In addition to the latent space discussed in Chapter 5, a conditional text-to-image GAN model also contains the linguistic space, in which word and sentence embeddings obtained by a text encoder are used to modulate the visual feature map for semantic consistency. The linguistic space significantly controls and directs the image-generation process, playing an important role in synthesizing perceptually plausible and semantically consistent samples. For example, the important attribute properties (e.g., color and object category) of the foreground objects and the backgrounds in the generated samples are determined by textual input probes. We expect that our identified meaningful directions in the latent space can provide valuable insight into the correspondences between the latent control space and the obtained image variation. In this chapter, we present two basic techniques to take a deep look into the linguistic space of a conditional text-to-image GAN model. We introduce the linear interpolation between pairs of keywords to study what the model has learned within the linguistic embeddings as well as testing the influence of individual, different words on the generated sample. Subsequently, we extend a linear interpolation to a triangular interpolation conditioned on three corners for further analyzing the model and better conducting data augmentation. Experimental results on the proposed DiverGAN generator trained on two benchmark data sets represent an improvement in explainability in the analyzed algorithm, which qualitatively suggests the effectiveness of our presented techniques.

This chapter is based on


6.1 Introduction

A conditional text-to-image GAN model usually maps a latent code and linguistic embeddings into an image sample semantically aligning with the input text via adversarial training. To improve the explainability of a conditional text-to-image GAN architecture, we need to investigate the latent space and the linguistic space. We expect that our proposed semantic-discovery method in Chapter 5 can provide valuable insight into the relationship between the latent control space and the obtained image variation. However, it remains particularly difficult to explain what a conditional text-to-image GAN model has learned within the text space. How to understand the relation between the textual (linguistic) probes and the generated image factors? This constitutes the last research topic of this thesis.

To alleviate the difficult but highly relevant problem, we take a closer look at the linguistic space of a conditional text-to-image GAN architecture using linear interpolation between latent codes for textual contrasts and triangular interpolation. We start with a well-trained GAN generator for text-to-image synthesis. We qualitatively analyze the roles played by linguistic embeddings in the synthetic-image semantic space through linear interpolation between pairs of keywords. More specifically, we visualize the samples generated by linearly interpolating two contrastive keywords, e.g., the color, the size and the length of the beak on the CUB bird data set and the background, the object and the action on the MSCOCO data set. After that, we can observe the transition process from the initial image to the final sample. We show that although semantic properties contained in the picture change continuously in the latent space, the visual appearance of the image does not always vary smoothly along with the contrasting word embeddings. In addition, when given adequate training images, a conditional text-to-image GAN architecture may be able to correctly capture the semantics of some significant words in the linguistic space of the conditional input-text probes. The model therefore has the ability to accurately control the background (e.g., from ‘grass’ to ‘beach’), the object (e.g., from ‘animals’ to ‘men’) and the action (e.g., from ‘grazing’ to ‘skiing’) of complex scenes as well as the color of the bird (e.g., from ‘blue’ to ‘yellow’) and the length of the beak (e.g., from ‘long’ to ‘men’) with the help of the ‘linguistic’ linear interpolation. However, multiple meaningful contrasts can be learned, but there are areas where the method is not able to capture important variations along a dimension. This may be due to architectural or data-related limitations.

In order to further improve our insights, we extend a linear interpolation to a triangular interpolation for simultaneously investigating three latent codes or three keywords in the given textual description. We show that the transitions towards the three corner points are natural as well as smooth. In the meantime, a generative model is able to produce various visually pleasing pictures using the triangular interpolation on three Good latent codes.
We carry out comprehensive experiments on the proposed DiverGAN generator trained on two widely used data sets, i.e., the CUB bird [108] and MSCOCO [63] data sets, since DiverGAN has the ability to employ a generator/discriminator pair to synthesize diverse and high-quality samples on the basis of a single textual description. Experimental results in the current study represent an improvement in explainability in the analyzed algorithm. The contributions of this chapter can be summarized as follows:

- We introduce the pairwise linear interpolation between contrastive keywords for an improved explainability of the linguistic space of a conditional text-to-image GAN model.
- We extend a linear interpolation to a triangular interpolation for further analysis using a 2-simplex, i.e., triplet contrast, and better data augmentation.

