. Significant improvement has been achieved over the past years with advances in representation learning but finding very specific images with detailed specifications remains challenging. A common way of the specification is through natural language queries, where a user inputs a description of the image and obtains a set of results. We focus on a common scenario where a user is trying to find an exact image, or similarly where the user has a very



Figure 3.1: An example of the interactive image retrieval with our Drill-down model, where a user generated query (U_t) progressively refines the search results (S_t) until the target image is among top search results.

specific idea of a target image, or is deciding on-the-fly while querying. We present empirical evidence that users are much more successful if they are allowed to refine their search results with subsequent textual queries. Users might start with a general query about the “concept” of the image they have in mind and then “drill down” onto more specific descriptions of objects or attributes in the image to refine the results.

Among previous efforts in image retrieval, a promising paradigm is to learn a visual-semantic embedding by minimizing the distance between a target image and an input textual query using a joint feature space. Pioneering approaches such as [35, 62, 69, 129, 134, 140] have demonstrated remarkable performance on large scale datasets such as Flickr30K [95] and COCO [74], and domain-specific tasks such as outfit composition [46]. However, we find that these methods are limited in their capacity for retrieving highly specific images, because it is either difficult for users to be specific enough with a single query or users may not have the full picture in mind beforehand. We show an example of this type of interaction in Fig. 3.1. While single-query retrieval might be more suited for domains such as product search where images typically contain only one object, requiring users to describe a whole scene in one sentence might be too demanding. More recently, dialog based search has been proposed to overcome some of the limitations of single-query retrieval [29, 42, 72, 117].

In this work, we propose Drill-down [121], an interactive image search framework

for retrieving complex scenes, which learns to capture the fine-grained alignments between images and multiple text queries. Our work is inspired by the observations that: (1) user queries at each turn may not exhaustively describe all the details of the target image, but focus on some local regions, which provide a natural decomposition of the whole scene. Therefore, we explicitly represent images as a list of object/stuff level features extracted from a pre-trained object detector [102]. This is also in line with recent research [69, 140] on learning region-phrase alignments for single-query methods; (2) complex scenes contain multiple objects that might share the same feature subspace. Particularly, existing state representations of sequential text queries, such as the hidden states of a RNN, condense all image properties in a single state vector, which makes it difficult to distinguish entities sharing the same feature subspace, such as multiple person instances. To address this, we propose to maintain a set of state vectors, encouraging each of the vectors to encode text queries corresponding to a distinct image region. Figure 3.2 shows an overview of our approach, images are represented with local feature representations, and the query state is represented by a fixed set of vectors that are selectively updated with each subsequent query.

We demonstrate the effectiveness of our approach on the Visual Genome dataset [65] in two scenarios: automatic image retrieval using region captions as queries, and interactive image retrieval with real queries from human evaluators. In both cases, our experimental results show that the proposed model outperforms existing methods, such as a hierarchical recurrent encoder model [110], while using less computational budget.

Our main contributions can be summarized as follows:

- We propose Drill-down, an interactive image search approach with multiple round queries which leverages region captions as a form of weak supervision

during training.

- We conduct experiments on a large-scale natural image dataset: Visual Genome [65], and demonstrate superior performance of our model on both simulated and real user queries;
- We show that our model, while producing a compact representation, outperforms competing baseline methods by a significant margin.

3.2 Related Work

Text-based image retrieval has been an active research topic for decades [20, 21, 106]. Prominent more contemporary works have recognized the need for richer user interactions in order to obtain higher quality results [5, 63, 64, 111]. Siddiquie et al [111] proposed an approach to use multiple query attributes. Kovashka et al [63, 64] further proposed using user feedback based on individual visual attributes to progressively improve search results. Arandjelovic et al [5] proposed a multiple query retrieval system that was used for querying specific objects within a large set of images. These works show that multiple independent queries generally outperform methods that jointly model the input set with a single query. Our work builds on these previous ideas but does not use an explicit notion of attributes and aims to support more general input text queries.

Remarkable results have been achieved by recent methods based on deep learning [35, 62, 134]. These methods typically explore mapping a text query and the target image into a common feature space. Learned feature representations are designated to capture both visual and semantic information in the same embedding space. In contrast, besides supporting multiple rounds of queries, our approach also has a richer region representation to explicitly map individual entities in images to textual

phrases. Another line of recent inquiry are dialog based image search systems [42, 72]. Liao et al [72] proposed to aggregate multi-round user responses from trained agents or human agents in order to iteratively refine a retrieved set of images using a hierarchical recurrent encoder-decoder framework [110]. We follow a similar protocol, but we explore a more open-ended domain of images corresponding to scenes depicting multiple objects. The method Guo et al [42] as in our work, used multiple rounds of natural language queries, and proposed collecting relative image captions as supervision for a product search task. In contrast, we pursue a weakly supervised approach where we leverage an image dataset with region captions that are used to simulate queries during training, thus bypassing the need to collect extra annotations. We demonstrate that training with simulated queries is surprisingly effective under human evaluations. As the hierarchical recurrent framework [110] was used in most of the previous dialog based methods [28, 29, 42, 72, 117], we provide a re-implementation of the hierarchical encoder (HRE) model with the queries as context and use it as one of our baselines. Different from the previous dialog based methods where the systems also provide textual responses, we explore a scenario where the system only responses with retrieved images, so no decoder module is required in our case.

Also relevant to our research are the existing works on learning image-word [45, 57, 69] or region-phrase [87] alignments for vision-language tasks. For instance, Karpathy et al [57] proposed to learn a bidirectional image-sentence mapping by jointly embedding fragments of images (objects) and sentences. The image fragments are extracted using a pre-trained object detector, while the sentence fragments are obtained using a dependency tree relation parser. Niu et al [87] extended this work by jointly learning hierarchical relations between phrases and image regions in an iterative refinement framework. Recently, Lee et al [69] developed a stacked cross attention

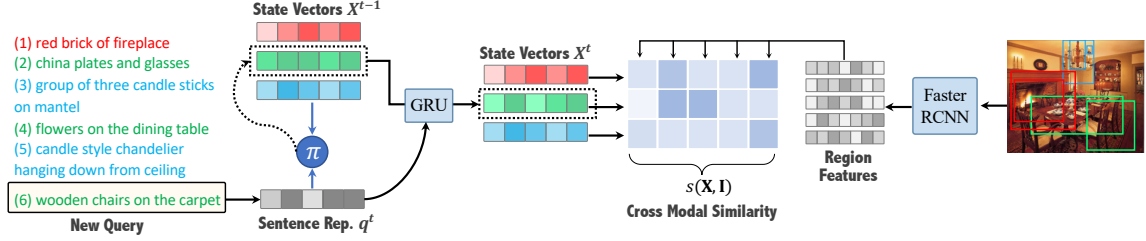


Figure 3.2: Overview of our model. Drill-down maintains a fixed set of state vectors \mathbf{X} , modeling the historical context of the user queries. Given a new query \mathbf{q}^t , our model selects and updates one of the state vectors. The updated state vectors \mathbf{X}^t and image region features are then projected to a cross-modal embedding space to measure the fine-grained alignment between each region-state pair.

network for word-region matching. Compared to these models, our proposed query state encoding aims at integrating multiple round queries while still using a compact representation of fixed size (i.e. independent of the number of queries), so that retrieval times do not depend on the number or the length of the queries. We show our compact representation to be both efficient and effective for interactive image search.

More closely related to our work are Memory Networks [59, 118, 137], which perform query and possibly update operations on a predefined memory space. In contrast to this line of research, we explore a more challenging scenario where the model needs to create and update the memory (i.e. the state vectors) on-the-fly so as to maintain the states of the queries.

3.3 Method

Retrieving images with multi-round refinements offers the potential benefit of reducing the ambiguity of each query but also raises challenges on how to integrate user queries from multiple rounds. Our model is inspired by the observation that users naturally underspecify in their queries by referring to local regions of the target image. We aim to capture these region level alignments by learning to map text queries

$\{\mathbf{s}_t\}_{t=1}^T$ and the target image \mathbf{I} into two sets of latent vectors $\{\mathbf{x}_i\}_{i=1}^M$ and $\{\mathbf{v}_j\}_{j=1}^N$ respectively, and computing the matching score of $\{\mathbf{s}_t\}_{t=1}^T$ and \mathbf{I} by measuring and aggregating fine-grained similarities between $\{\mathbf{x}_i\}_{i=1}^M$ and $\{\mathbf{v}_j\}_{j=1}^N$. Figure 3.2 provides an overview of our model.

3.3.1 Image Representation

To identify candidate regions referred in the queries, we follow [3, 69]. For each image \mathbf{I} , we first detect the potential objects and salient stuff using the FasterRCNN detector [102]. Corresponding features $\{\mathbf{c}_j\}$ are extracted from the ROI pooling layer of the detector. In practice, we leverage the object detector provided by [3], which is pre-trained on Visual Genome [65] with 1600 predefined object and stuff classes. A linear projection $\mathbf{v}_j = W_I \mathbf{c}_j + b_I$ is applied to reduce $\{\mathbf{c}_j\}$ into D-dimensional latent vectors $\mathbf{V} = \{\mathbf{v}_j\}_{j=1}^N$, $\mathbf{v}_j \in \mathbb{R}^D$. Here N is the number of regions in each image. The learnable parameters for the image representation $\{W_I, b_I\}$ are denoted as θ_I .

3.3.2 Query Representation

Supporting multi-round retrieval requires a state representation for integrating the queries from multiple turns. Solutions adopted by existing methods include applying a single recurrent network to the concatenation of all queries [35] or a hierarchical recurrent network [29, 42, 72, 117] modeling individual query and historical context in separate recurrent modules. These approaches produce a single latent vector that aggregates all queries. While state-of-the-art models [42, 72] show remarkable performance on domains such as fashion product search, we demonstrate that currently used single-vector representations are not the most effective for capturing complex scenes with multiple objects. Specifically, as image features used in existing methods are typically extracted from the penultimate layer of a pre-trained image classification

or object detection model, input instances of the same or very similar categories activate the same feature units in the extracted feature space. Therefore, it is nontrivial for these latent representations to encode and distinguish multiple entities from the same or very similar categories (i.e. multiple person instances).

We propose to maintain a set of latent representations $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^M, \mathbf{x}_i \in \mathbb{R}^D$ for multiple turn queries. Here M is the number of latent vectors. This parameter represents the computational budget, since retrieval time will depend on the compactness of this representation. While users might provide a general image description in the first round of querying, subsequent queries typically describe more specific regions. We aim at finding a good alignment between queries and image region representations $\{\mathbf{v}_j\}_{j=1}^N$. An ideal set of $\{\mathbf{x}_i\}_{i=1}^M$ should learn to group and encode the input queries into visually discriminative representations referring to distinct image regions. In the remaining section, we first introduce the cross modal similarity formula used in our model. We then explain how to update the state representations $\{\mathbf{x}_i\}_{i=1}^M$ from the queries $\{\mathbf{s}_t\}_{t=1}^T$ so as to optimize their matching score with the target image.

3.3.3 Cross Modal Similarity

To measure the similarity of $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^M$ and $\mathbf{V} = \{\mathbf{v}_j\}_{j=1}^N$, we first compute the cosine similarity of each possible state-region pair $(\mathbf{x}_i, \mathbf{v}_j)$: $s(\mathbf{x}_i, \mathbf{v}_j) = \mathbf{x}_i^T \mathbf{v}_j / \|\mathbf{x}_i\| \|\mathbf{v}_j\|$, where $\|\cdot\|$ denotes the $L2$ norm. Given $s(\mathbf{x}_i, \mathbf{v}_j)$, we define the similarity $s(\mathbf{x}_i, \mathbf{I})$ between a state vector \mathbf{x}_i and the target image \mathbf{I} as

$$s(\mathbf{x}_i, \mathbf{I}) = \frac{1}{N} \sum_{k=1}^N \alpha_{ik} s(\mathbf{x}_i, \mathbf{v}_k), \quad \alpha_{ik} = \frac{\exp(s(\mathbf{x}_i, \mathbf{v}_k)/\sigma)}{\sum_j \exp(s(\mathbf{x}_i, \mathbf{v}_j)/\sigma)} \quad (3.1)$$

Here σ is a temperature hyper-parameter. Note that this formulation is similar to measuring the cosine similarity of \mathbf{x}_i and a context vector $\sum_{k=1}^N \alpha_{ik} \mathbf{v}_k$ from an

attention module [69, 79]. The cross modal similarity between the state vectors $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^M$ and the target image \mathbf{I} is defined as $s(\mathbf{X}, \mathbf{I}) = \frac{1}{M} \sum_{k=1}^M s(\mathbf{x}_k, \mathbf{I})$.

3.3.4 Query Encoding

Given a query input \mathbf{s}_t at time t , our model maps each word token \mathbf{w}_k in \mathbf{s}_t to an E -dimensional vector via a linear projection: $\mathbf{e}_k = W_E \mathbf{w}_k$, $\mathbf{e}_k \in \mathbb{R}^E$, $k = 1, \dots, K$, then generates the sentence embedding via a uni-directional recurrent network ϕ with gated recurrent units (GRU) as: $\mathbf{h}_k = \phi(\mathbf{e}_k, \mathbf{h}_{k-1})$, $\mathbf{h}_k \in \mathbb{R}^D$. The first hidden state of ϕ is initialized as a zero vector, while the last hidden state is treated as the sentence representation: $\mathbf{q}^t = \mathbf{h}_K$. We also explore using a bidirectional encoder but find no improvement. Given the assumption that each text query describes a sub-region of the image, each \mathbf{q}^t only updates a subset of the state vectors. In this work, we focus on a simplified scenario where each \mathbf{q}^t only updates a single state vector $\mathbf{x}_k^{t-1} \in \mathbf{X}^{t-1}$. In detail, given the text query \mathbf{q}^t at time step t , our model samples \mathbf{x}_k^{t-1} from the previous state vector set $\mathbf{X}^{t-1} = \{\mathbf{x}_i^{t-1}\}_{i=1}^M$ based on the probability:

$$\pi(\mathbf{x}_k^{t-1} | \mathbf{X}^{t-1}, \mathbf{q}^t) = \begin{cases} \frac{\mathbb{1}(\mathbf{x}_k^{t-1} = \emptyset)}{\sum_j \mathbb{1}(\mathbf{x}_j^{t-1} = \emptyset)} & \text{if } \mathbf{X}^{t-1} \text{ has an empty vector} \\ \frac{\exp(f(\mathbf{x}_k^{t-1}, \mathbf{q}^t))}{\sum_j \exp(f(\mathbf{x}_j^{t-1}, \mathbf{q}^t))} & \text{otherwise} \end{cases} \quad (3.2)$$

$$f(\mathbf{x}_k^{t-1}, \mathbf{q}^t) = W_\pi^3(\delta(W_\pi^2(\delta(W_\pi^1[\mathbf{x}_k^{t-1}; \mathbf{q}^t] + b_\pi^1)) + b_\pi^2)) + b_\pi^3, \quad (3.3)$$

where $\mathbb{1}(\mathbf{x}_j^{t-1} = \emptyset)$ is an indicator function which returns 1 if \mathbf{x}_j^{t-1} is an empty vector and 0 otherwise. $f(\cdot)$ is a multilayer perceptron mapping the concatenation of \mathbf{x}_k^{t-1} and \mathbf{q}^t into a scalar value. Here δ is the ReLU activation function, $W_\pi^1 \in \mathbb{R}^{D \times 2D}$,

$W_\pi^2, \in \mathbb{R}^{D \times D}$, $W_\pi^3 \in \mathbb{R}^{1 \times D}$, $b_\pi^1, b_\pi^2 \in \mathbb{R}^D$, $b_\pi^3 \in \mathbb{R}$ are model parameters. An empty state vector is initialized with zero values. Ideally, an expressive sample policy should learn to allocate a new state vector when necessary. However, we empirically find it beneficial to update \mathbf{q}^t to an empty state vector whenever possible. Once \mathbf{x}_k^{t-1} is sampled, we update this state vector using a single uni-directional gated recurrent unit cell (GRU Cell) τ : $\mathbf{x}_k^t = \tau(\mathbf{q}^t, \mathbf{x}_k^{t-1})$. Note that our formulation is similar to a hard attention module [142]. Leveraging soft attention is possible, but it is more computationally expensive as it would need to update all state vectors. Our state vector update mechanism is inspired by the knowledge base methods with external memory [72]. Our method can be interpreted as building a knowledge base memory online from scratch, only from the query context, which can be trained end-to-end with other modules. We denote the learnable parameters for the state vector update policy function $\pi(\cdot)$ as $\theta_\pi = \{W_\pi^1, W_\pi^2, W_\pi^3, b_\pi^1, b_\pi^2, b_\pi^3\}$, and for the rest modules as $\theta_q = \{W_E, \phi, \tau\}$.

