



Multimodal Sparse Representation Learning and Cross-Modal Synthesis

Citation

Cha, Miriam. 2019. Multimodal Sparse Representation Learning and Cross-Modal Synthesis. Doctoral dissertation, Harvard University, Graduate School of Arts & Sciences.

Link

<http://nrs.harvard.edu/urn-3:HUL.InstRepos:42029738>

Terms of use

This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Other Posted Material (LAA), as set forth at

<https://harvardwiki.atlassian.net/wiki/external/NGY5NDE4ZjgzNTc5NDQzMGIzZWZhMGFIOWI2M2EwYTg>

Accessibility

<https://accessibility.huit.harvard.edu/digital-accessibility-policy>

Share Your Story

The Harvard community has made this article openly available.
Please share how this access benefits you. [Submit a story](#)

Multimodal Sparse Representation Learning and Cross-modal Synthesis

A dissertation presented

by

Miriam Cha

to

The Graduate School of Arts and Sciences

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in the subject of

Computer Science

Harvard University

Cambridge, Massachusetts

May 2019

© 2019 Miriam Cha
All rights reserved

Advisor
H. T. Kung

Author
Miriam Cha

Multimodal Sparse Representation Learning and Cross-modal Synthesis

Abstract

Humans have a natural ability to process and relate concurrent sensations in different sensory modalities such as vision, hearing, smell, and taste. In order for artificial intelligence to be more human-like in their capabilities, it needs to be able to interpret and translate multimodal information. However, multimodal data are heterogenous, and the relationship between modalities is often complex. For example, there exist a number of correct ways to draw an image given a text description. Similarly, many text descriptions can be valid for an image. Summarizing and translating multimodal data is therefore challenging.

In this thesis, I describe multimodal sparse coding schemes that can learn to represent multiple data modalities jointly. A key premise behind joint sparse coding is the representational power that captures complementary information while reducing statistical redundancy. As a result, my schemes can improve the performance of classification and retrieval tasks involving co-occurring data modalities.

Building on the deep learning framework, I also present probabilistic generative models that produce new data conditioned on an input from another data modality. Specifically, I develop text-to-image synthesis models based on generative adversarial networks (GAN). To improve the visual realism and the diversity of generated images, I propose additional objective functions and a new GAN architecture. Furthermore, I propose a novel sampling strategy for training data that promotes output diversity under adversarial setting.

Contents

Title Page	i
Abstract	iii
Acknowledgments	xi
Dedication	xiii
1 Introduction	1
1.1 Motivation	1
1.2 Related Work	3
1.2.1 Multimodal phenomenology	3
1.2.2 Cross-modal synthesis	4
1.3 Proposed Approach	5
1.4 Contributions	7
1.5 Organization	7
1.6 Related Publications	8
2 Preliminaries	9
2.1 Sparse Coding	9
2.1.1 Clustering view	9
2.1.2 Probabilistic interpretation	10
2.2 Generative Adversarial Models	12
2.2.1 Generative adversarial networks (GAN)	12
2.2.2 Conditional generative adversarial networks (CGAN)	13
2.2.3 Deep convolutional generative adversarial networks (DCGAN)	13
3 Learning Multimodal Sparse Representation	14
3.1 Introduction	14
3.2 Multimodal Feature Learning via Sparse Coding	15
3.2.1 Parallel unimodal sparse coding	16
3.2.2 Joint multimodal sparse coding	17

3.2.3	Hierarchical multimodal sparse coding	18
3.3	Experiments	20
3.3.1	Image denoising	20
3.3.2	Multimedia event detection using audio and video	21
3.3.3	Wikipedia category classification using images and text	24
3.3.4	Twitter sentiment classification using images and text	26
3.4	Summary	28
4	Applying Conditional Multimodality to Image Generation	29
4.1	Introduction	29
4.2	Method	31
4.2.1	Architecture	31
4.2.2	Adversarial loss	33
4.2.3	Perceptual loss	33
4.3	Experimental Results	35
4.3.1	Qualitative evaluation	36
4.3.2	Quantitative evaluation	37
4.4	Summary	38
5	Improving the Diversity of Generated Images with Semantic Relevance	40
5.1	Introduction	40
5.2	Comparison	42
5.2.1	Conditional GAN (CGAN)	42
5.2.2	Matching-aware Manifold-interpolated GAN (GAN-INT-CLS)	43
5.2.3	Auxiliary Classifier GAN (AC-GAN)	44
5.2.4	Text Conditioned Auxiliary Classifier GAN (TAC-GAN)	45
5.3	Text-conditioned Semantic Classifier GAN	46
5.3.1	Overview	46
5.3.2	Training objectives	46
5.3.3	Negative sampling	47
5.4	Experiments	50
5.4.1	Implementation details	50
5.4.2	Evaluation metrics	51
5.4.3	Quantitative analysis	52
5.4.4	Qualitative analysis	53
5.5	Conclusion	55
6	Adversarial Text-to-Image Synthesis by Semantic Sampling	56
6.1	Introduction	56
6.2	Method	58
6.2.1	Overview	58
6.2.2	Intuition: sampling under aliasing	60

6.2.3	Theory	60
6.2.4	Proposed semantic sampling	62
6.2.5	Training objectives	65
6.3	Experiments	66
6.3.1	Settings	66
6.3.2	Sampling scheme comparison	67
6.3.3	Extension to other models	68
6.4	Summary	69
7	Conclusion	71
	Bibliography	73

List of Figures

2.1	Sparse coding $SC(\mathbf{x}, \mathbf{D})$ encodes input data $\mathbf{x}$ as a sparse linear combination $\mathbf{y}$ of basis vectors from a dictionary $\mathbf{D}$ through the ℓ_0 - or ℓ_1 -regularized optimizations. The recovery path reconstructs the original data by the matrix-vector multiplication $\hat{\mathbf{x}} \approx \mathbf{D}\mathbf{y}$	10
3.1	Unimodal sparse coding and feature union	16
3.2	Multimodal sparse coding and feature union	16
3.3	Deep learning approaches for multimodal sparse coding	18
3.4	Example of the denoising results with $\sigma = 0.01$. From top to bottom, we visualize clean original image, noisy image, denoised images using DAE and joint sparse coding.	21
3.5	TRECVID audio-video data processing	22
3.6	Category classification comparison of joint sparse coding, cross-modal by text, and multimodal feature union on Wikipedia dataset	26
4.1	Four synthetic bird images generated for the given text, displaying different levels of contextual and perceptual relevance. The upper right image is most desirable, as it has higher relevance on both accounts.	30
4.2	Architecture of the generator and discriminator networks of our conditional GAN synthesis model.	32
4.3	Generated bird images by conditioning on text from unseen test categories using the baseline GAN-INT-CLS [20] and the baseline with three different perceptual loss functions: pixel (GAN-INT-CLS-Pixel), activation (GAN-INT-CLS-VGG), and texture (GAN-INT-CLS-Gram) reconstruction losses.	36
4.4	Generated flower images by conditioning on text from unseen test categories using the baseline GAN-INT-CLS [20] and the baseline with three different perceptual loss functions: pixel (GAN-INT-CLS-Pixel), activation (GAN-INT-CLS-VGG), and texture (GAN-INT-CLS-Gram) reconstruction losses.	37

5.1	An illustration of sample diversity in text-to-image synthesis. Top: poor diversity (generated images are nearly identical suffering from mode collapse); middle: modest diversity (generated images belong to a single class); bottom: good diversity (generated images are not limited to classes)	41
5.2	Architectural comparison of (a) CGAN [19], (b) matching-aware manifold-interpolated GAN [20], (c) auxiliary classifier GAN [34], (d) text conditioned auxiliary classifier GAN [35], and (e) our method.	43
5.3	Overall pipeline of Text-SeGAN. From training data, we first form mini-batches of N triplet examples $\{\mathbf{x}_p^{(i)}, \varphi(\mathbf{t}_p^{(i)}), \mathbf{x}_n^{(i)}\}_{i=1}^N$. The formed triplets are used to train the generator and discriminator networks.	44
5.4	Sampling negative images. Different shapes indicate different classes. Images are placed in the semantic space (reduced to 2D for illustrative purposes) where the Euclidean distance is used to indicate separation between points. The selection methods are (a) random negatives, (b) easy negatives, (c) hard negatives, (d) semi-easy negatives, and (e) semi-hard negatives.	47
5.5	Comparison of the class-wise MS-SSIM scores of the samples from the training data and the generated samples of Text-SeGAN using easy-to-hard negative sampling.	53
5.6	Flower images generated by GAN-INT-CLS, TAC-GAN, and Text-SeGAN (with easy-to-hard negatives).	54
6.1	Sampled positive and negative images using (a) baseline and (b) our sampling scheme given an anchor text from Oxford-102.	57
6.2	Overview of the proposed algorithm. Given a text embedding $\varphi(\mathbf{t}_a)$ for an anchor text $\mathbf{t}_a$, our algorithm first samples positive $\mathbf{x}_p$ and negative $\mathbf{x}_n$ images based on their semantic similarities to the encoded anchor text and forms triplets (Section 6.2.4). Then we train a text-to-image synthesis GAN using these triplets. Conditioning both generator G and discriminator D on $\varphi(\mathbf{t}_a)$, G generates an image, and D discriminates whether or not its input $(\mathbf{x}, \varphi(\mathbf{t}))$ contains a real image $\mathbf{x}$, and $\mathbf{x}$ and $\varphi(\mathbf{t}_a)$ semantically match (Section 6.2.5).	59
6.3	Training dataset only captures many-to-one mappings in the text-to-image direction (solid lines). In fact, image-text relationship is many-to-many as indicated with solid and dashed lines.	62
6.4	Sample images are placed according to cosine similarity between their corresponding text and an anchor text. When cosine similarity is high, semantic match between a sampled image and an anchor text is high.	64
6.5	Randomly selected positive images for an anchor text based on (a) classful sampling with $\alpha = 1$ (baseline), (b) classful sampling with $\alpha = 0.9$, and (c) classless sampling with $\alpha = 0.9$	68

List of Figures

6.6	Comparison of the class-wise MS-SSIM scores of the samples from the training data and the generated samples of GAN-INT-CLS using classless semantic sampling.	69
6.7	Generated flower images using baseline sampling and our classless semantic sampling on GAN-INT-CLS and HDGAN	70

List of Tables

3.1	Peak signal-to-noise ratio results (in decibels) on image denoising using denoising autoencoder and joint sparse coding with a range of standard deviation σ	20
3.2	Mean accuracy and mAP performance of sparse coding schemes	23
3.3	Mean accuracy and mAP performance for GMM and RBM on 10Ex/100Ex	24
3.4	Classification performance for image-text classification on Wikipedia dataset.	25
3.5	Classification performance for image-text classification on PhotoTweet dataset.	27
3.6	Classification performance comparison between multimodal sparse coding and other existing work on PhotoTweet dataset.	28
4.1	GoogLeNet bird and flower classification accuracies	38
5.1	Inception scores of the generated images using random negative sampling .	51
5.2	Inception scores of the generated images from Text-SeGAN using various negative sampling schemes	52
6.1	Inception scores of flower image generation using various sampling schemes on GAN-INT-CLS.	67
6.2	Inception scores on various models	69

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my advisor, HT Kung, for his enthusiasm, inspiration, and guidance throughout my 5 years of PhD study. HT was always open for new ideas and available when I needed his help regardless of time or day. Our status update meetings gave me the motivation to think deeper and make the best effort possible. His sharp intuition and fearless passion will never be forgotten.

I would also like to thank David Parkes and Yue Lu for being my committee members from the qualifying exam to the final defense. I have benefited from their constructive feedback and kind support on my research. Barbara Grosz also gave practical guidances, particularly when I was preparing to graduate. I thank her for her advice.

I am sincerely grateful to Youngjune Gwon, who is my research partner, co-worker, and brother-like figure. He was always eager to help, be it a research issue or a worldly matter. Without his breath of knowledge in the area of machine learning, finishing my thesis would have remained a distant dream.

Marios Savvides, my master research advisor at Carnegie Mellon University, opened the door of pattern recognition for me at the very beginning and encouraged me to further my graduate studies.

I owe a special debt of gratitude to MIT Lincoln Laboratory whose financial support through Lincoln Scholarship provided me invaluable resources and support to my study at Harvard. I would like to say thanks to my previous mentors at Lincoln, Rhonda Phillips, Patrick Wolfe, and Christ Richmond for their support and guidance that led me to pursue PhD program in the first place. I would also like to thank my Lincoln Scholarship mentor, Steven Smith, for thoughtful feedback.

I am also thankful for the National Science Foundation (NSF) Graduate Research Fel-

Acknowledgments

lowship that funded me; it provided me an even greater sense of research freedom.

My thanks also go to the members of the Kung Lab for many great collaborations and discussions: Steve Tarsa, Kevin Chen, Michael Crouse, Surat Teerapittayanon, Brad McDanel, Philippe Tillet, Marcus Comiter, Sai Zhang, James Yang, and Xin Dong.

Most of all, I would like to thank my parents who always trusted my decision, showed unconditional love, and prayed for me. Mom and Dad, this thesis would not have been possible without your love and support. To them, this thesis is dedicated.

To my parents for raising me to believe that anything is possible

Chapter 1

Introduction

1.1 Motivation

Human perception benefits from integrating multiple sensory inputs. Processing different sensory modalities (e.g., vision, auditory, olfaction) offers a possibility of capturing the correlation between modalities, which can strengthen our understanding of phenomena. When we observe an event such as a car accident, we tend to distill a combined effect of multiple sensations surrounding the event rather than perceiving its visible and audible attributes separately.

Multimodal machine learning aims to develop representations and algorithms that will allow a computer to interpret and translate multimodal information. Co-occurring data modalities are a creative source for human intelligence. When we hear the sound of an animal, our mind can immediately imagine its visuals, or vice versa. If reading a story about a colorful garden, an image of beautifully blossoming flowers could come into one's imagination.

Multimodal representation learning has a significant impact on many areas—such as navigation, recognition of events, people, and objects, health care, and surveillance—for better performance and higher reliability. Recently, generative models based on conditional multimodality have found creative data applications in visual editing, animation, and digital design.

This thesis develops on the two most fundamental problems in multimodal machine

learning. First, I address the problem of learning efficient multimodal representations. Secondly, I focus on the problem of creatively synthesizing a modality of data given in another modality. These problems underly many artificial intelligence tasks, including regression, recognition, retrieval, and generative modeling. I will discuss the multimodal challenges in representation learning and synthesis at a high level, provide related background, and summarize my contributions.

Multimodal representation learning aims to capture relationship between co-occurring modalities of data. For a recognition task, jointly captured modalities can be used to resolve an ambiguity surrounding a phenomenon. If one modality is impaired or corrupted, other modalities can help fill in the missing information for robustness. However, it is challenging to learn a succinct representation that exploits the complementarity of multiple modalities due to redundancy and heterogeneity. For example, visual modality is represented as numeric signals while language is often symbolic. Even when heterogeneous modalities are combined as high-dimensional feature representations, some features may be redundant or irrelevant to the task of interest.

The goal of cross-modal synthesis is to generate data from one modality using the other. Cross-modal synthesis is different from cross-modal retrieval, where the closest example from the existing data is found and used as the translated result. Data synthesis is a creative process. In particular, a probabilistic generative model for multimodality samples one data modality from the distribution conditioned on the other. Beyond heterogeneity, a difficulty in cross-modal synthesis is about how to handle many-to-many mapping between modalities. For example, there are many ways to draw an acceptable image for a text description “this blue bird has a pointed black beak.” Conversely, many text descriptions will be valid for an image of a blue bird.

Multimodal representation learning and synthesis problems are especially compelling now due to the widespread of multimodal data on the web—e.g., YouTube, Wikipedia, Twitter. Today, multimodal data and computational resources have significantly increased, yet jointly interpreting and translating between modalities present significant technical challenges. Thus, in the thesis, we aim to develop algorithms that can learn relationship through observing co-occurring multimodal data and probabilistic generative mapping that correlates one modality to another.

1.2 Related Work

This section describes related work on multimodal representation learning and cross-modal synthesis.

1.2.1 Multimodal phenomenology

A classical method for multimodal representation learning is canonical correlation analysis (CCA) [1]. CCA searches for linear projections that maximize the correlation between two data modalities. Kernel CCA (KCCA) [2,3] can be used to find nonlinear projections. Other variants include regularized CCA [4] and multi-view CCA for more than two modalities [5]. The CCA approaches are limited to wide-sense stationarity of each data modality and scale poorly with the training data size [6].

Andrew et al. [7] and Yan & Mikolajczyk [8] propose Deep CCA (DCCA) that models CCA in deep neural nets. DCCA is more flexible and can handle large-scale datasets. However, it is known that DCCA cannot guarantee a good solution to the generalized eigenvalue problem because covariance estimation in each mini-batch is unstable [6]. Ngiam et al. [9] propose a more genuine approach based on deep learning. They have developed a scalable architecture based on stacked autoencoder neural nets to learn joint audio-video representations. Poirson and Idrees [10] have used denoising autoencoders for imagery data annotated by text descriptions. Similarly, Srivastava et al. [11] propose a multimodal extension of deep Boltzmann machines (DBM). In summary, deep learning methods provide better representational capabilities.

More recently, sparse coding has been considered for multimodal feature learning. Jia et al. [12] exploit structured sparsity to learn a shared latent space of multi-view data (e.g., 2D image + depth). Monaci et al. [13] propose a sparse coding-based scheme that learns bimodal structure in audio-visual speech data. Additionally, Zhuang et al. [14] describe a supervised sparse coding scheme for cross-modal retrieval (e.g., text retrieval from an image query). However, there has been little research devoted to evaluating multimodal learning for various multimodal applications. We test our algorithms in three different applications namely, multimedia event detection and retrieval using audio and video data,

category classification on web encyclopedia dataset, sentiment classification on social media dataset, and image denoising.

To learn an efficient multimodal representation, I formulate a joint sparse coding problem motivated by an image super-resolution technique [15–17], which encodes the co-occurrence phenomena between low- and high-resolution image patches using a shared dictionary of basis vectors. Finding the missing pixel values of a high-resolution patch (from its low-resolution counterpart) in general poses an ill-posed regression problem. Sparse regularization alleviates the problem into a joint training that forces multiple modalities of data to share sparse representations that can lead to robust classification and retrieval performances.

1.2.2 Cross-modal synthesis

Cross-modal synthesis resembles a human task of visual imagination upon reading a book. Unlike cross-modal retrieval or translation as in image super-resolution, cross-modal synthesis is a more challenging process beyond regressing the value of missing modality. As a probabilistic generative model, sparse coding is less attractive for creative tasks (e.g., drawing images from text descriptions). Numerical optimization for probabilistic sparse coding is inconvenient as it cannot be easily mapped onto feed-forward processing. Alternately, this thesis explores deep architectures that can learn probabilistic generative mapping from one modality to another. Specifically, my interest is to directly generate image pixels from natural language.

Recently, generative adversarial network (GAN) [18] has been a promising approach for data synthesis. In particular, conditional GAN (CGAN) [19] provides a platform suitable for modeling the phenomenon that many different natural images can be mapped to the same text description. Conditioning gives a means to control the generative process that the original GAN lacks. Reed et al. [20] describe a CGAN architecture extended on deep convolutional generative adversarial network (DCGAN) [21]. They embed a text input onto the semantic word vector space for the use in the GAN training. With both generator and discriminator nets conditioned on text embedding, images corresponding to the descriptive text input can be generated.

StackGAN [22] and HDGAN [23] have improved the resolution of generated images in a two-stage and hierarchically nested CGANs. Notable other approaches have augmented text with synthesized captions [24] and emphasized object subregions prior to image generation [25–28]. ChatPainter [29] adds a dialogue to provide additional information about the scene that contains several objects.

It is known that GANs suffer from mode collapsing—i.e., the GAN-generated images lack diversity and fall largely into a few trends [30–33]. To utilize additional side information, an auxiliary supervised task can be added to the discriminator. Having the auxiliary task is shown to alleviate the mode collapse problem [34]. Notably, Dash et al. [35] make the use of the auxiliary class prediction task to mitigate mode collapsing as observed in Reed et al. [20]. However, as the same text can be used to describe images of different classes, we suspect that feeding scores on class prediction to generator can potentially bound the generator to produce images of limited number of classes.

A bulk of previous work on text-to-image synthesis uses class labels in an auxiliary classifier. A class label can be a helpful guide, but it can limit the generator from exploring the greater subspace of images given the text description. This thesis will present techniques to synthesize images that are diverse, realistic, and relevant to the input text regardless of class label.

1.3 Proposed Approach

The main goals of this thesis are 1) to learn an efficient representation that can interpret multimodal data for a variety of machine learning tasks and 2) to synthesize diverse and realistic images from text descriptions.

The most straightforward multimodal representation would be a vector of stacked unimodal representations. A unimodal representation is learned independently for each data modality. On the contrary, a joint multimodal representation requires the training with all data modalities simultaneously. An immediate benefit is that the joint learning reflects the statistical covariance of co-occurring data modalities, which can provide complementary, discriminative information for a machine learning task. The jointly learned representations

are also more efficient since stacked unimodal representations are statistically redundant. I propose a multimodal extension of sparse coding [36] that yields superior classification performance compared to parallel processing of individual modalities. In addition to joint sparse coding, I describe how to compute cross-modal sparse codes and use them to further improve classification performances.

