DISENTANGLED GENERATIVE MODELS AND THEIR APPLICATIONS

by

BINGCHEN LIU

A dissertation submitted to the
School of Graduate Studies
Rutgers, The State University of New Jersey
In partial fulfillment of the requirements
For the degree of
Doctor of Philosophy
Graduate Program in Computer Science

Written under the direction of
Ahmed Elgammal

and approved by

__________________________

__________________________

__________________________

__________________________

New Brunswick, New Jersey
May, 2022
Disentangled Generative Models and their Applications

By Bingchen Liu
Dissertation Director:
Ahmed Elgammal

Generative models, such as Auto-Encoders, Generative Adversarial Networks, Generative Flows, and Diffusion Models, are fascinating for their ability to synthesize versatile visual and audio information from merely noise. This generative process usually requires a model to perceive and compress high-dimensional data into a compact, low-dimensional latent space, where each dimension encodes some valuable semantic variations in the original data space.

How much we know about the latent space is vital because it determines how we can take advantage of the corresponding generative model. Imagine a GAN trained on human faces. If we know which dimension in its latent vector controls the concept “hair shape”, we can synthesize multiple images for the same face to try different hairstyles without changing other facial attributes. Disentangling generative models makes them more fun to play with, which is the topic of this thesis.

This thesis studies the unsupervised disentangling of the latent space in GANs focused on the image domain and extended to multi-modalities (image captioning and text-to-image synthesis). The proposed methods in this thesis enable the GAN model to disentangle its latent space automatically, thus sparing the expensive effort of collecting semantic labeling for the training data. Derived from disentanglement, this thesis also
covers studies on model interpretability and human-controllable data synthesis. This thesis contains three main topics:

First, we work on general-purpose disentanglement. A novel GAN-based disentanglement framework with One-Hot Sampling and Orthogonal Regularization (OOGAN) is proposed. While previous works primarily attempt to tackle disentanglement learning through VAE with various approximation-based methods, we show that GANs have a natural advantage in disentangling with an alternating latent variable (noise) sampling method that is straightforward and robust.

Second, we work on a more specific task: disentangling coarse and fine level style attributes for GAN. The proposed PIVQGAN facilities independent control and manipulation of coarse-level object arrangements (posture) and fine-grained level styling (identity) for both synthesized images from noise or sampled images from real datasets. We design a Vector-Quantized module for better pose-identity disentanglement and a novel joint-training scheme merging GAN and Auto-Encoder, which facilities several self-supervision tasks for the model to better separate the attributes.

Lastly, we study two applications taking advantage of a better disentangled GAN with mutual information learning. Focusing on text-to-image generation, we propose Text-and-Image Mutual-Translation Adversarial Networks (TIME), a lightweight but effective model that jointly learns a T2I generator and an image captioning discriminator under one single GAN framework. By maximizing the mutual information between the latent models of image and text. Focusing on sketch-to-image generation, we study exemplar-based sketch-to-image synthesis tasks in a self-supervised learning manner, eliminating the necessity of the paired sketch data via a better disentanglement between content information from sketch and style information from an exemplar RGB image.
Acknowledgements

Within the five years of my journey as a Ph.D. student, Prof. Ahmed Elgammal had the most significant impact on me. It is my genuine pleasure to express my gratitude and thanks to him. From the decision-making of diving into the world of computer vision and deep learning to exploring, testing, and figuring out research problems, topics, and directions, it was Prof. Elgammal who encouraged, supported, and guides me to make it here. Beyond the study, Prof. Elgammal’s charisma also inspires me in my life and career. I am fortunate to have him as my advisor.

I am also very grateful to the rest of my dissertation committee: Prof. Yongfeng Zhang, Prof. Vladimir Pavlovic, and Dr. Xiaohui Shen (Bytedance Research Manager), for their time reviewing my dissertation and their valuable suggestions. I received solid foundations from Prof. Pavlovic’s lecture series on Pattern-Recognition and Machine-Learning (I got two “A”s). I also would like to express my gratitude to Prof. Gerard de Melo, who helped me immensely in my multi-modal Learning and Nature-Language-Processing.

My sincere gratitude also goes to my beloved colleagues in our Art-and-AI-Lab. Dr. Mohamed Elhoseiny and Dr. Yizhe Zhu are both my respectful seniors. They offered on-hand mentoring and valuable advice throughout all stages of my Ph.D. study. Diana Kim, Shahrzad Ziaee, and Mohamed Latif were a joy to collaborate with, talk to, and learn from them. Kunpeng Song, my hard-working junior. I see myself from him sometimes. I wish them fruitful achievements in their researches.

I want to thank my beloved friends I met during my life at Rutgers: Jiahao Cai, Weitao Li, Yuchen Li, Xiao Qi, Tianwei Li, Fan Yang, Kun Wang, Zuohui Fu, Yuting Wang. They made my Ph.D. life more colorful and wonderful. They are great meal buddies, homework partners, and basketball teammates. Thanks for the sweet memories and sharing the excitement and frustrations along this unforgettable journey.

Last but not least, I never forget to thank my family for their generous support all these years. Mainly, my thanks go to my wife, Shiying Xu. I cannot make it here without her support. She went through the most challenging time with me and always motivated and inspired me to keep moving forward. I appreciate all the accompanies from her.
# Table of Contents

Abstract .................................................................................................................. ii  
Acknowledgements ............................................................................................... iv  
List of Tables .......................................................................................................... viii  
List of Figures ......................................................................................................... ix  

1. Introduction ......................................................................................................... 1  
   1.1. Unsupervised Image Disentanglement Learning ........................................ 1  
      1.1.1. Problem Definition ........................................................................... 1  
      1.1.2. Motivations and Challenges .............................................................. 2  
      1.1.3. Contributions .................................................................................. 2  
   1.2. Mutual-Information based Multi-modal Learning ....................................... 3  
      1.2.1. Motivations and Challenges .............................................................. 3  
      1.2.2. Contributions .................................................................................. 4  
   1.3. Posture and Identity Disentangled Image-to-image Translation .................. 4  
      1.3.1. Motivations and challenges ............................................................... 4  
      1.3.2. Contributions .................................................................................. 5  

2. Related work ....................................................................................................... 6  
   2.1. Unsupervised Image Disentangling Models .............................................. 6  
      2.1.1. $\beta$-VAE-based models: ................................................................. 6  
      2.1.2. GAN-based models: ........................................................................ 7  
   2.2. Text-to-Image Multi-Modal Learning ...................................................... 7  
      2.2.1. StackGAN as the Image Generation Backbone. ............................... 8  
      2.2.2. Dependence on Pre-trained modules. .............................................. 8  
      2.2.3. The Image-Text Attention Mechanism. ........................................... 8  
   2.3. Unsupervised Image-to-Image Translation ............................................... 9  
      2.3.1. Disentangled image synthesis ........................................................... 9
3. Disentangling GAN with One-Hot Sampling and Orthogonal Regularization

3.1. Introduction

3.2. Proposed Method

3.2.1. Alternating Continuous and One-hot Sampling

3.2.2. Compete-Free Generator

3.2.3. Orthogonal Regularized & Grouped Feature Extractor

3.3. Perceptual Diversity Metric

3.3.1. Limitations

3.4. Experiments

3.4.1. Quantitative Results on dSprites

3.4.2. Qualitative Results on 3D Chairs

3.4.3. Disentangling at a Higher Resolution on CelebA

3.4.4. Ablation Studies

3.4.5. More Qualitative Results

4. Text and Image Mutual-Translation Adversarial Networks

4.1. Introduction

4.2. Prior Works

4.2.1. StackGAN as the Image Generation Backbone

4.2.2. Dependence on Pre-trained modules

4.2.3. Difference to MirrorGAN

4.3. The Motivation of Mutual Translation

4.4. Proposed Method

4.4.1. Model Structures

4.4.2. Objectives

4.5. Experiments

4.5.1. Backbone Model Structure

4.5.2. Controllable G without Sentence-Embedding

4.5.3. Ablation Study

4.5.4. Comparison on T2I with State-of-the-Arts

4.5.5. More Qualitative Results

4.6. Conclusion
List of Tables

3.1. Disentanglement using Kim et al.’s metric ........................................ 23
3.2. Disentanglement using Perceptual Diversity metric ............................. 27
4.1. Comparison between stacked and aggregated model structures on the CUB dataset .............................................................. 43
4.2. Comparison of different attention settings on CUB ......................... 43
4.3. Ablation Study of TIME on CUB dataset ........................................ 46
4.4. Comparison between different attention settings on MS COCO dataset . . . . 47
4.5. Results on downstream Vision-Language tasks from TIME on COCO, compared with SOTA models ............................................. 48
4.6. Text-to-Image performance comparison between TIME and other models .... 48
5.1. Comparison to baselines ................................................................. 61
5.2. Ablation study .................................................................
List of Figures

3.1. OOGAN makes minimal changes upon a basic GAN. $c$ denotes the continuous control vector, $z$ is the noise vector, $c'$ is the feature representation of fake images. 12

3.2. Latent traversals trained on CelebA to showcase the competing and conflicting issue. The images are from the same set of $(z, c)$ on one fixed dimension of $c$ after different training iterations. We observe that InfoGAN begins to capture what appears to be a “wearing glasses” feature at a very early stage, but discards it during training in all dimensions of $c$. In contrast, when OOGAN begins to capture this feature, it consistently masters it in the end. 16

3.3. Model structures: (a) Input block of the compete-free $G$. (b) Orthogonal-regularized grouped $Q$. 18

3.4. Generated images for CelebA: In each group, the left-most image is generated from a randomly sampled $c$, and the following ones are generated by changing the value of each dimension in $c$ to 1. (a) OOGAN exhibits greater visual differences among each dimension, reflecting its ability to learn diverse latent factors. (b) Without the proposed one-hot sampling, OOGAN still manages to learn some distinguishable features, reflecting the advantage of its structural design. (c) The 4 top right images show that the learned features for an InfoGAN have a large overlap across the latent dimensions in $c$, lacking proper disentanglement. 21

3.5. Latent traversals on dSprites 22

3.6. Latent traversals on 3D Chair 23

3.7. Latent traversals trained on CelebA 24

3.8. Binary classification of each labeled attributes for each dimension 25

3.9. Binary classification of each labeled attributes for all dimensions 26

3.10. (a),(b): L1 losses between sampled $c$ and predicted $c'$. (c) TC estimation during training 26

3.11. OOGAN based on Vanilla GAN. Latent traversals along different dimensions 28

3.12. OOGAN based on Vanilla GAN. Latent traversals along different dimensions 29
3.13. OOGAN based on StyleGAN. Latent traversals along different dimensions. . . . 29
3.14. OOGAN based on StyleGAN. Latent traversals along different dimensions. . . . 30
3.15. OOGAN on MNIST dataset, the 4 sections are 4 selected indices from the discrete 
    vector (categorical one-hot vector), inside each section are latent traversals on 
    the same 6 continuous dimensions. . . . . . . . . . . . . . . . . . . . . . . . . . . . 30

4.1. Text-to-image results of TIME on the CUB dataset, where $D$ works as a stand-
    alone image-captioning model. . . . . . . . . . . . . . . . . . . . . . . . . . . . 32
4.2. (a) The StackGAN structure that serves as the backbone in SOTA T2I models 
    [113, 109, 90, 117, 11, 62, 31]. (b)&(c) Representative models build upon Stack-
    GAN, with red parts indicating modules that require pre-training. Note that our 
    proposed model TIME does not: 1. require pre-training the red modules in (b) 
    and (c); 2. require multiple discriminators (the green modules) in (a). . . . . . . 33
4.3. Model overview of TIME. The upper panel shows a high-level summary of our 
    architecture while the lower panel demonstrates the details of the individual 
    modules. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36
4.4. Differences between the attention of AttnGAN and TCIT. . . . . . . . . . . . 37
4.5. Visualization of 2-D positional embedding. . . . . . . . . . . . . . . . . . . . . . 38
4.6. Samples generated during the training of TIME, note the visual features emerge 
    in very early iterations. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 40
4.7. Images from TIME with fixed $z$ and varied sentences . . . . . . . . . . . . 44
4.8. Performance comparison on different annealing schedules of the hinged image-
    text consistency loss. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46
4.9. Learned word embeddings on CUB, and qualitative results on MS-COCO . . . 48
4.10. Image captioning results of TIME on the CUB dataset testing set. . . . . . . 50
4.11. Image captioning results of TIME on the MS-COCO dataset testing set. . . . 50
4.12. Uncurated Qualitative results of TIME on the CUB dataset: The images are 
    generated by TIME given the captions from testing dataset. . . . . . . . . . . 51
4.13. Uncurated Qualitative results of TIME on the MS-COCO dataset: The images 
    are generated by TIME given the captions from testing dataset. . . . . . . . 51
4.14. Uncurated Qualitative results of TIME on the CUB dataset: The images are 
    generated by TIME given the captions from testing dataset, each row containing 
    the generates samples from the same caption. . . . . . . . . . . . . . . . . . . . . 52
4.15. Uncurated Qualitative results of TIME on the MS-COCO dataset: The images are generated by TIME given the captions from testing dataset.

5.1. Unsupervised image-to-image translation results of PIVQGAN with disentangled posture and identity control. In each panel, the first row has input pose images, and the first column has referential identity images. The second and third rows are “segmentation-like” masks automatically learned by PIVQGAN, and bottom-right are the synthesized images.

5.2. **Top**: Illustration of the VQSN module. (1) A base feature-map is quantized by a certain number (e.g. 2 or 6) of trainable embeddings. (2) The VQ embeddings are used to perform the spatial-wise affine transformation on image feature-maps. **Bottom**: Training scheme of PIVQGAN. Apart from the regular GAN training (omitted), we train an encoder together with the convolution part of the generator with self-supervision tasks. In (3), we do latents-grafted reconstruction to help the model disentangle pose/identity attributes at different layers.

5.3. Qualitative comparison of PIVQGAN and other baselines on AFHQ dataset. Advantages of our model are highlighted.

5.4. **Left**: Object-mask editing. **Right**: Various augmentation methods used on the model’s identity inputs, enabling the model to learn the augment-variant attributes from only posture inputs.

5.5. **Left**: VQSN is controlled by the latent vectors, thus it has a smooth posture transaction when interpolating between two pose references. **Right**: When fixing the pose-latents and interpolating only the identity-latents, PIVQGAN shows a smooth transaction on identity-related attributes.

5.6. PIVQGAN is robust to out-domain images with unseen textures, and can generate meaningful in-domain counterparts from these out-domain inputs. (a,b) Out-domain images as identity input. (c,d) Out-domain images as posture input.

5.7. Generated results of PIVQGAN on more structural-complicated datasets: LSUN-cars and LSUN-churches. Note how the VQSN module is able to capture the abstract posture information, and the identity features can be accurately transferred among the same identity image among different postures.

5.8. Generated results of PIVQGAN on more structural-complicated dataset: cartoon stickers.
5.9. Cross-domain results of PIVQGAN trained on images joining cartoon stickers and AFHQ. PIVQGAN is able to stylized realistic animal faces into different stickers, as varied degrees. The style-transfer performance is superior and have the semantic-awareness property.

5.10. Cross-domain results of PIVQGAN trained only on anime girl faces. Note how to model is able to translate sticker images from unseen image domain into the anime girls. It shows the great potential on PIVQGAN on the task of zero-shot image domain translation.

5.11. Cross-domain results of PIVQGAN trained only on anime girl faces. The model translates real human face images from unseen image domain to anime girls.

5.12. Qualitative comparison between PIVQGAN and StarGAN-v2. PIVQGAN is superior in image quality, posture translation, and identity accuracy.

5.13. Different self-learned posture representations from differently configured VQSN modules.
Chapter 1
Introduction

A disentangled representation separates the underlying factors of variation such that each dimension exclusively encodes one semantic feature [8, 49]. While the benefits of the learned representation for downstream tasks is questioned by [77], disentangling a Deep Neural Network (DNN) is still of great value in terms of human-controllable data generation, data manipulation and post-processing, and increasing the model interpretability. Moreover, disentanglement learning in an unsupervised manner can effectively highlight the biased generative factors from a given dataset and yield appealing data-analytic properties. In this work, we focus on the unsupervised disentanglement learning using GANs [27] on images, which brings substantial advancement in tasks such as semantic image understanding and generation, and potentially aids research on zero-shot learning and multi-modal learning [8, 30, 57, 118, 119, 47, 22, 23].