3.3.5 End-to-end Training

Our model is trained to optimize θ_I , θ_π and θ_q so as to achieve high similarity score between the queries $\{\mathbf{s}_t\}_{t=1}^T$ and the target image \mathbf{I} . Thus, we follow [35, 69] and adopt a triplet loss on $s(\mathbf{X}, \mathbf{I})$ with hard negatives:

$$L_e = \operatorname{argmin}_{\theta_I, \theta_q} \sum_{\mathbf{X}, \mathbf{I}} \ell(\mathbf{X}, \mathbf{I}) \quad (3.4)$$

$$\ell(\mathbf{X}, \mathbf{I}) = \max_{\mathbf{I}'} [\alpha + s(\mathbf{X}, \mathbf{I}') - s(\mathbf{X}, \mathbf{I})]_+ + \max_{\mathbf{X}'} [\alpha + s(\mathbf{X}', \mathbf{I}) - s(\mathbf{X}, \mathbf{I})]_+$$

Here, α is a margin parameter, $[\cdot]_+ \equiv \max(\cdot, 0)$. \mathbf{I}' and \mathbf{X}' are decoy images and state vectors within the same mini-batch as the ground-truth pair (\mathbf{X}, \mathbf{I}) during training. Note that L_e will only optimize the parameters θ_I and θ_q . Directly optimizing θ_π is

difficult as sampling from Equation 3.2 is non-differentiable. We propose to train the policy parameters via Reinforcement Learning (RL). Formally, the state in our RL formulation is the set of state vectors $\mathbf{X}^t = \{\mathbf{x}_i^t\}_{i=1}^M$, and the action $k \in \{1, \dots, M\}$ is to select the state vector \mathbf{x}_k^t from \mathbf{X}^t when fusing information from the embedded query vector \mathbf{q}^{t+1} . The RL objective is to maximize the expected cumulative discounted rewards, so in our case we define the reward function as the similarity between the state vectors \mathbf{X}^t and the image \mathbf{I} , i.e. $s(\mathbf{X}^t, \mathbf{I})$. Note that our reward function evaluates the potential similarity at all future time steps instead of only the last step T , encouraging the model to find the target image with fewer turns.

Supervised pre-training As optimizing the sampling policy requires reward signals from the retrieval environment, we pre-train the model by optimizing L_e with a fixed policy: $\pi(\mathbf{x}_k^{t-1} | \mathbf{X}^{t-1}, \mathbf{q}^t) = \mathbb{1}(k \equiv t \pmod{M})$, where $\mathbb{1}(\cdot)$ is an indicator function and M is the number of state vectors. Intuitively, this policy circularly updates the state vectors in order.

Joint optimization Given the pre-trained environment, we then jointly optimize the sampling policy and the other modules (i.e. θ_I, θ_q and θ_π). Because the next state \mathbf{X}^{t+1} is a deterministic function given the current state \mathbf{X}^t and action k , we adopt the policy improvement strategy from [42] to update the policy. Specifically, we estimate the state-action value $Q(\mathbf{X}^t, k) = \sum_{t'=t}^{T-1} \gamma^{t'-t} s(\mathbf{X}^{t'+1}, \mathbf{I})$ for each state vector selection action k by sampling one look-ahead trajectory. γ is the discount factor. The policy is then optimized to predict the most rewarding action $k^* = \operatorname{argmax}_k Q(\mathbf{X}^t, k)$ via a cross entropy loss:

$$L_\pi = \operatorname{argmin}_{\theta_\pi} \sum_{\mathbf{X}^t, \mathbf{q}^{t+1}} -\log(\pi(\mathbf{x}_{k^*}^t | \mathbf{X}^t, \mathbf{q}^{t+1}; \theta_\pi)) \quad (3.5)$$

We also jointly finetune θ_I and θ_q by applying L_e on the rollout state vectors \mathbf{X}_* : $L_e^* = \operatorname{argmin}_{\theta_I, \theta_q} \sum_{\mathbf{X}_*, \mathbf{I}} \ell(\mathbf{X}_*, \mathbf{I})$. The model is trained with the multi-task loss: $L = L_e^* + \mu L_\pi$, where μ is a scalar factor determining the trade-off between the two terms.

3.4 Experiments

In this section, we first introduce the dataset for evaluation. Then, we describe the competing approaches we considered and more details about our method. We also present the experiments on simulated queries (section 3.4.1) and interaction with human evaluators (section 3.4.2).

Dataset. We evaluate the performance of our method on the Visual Genome dataset [65]. Each image in Visual Genome is annotated with multiple region captions. We preprocess the data by removing duplicate region captions (e.g. multiple captions that are exactly the same), and images with less than 10 region captions. This preprocessing results in 105,414 image samples, which are further split into 92,105/5,000/9,896 for training/validation/testing. We also ensure that the images in the test split are not used for the training of the object detector [3]. All the evaluations, including the human subject study, are performed on the test split, which contains 9,896 images. We use region captions as queries to train our model, thus bypassing the challenging issue of data collection for this task. The vocabulary of the queries is built with the words that appear more than 10 times in all region captions, resulting in a vocabulary size of 14,284. During training, queries and their orders are randomly sampled. During validation and testing, the queries and their orders are kept fixed.

Baselines. We compare our method with four baseline models: (1) **HRE**: a hierarchical recurrent encoder network, which is commonly adopted by recent dialog based approaches [42, 72, 117]. We consider the framework using text queries as context, which consists of a sentence encoder, a context encoder and an image encoder. The sentence encoder has the same word embedding (e.g. the linear projection W_E) and sentence embedding (e.g. the ϕ function) as the proposed model. The context encoder is a uni-directional GRU network ψ that sequentially integrates the sentence features \mathbf{q}^t from ϕ and generates the final query feature $\bar{\mathbf{x}}^t$: $\bar{\mathbf{x}}^t = \psi(\mathbf{q}^t, \bar{\mathbf{x}}^{t-1})$. $\bar{\mathbf{x}}^0$ is initialized as a zero vector. The image encoder maps the mean-pooled features of ResNet152 [49] into a one-dimensional feature vector $\bar{\mathbf{v}}$ via a linear projection. The ResNet model is pre-trained on ImageNet [30]. The model is trained to optimize the cosine similarity between $\bar{\mathbf{x}}^t$ and $\bar{\mathbf{v}}$ by a triplet loss with hard negatives as in [35]. (2) **R-HRE**: a model similar to baseline (1) but is trained with the region features $\{\mathbf{v}_j\}_{j=1}^N$, as in the proposed method. Specifically, the model learns to optimize the similarity term $s(\bar{\mathbf{x}}^t, \mathbf{I})$ defined in Eq.(3.1) by a triplet loss with hard negatives similar to L_e on one state vector. (3) **R-RE**: a model similar to baseline (2) but instead of using a hierarchical text encoder, this baseline uses a single uni-directional GRU network which encodes the concatenation of the queries. (4) **R-RankFusion**: a model where each query is encoded by a uni-directional GRU network and each image is represented as a set of region features $\{\mathbf{v}_j\}_{j=1}^N$. The ranks of all images are computed separately for each turn. The final ranks of the images are represented as the averages of the per-turn ranks.

Implementation details. We try to keep consistent configurations for all the models in our experiments to better evaluate the contribution of each component. In particular, all the models are trained with 10-turn queries ($T = 10$). We use ten turns as we'd like to track and demonstrate the performance of all methods in both

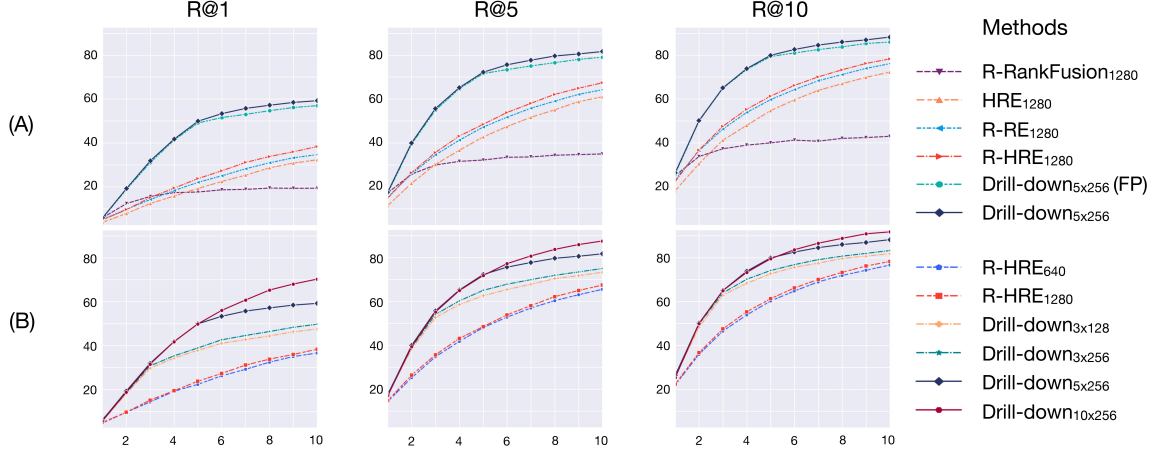


Figure 3.3: Quantitative evaluation of our models and the baselines. (A) Comparison of models using query representations of the same memory size; (B) Comparison of the models using query representations of different memory sizes. The horizontal axis represents the query turn.

short-term and long-term scenarios. For each image, we extract the top 36 regions ($N = 36$) detected by a pretrained Faster RCNN model, following [3]. Each embedded word vector has a dimension of 300 ($E = 300$). In all our experiments, we set the temperature parameter σ to 9, the margin parameter α to 0.2, the discount factor γ to 1.0, and the trade-off factor μ to 0.1. For optimization, we use Adam [61] with an initial learning rate of $2e - 4$ and a batch size of 128. We clip the gradients in the back-propagation such that the norm of the gradients is not larger than 10. All models are trained with at most 300 epochs, validated after each epoch. The models which perform best on the validation set are used for evaluation.

Evaluation metrics To measure the retrieval performance, we use the common R@K metric, i.e., recall at K - the ratio of queries for which the target image is among the top-K retrieved images. The R@1, R@5 and R@10 scores at each turn are reported as shown in Fig. 3.3.

Methods	HRE/R-RE ₁₂₈₀	R-HRE _{640/1280}	Drill-down _{3×128} / _{3×256} / _{5×256} / _{10×256}
# Query Rep.	1280	640 / 1280	384 / 768 / 1280 / 2560
# Image Rep.	1280 / 36 × 1280	36×640 / 36 × 1280	36 × 128 / 36 × 256 / 36 × 256 / 36 × 256
# Parameters	22820k	9866k / 22820k	4861k / 5830k / 5830k / 5830k

Table 3.1: Sizes of the query/image representations and the parameters in our models and the baselines.

3.4.1 Results on Simulated User Queries

Due to the lack of existing benchmarks for multiple turn image retrieval, we use the annotated region captions in Visual Genome to mimic the user queries. As region captions focus more on invariant information, such as image contents, and convey fewer irrelevant signals, such as different speaking/writing styles, they could be seen as the common "abstracts" of real queries in different forms. While we agree that strong supervisory signals such as real user queries could bridge the domain gap and would like to explore further in this direction, we choose at this stage to use only "weak but free" signals and investigate their potentials of being generalized to real scenarios. First, we compare our method against the baseline models when using query representations of the same memory size. In particular, we use 5 state vectors in our model ($M = 5$), each with a dimension of 256. Accordingly, the baseline models use a 1280-d query vector. Figure 3.3(A) shows the per-turn performance of the models on the test set. Here Drill-down_{5×256}(FP) indicates the supervised pre-trained model with the fixed policy, and Drill-down_{5×256} indicates the jointly optimized model with a learned policy. Both the R-RE₁₂₈₀ and R-HRE₁₂₈₀ baselines perform better than the HRE₁₂₈₀ model, demonstrating the benefit of incorporating region features. R-HRE₁₂₈₀ is superior to R-RE₁₂₈₀, demonstrating the benefit of hierarchical context encoding. R-RankFusion₁₂₈₀ performs inferior to all other models. Note that it also requires more memory to store the ranks of all images at each turn. Our models significantly outperform all baselines by a large margin. On the other hand, we observe that the performance of our model will degrade when different queries have to

share the same state vector. For example, after the 5th turn, the Drill-down_{5×256}(FP) model gains less improvement from each new query. Drill-down_{5×256} further improves Drill-down_{5×256}(FP) by learning to distribute the queries into the most rewarding state vectors.

To investigate the design space of the query representation, we further explore variants of our model with different numbers of state vectors and feature dimensions. Table 3.1 shows the sizes of the query/image representations and the parameters used in our models and the baselines. Note that the R-RankFusion and R-RE models have the same size of query/image representations and parameters. Here Drill-down_{M×D} indicates the model with M state vectors, each with a dimension of D. As shown in Figure 3.3(B), while both Drill-down and the R-HRE baseline can be improved by increasing the feature dimension, using more state vectors gains significantly more improvements with the same, or even less memory budget. For example, Drill-down_{3×128} significantly outperforms R-HRE₁₂₈₀ with 3 times fewer query features, 10 times fewer region features and 4 times fewer parameters. The highest performance is achieved by the model which stores each query in a distinct state vector: 10 state vectors for 10-turn queries. Integrating multiple queries into the same state vector could make the model “forget” the responses from earlier turns, especially when they activate the same semantic space as the new query.