Beyond multimodal representation learning, I aim to discover the conditional multimodal distribution of co-occurring data that can be used for cross-modal synthesis. Sparse coding can be formulated as a probabilistic generative model. Unlike deep neural nets, however, probabilistic sparse coding cannot be easily mapped onto feedforward processing, therefore less attractive for cross-modal synthesis. By conditioning both generator and discriminator on a text description, conditional GAN can model multimodal phenomena. Despite its overall success, the text-conditional GAN is susceptible to mode collapse, and the generated images often expose visual flaws. I present techniques to synthesize images that are diverse, realistic, and relevant to the input text regardless of class label. The previous work on text-to-image synthesis uses class label either directly in an auxiliary classifier or indirectly in sampling. Although class label can be a helpful side information, it can hinder the generator from modeling the greater subspace of images given the text description. Instead of using class label, I propose a sampling method called *semantic sampling*, which forms matching and non-matching pairs of images and text in a latent space of embedded text inputs. Such classless sampling forms diverse text-image pairs that encourage the generator to synthesize diverse images. In addition, I expand previous work leveraging an auxiliary class prediction in the discriminator. As class prediction can cause adverse effect on the generator training, I employ an auxiliary decoder that is tasked to regress semantic relevance between the two modalities. Beyond sampling, pre-trained latent space can be used in improving perceptual quality. I use high-level feature maps from a pre-trained image classification network to enforce the generated images to reside in the subspace of natural images.

1.4 Contributions

The main contributions are summarized as follows.

- This thesis extends the principle of sparse coding to formulate the joint multimodal representation for concision, cross-modal representation for regressing a missing modality, and union of the cross-modal representations, and applies them to a number of multimodal scenarios, namely multimedia event detection and retrieval on the TRECVID dataset (audio-video), categorical classification on the Wikipedia dataset (image-text), and sentiment classification on PhotoTweet (image-text).
- We show how hierarchical composition of sparse codes are effective for the case of highly nonlinear correlations between modalities.
- We extend state of the art for adversarial text-to-image synthesis by improving visual realism of generated images with augmented objective functions, which optimize on perceptual loss terms that measure pixel, feature activation, and texture differences against a natural image.
- We propose a new auxiliary augmentation technique to stabilize the conditional GAN training and also improve the diversity of the generated samples. Instead of class prediction, we modify the discriminator to regress semantic relevance between images and text.
- We introduce a simple semantic sampling strategy that forms image-text pairs in a high-dimensional space of embedded text inputs. This work is the first sampling method for text-to-image synthesis that does not rely on class labels of images. We demonstrate increased diversity on off-the-shelf text-to-image synthesis models that previously had mode collapse.

1.5 Organization

The rest of this thesis is organized as follows. Chapter 2 provides a background on sparse coding and generative adversarial network (GAN). In Chapter 3, I describe an extension

of the original sparse coding formulation that learns multimodal features, resulting in improved discriminative and retrieval performances. Chapter 4 presents a probabilistic generative mapping from one modality to the other with experimental results in text-to-image synthesis. This chapter focuses on generating realistic images from text descriptions. In Chapter 5, I describe a new architecture that improves the diversity of generated images. In Chapter 6, I formulate the task of generating diverse images from text as a problem of recovery under aliasing and develop semantic sampling schemes that further increase diversity while maintaining semantic correctness to a given text description. Chapter 7 provides a summary and the conclusion.

1.6 Related Publications

A portion of Chapter 3 has appeared in

- Y. Gwon, W. Campbell, K. Brady, D. Sturim, M. Cha, and H. T. Kung, “Multimodal Sparse Coding for Event Detection,” in *NIPS MMML*, 2015.

Chapter 4 is based on

- M. Cha, Y. Gwon, and H. T. Kung, “Adversarial Nets with Perceptual Losses for Text-to-image Synthesis,” in *Proc. of IEEE MLSP*, 2017.

Chapter 5 will appear in

- M. Cha, Y. Gwon, and H. T. Kung, “Adversarial Learning of Semantic Relevance in Text to Image Synthesis,” in *AAAI*, 2019.

Chapter 2

Preliminaries

In this chapter, I briefly review two learning models used in this thesis. The first is sparse coding, a method for discovering good basis vectors automatically using only unlabeled data. I provide both non-probabilistic (clustering view) and probabilistic interpretations. I extend the non-probabilistic sparse coding to learn features over multiple modalities in Chapter 3. The second is a version of generative adversarial networks designed to creatively synthesize data from one modality to the other. This module is used in Chapter 4, 5, and 6.

2.1 Sparse Coding

2.1.1 Clustering view

Sparse coding [36] is an unsupervised method that learns an efficient representation of data using a sparse linear combination of basis vectors. The premise of unsupervised learning is to figure out a useful representational mapping by running through unlabeled examples.

In sparse coding, the goal is to learn an efficient data representation using a small number of basis vectors. Given an input $\mathbf{x} \in \mathbb{R}^N$, sparse coding seeks to solve a representation $\mathbf{y} \in \mathbb{R}^K$ (i.e., sparse feature vector of $\mathbf{x}$) while simultaneously updating the dictionary $\mathbf{D} \in \mathbb{R}^{N \times K}$ in the following optimization:

$$\min_{\mathbf{D}, \mathbf{y}} \|\mathbf{x} - \mathbf{D}\mathbf{y}\|_2^2 + \lambda\psi(\mathbf{y}) \quad \text{s.t. } \|\mathbf{d}_i\|_2 \leq 1, \forall i \quad (2.1)$$

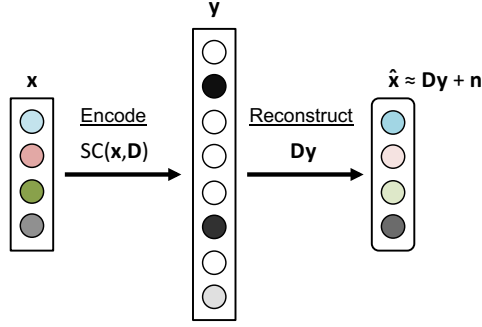


Figure 2.1: Sparse coding $SC(x, D)$ encodes input data x as a sparse linear combination y of basis vectors from a dictionary D through the ℓ_0 - or ℓ_1 -regularized optimizations. The recovery path reconstructs the original data by the matrix-vector multiplication $\hat{x} \approx Dy$.

where d_i is i th basis vector in D , λ is a regularization hyperparameter that induces a sparse solution y for a given x , and $\psi(\cdot)$ is the regularizer. We note that the sparse coding dictionary is an overcomplete matrix, thus $K > N$. This makes the sparse code y higher in dimension than x .

To obtain sparse representations, sparse coding employs ℓ_0 - or ℓ_1 -norm of y as $\psi(\cdot)$. We note that more direct way to control sparsity is to regularize on the ℓ_0 pseudo-norm $\|y\|_0$. However, computing the sparsest ℓ_0 solution in general is known to be intractable. Although greedy ℓ_0 methods such as orthogonal matching pursuit (OMP) [37] are comparable, we impose an ℓ_1 penalty, which is called least absolute shrinkage and selection operator (LASSO) [38] (This approach is a convex relaxation of the ℓ_0 sparse coding). We use the least angle regression (LARS) and the online dictionary learning algorithms by Mairal et al. [39] from SPAMS toolbox [40] to solve the LASSO problem. Figure 2.1 summarizes the data encoding and reconstructive aspect of sparse coding.

2.1.2 Probabilistic interpretation

Olshausen & Field [36] provide a probabilistic generative model for sparse coding

$$x = \sum_{i=1}^k y_i d_i + \epsilon \quad (2.2)$$

where ϵ is an additive noise term.

The goal of probabilistic sparse coding is to find a set of basis vectors $\mathbf{D}$ such that the likelihood $p(\mathbf{x}|\mathbf{D})$ is roughly equivalent to the empirical input distribution $p^*(\mathbf{x})$. Such task can be achieved by minimizing the KL divergence between $p^*(\mathbf{x})$ and $p(\mathbf{x}|\mathbf{D})$ as

$$\min D_{KL}(p^*(\mathbf{x})||p(\mathbf{x}|\mathbf{D})) = \min \int p^*(\mathbf{x}) \log \left(\frac{p^*(\mathbf{x})}{p(\mathbf{x}|\mathbf{D})} \right) d\mathbf{x} \quad (2.3)$$

Treating $p^*(\mathbf{x})$ a constant for a given $\mathbf{D}$, the optimization in Equation (2.3) becomes maximizing the log likelihood of $p(\mathbf{x}|\mathbf{D})$.

Assuming ϵ a Gaussian noise with variance σ^2 , $p(\mathbf{x}|\mathbf{y}, \mathbf{D})$ can be written as

$$p(\mathbf{x}|\mathbf{y}, \mathbf{D}) = \frac{1}{Z} \exp \left(-\frac{(x - \sum_{i=1}^k y_i \mathbf{d}_i)^2}{2\sigma^2} \right) \quad (2.4)$$

Using sparsity-inducing prior, we write

$$p(y_i) = \frac{1}{Z} \exp(-\beta C(y_i)) \quad (2.5)$$

where $C(\cdot)$ serves as a sparsity penalty (e.g., a Laplacian prior).

Having defined $p(\mathbf{x}|\mathbf{y}, \mathbf{D})$ and $p(\mathbf{y})$, the likelihood is $p(\mathbf{x}|\mathbf{D}) = \int p(\mathbf{x}|\mathbf{y}, \mathbf{D})p(\mathbf{y})d\mathbf{y}$, and our problem reduces to finding

$$\mathbf{D}^* = \operatorname{argmax}_{\mathbf{D}} \langle \log(p(\mathbf{x}|\mathbf{D})) \rangle \quad (2.6)$$

where $\langle \cdot \rangle$ denotes averaging over the input data.

Unfortunately, the integral over $\mathbf{y}$ to obtain $p(\mathbf{x}|\mathbf{D})$ is in general intractable. Olshausen & Field [36] suggest an approximate method for optimizing the dictionary $\mathbf{D}$

$$\mathbf{D}^{*'} = \operatorname{argmax}_{\mathbf{D}} \langle \max_{\mathbf{y}} \log(p(\mathbf{x}|\mathbf{D})) \rangle \quad (2.7)$$

Since maximizing $\log p(\mathbf{x}|\mathbf{D})$ is equivalent to minimizing $-\log(p(\mathbf{x}|\mathbf{y}, \mathbf{D})p(\mathbf{y}))$ where

$$-\log(p(\mathbf{x}|\mathbf{y}, \mathbf{D})p(\mathbf{y})) = \sum_{j=1}^m \left\| \mathbf{x}^{(j)} - \sum_{i=1}^k y_i^{(j)} \mathbf{d}_i \right\|^2 + \lambda \sum_{i=1}^k C(y_i^{(j)}) \quad (2.8)$$

when $\lambda = 2\sigma^2\beta$. We can recover the original optimization problem

$$\mathbf{D}^*, \mathbf{y}^* = \operatorname{argmin}_{\mathbf{D}, \mathbf{y}} \sum_{j=1}^m \left\| \mathbf{x}^{(j)} - \sum_{i=1}^k y_i^{(j)} \mathbf{d}_i \right\|^2 + \lambda \sum_{i=1}^k C(y_i^{(j)}) \quad (2.9)$$

It can be seen that the ℓ_1 penalty corresponds to the use of the Laplacian prior in $C(\cdot)$. Unlike deep neural nets, such numerical optimization is inconvenient as it cannot be trivially mapped onto feed-forward processing. Despite its solid theoretical foundation, probabilistic sparse coding is less attractive as a generative model.

2.2 Generative Adversarial Models

Generative adversarial networks (GAN) are learning algorithms that use a supervised loss as part of the training. In this section, we review adversarial training techniques for a generative model with an emphasis on its conditional variant, which can be viewed as a multi-modal extension.

2.2.1 Generative adversarial networks (GAN)

Goodfellow et al. [18] introduce GAN as a new means to learn probability distributions for data. GAN constitute a generative model G and a discriminative model D that are simultaneously trained in the competition described by a two-player minimax game:

$$\min_G \max_D \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} [\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}} [\log(1 - D(G(\mathbf{z})))] \quad (2.10)$$

Here, the objective of G is to produce a data estimate $\hat{\mathbf{x}}$ to the real $\mathbf{x}$, using a latent input variable $\mathbf{z}$ (e.g., noise) with a prior distribution $p_{\mathbf{z}}$. On the other hand, $D(\mathbf{x})$ represents the probability that $\mathbf{x}$ originates from the real data distribution p_{data} . Hence, $D(G(\mathbf{z}))$ can

be used to evaluate the quality of generated data $\hat{\mathbf{x}} = G(\mathbf{z})$ with respect to the real $\mathbf{x}$. In practice, G and D are typically implemented as deep neural nets. We can train G and D by backpropagation.

2.2.2 Conditional generative adversarial networks (CGAN)

Mirza & Osindero [19] extend GAN by conditioning G and D on side information. For CGAN, Equation (2.10) is rewritten as

$$\min_G \max_D \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} [\log D(\mathbf{x}|\mathbf{y})] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}} [\log(1 - D(G(\mathbf{z}|\mathbf{y})))] \quad (2.11)$$

where the extra information $\mathbf{y}$ can be a class label or data from another modality. For the latter case, since certain modalities are often observed together (e.g., audio-video and image-text), it is convenient to use CGAN. In neural net implementation, G takes in $\mathbf{z}|\mathbf{y}$ as a joint representation that concatenates $\mathbf{z}$ and $\mathbf{y}$ into a single input vector. Similarly, another joint representation is used to train D .

2.2.3 Deep convolutional generative adversarial networks (DCGAN)

Convolutional neural nets (CNNs) remain to be state-of-the-art for visual recognition tasks. Radford et al. [21] present convincing evidence in favor of deep CNN as a strong candidate architecture for adversarial learning. Their DCGAN have shown to learn a hierarchy of representations from object parts to scenes for image generation tasks. According to Radford et al., the success of DCGAN is attributed to their three architectural modifications to CNN. First, strided convolutions replace deterministic spatial pooling (e.g., max and average pooling) to learn spatial down- and upsampling. Secondly, fully connected layers following convolutional layers are removed to allow deeper representations. Lastly, batch normalization, which conditions the input to have zero mean and unit variance, induces to learn more useful features. In this thesis, we adopt the `DCGAN.torch` framework¹ to implement convolutional generator and discriminator.

¹<https://github.com/soumith/dcgan.torch>

Chapter 3

Learning Multimodal Sparse Representation

3.1 Introduction

Different types of multiple data modalities can be used to describe the same phenomenon. These data from multiple sources are semantically correlated, and sometimes provide complementary information to each other. In order to exchange such information, it is important to capture a high-level correlation between data modalities with a compact set of latent variables. Such semantic correlation reveals shared statistical association between modalities and thus can provide complementary information for each other.

Multiple modalities are believed to benefit discriminative machine learning tasks. Using different sensors simultaneously, a scene from the same event can be described in multiple data modalities. For example, consider a multimedia event detection (MED) problem with class names such as “Dog show,” “Firework,” “Playing fetch with dogs,” and “Shooting a gun.” Judging based on the video modality only, “Dog show” and “Playing fetch with dogs” come close, featuring both people and dogs in some coordinated actions. However, the two classes are easier to discriminate by incorporating the audio modality in which “Dog show” is characterized by crowd noise and microphone announcement absent in “Playing fetch with dog.” On the other hand, “Firework” and “Shooting a gun” are hard to discriminate with audio, but their visual differences are useful.

In multimodal feature learning, one wishes to learn good shared representations across heterogeneous data modalities. We can form a union of unimodal features after learning features for each modality separately. This approach, however, has a drawback for being unable to learn patterns that occur jointly or selectively across modalities since unimodal learning emphasizes on relating information within one modality.

In this chapter, we present a sparse coding framework that can model the correlation between modalities. The framework aims to learn relationships at higher level by forcing to share sparse representation. Sparse coding has been recognized widely in machine learning applications such as classification, denoising, and recognition [41–47]. In particular, it is known that multimedia data can be well-represented as a sparse linear combination of basis vectors. For example, the abundance of unlabeled, photos in the web makes good large dictionaries readily available for sparse coding.

Our contributions are two-fold. First, we set up an experimental deep architecture built on multiple layers of sparse coding and pooling units. From this, we report promising results on classification with multimodal datasets. Secondly, we demonstrate the performance of multimodal sparse coding in a comprehensive set of applications. In particular, we include our result with TRECVID MED tasks for detecting high-level complex events in user-generated videos. We examine various settings of multimodal sparse coding (detailed in Section 3.2) using several multimodal datasets of semantically correlated pairs (audio/video and web images/text). Such semantic correlation reveals shared statistical association between modalities and thus can provide complementary information for each other.

3.2 Multimodal Feature Learning via Sparse Coding

This section describes our multimodal feature learning schemes for sparse coding. Our schemes are general and can readily be extended for more than two modalities. For clarity of explanation, we use two modalities a and b throughout the section.

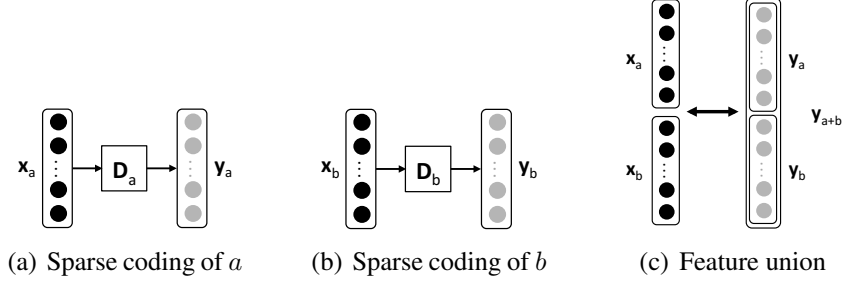


Figure 3.1: Unimodal sparse coding and feature union

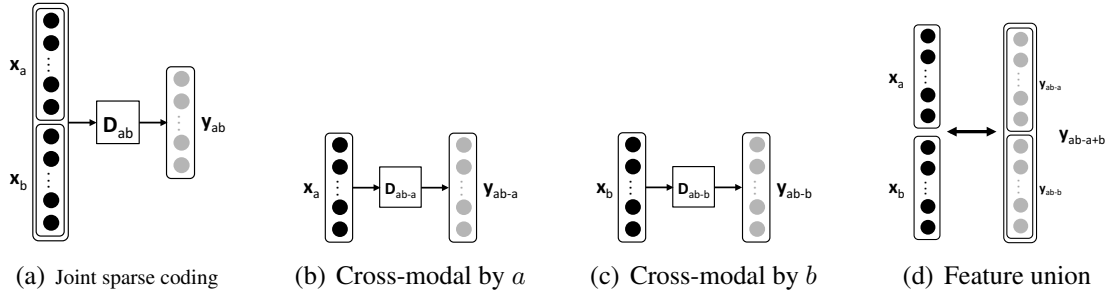


Figure 3.2: Multimodal sparse coding and feature union

3.2.1 Parallel unimodal sparse coding

A straightforward approach for sparse coding with two heterogeneous modalities (e.g., text and images) is to learn a *separate* dictionary of basis vectors for each modality. Figure 3.1 depicts unimodal sparse coding schemes for modalities a and b . We learn the two dictionaries D_a and D_b in parallel

$$\min_{D_a, Y_a^{(i)}} \sum_{i=1}^{n_a} \|x_a^{(i)} - D_a y_a^{(i)}\|_2^2 + \lambda \|y_a^{(i)}\|_1, \quad (3.1)$$

$$\min_{D_b, Y_b^{(i)}} \sum_{i=1}^{n_b} \|x_b^{(i)} - D_b y_b^{(i)}\|_2^2 + \lambda \|y_b^{(i)}\|_1. \quad (3.2)$$

Unimodal sparse coding of a takes in n_a unlabeled examples $x_a^{(i)}$ to train D_a while simultaneously computing corresponding sparse code $y_a^{(i)}$ under the regularization parameter λ . (We denote $x^{(i)}$ the i th training example.) Similarly for modality b , we train D_b from n_b unlabeled examples $x_b^{(i)}$ by computing $y_b^{(i)}$. As explained in Figure 3.1(c), we can form

$$\mathbf{y}_{a+b} = \begin{bmatrix} \mathbf{y}_a \\ \mathbf{y}_b \end{bmatrix}, \text{ a union of the unimodal feature vectors.}$$

3.2.2 Joint multimodal sparse coding

Union of the unimodal sparse codes $\mathbf{y}_{a+b}$ is a simple way to encapsulate the features from both modalities. However, the unimodal training model is flawed since it cannot capture correlations between the two modalities that could be beneficial for inference tasks. To remedy the lack of joint learning, we propose a multimodal sparse coding scheme illustrated in Figure 3.2(a). We use the joint sparse coding technique used in image super-resolution [48]

$$\min_{\mathbf{D}_{ab}, \mathbf{y}_{ab}^{(i)}} \sum_{i=1}^n \|\mathbf{x}_{ab}^{(i)} - \mathbf{D}_{ab} \mathbf{y}_{ab}^{(i)}\|_2^2 + \lambda' \|\mathbf{y}_{ab}^{(i)}\|_1. \quad (3.3)$$

Here, we train with concatenated input vectors $\mathbf{x}_{ab}^{(i)} = \begin{bmatrix} \frac{1}{\sqrt{N_a}} \mathbf{x}_a^{(i)} \\ \frac{1}{\sqrt{N_b}} \mathbf{x}_b^{(i)} \end{bmatrix}$, where N_a and N_b are dimensions of $\mathbf{x}_a$ and $\mathbf{x}_b$, respectively. As an interesting property, we can decompose the multimodal dictionary as $\mathbf{D}_{ab} = \begin{bmatrix} \frac{1}{\sqrt{N_a}} \mathbf{D}_{ab-a} \\ \frac{1}{\sqrt{N_b}} \mathbf{D}_{ab-b} \end{bmatrix}$ to perform the cross-modal sparse coding

$$\min_{\mathbf{y}_{ab-a}^{(i)}} \sum_{i=1}^{n_a} \|\mathbf{x}_a^{(i)} - \mathbf{D}_{ab-a} \mathbf{y}_{ab-a}^{(i)}\|_2^2 + \lambda'' \|\mathbf{y}_{ab-a}^{(i)}\|_1, \quad (3.4)$$

$$\min_{\mathbf{y}_{ab-b}^{(i)}} \sum_{i=1}^{n_b} \|\mathbf{x}_b^{(i)} - \mathbf{D}_{ab-b} \mathbf{y}_{ab-b}^{(i)}\|_2^2 + \lambda'' \|\mathbf{y}_{ab-b}^{(i)}\|_1. \quad (3.5)$$