6.2 PROPOSED METHODOLOGY

In this section, we introduce the textual linear interpolation to study what the model has learned within the linguistic space as well as testing the influence of individual, different words on the generated sample. Afterwards, triangular interpolation conditioned on three corners is presented for further analyzing the model and better performing data augmentation.

6.2.1 Preliminary

The goal of text-to-image generation is, given textual descriptions, to automatically produce perceptually realistic and semantically consistent image samples. Let \( S = \{(I_i, C_i)\}_{i=1}^{N} \) represent a set of \( N \) image-text pairs for training, where \( I_i \) indicates an image and \( C_i = (c^1_i, c^2_i, ..., c^K_i) \) denotes a suite of \( K \) natural-language descriptions, while \( S \) is cast into a training set and a testing set. Current conditional text-to-image GAN [127], [132], [140] approaches commonly follow the same paradigm. The generator \( G \) aims at yielding a visually plausible and semantically correlated sample \( \hat{I}_i \) according to a latent code randomly sampled from a fixed distribution and a text description \( c_i \) randomly picked from \( C_i \), where \( c_i = (w_1, w_2, ..., w_m) \) contains \( m \) words. The discriminator \( D \) is trained to distinguish the real image-text pair \( (I_i, c_i) \) from the synthetic image-text pair \( (\hat{I}_i, c_i) \).

Through the above training process, a conditional text-to-image GAN model is assumed to have the ability to synthesize high-quality, diverse and semantically related pictures, given a single textual description and different injected noise.

Chapter 5 studies the relationship between the latent control space and the obtained image variation. In addition to the latent space, a conditional text-to-image GAN model also contains the linguistic embeddings, in which word and sentence vectors are adopted to module the visual feature map for semantic
consistency. Despite high-quality pictures achieved by the existing approaches, we yet do not understand what a conditional text-to-image GAN model has learned within the linguistic space of the conditional input-text probes.

In order to understand ‘embeddings’ in deep learning, several methods have been proposed. A common method is to visualize the space using, e.g., t-SNE or k-means clustering. This may give some insights on the location of dominant image categories in the subspace. An alternative approach is to utilize - yet another - step of dimensionality reduction by applying standard PCA on the embedding. However, this still does not lead to good explanations and an easy controllability of the image-generation process. In this section, we start from Good latent vectors and introduce two basic techniques to provide useful insights into the explainability of the linguistic space of a conditional text-to-image GAN model.

6.2.2 Linear interpolation and semantic interpretability

We start with a well-trained and fixed generator $G(z, (w, s))$, which takes a random noise $z$, word embeddings $w$ of the given text description and the corresponding sentence vector $s$ as the inputs and outputs a photo-realistic and semantically consistent sample.

We study the linear interpolation between a pair of keywords in order to qualitatively explore how well the generator exploits the linguistic space of the conditional input-text probes as well as testing the influence of individual, different words on the generated sample. We can observe how the samples vary as a word in the given text is replaced with another word, for instance by using a polarity axis of qualifier key words (dark-light, red-blue, ...). More specifically, we can first acquire two word embeddings (i.e., $w_0$ and $w_1$) and two corresponding sentence vectors (i.e., $s_0$ and $s_1$) by only altering a significant word (e.g., the color attribute value and the background value) in the input natural-language description. Afterwards, the results are obtained by performing the linear interpolation between the initial textual description $(w_0, s_0)$ and the changed description $(w_1, s_1)$ while keeping the Good latent code $z$ frozen. Mathematically, this proposed text-space linear interpolation combines the latent code, the word and the sentence embeddings and is formulated as:

$$h(\gamma) = G(z, (1 - \gamma)w_0 + \gamma w_1, (1 - \gamma)s_0 + \gamma s_1)$$

(6.1)

where $\gamma \in [0, 1]$ is a scalar mixing parameter and $z$ is a successful latent code.