Figure 3.4 provides qualitative examples of the Drill-down_{3×128} model. Here the arrows indicate the predicted state vectors used to incorporate the queries. We show the top-3 regions of the target images that have the highest similarity scores with each state vector (illustrated with the same color). We observe that the model tends to group queries with entities that potentially coincide with each other. However, it could also lead to the “forgetting” of earlier queries. For instance, in the first example, when aggregating the queries “*child in a stroller*” and “*woman in a dress*” in order,

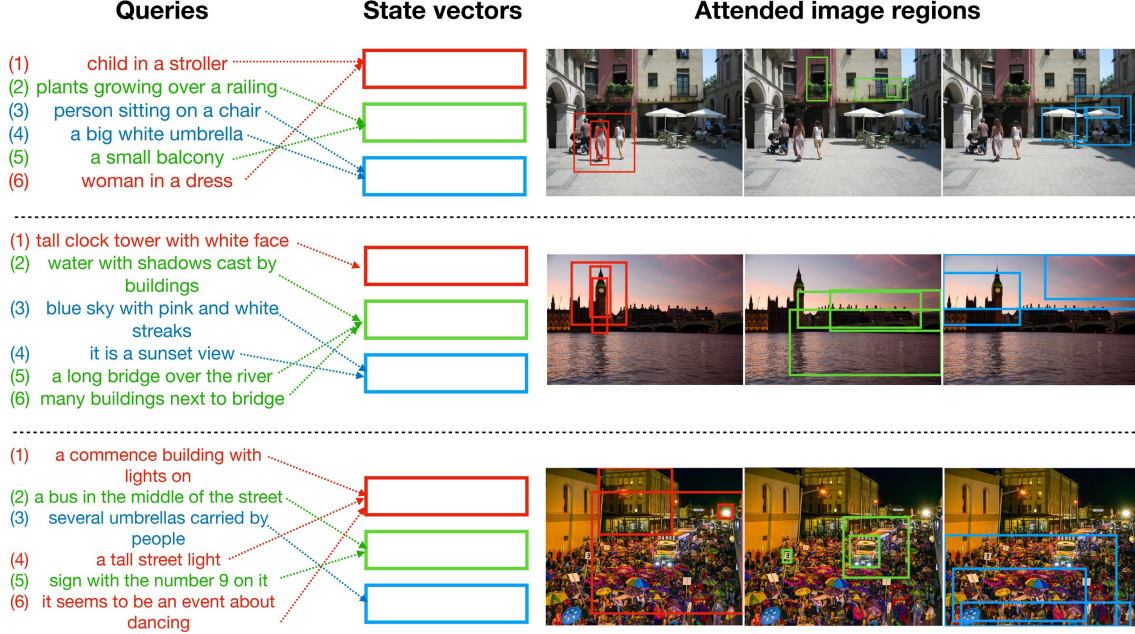


Figure 3.4: Qualitative examples of $\text{Drill-down}_{3 \times 128}$. The sequential queries and the corresponding state vectors used to integrate them are shown on the left; The top-3 regions of the target images attended by each state vector are shown on the right, with the same color as the corresponding state vector. Note that all these target images rank top-1 given the input queries.

the model tends to focus on “*woman*” while forgetting information about “*child*”, as “*woman*” and “*child*” potentially activate the same semantic subspace.

3.4.2 Results on Real User Queries

We evaluate our method with the queries from crowdsourced human users via a multi-round interactive system adapted from [19]. Given a target image, a user is asked to search for it by providing descriptions of the image content. The system shows top-5 retrieved images to the user per turn as context so that the user can improve the results by providing additional descriptions. This process is repeated until the image is found or it reaches 5 turns. We sample 80 random images from the test set and evaluate HRED_{1280} , R-HRED_{1280} and $\text{Drill-down}_{3 \times 256}$ on these images respectively. Each image is viewed by 3 different users. For each model, the best result on each



Figure 3.5: Examples of real user queries and the top-1 images from $\text{Drill-down}_{3 \times 256}$.

image is selected across users to ensure high quality responses.

As shown in Fig. 3.6, most users ($> 80\%$) successfully find the target image within 5 turns, demonstrating the effectiveness of the multi-round search paradigm and the quality of using region captions for training. In particular, $\text{Drill-down}_{3 \times 256}$ consistently outperforms HRE_{1280} and R-HRE_{1280} on all evaluation metrics. On the other hand, as real user queries have more flexible forms, e.g. longer sentences, repeated descriptions of the same region, etc, we also observe smaller performance gaps between our method and the baselines. We believe further efforts such as real query data collection are needed to systematically fill this domain gap. Figure 3.5 shows example real user queries and the retrieval sequences using $\text{Drill-down}_{3 \times 256}$.

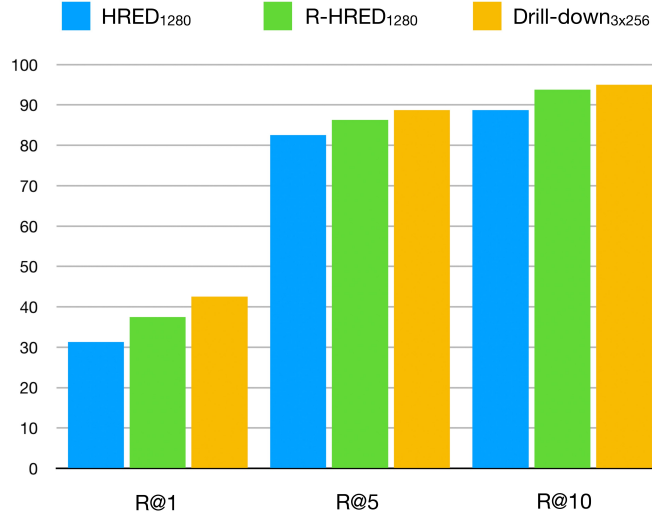


Figure 3.6: Human subject evaluation of the HRE₁₂₈₀, R-HRE₁₂₈₀ baselines and our Drill-down_{3x256} model.

3.5 Summary

In this chapter, we present Drill-down, a framework that is efficient and effective in interactive retrieval of specific images of complex scenes. Our method explores in depth and addresses several challenges in multiple round retrievals with natural language queries such as the compactness of query state representations, and the need for region-aware features. It also demonstrates the effectiveness of training a retrieval model with region captions as queries for interactive image search under human evaluations.

Chapter 4

Learning Visual Similarity of Images using Reranking Transformers

4.1 Introduction

In the Text2Scene and Drill-down projects, we leverage compositional (part-based) representations to model the correlation between visual and language data. In this chapter, we explore learning the correlation between pure visual data using local-based representations. Particularly, we study the problem of instance-level recognition/retrieval.

Instance recognition [113] aims to visually recognize an object/scene instance in an image. This is distinct from category-level recognition (e.g. the ILSVRC image classification [30]) that identifies only the object class. It is also a challenging problem in e-commerce where the objective is to find a specific product from a large image collection, and place identification where the objective is to use features from public landmarks to infer the identity of a place. Since the number of instances is typically

much larger than the number of categories of objects, instance recognition is typically cast as image retrieval instead of classification, and usually involves both metric learning and local feature matching strategies for reranking.

Over the last decade, instance recognition/retrieval continues to be a major focus of research. Early systems [94, 113] leverage hand-crafted local descriptors (e.g. Bag-of-Words (BOW)) and matching algorithms. With the advent of convolutional neural networks (CNNs) [47, 66], recent approaches incorporate both global and local descriptors extracted from deep learning models [8, 88]. On the one hand, global descriptors summarize the content of an image into a single vector, leading to a compact representation for large-scale search. On the other hand, local descriptors encode the spatial layout of visual elements for patch-level matching between images, which are shown to be essential to high retrieval precision [16, 126]. However, computing the similarity for every possible pair of images can be prohibitively expensive for large-scale search. Thus, the best existing methods [16, 112] typically first use a global descriptor to reduce the solution space, then use local descriptors to *re-rank* the top retrieved images. While extensive progress has been made towards learning more expressive global/local representations, fewer techniques are developed for local features based similarity learning. State-of-the-art approaches still rely on classic matching techniques, such as geometric verification (GV) [94] and aggregated selective match kernels (ASMK) [125]. Geometric verification assumes object instances are rigid and that local matches between images can be estimated as an affine transformation using RANSAC [37]. It is also an expensive process which requires iterative optimization on a large set of local descriptors. The performance of geometric verification deteriorates when deformable objects or challenging conditions (e.g. large variations in viewpoint or illumination) are present. ASMK focuses more on aggregating the similarities between features without explicitly modeling the geometric alignment, but requires off-line clustering and encoding procedures. It is mainly used as a global retrieval

technique in previous literature. Both geometric verification and ASMK require large amounts of local descriptors (e.g. 1000 per image) to ensure retrieval performance.

In this chapter, we take the initiative to advance the techniques for similarity learning using global/local descriptors. We propose *Reranking Transformers* (RRTs) [123], which learn to predict the similarity of an image-pair directly. Our method is general, flexible, and can be used as a drop-in replacement for other reranking approaches such as geometric verification. We conduct detailed experiments showing that as either a drop-in replacement or trained together with a global metric learning approach, the proposed method is the top performing across the standard benchmarks for instance recognition. Our approach is inspired by the Transformer architecture [130] which leads to significant improvements in many natural language processing [31, 75] and Vision-and-Language tasks [71, 78]. Most recently, it has also been introduced to pure visual tasks, such as image synthesis [91], recognition [33] and object detection [17]. Distinct from conventional neural networks (e.g. CNN, RNN), Transformers capture long-range dependencies among the input elements using self-attentions. It was initially designed for sequential modeling [130]. To the best of our knowledge, our work is the first to adopt transformers for a visual task involving the analysis of image pairs in the context of reranking image search results.

The proposed method is lightweight. Compared with the convolutional neural network (CNN) based feature extractors ((i.e. ResNet 50/101 as in [16])), which typically have over 20 million parameters (e.g. 25/44 million in ResNet 50/101), the proposed model, used as an extra module for similarity measurement in addition to the CNN backbone, has only 2.2 million parameters. It can also be easily parallelized so that re-ranking the top 100 retrieved images requires only a single neural network forward pass, allowing for more efficient model inference. Similar to geometric verification, our method aims to learn the region-wise alignment of an image-pair but with a more straightforward pipeline. As shown in Fig. 4.1, our method directly

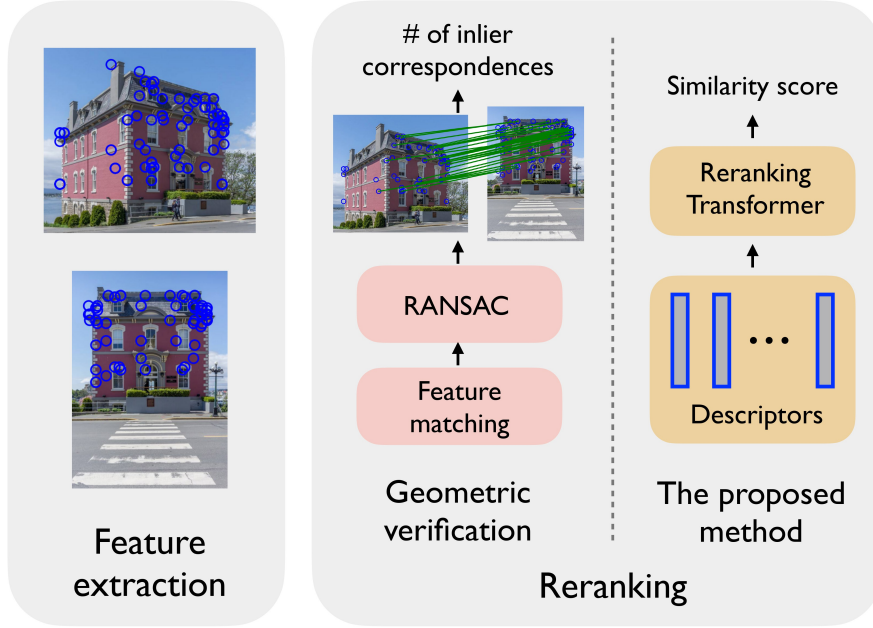


Figure 4.1: Top performing instance recognition methods often rely on reranking the top results using a score such as the number of inlier correspondences from geometric verification. We propose to replace this step with a Reranking Transformer (RRT) that can be learned with the underlying representations of the images.

predicts a similarity score of the matching images, instead of estimating a homography, which may be challenging under large viewpoint changes or may not even exist for deformable objects. Our method requires much fewer descriptors (i.e. 500 per image), but achieves superior performance, especially for challenging cases. Also, in current state-of-the-art models, the feature extraction and the matching modules are separately optimized, which is undesirable as it may lead to suboptimal feature representation. In this work, we first perform experiments using pretrained feature extractors. We then demonstrate the benefit of integrating the feature extractor and the proposed model into a unified framework and evaluate the resulting unified model on a benchmark of product images: Stanford Online Products [116]. We show that, by jointly optimizing the feature representation with our model, the re-ranking performance can be further improved.

Contributions. (1) We propose *Reranking Transformers* (RRTs), a lightweight

and effective model which learns to estimate the similarity of an image pair based on their global and local descriptors; (2) Compared with existing methods, the proposed method requires much fewer local descriptors and can be easily parallelized so that re-ranking the top neighbors for each query requires only a single neural network forward-pass; (3) We perform extensive experiments on three instance retrieval benchmarks: Revisited Oxford/Paris [98] and Google Landmarks v2 [138], and show that RRTs outperform prior reranking methods across a variety of settings. The results demonstrate the effectiveness of the transformer architecture on learning the visual correlation between images; (4) We further show the benefit of optimizing the proposed model jointly with the feature extractor on the Stanford Online Products (SOP) [116] benchmark.

4.2 Related Work

Local features for instance recognition/retrieval. Hand-crafted local descriptors [82], e.g. SIFT [77], were widely used in the earliest instance retrieval work [86, 113]. These descriptors were believed to be more invariant to image changes such as illumination, occlusion and truncation than global signatures, e.g. GIST [89]. Recently, local features extracted from convolution neural networks (CNN) have been shown to be more effective on various retrieval tasks [34, 88, 112, 124]. Siméoni *et al.* [112] detects local features from a pretrained CNN backbone using maximally stable extremal regions (MSER) originally developed by Matas *et al.* [81], while [16, 34, 88, 124, 126] propose to jointly learn feature representation and detection by either performing a non-local-maximum suppression on the feature responses [34, 126], or incorporating visual attentions [16, 88, 124]. The detected local descriptors are usually used for geometric verification [94] or ASMK [125]. Different from these works, we focus on similarity learning rather than feature detection or representation learning.

Global features for instance recognition/retrieval. Compared to local features, global descriptors provide a compact representation of an image for large-scale search. Most of the existing global descriptors are extracted from CNN models [8, 41] by spatially pooling the two-dimensional feature responses [7, 97, 128], which may not be ideal for modeling region-wise relations across images. Thus, state-of-the-art systems typically either use the global descriptor to reduce the solution space and then use local descriptors to *re-rank* the nearest neighbors, or encode the local descriptors using a large visual codebook, followed by image matching with an aggregated selective match kernel [124–126]. This work mainly follows the retrieve-and-rerank paradigm.

Reranking for instance recognition/retrieval. Geometric verification is the dominant image reranking approach and widely used in both traditional [94] and more recent work [16, 88, 112]. Geometric verification assumes rigid objects and seeks to estimate a linear transformation between images by iteratively aligning local descriptors. Inspired by text retrieval, query expansion techniques have also been introduced for image retrieval [25, 26, 127]. These methods differ from geometric verification and our work as they rely on analyzing the local nearest neighbor graph for each query during testing. On the other hand, diffusion based approaches [9, 10, 32, 52, 149] aim to learn the structure of the data manifold by similarity propagation over the global affinity graph built on a query and all the gallery images, which is nontrivial to scale. Overall, the motivation of image reranking is to make better use of test-time knowledge to refine retrieval results. Our work shares the same vision with this line of research but focuses more on learning the similarity of an image-pair directly.

Transformers for visual tasks. Transformers have become the dominant model architecture in natural language processing [31, 75]. Recently, it has also been

introduced to vision-and-language [71, 78] and pure vision tasks [17, 91]. Parmar et al [91] develop a transformer based autoregressive model for image synthesis. Carion et al [17] casts object detection as a direct set prediction problem using transformers. As the key ingredient of the transformer architecture, the self-attention mechanism has also been studied for visual recognition [12, 99, 151]. These prior works apply transformers for single image predictions while we leverage transformers to learn the visual relation of an image-pair.

4.3 Method

4.3.1 Background

We study the problem of learning the visual similarity of an image pair using global/local descriptors. In particular, we follow the retrieve-and-rerank paradigm [16, 112] and exploit a hierarchical framework where a global descriptor of the query is first used to retrieve the top-ranked neighbors and local descriptors are then used to *rerank* these candidates. The dominant approach for the latter task is geometric verification (GV) [94]. In modeling the relation of an image pair capturing the same object/scene, geometric verification assumes the underlying object/scene is rigid and seeks to estimate an affine transformation between the local descriptors of the image pair using RANSAC [37]. Despite its simplicity, it is shown to be surprisingly effective if sufficient local descriptors (e.g. 1000 per image) are provided. Given enough computing resources, geometric verification is still considered to be the state-of-the-art [112]. We explore an alternative solution to the reranking task by introducing a transformer based matching algorithm. The proposed method does not require the rigidity prior inherent to geometric verification and can potentially model more challenging objects/scenes.