In principle, joint sparse coding in Equation (3.3) combines the objectives of Equations (3.4) and (3.5), forcing the sparse codes $\mathbf{y}_{ab-a}^{(i)}$ and $\mathbf{y}_{ab-b}^{(i)}$ to share the same values when $\lambda' = (\frac{1}{N_a} + \frac{1}{N_b})\lambda''$. Ideally, we could have $\mathbf{y}_{ab}^{(i)} = \mathbf{y}_{ab-a}^{(i)} = \mathbf{y}_{ab-b}^{(i)}$, although empirical values determined by the three different optimizations would differ in reality. According to Yang et al. [48], $\mathbf{x}_a$ and $\mathbf{x}_b$ are highly correlated as the low and high resolution images are originated from the same source. However, if $\mathbf{x}_a$ and $\mathbf{x}_b$ come from two different modalities, their correlation is present in semantics. Thus, equality assumption among $\mathbf{y}_{ab}^{(i)}$, $\mathbf{y}_{ab-a}^{(i)}$, and $\mathbf{y}_{ab-b}^{(i)}$ is even less likely to be met. For that reason, we introduce cross-modal sparse cod-

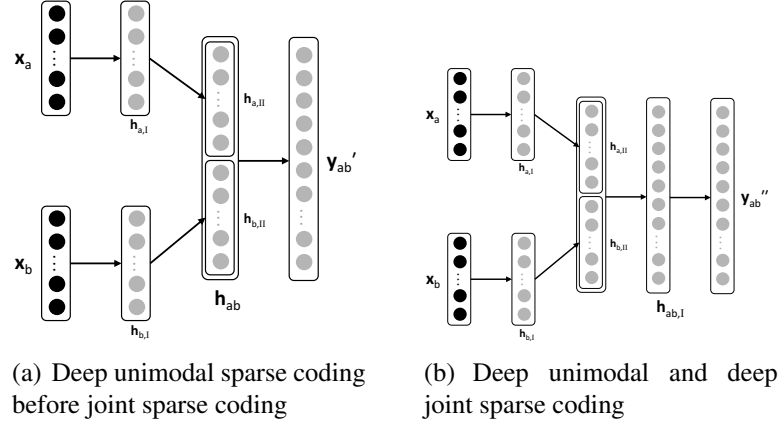


Figure 3.3: Deep learning approaches for multimodal sparse coding

ing that captures weak correlation between heterogeneous modalities, resulting sparse code that is more discriminative. Cross-modal sparse coding first trains a joint dictionary D_{ab} . Then in test time, cross-modal sparse codes are computed using sub-dictionaries D_{ab-a} and D_{ab-b} . Various feature formation possibilities on multimodal sparse coding—joint, cross-modal, and union of cross-modal sparse codes are explained in Figure 3.2.

3.2.3 Hierarchical multimodal sparse coding

So far, we have only considered shallow learning architectures using a single layer of sparse coding and dictionary learning. The shallow architecture is capable of learning the features jointly, but it may not be sufficient to capture the complex semantic correlations between the modalities fully. We expect modalities with high semantic correlation to be stable in hierarchical architecture as it can utilize a large number of hidden layers and parameters to extract more meaningful high-level representation from modalities. Composing higher-level representation using low-level features should be advantageous to contextual data such as human language, speech, audio, and sequence of image patterns. Hierarchical composition of sparse codes has shown to help unveil separability of data invisible from their lower-level features in unimodal settings [49,50]. Therefore, we consider deep architectures for multimodal sparse coding.

In Figure 3.3, we propose two possible architectures. Ngiam et al. [9] report their RBM-

based approaches beneficial when applying deep learning on each modality before joint training. Adopting their configuration, we use two layers of sparse coding for each modality followed by the joint sparse coding as illustrated in Figure 3.3(a). We write two-layer sparse coding for modality a

$$\min_{\mathbf{D}_{a,I}, \mathbf{y}_{a,I}^{(i)}} \sum_{i=1}^{n_a} \|\mathbf{x}_a^{(i)} - \mathbf{D}_{a,I} \mathbf{y}_{a,I}^{(i)}\|_2^2 + \gamma \|\mathbf{y}_{a,I}^{(i)}\|_1, \quad (3.6)$$

$$\{\mathbf{y}_{a,I}^{(i)}, \dots, \mathbf{y}_{a,I}^{(i+M_I-1)}\} \xrightarrow{\text{pool}} \mathbf{h}_{a,I}^{(j)}, \quad (3.7)$$

$$\min_{\mathbf{D}_{a,II}, \mathbf{y}_{a,II}^{(j)}} \sum_{j=1}^{n'_a} \|\mathbf{h}_{a,I}^{(j)} - \mathbf{D}_{a,II} \mathbf{y}_{a,II}^{(j)}\|_2^2 + \gamma \|\mathbf{y}_{a,II}^{(j)}\|_1, \quad (3.8)$$

$$\{\mathbf{y}_{a,II}^{(j)}, \dots, \mathbf{y}_{a,II}^{(j+M_{II}-1)}\} \xrightarrow{\text{pool}} \mathbf{h}_{a,II}^{(k)}. \quad (3.9)$$

We denote $\mathbf{D}_{a,I}$ and $\mathbf{D}_{a,II}$ the dictionaries learned by the two sparse coding layers for a , *unpooled* sparse codes $\mathbf{y}_{a,I}$ and $\mathbf{y}_{a,II}$, and the hidden activations $\mathbf{h}_{a,I}$ and $\mathbf{h}_{a,II}$ by max (or average) pooling sparse codes. Max pooling factors M_I and M_{II} refer to the number of sparse codes aggregated to one pooled representation. They are determined empirically. Since sparse coding takes in dense input vectors and produces sparse output vectors, it is a poor fit for multilayering. Hence, we interlace nonlinear pooling units between sparse coding layers and aggregate multiple sparse vectors nonlinearly to a dense vector before passing onto next layer. Similar to modality a , we can work out $\mathbf{D}_{b,I}$, $\mathbf{D}_{b,II}$, $\mathbf{y}_{b,I}$, $\mathbf{y}_{b,II}$, $\mathbf{h}_{b,I}$, and $\mathbf{h}_{b,II}$ for modality b . Ultimately, the joint feature representation $\mathbf{y}'_{ab}$ in Figure 3.3(a) is learned by

$$\min_{\mathbf{D}'_{ab}, \mathbf{y}'_{ab}} \sum_{k=1}^{n'} \|\mathbf{h}_{ab}^{(k)} - \mathbf{D}'_{ab} \mathbf{y}'_{ab}^{(k)}\|_2^2 + \gamma' \|\mathbf{y}'_{ab}^{(k)}\|_1 \quad (3.10)$$

where we input the concatenated deep hidden representations $\mathbf{h}_{ab}^{(k)} = \begin{bmatrix} \mathbf{h}_{a,II}^{(k)} \\ \mathbf{h}_{b,II}^{(k)} \end{bmatrix}$ to train $\mathbf{D}'_{ab}$.

Moreover, we consider additional joint sparse coding layer as illustrated in Figure 3.3(b). Again, we form dense hidden activations $\mathbf{h}_{ab,I}$ by pooling multiple $\mathbf{y}'_{ab}$ vectors. We compute

Table 3.1: Peak signal-to-noise ratio results (in decibels) on image denoising using denoising autoencoder and joint sparse coding with a range of standard deviation σ .

σ /PSNR	DAE	Joint sparse coding
0.005/23.20	25.61	25.71
0.01/20.28	24.14	23.87
0.1/11.52	18.74	18.19

the deeper representation $\mathbf{y}_{ab}''$ by

$$\min_{\mathbf{D}_{ab}'', \mathbf{y}_{ab}''^{(l)}} \sum_{l=1}^{n''} \|\mathbf{h}_{ab,I}^{(l)} - \mathbf{D}_{ab}'' \mathbf{y}_{ab}''^{(l)}\|_2^2 + \gamma'' \|\mathbf{y}_{ab}''^{(l)}\|_1. \quad (3.11)$$

3.3 Experiments

3.3.1 Image denoising

In our first experiment, we evaluate the image denoising problem, where zero-mean Gaussian noise is to be removed from a given image. The proposed learning algorithm is used to jointly learn associations between clean and noisy images. We consider clean and its noisy counterpart as two input modalities and recover a clean image from a noisy one. We randomly select 2,500 images from CIFAR-10 [51] and add zero-mean Gaussian noise with a range of standard deviation σ to generate noisy images. We use 2,000 pairs of clean and noisy images for training joint dictionary and 500 pairs for testing. To test, we input a noisy image and compute cross-modal (noisy to clean) sparse code to recover a clean estimate. We compare the performance of the joint multimodal sparse coding with denoising autoencoder (DAE) [52]. Table 3.1 summarizes denoising results for the joint sparse coding and DAE. In this experiment, the dictionaries used were of size $N \times K$, designed to handle clean and noisy image patches of size 4×4 pixels ($N = 32$, $K = 300$). Every result reported is an average over 5 experiments. For different realization of the noise, joint sparse coding improves the quality of noisy images, and DAE often achieves comparable results.

In Figure 3.4, we visualize the generated clean images from noisy images with $\sigma = 0.01$. Notice that joint sparse coding can learn shared association between clean and noisy images as in DAE to achieve comparable results.

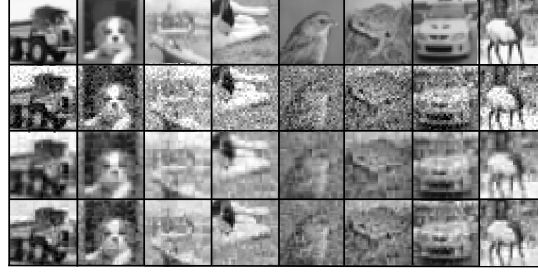


Figure 3.4: Example of the denoising results with $\sigma = 0.01$. From top to bottom, we visualize clean original image, noisy image, denoised images using DAE and joint sparse coding.

3.3.2 Multimedia event detection using audio and video

In this section, we apply our multimodal sparse coding schemes to multimedia event detection (MED) tasks. MED aims to identify complex activities encompassing various human actions, objects, and their interactions at different places and time. MED is considered more difficult than concept analysis such as action recognition and has received significant attention in computer vision and machine learning research.

Dataset, tasks, and metrics. We use the TRECVID 2014 dataset [53] to evaluate our schemes. We consider the event detection and retrieval tasks using the 10Ex and 100Ex data scenarios, where 10Ex gives 10 multimedia examples per event, and 100 examples per event for 100Ex. There are 20 event classes (E021 to E040) with event names such as “Bike trick,” “Dog show,” and “Marriage proposal.” We evaluate the MED performance trained on the features learned from our unimodal and multimodal sparse coding schemes. In particular, we compute classification accuracy and mean average precision (mAP) metrics according to the NIST standard on the following experiments: 1) cross-validation on 10Ex; and 2) train using 10Ex and test with 100Ex.

Data processing. For processing efficiency and scalability, we use keyframe-based feature extraction for audio-video data. We apply a simple two-pass algorithm that computes color histogram difference of any two successive frames and determines a keyframe candidate based on the threshold calculated on the mean and standard deviation of the histogram differences. We examine the number of different colors present in the keyframe candidates and discard the ones with less than 26 colors (e.g., all-black or all-white) to ensure non-

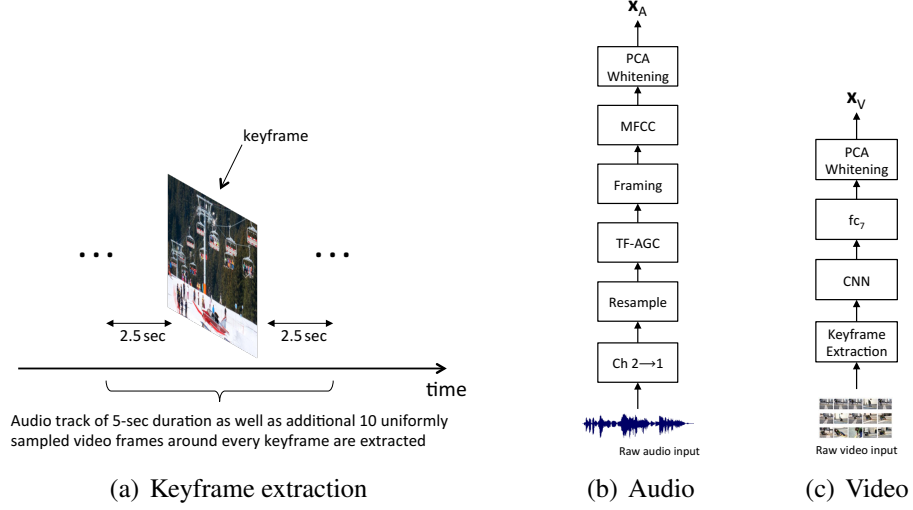


Figure 3.5: TRECVID audio-video data processing

blank keyframes.

Around each keyframe, we extract 5-sec audio data and additional 10 uniformly sampled video frames within the duration as illustrated in Figure 3.5(a). If extracted audio is stereo, we take only its left channel. The audio waveform is resampled to 22.05 kHz and regularized by the time-frequency automatic gain control (TF-AGC) to balance the energy in sub-bands. We form audio frames using a 46-msec Hann window with 50% overlap between successive frames for smoothing. For each frame, we compute 16 Mel-frequency cepstral coefficients (MFCCs) [54] as the low-level audio feature. In addition, we append 16 delta cepstral and 16 delta-delta cepstral coefficients, which make our low-level audio feature vectors 48 dimensional. We apply PCA whitening before unsupervised learning. The complete audio preprocessing steps are described in Figure 3.5(b).

For video preprocessing, we have tried out pretrained convolutional neural network (CNN) models and ended up choosing `VGG_ILSVRC_19_layers` by University of Oxford’s Visual Geometry Group (VGG) [55] for the ImageNet Large-scale Visual Recognition Challenge (ILSVRC). As depicted in Figure 3.5(c), we run the CNN feedforward passes with the extracted video frames and take 4,096-dimensional hidden activation from fc_7 , the highest hidden layer before the final ReLU. By PCA whitening, we reduce the dimensionality to 128.

Table 3.2: Mean accuracy and mAP performance of sparse coding schemes

	Audio-only (Fig. 3.1(a))	Unimodal Video-only (Fig. 3.1(b))	Union (Fig. 3.1(c))	Audio (Fig. 3.2(b))	Multimodal Video (Fig. 3.2(c))	Joint (Fig. 3.2(a))	Union (Fig. 3.2(d))
Mean accuracy (cross-val. 10Ex)	69%	86%	89%	75%	87%	90%	91%
mAP (cross-val. 10Ex)	20.0%	28.1%	34.8%	27.4%	33.1%	35.3%	37.9%
Mean accuracy (10Ex/100Ex)	56%	64%	71%	58%	67%	71%	74%
mAP (10Ex/100Ex)	17.3%	28.9%	30.5%	23.6%	28.0%	28.4%	33.2%

Feature learning for MED. We build feature vectors by sparse coding the preprocessed audio and video data. We use the number of basis vectors $K = 512$ same for all dictionaries under unimodal and multimodal sparse coding schemes. We aggregate sparse codes around each keyframe of a training example by max pooling to form feature vectors for classification. (The pooled feature vectors can scale to file level.) We train linear, 1-vs-all SVM classifiers for each event whose hyper-parameters are determined by 5-fold cross-validation on 10Ex.

Other feature learning schemes for comparison. We consider other unsupervised methods to learn audio-video features for comparison. We report the results for Gaussian mixture model (GMM) and restricted Boltzmann machine (RBM) under similar unimodal and multimodal settings. For GMM, we use the expectation-maximization (EM) to fit the preprocessed input vectors audio-only, video-only, concatenated audio-video in 512 mixtures and form GMM supervectors [56] containing posterior probabilities with respect to each Gaussian. The max-pooled GMM supervectors are used to train SVMs. We adopt the shallow bimodal pretraining model by Ngiam et al. [9] for RBM. For fairness, we use the hidden layer of a size 512, and the max-pooled activations are used to train SVMs. A target sparsity of 0.1 is applied to both GMM and RBM.

Results. Table 3.2 presents the classification accuracy and mAP performance of unimodal and multimodal sparse coding schemes. For the 10Ex/100Ex experiment, we have used the best parameter setting from the 10Ex cross-validation to test 100Ex examples. In general, we observe that the union of unimodal audio and video feature vectors perform better than using only unimodal or cross-modal features. The multimodal union scheme performs better than the joint schemes. The union schemes, however, double feature di-

Table 3.3: Mean accuracy and mAP performance for GMM and RBM on 10Ex/100Ex

Feature learning schemes	Mean accuracy	mAP
Union of unimodal GMM features (Figure 3.1(c))	66%	23.5%
Multimodal joint GMM feature (Figure 3.2(a))	68%	25.2%
Union of unimodal RBM features (Figure 3.1(c))	70%	30.1%
Multimodal joint RBM feature (Figure 3.2(a))	72%	31.3%

dimensionality since our union operation concatenates the two feature vectors. Joint feature vector is an economical way of combining both the audio and video features while keeping the same dimension as audio-only or video-only.

In Table 3.3, we report the mean accuracy and mAP for GMM and RBM under the union and joint feature learning schemes on the 10Ex/100Ex experiment. Our results show that sparse coding is better than GMM by 5–6% in accuracy and 7–8% in mAP. However, we find that the performance of RBM is on par with sparse coding.

3.3.3 Wikipedia category classification using images and text

Each article in Wikipedia dataset contains paragraphs of text on a subject with a corresponding picture. These text-image pairs are annotated with a label from 10 categories: art, biology, geography, history, literature, media, music, royalty, sport, and warfare. The corpus contains a total of 2,866 documents. We have split the documents into five folds. Along with the dataset, the authors have supplied the features for images and text. Each image is represented by a histogram of a 128-codeword SIFT [57] features, and each text is represented by a histogram of a 10-topic latent Dirichlet allocation (LDA) model [58]. We use the SIFT features for images and LDA features for text as input to sparse coding algorithms, and then train the 1-vs-all SVMs. After that, we predict the labels of the test image-text pairs and report the accuracy. In this experiment, we have used $K \approx 3N$.

Table 3.4 reports the results for various feature representations. In comparison between unimodal sparse coding of images or text alone, text features outperform the image features.

Table 3.4: Classification performance for image-text classification on Wikipedia dataset.

Feature Representation	Accuracy
(a) Sparse coding of images (Figure 3.1a)	16.93%
(b) Sparse coding of text (Figure 3.1b)	61.89%
(c) Unimodal feature union (Figure 3.1c)	63.38%
(d) Joint sparse coding (Figure 3.2a)	66.44%
(e) Cross-modal by images (Figure 3.2b)	21.28%
(f) Cross-modal by text (Figure 3.2c)	66.43%
(g) Multimodal feature union (Figure 3.2d)	67.23%

In Wikipedia, category membership is mostly driven by text. Categorization solely based on image is ambiguous and difficult even for a human. Thus, it is expected that image features would have lower accuracy than text features. The union of unimodal image and text features (Table 3.4c) further improves the accuracy. Joint sparse coding (Table 3.4d) is able to learn multimodal features that go beyond simply concatenating the two unimodal features.

We notice that learning shared association between image and text can improve classification accuracy even when only a single modality is available for training and testing. Cross-modal by images (Table 3.4e) improves accuracy by 4.35% compared to unimodal sparse coding of images (Table 3.4a), and cross-modal by text (Table 3.4f) achieves 4.54% higher accuracy than unimodal sparse coding of text (Table 3.4b). When the cross-modal features by images and text are concatenated, it outperforms the other feature combinations.

For all feature representations, multimodal features significantly outperform the unimodal features. This shows that the learned shared association between multiple modalities is useful for both cases when a single or multiple modalities are available for training and testing.

Figure 3.6 compares classification accuracies of joint sparse coding (Table 3.4d), cross-

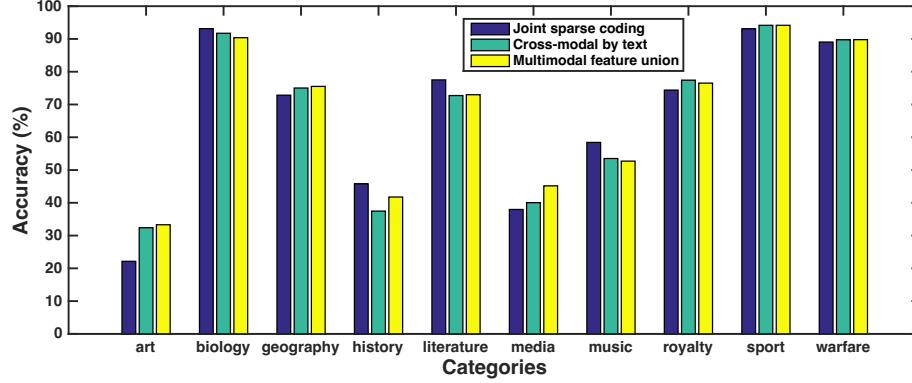


Figure 3.6: Category classification comparison of joint sparse coding, cross-modal by text, and multimodal feature union on Wikipedia dataset

modal by text (Table 3.4f), and multimodal feature union (Table 3.4g) in 10 categories. Although multimodal feature union does not achieve the best accuracy value for several categories, the classification accuracy is almost close to the best one.

