LatentVis: Investigating and Comparing Variational Auto-Encoders via Their Latent Space

Xiao Liu[a], Junpeng Wang[b]

[a]Department of Computer Science and Engineering, the Ohio State University, 2015 Neil Avenue, Columbus, Ohio, 43210, USA
[b]Visa Research, 385 Sherman Avenue, Palo Alto, California, 94306, USA

Abstract

As the result of compression and the source of reconstruction, the latent space of Variational Auto-Encoders (VAEs) captures the essences of the training data and hence plays a fundamental role in data understanding and analysis. Focused on revealing what data features/semantics are encoded and how they are related in the latent space, this paper proposes a visual analytics system, i.e., *LatentVis*, to interactively study the latent space for better understanding and diagnosing image-based VAEs. Specifically, we train a supervised linear model to relate the machine-learned latents with the human-understandable semantics. With this model, each important data feature is expressed along a unique direction in the latent space (i.e., semantic direction). Comparing the semantic directions of different features allows us to compare the feature similarity encoded in the latent space, and thus to better understand the encoding process of the corresponding VAE. Moreover, *LatentVis* empowers us to examine and compare latent spaces across various training stages, or different VAE models, which can provide useful insight into model diagnosis.

Keywords

Deep generative model, variational auto-encoder, latent space, semantics, visual analytics

1. Introduction

With the powerful capability in feature extractions, Deep Neural Networks (DNNs) have made a series of breakthroughs across a wide range of applications, e.g., image classification [1], object recognition [2], image segmentation [3], etc. More interestingly, DNNs also demonstrate excellent performance in feature generations, which has attracted more research attention [4]. For example, Generative Adversarial Nets (GANs) and Variational Auto-Encoders (VAEs) are able to generate data (including images [5], sounds [6]) that are almost indistinguishable from real data.

The outstanding performance of DNNs comes from their complicated internal model architectures and the long-time model training processes, which, however, have gone far beyond humans' interpretability. As a result, it is very difficult to explain how Deep Generative Models (DGMs) understand the extracted features and further use them to generate new features. The latent spaces of these models, located at the pivot point between extraction and generation, compress all the extracted features and control what to be generated. Investigating them could help to understand and diagnose the DGMs, and thus shed light on the mystery power of DGMs. However, those latent spaces are usually with high-dimensionality and the semantics of individual latent dimension is not human-understandable.

Recently, we have witnessed many works on interpreting the latent space of DNNs. Some considered a latent space as a high-dimensional manifold and focused on the geometric interpretation of the manifold. For example, [7] showed that geodesic curves on the latent space manifold are approximately straight in their experiments. [8] revealed that a stochastic Riemannian metric in the latent space could produce smoother interpolations than the conventional Euclidean distance. With static visualizations of the geometric path in the latent space, these studies have helped to understand the abstractive manifold holistically.

Others explored the semantics of different latent spaces by focusing on specific tasks. For example, [9, 10] analyzed the word embedding and verified the linear arithmetic of the semantics in the embedding/latent space, e.g., $queen - woman + man \approx king$. Similar linear arithmetic has also been found in the latent space of image-based DGMs [4]. These studies expose some structures of the latent spaces, but are still insufficient to comprehensively reveal their essential semantics.

This paper targets to diagnose image-based VAEs by interactively investigating their latent space, and hence answers three concrete research questions: (1) what semantics are embedded in the latent space of VAEs; (2) how can we transfer the machine-learned la-

Proceedings of the CIKM 2020 Workshops, October 19-20, 2020, Galway, Ireland
EMAIL: liu.5764@osu.edu (X. Liu); junpeng.wang.nk@gmail.com (J. Wang); The work was done while the author was at The Ohio State University. (J. Wang)
ORCID: 0000-0002-6303-0771 (X. Liu); 0000-0002-1130-9914 (J. Wang)

tent space to a human-understandable semantic space for better interpretation; (3) how to use the latent spaces of VAEs to track and compare VAE models. To the end, we design and develop a three-module visual analytics prototype, named *LatentVis*, for this matter. The Data module presents an interface to interact with the experimental dataset and select images with desired features. The Semantics module identifies and compares semantic directions of different image features, bridging the machine-encoded latents with human-understandable semantics. The Comparison module compares the latent space of (1) the same model in two different training stages, (2) the same model from two separate trainings with randomly initialized network parameters, and (3) two different VAE models. To sum up, the contributions of this paper are three-fold:

- We present *LatentVis*, a visual analytics system that helps to understand and diagnose VAEs by interactively revealing the encoded semantics of the latent space.
- Enlightened by the linear arithmetic of features, we use a linear model to transfer a machine-learned latent space into a human-understandable semantic space.
- Based on our analysis of the latent space, we propose a model-agnostic approach to compare VAEs, across training stages, separate trainings, or different VAE models.