For the CUB bird data set, when we vary the color attribute value in the given sentence, we empirically explore what happens in the color mix: Do we, e.g., get an average color interpolation in RGB space or does the network find another solution for the intermediate points between two disparate embeddings?
In general, our presented text-space linear interpolation has the following advantages:

- The linear interpolation between a pair of keywords can be utilized to quantitatively control the attribute of the synthetic sample, when the attribute varies smoothly with the variations of the word vectors. For example, the length of the beak of a bird can be adjusted precisely via the text-space linear interpolation between the word embeddings of ‘short’ and ‘long’.

- When the attribute of the synthesized sample does not change gradually along with the word embeddings, we can exploit a text-space linear interpolation to produce a variety of novel samples. Take bird synthesis as an example: When conducting the linear interpolation between color keywords, \( G(z, (w, s)) \) is likely to generate a new bird whose body contains two colors (e.g., red patches and blue patches) in the middle of the interpolation results, as shown in Fig. 6.1.

- Through the linear interpolation between contrastive keywords, we can take a deep look into which keywords play important roles in yielding foreground images as well as which image (background) regions are determined by the terms in the text probe.

### 6.2.3 Triangular interpolation

We extend the linear interpolation between two points to an interpolation between three points, i.e., in the 2-simplex, for further studying \( G(z, (w, s)) \) and better performing data augmentation. Since this kind of interpolation forms a triangular plane, we name it triangular interpolation. Triangular interpolation is able to generate more and more diverse samples conditioned on three corners (e.g., latent vectors and keywords), spanning a field rather than a line.

#### 6.2.3.1 Triangular interpolation of latent codes.

Similar as in Equation 4.5, triangular interpolation of latent codes is achieved by employing three successful latent codes (i.e., \( z_0, z_1 \) and \( z_2 \)) as corner points to perform an interpolation in the 2-simplex field. It is denoted as:

\[
f(\gamma_1, \gamma_2) = G(((1 - \gamma_1 - \gamma_2)z_0 + \gamma_1z_1 + \gamma_2z_2, (w, s))
\]

where \( \gamma_1 \in [0, 1] \) and \( \gamma_2 \in [0, 1] \) are mixing scalar parameters and \( z_0 \) represents the initial latent code. When we fix the value of \( \gamma_2 \), triangular interpolation between \( z_0, z_1 \) and \( z_2 \) can be viewed as the pairwise linear interpolation between \( z_0 \) and \( z_1 \). Additionally, the number of samples synthesized by a linear interpolation will decrease as \( \gamma_2 \) increases. Therefore, when we sample 10 points with the initial interpolation (i.e., \( \gamma_2 = 0 \)), we can obtain 55 pictures via the triangular interpolation of latent codes. We want to explore how semantic properties contained in the images change with the variations of the latent vectors for the
triangular interpolation of latent codes. Some linguistic contrast may represent a smooth transition, whereas others will involve clear transition points. As an example, the question may be asked whether the midpoint between a blue and red corner is represented by purple in RGB, or by a patchy transition involving red and blue image elements flowing into one another.

6.2.3.2 Triangular interpolation and semantic interpretability

Similar to the linear interpolation between a pair of keywords, we need to derive three word embeddings (i.e., \( w_0, w_1 \) and \( w_2 \)) and three corresponding sentence vectors (i.e., \( s_0, s_1 \) and \( s_2 \)) as corners to define the presented text-space triangular interpolation:

\[
h(\gamma_1, \gamma_2) = G(z, (1 - \gamma_1 - \gamma_2)w_0 + \gamma_1w_1 + \gamma_2w_2, (1 - \gamma_1 - \gamma_2)s_0 + \gamma_1s_1 + \gamma_2s_2)
\]

where \( \gamma_1 \in [0, 1] \) and \( \gamma_2 \in [0, 1] \) are mixing scalar parameters and \( z \) is a successful latent vector.