The most popular methods to tackle the unsupervised disentangling problem are based on GANs [27] or VAEs [52], and many instantiations of these [96, 1, 26, 2, 78] draw on information-theoretical concepts [97]. InfoGAN [13] seeks to maximize a Mutual Information (MI) lower bound between a sampled conditional vector and the generated data, with the expectation that the generator and discriminator will disentangle the vector with respect to the true underlying factors. In contrast, VAE-based approaches [12, 49, 24] attempt to optimize a Total Correlation (TC) [106] objective imposed on the inferred latent vector, which achieves disentanglement by encouraging inter-dimensional independence in the latent vector.

1.1 Unsupervised Image Disentanglement Learning

1.1.1 Problem Definition

We aim to accomplish both the task of disentangled feature extraction and human-controllable data generation in an unsupervised setting within the GAN framework. We define our problem as follows: A generator $G$ and a discriminator $D$ consist of multiple convolution layers. $G$ projects vector $v$ into image space $x \in \mathbb{R}^{3 \times h \times w}$ ($h, w$ for image shape), while $D$ takes image $x$ as
input and output one scalar value as real/fake score. For a continuous control vector \( c \) sampled from \( \text{uniform}(0, 1) \), we wish our generator \( G \) to be disentangled such that each dimension in \( c \) solely controls one feature of the generated data \( x = G(c, z) \) (\( z \) is the noise vector, the concatenation of \( c, z \) forms the vector \( v \)), and our feature extractor \( Q \) (mostly the discriminator \( D \) with a few more convolution layers on top that gives vector outputs) can emit a feature representation \( c' \), given \( x \), that is disentangled in the same way as \( c \).

### 1.1.2 Motivations and Challenges

TC-based VAE models have proven fruitful in disentangling. However, there is usually a trade-off between the degree of achievable disentanglement and the data-generating ability of VAE [49]. In practice, VAE struggles significantly when trained on higher-resolution images due to its restricted generative power. Furthermore, it only approximates the TC since both the marginal distribution of the learned latent representation and the product of its marginals are intractable in VAE, which makes the optimization process implicit and complicated. In contrast, with rapid advances in recent years [112, 84, 43], GANs have become more stable to train, and their generative power has become unparalleled even on high-resolution images.

Nonetheless, less attention has been paid to GANs in unsupervised disentanglement learning. Accordingly, we propose OOGAN, a novel framework based on GANs that can explicitly disentangle while generating high-quality images. The framework's components can readily be adapted to other GAN models.

Unlike in VAEs, where a latent vector has to be inferred, in GANs, noise is actively sampled as the latent vector during training. We exploit this property to enable OOGAN to directly learn a disentangled latent vector by means of one-hot vectors as latent representation to enforce exclusivity and encourage each dimension to capture different semantic features. This is achieved without sacrificing the continuous nature of the latent space through an alternating sampling procedure. We argue that our proposed OOGAN fully highlights the structural advantage of GANs over VAEs for disentanglement learning, which, to our knowledge, has not been exploited before.

### 1.1.3 Contributions

We achieve disentanglement in OOGAN through three contributions: 1) We propose an alternating one-hot sampling procedure for GANs to encourage greater disentanglement. 2) We adopt an orthogonal regularization on the model weights to better accompany our objective.
3) We identify a weakness in InfoGAN and related models with similar structure, which we summarize as the *compete and conflict issue*, and propose a model-structural change to resolve it. Moreover, we propose a compact and intuitive metric targeting the disentanglement of the generative part in the models. We present both quantitative and qualitative results along with further analysis of OOGAN, and compare its performance against VAEs and InfoGAN.

### 1.2 Mutual-Information based Multi-modal Learning

Taking advantage of the mutual-information maximization objective, we explore its potential to apply to the task of multi-modal learning. We propose Text and Image Mutual-Translation Adversarial Networks (TIME), a practical model that jointly learns a text-to-image (T2I) generator $G$ and an image captioning discriminator $D$ under a single GAN framework. While previous methods tackle the T2I problem as a uni-directional task and use pre-trained language models to enforce the image–text consistency, TIME requires neither extra modules nor pre-training. We show that the performance of $G$ can be boosted substantially by training it jointly with $D$ as a language model.

#### 1.2.1 Motivations and Challenges

While the T2I performance continues to advance \[90, 117, 11, 62, 110, 31\], the follow-up methods all share two common traits. First, they all adopt the same stacked structure of $G$ that requires multiple $D$s. Second, they all rely on the pre-trained DAMSM from AttnGAN to maintain the image–text consistency. However, these methods fail to take advantage of recent advances in both the GAN and NLP literature \[42, 44, 104, 19, 92\]. The rapidly progressing research in these two fields provides the opportunity to explore a substantial departure from previous work on text-to-image modeling. In particular, as StackGAN and follow-up works all depend on 1. a pre-trained text encoder for word and sentence embeddings, 2. an additional image encoder to ascertain image–text consistency, two crucial questions arise. First, can we skip the pre-training step and elegantly train the text encoder as part of $D$? Second, can we abandon the extra CNN (in the DAMSM module, which extracts image features) and use $D$ as the image encoder? If the answers are affirmative, two further questions can be explored. When $D$ and the text encoder are jointly trained to match the visual and text features, can we obtain an image captioning model from them? Furthermore, since $D$ is trained to extract text-relevant image features, will it benefit $G$ in generating more semantically consistent images?
1.2.2 Contributions

With these questions in mind, we present the Text and Image Mutual-translation adversarial nEtwork (TIME). To the best of our knowledge, this is the first work that jointly handles both text-to-image and image captioning in a single model using the GAN framework. Our contributions can be summarized as follows:

1. We propose an efficient model, Text and Image Mutual-Translation Adversarial Networks (TIME), for T2I tasks trained in an end-to-end fashion, without any need for pre-trained models or complex training strategies.

2. We introduce two techniques: 2-D positional encoding for a better attention operation and annealing hinge loss to dynamically balance the learning paces of $G$ and $D$.

3. We show that sentence-level text features are no longer needed for the T2I task, which leads to a more controllable T2I generation that is hard to achieve in previous models.

1.3 Posture and Identity Disentangled Image-to-image Translation

One popular objective for the image-to-image translation task is to independently control the coarse-level object arrangements (posture) and the fine-grained level styling (identity) of the generated image from two exemplar sources. We study a sub-task of image-to-image translation, which we call posture-and-identity disentangled translation. We view an image as a combination of multiple objects with varied scales, then define posture as the shape and composition of the larger-scale objects, and identity as both the color/texture of the objects and the shape of the finer-scale objects. Given two input images, one defines posture, and the other defines identity. Our goal is to develop a model that transfers the first image’s stance to the second one’s identity. This task has excellent potential in real-life applications such as illustration synthesis and motion driving for content-creation jobs. Note that a close task has been widely studied as content-style disentangled image synthesis, e.g. [16, 56, 60, 73, 70, 54, 46]. We find the terminology of “content-style” can be ambiguous sometimes, as both posture and identity has content and style related properties. In contrast, “posture-identity” are relatively precise to the task and datasets that we are working on, including human, animal, and animation faces.

1.3.1 Motivations and challenges

Training an image-to-image translation model usually requires image-level or set-level supervision, as represented by StarGAN-v2: [37, 105, 89, 116, 64, 61, 75, 16]. However, it is expensive
to obtain paired data and extra labeling information in real-world scenarios. More recently, fully unsupervised methods have gained more research interests such as CLUIT: [5, 56, 60], where no labeling information but only a set of images are needed. However, these methods lack controllability on disentangled image attributes and lack robustness on the degree of disentanglement among datasets. For our task of pose-identity translation, the limitations from prior works motivate us to seek a fully unsupervised method that is also controllable and robust. It is hard to train a model with precise control over posture and identity when there are no guidance labels of an image such as a segmentation or skeleton map, which are used by [86, 59, 53, 40]. Nonetheless, the need for such a segmentation label inspires us to develop a module that can learn a segmentation-like representation from the unlabeled image data in a self-supervised manner, therefore boosting the translation performance.

1.3.2 Contributions

With a carefully designed self-supervision training scheme, we show that VQSN can automatically learn meaningful object embeddings that provide useful posture information. The resulting model, which we call PIVQGAN, enables a precise control on posture and identity of the image generation and outperforms existing baselines in terms of visual quality and translation accuracy. Besides, we show a robust performance of PIVQGAN even on unseen out-of-domain images, where we further demonstrate its promising application beyond pose-identity translation, including zero-shot image style-transfer and semantic-aware image-editing.
Chapter 2
Related work

2.1 Unsupervised Image Disentangling Models

2.1.1 $\beta$-VAE-based models:

In the settings of $\beta$-VAE and its variants [30, 10, 12, 49, 25], a factorized posterior $p_\phi(z | x)$ is learned such that each dimension of a sampled $z_i$ is able to encode a disentangled representation of data $x$. The fundamental objective that $\beta$-VAE tries to maximize (also known as the Evidence Lower-Bound Optimization) is:

$$
L(\theta, \phi; x, z, \beta) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x | z)] - \beta D_{KL}(q_\phi(z | x) || p(z)),
$$

(2.1)

where $\beta > 1$ is usually selected to place stronger emphasis on the KL term for a better disentanglement learning. [10] motivate the effect of $\beta$ from an information-theoretical perspective, where the KL divergence term can be regarded as an upper bound that forces $q(z)$ to carry less information, thus becoming disentangled.

Follow-up research extends the explanation by deriving a Total Correlation from the KL term in the $\beta$-VAE objective, and highlights this TC term as the key factor to learning disentangled representations. Given a multi-dimensional continuous vector $z$, the TC quantifies the redundancy and dependency among each dimension $z_i$. It is formally defined as the KL divergence from the joint distribution $q(z_1, ..., z_n)$ to the independent distribution of $q(z_1)q(z_2)...q(z_n)$:

$$
L_{TC} = D_{KL}(\hat{q}(z)||\hat{q}(z)),
$$

(2.2)

where $\hat{q}(z) = \prod_{i=1}^{n} q(z_i)$. However, the TC term requires the evaluation of the density $q(z) = \mathbb{E}_{p(n)}[q(z|n)]$, which depends on the distribution of the entire dataset and usually is intractable. For the sake of a better optimization on the TC term, [58] propose TC-VAE, which uses a minibatch-weighted sampling method to approximate TC. [49] perform the same estimation using an auxiliary discriminator network in their Factor-VAE. Furthermore, [25] suggest a more generalized objective where the marginals $q(z_i)$ can be further decomposed into more TC terms,
in case each \( q(z_i) \) learns independent but entangled features, which leads to a hierarchically factorized VAE. [20] leverage the Gumbel Max trick [38] to enable disentangled learning of discrete features for VAE.

### 2.1.2 GAN-based models:

InfoGAN [13] reveals the potential of Generative Adversarial Networks [27] in the field of unsupervised disentanglement learning. In a typical GAN setting, a generator \( G \) and a discriminator \( D \) are trained by playing an adversarial game formulated as:

\[
\min_G \max_D \mathcal{L}_{GAN}(D, G) = \mathbb{E}_{p(x)}[\log(D(x))] + \mathbb{E}_{p(z)}[\log(1 - D(G(z))].
\] (2.3)

While this mini-max game guides \( G \) towards generating realistic \( x \) from noise \( z \) drawn from the isotropic Gaussian distribution, the variation of \( z \) often remains entangled. InfoGAN manages to make \( G \) learn a disentangled transformation from a latent code \( c \), which is concatenated to \( z \) before being fed to \( G \). InfoGAN achieves this by maximizing a Mutual Information (MI) lower-bound between \( c \) and the generated sample \( x = G(z, c) \), where the MI \( I(c, G(z, c)) \) can be calculated directly by matching \( c \) to \( \hat{c} = Q(G(z, c)) \), where \( Q \) is an auxiliary network that seeks to predict the sampled latent vector from \( x \). In practice, \( Q \) shares most weights with \( D \). However, such a lower-bound constraint only ensures \( c \) gains control over the generation process, but cannot guarantee any disentanglement as \( c \) increases its dimensionality, because this lower-bound does not encourage any independence across each dimension of \( c \).

A more recent GAN-based disentanglement work is the Information-Bottleneck-GAN [39]. However, it fails to take advantage of the GAN structure, instead trying to implicitly minimize the TC in the same way as \( \beta \)-VAE. The method requires an extra network that encodes noise \( z \) into a control vector \( c \) and lets the original \( G \) and \( D \) play the decoder’s role to reconstruct \( z \). This severely hurts the generation quality, since \( G \) starts the generation from \( c \), which has a much lower dimensionality than \( z \), and the increased network modules and loss objectives make the training scheme tedious and less likely to find the proper hyper-parameters that allow the model to converge.

### 2.2 Text-to-Image Multi-Modal Learning

Recent years have witnessed substantial progress in the text-to-image task [79, 85, 94, 93, 113, 109, 28] owing largely to the success of deep generative models [27, 52, 103]. [93] first
demonstrated the superior ability of conditional GANs to synthesize plausible images from text descriptions. StackGAN and AttnGAN then took the generation quality to the next level, which subsequent works built on [90, 117, 11, 62, 110, 31, 65]. Specifically, MirrorGAN [90] incorporates a pre-trained text re-description RNN to better align the images with the given texts, DMGAN [117] integrates a dynamic memory module on $G$, ControlGAN [62] employs a channel-wise attention in $G$, and SDGAN [110] includes a contrastive loss to strengthen the image–text correlation. In the following, we describe the key components of StackGAN and AttnGAN.

2.2.1 StackGAN as the Image Generation Backbone.

StackGAN adopts a coarse-to-fine structure that has shown substantial success on the T2I task. In practice, the generator $G$ takes three steps to produce a $256 \times 256$ image, where three discriminators ($D$) are required to train $G$. However, a notable reason for seeking an alternative architecture is that the multi-$D$ design is memory-demanding and has a high computational burden during training. If the image resolution increases, the respective higher-resolution $Ds$ can raise the cost particularly dramatically.

2.2.2 Dependence on Pre-trained modules.

While the overall framework for T2I models resembles a conditional GAN (cGAN), multiple modules have to be pre-trained in previous works. In particular, AttnGAN requires a DAMSM, which includes an Inception-v3 model [101] that is first pre-trained on ImageNet [18], and then used to pre-train an RNN text encoder. MirrorGAN further proposes the STREAM model, which is also an additional CNN+RNN structure pre-trained for image captioning.

Such pre-training has several drawbacks, including, first, the additional pre-trained CNN for image feature extraction introduces a significant amount of weights, which can be avoided as we shall later show. Second, using pre-trained modules leads to extra hyper-parameters that require dataset-specific tuning. For instance, in AttnGAN, the weight for the DAMSM loss can range from 0.2 to 100 across different datasets. Last but not least, empirical studies [90, 113] show that the pre-trained NLP components do not converge if jointly trained with the cGAN.

2.2.3 The Image-Text Attention Mechanism.

The attention mechanism employed in AttnGAN can be interpreted as a simplified version of the Transformer [104], where the three-dimensional image features (height $\times$ width $\times$ channel)
in the CNN are flattened into a two-dimensional sequence (seq-length × channel where seq-length = height × width). In this process, an image-context feature $f_{it}$ is derived via an attention operation on the reshaped image feature and the sequence of word embeddings. The resulting image-context features are then concatenated to the image features to generate the images. We will show that a full-fledged version of the Transformer can further improve the performance without a substantial additional computational burden.

2.3 Unsupervised Image-to-Image Translation

Well established image-to-image translation models mostly require labeled data to be trained, e.g. [37, 116, 74, 51, 36, 50, 75, 16]. Instance level paired images are required by models with an autoencoder training scheme like employed by [37, 89, 73], and coarse domain labels are needed for uni- or multi-modal translation models trained via cycle-consistency scheme as in [116, 36, 15].

However, the need for labels often becomes a bottleneck in real-world applications, limiting both the model performance and the application scenarios. Therefore, fully unsupervised image translation has drawn more research attention recently. One kind of unsupervised methods acquire pseudo-labels by image clustering [5, 6], another kind directly generates paired images via style-transfer from a small portion of labeled data [73, 73]. These methods lack robustness, as unintended translation results may occur when clustering algorithms fail to produce consistent clusters, or the synthesized pair data has low quality.

Focused on posture-and-identity image translation, methods without label-supervision are developed based on a reliable prior provided by StyleGAN [44, 44]. [56] proposed a diagonal attention structure which aims to encode the object shaping information. [60] developed a contrastive self-supervised training scheme, where a model is trained to implicitly learn and transfer visual features without explicit domain separation. However, the disentangled latent features from these methods are not well interpretable and controllable, thus lacks practicality in real world scenarios. Our work combines the strength from both perspectives. We propose a structural design that retains a more explicit object shaping representation, and leverages self-supervision to enforce the learning without labeling data.