2. Background and Related Works

Interpreting Latent Spaces. DNNs can be considered as functions that transfer data instances from the input data space to a latent space ($f : R^m \rightarrow R^n$). A well-trained DNN will preserve the essential information of the input data during this transformation. However, due to the complexity of DNNs, it is a non-trivial problem to reveal or verify what information is preserved and how it is preserved in the latent space. Targeted on this problem, many research efforts have been devoted to interpret the latent spaces of DNNs. For example, [11] showed how the statistics of data can be examined in the latent space representation. [12] interpreted the association between visual concepts and symbolic annotations captured by βVAE through parallel coordinates plots. Latent embedding learning methods (GLO, LEO, GLANN [13, 14, 15]) were also developed for the interpretation and understanding of latent spaces. Moreover, in natural language processing, the learned embedding of words/ paragraphs also form a latent space. [9] and [16] interpreted this space and found that the correlations between words/ paragraphs were well-captured in the space.

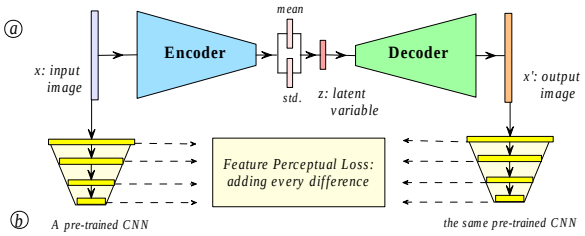

Figure 1: (a) The architecture of VAE; (b) the perceptual loss, introduced in DFC-VAE [27], for feature reconstruction.

Visual Analytics for Deep Learning (VIS4DL). There are two groups of VIS4DL works in general. One focuses on a specific model to reveal the internal working mechanism of the model, such as CNNVis [17], GANViz [18], and ActiVis [19]. These works usually design a visualization system to expose the hidden features and feature connections, for specific DNNs on specific datasets. Some works in this group also tried to generalize to different models on various datasets. For example, [20] proposed Network Dissection to quantify the interpretability of latent representations captured by CNNs (AlexNet, VGG, GoogLeNet, ResNet) via the alignment between hidden units and semantic concepts. The other group focuses on using only the model inputs and outputs to interpret/diagnose the model, without touching the intermediate model details (i.e., model-agnostic). For example, [21] proposed a model-agnostic approach to reveal the dominant regions of input images in controlling the prediction results of a classifier. More examples in this group also include [22, 23, 24]. Our work needs no examination on the internal working mechanism of VAEs (as semantics are encoded in the space formed by activations, rather than individual neurons [25, 26]), and thus belongs to the second group. Integrating a linear space transformer into our visual analytics process, we try to present a human-understandable latent space to diagnose DGMs.

Variational Auto-Encoder (VAE) [28] aims to reconstruct the input image from a latent representation of the image encoded/learned by itself. It is comprised of two neural networks: an encoder network encodes the image into a latent variable, and a decoder network decodes the image from the latent variable (Fig. 1a). Specifically, the encoder maps an input image x to a latent variable z (i.e., $z = encoder(x) \sim q(z|x)$), and

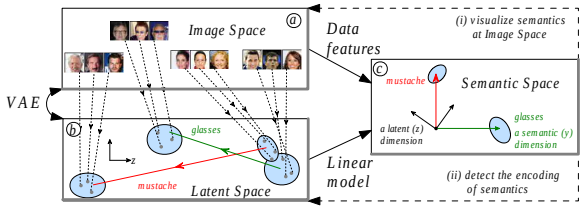

Figure 2: Three spaces: (a) the image space is where the CelebA images reside, each pixel is an independent dimension; (b) the latent space is the VAE learned representation of those images, the VAE encoder and decoder enable the transformation between the image space and latent space; (c) the semantic space is derived from the latent space under the supervision of the 40 binary features of the images (using our linear model).

the decoder maps a latent variable z to an output image x' (i.e., $x' = decoder(z) \sim p(x|z)$). The encoder and decoder, defined by trainable parameters θ and ϕ respectively, are optimized via minimizing the following loss function:

$$l(\theta, \phi) = -E_{q_\theta(z|x)}[log p_\phi(x|z)] + KL(q_\theta(z|x) \| p(z)).$$

By $z = encoder(x)$, the latent variable of a specific image from the VAE is readily accessible for further semantic explorations. One common issue of VAEs is that the generated images tend to be blurry, due to the aggregated pixel-wise image distance used in the loss function, i.e., the L_2 distance between x and x'.