For the sake of attribute analysis, we can obtain three new textual descriptions by replacing the attribute word in the initial natural-language description with another two attribute words. Then, through the triangular interpolation between keywords, the generator has the ability to yield pictures based on the above three attributes. Moreover, we expect that the text-space triangular interpolation should achieve the same visual smoothness as the text-space linear interpolation. In other words, when fixing the weight (i.e., \( \gamma_2 \)) of the third text in the triangular interpolation between keywords, the attributes of the image vary gradually along with the word embeddings if the interpolation results of a text-space linear interpolation between the first two textual descriptions change continuously.

The text-space triangular interpolation has obvious advantages over the linear interpolation between a pair of keywords. Firstly, the text-space triangular interpolation is able to produce more image variation to perform data augmentation than the pairwise linear interpolation. Secondly, we can simultaneously control two different attributes (e.g., color and the length of the beak) via the triangular interpolation between keywords. Thirdly, through the text-space triangular interpolation, three identical attributes (e.g., red, yellow and blue) can be combined to synthesize a novel sample.
6.3 experiments

In this section, to evaluate the effectiveness of the presented techniques, we conduct a set of experiments on the proposed DiverGAN generator trained on the CUB bird [108] and MSCOCO [63] data sets. To be specific, we clarify the details of experimental settings in Section 6.3.1. After that, the experiments in Section 6.3.2 are carried out to evaluate the textual linear-interpolation technique. Subsequently, the experiments in Section 6.3.3 and Section 6.3.4 are performed to prove the effectiveness of the proposed triangular-interpolation method.

6.3.1 Experimental settings

Implementation details. We take the DiverGAN generator as the backbone generator, which was pre-trained on the CUB bird and MSCOCO data sets. The steps of a linear interpolation are set to 10. We set the steps of \( \gamma_1 \) and \( \gamma_2 \) in a triangular interpolation to 10. Our approach is implemented by PyTorch [81]. A single NVIDIA Tesla V100 GPU (32 GB memory) is employed for all experiments.

6.3.2 Results of the linear interpolation between pairs of keywords

Fig. 6.1 shows the qualitative results of the textual linear interpolation of DiverGAN on the CUB bird data set, indicating that the attributes correlated with the synthesized sample do not always change gradually with the variations of word
Figure 6.2: The interpolated bird between <red> and <blue> is not simply purple.

Figure 6.3: Textual, ‘linguistic’ interpolation in case of semantically distant background keywords (e.g., beach and grass) of DiverGAN random latent-code samples on the MSCOCO data set, for four text input probes.

embeddings. For instance, the color of the bird does not vary continuously from ‘red’ to ‘blue’ in the first row. In the medium of interpolation results, DiverGAN generates multiple novel birds, whose bodies are composed of red and blue patches. However, the color attribute of the bird changes gradually from ‘red’ to ‘yellow’ in the second row. We are able to acquire an average color interpolation in RGB space by merging the first and second attributes. We can also see that in the third row, the length of the beak varies smoothly along with textual vectors, while other attributes remain unchanged. Furthermore, while the color of the beak changes continuously with the variations of word embeddings, the shape of the bird varies largely in the fourth row. The above results suggest that DiverGAN has the ability to capture the significant words (e.g., the color of the body and the length of the beak) in the given textual description. More importantly, by exploiting the characteristic as well as the linear interpolation between a pair of keywords, we can precisely control the image-generation process while producing various novel samples.
As can be seen in Fig. 6.2, the bird in the middle of the interpolation strip is not colored purple, but contains small red and blue patches. This illustrates the effect of pixel and channel-aware attention modules in DiverGAN.

To investigate the linguistic space of DiverGAN trained on the MSCOCO data set, we visualize the samples generated by linearly interpolating two contrastive keywords (i.e., two keywords from the background, the object and/or the action) in Fig. 6.3, Fig. 6.4 and Fig. 6.5, respectively. As shown in Fig. 6.3, DiverGAN correctly identifies ‘beach’, ‘snow’, ‘grass’ and ‘street’ while generating the corresponding image samples. We can also observe that the background appearance in the edited sample varies along with the background-attribute words in the textual description and other image contents (e.g., objects) remain unchanged, which suggests that DiverGAN can effectively disentangle the semantics of the input natural-language description while also accurately controlling the significant regions of the picture.