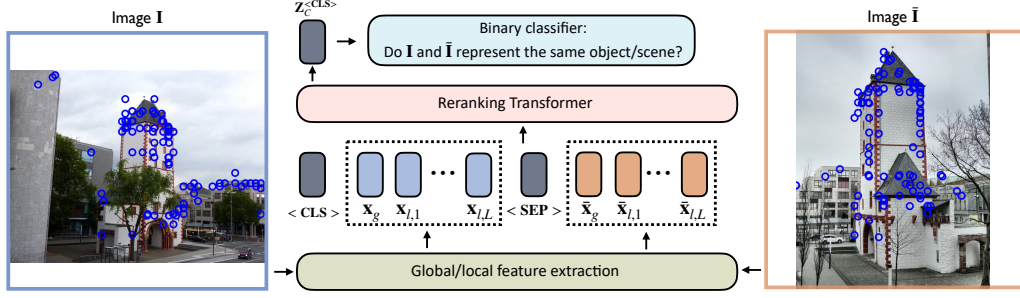


Figure 4.2: Illustration of the *Reranking Transformer* (RRT) model. The input of RRT is a sequence of global and local descriptors (circled in blue) extracted from an image-pair $(\mathbf{I}, \bar{\mathbf{I}})$. This sequence, together with two special tokens, are fed into a multi-layer transformer model which produces a similarity score of $(\mathbf{I}, \bar{\mathbf{I}})$. The model is trained to optimize a binary cross entropy loss.

4.3.2 Attention Modules in Transformers

First, we briefly review the key ingredients in the Transformer architecture: Single-Head Attention (SHA) and Multi-Head Attention (MHA).

Single-Head Attention (SHA): The input of a SHA layer comprises three sets of variables: the queries $\mathbf{Q} := \{\mathbf{q}_i \in \mathbb{R}^{d_q}\}_{i=1}^N$, the keys $\mathbf{K} := \{\mathbf{k}_j \in \mathbb{R}^{d_k}\}_{j=1}^M$, and the values $\mathbf{V} := \{\mathbf{v}_j \in \mathbb{R}^{d_v}\}_{j=1}^M$. Here, d_q , d_k , d_v are the dimensions of the corresponding feature vectors, while N and M are the sequence lengths. SHA produces a new feature sequence where each vector is a linear combination of $\{\mathbf{v}_j\}$. In doing this, \mathbf{Q} , \mathbf{K} , \mathbf{V} are first linearly projected as $\bar{\mathbf{Q}} = \mathbf{Q}W^Q$, $\bar{\mathbf{K}} = \mathbf{K}W^K$, $\bar{\mathbf{V}} = \mathbf{V}W^V$, using parameter tensors: $W^Q \in \mathbb{R}^{d_q \times d}$, $W^K \in \mathbb{R}^{d_k \times d}$, $W^V \in \mathbb{R}^{d_v \times d}$, where d is the new feature dimension. The output of a SHA layer is computed as:

$$\text{SHA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) := \text{SOFTMAX}\left(\frac{\bar{\mathbf{Q}}\bar{\mathbf{K}}^T}{\sqrt{d}}\right)\bar{\mathbf{V}} \quad (4.1)$$

Multi-Head Attention (MHA): Like SHA, MHA takes \mathbf{Q} , \mathbf{K} , \mathbf{V} as input and

comprises multiple SHA modules:

$$\begin{aligned} \text{MHA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) &:= [\text{HEAD}_1; \cdots; \text{HEAD}_h] W^O \\ \text{HEAD}_i &:= \text{SHA}_i(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \end{aligned} \tag{4.2}$$

Here $[\cdot]$ denotes the concatenation operator, h is the number of the SHA heads. $W^O \in \mathbb{R}^{d \times (hd)}$ is a linear projection with an output dimension of size $h \times d$.

4.3.3 Model

With the fundamental building blocks defined above, we introduce the detailed formulation of our model:

Image representations: An image \mathbf{I} is represented by a global descriptor of a dimension d_g : $\mathbf{x}_g \in \mathbb{R}^{d_g}$ and a set of L local descriptors: $\mathbf{x}_l = \{\mathbf{x}_{l,i} \in \mathbb{R}^{d_l}\}_{i=1}^L$, each of a dimension d_l . Both \mathbf{x}_g and \mathbf{x}_l are extracted from a CNN backbone (to be discussed in Sec. 4.4.2). Optionally, each $\mathbf{x}_{l,i}$ is associated with a coordinate tuple $\mathbf{p}_{l,i} = (u, v) \in \mathbb{R}^2$ and a scale factor $s_{l,i} \in \mathbb{R}$, indicating the pixel location and image scale where $\mathbf{x}_{l,i}$ is extracted from. In this work, $s_{l,i}$ is an integer, indexing a set of pre-defined image scales.

Input: As a sequence transduction model [31, 75], Transformers take as input a list of “tokens” (e.g. $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ in Sec. 4.3.2). In image retrieval, these “tokens” can be derived from the features of an image-pair $(\mathbf{I}, \bar{\mathbf{I}})$. Following the BERT transformer encoder [31], we define the input as:

$$\begin{aligned} \mathbf{X}(\mathbf{I}, \bar{\mathbf{I}}) &:= [\langle \text{CLS} \rangle; f_g(\mathbf{x}_g); f_l(\mathbf{x}_{l,1}); \cdots; f_l(\mathbf{x}_{l,L}); \\ &\quad \langle \text{SEP} \rangle; \bar{f}_g(\bar{\mathbf{x}}_g); \bar{f}_l(\bar{\mathbf{x}}_{l,1}); \cdots; \bar{f}_l(\bar{\mathbf{x}}_{l,L})], \end{aligned} \tag{4.3}$$

where:

$$\begin{aligned}
f_g(\mathbf{x}_g) &:= \mathbf{x}_g + \alpha; \\
f_l(\mathbf{x}_{l,i}) &:= \mathbf{x}_{l,i} + \varphi(\mathbf{p}_{l,i}) + \psi(s_{l,i}) + \beta \\
\bar{f}_g(\bar{\mathbf{x}}_g) &:= \bar{\mathbf{x}}_g + \bar{\alpha}; \\
\bar{f}_l(\bar{\mathbf{x}}_{l,i}) &:= \bar{\mathbf{x}}_{l,i} + \varphi(\bar{\mathbf{p}}_{l,i}) + \psi(\bar{s}_{l,i}) + \bar{\beta}.
\end{aligned} \tag{4.4}$$

Here, $\langle \text{CLS} \rangle$ is a special token used for summarizing the signals from both images. $\langle \text{SEP} \rangle$ is an extra separator token. $\alpha, \bar{\alpha}, \beta, \bar{\beta}$ are one dimensional segment embeddings, being used to distinguish the global and local descriptors of \mathbf{I} and $\bar{\mathbf{I}}$. φ is a linear position embedding, as used in [17]. ψ is a linear embedding taking the scale index $s_{l,i}$ as input.

Model architecture: With input $\mathbf{X}(\mathbf{I}, \bar{\mathbf{I}})$, we define a multi-layer transformer model, where each layer is formulated as:

$$\begin{aligned}
\bar{\mathbf{Z}}_{i+1} &= \text{LAYERNORM}(\mathbf{Z}_i + \text{MHA}(\mathbf{Z}_i)), \\
\mathbf{Z}_{i+1} &= \text{LAYERNORM}(\text{MLP}(\bar{\mathbf{Z}}_{i+1})), \\
\text{MLP}(\bar{\mathbf{Z}}_{i+1}) &= \text{RELU}(\bar{\mathbf{Z}}_{i+1} W_1^T) W_2^T, \\
i &= 0, \dots, C-1.
\end{aligned} \tag{4.5}$$

In this setting, the Q, K, V features for MHA are the same set of vectors \mathbf{Z}_i , with $\mathbf{Z}_0 = \mathbf{X}(\mathbf{I}, \bar{\mathbf{I}})$. MLP is a two-layer perceptron with parameter matrices $W_1 \in \mathbb{R}^{(hd) \times d_c}$ and $W_2 \in \mathbb{R}^{d_c \times (hd)}$, and an intermediate dimension d_c . LAYERNORM is a layer normalization function proposed in [6]. The model includes C transformer layers in total.

Training objective: Our model is trained to optimize a binary cross entropy

loss:

$$E(\mathbf{I}, \bar{\mathbf{I}}) = \text{BCE}(\text{SIGMOID}(\mathbf{Z}_C^{(\text{CLS})} W_z^T), \mathbb{1}(\mathbf{I}, \bar{\mathbf{I}})), \quad (4.6)$$

$$\mathbb{1}(\mathbf{I}, \bar{\mathbf{I}}) = \begin{cases} 1.0, & \mathbf{I}, \bar{\mathbf{I}} \text{ represent the same instance} \\ 0.0, & \text{otherwise} \end{cases} \quad (4.7)$$

$\mathbf{Z}_C^{(\text{CLS})} \in \mathbb{R}^{hd}$ is a feature vector, corresponding to the $\langle \text{CLS} \rangle$ token. It is extracted from the last transformer layer. $W_z^T \in \mathbb{R}^{(hd) \times 1}$ is a linear function mapping $\mathbf{Z}_C^{(\text{CLS})}$ to a logit scalar. $\mathbb{1}(\mathbf{I}, \bar{\mathbf{I}})$ is an indicator function which equals to one when \mathbf{I} and $\bar{\mathbf{I}}$ represent the same object, or zero otherwise. Fig. 4.2 provides an illustration of the proposed model.

4.4 Experiments

Next, we describe the datasets we use to evaluate our approach, and details about our implementation.

4.4.1 Datasets

We perform experiments on three datasets, Google Landmarks v2 [138], Revisited Oxford/Paris [98], and Stanford Online Products [116]. The former two are used for instance matching and showcase landmark locations where geometry verification plays a prominent role. The Stanford Online Products dataset showcases images of products that may be deformable so that correspondences between images cannot be modeled with an affine transformation. It has been mostly used for metric learning. We briefly describe each of these resources:

GLDv2: Google Landmarks v2 (GLDv2) [138] is a new benchmark for instance recognition that includes over five million images from 200k natural landmarks. As the

proposed Reranking Transformer has limited parameters (e.g. 2.2 million), we sample a small subset of the images from the “v2-clean” split of GLDV2 for training. The “v2-clean” split consists of 1,580,470 images from 81,313 landmarks. We *randomly* sample 12,000 landmarks where each landmark has *at least* 10 images. For each landmark, we *randomly* sample *at most* 500 images. This results in 322,008 images, which is 20% of the “v2-clean” split and 8% of the original training set. For testing, we evaluate on the standard test set for the retrieval task, which contains 1,129 query images and 761,757 gallery images.

ROxf and **RPar**: Revisited Oxford (ROxf) and Paris (RPar) [98] are standard benchmarks for instance recognition, which have 4,993 and 6,322 gallery images respectively. They both have 70 query images, each with a bounding box depicting the location and span of the prominent landmark. An extra distractor set (R1M) with 1,001,001 images is included for large-scale experiments. We follow the standard evaluation protocol [16, 98] and crop the query image using the provided bounding box. We report mean Average Precision (mAP) on the Medium and Hard setups.

SOP: To further investigate the benefit of jointly optimizing the feature representation and our Reranking Transformer, we perform experiments on a dataset of product images: Stanford Online Products (SOP) [116]. SOP is a commonly used benchmark for metric learning [13, 15, 104, 105, 109, 135, 139], which includes 120,053 images, 59,551 for training, 60,502 for testing. We follow the evaluation protocol for metric learning and measure the R@K scores.

4.4.2 Implementation

Experiments on pretrained features: As this work mainly focuses on similarity learning rather than feature learning, we leverage image descriptors obtained from state-of-the-art feature extractors. In particular, we use the pretrained DELG models

provided by [16] with ResNet50 [47] as the CNN backbone. DELG provides a unified framework for global/local feature extraction. The local descriptors are extracted from 7 image scales ranging from 0.25 to 2.0, each with a dimension of 128. The global descriptor is extracted from 3 image scales: $\{\frac{1}{\sqrt{2}}, 1, \sqrt{2}\}$, with a dimension of 2048. We apply an extra linear projection to the global descriptor to reduce its dimension to 128. In the original DELG model, each local descriptor comes with an attention score. The top 1000 local descriptors with the highest attention scores are selected for image reranking. We observe that RRT does not require this amount of descriptors, and the retrieval performance saturates at 500 local descriptors. Thus, in our experiments we choose the top 500 local descriptors and set $L = 500$, $d_g = d_l = 128$. For images with fewer descriptors, we pad the feature sequence with empty vectors and use a binary attention mask, as in BERT [31], to indicate the padding locations. Both the global and local features are L2 normalized to unit norm. During training, the positive image is randomly sampled from the images sharing the same label as the query. The negative image is randomly sampled from the top-100 neighbors returned by the global retrieval, which have a different label from the query. DELG are pretrained on both Google Landmarks (GLD) v1 [88] and v2-clean [138]. Thus, we perform experiments on two sets of descriptors from these two pretrained models. For the architecture, we use 4 SHA heads ($h = 4$) and 6 transformer layers ($C = 6$). d_q , d_k , d_v and d in SHA are set to 128, while d_c in MLP (Eq. 4.5) is set to 1024. The number of learnable parameters is 2,243,201, which is 9% of the amount in ResNet50. The model is trained with AdamW [76] for 15 epochs, using a learning rate of 0.0001 and a weight decay of 0.0005.

Experiments on SOP: The DELG descriptors are extracted from multiple image scales, it is nontrivial to jointly optimize them with RRT. Instead, we perform experiments on SOP [116] using a single image scale, following the protocol for met-

ric learning [135]. During training, each image is randomly cropped to 224×224 , followed by a random flip. During testing, each image is first resized to of 256×256 then cropped at the center to 224×224 . We use ResNet50 and extract features from the last convolutional layer, which leads to 49 (7×7) local descriptors for each image. We use all these local descriptors. The global descriptor is obtained by spatially averaging the local responses. The RRT architecture and most of the training details remain the same as in the DELG experiments. Here we only describe the main differences. The global retrieval model is trained with a contrastive loss, as in [135]. Different from [135], we do not rely on a cross batch memory but simply use a large batch size of 800. As all the local features are used, we do not incorporate the global descriptor term $(f_g(\mathbf{x}_g), \bar{f}_g(\bar{\mathbf{x}}_g))$ in Eq. 4.3. We also drop the scale embedding (ψ) as only one image scale is used. The global model is trained using SGD with Nesterov momentum for 100 epochs, using a learning rate of 0.001, a weight decay of 0.0005 and a momentum of 0.9. The learning rate drops by a factor of 10 after 60 and 80 epochs. We train an RRT model on top of the pretrained global model, either freezing or finetuning the CNN backbone. Both models are trained with AdamW [76] for 100 epochs, using a learning rate of 0.0001. The learning rate drops by a factor of 10 after 60 and 80 epochs. We implement RRT in PyTorch [92].

Position embedding: For the experiments on the DELG descriptors, we observe a limited benefit in applying position embeddings and do not use the φ term in Eq. 4.4. For the experiments on SOP, we observe the position embedding is indeed helpful, especially when the feature representations are jointly optimized with the Reranking Transformer.



Figure 4.3: An extreme example where the target images are some crops of the query. In this case the global descriptor + cosine similarity retrieval paradigm is not ideal.

4.5 Results

We demonstrate the effectiveness of Reranking Transformers (RRTs) across different settings, benchmarks and use cases.