2.3.1 Disentangled image synthesis

A broader task of disentangled image synthesis usually aims to learn one vector representation for the images in an image domain, where each vector entry controls one mutually independent
visual attribute. Autoencoder-based methods, e.g. [30, 10, 49, 25], approach disentanglement via the total-correlation objective, while GAN based models, e.g. [13, 69, 71], rely on maximizing a mutual-information loss. However, the vector representation has a limited capacity, and the per-attribute disentanglement is still hard to achieve. In this paper, we study the coarse-level disentanglement between two sets of attributes, which can be seen as the very first step in a hierarchically-disentangled task as proposed by [25, 99].

The coarse-level disentanglement is usually approached by the inductive biases of the layer designs and model structures. For example, [44, 41, 72] showed that controlling the attributes of a generated image from multiple scales in the generator is effective. [89] developed SPADE which applies different affine transformations to different spatial regions of the feature-map, to highlight different object shapes and diversify the object appearances. [56] proposed to learn “content” branch beside the original “style” branch in StyleGAN, and uses a one-channel spatial mask to guide the content information. [48] further replaced the vector representation in the “style” branch with spatial masks to embed better shaping information to the generated images.

We also bet on the structural bias to gain better control over the posture generation. However, unlike previous methods, our model learns the posture information in an interpretable, transferable, and editable way while disentangled well from the identity attributes.
Chapter 3
Disentangling GAN with One-Hot Sampling and Orthogonal Regularization

3.1 Introduction

Exploring the potential of GANs for unsupervised disentanglement learning, this chapter proposes a novel GAN-based disentanglement framework with One-Hot Sampling and Orthogonal Regularization (OOGAN). While previous works mostly attempt to tackle disentanglement learning through VAE and seek to implicitly minimize the Total Correlation (TC) objective with various sorts of approximation methods, we show that GANs have a natural advantage in disentangling with an alternating latent variable (noise) sampling method that is straightforward and robust. Furthermore, we provide a brand-new perspective on designing the structure of the generator and discriminator, demonstrating that a minor structural change and an orthogonal regularization on model weights entails an improved disentanglement. Instead of experimenting on simple toy datasets, we conduct experiments on higher-resolution images and show that OOGAN greatly pushes the boundary of unsupervised disentanglement.

A disentangled representation is one that separates the underlying factors of variation such that each dimension exclusively encodes one semantic feature [8, 49]. While the benefits of the learned representation for downstream tasks is questioned by [77], disentangling a Deep Neural Network (DNN) is still of great value in terms of human-controllable data generation, data manipulation and post-processing, and increasing the model interpretability. Moreover, disentanglement learning in an unsupervised manner can effectively highlight the biased generative factors from a given dataset, and yield appealing data-analytic properties. In this work, we focus on the unsupervised disentanglement learning using GANs [27] on images, which brings substantial advancement in tasks such as semantic image understanding and generation, and potentially aids research on zero-shot learning and reinforcement learning [8, 30, 57, 118, 119, 47, 22, 23].

The most popular methods to tackle the unsupervised disentangling problem are based on GANs [27] or VAEs [52], and many instantiations of these [96, 1, 26, 2, 78] draw on information-theoretical concepts [97]. InfoGAN [13] seeks to maximize a **Mutual Information** (MI) lower
Figure 3.1: OOGAN makes minimal changes upon a basic GAN. $c$ denotes the continuous control vector, $z$ is the noise vector, $c'$ is the feature representation of fake images.

bound between a sampled conditional vector and the generated data, with the expectation that the generator and discriminator will disentangle the vector with respect to the true underlying factors. In contrast, VAE-based approaches [12, 49, 24] attempt to optimize a Total Correlation (TC) [106] objective imposed on the inferred latent vector, which achieves disentanglement by encouraging inter-dimensional independence in the latent vector.

We achieve disentanglement in OOGAN through three contributions: 1) We propose an alternating one-hot sampling procedure for GANs to encourage greater disentanglement. 2) We adopt an orthogonal regularization on the model weights to better accompany our objective. 3) We identify a weakness in InfoGAN and related models with similar structure, which we summarize as the compete and conflict issue, and propose a model-structural change to resolve it. Moreover, we propose a compact and intuitive metric targeting the disentanglement of the generative part in the models. We present both quantitative and qualitative results along with further analysis of OOGAN, and compare its performance against VAEs and InfoGAN.

3.2 Proposed Method

Our approach accomplishes both the task of disentangled feature extraction and human-controllable data generation in an unsupervised setting within the GAN framework. We define our problem as follows: For a continuous control vector $c$ sampled from uniform$(0, 1)$, we wish our generator $G$ to be disentangled such that each dimension in $c$ solely controls one feature of the generated data $x = G(c, z)$ ($z$ is the noise vector), and our feature extractor $Q$ (mostly the discriminator $D$ with a few layers on top that gives vector outputs) is able to emit a feature representation
$c'$, given $x$, that is disentangled in the same way as $c$.

Our model is illustrated in Figure 3.1, and the complete training process is described in Algorithm 1. Similar to the design of InfoGAN, we let the feature extractor $Q$ be a sub-module that shares weights with the discriminator $D$. $Q$ takes the feature map of a generated image $G(c, z)$ as input and tries to predict the control vector $c$ used by the generator $G$. We describe the three components of our OOGAN framework in the following sub-sections.

**Algorithm 1: OOGAN training algorithm**

**Input**: generator $G$, discriminator $D$, feature extractor $Q$, batch size $B$, real data $X$, iteration $n$, control vector dimension $d$.

**Output**: well trained $G$, $D$, $Q$

for $i$ in $n$ iterations do

$x_{real} \leftarrow X$ ;

$z \leftarrow \mathcal{N}(0,1)$ ;

if $i$ is odd then

indices $\leftarrow \text{randint}(d)$ ;

$c \leftarrow \text{onehot}(\text{indices})$ ;

else

$c \leftarrow \text{uniform}(0,1)$ ;

$x_{fake} \leftarrow G(z, c)$;

$\text{loss}_d \leftarrow \text{relu}[1 - D(x_{real})] + \text{relu}[1 + D(x_{fake})]$ ;

update $D$ via $\text{loss}_d$ ;

$\text{loss}_g \leftarrow -D(x_{fake})$ ;

update $G$ via $\text{loss}_g$ ;

$\text{loss}_{MI} \leftarrow L_1(Q(x_{fake}), c)$ ;

if $i$ is odd then

$\text{loss}_{MI} \leftarrow \text{loss}_{MI} + L_{\text{cross-entropy}}(Q(x_{fake}), c)$ ;

update $G, Q$ via $\text{loss}_{MI}$ ;

end if

end if

end for
3.2.1 Alternating Continuous and One-hot Sampling

Previous methods of minimizing TC to achieve disentanglement have two limitations. First, due to the intractability, extra network modules and objectives have to be invoked to approximate TC, which leads to undesired hyper-parameter tuning, a non-trivial training regime, and a high computational overhead. Second, to optimize the derived TC objectives in VAE-based models, the data generation quality is sacrificed [49, 12], and can hardly perform well on higher-resolution image data. In contrast, in the GAN setting, the latent vector is sampled instead of inferred as in VAEs. This motivates us to approach disentanglement by deliberately sampling latent vectors that possess the property of inter-dimensional independence and training the networks using these sampled vectors.

To this end, we propose an alternating continuous-discrete sampling procedure: we alternate between sampling continuous $c$ from $\text{uniform}(0, 1)$ (as typically done in InfoGAN) and sampling $c$ as one-hot vectors. The one-hot vector $c$ implies that the generated image should only exhibit one feature, and, ideally, the prediction $c'$ from $Q$ should also be a one-hot vector. On both the $G$ and $Q$ sides, any presence of other features should be penalized, while alternating with continuous uniform sampling is necessary to ensure the continuity of the representation. Interestingly, such a one-hot sampling resembles a classification task. Therefore, we can jointly train $Q$ and $G$ directly via a cross-entropy loss. In such a process, $G$ is trained to generate images that possess the specified features and avoids retaining any other features, while $Q$ is trained to summarize the highlighted feature only in one dimension and refrain from spreading the feature representation into multiple dimensions.

Note that we treat $c$ as a continuous vector in the whole training process, and the alternating one-hot sampling can be seen as an regularizer for $G$ and $Q$. When we sample $c$ from $\text{uniform}(0, 1)$ as in InfoGAN, we ensure the correlation between $c$ and $x$ remains. Furthermore, when we interleave that with one-hot samples, the process can be interpreted as getting the extremely typical samples (those samples that lie on the boundary of the uniform distribution) for the model. We argue that sampling data at the distribution boundaries makes the model pay more attention to these boundaries, yielding a clearer distribution shape highlighting the semantics of these boundary factors. These typical samples are vital for the model to learn inter-dimension exclusivity, as a one-hot $c$ regularizes $G$ to generate images with only one factor and $Q$ to only capture this one factor.

In other words, alternating one-hot sampling and uniform sampling results in a more desirable prior distribution for disentangling GANs, which provides much more typical samples on
the margin than a single uniform distribution. Such an alternating procedure, which injects the
categorical sampling (i.e., the one-hot sampling) into a continuous $c$, makes it possible that $c$
gains continuous control over the generation process while simultaneously paying more attention
to those typical examples, and therefore achieves better disentanglement.

Formally, our complete objective for OOGAN is:

$$\min_G \max_D \mathcal{L}_{OOGAN}(D, G) = \mathcal{L}_{GAN}(D, G) + \lambda I(c_{\text{continuous}}, G(c_{\text{continuous}}, z)) + \gamma \mathcal{L}_{\text{Cross-Entropy}}(Q(G(c_{\text{one-hot}}(d), z)), c_{\text{one-hot}}(d)).$$

(3.1)

Despite the lack of any TC terms in our objective, the one-hot sampling still ensures that we
have a well-disentangled feature extractor $Q$ and generator $G$ that learn features with no overlap
between each dimension in $c$, without any approximations and extra network modules involved.

### 3.2.2 Compete-Free Generator

InfoGAN [13] and many conditional-GAN variants leverage an auxiliary vector $c$ that is con-
catenated with noise $z$ before being fed into $G$, with the expectation that $c$ carries the human-
controllable information. From a size perspective, the dimensionality of $z$ is usually much more
significant than $c$ ($z$ typically has around a hundred dimensions, while $c$ has in the order of
10). Intuitively, $c$ will have much less impact in the generation process. With the objective
of the unsupervised disentanglement learning, the large portion of influence $z$ is taking in the
generation process is undesirable, which we refer to as the competing and conflicting issue.

Usually, a disentangled feature learned by $c$ can also be entangled in $z$. During the training
process, if $c$ with $c_i$ holding a high signal on a certain feature is paired with some $z$ with many
dimensions holding the same feature with a conflict signal, this signal, entangled in $z$, will
easily overpower $c$. Thus, the generated images will not present $c_i$'s signal. Such a conflict will
discourage $c$ from mastering the learned feature and cause it to stray away to some easier-to-
achieve but less distinct features. An example is shown in Figure 3.2. More discussion can be
found in the experiments section.

To avoid the aforementioned competing and conflicting issue, we propose a new Compete-
Free Design of the generator's input block, which switches the role between $c$ and $z$ by letting
$c$ control the fundamental content even when the dimensionality of $c$ is low, and ensures that $z$
has limited influence in the generation process.

To start with, we project the low-dimensional control vector $c$ into a multi-channel $4 \times 4$
feature map by a convTransposed layer. Then, we add this feature map to a learned constant
Figure 3.2: Latent traversals trained on CelebA to showcase the competing and conflicting issue. The images are from the same set of $(z, c)$ on one fixed dimension of $c$ after different training iterations. We observe that InfoGAN begins to capture what appears to be a “wearing glasses” feature at a very early stage, but discards it during training in all dimensions of $c$. In contrast, when OOGAN begins to capture this feature, it consistently masters it in the end.
tensor with the same dimensionality.

The weights for the constant tensor are randomly initialized before training and will be used for all generations. These weights are trained via back-propagation just like all other model weights. This learned constant can be regarded as an additive bias that is learned from the dataset, and is necessary since it is responsible for representing the features that are not captured in \( c \). Ideally, when given a \( c \) with all zeros as input, this constant should let the generator output the most “neutral” \( x \). In our experiments, we find such constant important for providing a stabilized learning process. It makes OOGAN faster to converge to the disentangled factors. Intuitively, one can imagine this constant as placing an anchor at the center of the target distribution, such that all latent factors can expand towards different directions. This behavior encourages the model to focus on learning the correlation between \( c \) and the generated images. Without this ”constant”, OOGAN will still work as it is but will be slower to converge for \( c \).

To encourage the variance and complement the details for a higher-quality generation, the traditional noise \( z \) is still taken into the generator, but only after the 8 \( \times \) 8 feature map level. To prevent \( z \) from causing the competing and conflicting issue, we leverage an attention mask generated from \( c \) on the features from \( z \), which means that only the approved part of \( z \) by \( c \) can join in the generation process. Different layers in CNNs have been studied extensively [43], where the first few layers tend to generate fundamental compositions, and higher layers only refine the details. So our design makes \( c \) more natural to control the key generative factors without the interference of \( z \). The design details are illustrated in Figure 3.3-(a).

Our generator design resembles the one proposed for StyleGAN [43], as we both base the generation on a fixed multi-dimensional feature map instead of an input vector \( z \), and take \( z \) as input only in later layers. As claimed by Karras et al., such a design leads to a better separation in the data attributes and a more linear interpolation along latent factors. However, both the motivation and structure details are different. The disentanglement we study here is a more strict term than what Karras et al. used. The fundamental difference is that the fixed weights in our proposed model only serve as a supportive bias, and will be directly changed by \( c \), while in StyleGAN the fixed weights are solely used to start the image generation process.

### 3.2.3 Orthogonal Regularized & Grouped Feature Extractor

To learn a disentangled representation, we propose a new structure of \( Q \) that uses grouped convolutions [55, 115] instead of traditional fully connected ones, with an orthogonal regularization on the weights among every convolution kernel. The intuition is, since we hope that \( Q \) will be a
Figure 3.3: Model structures: (a) Input block of the compete-free $G$. (b) Orthogonal-regularized grouped $Q$

highly disentangled feature extractor, a fully connected (FC) design is not favorable, since, in a FC convolution, each feature prediction has to take into consideration all the feature maps from the previous layer. A grouped convolution, on the other hand, can focus its decision making on a much smaller group of previous features, and may thus be less distracted by potentially irrelevant features.

To make sure that each group is indeed attending to different features, we impose an additional loss function on the weights of the convolutional layers to enforce the orthogonality between different kernels. Weight orthogonality in DNNs has been studied [9, 33, 7]. However, these studies each focused on different tasks, and none of them revealed the potential for disentanglement learning.

The orthogonal regularization we use is straightforward: during each forward pass of the OOGAN, compute and minimize the cosine similarity between every convolutional kernel. With grouped feature extraction and orthogonal regularization, $Q$ structurally more easily captures diversified features in each dimension. Note that the group design is not only applicable to convolutional layers but also to grouped linear layers or other weights indicated as “transformer” in Figure 3.3-(b). Similarly, the orthogonal regularization can be applied on weights of all these grouped layers.

3.3 Perceptual Diversity Metric

Quantitative metrics for the disentanglement are mostly proposed in VAE-based works and for simulated toy datasets with available ground truth information. [30] suggest training a low-capacity linear classifier on the obtained latent representations of the simulated data from the trained encoder, and report the error rate of the classifier as the disentanglement score of the generative model. [49] argue that the introduction of an extra classifier could lead to undesirable
uncertainties due to the increased hyper-parameters to tune. Thus, they favor a majority-vote classifier that is achievable directly from the latent representations.

We concur with [49] in arguing that, to the best of our knowledge, there is no convincing metric for disentanglement on a dataset for which no ground truth latent factor is provided. Therefore, we propose a method that is capable of relatively evaluating partial properties of a disentangling model when certain conditions are satisfied.

Our intuition is that if a generative model is well-disentangled, then varying each dimension of the controlling vector $c$ should yield different feature changes of the generated data $x$. Suppose the feasible value range for $c$ is $[a, b]$, and for a pair of $(c^o, i, j)$ where $c^o$ is a uniformly sampled vector and $i$ and $j$ are two randomly selected indices, we get $c^i$ by setting $c^o[i] = b$ and $c^o[j] = a$, and $c^j$ by setting $c^o[j] = b$ and $c^o[i] = a$. Given the fact that $i$ and $j$ each control different factors, we expect $x^i = G(c^i)$ and $x^j = G(c^j)$ to be different. Therefore, we can use a pre-trained VGG [98] model $V$ to extract the feature map of $x^i$ and $x^j$, and report their $L_1$ distance as the disentanglement score, with a higher $L_1$ distance indicating dimensions $i$ and $j$ are more independent. The final score of this proposed perceptual diversity metric will be the average score of many samples of paired $(c^o, i, j)$. A formalized algorithmic procedure is in 2:

We argue that such a metric can adequately reflect the separability and diversity of the learned factors, especially when used for comparing similarly structured models on high-resolution datasets, where higher diversity should already be considered better, and on datasets in which latent factors are known to control a good amount of visual differences. As shown in Figure 3.4, the proposed metric can efficiently capture the disentangle performance in terms of how diversified each dimension is in $c$.