Deep Feature Consistent VAE (DFC-VAE) [27] is a variant of the regular VAE. It improves the quality of the reconstructed images by replacing the pixel-wise reconstruction loss with a feature perceptual loss [29]. In DFC-VAE, multiple levels of features are extracted from both the input and reconstructed images by passing them into a pre-trained CNN. Each layer of the CNN extracts certain levels of image features. The features from the input and reconstructed images are then used to measure their perceptual distance (Fig. 1b, the L_2 loss between the corresponding feature maps). In this work, we adopted this perceptual loss to improve the reconstruction quality. The VGG19 [30] pre-trained on the ImageNet data is used as our pre-trained CNN.

3. Methodology

3.1. Fundamental Concepts

Image Dataset. We focus on a face image dataset, i.e., CelebA [31], to explore the latent space of VAE models in this work. This dataset is constituted of 202599 human face images. Each image has 40 binary attributes (e.g., the image is a male face or not, a face with glasses or not) with a resolution of 178×218. We pre-processed those images by cropping them into 148×148 and scaling down to 64×64 for our VAEs. Images with the same feature (i.e., have the same value on a binary attribute) belong to the same feature category.

We focused on this face image dataset for two reasons. First, this dataset presents rich attributes for the same object (the human face) in the same scale. Compared to other datasets with numerous objects in different scales (e.g., ImageNet), a VAE can more accurately capture the underlying data distribution. Second, the well-labeled attributes in this dataset can help to interpret the semantics encoded in the latent space of VAEs, through which, we derived the semantic space using our linear model (Fig. 2).

Image Semantics. The semantics of face images is the existence and scale of the 40 features in the CelebA dataset. A well-trained VAE can transfer the semantics from the image space to the VAE's latent space (i.e., from Fig. 2a to 2b). However, the transferred semantics in the latent space is not human-understandable. Hence, our goal is to interpret them via a semantic space (Fig. 2c), in which, we can explore if the image semantics have been accurately encoded (Fig. 2i) and how they are encoded (Fig. 2ii).

Semantic Direction. In Fig. 2b, we use a point to denote the VAE encoded latent variable for the corresponding image in the image space. All latent variables for images of the same category (e.g. "glasses", "mustache") form a cluster in the space, denoted as a blue bubble in Fig. 2b. We identify the direction from one cluster without a particular feature to the cluster with that feature as the *semantic direction* for the feature. For example, in Fig. 2b, the directions on the red and green lines reflect the semantic directions for "mustache" and "glasses". Moving the latent variable of one image along a semantic direction will change the corresponding feature of the reconstructed image **the most**. Along this semantic direction, a vector with a certain length is referred to as a *Semantic Vector*. For CelebA, there are 40 features, and we have 40 unique semantic directions.

3.2. Our Contributions

The Linear Model. Enlightened by the linear arithmetic of features (e.g., $woman\ face\ without\ glasses + (man\ face\ with\ glasses - man\ face\ without\ glasses) \approx woman\ face\ with\ glasses$) [4], we trained a linear model to quantify the semantic directions, as well as to trans-

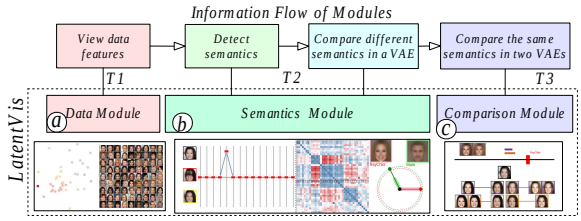

Figure 3: The framework of *LatentVis* system.

form the latent space to a human-understandable semantic space (i.e., from Fig. 2b to 2c). The linear model, $\mathbf{y} = W^T \cdot \mathbf{z} + \mathbf{b}$, is trained using the latent variable of all CelebA images encoded by VAEs (denoted as $\mathbf{z}$) and the 40 binary attributes of the images (denoted as $\mathbf{y}$). $\mathbf{z}$ is a vector with many dimensions (the black arrow in Fig. 2b), and $\mathbf{y}$ is a vector with 40 dimensions (the green arrow in Fig. 2c). Each row of the weight matrix W^T is the derivatives of a certain feature to all the latent dimensions, which represents a semantic direction in the latent space. Each column of W^T represents the contributions of a latent dimension to all the semantics.