4.5.1 Baselines

We consider geometry verification [94] and α QE [26] as the main baselines as they share the same spirit with our method: they make better use of test-time information. When comparing the query and target images, geometry verification attends to *different* sub-regions of the query image when the target image is *different*, and vice versa, which is very similar to the proposed Reranking Transformers (RRTs). α QE also leverages test-time knowledge, but relies on analyzing the local affinity graph created during testing. Incorporating test-time knowledge is the key motivation of image reranking, and we believe that the attention modules in the transformer architecture (sec. 4.3.2) is very suitable for this task. It also distinguishes our method from most of the previous approaches that focus on feature learning. Note that we use pretrained and fixed feature representation in most of our experiments.

Fig. 4.3 provides another intuitive example of the partial-matching cases. In

this example, the target images are some crops of the query. We believe the global descriptor + cosine similarity paradigm is not ideal for this case, as no matter how large is the global descriptor, it contains irrelevant information that hinders the cosine similarity measurement. On the other hand, the cross-image attention in our model is shown to be very helpful for these situations. We trained a variant of our model that *disables* cross-image attentions and uses the cosine-similarity of the $\mathbf{Z}_C^{(\text{CLS})}$ vectors as the score. This baseline performs even worse than the global-only retrieval. We posit that without finetuning the backbone, the extra transformer module may not help a lot with feature learning. On the other hand, it also demonstrates that any benefit of the proposed method can only be from the cross-image attentions.

Aggregated Selective Match Kernel (ASMK) [125] was previously not used for image reranking but as a global-only retrieval approach. Specifically, it proposes to create a set of new filters (i.e. visual codebook) by clustering. It then remaps/aggregates the local descriptors of each image into a global vector. We perform experiments on ASMK as it also relies on local descriptors.

4.5.2 Comparison with Geometric Verification

We perform experiments on comparing GV and RRT using the same set of descriptors, i.e. the pretrained DELG [16] descriptors. Following the protocol in [16], given a query, we use its global descriptor to retrieve a set of top-ranked images. The top-100 neighbors are reranked by GV and RRT. We present results on two sets of descriptors: DELG pretrained on GLD v1 [88] and v2-clean [138].

On \mathcal{ROxf} and \mathcal{RPar} , both GV and RRT significantly outperform global-only retrieval, as shown in Table 4.1. RRT shows further advantages over GV, with much fewer local descriptors. On \mathcal{ROxf} (+ $\mathcal{R1M}$), RRT performs on par with GV on the Medium setup and consistently better on the Hard setup. On \mathcal{RPar} (+ $\mathcal{R1M}$), RRT

Method	# local desc.	# Reranked images	Desc. version	Medium				Hard			
				\mathcal{ROxf}	$+\mathcal{R1M}$	\mathcal{RPar}	$+\mathcal{R1M}$	\mathcal{ROxf}	$+\mathcal{R1M}$	\mathcal{RPar}	$+\mathcal{R1M}$
DELG global	0	0	v1	69.7	55.0	81.6	59.7	45.1	27.8	63.4	34.1
GV	1000	100	v1	75.4	61.1	82.3	60.5	54.2	36.8	64.9	34.8
RRT (ours)	500	100	v1	75.5	61.2	82.7	60.7	56.4	37.0	68.6	37.5
GV*	1000	200	v1	77.2	63.1	82.5	60.9	55.4	37.9	63.2	34.7
RRT (ours)	500	200	v1	77.9	63.5	84.4	62.1	58.8	39.5	71.6	39.5
DELG global	0	0	v2-clean	73.6	60.6	85.7	68.6	51.0	32.7	71.5	44.4
GV	1000	100	v2-clean	78.3	67.2	85.7	69.6	57.9	43.6	71.0	45.7
RRT (ours)	500	100	v2-clean	78.1	67.0	86.7	69.8	60.2	44.1	75.1	49.4
GV*	1000	200	v2-clean	79.2	68.2	85.5	69.6	57.5	42.9	67.2	44.5
RRT (ours)	500	200	v2-clean	79.5	68.6	87.8	71.5	62.5	46.3	77.1	52.3

Table 4.1: Comparison to geometric verification on Revisited Oxford/Paris [98]. The mAP scores on the Medium ($+\mathcal{R1M}$) and Hard ($+\mathcal{R1M}$) setups are reported. Results marked by * are evaluated by us using the public models provided by [16].

Method	# local desc.	Desc. version	Retrieval	
			Public	Private
DELG global	0	v1	18.3	20.4
GV	1000	v1	20.4	22.3
RRT (ours)	500	v1	21.5	23.1
DELG global	0	v2-clean	22.2	24.2
GV	1000	v2-clean	—	24.3
RRT (ours)	500	v2-clean	24.6	27.0

Table 4.2: Comparison to geometric verification on the GLDv2 retrieval task [98]. The mAP@100 scores on the public and private test sets are reported.

consistently outperforms GV. The largest performance gap appears on the Hard setup. RRT obtains 2.2 (3.7) absolute improvements over GV on \mathcal{ROxf} (\mathcal{RPar}), when using the “v1” descriptors. We posit that, while GV is very effective for sufficiently similar images, it has difficulty handling challenging cases, e.g. large variations in viewpoint. To verify this hypothesis, we try re-ranking more images (e.g. top-200). The performance gap becomes larger indeed. RRT obtains 3.4 (8.4) absolute improvements over GV on \mathcal{ROxf} (\mathcal{RPar}), when using the “v1” descriptors.

We present results on the GLDv2 retrieval task [138] in Table 4.2. Following [16], we report the mAP@100 scores on both the public and private test sets. Compared

with \mathcal{ROxf} and \mathcal{RPar} , the improvement of applying re-ranking on GLDv2 becomes smaller. On the other hand, RRT performs consistently better than the global retrieval baseline and GV. When using “v2-clean” descriptors, the absolute improvements of RRT over global-only (GV) on the private set are 2.8 (2.7).

4.5.3 Ablation on the Number of Local Descriptors

In the DELG model, for each image, a maximum of 1000 local descriptors with a predefined minimum attention score are extracted for geometric verification. In our experiment, we observe that for most of the images, the number of the extracted local descriptors is close to 1000. For example, on the sampled GLDv2 training set, the query and gallery sets of the Revisited Oxford (\mathcal{ROxf}) [98] benchmark, DELG extracts 955/759/987 local descriptors per image on average.

We perform ablation experiments by setting the maximum number of local descriptors used for each image to different values. The features used in the experiments were pretrained on the v2-clean split of GLDv2. For purposes of comparison, we include the results of geometry verification (GV) and the proposed method (RRT). We report the mAP scores on the Revisited Oxford (\mathcal{ROxf}) benchmark in Table 4.3.

Both GV and RRT benefit from using more local descriptors in general. Nevertheless, the performance of RRT saturates at 500 local descriptors. As the local descriptors are extracted from seven image scales, we conjecture that in each image there are descriptors extracted from the same geometry location, thus providing duplicate information. To verify this, we compute the number of *distinct* local descriptors extracted from different grid locations. In particular, we assign each local descriptor $\mathbf{x}_{l,i}$ to a grid location (gu, gv) by $(gu, gv) = (\lfloor u/16 \rfloor, \lfloor v/16 \rfloor)$. Here (u, v) is the coordinate of $\mathbf{x}_{l,i}$ provided by the DELG model, 16 is the stride of the convolutional feature map where $\mathbf{x}_{l,i}$ is extracted from. We then group the descriptors sharing the

# Local Desc.	Medium		Hard	
	GV	RRT	GV	RRT
200	72.1	76.7	48.3	58.9
400	75.2	77.6	53.8	58.6
500	75.7	78.1	53.4	60.2
600	77.4	77.9	55.9	59.6
800	77.9	76.9	56.7	57.4
1000	78.3	78.1	57.9	60.4

Table 4.3: Ablation on the number of local descriptors used per image. We compare the proposed Reranking Transformer (RRT) model to geometric verification (GV) on Revisited Oxford [98]. The mAP scores on the Medium and Hard setups are reported.

same grid location as a *distinct* descriptor. We observe that, the number of *distinct* local descriptors is significantly smaller than the number of all local descriptors per image. For example, on the sampled GLDv2 training set, the query and gallery sets of Revisited Oxford (\mathcal{ROxf}), the numbers of *distinct* local descriptors per image are 585/465/655 on average.

When using the same amount of local descriptors, the proposed method outperforms geometry verification in four of the six experiments on the Medium setup, and consistently outperforms geometry verification in all experiments on the Hard setup.

4.5.4 Comparison with Query Expansion

Query expansion (QE) [25, 26, 127] is another popular reranking technique for image retrieval. Different from GV and RRT, QE aggregates the query image and a number of top-ranked neighbors into a new query. This new query is used to rerank all the gallery images rather than the nearest ones as in GV and RRT. We compare RRT with one of the most widely used query expansion methods: α -weighted query expansion (α QE) proposed in [97]. We use the public implementation of α QE released by [103]. α QE has two hyper-parameters: (1) nQE, the number of top-ranked neighbors to aggregate; (2) α , the exponential weight. In [103], they are set

Method	# Reranked images	Desc. version	Medium				Hard			
			\mathcal{ROxf}	$+\mathcal{R1M}$	\mathcal{RPar}	$+\mathcal{R1M}$	\mathcal{ROxf}	$+\mathcal{R1M}$	\mathcal{RPar}	$+\mathcal{R1M}$
DELG global		v1	69.7	55.0	81.6	59.7	45.1	27.8	63.4	34.1
α QE		v1	72.9	60.7	<u>83.4</u>	<u>63.7</u>	49.4	33.6	66.1	<u>38.1</u>
RRT (ours)	100	v1	75.5	61.2	82.7	60.7	56.4	37.0	68.6	37.5
RRT (ours)	200	v1	77.9	63.5	84.4	62.1	58.8	39.5	71.6	39.5
RRT (ours)	400	v1	79.2	66.2	86.3	64.0	60.5	42.6	74.1	41.6
α QE		v1	72.9	60.7	83.4	63.7	49.4	33.6	66.1	38.1
α QE + RRT (ours)	100	v1	78.7	66.2	85.6	65.4	59.8	42.1	72.8	43.1
DELG global		v2-clean	73.6	60.6	85.7	68.6	51.0	32.7	71.5	44.4
α QE		v2-clean	76.6	66.4	86.7	<u>72.8</u>	54.6	39.5	73.2	<u>51.2</u>
RRT (ours)	100	v2-clean	78.1	67.0	86.7	69.8	60.2	44.1	75.1	49.4
RRT (ours)	200	v2-clean	79.5	68.6	87.8	71.5	62.5	46.3	77.1	52.3
RRT (ours)	400	v2-clean	80.5	70.6	89.1	73.8	64.2	49.5	78.1	55.6
α QE		v2-clean	76.6	66.4	86.7	72.8	54.6	39.5	73.2	51.2
α QE + RRT (ours)	100	v2-clean	80.4	71.7	88.5	74.8	64.0	50.9	77.7	57.1

Table 4.4: Comparison to α QE [97] on Revisited Oxford/Paris [98]. The mAP scores on the Medium ($+\mathcal{R1M}$) and Hard ($+\mathcal{R1M}$) setups are reported. We underline the scores of α QE that RRT cannot match by just reranking the top-100 neighbors. RRT consistently outperforms α QE when reranking the top-400 neighbors for each query. Moreover, combining α QE with RRT significantly outperforms using α QE only, showing that RRT and α QE are complementary to each other.

as $(n\text{QE}, \alpha) = (10, 2.0)$. Our experiment shows that these values do not work out of the box for the DELG descriptors. We tune these parameters on \mathcal{ROxf} over the ranges: $n\text{QE} \in [2, 15]$, $\alpha \in [0.1, 3.0]$, and eventually set them as $(n\text{QE}, \alpha) = (2, 0.3)$.

We present the results on \mathcal{ROxf} and \mathcal{RPar} in Table 4.4. When reranking only the top-100 neighbors, the performance of RRT is superior to α QE on five of the eight settings, except for $\mathcal{RPar}+\text{Medium}$, $\mathcal{RPar}+\mathcal{R1M}+\text{Medium}$, $\mathcal{RPar}+\mathcal{R1M}+\text{Hard}$ (underlined numbers). We believe it is because α QE reranks all the gallery images while RRT reranks only 100 neighbors and keeps the ranks of all the other images unchanged. By reranking more neighbors, e.g. 200, 400, we show that the performance of RRT progressively improves and eventually surpasses α QE by significant margins across all settings. On the Hard setup with the “v1” descriptors, the absolute gains of RRT over α QE on $(\mathcal{ROxf}, \mathcal{ROxf}+\mathcal{R1M}, \mathcal{RPar}, \mathcal{RPar}+\mathcal{R1M})$ are (11.1, 9.0, 8.0, 3.5).

Method	# local desc.	Desc. version	Medium		Hard	
			\mathcal{ROxf}	\mathcal{RPar}	\mathcal{ROxf}	\mathcal{RPar}
DELG global	0	v1	69.7	81.6	45.1	63.4
ASMK global	1000	v1	71.2	80.8	47.1	61.6
ASMK rerank	1000	v1	71.3	82.6	47.5	66.2
RRT (ours)	500	v1	75.5	82.7	56.4	68.6
DELG global	0	v2-clean	73.6	85.7	51.0	71.5
ASMK global	1000	v2-clean	70.4	80.9	45.8	62.0
ASMK rerank	1000	v2-clean	73.1	86.3	49.3	71.9
RRT (ours)	500	v2-clean	78.1	86.7	60.2	75.1

Table 4.5: Comparison to Aggregated Selective Match Kernel (ASMK) on Revisited Oxford/Paris [98]. The mAP scores on the Medium and Hard setups are reported.

We also perform experiments on combining α QE and RRT by reranking the top neighbors produced by α QE. As shown in Table 4.4, combining α QE and RRT considerably improves over using α QE only, with improvements of (10.4, 8.5, 6.7, 5.0) on the Hard setup of (\mathcal{ROxf} , $\mathcal{ROxf}+\mathcal{R1M}$, \mathcal{RPar} , $\mathcal{RPar}+\mathcal{R1M}$) for the “v1” descriptors. We consider query expansion and RRT are thus complementary.

4.5.5 Comparison with Aggregated Selective Match Kernel (ASMK)

Aggregated Selective Match Kernel (ASMK) [125] also leverages local descriptors for image retrieval. The key idea is to create a large visual codebook (i.e. filter banks) by clustering the local descriptors. This visual codebook is used to encode the query and gallery images into global descriptors. The clustering and encoding procedures are typically performed off-line as they’re relatively time-consuming. Previously, ASMK was mainly considered as a global retrieval technique. In this work, we treat ASMK as both a global retrieval baseline and a reranking baseline. We use the public implementation of ASMK released by [124]. Following the common practice proposed in [124], we train a codebook of 65,536 visual words on \mathcal{ROxf} for retrieval exper-

Method	Desc.	SOP			
	Dim	$R@1$	$R@10$	$R@100$	$R@1k$
<i>Global-only retrieval</i>					
Margin [105, 139]	128	76.1	88.4	95.1	98.3
Divide [109]	128	75.9	88.4	94.9	98.1
MIC [104]	128	77.2	89.4	-	95.6
FastAP [15]	128	73.8	88.0	94.9	98.3
XBM [135]	128	80.6	91.6	96.2	98.7
CE [13]	2048	81.1	91.7	96.3	98.8
CO	128	80.7	91.9	96.6	99.0
CO + RRT (frozen)	128	81.8	92.4	96.6	99.0
CO + RRT (finetuned)	128	84.5	93.2	96.6	99.0

Table 4.6: Results of the experiments on jointly optimizing the feature extractor and RRT. The $R@K$ ($K=1, 10, 100, 1000$) scores on SOP [116] are reported.

iments on $\mathcal{R}Par$, and vice-versa. We conduct two experiments: a) ASMK global: using ASMK for global retrieval, as in all the previous literature [124–126]; b) ASMK rerank: using ASMK for image reranking, e.g. reranking the top-100 images from DELG global retrieval.