Karras et al. [49] in their StyleGAN work propose the Perceptual Path Length criterion, which is a pairwise image difference between two DNN embeddings of any small cut along with interpolation, and a minor change between the cuts indicates more focused information. However, such a perceptual pairwise distance only reflects the linearity of the interpolated images but cannot determine if a linear change entangles multiple factors or not. Additionally, it has a failure case that when there is no change along with the interpolation, this dimension will get the best score, but actually, nothing is learned.

On the other hand, our proposed Perceptual Diversity metric measures how each dimension in $c$ encodes different features that will reflect on the generated images. Moreover, in our experiment on celebA data, we fine-tune the VGG model on the celebA dataset with the provided 40 attributes describing the visual features, to make the VGG model better extract the on-point features, thus making the final score more meaningful.
Algorithm 2: Metric: Perceptual diversity L1 difference

**Input**: generative model $G$, pre-trained VGG model $V$, sample times $n$, latent dimension $d$, latent variable range $[-k, k]$.

**Output**: score $s$

```plaintext
score = 0;

for $n$ iterations do
    Sample $c \in \mathbb{R}^d$ from Uniform$(0,1)$;
    Sample $i, j$ from RandomInt$(0, d)$;
    $c_1 = c$;
    $c_2 = c$;
    $c_1[i] = -k, c_1[j] = k$;
    $c_2[i] = k, c_2[j] = -k$;
    $feat_1 = V(G(c1))$;
    $feat_2 = V(G(c2))$;
    score = score + L1_distance(feat_1, feat_2)

return score/n;
```

Note that our proposed metric involves computing the paired L1 difference between two dimensions, while setting one dimension to the maximum value $b$, we will also set the value on the other dimension to the minimum value $a$, to better highlight the visual differences between the dimensions. Since a $c^o$ is uniformly sampled, the original value in $c^o$ on $i$ and $j$ dimension could already be high and close to $b$, so to ensure the two samples $c^i$ and $c^j$ are comparable, we have to force the values in one dimension to $a$.

### 3.3.1 Limitations:

The perceptual diversity metric should not be generally used to compare differently structured models, and cannot solely capture the disentanglement ability of a model. First, the L1 distance between feature maps is not an absolute measure. For example, a VAE model $A$ produces blurry images that could lead to a lower value from this metric compared to a GAN model $B$, where images are sharp and high-contrast, but this does not necessarily imply that $A$ disentangles worse than $B$.
Figure 3.4: Generated images for CelebA: In each group, the left-most image is generated from a randomly sampled $c$, and the following ones are generated by changing the value of each dimension in $c$ to 1. (a) OOGAN exhibits greater visual differences among each dimension, reflecting its ability to learn diverse latent factors. (b) Without the proposed one-hot sampling, OOGAN still manages to learn some distinguishable features, reflecting the advantage of its structural design. (c) The 4 top right images show that the learned features for an InfoGAN have a large overlap across the latent dimensions in $c$, lacking proper disentanglement.

Hence, we only use this metric on the CelebA dataset for comparisons within GAN models, where the aforementioned drawbacks are irrelevant. In such cases, this metric remains valuable in providing an intuitive and direct measure of how well the generative part of the models disentangle.

3.4 Experiments

We conduct quantitative and qualitative experiments to demonstrate the advantages of our method on several datasets. First, we perform quantitative experiments on the dSprites datasets [80] following the metric proposed by [49]. After that, we show the superiority of OOGAN in terms of generating quality images while maintaining competitive disentanglement compared to VAE-based models on CelebA [76] and 3D-chair [4] data. Based on the disentangling benchmark guidance from [21], we also present an elaborated learned-factor identification experiment to showcase the effectiveness of OOGAN and validate our compete-and-conflicting issue observations. Finally, we conduct an ablation study on the proposed components in OOGAN with our metric. All the model structures, training details, and more qualitative results are given in the Appendix, and all the code for our experiments is also submitted.
3.4.1 Quantitative Results on dSprites:

Several quantitative metrics have been proposed on the dSprites dataset [30, 49, 21, 12]. While these metrics achieve a thorough evaluation of the disentanglement abilities of the feature-extractor (i.e., the encoder in VAE and Q in GANs), they pay no attention to the generative part of the models. Therefore, we only select [49]'s metric for its intuitiveness and simplicity to demonstrate our model’s competitiveness on the feature extractor’s end.

For all the models, we follow the same setup as [49] and [39]. Due to the simplicity of the dataset, we train all the GAN models with the “instance noise technique” introduced by [100] to get stable and good quality results.

As can be seen from Table 3.1 and Figure 3.5, our proposed OOGAN genuinely does a better job on both the feature extractor and generator parts. While Factor-VAE is only able to disentangle three out of the five ground truth factors effectively, OOGAN retrieves all the generative factors and manages to put the variables of the discrete factor “shape” into different dimensions. Additionally, we would like to highlight the robustness of our model, where varying the hyper-parameters of $\lambda$ (1 to 5) and $\gamma$ (0.2 to 2) in our loss function always yields consistent performance.

3.4.2 Qualitative Results on 3D Chairs:

On the 3D Chairs data, we use $64 \times 64$ RGB images with batch size 64 for all training runs. To demonstrate the robustness and performance of OOGAN in generating higher-quality images and potentially learning more latent factors, we consider the dimensionality of $c$ for up to 16,
where previous works only experiment on a smaller dimensionality such as 6.

In Factor-VAE, when we increase the dimensionality of $c$ from 6 to 16, it struggles to disentangle at a similar quality, and the reconstruction ability is severely sacrificed. In contrast, our model is not affected by an increase of the dimensionality, and apart from learning somewhat more obvious features such as scale and azimuth, our model also discovers several exciting features that have never been reported in previous work. For example, Figure 3.6-(c) shows a linear transformation of different back styles and leg thickness of the chairs, and Figure 3.6-(d) shows that our model successfully disentangles discrete features such as “color” and “chair type” without any additional tweaks and tricks, for which additional tweaks such as various approximation approaches would have to be incorporated in a VAE approach.

### 3.4.3 Disentangling at a Higher Resolution on CelebA:

We consider OOGAN as a suite of three modules that can be plugged into any GAN frameworks, and it is orthogonal to other disentanglement approaches based on GANs such as IBGAN. In other words, it can be incorporated into other methods and inherits the breakthroughs made
Figure 3.7: Latent traversals trained on CelebA

in GANs [112, 84, 43]. Therefore, we focus on demonstrating the advantages OOGAN has over VAE-based models in qualitative experiments. The comparison with InfoGAN will be presented as quantitative results in an ablation study.

On the CelebA dataset, while previous work operates at a resolution of $64 \times 64$, we train all the models on the resolution of $256 \times 256$ to showcase the advantage of OOGAN and expose a shortcoming of the VAE-based models. Figure 3.4-(a) shows the images trained in a plain DCGAN manner [91] and Figure 3.7-(a) shows the images trained in a progressively up-scaling manner [42], demonstrating a strong ability to disentangle while maintaining a high image quality. On the other hand, VAE-based models deteriorate when reconstructing high-contrast images, and are unable to maintain the same disentangle performance as the resolution increases. Thanks to the detail richness of the generated images, OOGAN can discover more interesting facial features such as “chin” and “cheek”, which no VAE-based models have achieved.

**Learned attribute analysis:** To provide a more transparent breakdown on what is learned [21], we train 40 binary classifiers on the 40 provided visual attributes from the CelebA dataset, each only predicting one attribute. Then we use these classifiers to monitor the generated images across the training iterations of InfoGAN and OOGAN.

In Figure 3.8, there are 40 different colors on the lines, each color representing an attribute, with 16 lines per color representing the 16 dimensions in $c$. In terms of sampling $c$, we set one dimension’s value to one and sample the remaining dimensions from $\text{uniform}(0, 1)$, repeating the
same operation for all dimensions. For InfoGAN, the lines with the same color stick together and have the same tendency to change, which means different dimensions in $c$ are learning similar factors. In contrast, for OOGAN, we observe that lines in the same color (the same attribute) develop differently during the iterations. Moreover, at each iteration, the lines in the same color get different prediction scores, which implies that different dimensions in $c$ are learning different factors.

We then average the curves of each attribute into one to show the prediction score for each ground truth attribute in Figure 3.9, which confirms the competing and conflicting issue. If a prediction score is constantly increasing, this means that $c$ is improving its ability to represent the respective attribute. For InfoGAN, the attribute predictions show rising and falling fluctuations, a sign of an unstable learning process. In contrast, OOGAN steadily increases the prediction score for some visual attributes.

### 3.4.4 Ablation Studies:

Based on the plain InfoGAN setting, we conduct ablation studies on the effectiveness of our proposed three modules quantitatively using our proposed metric, with InfoGAN as the baseline. The experiments are conducted only on learning continuous factors, as InfoGAN already performs well in disentangling categorical latent variables.

There are two types of $Q$ we can choose from when training OOGAN or InfoGAN. A deterministic $Q$ will try to directly output $c'$, which is considered as a reconstruction of $c$; and a probabilistic $Q$ will assume that each dimension of $c_i$ is from a Gaussian and try to output the mean and standard deviation of that distribution given different input images. To optimize a
When the dimensionality of the latent factors $c$ goes beyond 6, InfoGAN fails to disentangle them, which is also confirmed in previous work [49, 39]. As shown in Figure 3.4, most dimensions in InfoGAN produce similar images as many features remain entangled. In the meantime, OOGAN has a better tendency to learn disentangled representations thanks to its structural design, and a direct objective to learn independent features driven by the proposed alternating one-hot sampling.

For the proposed perceptual diversity metric, we fine-tune the VGG model on the CelebA
Table 3.2: Disentanglement using Perceptual Diversity metric

<table>
<thead>
<tr>
<th>Model</th>
<th>Score</th>
<th>Cos-simil. in Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>InfoGAN</td>
<td>2.39 ±0.03</td>
<td>0.21 ± 0.01</td>
</tr>
<tr>
<td>OOGAN w/o One-hot</td>
<td>2.44 ±0.05</td>
<td>0.09 ± 0.03</td>
</tr>
<tr>
<td>OOGAN w/o Ortho-reg</td>
<td>2.65 ±0.05</td>
<td>0.21 ± 0.01</td>
</tr>
<tr>
<td>OOGAN w/o Compt-free G</td>
<td>2.69 ±0.03</td>
<td>0.09 ± 0.03</td>
</tr>
<tr>
<td>OOGAN</td>
<td>2.77 ±0.06</td>
<td>0.09 ± 0.03</td>
</tr>
</tbody>
</table>

dataset with the provided 40 facial attributes, to make it more sensitive to the visual attributes.

As shown in Table 3.2, the one-hot sampling makes the most substantial contribution, while orthogonal-regularized Q and compete-free generator also provide significant improvements. The averaged cosine similarity among the weights is effectively minimized with the proposed orthogonal regularization. In Figures 3.10-(a) and 3.10-(b), we train the models with a deterministic Q that directly attempts to reconstruct $c$, and plot the L1 distance between the sampled $c$ and predicted $c'$ (the L1 distance for one-hot $c$ is not used as an objective loss to train InfoGAN). Note how InfoGAN’s L1 loss is similarly minimized when $c$ is uniformly sampled, but struggles to decrease when $c$ is one-hot, which means that the output $c'$ of InfoGAN is highly correlated (there are correlated latent factors encoded into multiple dimensions, implying poor disentanglement), while OOGAN’s $c'$ is not. In Figure 3.10-(c), we train the models with probabilistic Q and estimate TC following the method from [12]. The TC from InfoGAN remains high, while OOGAN can maintain a low TC consistently, which shows the effectiveness of our method.

3.4.5 More Qualitative Results
Figure 3.11: OOGAN based on Vanilla GAN. Latent traversals along different dimensions.
Figure 3.12: OOGAN based on Vanilla GAN. Latent traversals along different dimensions.

Figure 3.13: OOGAN based on StyleGAN. Latent traversals along different dimensions.
Figure 3.14: OOGAN based on StyleGAN. Latent traversals along different dimensions.

Figure 3.15: OOGAN on MNIST dataset, the 4 sections are 4 selected indices from the discrete vector (categorical one-hot vector), inside each section are latent traversals on the same 6 continuous dimensions.
Chapter 4

Text and Image Mutual-Translation Adversarial Networks

4.1 Introduction

There are two main aspects to consider when approaching the text-to-image (T2I) task: the image generation quality and the image–text semantic consistency. The T2I task is commonly modeled by a conditional Generative Adversarial Network (cGAN) [83, 27], where a Generator (G) is trained to generate images given the texts describing the contents, and a Discriminator (D) learns to determine the authenticity of the images, conditioned on the semantics defined by the given texts.

To address the first aspect, [113] introduced StackGAN by letting G generate images at multiple resolutions, and adopted multiple Ds to jointly refine G from coarse to fine levels. StackGAN invokes a pre-trained Recurrent-Neural-Network (RNN) [32, 82] to provide text conditioning for the image generation. To approach the second aspect, [109] take StackGAN as the base model and propose AttnGAN, which incorporates word embeddings into the generation and consistency-checking processes. A pre-trained Deep-Attentional-Multimodal-Similarity-Model (DAMSM) is introduced, which better aligns the image features and word embeddings via attention mechanism.

While the T2I performance continues to advance [90, 117, 11, 62, 110, 31], the follow-up methods all share two common traits. First, they all adopt the same stacked structure of G that requires multiple Ds. Second, they all rely on the pre-trained DAMSM from AttnGAN to maintain the image–text consistency. However, these methods fail to take advantage of recent advances in both the GAN and NLP literature [42, 44, 104, 19, 92]. The rapidly progressing research in these two fields provides the opportunity to explore a substantial departure from previous works on text-to-image modeling. In particular, as StackGAN and follow-up works all depend on: 1. a pre-trained text encoder for word and sentence embeddings; 2. an additional image encoder to ascertain image–text consistency; two important questions arise. First, can we skip the pre-training step and elegantly train the text encoder as part of D? Second, can we abandon the extra CNN (in the DAMSM module which extracts image features) and use D as
This bird has wings that are black and has a yellow belly.

This bird has wings that are brown and has a white belly.

This bird is black all over, and has a black tarsus and feet with some white.

AttnGAN

TIME

TIME

TIME

TIME

TIME

1. This bird has wings that are brown and has a white belly.
2. A brown bird with white and brown feathered belly, and a small beak.
3. A black bird with slick feathers, and a black bill.
4. A white seagull with yellow beak has black wings and tail, and pink legs.
5. A brown bird with white and brown feathered belly, and a small beak.
6. A black bird with slick feathers, and a black bill.
7. A white bird with black and has a very orange beak.
8. A black bird with black and has a white belly.
9. A white bird with black and has a very orange beak.
10. A black bird with black and has a white belly.

(a) Real caption
(b) Worst 2 out of 10 random synthesised samples
(c) Best 3 out of the same 10 samples
(d) Real sample
(e) Re-caption samples by Discriminator

Figure 4.1: Text-to-image results of TIME on the CUB dataset, where $D$ works as a stand-alone image-captioning model.

the image encoder? If the answers are affirmative, two further questions can be explored. When $D$ and the text encoder are jointly trained to match the visual and text features, can we obtain an image captioning model from them? Furthermore, since $D$ is trained to extract text-relevant image features, will it benefit $G$ in generating more semantically consistent images?

With these questions in mind, we present the Text and Image Mutual-translation adversarial network (TIME). To the best of our knowledge, this is the first work that jointly handles both text-to-image and image captioning in a single model using the GAN framework. Our contributions can be summarized as follows:

1. We propose an efficient model: Text and Image Mutual-Translation Adversarial Networks (TIME), for T2I tasks trained in an end-to-end fashion, without any need for pre-trained models or complex training strategies.
2. We introduce two techniques: 2-D positional encoding for a better attention operation and annealing hinge loss to dynamically balance the learning paces of $G$ and $D$.
3. We show that the sentence-level text features are no longer needed in T2I task, which leads to a more controllable T2I generation that is hard to achieve in previous models.
4. Extensive experiments show that our proposed TIME achieves superior results on text-to-image tasks and promising results on image captioning. Fig. 4.1-(c) showcases the superior synthetic image quality from TIME, while Fig. 4.1-(e) demonstrates TIME’s image captioning capability.
Figure 4.2: (a) The StackGAN structure that serves as the backbone in SOTA T2I models [113, 109, 90, 117, 11, 62, 31]. (b)&(c) Representative models build upon StackGAN, with red parts indicating modules that require pre-training. Note that our proposed model TIME does not: 1. require pre-training the red modules in (b) and (c); 2. require multiple discriminators (the green modules) in (a).