The qualitative results of the linear interpolation between two object-property keywords on the MSCOCO data set are presented in Fig. 6.4. We can see that DiverGAN synthesizes semantically consistent pictures according to the modified text, which indicates that the model learns the linguistic embeddings of the words ‘sheep’, ‘skiers’, ‘men’ and ‘pizza’. It can also be observed that although we change the ‘object’ word from ‘animals’ to ‘skiers’ (2nd row) and from ‘people’ to ‘skiers’ (4th row), the background significantly varies from ‘grass’ to ‘snow’ and
Figure 6.5: Textual, ‘linguistic’ interpolation in case of semantically distant action keywords (e.g., surfing and skiing) of DiverGAN random latent-code samples on the MSCOCO data set, for three text input probes.

Figure 6.6: Unsuccessful textual interpolation of DiverGAN random latent-code samples on the CUB data set, for four text input probes.
Figure 6.7: Challenging ‘linguistic’ interpolation in case of semantically distant background keywords in DiverGAN random latent-code samples on the MSCOCO data set, for four text input probes. For the third row, the desired attribute (i.e., a street) is not emerging.

from ‘beach’ to ‘snow’, which demonstrates that some words (e.g., ‘skiers’) play a vital role in the generation process of image samples. More importantly, a pizza appears on the waves when we modify the ‘object’ word from ‘man’ to ‘pizza’. It shows that DiverGAN may have the ability to decompose some complex scenes into foreground objects and backgrounds that can be edited using text.

As can be observed in Fig. 6.5, the background (i.e., beach and grass) and the objects (i.e., animals and men) vary along with the linguistic embeddings when changing the ‘action’ word in the input textual description, which reveals that some ‘action’ words contribute to producing the background and object regions in the image-generation process. For example, when we modify the ‘action’ word from ‘grazing’ to ‘skiing’ in the 2nd row, the background varies smoothly from ‘grass’ to ‘snow’ and the objects change continuously along with the word embeddings.

The above analysis indicates that when given adequate training images, DiverGAN is able to control the background (e.g., from ‘grass’ to ‘beach’), the objects (e.g., from ‘animals’ to ‘men’) and the action (e.g., from ‘grazing’ to ‘skiing’) of complex scenes with the help of the textual, ‘linguistic’ linear interpolation, since DiverGAN is able to capture the corresponding semantics in the linguistic space of the conditional input-text probes.

In addition to visualizing the effective examples of the linear interpolation between a pair of keywords, we also present some unsuccessful results in Fig. 6.6, Fig. 6.7 and Fig. 6.8, respectively. Fig. 6.6 shows the unsuccessful ‘linguistic’-interpolation samples obtained by DiverGAN trained on the CUB data set. We can see that the size of the bird (1st and 2nd row) does not vary with the variations
of the words (from ‘small’ to ‘big’ and from ‘small’ to ‘medium’). Furthermore, the color of the eyes (2
row) and the length of the tarsus (3
row) unfortunately do not change along with the words (from ‘black’ to ‘red’ and from ‘medium’ to ‘short’). It suggests that the model does not capture the size, the eyes and the tarsus of the bird.

The unsuccessful results of the linear interpolation between two background-attribute words on the MSCOCO data set are depicted in Fig. 6.7. Users may experiment with semantically unrelated words in order to try to generate interesting patterns and we need to know what happens. It can be observed that ‘street’ and ‘grass’ can be learned (shown in Fig. 6.3), but the background does not change or slightly changes from ‘waves’ to ‘grass’ in the 2
row and from ‘grass’ to ‘street’ in the 3
row according to the modified text. This may be due to the limitation of the modulation modules that fine-tune the feature map using the attention weights of the individual, different words in the given textual description. For example, ‘wave’ is assigned greater weight than ‘grass’ in the sentence: ‘A man riding a wave on top of the grass’. ‘Grazing’ plays a more significant role than ‘street’ when yielding the samples according to the natural-language description: ‘A bunch of animals grazing on a street’. We can also see that the
background does not change along with the words from ‘beach’ to ‘ocean’ in the 1st row and from ‘grass’ to ‘tower’ in the 4th row. The reason behind this result may be that DiverGAN cannot learn the semantic meaning of some contrastive keywords owing to the data-related limitation.