We present the results on $\mathcal{R}Oxf$ and $\mathcal{R}Par$ in Table 4.5. ASMK, when used as a global retrieval approach, demonstrates comparable or inferior performance to the DELG global retrieval. When used as a reranking approach, ASMK gains further improvement over the DELG global retrieval, showing that they are complementary. The proposed method consistently outperforms ASMK global/rerank in all settings. We posit that compared with the hand-crafted kernel matching paradigm, RRTs learn a more holistic region-wise alignment between the image-pair.

4.5.6 Feature Learning & RRT: Joint Optimization

To further explore the benefit of jointly optimizing feature representations and RRTs, we perform experiments on the Stanford Online Products (SOP) dataset [116]. We study three models: (1) *CO*: A global retrieval model trained with a contrastive

loss [135], following the metric learning protocol. The global descriptor has a dimension of 128, as in most prior work [15, 104, 105, 109, 135]; (2) *CO + RRT (frozen)*: a RRT model trained on top of *CO*. The pretrained *CO* remains frozen and an extra linear projection is used to reduce the dimension of the local descriptors to 128; (3) *CO + RRT (finetune)*: a model with the same architecture as *CO + RRT (frozen)* but the backbone is also finetuned. It is also initialized by *CO + RRT (frozen)*. During testing, we perform global retrieval using the global descriptor from *CO*. The top-100 neighbors for each query are reranked by either *CO + RRT (frozen)* or *CO + RRT (finetune)*. While there is no direct comparison between our method and global-only retrieval work, we present the results of the most recent metric learning approaches [13, 15, 105, 135] to provide an overview of the state-of-the-art performance on SOP.

As shown in Table 4.6, the global *CO* model, which is trained with a contrastive loss using a relatively large batch size, performs surprisingly well. It achieves the same level of accuracy as well-established works on metric learning. This aligns with the recent research on self-supervised learning [23, 48] suggesting that contrastive loss is very effective for feature learning. *CO + RRT (frozen)* further improve the performance, demonstrating the effectiveness of reranking. Note that, as only the top-100 images are reranked, the R@100 and R@1k scores remain unchanged. *CO + RRT (finetuned)* achieves the best reranking performance, with an absolute improvement of 3.8 over the global-only retrieval on R@1. We believe it is because jointly optimizing the backbone and our model leads to better local features that are tailored to the reranking tasks.

Method	Training set	Net	# local desc.	Medium				Hard			
				\mathcal{ROxf}	$+\mathcal{R1M}$	\mathcal{RPar}	$+\mathcal{R1M}$	\mathcal{ROxf}	$+\mathcal{R1M}$	\mathcal{RPar}	$+\mathcal{R1M}$
<i>(A) Global features</i>											
R-MAC [41]	Landmarks	R101	0	60.9	39.3	78.9	54.8	32.4	12.5	59.4	28.0
GeM [97]	SfM-120k	R101	0	64.7	45.2	77.2	52.3	38.5	19.9	56.3	24.7
GeM-AP [103]	SfM-120k	R101	0	67.5	47.5	80.1	52.5	42.8	23.2	60.5	25.1
DELG [16]	GLDv1	R50	0	69.7	55.0	81.6	59.7	45.1	27.8	63.4	34.1
<i>(B) Local feature aggregation</i>											
DELG-ASMK[124]	Landmarks	R50	1000	67.8	53.8	76.9	57.3	43.1	31.2	55.4	26.4
HOW-ASMK[126]	SfM-120k	R50	1000	78.3	63.6	80.1	58.4	55.8	36.8	60.1	30.7
HOW-ASMK[126]	SfM-120k	R50	2000	79.4	65.8	81.6	61.8	56.9	38.9	62.4	33.7
<i>(C) Global features + Re-ranking</i>											
GeM \uparrow +DSM [112]	SfM-120k	R101	1000	65.3	47.6	77.4	52.8	39.2	23.2	56.2	25.0
DELG [16] + GV	GLDv1	R50	1000	75.1	61.1	82.3	60.5	54.2	36.8	64.9	34.8
DELG [16] + RRT (ours)	GLDv1&v2-clean	R50	500	75.5	61.2	82.7	60.7	56.4	37.0	68.6	37.5
DELG [16] + GV	GLDv2-clean	R50	1000	78.3	67.2	85.7	69.6	57.9	43.6	71.0	45.7
DELG [16] + RRT (ours)	GLDv2-clean	R50	500	78.1	67.0	86.7	69.8	60.2	44.1	75.1	49.4

Table 4.7: Comparison to state-of-the-art methods on Revisited Oxford/Paris [98]. mAP scores on the Medium and Hard setups are reported.

4.5.7 Comparison with the State-of-the-Art

In Table 4.7, we compare the proposed method with the state-of-the-art on the \mathcal{ROxf} ($+\mathcal{R1M}$) and \mathcal{RPar} ($+\mathcal{R1M}$) benchmarks. We include the most recent instance recognition/retrieval models in three different groups: (A) Retrieval by global features only; (B) Retrieval by local feature aggregation; (C) Retrieval by combining global features with re-ranking. While our method performs favorably on most of the settings (except for \mathcal{ROxf} , $\mathcal{ROxf}+\mathcal{R1M}$), these results include comparisons to other methods that differ on the training data used, the CNN backbones, and the number of local features. For context we provide as much information about each method regarding these differences.

4.5.8 Limitation

Interpretability. Compared to the homography that explicitly models the alignment of the image-pair, the similarity score predicted by our model is less interpretable. In the future, we’d like to extend the work to learning more visual relation

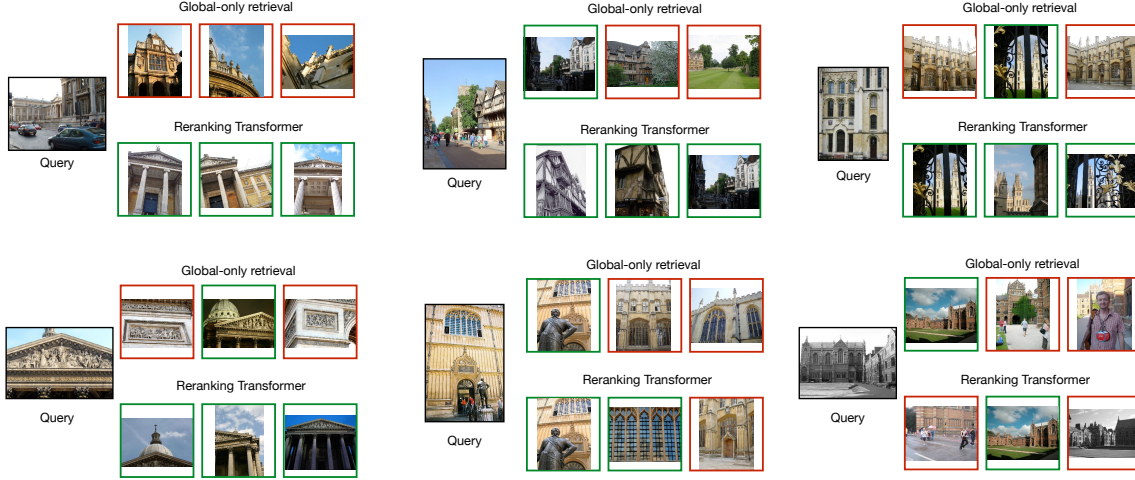


Figure 4.4: Qualitative examples from Revisited Oxford/Paris [98]. For each query, the top-3 neighbors ranked by the global retrieval and reranked by RRT are presented. Correct/incorrect neighbors are marked with green/red borders.

concepts, e.g. homography, dense matching, optical flow, which may lead to more interpretable results.

Domain shift. In the experiment on the pretrained DELG feature descriptors [16], our method is trained on Google Landmarks v2 [138] and tested on Revisited Oxford/Paris [98]. In the experiment on the Stanford Online Products benchmark [116], the training and test sets have no overlapping instance categories. Both experiments demonstrate that the proposed Reranking Transformer can transfer the knowledge across different instance categories to a certain extent. On the other hand, similar to all learning-based approaches, our method might have difficulty in handling large domain shifts. It is also a major challenge for most of the recent approaches as another key component of the image retrieval pipeline, the feature extractor, may also suffer from domain shift. Learning transferable feature representation/matching could be an interesting topic for future research.

4.5.9 Qualitative Examples

In Fig. 4.4, we present qualitative examples when using only global features and when using our full reranking approach on Revisited Oxford/Paris [98]. While global-only retrieval can return highly similar images in general, reranking by global/local descriptors captures a more fine-grained matching between images, leading to better recognition accuracy.

In Fig. 4.5, we provide qualitative examples on Stanford Online Products [116]. Here, we compare the results from the global-only model (*CO*) and the proposed model (*CO + RRT (finetuned)*). In particular, we showcase the examples of rigid objects (e.g. coffee maker, kettle) and deformable objects (e.g. stapler, lamp). The proposed method outperforms the global-only retrieval on challenging cases such as partial-matching (example (A)(C)(D)), articulated objects (example (E)(F)), and irrelevant context (example (B)).

In Fig. 4.6, we provide reranking examples produced by geometry verification and the proposed Reranking Transformer on Revisited Oxford/Paris [98]. It is shown that, compared to geometry verification, the proposed method performs favorably when large viewpoint variations are present. For example, the queries in example (A) and (B) represent the same landmark but exhibit a large viewpoint change. While geometry verification predicts two different sets of top neighbors, our model predicts the same set of top-ranked images for the two queries. Example (E) and (F) also show some failure cases of our model.

4.6 Summary

In this chapter, we introduce *Reranking Transformers* (RRTs) as effective reranking models for instance image retrieval. We show with extensive experiments that the

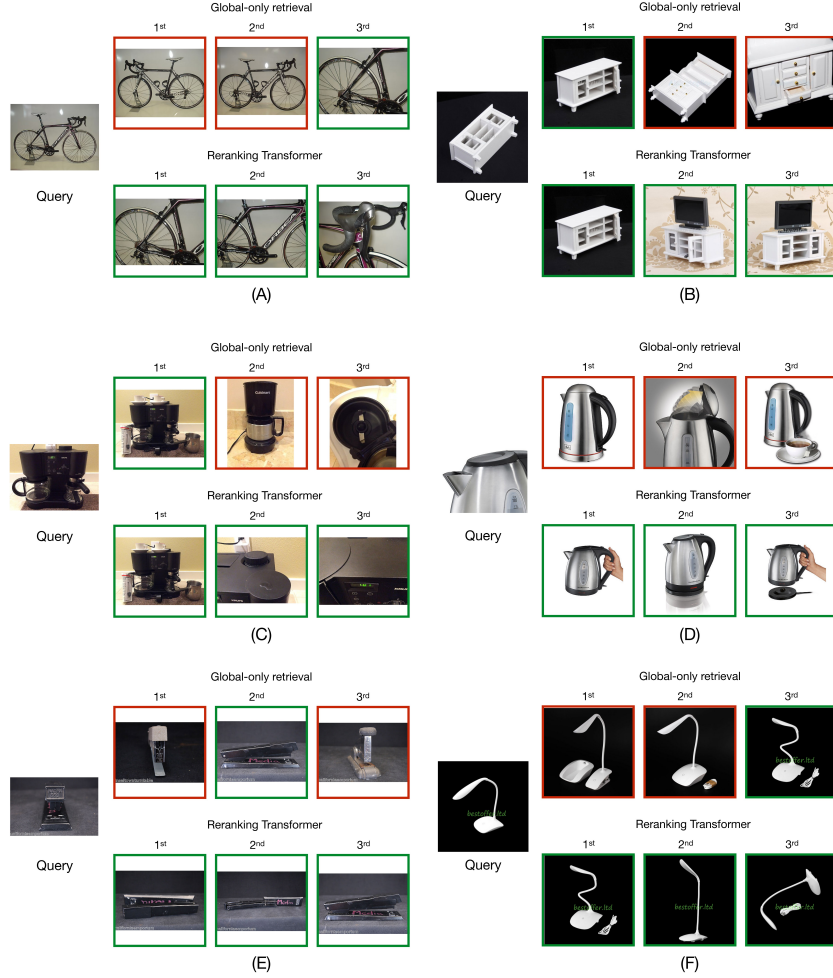


Figure 4.5: Qualitative examples from Stanford Online Products [116]. For each query, the top-3 neighbors predicted by the global-only retrieval and the proposed Reranking Transformer are presented. Correct/incorrect neighbors are marked with green/red borders.

proposed method outperforms prior reranking approaches across a variety of settings. Compared to geometric verification [94] and other local feature based methods [125], RRTs use much fewer descriptors and can be easily parallelized such that reranking requires a single neural network forward pass. We also demonstrate that, unlike previous reranking approaches, RRTs can be optimized jointly with the feature extractor, which leads to improved accuracy.

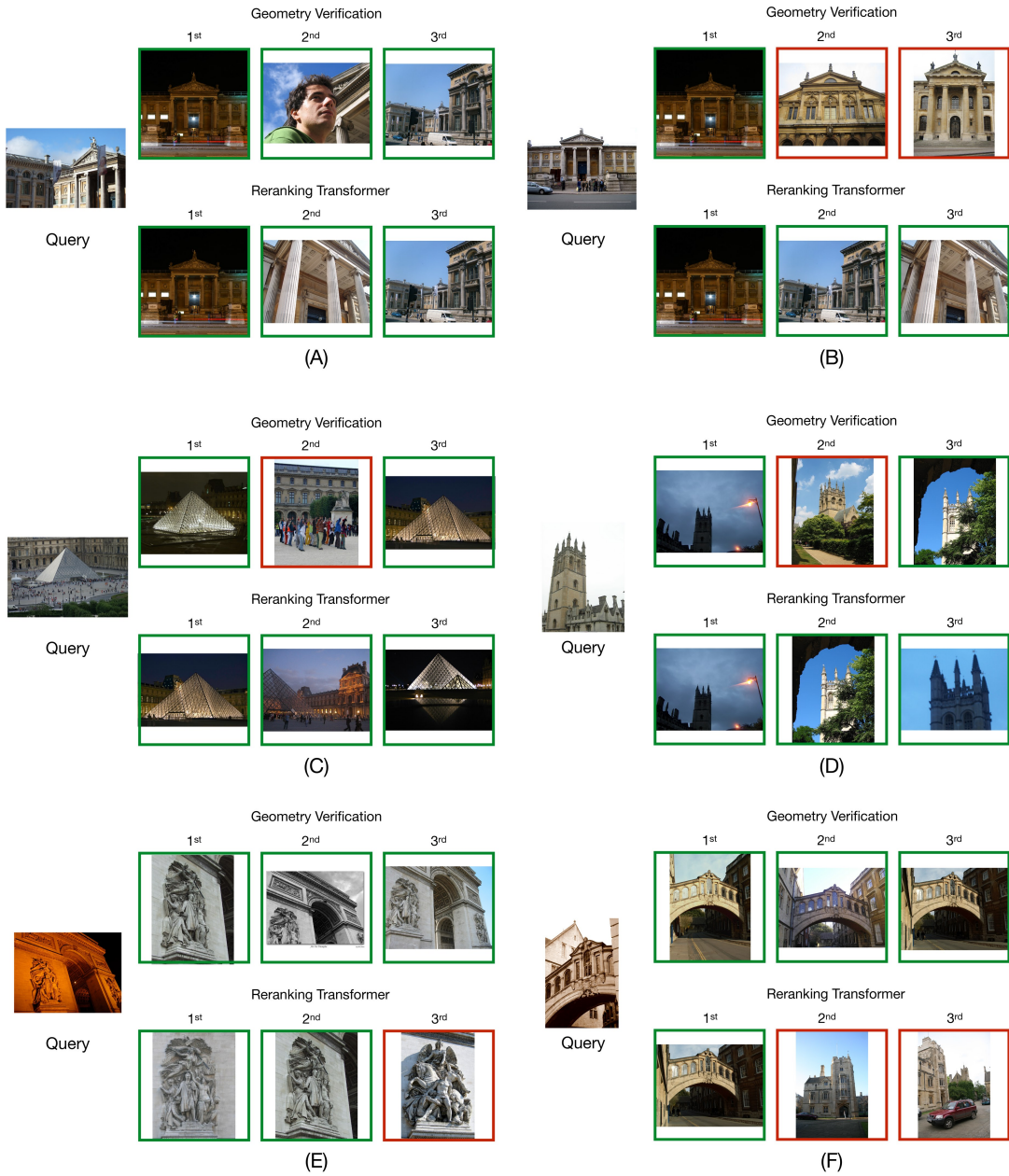


Figure 4.6: Qualitative examples from Revisited Oxford/Paris [98]. For each query, the top-3 neighbors predicted by geometry verification and the proposed Reranking Transformer are presented. Correct/incorrect neighbors are marked with green/red borders.