4.2 Prior Works

In this section, we want to highlight the differences between TIME and previous work.

4.2.1 StackGAN as the Image Generation Backbone

StackGAN adopts a coarse-to-fine structure that has shown substantial success on the T2I task. The generator $G$ takes three steps to produce a $256 \times 256$ image as shown in Figure 4.2-(a). In stage-I, a $64 \times 64$ image with coarse shapes is generated. In stage-II and III, the feature maps are further up-sampled to produce more detailed images with better textures. Three discriminators are required, where the lowest-resolution $D$ guides $G$ with regard to coarse shapes, while localized defects are refined by the higher-resolution $D$s.

However, there are several reasons for seeking an alternative architecture. First, the multi-$D$ design is memory-demanding and has a high computational burden during training. As the image resolution increases, the respective higher-resolution $D$s can raise the cost dramatically. Second, it is hard to balance the effects of the multiple $D$s. Since $D$s are trained on different resolutions, their learning paces diverge, and can result in conflicting signals when training $G$.

In our experiments, we notice a consistently slower convergence rate of the stacked structure.
compared to a single-\(D\) design.

### 4.2.2 Dependence on Pre-trained modules

While the overall framework for T2I models resembles a conditional GAN (cGAN), multiple modules have to be pre-trained in previous works. As illustrated in Figure 4.2-(b), AttnGAN requires a DAMSM, which includes an Inception-v3 model [101] that is first pre-trained on ImageNet [18], and then used to pre-train an RNN text encoder. MirrorGAN further proposes a pre-trained STREAM module as shown in Fig. 4.2-(c).

Such pre-training has a number of drawbacks, including, first and foremost, the computational burden. Second, the additional pre-trained CNN for image feature extraction introduces a significant amount of weights, which can be avoided as we shall later show. Third, using pre-trained modules leads to extra hyper-parameters that require dataset-specific tuning. For instance, in AttnGAN, the weight for the DAMSM loss can range from 0.2 to 100 across different datasets. While these pre-trained models boost the performance, empirical studies in MirrorGAN and StackGAN show that they do not converge if jointly trained with cGAN.

### 4.2.3 Difference to MirrorGAN

Our model shares a similar idea with MirrorGAN in “matching the image captions for a better text-to-image generation consistency”. However, there are substantial difference between our work and MirrorGAN. First, while MirrorGAN focused on the T2I task and uses a fixed captioning model to support the training of \(G\), our idea can better be described as “image–text mutual translation”. The objective is not merely to train a T2I \(G\), but also to train an image captioning model at the same time. Second, the study in MirrorGAN suggests that the image-captioning module with extra CNN and RNN has trouble converging if trained together with the T2I GAN, and thus can only be pre-trained. However, in our work, we show that the image-captioning module can be jointly trained under the same framework, and the joint training leads to better performance for \(G\) while does not hurt the performance on captioning for \(D\). Third, we believe “image–text mutual translation” is a topic that can attract substantial research attention and that it is particularly intriguing, as humans can easily relate visual and lingual information together.

TIME differs from MirrorGAN in several respects to realize the joint training. i) MirrorGAN uses LSTMs to model the language, which suffers from vanishing gradient issues, in particular that the discrete word embeddings are not very compatible with image features, which are
more continuous when trained jointly. In contrast, Transformer is more robust in dealing
with the combined features of the text and image modalities. ii) MirrorGAN tries to train
directly on word embeddings, which is common in T2I tasks. In contrast, in our model, the
word embeddings that the CNNs (G and D) take are not the initial word embeddings. We
have a text encoder to endow the word sequence embeddings with contextual signals, making
the resulting sentence representations much smoother, thereby facilitating training. iii) The
language part of TIME is only trained with the discriminator on the image captioning task.
The embeddings are not trained to help the Generator for better image quality. This focused
objective frees the text encoder from potentially conflicting gradients, aiding convergence, and
making mutual translation feasible with our model.

4.3 The Motivation of Mutual Translation

One may ask that since training the text-to-image model already achieves fruitful results with
a pre-trained NLP model, is it necessary to explore the joint-training method? We can answer
this question from several aspects.

First, a pre-trained NLP model is not always available given some image datasets. In cases
where the given texts do not have a pre-trained NLP model, one can save the separated pre-
training time and get a model that translates towards both directions with TIME. In case a
pre-trained NLP model is available, it is still not guaranteed that the fixed word embeddings
are the best for training the image generator. Tuning the hyper-parameters (such as weights
of loss objectives from the pre-trained NLP model) for the pre-training methods can be very
costly and may not be optimal.

Second, under the GAN framework, balancing the joint training between the Discriminator
D and Generator G is vital. G is unlikely to converge if trained with a fixed D. In the text-to-
image task, the pre-trained NLP model serves as a part of D that provides authenticity signals
to G. Using a pre-trained NLP model is equivalent to fixing a part of D, which undermines the
performance of the whole training schema as a GAN. Instead, the joint training in TIME does
not have such restrictions. The NLP parts in TIME learn together with G and dynamically
adjust the word embeddings to serve the training best, leading to better image synthesis quality.

Finally, mutual translation itself can be a crucial pre-training method, which is also studied
in [34, 66]. As we show in the chapter, the NLP models learned in TIME have a promising
performance on downstream vision-language tasks. Therefore, instead of pre-training only on
texts, mutual translation between image and text itself has the potential to be a powerful
4.4 Proposed Method

In this section, we present our proposed approach. The upper panel in Fig. 4.3 shows the overall structure of TIME, consisting of a Text-to-Image Generator $G$ and an Image-Captioning Discriminator $D$. We treat a text encoder $Enc$ and a text decoder $Dec$ as parts of $D$. $G$’s Text-Conditioned Image Transformer accepts a series of word embeddings from $Enc$ and produces an image-context representation for $G$ to generate a corresponding image. $D$ is trained on three kinds of input pairs, consisting of captions $T^{\text{real}}$ alongside: (a) matched real images $I_{\text{match}}$; (b) randomly mismatched real images $I_{\text{mis}}$; and (c) generated images $I_{\text{fake}}$ from $G$.

4.4.1 Model Structures

Aggregated Generator To trim the model size of the StackGAN structure, we present the design of an aggregated $G$ as shown in the upper panel of Fig. 4.3. $G$ still yields RGB outputs
at multiple resolutions, but these RGB outputs are re-scaled and added together as a single aggregated image output. Therefore, only one $D$ is needed to train $G$.

**Text-Conditioned Image Transformer** While prior works \cite{112, 109} show the benefit of an attention mechanism for the image generative task, none of them dive deeper towards the more comprehensive “multi-head and multi-layer” Transformer design \cite{104}. To explore a better baseline for the T2I task, we redesign the attention in AttnGAN with the Text-Conditioned Image Transformer (TCIT) as illustrated in Fig. 4.3-(a). In Fig. 4.4, we show three main differences between TCIT and the form of attention that is widely used in previous T2I models such as AttnGAN. All the attention modules take two inputs: the image feature $f_i$, and the sequence of word embeddings $f_t$, and gives one output: the revised image feature $f_{it}$ according to the word embeddings $f_t$.

First, Fig. 4.4-(a) shows the attention module from AttnGAN, where the projected key ($K$) from $f_t$ is used for both matching with query ($Q$) from $f_i$ and calculating $f_{it}$. Instead, TCIT has two separate linear layers to project $f_t$ as illustrated in Fig. 4.4-(b). The intuition is, as $K$ focuses on matching with $f_i$, the other projection value $V$ can better be optimized towards refining $f_i$ for a better $f_{it}$. Second, TCIT adopts a multi-head structure as shown in Fig. 4.4-(c). Unlike in AttnGAN where only one attention map is applied, the Transformer replicates the attention module, thus adding more flexibility for each image region to account for multiple words. Third, TCIT stacks the attention layers in a residual structure as in certain NLP models \cite{19, 92} as illustrated in Fig. 4.4-(d), for a better performance by provisioning multiple attention layers and recurrently revising the learned features. In contrast, previous GAN models (AttnGAN, SAGAN) adopt attention only in a one-layer fashion.

**Image-Captioning Discriminator** We treat the text encoder $Enc$ and text decoder $Dec$
Before 2-D embedding

After 2-D embedding

Embedding the first half channels with y-axis spatial information

Embedding the second half channels with x-axis spatial information

(a) Features at faraway regions can have similar values, thus hard to be distinguished by attention

(b) Features at faraway regions no longer have similar values after positional embedding, thus will be treated differently by attention.

Figure 4.5: Visualization of 2-D positional embedding.

as a part of our $D$. Specifically, $Enc$ is a Transformer that maps the word indices into the embeddings while adding contextual information to them. To train $Dec$ to actively generate text descriptions of an image, an attention mask is applied on the input of $Enc$, such that each word can only attend to the words preceding it in a sentence. $Dec$ is a Transformer decoder that performs image captioning by predicting the next word’s probability from the masked word embeddings and the image features.

**Image-Captioning Transformer** Symmetric to TCIT, the inverse operation, in which $f_t$ is revised by $f_i$, is leveraged for image captioning in $Dec$, as shown in Fig. 4.3-(b). Such a design has been widely used in recent captioning works. In TIME, we show that a simple 4-layer 4-head Transformer is sufficient to obtain high-quality captions and facilitate the consistency checking in the T2I task.

### 2-D Positional Encoding for Image Features

When we reshape the image features $f_i$ for the attention operation, there is no way for the Transformer to discern spatial information from the flattened features. To take advantage of coordinate signals, we propose 2-D positional encoding as a counter-part to the 1-D positional encoding in the Transformer [104]. The encoding at each position has the same dimensionality as the channel size $c$ of $f_i$, and is directly added to the reshaped image feature $f_i^T \in R^{d \times c}$. The first half of dimensions encode the y-axis positions and the second half encode the x-axis, with sinusoidal functions of different frequencies. Such 2-D encoding ensures that closer visual features have a more similar representation compared to features that are spatially more remote from each other. An example $32 \times 32$ feature-map from a trained TIME is visualized in Fig. 4.5, where we visualize three feature channels as an RGB image. In practice, we apply 2-D positional encoding on the image features for both TCIT and $Dec$ in $D$. Please refer to the appendix for
more details.

The formal equation of the 2-D positional encoding is:

\[ P_{i \in [1:4]}(y, 2i) = \sin \left( \frac{y}{10000 \cdot \frac{x}{4}} \right); \]  

\[ P_{i \in [\frac{5}{4}:2]}(x, 2i) = \cos \left( \frac{x}{10000 \cdot \frac{y}{4}} \right); \]  

\[ P_{i \in [1:4]}(y, 2i - 1) = \cos \left( \frac{y}{10000 \cdot \frac{x}{4}} \right); \]  

\[ P_{i \in [\frac{5}{4}:2]}(x, 2i - 1) = \sin \left( \frac{x}{10000 \cdot \frac{y}{4}} \right), \]  

where \( x, y \) are the coordinates of each pixel location, and \( i \) is the dimension index along the channel.

One interesting question about 2-D positional encoding is: “if we cut off the boundary parts of the input image, which leads to major value changes of the position encoding on all pixels, how will it affect the captioning result?”

The purpose of position encoding is to re-value the feature vectors based on each spatial location, making the attention operation better at attending to the related positions. We believe the “relative value” between each spatial location is the key and it should be invariant in cases such as the one just described. I.e., the neighbor pixels’ relative value difference should not be changed even when the boundary pixels are cut off. Therefore, each embedding will still attend to its most relative embeddings that have similar embedding values, so the attention result and the resulting captions will show little change. We conducted experiments that confirmed our hypothesis, and showed that the 2D-positional-encoding based attention operation has such a “content invariant” property.

4.4.2 Objectives

**Discriminator Objectives** Formally, we denote the three kinds of outputs from \( D \) as: \( D_f() \), the image feature at \( 8 \times 8 \) resolution; \( D_u() \), the unconditional image real/fake score; and \( D_c() \), the conditional image real/fake score. Therefore, the predicted next word distribution from \( Dec \) is: \( P_k = Dec(Enc(T_{1:k-1}^{real}), D_f(I_{match})) \). Finally, the objectives for \( D, Enc, \) and \( Dec \) to jointly minimize are:

\[ L_{caption} = - \sum_{k=1}^{t} \log(P_k(T_k^{real}, D_f(I_{match}))); \]  

\[ L_{uncond} = - \mathbb{E}[\log(D_u(I_{match}))] \]  

\[ - \mathbb{E}[\log(1 - D_u(I_{fake}))]; \]
A small bird that is grayish brown with a striking blue color on the head, wing and breast area.

This bird has a yellow crown, a rounded breast, and grey wings.

Figure 4.6: Samples generated during the training of TIME, note the visual features emerge in very early iterations.

along with $\mathcal{L}_{\text{cond}}$, which we shall discuss next.

**Annealing Image–Text Matching Loss** During training, we find that $G$ can learn a good semantic visual translation at very early iterations. As shown in Fig. 4.6, while the convention is to train the model for 600 epochs on the CUB dataset, we observe that the semantic features of $T_{\text{real}}$ begin to emerge on $I_{\text{fake}}$ as early as after 20 epochs. Thus, we argue that it is not ideal to penalize $I_{\text{fake}}$ by the conditional loss on $D$ in a static manner. Since $I_{\text{fake}}$ is already very consistent to the given $T_{\text{real}}$, if we let $D$ consider an already well-matched input as inconsistent, this may confuse $D$ and in turn hurt the consistency-checking performance. Therefore, we employ a hinge loss [67, 102] and dynamically anneal the penalty on $I_{\text{fake}}$ according to how confidently $D$ predicts the matched real pairs:

$$s_{\text{pivot}} = \text{detach}(\mathbb{E}[D_c(I_{\text{match}}, \text{Enc}(T_{\text{real}}))]);$$

$$\mathcal{L}_{\text{cond}} = \mathbb{E}[\min(0, 1 - D_c(I_{\text{match}}, \text{Enc}(T_{\text{real}})))]$$
$$+ \mathbb{E}[\min(0, 1 + D_c(I_{\text{mismatch}}, \text{Enc}(T_{\text{real}})))]$$
$$+ \mathbb{E}[\min(0, -s_{\text{pivot}} \times p + D_c(I_{\text{fake}}, \text{Enc}(T_{\text{real}}))]].$$

Here, detach(,) denotes that the gradient is not computed for the enclosed function, and $p = \frac{i_{\text{epoch}}}{n_{\text{epochs}}}$ (current epoch divided by total number) is the annealing factor. The hinge loss ensures that $D$ yields a lower score on $I_{\text{fake}}$ compared to $I_{\text{match}}$, while the annealing term $p$ ensures that $D$ penalizes $I_{\text{fake}}$ sufficiently in early epochs.

**Generator Objectives** On the other side, $G$ considers random noise $z$ and word embeddings from $\text{Enc}$ as inputs, and is trained to generate images that can fool $D$ into giving high scores on authenticity and semantic consistency with the text. Moreover, $G$ is also encouraged to make $D$ reconstruct the same sentences as provided as input. Thus, the objectives for $G$ to
minimize are:

\[ \mathcal{L}_{\text{caption}} = - \sum_{k=1}^{l} \log(P_k(T_{k}^{\text{real}}, D_t(G(z, Enc(T_{k}^{\text{real}}))))); \]  

(4.9)

\[ \mathcal{L}_{\text{uncond}} = -\mathbb{E}[\log(D_u(G(z, Enc(T^{\text{real}}))))]; \]  

(4.10)

\[ \mathcal{L}_{\text{cond}} = -\mathbb{E}[D_c(G(z, Enc(T^{\text{real}})), Enc(T^{\text{real}}))]. \]  

(4.11)

**Vanilla Conditional Loss** Apart from the proposed image-captioning loss, which serves as a conditioning factor to train \(D\) and \(G\), we also applied another simple conditional loss by combining the image feature and text embeddings (this vanilla conditional loss is reflected in Fig. 2 upper-left). Such conditional loss serves as a basic regulation term for \(D\), just like the conditional loss used in StackGAN.