3.3.4 Twitter sentiment classification using images and text

PhotoTweet dataset includes 603 Twitter messages (*tweets*) with their associated photos. The benchmark is designed for Twitter user’s sentiment prediction using visual and textual features. PhotoTweet is collected in November 2012 via the PeopleBrowsr API. Ground truths labels are obtained by Amazon Mechanic Turk annotation, resulting in 470 positive and 133 negative sentiments. The authors of the dataset have partitioned the dataset into five folds for cross-validation.

We represent the textual data using binary *bag-of-words* embedding, producing 2,688 dimensions vector. We process the embedded tweet data to consecutive patches of a configurable size, $N_a = 42$. The dimension of resulting patches are reduced to 12 dimensions with PCA whitening. For images, we consider a *per-image* feature vector from non-overlapping patches drawn from a receptive field with width $w = 4$ pixels. Thus, each (colored) patch has size $N_b = 3 \times 4 \times 4 = 48$. In unsupervised learning, we precondition visual and textual patches with mean removal and whitening before sparse coding. We have used a dictionary size $K \approx 4N$ for both unimodal and multimodal settings. For max pooling, we use pooling factors M in 10s.

Table 3.5: Classification performance for image-text classification on PhotoTweet dataset.

Feature Representation	Accuracy
(a) Sparse coding of images (Figure 3.1a)	60.91%
(b) Sparse coding of text (Figure 3.1b)	58.07%
(c) Unimodal feature union (Figure 3.1c)	66.10%
(d) Joint sparse coding (Figure 3.2a)	70.29%
(e) Cross-modal by images (Figure 3.2b)	66.64%
(f) Cross-modal by text (Figure 3.2c)	62.01%
(g) Multimodal feature union (Figure 3.2d)	71.95%
(h) Deep multimodal sparse coding	75.16%

In Table 3.5, we present the sentiment classification performances using linear 1-vs-all SVM. We compare performances of sparse coding methods discussed in Section 3.2. On Twitter sentiment classification task, image and text features alone are equally useful. Thus, the image features complement the text features in unimodal feature union achieving 66.1%. Shared representation learned from unsupervised learning stage helps increase classification performance by 4–6% in cross-modal features compared to unimodal features. The best feature in shallow model is multimodal feature union (Table 3.5g). Finally, we further improve the performance by building deep representations that model the correlation across the learned shallow representations. After having tried both architectures in Figures 3.3a and b, we obtained the result summarized the better of two in Table 3.5.

Table 3.6 presents a summary that compares the sentiment classification performances of our approach and the previous work. The authors of the PhotoTweet dataset use a combination of SentiStrength [61] for textual features and SentiBank [59] for mid-level visual features. The combined method simply concatenates the features from SentiStrength and SentiBank and does not learn shared association between modalities. We notice that our multimodal feature union (Table 3.5g) outperforms SentiStrength+SentiBank, emphasizing the importance of shared learning across multiple modalities. Baecchi et al. [60] use

Table 3.6: Classification performance comparison between multimodal sparse coding and other existing work on PhotoTweet dataset.

Feature Representation	Accuracy	
	Linear SVM	Logistic Regr.
SentiStrength + SentiBank [59]	68%	72%
Shallow multimodal sparse coding	71.95%	74.65%
CBOW-DA-LR [60]	N/A	79%
Deep multimodal sparse coding	75.16%	80.70%

an extension of Continuous bag-of-words model for text and denoising autoencoder for images. Again, the textual and visual features are concatenated. We compare this method with hierarchical learning with our deep multimodal sparse coding (Table 3.5h) and show that our method yields better classification result.

3.4 Summary

We have presented multimodal sparse coding algorithms that model the semantic correlation between modalities. We have shown that multimodal features significantly outperform the unimodal features. Our experimental results also indicate that the multimodal features learned by our algorithms are more discriminative than the feature formed by concatenating multiple unimodal features. In addition, cross-modal features computed using only one modality also lead to better performance than unimodal features. This suggests we can learn better features for one modality from joint learning of multiple modalities. The effectiveness of our approach is demonstrated in various multimedia applications such as image denoising, MED, category and sentiment classification.

Beyond learning multimodal representations, sparse coding can be probabilistically interpreted as a generative model and used for cross-modal synthesis. Given an entity in one modality, the task is to synthesize the same entity in a different modality. However, as discussed in Section 2.1.2, the numerical optimization required in probabilistic sparse coding is not conveniently mapped to feed-forward processing compared to deep neural nets.

Chapter 4

Applying Conditional Multimodality to Image Generation

4.1 Introduction

The recently introduced generative adversarial network (GAN) [18] is designed to benefit from contradictory objectives between a pair of simultaneously trained learning models. In the training of a generative model, GAN uses the discriminator’s prediction output as a feedback to compute the generator’s loss function. Desirably, when the training reaches to a steady-state equilibrium where the discriminator confusion is at maximum on synthetic data from the generator, one can conclude the overall GAN objective accomplished.

In this chapter, we aim to extend the GAN framework for automatic text-to-image synthesis. We are interested in generating a perceptually high-quality image that matches a descriptive text input. In particular, we care for two important aspects: contextual match to the input text and realistic image production. Good cross-modal translation results in a generated image closely matching the given text description where a realistic image can hardly be distinguished from natural images.

To ensure good cross-modal translation, we adopt a *contextual* loss term in the generator following the conditional GAN framework [19]. To generate realistic images, we in addition introduce *perceptual* loss terms for the generator, corresponding to pixel, feature activation, and texture reconstruction losses. Thus our approach is to regularize the original

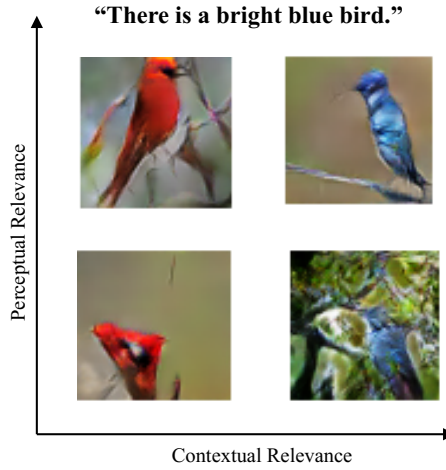


Figure 4.1: Four synthetic bird images generated for the given text, displaying different levels of contextual and perceptual relevance. The upper right image is most desirable, as it has higher relevance on both accounts.

minimax optimization for GAN with both contextual and perceptual loss terms.

We illustrate the effect of contextual and perceptual losses in Figure 4.1. Here, the task is to generate an image for text, “There is a bright blue bird.” The two axes in the figure indicate the degree of contextual and perceptual relevances. The red birds on the left are contextual mismatches because the input text mentions only ‘blue’ bird. Hinted by higher perceptual relevance, the bird images on the top have more natural and better overall perceptual quality than the bottom. We are interested in synthesizing images in the upper right corner which have higher relevance in both context and perception.

The use of GAN to generate realistic images has been attempted by others. Radford et al. [21] propose deep convolutional generative adversarial net (DCGAN) that takes in a random noise vector to generate synthetic images. We base our approach on conditional GAN by Mirza & Osindero [19]. By conditioning both generator and discriminator networks on the text input, conditional DCGAN can model the phenomenon that many different natural images can be mapped to the same text description.

Several encouraging results follow in recent literature. Gauthier [62] applies conditional GAN to generating MNIST and facial images by using class labels as side information. Denton et al. [63] introduce a Laplacian pyramid extension to conditional GAN

that can generate an image at different resolutions. The notion of perceptual loss in image processing is originated by Gatys et al. [64]. They have proposed to use the Gram matrix of convolutional neural net activations (i.e., high-level image feature vectors) between two images to make artistic style transfer possible. In Johnson et al. [65], the same perceptual loss function is used for image super-resolution. Ledig et al. [66] propose super-resolution GAN (SRGAN) that combines per-pixel, VGG, and adversarial losses as the perceptual loss function. For VGG loss, they take the ReLU activation layers of the pre-trained 19-layer convolutional neural net by Oxford’s Visual Geometry Group (VGG) [67].

Reed et al. [20] present an architecture based on conditional DCGAN. They have trained bird and flower image generators conditioned on text features computed from a character-level recurrent neural net. Our work builds on Reed et al. [20]. We tackle the same bimodal text-to-image task and use the same datasets. However, our generator minimizes on a perceptual loss term given by the Gram, per-pixel, or VGG loss function alongside the contextual loss obtained from the discriminator in conditional GAN. Ledig et al. have explored the per-pixel, VGG, and adversarial loss functions on the image super-resolution task, not text-to-image synthesis.

4.2 Method

Our task is to generate realistic images that match text input. At a high-level, we build a DCGAN as in Reed et al. [20] and train it with contextual and perceptual loss terms by conditioning on the input text. The main contribution of our approach is the inclusion of additional perceptual loss in training the generator. In particular, we compare the use of three different perceptual loss functions based on pixel, activation, and texture reconstruction. This section first presents our network architecture and then describes the formulation of loss functions used to train our DCGAN.

4.2.1 Architecture

Figure 4.2 depicts our network architecture. We use input text encoding $\varphi(t)$ at both generator and discriminator networks. In the generator network G , $\varphi(t)$ (in reduced dimension-

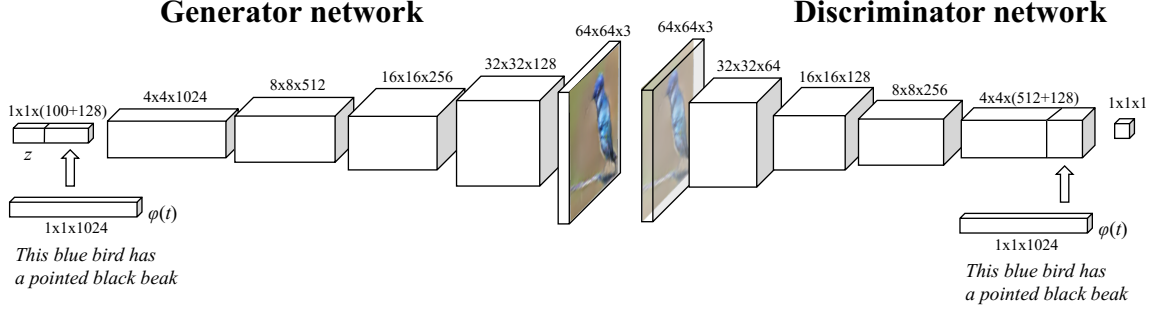


Figure 4.2: Architecture of the generator and discriminator networks of our conditional GAN synthesis model.

ality) is concatenated with a noise sample $\mathbf{z}$ and propagated through stages of fractional-strided convolution processing. In the discriminator network D , an input image is processed through layers of strided convolution before concatenated to $\varphi(t)$ (in reduced dimensionality) for computing the final discriminator score. Both G and D are optimized by gradient descent on their loss functions $\mathcal{L}_G$ and $\mathcal{L}_D$, respectively.

We denote $\mathbf{x}$ for image data, t for text data, $\mathbf{h} = \varphi(t)$ for text encoding, and $\mathbf{z}$ for noise input. A superscript (i) is used to designate i th example. For our task, a training example is an image-text pair. When applying m training examples (e.g., mini-batches) $\{(\mathbf{x}^{(1)}, \mathbf{h}^{(1)}), \dots, (\mathbf{x}^{(m)}, \mathbf{h}^{(m)})\}$ with noise input $\{\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(m)}\}$ sampled from the prior p_z , G generates corresponding synthetic images $\{\hat{\mathbf{x}}^{(1)}, \dots, \hat{\mathbf{x}}^{(m)}\}$.

The discriminator loss function is given by

$$\begin{aligned} \mathcal{L}_D = & -\frac{1}{m} \sum_i \log(D(\mathbf{x}^{(i)} | \mathbf{h}^{(i)})) \\ & -\frac{1}{2} [\log(1 - D(\mathbf{x}^{(i)} | \hat{\mathbf{h}}^{(i)})) + \log(1 - D(\hat{\mathbf{x}}^{(i)} | \mathbf{h}^{(i)}))] \end{aligned} \quad (4.1)$$

where $\{\mathbf{x}^{(i)} | \mathbf{h}^{(i)}\}$ corresponds a real image with its corresponding right text, $\{\mathbf{x}^{(i)} | \hat{\mathbf{h}}^{(i)}\}$ real image with arbitrary text (randomly chosen from a different image class), and $\{\hat{\mathbf{x}}^{(i)} | \mathbf{h}^{(i)}\}$ synthetic image with right text.

The generator loss function is a weighted sum of two parts

$$\mathcal{L}_G = \ell_{\text{cont}} + \lambda \ell_{\text{perc}} \quad (4.2)$$

where ℓ_{cont} is contextual loss, ℓ_{perc} perceptual loss, and a weight parameter λ .

4.2.2 Adversarial loss

The contextual loss used in our implementation is:

$$\ell_{\text{cont}} = -\frac{1}{m} \sum_i \log(D(G(\mathbf{z}^{(i)}|\mathbf{h}^{(i)}))) \quad (4.3)$$

where $D(G(\mathbf{z}|\mathbf{h}))$ is the probability that the generated image and the conditioned text form a real contextual pair. For better gradient behavior, we minimize $-\log(D(G(\mathbf{z}|\mathbf{h})))$ rather than minimizing the original $\log(1 - D(G(\mathbf{z}|\mathbf{h})))$ [18].

4.2.3 Perceptual loss

The contextual (adversarial) loss for the generator, as presented in Reed et al. [20], only considers the semantic relatedness between images and text descriptions without an explicit regularization term that penalizes large visual changes between the synthetic and real images. In contrast, in addition to the contextual loss term, we adopt a perceptual loss ℓ_{perc} as a part of the loss in training the generator. We propose three perceptual loss functions each aiming to enforce perceptual similarity between the real and the generated images. That is, by minimizing the perceptual loss the visual difference between the synthetic and the real images can be lowered. This helps preserve the perceptual realism.

Pixel reconstruction loss. Making image adjustment to minimize pixel-wise losses is a simple approach to encourage visual similarity between images. The pixel reconstruction loss calculates the mean squared error between a real image $\mathbf{x}^{(i)}$ and a corresponding synthetic image $\hat{\mathbf{x}}^{(i)}$ as

$$\ell_{\text{perc.Pix}} = \frac{1}{m} \sum_i \|\mathbf{x}^{(i)} - \hat{\mathbf{x}}^{(i)}\|_2^2 \quad (4.4)$$

where $\|\cdot\|_2$ is the ℓ_2 norm. The pixel reconstruction loss function encourages the pixels of the two images to match. Unlike image super-resolution, text-to-image synthesis involves one-to-many mapping between two different data kinds. Thus, we find usage of high-level image features more appropriate.

Activation reconstruction loss. Instead of promoting pixel-wise match between synthetic and real images, we can encourage high-level feature representations of the images to be similar. Let $\mathbf{A}_{\mathbf{x}^{(i)}}^j$ or $\mathbf{A}_{\hat{\mathbf{x}}^{(i)}}^j$ be the rectified linear unit (ReLU) activation outputs of the j th convolutional layer within a classification network such as VGG when $\mathbf{x}^{(i)}$ or $\hat{\mathbf{x}}^{(i)}$ is used as an input, respectively. The feature reconstruction loss is defined as

$$\ell_{\text{perc.VGG}} = \frac{1}{m} \sum_i \|\mathbf{A}_{\mathbf{x}^{(i)}}^j - \mathbf{A}_{\hat{\mathbf{x}}^{(i)}}^j\|_F^2 \quad (4.5)$$

where $\|\cdot\|_F$ is the Frobenius norm. Here, we use a 19-layer VGG network pretrained on the ImageNet dataset [68]. As the VGG network is trained on ImageNet, which consists of more than 14 million images, activation outputs derived from high levels capture image content and overall structures such as object shapes that may be useful for classifying objects. By minimizing the differences in the activation outputs, we encourage the generated image to be classified similarly as the real image, thereby containing objects of the same class as those in the real image.

Texture reconstruction loss. Although image content and overall structures are well captured in the activation outputs, style-related features such as texture and recurring patterns may not. In order to capture whether the generated image and the real image use combinations of nearly identical set of supporting filters, we compare Gram matrices of the activation outputs, as previously being proposed for image super-resolution [65] and style transfer [64, 69]:

$$\ell_{\text{perc.Gram}} = \frac{1}{m} \sum_i \|S(\mathbf{A}_{\mathbf{x}^{(i)}}^j) - S(\mathbf{A}_{\hat{\mathbf{x}}^{(i)}}^j)\|_F^2 \quad (4.6)$$

where $S(\mathbf{A}_{\mathbf{x}^{(i)}}^j)$ and $S(\mathbf{A}_{\hat{\mathbf{x}}^{(i)}}^j)$ are the Gram matrix of activation vectors resulting from filters at j th convolutional layer for input $\mathbf{x}^{(i)}$ and $\hat{\mathbf{x}}^{(i)}$, respectively. Note that since the Gram matrix captures the correlations between these filters, it effectively describes an object-level style of an image.

4.3 Experimental Results

We evaluate the proposed models with perceptual losses on Caltech-UCSD Bird (CUB) [70] and Oxford-102 flower [71] datasets. CUB consists of 11,788 images of 200 bird species. The Oxford-102 has 8,189 images of flowers from 102 different types. A recent extension of these datasets has collected 10 visual description sentences for each image [72]. We follow train/test split by Reed et al. [72]. That is, the images in CUB are split into 150 training/validation and 50 test classes. The flower dataset is split into 82 training/validation and 20 test categories. Thus during testing, an input text is from a category not present in the training set.

All images are resized to $64 \times 64 \times 3$. For word representation, we adopt the state-of-the-art pre-trained character-based embedding on the visual descriptions, namely character-level ConvNet with a recurrent neural network (char-CNN-RNN), which outputs text embedding vectors of dimensionality 1,024. In our training, we sample n out of 10 sentences and use the average text embedding of the sampled sentences. We use $n = 4$ determined empirically.

As shown in Figure 4.2, both the generator and discriminator are deep convolutional neural networks. The text embedding is projected to a 128-dimensional vector by linear transformation. In the generator, the input is formed by concatenating this text embedding vector with a 100-dimensional noise sampled from unit normal distribution. In the discriminator, the projected text-embedding vector is depth concatenated with the final convolutional feature map.

All models are trained with mini-batch stochastic gradient descent with a mini-batch size of 64. We adopt the ADAM optimizer [73] with momentum 0.5 and learning rate 0.0002 as used in Radford et al. [21]. We use for Equation (4.2) the hyperparameter $\lambda = 10^{-6}$ to balance contextual and perceptual losses.

In the experiments, we evaluate our models based on human and machine judgments. For human judgment, we qualitatively evaluate the generated synthetic images given query text descriptions by visual inspection. For machine judgment, we input the generated image to an object classifier to assess quantitatively the realism of the synthetic images.

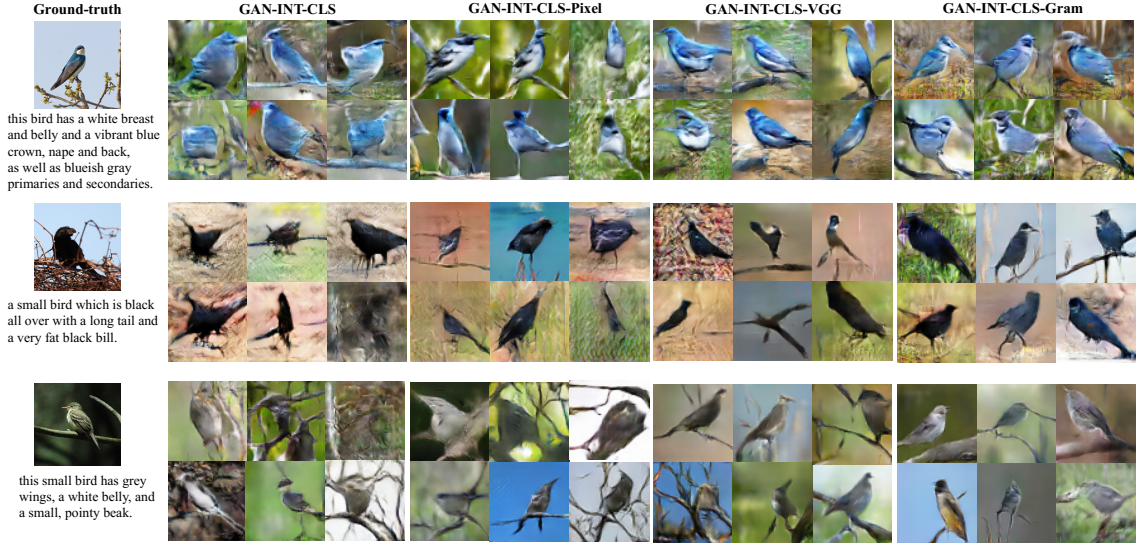


Figure 4.3: Generated bird images by conditioning on text from unseen test categories using the baseline GAN-INT-CLS [20] and the baseline with three different perceptual loss functions: pixel (GAN-INT-CLS-Pixel), activation (GAN-INT-CLS-VGG), and texture (GAN-INT-CLS-Gram) reconstruction losses.