Chapter 5

Conclusion

This thesis presents my Ph.D. research on learning local representations of images and text for visual synthesis, retrieval, and recognition. Here, we summarize the contributions made in this thesis:

In Chapter 2, we develop a sequence-to-sequence model to generate compositional image representations from visually descriptive language. The proposed framework is general and capable to generate various forms of scenes, e.g. cartoon-like images, object layouts, composite images. Compared to the GAN-based approaches [40, 56, 143] resorting to pixel-wise synthesis, the proposed compositional pipeline demonstrates superior performance in both automatic evaluations and human subject study, while being more interpretable and data-efficient.

In Chapter 3, we present an effective framework for interactive retrieval of specific images of complex scenes. The method explores in-depth and addresses several challenges in multiple round retrievals with natural language queries, e.g. learning instance-aware features with a compositional text representation. We demonstrate that the proposed model performs favorably to the existing approaches on two proposed benchmarks: automatic image retrieval on a simulated scenario that uses region captions as queries, and interactive image retrieval using real queries from human

evaluators.

In Chapter 4, we propose a novel model which learns to predict the visual similarity of an image-pair by analyzing the correlation of both the global and local representations using a transformer architecture. The proposed method is applied to the instance image recognition problem. Compared to domain-specific works that rely on optimizing the matching of large amounts of descriptors, we show that our approach is more general, more flexible, requires much fewer descriptors but achieves higher retrieval accuracy, especially for challenging cases.

We hope that the research in this thesis could help inspire future research in vision and language. Particularly, I would prefer to further explore the research directions as discussed below:

Compositional Representations for Few-shot Learning. As shown in Chapter 2, learning the pixel level compositionality of visual data requires access to large amounts of labeled data. This limits the deployment of deep learning models to the domains where annotations are difficult to collect. The research conducted in Chapter 2 suggests a new direction: decouple the learning of the hierarchical concepts (e.g. image, event, segment, pixel) into different stages using compositional representations. I envision that this line of research could be beneficial for few-shot learning. Potential applications include (1) the analysis of multimodal documents which contains diagrams, text, tables, listitems, etc; (2) learning visual relationship (e.g. human-object interaction) presented in images and videos that involves modeling the intra-object representations and inter-object correlation.

Learning Generalizable Visual Representations from Language. Depending on the specific tasks, existing systems typically train different model instances on different datasets. Consequently, the learned representations may be dramatically

different even though the models may have similar architectures, or the datasets may share common concepts, e.g. both ImageNet [30] and COCO [74] contain images of “cat” and “dog”. I believe this will result in redundant efforts on data annotations and difficulties in transferring the learned knowledge for different tasks. To address this problem, I would like to explore learning generalizable visual representations using common supervised signals from language. On the one hand, it could simplify the data labeling process to a certain extent as annotations in the form of language are relatively easy to collect. On the other hand, it could also learn common visual representations for diverse supervised problems, improving the generalization of the trained models.

Bibliography

- [1] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. “Building Rome in a day”. In: *2009 IEEE 12th International Conference on Computer Vision*. 2009, pp. 72–79.
- [2] P. Anderson, B. Fernando, M. Johnson, and S. Gould. “Spice: Semantic propositional image caption evaluation”. In: *Eur. Conf. Comput. Vis.* Springer. 2016, pp. 382–398.
- [3] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. “Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2018.
- [4] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. “VQA: Visual Question Answering”. In: *Int. Conf. Comput. Vis.* 2015.
- [5] R. Arandjelovic and A. Zisserman. “Multiple Queries for Large Scale Specific Object Retrieval.” In: *Brit. Mach. Vis. Conf.* 2012, pp. 1–11.
- [6] J. L. Ba, J. R. Kiros, and G. E. Hinton. “Layer Normalization”. In: *arXiv preprint arXiv:1607.06450* (2016).
- [7] A. Babenko and V. S. Lempitsky. “Aggregating Deep Convolutional Features for Image Retrieval”. In: *Int. Conf. Comput. Vis.* 2015.
- [8] A. Babenko, A. Slesarev, A. Chigorin, and V. S. Lempitsky. “Neural Codes for Image Retrieval”. In: *Eur. Conf. Comput. Vis.* 2014.

- [9] S. Bai, X. Bai, Q. Tian, and L. J. Latecki. “Regularized Diffusion Process on Bidirectional Context for Object Retrieval”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 41.5 (2019), pp. 1213–1226. DOI: 10.1109/TPAMI.2018.2828815.
- [10] S. Bai, P. Tang, P. H. Torr, and L. J. Latecki. “Re-Ranking via Metric Fusion for Object Retrieval and Person Re-Identification”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2019.
- [11] S. Banerjee and A. Lavie. “METEOR: An automatic metric for MT evaluation with improved correlation with human judgments”. In: *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization.* 2005, pp. 65–72.
- [12] I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le. “Attention Augmented Convolutional Networks”. In: *Int. Conf. Comput. Vis.* 2019.
- [13] M. Boudiaf, J. Rony, I. M. Ziko, É. Granger, M. Pedersoli, P. Piantanida, and I. B. Ayed. “Metric learning: cross-entropy vs. pairwise losses”. In: *Eur. Conf. Comput. Vis.* 2020.
- [14] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. “Language Models are Few-Shot Learners”. In: *Adv. Neural Inform. Process. Syst.* 2020.
- [15] F. Cakir, K. He, X. Xia, B. Kulis, and S. Sclaroff. “Deep Metric Learning to Rank”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2019.

- [16] B. Cao, A. Araujo, and J. Sim. “Unifying Deep Local and Global Features for Image Search”. In: *Eur. Conf. Comput. Vis.* 2020.
- [17] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. “End-to-End Object Detection with Transformers”. In: *Eur. Conf. Comput. Vis.* 2020.
- [18] X. Carreras and L. Màrquez. “Introduction to the CoNLL-2005 shared task: Semantic role labeling”. In: *Proceedings of the ninth conference on computational natural language learning*. Association for Computational Linguistics. 2005, pp. 152–164.
- [19] P. Cascante-Bonilla, X. Yin, V. Ordonez, and S. Feng. “Chat-crowd: A Dialog-based Platform for Visual Layout Composition”. In: *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*. 2019.
- [20] N.-S. Chang and K.-S. Fu. “A Relational Database System for Images”. In: *Pictorial Information Systems*. 1980.
- [21] N.-S. Chang and K.-S. Fu. “Query-by-Pictorial-Example”. In: *IEEE Trans. Softw. Eng.* 6.6 (Nov. 1980), pp. 519–524.
- [22] Q. Chen and V. Koltun. “Photographic Image Synthesis with Cascaded Refinement Networks”. In: *Int. Conf. Comput. Vis.* 2017.
- [23] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton. “A Simple Framework for Contrastive Learning of Visual Representations”. In: *Int. Conf. Mach. Learn.* 2020.
- [24] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. “Learning Phrase Representations using RNN Encoder–

- Decoder for Statistical Machine Translation”. In: *Empirical Methods in Natural Language Processing (EMNLP)*. 2014.
- [25] O. Chum, A. Mikulík, M. Perdoch, and J. Matas. “Total recall II: Query expansion revisited”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2011, pp. 889–896.
 - [26] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. “Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval”. In: *Int. Conf. Comput. Vis.* 2007.
 - [27] N. Dalal and B. Triggs. “Histograms of oriented gradients for human detection”. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*. Vol. 1. 2005, 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177.
 - [28] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. Moura, D. Parikh, and D. Batra. “Visual Dialog”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2017.
 - [29] A. Das, S. Kottur, J. M. F. Moura, S. Lee, and D. Batra. “Learning Cooperative Visual Dialog Agents With Deep Reinforcement Learning”. In: *Int. Conf. Comput. Vis.* 2017.
 - [30] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. “ImageNet: A Large-Scale Hierarchical Image Database”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2009.
 - [31] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*. Minneapolis, Minnesota: Association for Computational

- Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: <https://www.aclweb.org/anthology/N19-1423>.
- [32] M. Donoser and H. Bischof. “Diffusion Processes for Retrieval Revisited”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2013, pp. 1320–1327. DOI: 10.1109/CVPR.2013.174.
 - [33] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *arXiv preprint arXiv:2010.11929* (2020).
 - [34] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler. “D2-Net: A Trainable CNN for Joint Detection and Description of Local Features”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2019.
 - [35] F. Faghri, D. J. Fleet, J. R. Kiros, and S. Fidler. “VSE++: Improving Visual-Semantic Embeddings with Hard Negatives”. In: *Brit. Mach. Vis. Conf.* 2018.
 - [36] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. “Object Detection with Discriminatively Trained Part-Based Models”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 32.9 (Sept. 2010), 1627–1645. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2009.167. URL: <https://doi.org/10.1109/TPAMI.2009.167>.
 - [37] M. A. Fischler and R. C. Bolles. “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”. In: *Commun. ACM* 24.6 (1981), 381–395.
 - [38] J. P. Florian Schroff Dmitry Kalenichenko. “FaceNet: A Unified Embedding for Face Recognition and Clustering”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2015.

- [39] D. F. Fouhey and C. L. Zitnick. “Predicting object dynamics in scenes”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2014, pp. 2019–2026.
- [40] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. “Generative adversarial nets”. In: *Adv. Neural Inform. Process. Syst.* 2014, pp. 2672–2680.
- [41] A. Gordo, J. Almazán, J. Revaud, and D. Larlus. “End-to-end Learning of Deep Visual Representations for Image Retrieval”. In: *Int. J. Comput. Vis.* (2017).
- [42] X. Guo, H. Wu, Y. Cheng, S. Rennie, G. Tesauro, and R. Feris. “Dialog-based Interactive Image Retrieval”. In: *Adv. Neural Inform. Process. Syst.* 2018, pp. 676–686.
- [43] T. Gupta, D. Schwenk, A. Farhadi, D. Hoiem, and A. Kembhavi. “Imagine This! Scripts to Compositions to Videos”. In: *Eur. Conf. Comput. Vis.* 2018.
- [44] T. Gupta, A. Schwing, and D. Hoiem. “ViCo: Word Embeddings From Visual Co-Occurrences”. In: *Int. Conf. Comput. Vis.* 2019.
- [45] T. Gupta, K. J. Shih, S. Singh, and D. Hoiem. “Aligned Image-Word Representations Improve Inductive Transfer Across Vision-Language Tasks”. In: *Int. Conf. Comput. Vis.* 2017.
- [46] X. Han, Z. Wu, Y.-G. Jiang, and L. S. Davis. “Learning Fashion Compatibility with Bidirectional LSTMs”. In: *ACM Multimedia.* 2017.
- [47] K. He, X. Zhang, S. Ren, and J. Sun. “Deep Residual Learning for Image Recognition”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2016, pp. 770–778.
- [48] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. “Momentum Contrast for Unsupervised Visual Representation Learning”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2020.

- [49] K. He, X. Zhang, S. Ren, and J. Sun. “Deep Residual Learning for Image Recognition”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2016.
- [50] J. U. Holger Caesar and V. Ferrari. “COCO-Stuff: Thing and Stuff Classes in Context”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2018.
- [51] S. Hong, D. Yang, J. Choi, and H. Lee. “Inferring Semantic Layout for Hierarchical Text-to-Image Synthesis”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2018.
- [52] A. Iscen, G. Tolias, Y. Avrithis, T. Furon, and O. Chum. “Efficient Diffusion on Region Manifolds: Recovering Small Objects with Compact CNN Representations”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2017.
- [53] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. “Image-to-Image Translation with Conditional Adversarial Networks”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2017.
- [54] H. Jiang, I. Misra, M. Rohrbach, E. Learned-Miller, and X. Chen. “In Defense of Grid Features for Visual Question Answering”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2020.
- [55] Y. Jin, D. Mishkin, A. Mishchuk, J. Matas, P. Fua, K. Yi, and E. Trulls. “Image Matching across Wide Baselines: From Paper to Practice”. In: *Int. J. Comput. Vis.* 2020.
- [56] J. Johnson, A. Gupta, and L. Fei-Fei. “Image Generation from Scene Graphs”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2018.
- [57] A. Karpathy, A. Joulin, and L. Fei-Fei. “Deep Fragment Embeddings for Bidirectional Image Sentence Mapping”. In: *Adv. Neural Inform. Process. Syst.* 2014, pp. 1889–1897.

- [58] S. Kazemzadeh, V. Ordonez, M. Matten, and T. L. Berg. “Referitgame: Referring to objects in photographs of natural scenes”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 787–798.
- [59] C. Kiddon, L. S. Zettlemoyer, and Y. Choi. “Globally Coherent Text Generation with Neural Checklist Models”. In: *Empirical Methods in Natural Language Processing (EMNLP)*. 2016.
- [60] J.-H. Kim, D. Parikh, D. Batra, B.-T. Zhang, and Y. Tian. “CoDraw: Visual Dialog for Collaborative Drawing”. In: *arXiv preprint arXiv:1712.05558* (2017).
- [61] D. P. Kingma and J. Ba. “Adam: A Method for Stochastic Optimization”. In: *Int. Conf. Learn. Represent.* (2015).
- [62] R. Kiros, R. Salakhutdinov, and R. S. Zemel. “Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models”. In: *arXiv preprint arXiv:1411.2539* (2014).
- [63] A. Kovashka and K. Grauman. “Attribute Pivots for Guiding Relevance Feedback in Image Search”. In: *Int. Conf. Comput. Vis.* 2013.
- [64] A. Kovashka, D. Parikh, and K. Grauman. “Whittlesearch: Interactive image search with relative attribute feedback”. In: *Int. J. Comput. Vis.* 115.2 (2015), pp. 185–210.
- [65] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei. “Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations”. In: *Int. J. Comput. Vis.* 123.1 (May 2017), 32–73. ISSN: 0920-

5691. DOI: 10.1007/s11263-016-0981-7. URL: <https://doi.org/10.1007/s11263-016-0981-7>.
- [66] A. Krizhevsky, I. Sutskever, and G. E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Adv. Neural Inform. Process. Syst.* Vol. 25. 2012, pp. 1097–1105. URL: <https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>.
 - [67] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. “Baby talk: Understanding and generating simple image descriptions”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2011, pp. 1601–1608. DOI: 10.1109/CVPR.2011.5995466.
 - [68] S. Lazebnik, C. Schmid, and J. Ponce. “Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* Vol. 2. 2006, pp. 2169–2178. DOI: 10.1109/CVPR.2006.68.
 - [69] K.-H. Lee, X. Chen, G. Hua, H. Hu, and X. He. “Stacked Cross Attention for Image-Text Matching”. In: *Eur. Conf. Comput. Vis.* 2018.
 - [70] T. Leung and J. Malik. “Representing and Recognizing the Visual Appearance of Materials Using Three-Dimensional Textons”. In: *Int. J. Comput. Vis.* 43.1 (June 2001), 29–44. ISSN: 0920-5691. DOI: 10.1023/A:1011126920638. URL: <https://doi.org/10.1023/A:1011126920638>.
 - [71] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang. “VisualBERT: A Simple and Performant Baseline for Vision and Language”. In: *arXiv preprint arXiv:1908.03557* (2019).