In practice, we can extract \(8 \times 8\) or \(4 \times 4\) image features from the CNN part of \(D\), and concatenate the image features with the reshaped word embeddings along the channel dimension. Specifically, we re-shape the word embeddings by fitting each of them into one pixel location of an image feature map. For example, if we have an image feature with the shape of \(512 \times 4 \times 4\) (4×4 spatial size with 512 channels), and a text description of 16 words, where each word is represented in 512 dimensions, we reshape the word embeddings into \(512 \times 4 \times 4\), where the top-left spatial location is the first embedding, the second top-left location is the second embedding, and the bottom-right contains the last embedding, etc. The image features and the reshaped word embeddings can then be concatenated into a image-context feature with the shape of \(1024 \times 4 \times 4\). Such image-context feature is then processed by two convolution layers, giving rise to the final image–text matching score.

When the word sequence is longer than 16, we just select the first 16 words, and when the sequence is less than 16, we use 0-padding for the remaining space. This sort of procedure is fairly naive, and can potentially be improved. However, this is not the primary concern in this work and we find it already works sufficiently well across all of our experiments. Since using such a conditional loss is a common practice in prior works, we only conduct a simple ablation experiment on TIME and AttnGAN to confirm the effect of this loss. The results shows a consistent small performance gain for both TIME and AttnGAN, where abandoning such a conditional loss only leads to about 2% IS loss on the CUB dataset. As a result, we pick this conditional loss for all models. Further details of the implementation can be found in the code.
4.5 Experiments

In this section, we evaluate the proposed model from both the text-to-image and image-captioning directions, and analyze each module’s effectiveness individually. Moreover, we highlight the desirable property of TIME being a more controllable generator compared to other T2I models.

Experiments are conducted on two datasets: CUB [107] and MS-COCO [68]. We follow the same convention as in previous T2I works to split the training/testing set. We benchmark the image quality by the Inception Score (IS) [95] and Fréchet Inception Distance (FID) [29], and measure the image–text consistency by R-precision [109] and SOA-C [31].

4.5.1 Backbone Model Structure

Table 4.1 demonstrates the performance comparison between the StackGAN structure and our proposed “aggregating” structure. AttnGAN as the T2I backbone has been revised by recent advances in the GAN literature [117, 90]. Seeking a better baseline, we also incorporated recent advances. In particular, columns with “+new” imply that we train the model with equalized learning rate [42] and R-1 regularization [81]. “Aggr” means we replace the “stacked” $G$ and multiple $D$s with the proposed aggregated $G$ and a single $D$. To show the computing cost, we list the relative training times of all models with respect to StackGAN. All models are trained with the optimal hyper-parameter settings and the same loss functions from StackGAN and AttnGAN respectively.

The aggregated structure with new GAN advances achieve the best performance/compute-cost ratio in both the image quality and the image–text consistency. Moreover, we find that the abandoned lower-resolution $D$s in StackGAN have limited effect on image–text consistency, which instead appears more related to the generated image quality, as a higher IS always yields a better R-precision.

Attention Mechanisms

We conduct experiments to explore the best attention settings for the T2I task from the mechanisms discussed in Section 4.4.1.

Table 4.2 lists the settings we tested, where all the models are configured the same based on AttnGAN, except for the attention mechanisms used in $G$. In particular, column 1 shows the baseline performance that employs the basic attention operation, described in Fig. 4.4-(a), from AttnGAN. The following columns show the results of using the Transformer illustrated in Fig. 4.4-(d) with different numbers of heads and layers (e.g., Tf-h4-l2 means a Transformer
<table>
<thead>
<tr>
<th></th>
<th>Inception Score ↑</th>
<th>R-precision ↑</th>
<th>Training time ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>StackGAN w/o stack</td>
<td>3.42 ± 0.05</td>
<td>9.25 ± 3.12</td>
<td>0.57</td>
</tr>
<tr>
<td>StackGAN</td>
<td>3.82 ± 0.06</td>
<td>10.37 ± 5.88</td>
<td>1.0</td>
</tr>
<tr>
<td>Aggr GAN</td>
<td>3.78 ± 0.03</td>
<td>10.21 ± 5.42</td>
<td>0.78</td>
</tr>
<tr>
<td>Aggr GAN +new</td>
<td>4.12 ± 0.03</td>
<td>12.26 ± 4.76</td>
<td>0.85</td>
</tr>
<tr>
<td>AttnGAN w/o stack</td>
<td>4.28 ± 0.02</td>
<td>64.82 ± 4.43</td>
<td>0.71</td>
</tr>
<tr>
<td>AttnGAN</td>
<td>4.36 ± 0.03</td>
<td>67.82 ± 4.43</td>
<td>1.14</td>
</tr>
<tr>
<td>Aggr AttnGAN</td>
<td>4.34 ± 0.02</td>
<td>67.63 ± 5.43</td>
<td>0.86</td>
</tr>
<tr>
<td>Aggr AttnGAN +new</td>
<td><strong>4.52 ± 0.02</strong></td>
<td><strong>70.32 ± 4.19</strong></td>
<td>1.0</td>
</tr>
</tbody>
</table>

Table 4.1: Comparison between **stacked** and **aggregated** model structures on the CUB dataset.

<table>
<thead>
<tr>
<th></th>
<th>Inception Score ↑</th>
<th>R-precision ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>AttnGAN</td>
<td>4.36 ± 0.03</td>
<td>67.82 ± 4.43</td>
</tr>
<tr>
<td>Tf-h1-l1</td>
<td>4.38 ± 0.06</td>
<td>66.96 ± 5.21</td>
</tr>
<tr>
<td>Tf-h4-l1</td>
<td>4.42 ± 0.06</td>
<td>68.58 ± 4.39</td>
</tr>
<tr>
<td>Tf-h4-l2</td>
<td><strong>4.48 ± 0.03</strong></td>
<td><strong>69.72 ± 4.23</strong></td>
</tr>
<tr>
<td>Tf-h4-l4</td>
<td>4.33 ± 0.02</td>
<td>67.42 ± 4.31</td>
</tr>
<tr>
<td>Tf-h8-l4</td>
<td>4.28 ± 0.03</td>
<td>62.32 ± 4.25</td>
</tr>
</tbody>
</table>

Table 4.2: Comparison of different attention settings on CUB.
This is a small bird with white belly and a short beak.
This is a small bird with white belly and a long beak.
A yellow bird with a yellow belly and brown wings.
A yellow bird with a yellow belly and black wings.
The bird has a beautiful red body and grey wings.
The bird has a beautiful red body and grey head.
This bird has wings that are black and has a red belly.
This bird has wings that are black and has a yellow belly.
This is a green bird with a brown crown and a white and black belly.
This small bird has black eyerings and cheek patches, with a white breast.
A brown bird with white and brown feathered belly and breast, and small pointed yellow bill.

Figure 4.7: Images from TIME with fixed $z$ and varied sentences

with 4 heads and 2 layers). The results suggest that a Transformer with a more comprehensive attention yields a better performance than the baseline. However, when increasing the number of layers and heads beyond a threshold, a clear performance degradation emerges on the CUB dataset. More discussion and the results on MS COCO can be found in the appendix.

4.5.2 Controllable G without Sentence-Embedding

Most previous T2I models rely on a sentence-level embedding $f_s$ as a vital conditioning factor for $G$ [113, 109, 90, 117, 62]. Specifically, $f_s$ is concatenated with noise $z$ as the input for $G$, and is leveraged to compute the conditional authenticity of the images in $D$. Sentence embeddings are preferred over word embeddings, as the latter lack contextual meaning and semantic concepts are often expressed in multiple words.

However, since $f_s$ is a part of the input alongside $z$, any slight changes in $f_s$ can lead to major visual changes in the resulting images, even when $z$ is fixed. This is undesirable when we like the shape of a generated image but want to slightly revise it by altering the text description. Examples are given in Fig. 4.7-(a), where changing just a single word leads to unpredictably large changes in the image. In contrast, since we adopt the Transformer as the text encoder, where the word embeddings already come with contextual information, $f_s$ is no longer needed in TIME. Via our Transformer text encoder, the same word in different sentences or at different positions will have different embeddings. As a result, the word embeddings are sufficient to provide semantic information, and we can abandon the sentence embedding.

In Fig. 4.7-(b) and (c), TIME shows a more controllable generation when changing the captions while fixing $z$. TIME provides a new perspective that naturally enables fine-grained manipulation of synthetic images via their text descriptions.

**Discussion on Controllability** The reason we gain controllability with TIME is fairly
intuitive. Here, we extend the discussion in the chapter on "Controllable G without Sentence-Level Embedding" with more implementation details. Unlike previous T2I models, the text information is introduced to the model only on high resolution feature-maps (e.g., 32x32 and 64x64 as illustrated in Fig. 2 of the chapter), thus it is unlikely to change the content shape. This intuition is validated in many GAN works, a typical example being StyleGAN, where changing the input noise at the 8×8 feature level leads to major changes in content, while changing the input noise at the 32×32 level does not change the shapes in the image at all, but instead only leads to stylistic changes such as the coloring. Our experiments also confirmed this observation: when we tested feeding text embeddings to a lower-resolution layer (e.g., 8×8), the model controllability indeed disappeared.

Note that no additional tweaks were used (extra loss, extra labeled data) to gain such controllability, and thus the degree of controllability is highly biased on the data. Taking the CUB dataset as an example, although the paired text description only describes how the bird looks like, there is a high correlation between the color of birds and the background, e.g., yellow birds tend to be found more in green trees and white birds more in the blue sky. Such "background & bird-style" correlation can easily be learned by the model without any extra guidance. As a result, the background color also changes when we change text that merely describes the bird, and hence the degree of controllability pertains more to the invariant shape of the bird but not to an invariant background.

4.5.3 Ablation Study

Based on AttnGAN objectives, we combine the model settings from Table 4.1 row.7 and Table 4.4 row.5 as the baseline, and perform ablation study in Table 4.3. First, we remove the image captioning text decoder $Dec$ to show its positive impact. Then, we add $Dec$ back and show that dropping the sentence-level embedding does not hurt the performance. Adding 2-D positional encoding brings improvements in both image–text consistency and the overall image quality. Lastly, the proposed hinge loss $L_{hinge}$ (eq. 4) releases $D$ from a potentially conflicting signal, resulting in the most substantial boost in image quality.

To emphasize the contribution of the proposed image-text hinge loss $L_{hinge}$, we conduct ablation study of it with different annealing schedules, including: stop training on $L_{hinge}$ after 400 epoches (early-stop), start training on $L_{hinge}$ after 100 epoches (late-begin), and annealing $L_{hinge}$ with a constant factor 1. Fig. 4.8 records the model performance along the training
<table>
<thead>
<tr>
<th></th>
<th>Inception Score ↑</th>
<th>R-precision ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>4.64 ± 0.03</td>
<td>70.72 ± 1.43</td>
</tr>
<tr>
<td>B - img captioning</td>
<td>4.58 ± 0.02</td>
<td>69.72 ± 1.43</td>
</tr>
<tr>
<td>B - Sentence emb</td>
<td>4.64 ± 0.06</td>
<td>68.96 ± 2.21</td>
</tr>
<tr>
<td>B + 2D-Pos Encode</td>
<td>4.72 ± 0.06</td>
<td>71.58 ± 2.39</td>
</tr>
<tr>
<td>B + Hinged loss</td>
<td>4.91 ± 0.03</td>
<td>71.57 ± 1.23</td>
</tr>
</tbody>
</table>

Table 4.3: Ablation Study of TIME on CUB dataset

![Graph showing Inception Score over Epoches for different annealing schedules](image)

Figure 4.8: Performance comparison on different annealing schedules of the hinged image-text consistency loss.
Table 4.4: Comparison between different attention settings on MS COCO dataset

<table>
<thead>
<tr>
<th></th>
<th>AttnGAN</th>
<th>Tf-h1-l1</th>
<th>Tf-h4-l1</th>
<th>Tf-h4-l2</th>
<th>Tf-h4-l4</th>
<th>Tf-h8-l4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Inception Score ↑</strong></td>
<td>25.89</td>
<td>26.58</td>
<td>26.42</td>
<td>27.48</td>
<td>27.85</td>
<td>27.153</td>
</tr>
<tr>
<td><strong>R-precision ↑</strong></td>
<td>83.53</td>
<td>86.46</td>
<td>88.58</td>
<td>89.72</td>
<td>89.57</td>
<td>88.32</td>
</tr>
</tbody>
</table>

iterations. Firstly, it shows the effectiveness of the proposed $L_{hinge}$ within all the anneal schedules. Moreover, early-stop leads to a direct performance downgrade in late iterations, while late-begin performs the worst in early iterations. Annealing with a constant factor yields a similar performance as the dynamic annealing in early iterations, but falls back later when the models converge.

**Transformer configuration exploration** We conducted the same experiments to explore the best attention settings for the T2I task on the MS COCO dataset. Table 4.4 lists the settings we tested, where all the models are configured in the same way based on AttnGAN, except for the attention mechanisms used in $G$.

The results are consistent with the experiment on the CUB dataset in showing that the “multi-head and multi-layer” (MHL) design boosts the performance. We hypothesize that the optimal number of heads and layers depends on the dataset, where the “4-heads 2-layers” and “4-heads-4-layers” settings are the sweet points for the CUB and COCO datasets, respectively. While the motivation and design of MHL is quite intuitive, it is still possible that the performance gain results from the increased number of parameters. This suspicion is beyond this work. On the other hand, as shown in the last two columns, making the Transformer too “big” leads to worse performance. A possible reason is that the increase in parameters makes it harder for the model to converge, and makes it more susceptible to overfitting the training data.

**Language Model Performance** Apart from a strong T2I performance, TIME also yields $D$ as a well-performed stand-alone image captioning model.

Table 4.5 shows the comparison between TIME and more complicated NLP models, reflects the practicality and competence of TIME on the more general Vision-Language (VL) tasks. Note that we compete with Bert [17, 88], UNITER [14] and OSCAR [66] which all are large scale models with 24 Transformer (TF) blocks, and pre-trained on multiple VL tasks for an optimal performance.

In contrast, TIME is only trained on the studied text-image mutual translation task, with a tiny model size (only 8 TF blocks) and without any pre-training. It gains close performance to
Table 4.5: Results on downstream Vision-Language tasks from TIME on COCO, compared with SOTA models.

<table>
<thead>
<tr>
<th>model</th>
<th>Captioning BLEU-4</th>
<th>Image Retrieval@5↑</th>
<th>Text Retrieval@5↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bert Basic (with 24 TF)</td>
<td>0.389</td>
<td>69.3</td>
<td>82.2</td>
</tr>
<tr>
<td>UNITER (with 24 TF)</td>
<td>0.395</td>
<td>76.7</td>
<td>87.0</td>
</tr>
<tr>
<td>OSCAR (with 24 TF)</td>
<td>0.405</td>
<td>80.8</td>
<td>91.1</td>
</tr>
<tr>
<td>TIME (with 8 TF)</td>
<td>0.381</td>
<td>75.1</td>
<td>78.2</td>
</tr>
</tbody>
</table>

Figure 4.9: Learned word embeddings on CUB, and qualitative results on MS-COCO

Table 4.6: Text-to-Image performance comparison between TIME and other models.

<table>
<thead>
<tr>
<th></th>
<th>StackGAN</th>
<th>AttnGAN</th>
<th>ControlGAN</th>
<th>MirrorGAN</th>
<th>DMGAN</th>
<th>TIME</th>
<th>Real-Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUB Inception Score ↑</td>
<td>3.82 ± 0.06</td>
<td>4.36 ± 0.03</td>
<td>4.51 ± 0.06</td>
<td>4.56 ± 0.05</td>
<td>4.71 ± 0.02</td>
<td><strong>4.91 ± 0.03</strong></td>
<td>5.04</td>
</tr>
<tr>
<td>FID ↓</td>
<td>N/A</td>
<td>23.98</td>
<td>N/A</td>
<td>N/A</td>
<td>16.09</td>
<td><strong>14.3</strong></td>
<td>0</td>
</tr>
<tr>
<td>R-precision ↑</td>
<td>10.37 ± 5.88</td>
<td>67.82 ± 4.43</td>
<td>69.33 ± 3.21</td>
<td>69.58 ± 4.39</td>
<td>72.31 ± 0.91</td>
<td>71.57 ± 1.2</td>
<td>N/A</td>
</tr>
<tr>
<td>COCO Inception Score ↑</td>
<td>8.45 ± 0.03</td>
<td>25.89 ± 0.47</td>
<td>24.06 ± 0.6</td>
<td>26.47 ± 0.4</td>
<td>30.49 ± 0.5</td>
<td><strong>30.85 ± 0.7</strong></td>
<td>36.5</td>
</tr>
<tr>
<td>FID ↓</td>
<td>N/A</td>
<td>35.49</td>
<td>N/A</td>
<td>N/A</td>
<td>32.64</td>
<td><strong>31.14</strong></td>
<td>0</td>
</tr>
<tr>
<td>R-precision ↑</td>
<td>N/A</td>
<td>83.53 ± 0.43</td>
<td>82.43 ± 2.21</td>
<td>84.21 ± 0.39</td>
<td>91.87 ± 0.28</td>
<td>89.57 ± 0.9</td>
<td>N/A</td>
</tr>
<tr>
<td>SOA-C ↑</td>
<td>N/A</td>
<td>25.88</td>
<td>25.64</td>
<td>27.52</td>
<td>33.44</td>
<td>32.78</td>
<td>74.97</td>
</tr>
</tbody>
</table>
the SOTA models, which reveals a promising area for future research towards mutual-translation in a single framework. Fig. 4.9 shows the qualitative results from TIME on language tasks. In Fig. 4.9-(a), words with similar meanings reside close to each other. “Large” ends up close to “red”, as the latter often applies to large birds, while “small” is close to “brown” and “grey”, which often apply to small birds.