4.3.1 Qualitative evaluation

Figure 4.3 shows qualitative examples comparing our results with the results of baseline method (GAN-INT-CLS) by Reed et al. [20] for the CUB dataset. The generated bird images are conditioned on text from unseen test categories. Our three methods are the baseline combined with pixel (GAN-INT-CLS-Pixel), activation (GAN-INT-CLS-VGG), and texture (GAN-INT-CLS-Gram) reconstruction losses. In these results, it is clear that GAN-INT-CLS is aware of the semantic relatedness between images and text. However, the bird images by GAN-INT-CLS generally lack structural definition. The baseline method with pixel reconstruction loss (GAN-INT-CLS-Pixel) yields slightly better bird structures. We find that GAN-INT-CLS-VGG and GAN-INT-CLS-Gram yield the perceptually and contextually most compelling results. We speculate that the improvements stem from the feature maps that focus on the object-specific structures while leaving the contextual loss focusing on semantic relatedness.

In Figure 4.4, we present results on the Oxford-102 flower dataset using the baseline and our methods. As shown, all four methods have generated plausible looking flower

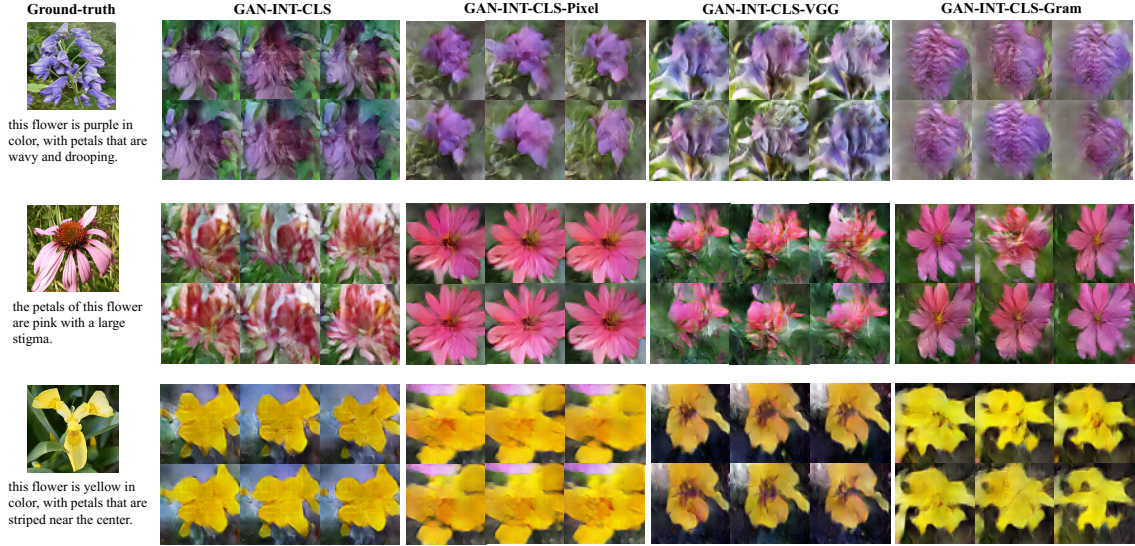


Figure 4.4: Generated flower images by conditioning on text from unseen test categories using the baseline GAN-INT-CLS [20] and the baseline with three different perceptual loss functions: pixel (GAN-INT-CLS-Pixel), activation (GAN-INT-CLS-VGG), and texture (GAN-INT-CLS-Gram) reconstruction losses.

images according to the query text. The generated flowers look like flowers from different classes. This is due to one-to-many mapping where the same text can be used to describe many different classes. In addition, the descriptions provided by human may not have been descriptive enough to distinguish between different types of flowers.

4.3.2 Quantitative evaluation

For quantitative evaluation, we utilize GoogLeNet [74] to evaluate how well the generated images are correctly classified as birds or flowers. We use the pretrained GoogLeNet model from `inception-v3.torch`.¹ For the evaluation, we input three query texts: “the bright blue bird has a white colored belly,” “this bird is yellowish orange with black wings,” and “this vibrant red bird has a pointed black beak.” For each text query, we generate 500 synthetic bird images and input the total 1,500 such images to GoogLeNet to do bird classification. Since there are many different types of birds from 1,000 object classes of

¹<https://github.com/Moodstocks/inception-v3.torch>

Table 4.1: GoogLeNet bird and flower classification accuracies

	Bird	Flower
GAN-INT-CLS	0.76	0.66
GAN-INT-CLS-Pixel	0.83	0.66
GAN-INT-CLS-VGG	0.80	0.76
GAN-INT-CLS-Gram	0.85	0.89

the GoogLeNet output, we use a world list of birds² to make a conglomerate bird classifier, such that all types of birds are classified as a bird.

We perform similar tasks for the flower dataset and evaluate the performance of flower classification. We use query texts, “this flower has white petals and a yellow stamen,” “the center is yellow surrounded by wavy dark purple petals,” and “this flower has lots of small round pink petals.” A comprehensive list of flowers is obtained from Lyons [75].

Table 4.1 presents the GoogLeNet bird and flower classification results on images generated by GAN-INT-CLS, GAN-INT-CLS-Pixel, GAN-INT-CLS-VGG, and GAN-INT-CLS-Gram. We observe the trend that more images are correctly classified as birds and flowers when perceptual loss functions are used. The quantitative evaluation confirms the qualitative improvements observed in Figure 4.3 and Figure 4.4, showing a combination of contextual and perceptual loss terms generates better images for both human perception and machine classification. We notice that for the flower case our results using GAN-INT-CLS-Gram are significantly better than other methods. This can be explained by observing that there are non-flower objects such as hen-of-the-woods and earthstar, which have subtle differences in styles as compared to flowers that humans can pick up. However, without incorporating a proper perceptual loss in the generator, GAN will not be able to synthesize flowers with distinctive features.

4.4 Summary

In this chapter, we have described GAN-based text-to-image synthesis methods that use both contextual and perceptual losses. The contextual loss in existing GAN literature fo-

²<http://ces.iisc.ernet.in/hpg/envis/sibleydoc63.html>

cuses on semantic relatedness between text and image, whereas the proposed perceptual loss focuses on the object-specific structure. Our results on CUB and Oxford-102 datasets suggest that adopting perceptual loss functions is helpful for improving visual realism in text-to-image synthesis.

Chapter 5

Improving the Diversity of Generated Images with Semantic Relevance

5.1 Introduction

This chapter further develops a text-to-image synthesis model built on conditional GAN (CGAN) by Mirza & Osindero [19]. The task of translating a short text description into an image has drawn much interest, attempted in a number of GAN approaches [20, 22, 25, 35, 76, 77]. Conditioning both the generator and discriminator on a text description, these approaches are capable of creating realistic images that correspond to the text description given.

Despite its success, GAN is known to suffer from a *mode collapse* problem in which generated images lack diversity and fall largely into a few trends. One way to mitigate mode collapse is to encourage a bijective mapping between the generated image output and the input latent space [78–80]. In Odena et al. [34], auxiliary classifier GAN (AC-GAN) is tasked to recover side information about the generated image such as a class label. The extra task is shown to promote the bijective mapping and discourages different input latent code from generating the same image output. Dash et al. [35] describe text conditioned auxiliary classifier GAN (TAC-GAN). During the training, the auxiliary classifier in TAC-GAN predicts the class of a generated image. The predicted class is compared against the ground-truth class of a training image whose text description is applied as the input.

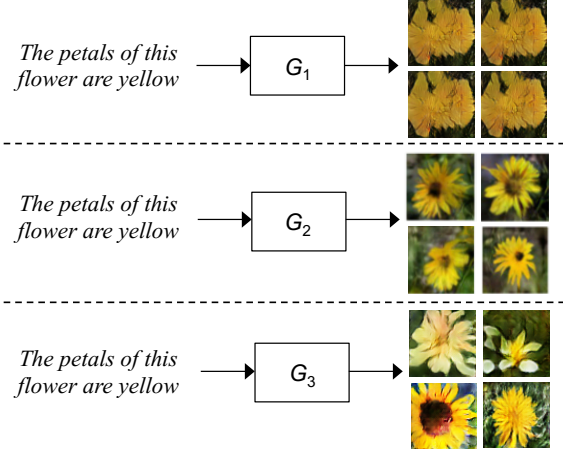


Figure 5.1: An illustration of sample diversity in text-to-image synthesis. Top: poor diversity (generated images are nearly identical suffering from mode collapse); middle: modest diversity (generated images belong to a single class); bottom: good diversity (generated images are not limited to classes)

Text is an indiscriminate input choice for CGAN (e.g., compared to class label). The implicit binding of text input to the (predicted) class label output in TAC-GAN is a source of ambiguity because some text can rightfully describe different class of images. Figure 5.1 illustrates diversity in generated samples. Given a text input, the baseline G_1 collapses to synthesizing images of the same trend. When the discriminator is forced to recover a class label of input text, the generator is implicitly bounded to synthesize a single image class as in G_2 , even if descriptive text is suitable for many different classes of flowers. We are interested in synthesizing images as in G_3 that are diverse, realistic, and relevant to the input text regardless of class.

Instead of class prediction, we modify the discriminator to regress semantic relevance between the two modalities of data, text and image. Similar to the AC-GAN and CGAN text-to-image synthesis, we explore the benefit of an additional supervised task in the discriminator. We train the discriminator with an extra regression task to estimate semantic correctness measure, a fractional value ranging between 0 and 1, with a higher value reflecting more semantic relevance between the image and text. We find training with the extra regression beneficial for the generator diversifying its generated examples, alleviating the mode collapse problem.

To support the learning through semantic correctness, we devise a training method that selects positive and negative examples. Unlike existing approaches that select a random image outside its class as a negative example, we distinguish easy and hard negatives measured in the semantic space of image’s text embedding. The idea of selecting negatives for text-image data based on some distance metric is not new, as it has been explored for image-text matching tasks [81]. We gradually decrease semantic distance between reference text and its negative image. The idea of progressively increasing semantic difficulty is related to curriculum learning [82] that introduces gradually more complex concepts instead of randomly presenting training data. Curriculum learning has been successfully applied to GAN in several domains. Subramanian et al. [83] and Press et al. [84] use curriculum learning for text generation from gradually increasing the length of character sequences in text as the training progresses. Karras et al. [85] apply curriculum learning on image generation by increasing the image resolution. We are the first to apply curriculum learning based on semantic difficulty for text-to-image synthesis. We validate empirically that our training method significantly improves the diversity of generated images and semantic correctness to the input text.

5.2 Comparison

This section reviews the GAN extensions that condition on or train an auxiliary supervised task with the side information. For text-to-image synthesis, we focus on methods by Reed et al. [20] and Dash et al. [35] as they are the closet schemes to ours. We describe our approach in contrast to these extensions.

5.2.1 Conditional GAN (CGAN)

Goodfellow et al. [18] suggest a possibility of conditional generative models. Mirza & Osindero [19] propose CGAN that makes the use of side information at both the generator G and discriminator D . A mathematical optimization for CGAN is given by

$$\min_G \max_D \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} [\log D(\mathbf{x}|\mathbf{y})] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}} [\log(1 - D(G(\mathbf{z}|\mathbf{y})))] \quad (5.1)$$

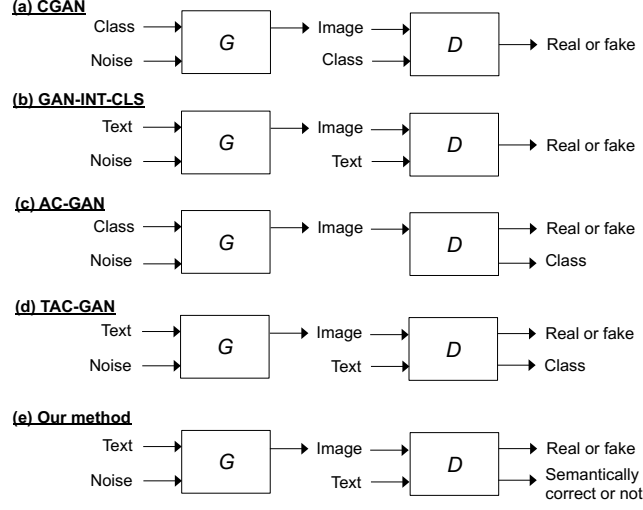


Figure 5.2: Architectural comparison of (a) CGAN [19], (b) matching-aware manifold-interpolated GAN [20], (c) auxiliary classifier GAN [34], (d) text conditioned auxiliary classifier GAN [35], and (e) our method.

where x is a data example, and the side information y can be a class label or data from another modality. Figure 5.2 (a) shows the structure of CGAN when a class label is used as the side information. GAN data generation requires to sample a random noise input z from the prior distribution p_z . This approach for multimodal CGAN is convenient and powerful for modalities that are typically observed together (e.g., audio-video, image-text annotation). For practical consideration, G takes in $z|y$ as a joint representation that (typically) concatenates z to y into a single vector. Similarly, joint representation can be formed for D .

5.2.2 Matching-aware Manifold-interpolated GAN (GAN-INT-CLS)

Reed et al. [20] propose GAN-INT-CLS for automatic synthesis of realistic images from text input. As in Figure 5.2 (b), GAN-INT-CLS can be viewed as a multimodal CGAN trained on text-annotated images. Text features are pre-computed from a character-level recurrent neural net and fed to both G and D as side information for an image. In G , the text feature vector is concatenated with a noise z and propagated through stages of fractional-strided convolution processing. In D , the generated or real image is processed

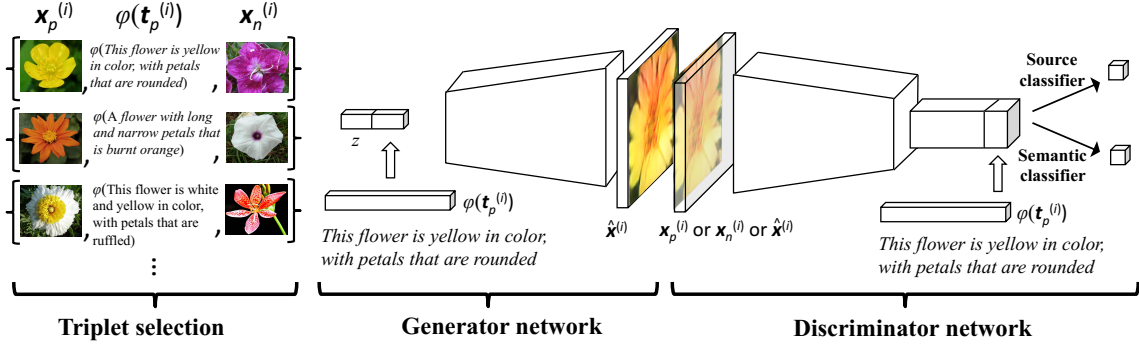


Figure 5.3: Overall pipeline of Text-SeGAN. From training data, we first form mini-batches of N triplet examples $\{\mathbf{x}_p^{(i)}, \varphi(\mathbf{t}_p^{(i)}), \mathbf{x}_n^{(i)}\}_{i=1}^N$. The formed triplets are used to train the generator and discriminator networks.

through layers of strided convolution before concatenated to the text feature for computing the final discriminator score.

5.2.3 Auxiliary Classifier GAN (AC-GAN)

AC-GAN [34] uses an extra discriminative task at D . For example, the AC-GAN D could perform class label prediction for a generated example from G . That is, instead of supplying the class label side information as an input to D , AC-GAN trains an additional label classifier at D for input samples. The structure of AC-GAN is illustrated in Figure 5.2 (c). The log-likelihood functions of the AC-GAN D consist of

$$L_s = \mathbb{E}[\log p(s = \text{"real"}|\mathbf{x})] + \mathbb{E}[\log p(s = \text{"fake"}|\hat{\mathbf{x}})] \quad (5.2)$$

$$L_c = \mathbb{E}[\log p(C = c|\mathbf{x})] + \mathbb{E}[\log p(C = c|\hat{\mathbf{x}})] \quad (5.3)$$

Here, $p(s = \text{"real"}|\mathbf{x})$ and $p(s = \text{"fake"}|\hat{\mathbf{x}})$ are the source (s) probability distributions given a real input $\mathbf{x}$ or generated input $\hat{\mathbf{x}}$. Similarly, $p(C = c|\mathbf{x})$ and $p(C = c|\hat{\mathbf{x}})$ are the class probability distributions over the labels. During the training, D maximizes $L_s + L_c$ while G maximizes $L_c - L_s$.

5.2.4 Text Conditioned Auxiliary Classifier GAN (TAC-GAN)

Simply put, TAC-GAN [35] combines AC-GAN and GAN-INT-CLS. Figure 5.2 (d) shows the structure of TAC-GAN. As in GAN-INT-CLS, TAC-GAN performs image synthesis on the text embedding $\varphi(\mathbf{t})$ with a sampled noise input. Following AC-GAN, the TAC-GAN D performs source and label classifications of the input image. The log-likelihood functions of the source classification and the label classification are given by

$$L_s = \mathbb{E}[\log p(s = \text{“real”} | \mathbf{x}, \varphi(\mathbf{t}))] + \mathbb{E}[\log p(s = \text{“fake”} | \hat{\mathbf{x}}, \varphi(\mathbf{t}))] \quad (5.4)$$

$$L_c = \mathbb{E}[\log p(C = c | \mathbf{x}, \varphi(\mathbf{t}))] + \mathbb{E}[\log p(C = c | \hat{\mathbf{x}}, \varphi(\mathbf{t}))] \quad (5.5)$$

The goal for D is to maximize $L_s + L_c$, and $L_c - L_s$ for G .

The label prediction in D aims to correctly recover the class label of its input. If an input is real, its ground-truth label is available in the training examples. If the input is fake (i.e., generated), its class label is set with the same label as the class of the image associated with the text input to G used to generate the fake image. Thus, G will be penalized if its generated image does not correspond to the class of the image associated with its text input.

All architectures in Figure 5.2 take in the side information input. While CGAN and GAN-INT-CLS discriminators are single-task, AC-GAN, TAC-GAN, and our approach have an extra discriminative task. Noticeably, our approach does not rely on class prediction of generated images. This is because there could be a potential one-to-many mapping problem where one text description broadly covers multiple classes of images. As a result, it may cause adverse effect on the generator training. Instead, we wish to weigh in whether or not the input (text description) explains the generated image correctly as shown in Figure 5.2 (e). In the next section, we describe our approach in detail.

5.3 Text-conditioned Semantic Classifier GAN

5.3.1 Overview

We propose Text-conditioned Semantic Classifier GAN (Text-SeGAN), a variant of the TAC-GAN architecture and its training strategies. Figure 5.3 illustrates an overview of our approach. Unlike TAC-GAN, there is no class label prediction in the discriminator network. Using a text-image pair from the training dataset, we form a triplet $\{\mathbf{x}_p, \varphi(\mathbf{t}_p), \mathbf{x}_n\}$. We denote $\mathbf{x}_p$ a positive image that corresponds to the description of encoded text $\varphi(\mathbf{t}_p)$. On the contrary, a negative image $\mathbf{x}_n$ is selected not to correspond to $\varphi(\mathbf{t}_p)$. Instead of randomly sampling a negative image, we introduce various training strategies that select a negative image based on a semantic distance to its encoded text. Our algorithm for selecting negative images will be explained in detail. We train the generator and discriminator networks using the triplets. By conditioning on the encoded positive text $\varphi(\mathbf{t}_p)$, our generator synthesizes an image $\hat{\mathbf{x}}$ (fake). Taking $\mathbf{x}_p$, $\mathbf{x}_n$, or $\hat{\mathbf{x}}$ as an input conditioned on $\varphi(\mathbf{t}_p)$, the discriminator predicts a source of the input image and semantic relevance to $\varphi(\mathbf{t}_p)$.

5.3.2 Training objectives

We train the generator and the discriminator by mini-batch stochastic gradient ascent on their objective functions. Similar to AC-GAN and TAC-GAN, the log-likelihood objective L_s is used to evaluate whether or not a (source) image applied to the discriminator is real or fake

$$L_s = \mathbb{E}[\log p(s = \text{“real”} | \mathbf{x}, \varphi(\mathbf{t}))] + \mathbb{E}[\log p(s = \text{“fake”} | \hat{\mathbf{x}}, \varphi(\mathbf{t}))] \quad (5.6)$$

The additional task at our discriminator is to determine how well the applied image matches the text encoding. In other words, the discriminator is tasked to predict semantic relevance between the applied image and text. The log-likelihood objective L_r for semantic

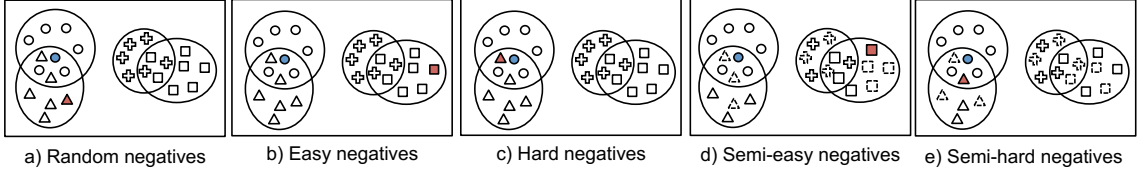


Figure 5.4: Sampling negative images. Different shapes indicate different classes. Images are placed in the semantic space (reduced to 2D for illustrative purposes) where the Euclidean distance is used to indicate separation between points. The selection methods are (a) random negatives, (b) easy negatives, (c) hard negatives, (d) semi-easy negatives, and (e) semi-hard negatives.

relevance matching is

$$L_r = \mathbb{E}[\log p(r = \text{“match”} | \mathbf{x}, \varphi(\mathbf{t}))] + \mathbb{E}[\log p(r = \text{“match”} | \hat{\mathbf{x}}, \varphi(\mathbf{t}))] \quad (5.7)$$

Regardless of the applied image source (real or fake), we want to make sure that the image matches the text description. Using the likelihoods L_s and L_r , we describe the training objectives for our discriminator and generator. For training the discriminator, we maximize $L_s + L_r$. For training the generator, however, we want to maximize L_r while minimizing L_s . For realistic fakes, $\mathbb{E}[\log p(s = \text{“fake”} | \hat{\mathbf{x}}, \varphi(\mathbf{t}))]$ should be low. Hence, we maximize $L_r - L_s$ for G .