Fig. 6.8 visualizes the unsuccessful pictures synthesized by linearly interpolating two attribute keywords, i.e., the object and the action. We can see that the objects in the synthesized sample (1st, 2nd and 4th row) unfortunately do not change or slightly change with the variations of the words (from ‘man’ to ‘sheep’, from ‘people’ to ‘sheep’ and from ‘animals’ to ‘cows’). Moreover, although the objects on the grass vary from animals to be like a pizza in the 3rd row, the textures remain green. It can also be observed that the image does not change in the 5th row when modifying the word from ‘standing’ to ‘grazing’. The size
of the object in the 6th row varies along with the action word from ‘grazing’ to ‘walking’.

At this point we can conclude that many meaningful contrasts can be learned (shown in Fig. 6.3, Fig. 6.4 and Fig. 6.5), but there are areas where the method is not able to capture important variations along a dimension. This may be due to architectural or data-related limitations. In order to improve our insights, we will look at a triangular interpolation in the next subsection.

6.3.3 Results of the triangular interpolation of latent codes

To better understand a triangular interpolation, we show the samples generated by linearly interpolating three Good latent vectors on the CUB bird and MSCOCO
data sets in Fig. 6.9 and Fig. 6.10, respectively. As can be observed in Fig. 6.9, the semantics contained in the generated image, e.g., the background, the visual appearances of footholds, the shape, pose and orientation of the bird, change gradually with the variations of latent codes in two dimensions. In addition, the interpolation results achieve a balanced triangular shape within the triangle, such that the center marked in red is the combination of three latent vectors. If the application concerns data augmentation, 55 believable samples are obtained by performing the triangular interpolation of three successful latent vectors.

The qualitative results of the triangular interpolation of latent vectors on the MSCOCO data set are presented in Fig. 6.10. We can see that both the background appearance and the foreground objects vary smoothly along with the latent codes. More importantly, the objects of three initial image may appear in the same sample marked in red.
6.3.4 Results of a triangular textual interpolation

The triangular interpolation for linguistic attributes (i.e., the points between \textit{blue, red, yellow} and the points between \textit{snow, grass, beach}) in two dimensions is shown in Fig. 6.11 and Fig. 6.12, respectively. As can be observed in Fig. 6.11, the color attribute changes continuously with the variations of the word from ‘red’ to ‘yellow’, whereas the transition process from ‘red’ to ‘blue’ is not smooth, such that multiple birds are composed of patches of different colors, i.e., \textit{(orange, blue)}, \textit{(yellow, blue)} and \textit{(red, blue)}. Furthermore, the color of the center bird is the combination of three attributes, i.e., \textit{red, blue, yellow}.

Fig. 6.12 visualizes the results of the triangular textual interpolation on the 
\textit{MSCOCO} data set. We can observe that the transitions towards the three cor-
ner points are natural as well as smooth. The background significantly varies continuously along with the word embeddings in two dimensions.

6.4 CONCLUSION

In this chapter, we present two basic techniques for an improved explainability of the linguistic space of a conditional text-to-image GAN model. To provide valuable insight into the relationship between the linguistic embeddings and the synthetic-sample semantic space, we conduct the linear-interpolation analysis between pairs of keywords. Meanwhile, we extend a pairwise linear interpolation to a triangular interpolation conditioned on three corners to further analyze the model and better conduct data augmentation. We evaluate our proposed approaches on the DiverGAN generator trained on two very different data sets, i.e., the single-object CUB bird data set and the complex-scene MSCOCO data set. We show that semantic properties contained in the generated image change gradually with the variations of latent codes, but the attributes of the synthesized sample do not always vary continuously along with the word embeddings. Furthermore, we find that the generator of DiverGAN cannot capture the size of the object due to the mechanism of the convolutional neural network and cannot understand some words in the given textual description owing to the limitation of the data set. More importantly, we observe that the samples generated by a triangular interpolation achieve a balanced triangular shape within the triangular and the transition towards the three latent vectors or three significant words are continuous as well as natural.