4.5.4 Comparison on T2I with State-of-the-Arts

We next compare TIME with several SOTA text-to-image models. Qualitative results of TIME can be found in Figs. 4.1, 4.7, and 4.9. On CUB, TIME yields a more consistent image synthesis quality, while AttnGAN is more likely to generate failure samples. On MS-COCO, where the images are much more diverse and complex, TIME is still able to generate the essential contents that is consistent with the given text. The overall performance of TIME proves its effectiveness, given that it also provides image captioning besides T2I, and does not rely on any pre-trained modules.

As shown in Table 4.6, TIME demonstrates competitive performance on MS-COCO and CUB datasets with the new state-of-the-art IS and FID. Unlike the other models that require a well pre-trained language module and an Inception-v3 image encoder, TIME itself is sufficient to learn the cross-modal relationships between image and language. Regarding the image–text consistency performance, TIME is also among the top performers on both datasets. Specifically, we do not tune the model structure to get an optimal performance on MS-COCO dataset. As our text decoder in $D$ performs image captioning with image feature-map of size $8 \times 8$, such a size choice may not able to capture small objects in images from MS-COCO. On the other hand, $8 \times 8$ is the suitable size to capture features of bird parts, for images from CUB dataset.

Importantly, TIME is considerably different from AttnGAN (no pre-training, no extra CNN/RNN modules, no stacked structure, no sentence embedding), while the other models all based on AttnGAN with orthogonal contributions to TIME. These technique contributions (e.g. DMGAN, SD-GAN, OP-GAN) could also be incorporated into TIME, with foreseeable performance boost. Due to the content limitation, we omit the integration of such ideas.

4.5.5 More Qualitative Results
Figure 4.10: Image captioning results of TIME on the CUB dataset testing set.

Figure 4.11: Image captioning results of TIME on the MS-COCO dataset testing set.
A small bird with a small beak compared to its head size, that is covered in red, brown and white.

This bird has a red crown with black wings and a black beak.

A brown bird with white and brown feathered belly and breast and small pointed yellow bill.

This bird has a dark grey color, with a large bill and long wingspan.

The bird has a yellow crown, a rounded breast, and grey wings.

This small bird features a yellow plumage with streaks of dark brown and a pink, stout beak.

A small bird with wide white eyes sitting on rock edges.

This is a green bird with a brown crown and a white and black belly.

This bird has a red crown with grey and has a long, pointy beak.

This small bird has black eyerings and cheek patches, with a white breast.

This small bird has a yellow, grey, and red and has a very short beak.

A booth for or by firefighters at a fair.

Three giraffes walking around their pen in a zoo.

A person on a snowboard jumping over a railing into snow.

The television is mounted on the wall above the fireplace.

Figure 4.12: Uncurated Qualitative results of TIME on the CUB dataset: The images are generated by TIME given the captions from testing dataset.

An empty living room with a fire place and a mirror.

A group of young people sitting around each other.

A powered on computer monitor sitting on an office desk.

A large ornate clock hangs on the side of a building.

A booth for or by firefighters at a fair.

A couple of large pizzas that are on a table.

A sleek looking red train parked at a train station.

The children are playing the game of baseball.

A group of sheep grazing in a grassy field.

A bed with patterned red sheets and orange bedspread.

A rusted area in the floor where a toilet used to sit.

A red stop sign has a street sign on top of it.

The children are playing the game of baseball.

The television is mounted on the wall above the fireplace.

Figure 4.13: Uncurated Qualitative results of TIME on the MS-COCO dataset: The images are generated by TIME given the captions from testing dataset.
The medium sized bird has a dark grey color, a black downward curved beak, and long wings.

This brown bird has a little bit of red on its belly and a gray head.

Small bird, with a white chest, a blue crown, and black and blue wings.

A small colorful bird, with a long orange beak, blue wing and tail feathers, and a black and orange under belly and head.

This is a small grey bird with a straight, flat beak.

Figure 4.14: Uncurated Qualitative results of TIME on the CUB dataset: The images are generated by TIME given the captions from testing dataset, each row containing the generates samples from the same caption.
Figure 4.15: Uncurated Qualitative results of TIME on the MS-COCO dataset: The images are generated by TIME given the captions from testing dataset.
4.6 Conclusion

In this chapter, we propose the Text and Image Mutual-translation adversarial nEtwork (TIME), a unified framework trained with an adversarial schema that accomplishes both the text-to-image and image-captioning tasks. Via TIME, we provide affirmative answers to the four questions we raised in Section 1.2. While previous works in the T2I field require pre-training several supportive modules, TIME achieves the new state-of-the-art T2I performance without pre-training. The joint process of learning both a text-to-image and an image-captioning model fully harnesses the power of GANs (since in related works, $D$ is typically abandoned after training $G$), yielding a promising Vision-Language performance using $D$. TIME bridges the gap between the visual and language domains, unveiling the immense potential of mutual translations between the two modalities within a single model.
Chapter 5

Posture and Identity Disentangled Image-to-Image Translation

5.1 Introduction

We study a sub-task of image-to-image translation, which we call posture-and-identity disentangled translation. We view an image as a combination of multiple objects with varied scales, then define posture as the shape and composition of the larger-scale objects, and identity as both the color/texture of the objects and the shape of the finer-scale objects. Given two input images, one defines posture, and the other defines identity. Our goal is to develop a model that transfers the first image’s stance to the second one’s identity. This task has excellent potential in real-life applications such as illustration synthesis and motion driving for content-creation jobs. Note that a close task has been widely studied as content-style disentangled image synthesis, e.g. [16, 56, 60, 73, 70, 54, 46]. We find the terminology of “content-style” can be ambiguous sometimes, as both posture and identity has content and style related properties. In contrast, “posture-identity” are relatively precise to the task and datasets that we are working on, including human, animal, and animation faces.

Given an image generative network, many works bid on the model’s structural designs to help separate the spatial information of posture from the finer identity information. For example, [35] proposed AdaIN which encodes the color features into a vector without spatial dimensions, thus minimizes the vector’s influence on spatial shapes (when doing affine transformation on feature-maps). [45] proposed StyleGAN that exploits the structural inductive bias of convolutional network, in which the relative size of the conv kernel and the feature-map determines the feature-map’s changing scope. By injecting latent vectors to feature-maps at different resolutions, the lower-resolution conv-layers naturally tend to control the larger-scale shapes of the generated images, while the higher layers work only on coloring and texturing.

Our key idea is to let the image generator automatically learn a semantic mask of the posture image. Different regions in the mask represent a semantic object in the image. The synthesis is then based on the objects’ shape and composition in the mask. To facilitate the idea, we devise
Figure 5.1: Unsupervised image-to-image translation results of PIVQGAN with disentangled posture and identity control. In each panel, the first row has input pose images, and the first column has referential identity images. The second and third rows are “segmentation-like” masks automatically learned by PIVQGAN, and bottom-right are the synthesized images.

We show that, with a carefully designed self-supervision training scheme, VQSN can automatically learn meaningful object embeddings that provide useful posture information. The resulting model, which we call PIVQGAN, enables a precise control on posture and identity of the image generation, and outperforms existing baselines in terms of visual quality and translation accuracy. Besides, we show a robust performance of PIVQGAN even on unseen out-of-domain images, where we further demonstrate its promising application beyond pose-identity translation, including zero-shot image style-transfer and semantic-aware image-editing. Qualitative results can be found in Fig 5.1

a Vector-Quantized Spatial Normalization layer (VQSN) that combines vector-quantization [87] and spatially-adaptive normalization [89]. Specifically, we first leverage the VQ mechanism to reduce a feature-map into a certain number of object embeddings. Each embedding corresponds to one common object shared among images of the same domain (e.g. eye, mouth, and nose for face dataset; wall, window, and roof for building dataset). We then get a spatial object mask that we refer to when performing an affine transformation on the feature-map. Each spatial location is transformed differently according to its corresponding object.
5.2 Methods

In this section, we first describe PIVQGAN’s synthesis flow. Then we introduce Vector-Quantized Spatial Normalization (VQSN). Finally, we talk about the self-supervised training scheme and all the objective functions.

5.2.1 Model overview

PIVQGAN consists of three models, the discriminator $D$ and the generator $G$ (modified with VQSN layers) from StyleGAN2, and an encoder $E$ that converts images to vectors in $G$’s latent space.

In each training iteration, $D$ and $G$ are first trained as GAN [27], then $G$ and $E$ are trained for GAN inversion tasks (as in [108]) like an Autoencoder. A similar joint-training schema without dedicated self-supervision is also explored by [48] in StyleMapGAN. Such a joint-training is required in our case to implement the proposed self-supervisions as going to be described in Sec 5.2.3.

We separate the generation of posture and identity attributes on different convolutional layers of $G$. Specifically, given that $G$ synthesizes an image at the resolution of $256 \times 256$, a base feature-map of size $4 \times 4$ is up-scaled six times by 14 conv-layers, each conv-layer modulated by a corresponding latent vector. We take 2 to 3 layers between 2nd to 7th layers (varied on datasets) as the pose-layer. The pose-layers are forced to learn only posture-related information, which has to be independent of the other layers. Meanwhile, given an image, $E$ is trained to generate the 14 latent vectors that can let $G$ reconstruct the image. The goal of this setting is to let the latent vectors for pose-layers encode only the pose information, while the rest carry other attributes.

In inference time, $E$ gets two latent sets $l_1^{[1:14]}, l_2^{[1:14]}$ from two images $i_1, i_2$. And we graft the latent sets by swapping the pose-layers’ vectors to transfer the posture or identity features between the two images. For example, suppose layer 4 to 6 are pose-layers, the grafted latent sets $[l_1^{[1:3]}, l_2^{[4:6]}, l_1^{[7:14]}]$ will let $G$ generate a image with the identity features of $i_1$, while having the posture of $i_2$. Moreover, we can also randomly sample $l$ from the latent space of $G$, for more diverse translations.
**Figure 5.2:** Top: Illustration of the VQSN module. (1) A base feature-map is quantized by a certain number (e.g. 2 or 6) of trainable embeddings. (2) The VQ embeddings are used to perform the spatial-wise affine transformation on image feature-maps.

Bottom: Training scheme of PIVQGAN. Apart from the regular GAN training (omitted), we train an encoder together with the convolution part of the generator with self-supervision tasks. In (3), we do latents-grafted reconstruction to help the model disentangle pose/identity attributes at different layers.
5.2.2 Vector-Quantized Spatial Normalization

With the hope of letting the selected pose-layers learn only pose-related features while not influencing the rest layers to maintain a high synthesis quality, we propose the VQSN layer. VQSN is a plug-in module that works on the output feature-map of common convolutional layers.

The VQSN module takes two inputs: a vector $l^p \in \mathbb{R}^{512}$ (from StyleGAN’s latent space) that defines the posture attributes, and an image feature-map $f \in \mathbb{R}^{c \times h \times w}$, e.g. $c, h, w = (512, 16, 16)$. And it outputs a revised feature-map $f'$ in the same shape. There are two parts inside the VQSN module. First, a pose generator has a trainable basis in the shape of $512 \times 4 \times 4$ and three modulated convolutional layers (like a tiny StyleGAN2 generator). This pose generator uses $l^p$ to up-sample the basis into a base feature-map $f_b \in \mathbb{R}^{c \times h \times w}$. Second, as shown in Fig. 5.2-left, $f_b$ is quantized into $f_{vq}$ according to a certain number of trainable embedding vectors. Finally, $f_{vq}$ is converted into alpha and beta via $1 \times 1$ convolutional layers, and performs affine transformation on $f$ to get the output $f'$.

Two intuitions guide our design of VQSN. First, we hope to compress the representation capability of the pose-layers so that it has no room to carry unwanted shape-invariant information. Vector-quantization fits well in this case, for it reduces the continuous representation space into a few discrete embeddings. The VQ module categorizes each spatial location into one object. And the categorization process naturally defines the shape of each object and the composition of all the objects. Second, to highlight shape-variant information, we want to treat each spatial location differently in a feature-map (e.g. the eye area has a value distribution that is different from the mouth area). We apply unique affine-transformation at different pixel locations according to the quantized object mask to address such a difference.

5.2.3 Self-supervised training scheme

The encoder $E$ aims at projecting an image into the latent space of $G$, where $G$ can faithfully reconstruct the input image. There are two main training stages when we jointly train $E$ and $G$. First, we sample latent vectors $l_{gt} \in \mathbb{R}^{h \times 14 \times 512}$ from the mapping network [44] in $G$ with random noise as input. Here we train the decoder network in $G$ and $E$ to reconstruct the sampled latent vectors:

$$\mathcal{L}_{\text{latent}} = E[\|E(G(l_{gt})) - l_{gt}\|_2]$$ (5.1)
\( \mathcal{L}_{\text{latent}} \) ensures \( E \) and \( G \) to be well communicated, thus helps \( E \) to always produce in-domain latent vectors in \( G \)'s latent space. Second, we sample real images \( i_{\text{real}} \) and train \( E \) and \( G \) to reconstruct them with LPIPS loss [114]:

\[
\mathcal{L}_{\text{rec}} = \mathbb{E}[||i_{\text{real}} - G(E(i_{\text{real}}))||_2] + \text{LPIPS}(i_{\text{real}}, G(E(i_{\text{real}})))
\] (5.2)

Importantly, while the VQSN module captures large-scale posture features well (e.g. head direction), it can hardly capture smaller-scale pose information, such as mouth and eyes openness, by itself. To help VQSN capture those detailed postures, we apply a series of augmentations to the real images and train \( E \) and \( G \) on the augmented image reconstruction task. As shown in Fig. 5.4, we use shape-oriented augmentations, including random cutout, flipping, and re-scaling, to create pose and identity image pairs \( [i_{\text{org}}, i_{\text{aug}}] \). Then, as shown in Fig. 5.2-bottom-(3), we train \( E \) and \( G \) on reconstruction tasks from the grafted latents (again suppose layer 4 to 6 are pose layers):

\[
\begin{align*}
 l^{\text{org}}_{[1:14]} &= E(i_{\text{org}}); \quad l^{\text{aug}}_{[1:14]} = E(i_{\text{aug}}) \\
 \mathcal{L}_{\text{aug}} &= \text{LPIPS}(G([l^{\text{org}}_{[1:3]}, l^{\text{aug}}_{[4:6]}, l^{\text{org}}_{[7:14]}]), i_{\text{aug}}) + \text{LPIPS}(G([l^{\text{aug}}_{[1:3]}, l^{\text{org}}_{[4:6]}, l^{\text{aug}}_{[7:14]}]), i_{\text{org}})
\end{align*}
\] (5.3)

\( \mathcal{L}_{\text{aug}} \) makes sure that only the selected pose-layers encode these augmentation-related shape information while the rest do not carry them. With this training scheme, VQSN automatically learns the proper semantics of the commonly shared objects between training images.

### 5.3 Experiments

In this section, we describe the evaluation metrics and experiment results. We first compare PIVQGAN to the state-of-the-art methods, then present additional analysis and demonstrate applications of it for further insights. In each experiment, we train all the models at 256 × 256 resolution. Our model is trained five times in each setting, and the median score is reported.

**Baselines.** We compare PIVQGAN with both latest and some earlier baseline models that perform examplar-based image-to-image translation. Including DATGAN by [56], CLUIT by [60], SNI by [3], StarGAN-v2 by [16], MUNIT by [36], and TUNIT by [5]. For models that are not open-source, we use the scores reported from the authors.