5.3.3 Negative sampling

Optimizing over all possible triplets is computationally infeasible. To form our triplet $\{\mathbf{x}_p, \varphi(\mathbf{t}_p), \mathbf{x}_n\}$, we perform class negative sampling. A positive example is an image with matching text description and a class label drawn from the training dataset. Our class negative sampling looks for an example outside the positive class. Additionally, we use the Euclidean distance between the text embeddings of the reference positive and negative examples.

Recall that our discriminator takes in image and text modalities. The discriminator is applied with the following pairs formed from each triplet.

1. Real image with matching text: any reference example as-is from the training dataset

is this kind;

2. Real image with non-matching text: any example from negative sampling paired with the reference text description results in this kind;
3. Fake image with matching text: any image generated using the reference text description results in this kind.

We now discuss methods to select negatives for a given reference (positive) example. Existing text-to-image synthesis approaches [20, 22, 35, 76, 77] select a random image outside the class of the positive example. According to Reed et al. [20], negative images are real images that do not match the reference text description. We note that negative samples could have partial or complete matching to the reference text but a different label from the reference example. Suppose a reference image $\mathbf{x}^{(i)}$ with a label $l^{(i)}$ has an encoded text description $\varphi(\mathbf{t}^{(i)})$. We define various types of negative images and evaluate their effects on text-to-image synthesis. Figure 5.4 illustrates different negative sampling schemes in the semantic metric space of encoded text. Samples for different classes are denoted by different shapes. Blue circle indicates a reference sample, and one of its negative samples is shown in red.

Random negatives. As used in other approaches [20, 22, 35, 76, 77], choose any image outside the class of the reference image as a random negative. Given a positive example $\mathbf{x}^{(i)}$, choose

$$\mathbf{x}^{(j)} \text{ s.t. } l^{(i)} \neq l^{(j)}, \quad (5.8)$$

where $l^{(i)}$ is the class label of $\mathbf{x}^{(i)}$, and $l^{(j)}$ for that of the random negative $\mathbf{x}^{(j)}$. Figure 5.4 (a) illustrates a random negative example (in red triangle).

Easy negatives. We use encoded text vectors to measure semantic similarity between images. For easy negative, we find an image that belongs to the outer class of the reference class and has its corresponding text vector farthest from the reference text. We select

$$\mathbf{x}^{(j)} \text{ s.t. } l^{(i)} \neq l^{(j)} \text{ and } \max_j \|\varphi(\mathbf{t}^{(i)}) - \varphi(\mathbf{t}^{(j)})\|_2^2 \quad \forall j \quad (5.9)$$

In Figure 5.4 (b), we note that the red square is the farthest outer class sample from the

reference positive image.

Hard negatives. As denoted with red triangle in Figure 5.4 (c), the hard negative corresponds to an image that belongs to the outer class of the reference image and has its text vector closest to the encoded reference text. Thus, we select

$$\mathbf{x}^{(j)} \text{ s.t. } l^{(i)} \neq l^{(j)} \text{ and } \min_j \|\varphi(\mathbf{t}^{(i)}) - \varphi(\mathbf{t}^{(j)})\|_2^2 \quad \forall j \neq i \quad (5.10)$$

Semi-easy negatives. Selecting easiest negatives in practice leads to a poorly trained discriminator. To mitigate the problem, it helps to select a negative

$$\mathbf{x}^{(j)} \text{ s.t. } l^{(i)} \neq l^{(j)} \text{ and } \|\varphi(\mathbf{t}^{(i)}) - \varphi(\mathbf{t}^{(j)})\|_2^2 > \alpha \quad (5.11)$$

for some α . In practice, we randomly select M samples from outer classes and apply Equation (5.9) among the samples in the outer classes. We call these negative samples semi-easy negatives. In Figure 5.4 (d), dotted lines indicate samples not included in M outer samples. Among the selected outer samples, the red square represents an easy negative sample.

Semi-hard negatives. It is crucial to select hard negatives that can contribute to improving the semantic discriminator. However, selecting only the hardest negatives can result in a collapsed model. Therefore, we apply Equation (5.10) in randomly selected M outer samples. We call these negative examples semi-hard negatives. In Figure 5.4 (e), a semi-hard negative is depicted as the red triangle.

Easy-to-hard negatives. In early training, hard negatives that have similar features to the reference example may remove relevant features in representing the positive sample, leading to mode collapse. As a systematic way to provide negative examples of incremental semantic difficulty, we use curriculum learning [82] by gradually increasing the semantic similarity between the input encoded text and negative image. We use the following method for easy-to-hard negative selection.

1. Randomly select negative text from M outer samples;
2. Generate a histogram of cosine similarity values between positive and M negative text;

3. Select 100β -th percentile of the histogram ($0 < \beta \leq 1$);
4. Increase β gradually.

Low β induces the selection of easy negatives, whereas high β leads to hard negatives. We sample negative training example from a distribution and continue a sequence of sampling of the distribution which gradually gives more weight β to the more semantically difficult negatives, until all examples have equal weight of 1 ($\beta = 1$).

5.4 Experiments

We evaluate our models using Oxford-102 flower dataset [71]. We use the same settings described in Section 4.3.

5.4.1 Implementation details

As shown in Figure 5.3, both the generator and discriminator are implemented as deep convolutional neural nets. We build our GAN models based on the GAN-INT-CLS¹ architecture. We perform dimensionality reduction on the 1,024-dimensional text embedding vectors using a linear projection onto 128 dimensions. The generator input is formed by concatenating the reduced text vector with a 100-dimensional noise sampled from a unit normal distribution. In the discriminator, the text vector is depth-concatenated with the final convolutional feature map. We add an auxiliary classification task at the last layer of the discriminator. The auxiliary task predicts a semantic relevance measure between the input text and image.

In the model training, we perform mini-batch stochastic gradient ascent with a batch size $N = 64$ for 600 epochs. We use the ADAM optimizer [73] with a momentum of 0.5 and a learning rate of 0.0002 as suggested by Radford et al. [21]. We use number of outer samples $M = 1000$. We increase β from 0.6 to 1 by 0.1 for every 100 epoch. We stay with max β once it is reached.

¹<https://github.com/reedscot/icml2016>

Table 5.1: Inception scores of the generated images using random negative sampling

	Inception score
GAN-INT-CLS [20]	2.66 ± 0.03
StackGAN [22]	3.20 ± 0.01
TAC-GAN [35]	3.45 ± 0.05
HDGAN [76]	3.45 ± 0.07
Text-SeGAN	3.65 ± 0.06

5.4.2 Evaluation metrics

We evaluate Text-SeGAN both qualitatively and quantitatively. In our quantitative analysis, we compute the inception score [86, 87] and the multi-scale structural similarity (MS-SSIM) metric [88] for comparative evaluation against other models. The inception score measures whether or not a generated image contains a distinctive class of objects. It also measures how diverse classes of images are produced. The analytical inception score is given by

$$IS(G) = \exp(\mathbb{E}_{\mathbf{x}} D_{KL}(p(y|\mathbf{x})||p(y))) \quad (5.12)$$

where $\mathbf{x}$ is a generated image by G , and y indicates the label predicted by the pre-trained inception model [87]. $p(y|\mathbf{x})$ is the conditional class distribution and $p(y)$ is the marginal class distribution. Images that contain distinctive objects will have the conditional class distribution with low entropy. G that outputs diverse class of images will have a marginal class distribution with a high entropy value. Therefore, high KL divergence between the two distributions that leads to high $IS(G)$ is desirable. As suggested by Salimans et al. [86], we evaluate the metric on 30k generated images for each generative model.

Additionally, we use the MS-SSIM metric to measure interclass diversity of the generated images. In image processing, MS-SSIM is used to indicate similarity of two images in terms of luminance, contrast, and structure. Its use in GAN is primarily for measuring dissimilarity (i.e., diversity) of the generated images [34, 35, 76]. A low MS-SSIM value indicates higher diversity or a less likelihood of mode collapsing. Following Odena et al. [34], we sample 400 image pairs randomly within a training class and report their MS-SSIM scores.

Table 5.2: Inception scores of the generated images from Text-SeGAN using various negative sampling schemes

	Inception score
Hard negatives	3.33 ± 0.03
Semi-easy negatives	3.69 ± 0.04
Semi-hard negatives	3.70 ± 0.04
Easy-to-hard negatives	4.03 ± 0.07

5.4.3 Quantitative analysis

We first evaluate the effect of architectural variations using random negative sampling. We compare Text-SeGAN with GAN-INT-CLS [20] and TAC-GAN [35] as they are the closest schemes to ours. Table 5.1 shows the inception scores for GAN-INT-CLS, TAC-GAN, and Text-SeGAN using the random negative sampling scheme. For broader comparison, we also include the results by StackGAN [22] and HDGAN [76]. The primary goal of StackGAN and HDGAN is to enhance the resolution of generated images. We achieve a significant improvement over GAN-INT-CLS and competitive results against StackGAN, HDGAN, and TAC-GAN despite the difference in image sizes. Note that our generated images ($64 \times 64 \times 3$) are half the size of TAC-GAN ($128 \times 128 \times 3$) and a quarter of HDGAN ($256 \times 256 \times 3$) and StackGAN ($256 \times 256 \times 3$) images. Text-SeGAN improves inception scores of GAN-INT-CLS by 0.99, StackGAN by 0.45, and HDGAN and TAC-GAN by 0.2. It is known that images with higher resolution generally improve discriminability [34]. Our improvement in the inception score is significant considering a relatively small size of the generated images.

Next, we evaluate the effects of different triplet selection schemes on the proposed architecture. Table 5.2 compares the inception scores for Text-SeGAN using hard, semi-easy, semi-hard, and easy-to-hard negative selection schemes. Easy negative selection turns out choosing negative examples that have a little effect in training of our model. Under easy negative selection, generated images match the text description unreliably, and no visible improvement to the model could be observed. Therefore, we omit reporting its results here. Finding the hardest or the easiest negative of a positive sample evidently results in deterministic pairing. We suspect that a lack of variety in triplets may have led to poor performance in easy and hard negatives. In practice, mislabelled and poorly captured images

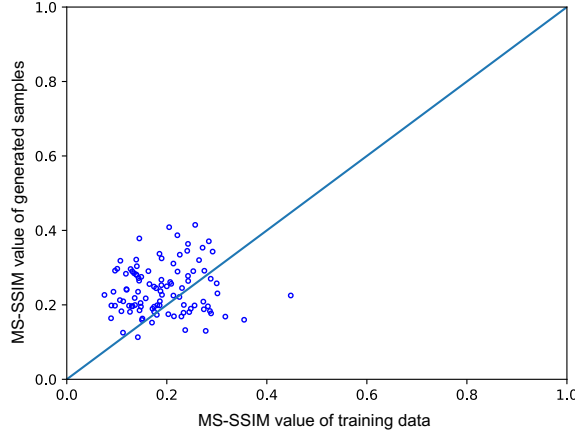


Figure 5.5: Comparison of the class-wise MS-SSIM scores of the samples from the training data and the generated samples of Text-SeGAN using easy-to-hard negative sampling.

would dominate the easy and hard negatives. Semi-easy negatives have similar performance to random negatives. Semi-hard negatives further increase the inception score by introducing diverse triplets that contribute more to improving the model. Among various negative sampling schemes, easy-to-hard negative sampling achieves the best performance. This result suggests that training with triplets of gradually more difficult semantics can benefit text-to-image synthesis.

As in Dash et al. [35], we use MS-SSIM to validate that our generated images are as diverse as training data. Figure 5.5 compares the mean MS-SSIM for each class in the training dataset compared to that of the generated images by our best scheme (Text-SeGAN with easy-to-hard negative sampling). Each point represents a class and can be interpreted as how similar the images of the same class are one another in the two datasets. Our model produces as diverse images as the real images in the training dataset.

5.4.4 Qualitative analysis

Figure 5.6 shows the generated images from the three text-conditioned models, GAN-INT-CLS ($64 \times 64 \times 3$), TAC-GAN ($128 \times 128 \times 3$), and Text-SeGAN ($64 \times 64 \times 3$) using easy-to-hard negative sampling. At first glance, all images seem reasonable, matching the whole or part of the text descriptions. Despite the semantic relevance and visual realism, we

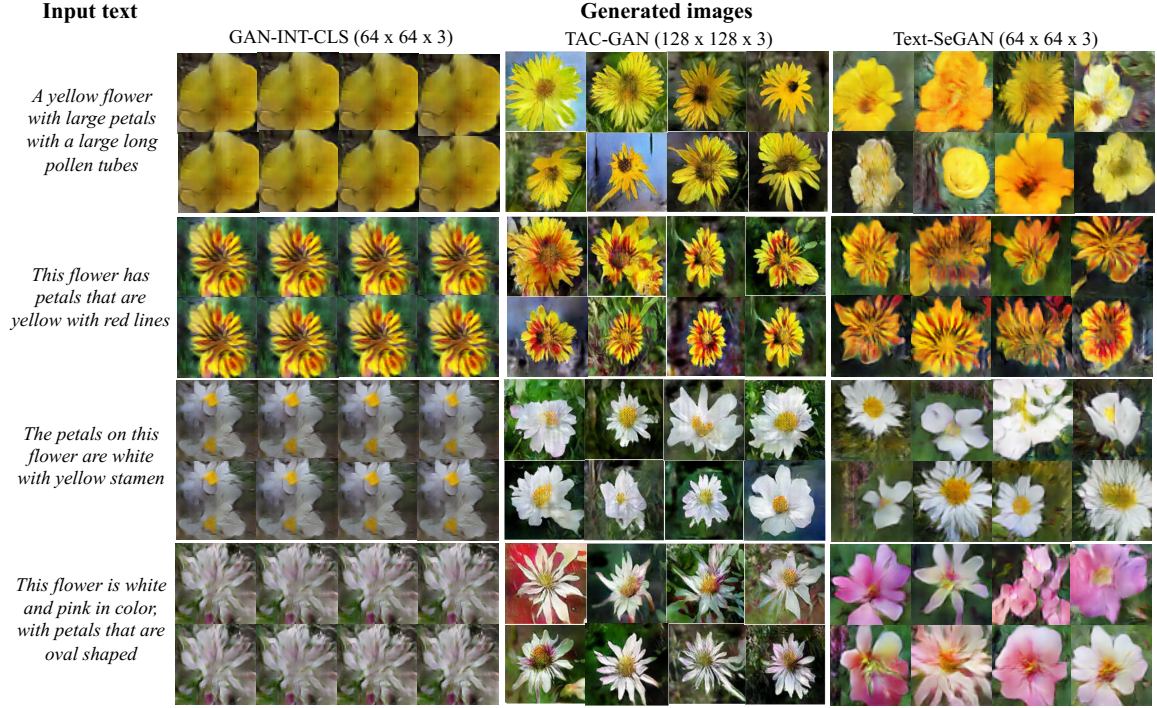


Figure 5.6: Flower images generated by GAN-INT-CLS, TAC-GAN, and Text-SeGAN (with easy-to-hard negatives).

notice that GAN-INT-CLS collapses to nearly an identical image output. In other words, different latent codes are mapped to a few (or singular) output trends. TAC-GAN avoids such collapse, but the generated images tend to belong to a single class. Adopting an auxiliary class prediction suppresses different latent codes being mapped to the same output, but enforcing the generated image classes to match the class of the input text in TAC-GAN has restricted generating images of diverse classes. Since the goal of text-to-image synthesis is simply generating images that match the input text, we modify the auxiliary classifier to measure semantic relevance instead of reinforcing the class label of the text description attached to training images. In addition, we gradually introduce semantically more difficult triplets rather than in a random order during training. With such modifications, we observe that the generated flowers have more variations in their shapes and colors (in spite of their relative small size of $64 \times 64 \times 3$) while matching the text description. For example, given an input text, “This flower is white and pink in color, with petals that are oval shaped,” our approach generates diversely shaped flowers with both pink and white petals.

5.5 Conclusion

We present a new architecture and training strategies for text-to-image synthesis that can improve diversity of generated images. Discriminator in existing AC-GAN is tasked to predict class label along with source classification (real or fake). Due to one-to-many mapping (the same text can describe images of different classes) in text-to-image synthesis, feeding scores on class prediction to generator can potentially bound the generator to produce images of limited number of classes. In order to mitigate this, we introduce a variant of AC-GAN whose discriminator measures semantic relevance between image and text instead of class prediction. We also provide several strategies of selecting training triplet examples. Instead of randomly presenting the triplets during training, we introduce gradually more complex triplets in their semantics. Experiment results on Oxford-102 flower dataset demonstrate that our model with easy-to-hard negative selection scheme can generate diverse images that are semantically relevant to the input text and significantly improves inception score compared to existing state-of-the-art methods.

Chapter 6

Adversarial Text-to-Image Synthesis by Semantic Sampling

6.1 Introduction

The task of generating an image from natural language (e.g., human-written text description) has drawn much attention in recent machine learning research. A popular approach is to train generative adversarial nets (GANs). In particular, conditional GAN (CGAN) formulations [20, 22, 23] have been successful to capture the bi-modality between image and text. A CGAN-based approach can use word embeddings to transform a text input into a vector representation, on which the generator and discriminator networks are conditioned as the side information. On the other hand, deep convolutional GAN (DCGAN) [21] has provided an architecturally stable topology based on deep CNNs that enable learning to synthesize a specific image through adversarial training.

Word embeddings via deep CNN or RNN yield discernible and generalizable vector representations for text. In zero-shot caption-based retrieval, the use of such embeddings is found to achieve the state-of-the-art result on the Oxford-102 flower database [71, 72], thereby hinting a learnable mapping to image pixels from text characters. Yet, a real difficulty, according to Reed et al. [72], is a highly multimodal nature of the problem that there are many images correctly described by a text input and vice versa.

In this chapter, we formulate the task of generating *diverse* images from text as a prob-

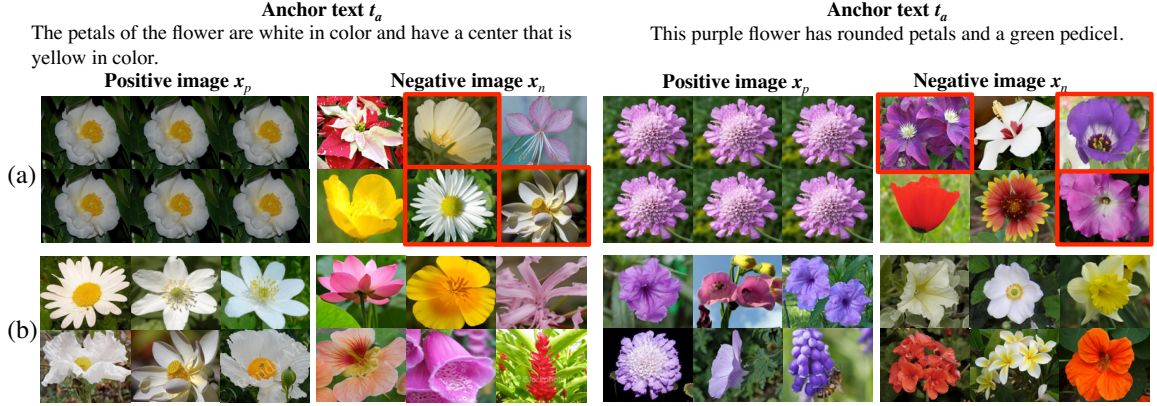


Figure 6.1: Sampled positive and negative images using (a) baseline and (b) our sampling scheme given an anchor text from Oxford-102.

lem of recovery under aliasing with incomplete sampling. For a text description, we notice that there could be multiple images that match the text equally well (“one-to-many”); see, e.g., Figure 6.1 (b). In an analogy to the classical aliasing theory in signal processing, we call these image aliases of the text, and observing images matched by the same text is equivalent to a signal sampling process. Since a single image or just a few of them are hardly the sole realization of a text description, we would have to sample many images in order to characterize the text. Similarly, there could be multiple texts which describe the same image equally well (“many-to-one”); e.g., Reed et al. [72] collect ten sentences to describe an image. In this case, we would have to examine many texts to best characterize the image.

Due to scarcity of labeled training data, our problem is related to learning from positive and unlabeled examples. Liu et al. [89, 90] have studied the problem for text classification where there are no negatively labeled examples. They propose a two-step approach that first identifies a set of likely negative documents from unlabeled set and applies the EM algorithm or an SVM to build a classifier iteratively. Scholkopf et al. [91] propose one-class SVM that uses only positive data to build an SVM classifier. However, the method is known highly sensitive to representation and dimensionality of input [92]. Our approach unveils images aliased to each text description and also text descriptions aliased to each image via positive and negative sampling. We have devised such a *semantic sampling* method that works in a high-dimensional space of embedded text inputs. Our CGAN model learns

to synthesize diverse images by training with the text-image pairs formed by semantic sampling.

This chapter makes the following contributions:

- Demonstration of how positive and negative sampling can benefit from aliasing images and text, respectively, in mitigating mode collapsing (Section 6.2.2). In signal processing, an aliasing problem has been regarded as a negative effect interfering with information recovery. In this sense, in adversarial text-to-image synthesis, image and text aliasing is a blessing.
- Justifying the use of negative sampling in the text embedding space for the scenario where training data does not provide non-matching image-text pairs, based on a general optimization theory about building classifiers without negative training data (Section 6.2.3).
- Achieving high inception scores with semantic sampling. Generated images by our sampling scheme lead to higher diversity as measured with inception scores compared to images generated by the previous work [20, 22, 23, 25, 29] (Table 6.2) and are as diverse as images in the training set (Figure 6.6).
- Application of the proposed sampling on other GAN-based text-to-image synthesis models. We extend our sampling scheme to other models and show that our semantic sampling increases image diversity (Figure 6.7).