- [72] L. Liao, Y. Ma, X. He, R. Hong, and T.-S. Chua. “Knowledge-aware Multimodal Dialogue Systems”. In: *ACM Int. Conf. Multimedia*. 2018, pp. 801–809.
- [73] C.-Y. Lin. “Rouge: A package for automatic evaluation of summaries”. In: *Text Summarization Branches Out* (2004).
- [74] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. “Microsoft COCO: Common Objects in Context”. In: *Eur. Conf. Comput. Vis.* (2014).
- [75] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. “RoBERTa: A Robustly Optimized BERT Pretraining Approach”. In: *1907.11692* (2019).
- [76] I. Loshchilov and F. Hutter. “Decoupled Weight Decay Regularization”. In: *Int. Conf. Learn. Represent.* 2019. URL: <https://openreview.net/forum?id=Bkg6RiCqY7>.
- [77] D. G. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. In: *Int. J. Comput. Vis.* (2004).
- [78] J. Lu, D. Batra, D. Parikh, and S. Lee. “ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks”. In: *Adv. Neural Inform. Process. Syst.* Curran Associates, Inc., 2019, pp. 13–23. URL: <http://papers.nips.cc/paper/8297-vilbert-pretraining-task-agnostic-visiolinguistic-representations-for-vision-and-language-tasks.pdf>.
- [79] M. Luong, H. Pham, and C. D. Manning. “Effective Approaches to Attention-based Neural Machine Translation”. In: *Empirical Methods in Natural Language Processing (EMNLP)*. 2015, pp. 1412–1421.

- [80] T. Luong, H. Pham, and C. D. Manning. “Effective Approaches to Attention-based Neural Machine Translation”. In: *Empirical Methods in Natural Language Processing (EMNLP)*. 2015, pp. 1412–1421.
- [81] J Matas, O Chum, M Urban, and T Pajdla. “Robust wide-baseline stereo from maximally stable extremal regions”. In: *Brit. Mach. Vis. Conf.* 2002.
- [82] K. Mikolajczyk and C. Schmid. “A performance evaluation of local descriptors”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 27.10 (2005), pp. 1615–1630.
- [83] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. “Distributed Representations of Words and Phrases and their Compositionality”. In: *Advances in Neural Information Processing Systems*. Vol. 26. 2013, pp. 3111–3119. URL: <https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf>.
- [84] J. Mitchell and M. Lapata. “Vector-based Models of Semantic Composition”. In: *Proceedings of ACL-08: HLT*. Columbus, Ohio: Association for Computational Linguistics, June 2008, pp. 236–244. URL: <https://www.aclweb.org/anthology/P08-1028>.
- [85] M.-E. Nilsback and A. Zisserman. “A Visual Vocabulary for Flower Classification”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2006, pp. 1447–1454.
- [86] D. Nister and H. Stewenius. “Scalable Recognition with a Vocabulary Tree”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* Vol. 2. 2006, pp. 2161–2168.
- [87] Z. Niu, M. Zhou, L. Wang, X. Gao, and G. Hua. “Hierarchical Multimodal LSTM for Dense Visual-Semantic Embedding”. In: *Int. Conf. Comput. Vis.* 2017.

- [88] H. Noh, A. Araujo, J. Sim, and B. Han. “Image Retrieval with Deep Local Features and Attention-based Keypoints”. In: *Int. Conf. Comput. Vis.* 2017.
- [89] A. Oliva and A. Torralba. “Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope”. In: *Int. J. Comput. Vis.* 42 (2004), pp. 145–175.
- [90] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. “BLEU: a method for automatic evaluation of machine translation”. In: *Annual Meeting of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics. 2002, pp. 311–318.
- [91] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, and A. Ku. “Image Transformer”. In: *Int. Conf. Mach. Learn.* 2018.
- [92] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. 2019, pp. 8024–8035. URL: <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.
- [93] J. Pennington, R. Socher, and C. D. Manning. “GloVe: Global Vectors for Word Representation”. In: *EMNLP*. 2014, pp. 1532–1543.
- [94] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. “Object retrieval with large vocabularies and fast spatial matching”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2007, pp. 1–8.

- [95] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik. “Flickr30K Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models”. In: *Int. Conf. Comput. Vis.* 2015, pp. 2641–2649.
- [96] X. Qi, Q. Chen, J. Jia, and V. Koltun. “Semi-parametric Image Synthesis”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2018.
- [97] F. Radenovic, G. Tolias, and O. Chum. “Fine-tuning CNN Image Retrieval with No Human Annotation”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* (2018).
- [98] F. Radenović, A. Iscen, G. Tolias, Y. Avrithis, and O. Chum. “Revisiting Oxford and Paris: Large-Scale Image Retrieval Benchmarking”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2018.
- [99] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens. “Stand-Alone Self-Attention in Vision Models”. In: *Adv. Neural Inform. Process. Syst.* Vol. 32. 2019, pp. 68–80. URL: <https://proceedings.neurips.cc/paper/2019/file/3416a75f4cea9109507cacd8e2f2aefc-Paper.pdf>.
- [100] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. “Generative Adversarial Text to Image Synthesis”. In: *Int. Conf. Mach. Learn.* Vol. 48. Proceedings of Machine Learning Research. PMLR, 2016, pp. 1060–1069. URL: <http://proceedings.mlr.press/v48/reed16.html>.
- [101] S. E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee. “Learning what and where to draw”. In: *Adv. Neural Inform. Process. Syst.* 2016, pp. 217–225.

- [102] S. Ren, K. He, R. Girshick, and J. Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *Adv. Neural Inform. Process. Syst.* 2015.
- [103] J. Revaud, J. Almazan, R. S. Rezende, and C. R. d. Souza. “Learning With Average Precision: Training Image Retrieval With a Listwise Loss”. In: *Int. Conf. Comput. Vis.* 2019.
- [104] K. Roth, B. Brattoli, and B. Ommer. “MIC: Mining Interclass Characteristics for Improved Metric Learning”. In: *Int. Conf. Comput. Vis.* 2019.
- [105] K. Roth, T. Milbich, S. Sinha, P. Gupta, B. Ommer, and J. P. Cohen. “Revisiting Training Strategies and Generalization Performance in Deep Metric Learning”. In: *Int. Conf. Mach. Learn.* 2020.
- [106] Y. Rui, T. S. Huang, and S.-F. Chang. “Image Retrieval: Current Techniques, Promising Directions, and Open Issues”. In: *Journal of Visual Communication and Image Representation* 10.1 (1999), pp. 39–62.
- [107] M. A. Sadeghi and A. Farhadi. “Recognition using visual phrases”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2011, pp. 1745–1752. DOI: 10.1109/CVPR.2011.5995711.
- [108] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. “Improved Techniques for Training GANs”. In: *Adv. Neural Inform. Process. Syst.* 2016.
- [109] A. Sanakoyeu, V. Tschernezki, U. Buchler, and B. Ommer. “Divide and Conquer the Embedding Space for Metric Learning”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2019.

- [110] I. V. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau. “Building End-to-end Dialogue Systems Using Generative Hierarchical Neural Network Models”. In: *AAAI*. 2016, pp. 3776–3783.
- [111] B. Siddiquie, R. S. Feris, and L. S. Davis. “Image Ranking and Retrieval based on Multi-attribute Queries”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2011, pp. 801–808.
- [112] O. Siméoni, Y. Avrithis, and O. Chum. “Local Features and Visual Words Emerge in Activations”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2019.
- [113] J. Sivic and A. Zisserman. “Video Google: A Text Retrieval Approach to Object Matching in Videos”. In: *Int. Conf. Comput. Vis.* 2003, 1470–1477 vol.2.
- [114] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng. “Semantic Compositionality through Recursive Matrix-Vector Spaces”. In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Jeju Island, Korea: Association for Computational Linguistics, July 2012, pp. 1201–1211. URL: <https://www.aclweb.org/anthology/D12-1110>.
- [115] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts. “Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank”. In: *Empirical Methods in Natural Language Processing (EMNLP)*. Seattle, Washington, USA: Association for Computational Linguistics, Oct. 2013, pp. 1631–1642. URL: <https://www.aclweb.org/anthology/D13-1170>.
- [116] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese. “Deep Metric Learning via Lifted Structured Feature Embedding”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2016.

- [117] A. Sordoni, Y. Bengio, H. Vahabi, C. Lioma, J. Grue Simonsen, and J.-Y. Nie. “A Hierarchical Recurrent Encoder-Decoder for Generative Context-Aware Query Suggestion”. In: *ACM International on Conference on Information and Knowledge Management (CIKM)*. 2015, pp. 553–562.
- [118] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus. “End-To-End Memory Networks”. In: *Adv. Neural Inform. Process. Syst.* 2015.
- [119] C. Sun, A. Shrivastava, S. Singh, and A. Gupta. “Revisiting Unreasonable Effectiveness of Data in Deep Learning Era”. In: *Int. Conf. Comput. Vis.* 2017.
- [120] I. Sutskever, O. Vinyals, and Q. V. Le. “Sequence to Sequence Learning with Neural Networks”. In: *Adv. Neural Inform. Process. Syst.* 2014.
- [121] F. Tan, P. Cascante-Bonilla, X. Guo, H. Wu, S. Feng, and V. Ordonez. “Drill-down: Interactive Retrieval of Complex Scenes using Natural Language Queries”. In: *Adv. Neural Inform. Process. Syst.* 2019.
- [122] F. Tan, S. Feng, and V. Ordonez. “Text2Scene: Generating Compositional Scenes from Textual Descriptions”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2019.
- [123] F. Tan, J. Yuan, and V. Ordonez. “Instance-level Image Retrieval using Reranking Transformers”. In: *arXiv preprint arXiv:2103.12236* (2021).
- [124] M. Teichmann, A. Araujo, M. Zhu, and J. Sim. “Detect-to-Retrieve: Efficient Regional Aggregation for Image Search”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2019.
- [125] G. Tolias, Y. Avrithis, and H. Jégou. “Image Search with Selective Match Kernels: Aggregation Across Single and Multiple Images.” In: *Int. J. Comput. Vis.* 116.3 (2016), pp. 247–261.

- [126] G. Tolias, T. Jenicek, and O. Chum. “Learning and aggregating deep local descriptors for instance-level recognition”. In: *Eur. Conf. Comput. Vis.* 2020.
- [127] G. Tolias and H. Jégou. “Visual query expansion with or without geometry: Refining local descriptors by feature aggregation”. In: *Pattern Recognition* 47.10 (2014), pp. 3466–3476.
- [128] G. Tolias, R. Sivic, and H. Jégou. “Particular object retrieval with integral max-pooling of CNN activations”. In: *Int. Conf. Learn. Represent.* 2016.
- [129] M. I. Vasileva, B. A. Plummer, K. Dusad, S. Rajpal, R. Kumar, and D. Forsyth. “Learning Type-Aware Embeddings for Fashion Compatibility”. In: *Eur. Conf. Comput. Vis.* 2018.
- [130] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. “Attention is All you Need”. In: *Adv. Neural Inform. Process. Syst.* Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Curran Associates, Inc., 2017, pp. 5998–6008. URL: <http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf>.
- [131] R. Vedantam, C Lawrence Zitnick, and D. Parikh. “Cider: Consensus-based image description evaluation”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2015, pp. 4566–4575.
- [132] R. Vedantam, X. Lin, T. Batra, C Lawrence Zitnick, and D. Parikh. “Learning common sense through visual abstraction”. In: *Proceedings of the IEEE international conference on computer vision.* 2015, pp. 2542–2550.
- [133] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. “Show and tell: A neural image caption generator”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2015, pp. 3156–3164.

- [134] L. Wang, Y. Li, and S. Lazebnik. “Learning Deep Structure-Preserving Image-Text Embeddings”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2016, pp. 5005–5013.
- [135] X. Wang, H. Zhang, W. Huang, and M. R. Scott. “Cross-Batch Memory for Embedding Learning”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2020.
- [136] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. *Caltech-UCSD Birds 200*. Tech. rep. CNS-TR-2010-001. California Institute of Technology, 2010.
- [137] J. Weston, S. Chopra, and A. Bordes. “Memory Networks”. In: *Int. Conf. Learn. Represent.* 2015.
- [138] T. Weyand, A. Araujo, B. Cao, and J. Sim. “Google Landmarks Dataset v2 – A Large-Scale Benchmark for Instance-Level Recognition and Retrieval”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2020.
- [139] C.-Y. Wu, R. Manmatha, A. J. Smola, and P. Krahenbuhl. “Sampling Matters in Deep Embedding Learning”. In: *Int. Conf. Comput. Vis.* 2017.
- [140] H. Wu, J. Mao, Y. Zhang, Y. Jiang, L. Li, W. Sun, and W.-Y. Ma. “Unified Visual-Semantic Embeddings: Bridging Vision and Language with Structured Meaning Representations”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2019.
- [141] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention”. In: *Int. Conf. Mach. Learn.* Vol. 37. Proceedings of Machine Learning Research. PMLR, 2015, pp. 2048–2057.
- [142] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention”. In: *Int. Conf. Mach. Learn.* Vol. 37. 2015, pp. 2048–2057.

- [143] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He. “AttnGAN: Fine-Grained Text to Image Generation With Attentional Generative Adversarial Networks”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2018.
- [144] M. Yatskar, V. Ordonez, and A. Farhadi. “Stating the obvious: Extracting visual common sense knowledge”. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2016, pp. 193–198.
- [145] X. Yin and V. Ordonez. “Obj2Text: Generating Visually Descriptive Language from Object Layouts”. In: *Empirical Methods in Natural Language Processing (EMNLP)*. 2017.
- [146] A. L. Yuille. “Deformable Templates for Face Recognition”. In: *Journal of Cognitive Neuroscience* 3.1 (1991), pp. 59–70.
- [147] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. “Understanding deep learning requires rethinking generalization”. In: *Int. Conf. Learn. Represent.* 2017.
- [148] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. Metaxas. “Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks”. In: *Int. Conf. Comput. Vis.* 2017, pp. 5907–5915.
- [149] S. Zhang, M. Yang, T. Cour, K. Yu, and D. N. Metaxas. “Query Specific Rank Fusion for Image Retrieval”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 37.4 (2015), pp. 803–815. DOI: 10.1109/TPAMI.2014.2346201.
- [150] Z. Zhang, Y. Xie, and L. Yang. “Photographic Text-to-Image Synthesis with a Hierarchically-nested Adversarial Network”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2018.

- [151] H. Zhao, J. Jia, and V. Koltun. “Exploring Self-attention for Image Recognition”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2020.
- [152] X. Zhu and E. Grefenstette. “Deep Learning for Semantic Composition”. In: *ACL tutorial*. 2017.
- [153] C. L. Zitnick and D. Parikh. “Bringing Semantics into Focus Using Visual Abstraction”. In: *IEEE Conf. Comput. Vis. Pattern Recog.* 2013.
- [154] C. L. Zitnick, D. Parikh, and L. Vanderwende. “Learning the Visual Interpretation of Sentences”. In: *Int. Conf. Comput. Vis.* 2013.
- [155] Z. Zuo, B. Shuai, G. Wang, X. Liu, X. Wang, B. Wang, and Y. Chen. “Convolutional recurrent neural networks: Learning spatial dependencies for image representation”. In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2015, pp. 18–26.