**Datasets.** We show qualitative results of PIVQGAN on five datasets covering a wide range of image domains, including CelebA [76], AFHQ [16], LSUN car and church [111], and Anime-Girls [63]. And we conduct quantitative comparisons with other models on CelebA and AFHQ datasets. We do not use any labeling data; only images are needed to train our model.
Metrics. For quantitative comparisons, we use Frechét inception distance (FID, [29]) and learned perceptual image patch similarity (LPIPS, [114]) to evaluate the synthesized image quality and diversity. Perceptual path length (PPL, [44]) is used to measure the smoothness and linearity of the latent space of $G$. We also include a simple new metric called mix-back LPIPS. Given two images $i_a, i_b$, we exchange their pose by the models to get $i_{ab}, i_{ba}$ ($i_{ab}$ has the identity of $i_a$ and pose of $i_b$), then we exchange their pose again to get $i_{aba}, i_{bab}$. If pose and identity are perfectly disentangled, and the image synthesis quality is high, $i_{aba}$ should look exactly the same as $i_a$, so does $i_{bab}$ and $i_b$. For all these metrics, a lower score means better performance.

5.3.1 Comparison to baselines

<table>
<thead>
<tr>
<th></th>
<th>AFHQ</th>
<th>CelebA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FID m-LPIPS r-LPIPS PPL</td>
<td>FID m-LPIPS r-LPIPS PPL</td>
</tr>
<tr>
<td>MUNIT</td>
<td>68.32 0.842 0.512 -</td>
<td>85.74 0.856 0.537 -</td>
</tr>
<tr>
<td>TUNIT</td>
<td>48.92 0.721 0.491 -</td>
<td>36.88 0.766 0.451 -</td>
</tr>
<tr>
<td>SNI</td>
<td>16.02 0.481 0.385 65.22</td>
<td>17.45 0.461 0.382 57.63</td>
</tr>
<tr>
<td>StarGAN-v2</td>
<td>19.93 0.671 0.436 97.83</td>
<td>22.35 0.671 0.405 72.26</td>
</tr>
<tr>
<td>DATGAN</td>
<td>16.09 0.472 0.378 63.44</td>
<td>18.11 0.472 0.407 48.12</td>
</tr>
<tr>
<td>ours</td>
<td><strong>12.54</strong> 0.463 0.373 <strong>58.63</strong></td>
<td><strong>12.65</strong> 0.447 0.331 51.32</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>AFHQ</th>
<th>CelebA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FID m-LPIPS r-LPIPS PPL</td>
<td>FID m-LPIPS r-LPIPS PPL</td>
</tr>
<tr>
<td>VQ-2</td>
<td>16.18 0.493 0.401</td>
<td>12.65 0.447 0.331</td>
</tr>
<tr>
<td>VQ-3</td>
<td><strong>10.98</strong> 0.512 <strong>0.358</strong></td>
<td><strong>10.82</strong> 0.451 0.341</td>
</tr>
<tr>
<td>w/o augment</td>
<td>12.32 0.532 0.366</td>
<td>12.72 0.522 0.346</td>
</tr>
<tr>
<td>w/o VQ</td>
<td>12.93 0.518 0.362</td>
<td>13.53 0.538 0.352</td>
</tr>
<tr>
<td>VQ-1</td>
<td>12.54 0.463 0.373</td>
<td>12.46 0.449 0.339</td>
</tr>
</tbody>
</table>

Table 5.1: Comparison to baselines.

StarGAN-v2 and CLUIT have the best visual quality among the baseline models, so we only show quality comparison with these two models on the AFHQ dataset in Fig. 5.3. We consider AFHQ data harder than CelebA because it has more prominent identity feature variance with different animal breeds. PIVQGAN outperforms its competitors in both translation accuracy and synthesis quality. Our model produces a sharper image with more precise details and correct coloring in overall image quality. In terms of translation accuracy, PIVQGAN overcomes some common defects shared by prior methods. Specifically, models before CLUIT can hardly maintain a good identity consistency. StarGAN-v2, TUNIT, and MUNIT even generate animals with the wrong species. CLUIT fails to translate the detailed posture attributes, including eye
Figure 5.3: Qualitative comparison of PIVQGAN and other baselines on AFHQ dataset. Advantages of our model are highlighted.

and mouth open degree and ear fold degree. In contrast, PIVQGAN not only translates these postures well but also captures the fine-grained identity features better, such as the texture of leopard, the ear color of fox, and the eye color of the cats, where other models fail to capture.

To compute FID, we use images from the testing set as pose and identity reference images, randomly matching them to get 10000 input pairs and generate 10000 testing samples. These samples are then used to compute FID with all the training set images. The same input pairs are also used to calculate the mix-back LPIPS. We also use LPIPS to measure the models' reconstruction performance, where we use the same testing image as both pose and identity reference, and compute the mean LPIPS of 10000 samples. To calculate PPL, we randomly sample two images from the testing set as a pair and get latent vectors from them, then interpolate the two latent sets to compute the score. We also use 10000 sample pairs and report their mean value.

Table. 1 shows the metrics comparison results. For both datasets, our model stands out with almost 25% improvement on FID over the previous state-of-the-arts, indicating a better image quality and a more diverse coverage on identity and posture features. The better mix-back LPIPS shows that our model separates and preserves the pose and identity features well.
Figure 5.4: **Left:** Object-mask editing. **Right:** Various augmentation methods used on the model’s identity inputs, enabling the model to learn the augment-variant attributes from only posture inputs.

### 5.3.2 Analysis of model components

**VQSN vs Gumbel-softmax.** To learn a segmentation-like representation that captures the object shape and composition feature, Gumbel-softmax (GS, [38]) is also a viable approach besides our employed Vector-Quantization. We also implemented the GS version of our spatial transformation module. We use the differentiable soft-max trick in GS to classify each pixel location into one of the object labels. Unfortunately, with GS, the model is unable to learn interpretable semantic masks as VQSN does. Moreover, the object mask from GS looks like random noise and leads to a worse image synthesis quality of the model. Nonetheless, we think GS is worth more study; a better parameter tuning and structure design may lead to viable performance.

**VQSN structure design.** We explore multiple structural settings of our VQSN module to have a better understanding of its performance. Among various trials, we present three notable settings:

1) VQ-1: the basic design, where the object-mask is got from a base feature-map generated from the input latent vector and modulated conv-layers.

2) VQ-2: the base feature-map is generated from the previous object mask’s VQ embeddings with modulated conv-layer and the current latent vector.

3) VQ-3: the VQ-embeddings will first concatenate to the image feature-map, then generate
the affine transformation parameters.

The intuition for VQ-2 is to get hierarchically coarse-to-fine object masks over multiple pose-layers. So the object-mask is generated based on the previous layer’s output mask. For example, the first VQ layer learns the large-scale object information such as head region and background region; then, the second VQ layer can learn eye, mouth, and nose shapes inside the head region. For VQ-3, we aim to smooth the affine transformation on object contour areas because the object mask has a sharp difference at object boundary. Therefore, we concatenate the spatially discrete VQ embeddings with the continuous image feature-map before using the VQ embeddings to revise the image feature-map.

Interestingly, VQ-2 does not do what we expected. Instead, it leads to worse visual semantics on object masks and worse performance across the metric. We hypothesize this depends on the image visual properties of the datasets. For example, as shown in Fig. 5.4-right and Fig. 5.5-left, on AFHQ, the VQ layers first learn head shape then smaller objects. In reverse order, on CelebA, the model first learns eye, mouth, and nose objects, then captures head and hair in the next layer. We do not extend more on it in this chapter but will further study the reason behind it.

Meanwhile, VQ-3 improves the image quality as expected. However, the pose translation performance slightly downgrades. The concatenated image feature-map may introduce noise to the VQ module and weaken its ability to learn object shaping features. The quantitative results on AFHQ are presented in Fig. 5.3.1. While VQ-3 has better FID and reconstruction LPIPS, it performs marginally worse than VQ-1 in disentangling and preserving the pose and identity features.

**Shape-related image augmentation.** Self-supervised reconstruction task with image augmentation (Fig. 5.2-right) is the major reason that PIVQGAN captures the small-scale object semantics at VQSN layers. PIVQGAN can hardly capture the mouth and eye pose without this training scheme, which failed by other baseline models. Both the VQ module and the augmented training bring better translation performance, as can be found according to the mix-back LPIPS score in Table. 2. The augmented training makes the model harder to learn the original image distribution because the training images now contain a larger variance on object shapes. This is the reason for a slightly worse FID and r-LPIPS score when trained with augmentations.
5.3.3 Qualitative evaluation and applications

Latent space smoothness. In the VQSN module, we still let a vector control the synthesis of the object-mask. The design is to get a smooth latent space of the object-mask. Thus we can generate continuous transactions between different postures. Fig. 5.5-right shows the interpolation between two posture vector sets. In each panel, the left-most image defines the identity and the initial posture, and the right-most image defines the target posture. Both inputs are real images from the testing set. We can see that both the generated images and the generated object-masks have a smooth transaction. On the third panel, we show one problem of PIVQGAN that remains to be solved. On CelebA, our model learns the hairstyle as a posture attribute rather than an identity attribute because the hairstyle is more of a shape-related property than a shape-invariant one like color and texture. Such a problem is hard to address by neither the VQSN module nor the augmentation training scheme.

Fig. 5.5-right further shows the smoothness of the identity latent space in PIVQGAN and also reflects the excellent disentanglement between posture and identity features of our model. The left-most image defines the pose and the initial identity in each row, while the right-most one defines the target identity. Both inputs are images from the testing set. All the synthesized images remain of high quality while maintains the original posture well. On CelebA, PIVQGAN
Figure 5.6: PIVQGAN is robust to out-domain images with unseen textures, and can generate meaningful in-domain counterparts from these out-domain inputs. (a,b) Out-domain images as identity input. (c,d) Out-domain images as posture input.

generates expressive human faces among the interpolations. On AFHQ, the interpolated images present the breed-blending visuals between the species. It shows the potential of our model for creative material synthesis.

Object-mask editing. After training, the encoder $E$ combined with the VQSN layer in $G$ becomes a segmentation network. Given an image, the combined model generates object masks that highlight each component of the image. Of course, the produced masks are far less accurate than actual segmentation maps. However, these object-masks still get the essential semantics of the input image. And by editing the masks, we can create novel images from $G$ that follow our revisions.

Fig. 5.4-left presents some editing examples with testing images as referential inputs. In Fig. 5.4-left-(a), we find that the red region represents the eyes, so if we enlarge the red parts, the eyes are also getting bigger in the generated images. Similarly, the green parts correlate well with the nose in the images, so if we change the size of the green part, we achieve some funny pictures of tigers and cats having big noses. One last example is that the yellow part highlights the mouth area, so if we gradually remove the yellow part, we can see that the generated animals have their mouths closed.
Robust on out-domain images. We consider the robustness to unseen out-domain images to be an essential feature of PIVQGAN. The model performance is inevitably downgraded in prior works when the testing images have unseen texture, color, or objects. However, our model can maintain its high image synthesis quality as long as the out-domain images are not too far away from the training image domain. Two factors contribute to such robustness. Firstly, the joint training scheme between GAN and the GAN-inversion encoder guarantees that the encoded latent vectors always lay in the latent space. Therefore, even for images with unseen visual attributes, $E$ still projects the image into the closest vector inside the latent space. Secondly, the VQ module will always reduce the shape feature-map into a certain number of learned object embeddings. It means the produced object-mask is always understandable to $G$, even for input images with unseen objects.

With the factors mentioned above, we show the synthesized images of PIVQGAN with unseen image domains in Fig. 5.6. We train our model on the Anime-Girl dataset in (a) and (c), which only contains animated female portraits. When we input a photo-realistic face as the identity image, the model automatically generates an “anime-girl” version of the input image. We can see that since the model has formed a solid prior bias on animated young girls, it translates all the faces into a young girl regardless of the input image’s gender and age. On the other hand, it translates the other facial attributes correctly, including hair color, skin tone, and posture, because all these attributes are already learned from the training dataset. When we input real human faces as the posture reference, the model transfers the pose without any problem. Similarly, in (b) and (d), the model is trained only on photo-realistic animal images while we fed toy animal pictures as pose or identity inputs. Again, the toy’s identity is well translated into a counterpart of the natural animal face, and our model captures the posture perfectly. In contrast, all the baseline models suffer from generating good quality results given the same training and input data. The out-domain performance reveals a promising performance of PIVQGAN on a zero-shot image domain translation task, which has not gained much research attention before to our knowledge.

5.3.4 More Qualitative Results
Figure 5.7: Generated results of PIVQGAN on more structural-complicated datasets: LSUN-cars and LSUN-churches. Note how the VQSN module is able to capture the abstract posture information, and the identity features can be accurately transferred among the same identity image among different postures.
Figure 5.8: Generated results of PIVQGAN on more structural-complicated dataset: cartoon stickers.

Figure 5.9: Cross-domain results of PIVQGAN trained on images joining cartoon stickers and AFHQ. PIVQGAN is able to stylized realistic animal faces into different stickers, as varied degrees. The style-transfer performance is superior and have the semantic-awareness property
Figure 5.10: Cross-domain results of PIVQGAN trained only on anime girl faces. Note how the model is able to translate sticker images from unseen image domain into the anime girls. It shows the great potential on PIVQGAN on the task of zero-shot image domain translation.

Figure 5.11: Cross-domain results of PIVQGAN trained only on anime girl faces. The model translates real human face images from unseen image domain to anime girls.
Figure 5.12: Qualitative comparison between PIVQGAN and StarGAN-v2. PIVQGAN is superior in image quality, posture translation, and identity accuracy.

Figure 5.13: Different self-learned posture representations from differently configured VQSN modules.
5.4 Conclusion

We proposed the VQSN module and the self-supervised joint training scheme. The resulted model, PIVQGAN, tackles the problem of unsupervised posture-and-identity translation and improves the state-of-the-art performance on this task. The experimental results showed that our model outperforms prior baselines in both image quality and translation accuracy.

Beyond posture-and-identity translation, the proposed VQSN module also unveils the immense potential of fully unsupervised image segmentation and semantic understanding. Moreover, the robust performance of PIVQGAN on unseen-domain images gives hints to a new task of zero-shot image domain translation. We hope our model paves the way for future studies in this new direction.
Chapter 6

Conclusion

This dissertation has investigated disentangling the Generative Adversarial Networks, and the applications of GAN disentanglement from three aspects: the unsupervised image latent space disentangling via mutual-information maximization; the text-and-image mutual translation via mutual-information maximization; and the image-to-image translation via posture-and-identity disentanglement.

First, we propose OOGAN, a robust framework that disentangles high-resolution images with high generation quality. Our one-hot sampling highlights the structural advantage of GANs for easy manipulation of the input distribution that can lead to disentangled representation learning, while the architectural design provides a new perspective on GAN designs. Instead of tweaking the loss functions (designing a new loss, adjusting loss weights, which are highly unstable and inconsistent across datasets), we show that sampling noise from multiple distributions to achieve disentanglement and interpretability is robust and straightforward. It leads to a promising new direction of training GANs, opening up important avenues for future research, e.g., choosing what distribution to sample from, allocating alternating ratios, etc. The impact of this goes beyond disentangling, as future research can also be conducted on model interpretability and human-controllable data generation. In the future, we plan to explore more dynamic and fluent sampling methods that can be integrated into the GAN framework for better performance, and we will attempt to validate the benefits of these sampling methods theoretically.

Second, we propose TIME, the Text and Image Mutual-translation adversarial nEtwork, a unified framework trained with an adversarial schema that accomplishes both the text-to-image and image-captioning tasks. Via TIME, we provide affirmative answers to the four questions we raised in Section 1. While previous works in the T2I field require pre-training several supportive modules, TIME achieves the new state-of-the-art T2I performance without pre-training. The joint process of learning both a text-to-image and an image-captioning model fully harnesses the power of GANs (since in related works, $D$ is typically abandoned after training $G$), yielding a promising Vision-Language performance using $D$. TIME bridges the gap between the visual
and language domains, unveiling the immense potential of mutual translations between the two modalities within a single model.

Finally, we propose PIVQGAN, a model consisting of the VQSN module and the self-supervised joint training scheme. The resulted model, PIVQGAN, tackles the problem of unsupervised posture-and-identity translation and improves the state-of-the-art performance on this task. The experimental results showed that our model outperformed prior baselines in image quality and translation accuracy. Beyond posture-and-identity translation, the proposed VQSN module also unveils the immense potential of fully unsupervised image segmentation and semantic understanding. Moreover, the robust performance of PIVQGAN on unseen-domain images gives hints to a new task of zero-shot image domain translation. We hope our model paves the way for future studies in this new direction.
References