6.2 Method

6.2.1 Overview

We propose an unsupervised scheme for sampling training examples that leads to generation of diverse images on a text input. Our scheme is unsupervised in the sense that we do not consider class labels of images when sampling. Given encoded anchor text $\varphi(\mathbf{t}_a)$ (via embedding function $\varphi(\cdot)$), we form a triplet $\{\mathbf{x}_p, \varphi(\mathbf{t}_a), \mathbf{x}_n\}$. We sample positive and negative images, $\mathbf{x}_p$ and $\mathbf{x}_n$, with respect to anchor text. A positive image is paired with

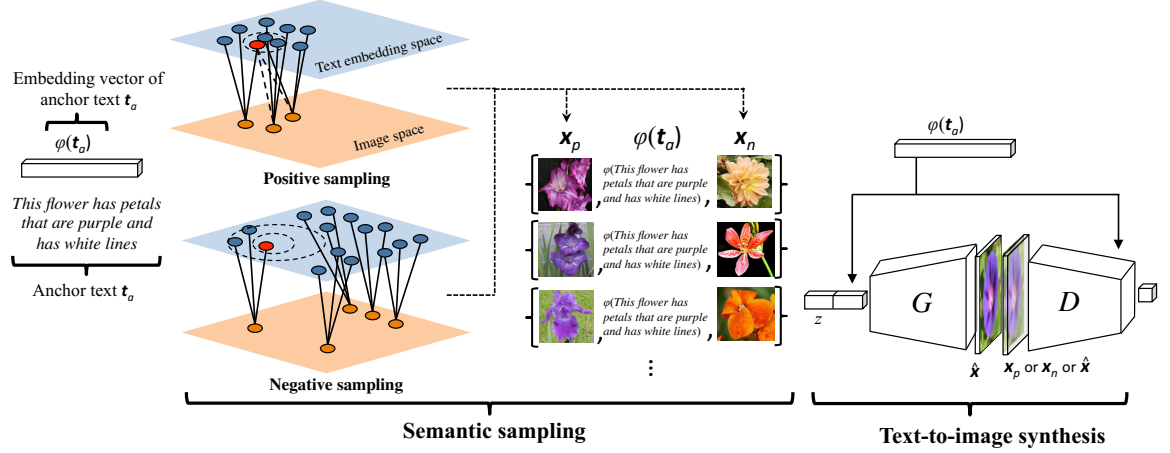


Figure 6.2: Overview of the proposed algorithm. Given a text embedding $\varphi(t_a)$ for an anchor text t_a , our algorithm first samples positive x_p and negative x_n images based on their semantic similarities to the encoded anchor text and forms triplets (Section 6.2.4). Then we train a text-to-image synthesis GAN using these triplets. Conditioning both generator G and discriminator D on $\varphi(t_a)$, G generates an image, and D discriminates whether or not its input $(x, \varphi(t))$ contains a real image x , and x and $\varphi(t_a)$ semantically match (Section 6.2.5).

text whose encoding has a high cosine similarity to encoded anchor text. Conversely, text paired with a negative image is dissimilar to the anchor.

Figure 6.2 illustrates an overview of our method. The triplets $\{x_p, \varphi(t_a), x_n\}$ formed by sampling are used to train the generator and discriminator nets. By conditioning on $\varphi(t_a)$, our generator synthesizes an image $\hat{x}$. Taking x_p , x_n , or $\hat{x}$ as an input, also conditioned on $\varphi(t_a)$, the discriminator predicts the nature (real or synthesized) of the image input and its semantic relevance (match or mismatch) to $\varphi(t_a)$.

In Figure 6.1, we compare positive and negative images selected by a baseline sampling scheme found in previous methods [20, 22, 23, 25, 29] against our semantic sampling. The baseline is a supervised method requiring class labels of images. That is, given an anchor text, the baseline selects the image paired with the anchor text as a positive example. Therefore, as in Figure 6.1 (a), the pool of positive images is identical given an anchor text. For negative sampling, the baseline looks for any image outside the class of a positive image. Such scheme can sample a semantically matching image as a negative image (highlighted in red). A goal of text-to-image synthesis is to generate diverse images that correspond to the input text regardless of class. As shown in Figure 6.1 (b), our scheme selects more di-

verse images than the baseline. In addition, our negative sampling scheme acts to mitigate positive images selected as negatives.

6.2.2 Intuition: sampling under aliasing

The intuition behind our approach can be explained with sampling under aliasing. When we observe one text description plausible for many images, we say that the images are aliased to the text. Similarly, when many text descriptions are valid for an image, the text descriptions are aliased to the image.

Using the notion of aliasing, we make the following two observations. (1) In positive sampling, *one-to-many* mappings in the text-to-image direction help increase diversity of synthesized images. That is, we can include all images aliased to given anchor text as positive images. This enlarges the set of positive examples and promotes diversity of synthesized images. (2) In negative sampling, *many-to-one* mappings in the text-to-image direction help mitigate the risk of incorrect selection of images. That is, we consider an image a negative sample only when all its aliasing text descriptions are far away from the anchor text. Also for positive sampling, many-to-one mappings are helpful to obtain an aggregate text representation by combining text aliases for an image (e.g., in this thesis, we take an average embedding for these text aliases).

We note that sampling methods insensitive to many-to-one or one-to-many scenarios tend to underperform for datasets that exhibit image or text aliasing. For example, under the assumption of one-to-one mapping only, images corresponding to nearby text descriptions would constitute distinct positive images for a given anchor text. This would not hold, however, without the one-to-one assumption because in this case these texts will likely describe just one or a few images. To resolve this issue, we would need to use our proposed semantic sampling approach to expand allowed positive samples.

6.2.3 Theory

In this section, we establish a theoretical foundation for training with datasets only containing matched pairs of image and text (i.e., no mismatch examples). We find our problem similar to partially supervised training of a classifier (i.e., discriminator). Let $y \in \{+1, -1\}$

be a label indicating whether or not an image $\mathbf{x}$ and a text embedding $\varphi(\mathbf{t})$ match. Text-to-image synthesis aims to design a generator G that estimates a conditional generative model $p(\mathbf{x}|\varphi(\mathbf{t}), y = +1)$ by training a discriminator D that computes $p(y|\mathbf{x}, \varphi(\mathbf{t}))$, the probability that $\mathbf{x}$ and $\varphi(\mathbf{t})$ match semantically. In order to train such discriminator, a training pair $(\mathbf{x}, \varphi(\mathbf{t}))$ should be sufficiently diverse and representative. In a practical scenario, the problem is solved in a partially supervised manner. That is, labeled negative pairs are usually unavailable and assumed to be mixed with positive pairs in an unlabeled dataset, and only a limited amount of labeled positive pairs is available.

To obtain insights into why learning is possible in partially supervised case, we rewrite the probability of error in D

$$\begin{aligned} p(D(\mathbf{x}, \varphi(\mathbf{t})) \neq y) &= p(D(\mathbf{x}, \varphi(\mathbf{t})) = +1, y = -1) \\ &\quad + p(D(\mathbf{x}, \varphi(\mathbf{t})) = -1, y = +1). \end{aligned} \quad (6.1)$$

We can rewrite $p(D(\mathbf{x}, \varphi(\mathbf{t})) = +1, y = -1)$ as

$$\begin{aligned} p(D(\mathbf{x}, \varphi(\mathbf{t})) = +1, y = -1) &= p(D(\mathbf{x}, \varphi(\mathbf{t})) = +1) \\ &\quad - [p(y = +1) - p(D(\mathbf{x}, \varphi(\mathbf{t})) = -1, y = +1)]. \end{aligned} \quad (6.2)$$

Substituting into Equation (6.1), we obtain

$$\begin{aligned} p(D(\mathbf{x}, \varphi(\mathbf{t})) \neq y) &= p(D(\mathbf{x}, \varphi(\mathbf{t})) = +1) - p(y = +1) \\ &\quad + 2p(D(\mathbf{x}, \varphi(\mathbf{t})) = -1|y = +1)p(y = +1). \end{aligned} \quad (6.3)$$

Our goal is to minimize the discriminator error $p(D(\mathbf{x}, \varphi(\mathbf{t})) \neq y)$. To this end, we may assume without loss of generality that $p(y = +1)$ is a constant. If we are able to hold $p(D(\mathbf{x}, \varphi(\mathbf{t})) = -1|y = +1)$ small, then learning is the same as minimizing $p(D(\mathbf{x}, \varphi(\mathbf{t})) = +1)$. Holding $p(D(\mathbf{x}, \varphi(\mathbf{t})) = -1|y = +1)$ small while minimizing $p(D(\mathbf{x}, \varphi(\mathbf{t})) = +1)$ is the same as minimizing $p_U(D(\mathbf{x}, \varphi(\mathbf{t})) = +1)$ while holding $p_P(D(\mathbf{x}, \varphi(\mathbf{t})) = +1) \geq r$, where r is a recall. Once we fix r , our remaining control in sampling is to make $p_U(D(\mathbf{x}, \varphi(\mathbf{t})) = +1)$ small. This optimization shows that to train a

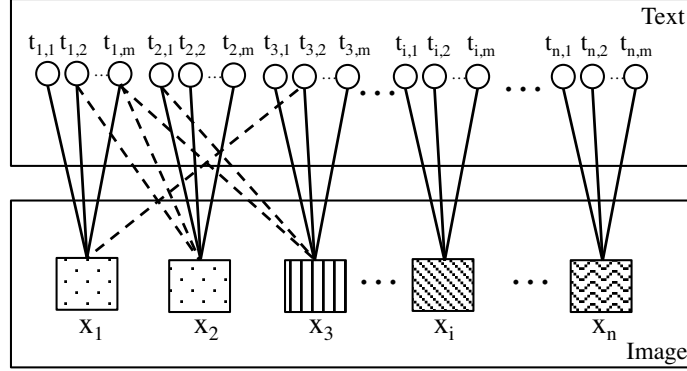


Figure 6.3: Training dataset only captures many-to-one mappings in the text-to-image direction (solid lines). In fact, image-text relationship is many-to-many as indicated with solid and dashed lines.

successful D , among unlabeled samples we must choose those which D will classify as positive with high probability, subject to a fraction r of the positive pairs being correctly classified.

6.2.4 Proposed semantic sampling

Although it is theoretically possible to train a classifier without labeled negative examples, our constrained optimization problem is difficult to solve analytically without the closed form discriminator function $D(\cdot)$, which is a deep convolutional neural net for our case. To overcome the challenge, we turn to an unsupervised technique to sample additional positive images with high probability. To this end we use negative sampling to reduce the space from which we will draw positive samples.

Figure 6.3 illustrates a relationship of images $\mathbf{x}$ and text $\mathbf{t}$. Suppose we have n images where each image is described with m matching text descriptions. We use notation $\mathbf{x}_i$ for i th image and $t_{i,j}$ for j th matching text for $\mathbf{x}_i$. Patterns in $\mathbf{x}$ correspond to class labels l . Solid line indicates labeled matching $(\mathbf{x}, \mathbf{t})$ pairs with $y = +1$ from training data or a set P of positive class data. We also have a set U of unlabeled pairs with instances from P (in dashed line) and from a negative class set $\bar{P}$ (pairs with $y = -1$). Referring to Figure 6.3, if we sample $t_{1,m}$ as anchor text, only positive image available from the training data is $\mathbf{x}_1$. However, some text can correctly describe multiple images (i.e., the mapping in the text-to-

image direction may be one-to-many). These multiple positive images are aliasing images to the anchor text. Utilizing pairs from both P and U , we can sample diverse positive images, e.g., images $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$ for anchor text $\mathbf{t}_{1,m}$ in Figure 6.3. Similarly, only using solid line can potentially sample matching image (e.g., $\mathbf{x}_2$ or $\mathbf{x}_3$) as a negative. As the set P and $\bar{P}$ in U are unlabeled, we aim to infer positive and negative pairs using the cosine similarity metric in the text embedding space. Definitions follow.

Definition 6.2.1. (Positive Image). A positive image has its matching text similar to the anchor text in the text embedding space.

Text can fit multiple images equally well. We use cosine similarity to sample images aliased to the anchor text. Given encoded anchor text, we encourage GAN to generate positive images.

Definition 6.2.2. (Negative Image). A negative image has its matching text dissimilar to the anchor text in the text embedding space.

A negative image does not correspond to the anchor text because it matches sentences different from the anchor text. Thus, we discourage GAN from generating negative images. Figure 6.4 shows sampled images and cosine similarities between their corresponding text and the anchor text. Notice that when the similarity is high, sampled images match the anchor text well. As the similarity value decreases, the semantic relevance between the sample image and the anchor text decreases.

We describe supervised (classful) and unsupervised (classless) sampling of positive and negative images below.

Classful Sampling. As used by other approaches [20,22,23,35], choose an image from the class of a reference image above a cosine similarity threshold α as a classful positive and any image outside the class of the reference image below a cosine similarity threshold β as a negative. Given encoded anchor text $\varphi(\mathbf{t}^{(i)})$, choose a classful positive image

$$\mathbf{x}^{(j)} \text{ s.t. } l^{(i)} = l^{(j)} \text{ and } \text{cossim}(\varphi(\mathbf{t}^{(i)}), \varphi(\mathbf{t}^{(j)})) \geq \alpha \quad (6.4)$$

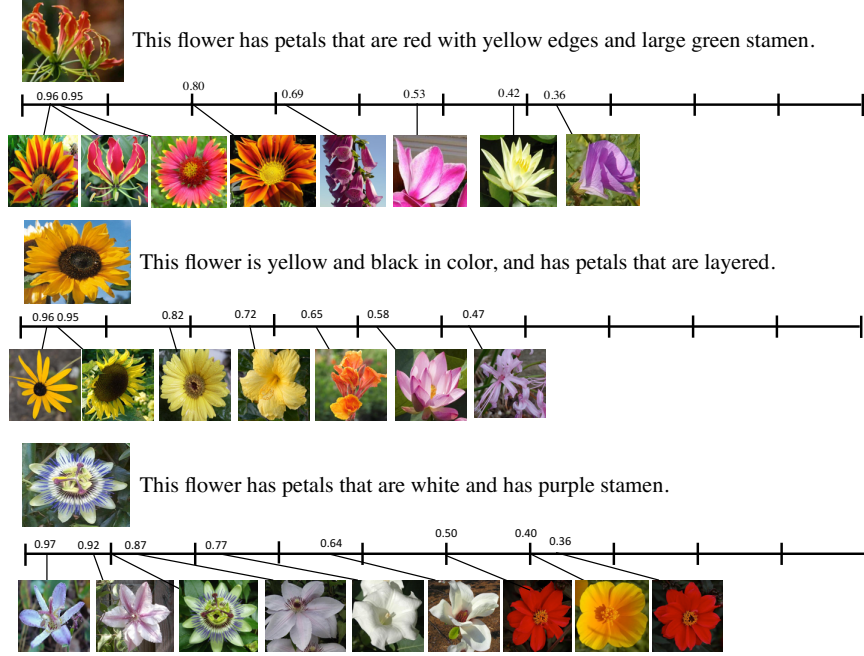


Figure 6.4: Sample images are placed according to cosine similarity between their corresponding text and an anchor text. When cosine similarity is high, semantic match between a sampled image and an anchor text is high.

and choose a classful negative image

$$\mathbf{x}^{(j)} \text{ s.t. } l^{(i)} \neq l^{(j)} \text{ and } \text{cossim}(\varphi(\mathbf{t}^{(i)}), \varphi(\mathbf{t}^{(j)})) \leq \beta \quad (6.5)$$

where $l^{(i)}$ and $l^{(j)}$ denote class labels for image i and j , respectively. Here, $\beta = \alpha - \gamma$ where γ is some margin.

Classless Sampling. Some sentences are ambiguous and could match images across multiple classes. Furthermore, images of the same class can be described by very different sentences. We should consider removal of class constraint in sampling positive and negative images. Given an anchor text encoding $\varphi(\mathbf{t}^{(i)})$, choose a classless positive image

$$\mathbf{x}^{(j)} \text{ s.t. } \text{cossim}(\varphi(\mathbf{t}^{(i)}), \varphi(\mathbf{t}^{(j)})) \geq \alpha \quad (6.6)$$

and choose a classless negative image

$$\mathbf{x}^{(j)} \text{ s.t. } \text{cossim}(\varphi(\mathbf{t}^{(i)}), \varphi(\mathbf{t}^{(j)})) \leq \beta. \quad (6.7)$$

Our principle to set positive sampling parameter α is by averaging cosine similarities among corresponding texts of all training images. For Oxford-102, we get 0.8524. We then conservatively set $\alpha = 0.9$. Our discriminator aims to separate the positive and negative text-image pairs by a margin ($\gamma = 0.2$). Negative sampling further removes the possibility of generating positive images of mismatching semantics. We have found that the same principle can be used in different datasets.

Here we elaborate a point we made earlier in Section 6.2.2 that many-to-one mappings in the text-to-image direction help mitigate the risk of incorrect selection of images. According to Equation (6.7), we note that a selected negative image must have its cosine similarity with the anchor text in the text embedding space to be no greater than β . Recall that the text embedding is the average text embedding of N description texts aliased to the image (we use $N = 4$ as noted in Section 6.3.1). By using aliasing texts, the chance of selecting an incorrect image due to a single outlier description text is greatly reduced for both positive and negative sampling.

6.2.5 Training objectives

We train the generator G and the discriminator D by mini-batch stochastic gradient ascent. The log-likelihood objective for D is used to evaluate whether or not an image applied to the discriminator is real or fake. It also determines if the input image matches the text encoding

$$\begin{aligned} L_D = & \mathbb{E}[\log p(s = \text{“real and match”} | \mathbf{x}_p, \varphi(\mathbf{t}_a))] \\ & + \mathbb{E}[\log p(s = \text{“fake”} | \hat{\mathbf{x}}, \varphi(\mathbf{t}_a))] \\ & + \mathbb{E}[\log p(s = \text{“mismatch”} | \mathbf{x}_n, \varphi(\mathbf{t}_a))] \end{aligned} \quad (6.8)$$

The log-likelihood objective for G is

$$L_G = \mathbb{E}[\log p(r = \text{“real and match”} | \hat{\mathbf{x}}, \varphi(\mathbf{t}_a))] \quad (6.9)$$

For training the generator, we maximize L_G .

6.3 Experiments

We analyze the effect of sampling schemes on text-to-image synthesis. We evaluate diversity of the generated images by the inception score [87] and the multi-scale structural similarity index (MS-SSIM) [88] metrics. The inception score measures how well a generated image contains a distinctive class of objects and also measures inter-class diversity in the generated images. The MS-SSIM metric is used to measure intra-class diversity. We sample 400 image pairs randomly from each class and report their MS-SSIM scores. Finally, we apply our semantic sampling scheme to previous approaches GAN-INT-CLS [20] and HDGAN [23] and show how a simple change in sampling can improve diversity of generated images.

6.3.1 Settings

We use images from the Oxford-102 flower dataset [71] that consists of 8,189 images in 102 flower types. Following Reed et al. [20], we split the dataset into 82 training-validation and 20 test classes.

As described in the previous chapters, Reed et al. collected 10 corresponding sentences for each image using Amazon Mechanical Turk. We encode a text sentence into a 1,024-dimensional vector using a pretrained character-level ConvNet with a recurrent neural network (char-CNN-RNN). We randomly sample 4 out of 10 sentences and aggregate by averaging the sampled sentence text embeddings.

Using the same setup as Reed et al. [72], we train our models with mini-batch stochastic gradient ascent with a mini-batch size of 64. We use the ADAM solver [73] with a momentum 0.5 and a learning rate 0.0002.

Table 6.1: Inception scores of flower image generation using various sampling schemes on GAN-INT-CLS.

Positive		Negative		Inception Score
Classless	α	Classless	β	
No	1	No	1	2.66 ± 0.03
No	1	-	-	1.34 ± 0.01
No	0.9	-	-	1.41 ± 0.01
Yes	0.9	-	-	1.55 ± 0.01
No	0.9	No	0.7	2.95 ± 0.03
Yes	0.9	Yes	0.7	4.00 ± 0.06

6.3.2 Sampling scheme comparison

Table 6.1 shows the inception scores of various sampling schemes when applied to the GAN-INT-CLS model [20]. The baseline sampling scheme selects an image in the same class that directly corresponds to the anchor text as a positive image. The baseline is classful with $\alpha = 1$. For negative sampling, the baseline scheme randomly selects any image outside the class of the image that anchor text corresponds to. Such negative sampling scheme is classful with $\beta = 1$. The inception score for the baseline is 2.66 ± 0.03 .

We first analyze the effect of positive sampling only (without negative enforcement). Decreasing α effectively increases the number of positive samples for an anchor text. Changing classful positive sampling from $\alpha = 1$ to 0.9 increases inception score by 0.07. Removing the classful constraint on positive sampling further increases the diversity and yields a better inception score, 1.55 ± 0.01 . Figure 6.5 presents positive images for anchor text using (a) classful sampling with $\alpha = 1$, (b) with $\alpha = 0.9$, and (c) classless sampling with $\alpha = 0.9$. A simple modification to classless positive sampling increases the diversity of positive samples. Compared to the baseline, the inception scores are much lower without negative sampling.

Next, we evaluate classful sampling of both positive and negative images. Compared to the baseline scheme, decreasing α increases diversity and decreasing β ensures that semantically matching images are not sampled as negative images. Classful sampling of disjoint positive and negative sets (by $\gamma = 0.2$) further improves the inception score to 2.95 ± 0.03 . Changing from classful to classless significantly increases the inception score. This result shows that diversity of positive images plays an important role in improving text-to-image

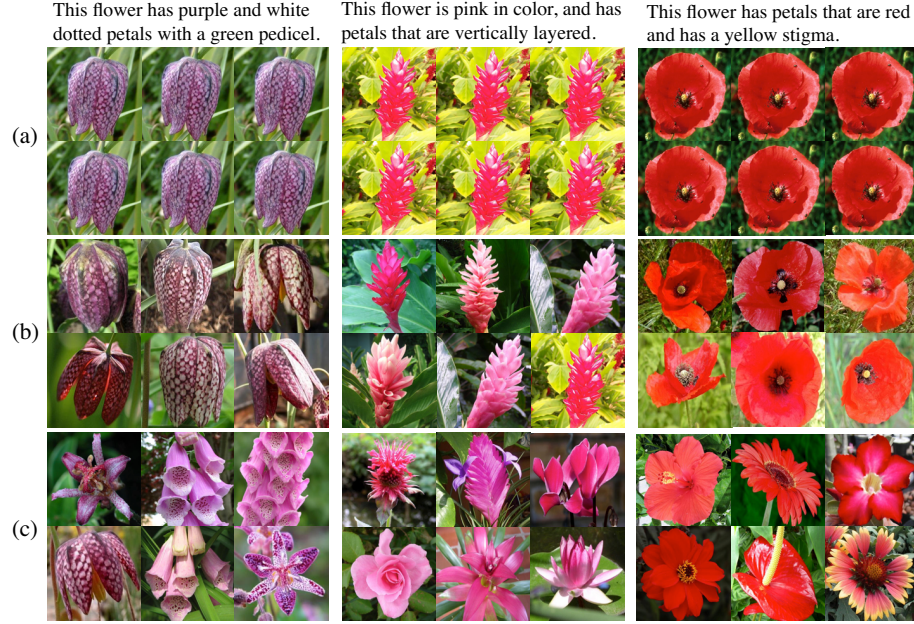


Figure 6.5: Randomly selected positive images for an anchor text based on (a) classful sampling with $\alpha = 1$ (baseline), (b) classful sampling with $\alpha = 0.9$, and (c) classless sampling with $\alpha = 0.9$.

synthesis results. Compared to the baseline scheme, classless sampling scheme extends the variety of positive images, and sampled negative images have low semantic match to the anchor text. To validate that the discriminator trained with our classless sampling is better than the baseline sampling, we compare averaged discriminator outputs on detecting semantic match from test data. The averaged output for classless sampling scheme came out to be 0.28 higher than the averaged output for the baseline scheme.

We compute MS-SSIM to show that the generated images from our classless semantic sampling are as diverse as training images. Figure 6.6 compares the mean MS-SSIM for each class in the training dataset compared to that of the generated images with our classless semantic sampling. Our scheme produces as diverse images as the training images.

6.3.3 Extension to other models

Table 6.2 compares the inception scores of various models, namely GAN-INT-CLS [20], StackGAN [22], TAC-GAN [35], and HDGAN [23]. Note these models used baseline sam-

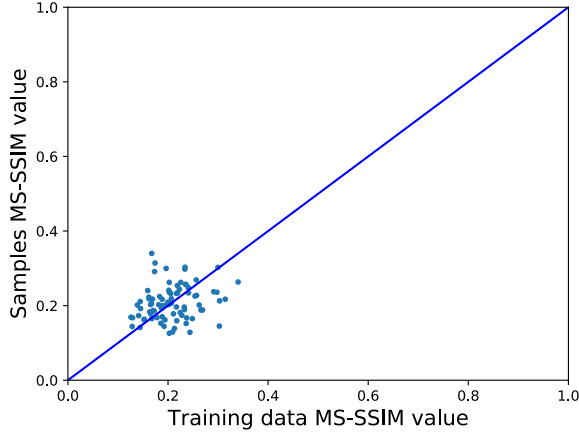


Figure 6.6: Comparison of the class-wise MS-SSIM scores of the samples from the training data and the generated samples of GAN-INT-CLS using classless semantic sampling.

Table 6.2: Inception scores on various models

Model	Inception score
GAN-INT-CLS [20]	2.66 ± 0.03
StackGAN [22]	3.20 ± 0.01
TAC-GAN [35]	3.45 ± 0.05
HDGAN [23]	3.45 ± 0.07
GAN-INT-CLS /w classless sampling	4.00 ± 0.06

pling scheme. We report that GAN-INT-CLS with our classless sampling has superior inception score to other models using the baseline scheme.

Figure 6.7 shows generated images with GAN-INT-CLS and HDGAN using baseline sampling and our semantic sampling schemes. For both models, our sampling method increases diversity of generated images while maintaining semantic match to input text.

6.4 Summary

This chapter proposes semantic sampling schemes to further increase diversity while maintaining semantic correctness to a given sentence. Our semantic sampling is able to exploit image and text aliases often present in (image, text) pairs in the training data. Both quantitative and qualitative analyses suggest that our sampling scheme improves diversity of

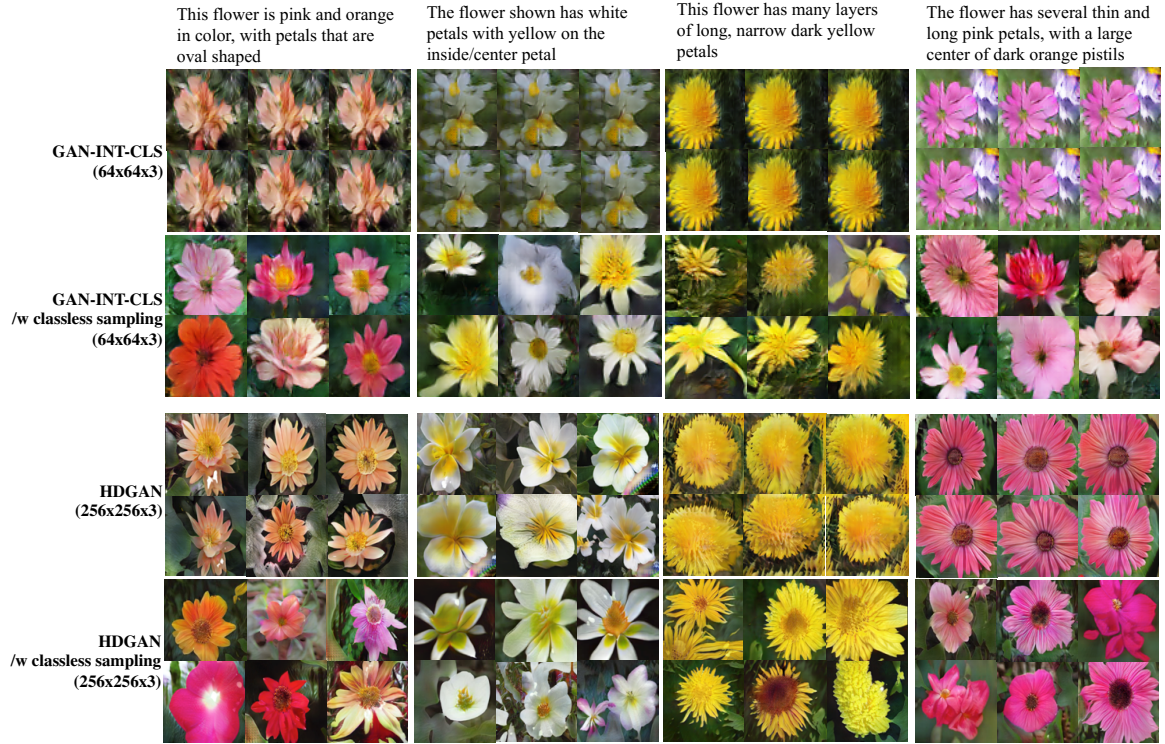


Figure 6.7: Generated flower images using baseline sampling and our classless semantic sampling on GAN-INT-CLS and HDGAN

text-to-image synthesis. We also show that our sampling scheme is applicable to a broader range of text-to-image synthesis models.

Chapter 7

Conclusion

Inspired by human ability to process multimodal data, this thesis has explored methods that can efficiently interpret and creatively connect different sources of multimodal information. Along the growth of multimodal contents on the web, machine learning plays a key role to learn good multimodal features and translate between modalities. This thesis has emphasized on the empirical evaluation of the proposed approaches on a variety of multimodal tasks including regression, recognition, retrieval, and synthesis.

This thesis has first examined a multimodal sparse coding scheme for learning a joint feature representation that integrates information across different modalities. By jointly training dictionaries, one can force the multimodal data to share a similar sparse representation, which discovers correlation across the modalities. We showed that the learned sparse representation were highly discriminative for various recognition and retrieval tasks. From cross-modal sparse codes, we find that unimodal features learned from multimodal training performed better than unimodal features learned from unimodal training. In essence, a machine can learn better with paired multimodal data even when it is tested on a single modality. Here, we have used sparse coding as a recovery engine. It can alternatively be viewed as a probabilistic generative model. However, probabilistic sparse coding has become less attractive compared to deep neural networks that can incorporate a more flexible choice of function approximators.

Our conditional image synthesis model based on deep convolutional GAN learned to generate realistic images according to text descriptions. Despite the overall fair quality, an

early work in adversarial text-to-image synthesis has failed to contain structural details of object parts. We exploit the feature learning capability of pre-trained discriminative models (e.g., VGG) to guide the training process of the generative model. We have found that the high-level feature representation can teach the generative model to extract object-specific structures in natural images.

Beyond visual realism, we have developed a new approach that improves the training of GAN for synthesizing diverse images from a text input. We have expanded on CGAN leveraging an auxiliary task in the discriminator. Instead of class prediction, we modify the discriminator to regress semantic relevance between images and text. We find the training with extra semantic regression beneficial for the generator diversifying its generated examples, alleviating the mode collapse problem.

The main reason for mode collapsing in text-to-image synthesis is the loss of many-to-many mapping between an image and text during training. We have devised an unsupervised sampling scheme for pairing up an image with text that preserves the many-to-many relationship. Our sampling scheme forms training data in a high-dimensional space of embedded text inputs. We have demonstrated a substantially improved diversity of the generated images after applying our scheme to off-the-shelf text-to-image synthesis models.

This thesis has explored the problems of learning joint representations from multimodal data and synthesizing images based on conditional multimodality. As advanced multimodal feature learning, I have developed algorithms that exploit sparsity and hierarchy while demonstrating good classification and retrieval performances in many multimedia applications. Furthermore, I have proposed approaches to generate more realistic and diverse images that depict a sentence. I believe that the abilities for algorithms to interpret and transfer multimodal messages show the ability of the models to think more like humans. Human-level intelligence to process multimodal data is still beyond reach. On the positive side, multimodal data and computational resources have significantly increased, creating opportunities to explore more humanlike learning capabilities. We hope that over time our research efforts will allow computers to approach human-like capacity for understanding multimodal data and consequently be more useful in solving everyday problems.

Bibliography

- [1] H. Hotelling. Relations between two sets of variates. In *Biometrika*, 1936.
- [2] C. Fyfe and P. L. Lai. Kernel and nonlinear canonical correlation analysis. In *International Journal of Neural Systems*, 2001.
- [3] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. In *Neural computation*, 2004.
- [4] N. Martin and H. Maes. Multivariate analysis. 1979.
- [5] J. R. Ketterling. Canonical analysis of several sets of variables. In *Biometrika*, 1971.
- [6] Z. Ma, Y. Lu, and D. Foster. Finding linear structure in large datasets with scalable canonical correlation analysis. In *ICML*, 2015.
- [7] F. Y. K. Mikolajczyk. Deep correlation for matching images and text. In *CVPR*, 2015.
- [8] G. Andrew, R. Arora, J. Bilmes, and K. Livescu. Deep canonical correlation analysis. In *ICML*, 2013.
- [9] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Ng. Multimodal deep learning. In *ICML*, 2011.
- [10] P. Poirson and H. Idrees. Multimodal stacked denoising autoencoders. *Technical Report*, 2013.
- [11] N. Srivastava and R. Salakhutdinov. Multimodal learning with deep Boltzmann machines. *NIPS*, 2012.
- [12] Y. Jia, M. Salzmann, and D. Trevor. Factorized Latent Spaces with Structured Sparsity. In *NIPS*, 2010.
- [13] G. Monaci, P. Vandergheynst, and F. Sommer. Learning Bimodal Structure in Audio-Visual Data. In *IEEE Trans. on Neural Networks*, 2009.
- [14] Y. Zhuang, Y. Wang, F. Wu, Y. Zhang, and W. Lu. Supervised Coupled Dictionary Learning with Group Structures for Multi-Modal Retrieval. In *AAAI*, 2013.

- [15] J. Wright, A.Y. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. In *PAMI*, 2009.
- [16] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution as sparse representation of raw image patches. In *CVPR*, 2008.
- [17] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution via sparse representation. In *IEEE Transactions on Image Processing (TIP)*, 2010.
- [18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
- [19] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.
- [20] Scott Reed, Zeynep Akata, Xinchun Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 3, 2016.
- [21] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.
- [22] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiao lei Huang, Xiaogang Wang, and Dimitris N. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. *ICCV*, 2017.
- [23] Zizhao Zhang, Yuanpu Xie, and Lin Yang. Photographic text-to-image synthesis with a hierarchically-nested adversarial network. *CVPR*, 2018.
- [24] Hao Dong, Jingqing Zhang, Douglas McIlwraith, and Yike Guo. I2t2i: Learning text to image synthesis with textual data augmentation. *ICIP*, 2017.
- [25] Scott E. Reed, Zeynep Akata, Santosh Mohan, Samuel Tenka, Bernt Schiele, and Honglak Lee. Learning what and where to draw. *arXiv preprint arXiv:1610.02454*, 2016.
- [26] Seunghoon Hong, Dingdong Yang, Jongwook Choi, and Honglak Lee. Inferring semantic layout for hierarchical text-to-image synthesis. *CVPR*, 2018.
- [27] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiao lei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. *arXiv preprint arXiv:1711.10485*, 2017.

- [28] Jianbo Chen, Yelong Shen, Jianfeng Gao, Jingjing Liu, and Xiaodong Liu. Language-based image editing with recurrent attentive models. In *CVPR*, 2018.
- [29] Shikhar Sharma, Dendi Suhubdy, Vincent Michalski, Samira Ebrahimi Kahou, and Yoshua Bengio. Chatpainter: Improving text to image generation using dialogue. *arXiv preprint arXiv:1802.08216*, 2018.
- [30] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. *ICLR*, 2017.
- [31] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *NIPS*. 2017.
- [32] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. CVAE-GAN: fine-grained image generation through asymmetric training. *ICCV*, 2017.
- [33] Yunus Saatchi and Andrew Gordon Wilson. Bayesian gan. In *NIPS*. 2017.
- [34] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. *ICML*, 2017.
- [35] Ayushman Dash, John Cristian Borges Gamboa, Sheraz Ahmed, Marcus Liwicki, and Muhammad Zeshan Afzal. TAC-GAN - text conditioned auxiliary classifier generative adversarial network. *arXiv preprint arXiv:1703.06412*, 2017.
- [36] Bruno A Olshausen and David J Field. Sparse Coding with an Overcomplete Basis Set: Strategy Employed by V1? *Vision research*, 1997.
- [37] J.A. Tropp and A.C. Gilbert. Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit. *IEEE Trans. on Information Theory*, 2007.
- [38] Robert Tibshirani. Regression Shrinkage and Selection via Lasso. *Journ. of Royal Statistical Society*, 1994.
- [39] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online Dictionary Learning for Sparse Coding. In *ICML*, 2009.
- [40] INRIA. Sparse modeling software. <http://spams-devel.gforge.inria.fr/>.
- [41] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 2006.
- [42] J. Wright, Y. Ma, J. Maral, G. Sapiro, T. Huang, and S. Yan. Sparse representation for computer vision and pattern recognition. In *Proceedings of the IEEE*, 2010.

- [43] M. Cha, Y. Gwon, and H. T. Kung. Twitter Geolocation and Regional Classification via Sparse Coding. In *ICWSM*, 2015.
- [44] James R Williamson, Elizabeth Godoy, Miriam Cha, Adrienne Schwarzentru-ber, Pooya Khorrami, Youngjune Gwon, Hsiang-Tsung Kung, Charlie Dagli, and Thomas F Quatieri. Detecting depression using vocal, facial and semantic communi-cation cues. In *ACM MM AVEC*, 2016.
- [45] Y. Gwon, W. Campbell, D. Sturim, and H. T. Kung. Language Recognition via Sparse Coding. In *Interspeech*, 2016.
- [46] GKevin Brady, Youngjune Gwon, Pooya Khorrami, Elizabeth Godoy, William Camp-bell, Charlie Dagli, and Thomas S Huang. Multi-modal audio, video and physiologi-cal sensor learning for continuous emotion prediction. In *ACM MM AVEC*, 2016.
- [47] G. S. V. S. Sivaram, S. K. Nemala, M. Elhilali, T. D. Tran, and H. Hermansky. Sparse coding for speech recognition. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010.
- [48] J. Yang, J. Wright, T. Huang, and Y. Ma. Image Super-Resolution Via Sparse Repre-sentation. *IEEE Trans. on Image Processing*, 19(11):2861–2873, 2010.
- [49] L. Bo, X. Ren, and D. Fox. Multipath Sparse Coding Using Hierarchical Matching Pursuit. In *CVPR*, 2013.
- [50] K. Yu, Y. Lin, and J. Lafferty. Learning Image Representations from the Pixel Level via Hierarchical Sparse Coding. In *CVPR*, 2011.
- [51] A. Krizhevsky. Learning multiple layers of features from tiny images. *Master’s thesis, University of Toronto*, 2009.
- [52] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008.
- [53] NIST. 2014 TRECVID Multimedia Event Detection & Multimedia Event Recounting Tracks. <http://nist.gov/itl/iad/mig/med14.cfm>.
- [54] D. G. Childers, D. P. Skinner, and R. C. Kemerait. The cepstrum: A guide to process-ing. In *Proceedings of the IEEE*. 1977.
- [55] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*, abs/1409.1556, 2014.
- [56] W. M. Campbell, D. E. Sturim, and D. A. Reynolds. Support Vector Machines Us-ing GMM Supervectors for Speaker Verification. *IEEE Signal Processing Letters*, 13(5):308–311, May 2006.

- [57] D. Lowe. Distinctive image features from scale-invariant keypoints. In *IJCV*, 2004.
- [58] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. In *JMLR*, 2003.
- [59] D. Borth, R. Ji, T. Chen, T. Breuel, and S. Chang. Large-scale Visual Sentiment Ontology and Detectors Using Adjective Noun Pairs. *ACM Multimedia Conference*, 2013.
- [60] C. Baecchi, T. Uricchio, M. Bertini, and A.D. Bimbo. A multimodal feature learning approach for sentiment analysis of social network multimedia. In *Multimed Tools Appl*, 2015.
- [61] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, 2010.
- [62] Jon Gauthier. Conditional generative adversarial nets for convolutional face generation. *Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition, Winter semester*, 2014(5):2, 2014.
- [63] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, pages 1486–1494, 2015.
- [64] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 262–270, 2015.
- [65] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016.
- [66] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint arXiv:1609.04802*, 2016.
- [67] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [68] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

- [69] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.
- [70] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.
- [71] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on*, pages 722–729. IEEE, 2008.
- [72] Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–58, 2016.
- [73] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [74] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.
- [75] A.B. Lyons. *Plant Names: Scientific and Popular*. Nelson, Baker & Company, 1900.
- [76] Zizhao Zhang, Yuanpu Xie, and Lin Yang. Photographic text-to-image synthesis with a hierarchically-nested adversarial network. *CVPR*, 2018.
- [77] Miriam Cha, Youngjune Gwon, and H.T. Kung. Adversarial nets with perceptual losses for text-to-image synthesis. *MLSP*, 2017.
- [78] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. *NIPS*, 2017.
- [79] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *ICCV*, 2017.
- [80] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. mage-to-image translation with conditional adversarial networks. *CVPR*, 2017.
- [81] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. *CVPR*, 2016.
- [82] Yoshua Bengio, Jerome Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. *ICML*, 2009.

- [83] Sandeep Subramanian, Sai Rajeswar, Francis Dutil, Chris Pal, and Aaron Courville. Adversarial generation of natural language. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*. ACL, 2017.
- [84] Ofir Press, Amir Bar, Ben Bogin, Jonathan Berant, and Lior Wolf. Language generation with recurrent generative adversarial networks without pre-training. *Proceedings of the 1st Workshop on Learning to Generate Natural Languages, ICLR*, 2017.
- [85] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *ICLR*, 2018.
- [86] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *arXiv preprint arXiv:1606.03498*, 2016.
- [87] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *arXiv preprint arXiv:1512.00567*, 2015.
- [88] Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*, 2003.
- [89] Bing Liu, Wee Sun Lee, Philip S. Yu, and Xiaoli Li. Partially supervised classification of text documents. In *ICML*, 2002.
- [90] Xiaoli Li and Bing Liu. Learning to classify texts using positive and unlabeled data. In *IJCAI*, 2003.
- [91] Bernhard Schölkopf, John C. Platt, John C. Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. Estimating the support of a high-dimensional distribution. *Technical Report. Microsoft Research MSR-TR-99-87*, 1999.
- [92] Larry M. Manevitz and Malik Yousef. One-class svms for document classification. *Journal of Machine Learning Research*, 2002.