Analytic Tasks and *LatentVis*. We focus on three analytic tasks to better understand and diagnose VAEs in a lens of their latent space: (T1) navigating the dataset and feature selection; (T2) visualizing and comparing the image semantics/ semantic directions in a latent space; and (T3) facilitating model comparisons and diagnosis through comparing their latent spaces. Following these tasks, we propose a visual analytics system, *LatentVis* (Fig. 3, bottom), which contains three analytical modules corresponding to an hierarchical information flow (Fig. 3, top).

The *Data Module* (Fig. 3a) gives an overview of the studied dataset, allowing us to flexibly explore data instances. It is also an interface to select any interested feature category and data instance for further analysis in other modules (please check details in the Appendix).

The *Semantics Module* (Fig. 3b) demonstrates what semantics has been captured by the VAE, and how the data features are correlated in the latent space by connecting the image space with the VAE latent space. Its three views follow an hierarchical information flow to detect, cluster, and compare semantics (see details in Sec. 4.1).

The *Comparison Module* (Fig. 3c) compares the semantic directions of latent space from different VAEs to diagnose these models. The diagnosis for the compared VAEs is performed by examining the learning work division between the encoder and decoder (see details in Sec 4.2).

4. Experiments and Results

Neural Network Structure. We worked with one regular VAE and one DFC-VAE, both with an encoder and a decoder of four convolutional layers. The four-layer encoder compresses the 64×64×3 CelebA images to 32×32 ×32, 16×16×64, 8×8×128, and 4×4×256. The compression result is then flattened and mapped to a 100D Gaussian distribution, represented by a 100D mean and a 100D standard deviation, through two fully-connected layers. The decoder has a symmetric structure with the encoder, but with a reversed order of the layers to up-sample the 100D latent variables (sampled from the 100D Gaussian).

The difference between the VAE and DFC-VAE is whether a pre-trained VGG19 model was used to compute the perceptual loss. We trained the VAE once and the DFC-VAE twice with the same batch size and the Adam optimizer in all trainings. However, notice that the hyperparamters for these trainings could be the same or be different on-purpose to compare the trained models and investigate the effect of the hyperparameters, e.g., comparing models trained with different learning rates to study their convergence speed.

All three trainings used the 202599 CelebA images and the batch size is 64. Every 800 batches were considered as a training stage to collect model statistics, like loss-values, and all the three trainings were run for 197 stages (i.e., 157600 batches).

4.1. Detecting and Comparing Semantic Directions

We propose Algorithm 1 to detect and compare semantic directions in the Semantics Module (Fig. 4). First, we give all CelebA images to the well-trained VAE model to obtain their latent variables. Then, we use these latent variables and their corresponding 40 binary attributes to train a linear model to capture the semantic directions encoded in the latent space. To verify the effectiveness of the semantic directions, we also generate many random directions in the latent space. Given any selected image, we visualize the modified version of the image resulted from changing its latent variable along the 40 semantic directions and a random direction. For example, by dragging the control point for the feature “bangs" and “glasses" in Fig. 4a (i.e., change the length of a semantic vector, $\lambda \in [-10, 10]$), we observed how those two features were added (Fig. 4-a2) to the selected image (Fig. 4-a1). However, no obvi-

ous changes towards a particular feature were found in the image when dragging the control point on the axis representing random directions (Fig. 4-a3).

Algorithm 1 : Detecting and Comparing Semantic Directions

Require: images $X = \{\mathbf{x}_i\}_{i=1}^{n}$, feature labels $Y = \{\mathbf{y}_i\}_{i=1}^{n}$
Require: selected image $\hat{\mathbf{x}} \in X$, selected feature f, compared feature f', the length of a semantic vector λ
Require: the VAE model with an *encoder* and a *decoder*

for i in $range(n)$ **do**
  $\mathbf{z}_i \leftarrow encoder(\mathbf{x}_i)$
end for
train a linear model $\mathbf{y} = W^T \cdot \mathbf{z} + \mathbf{b}$
semantic directions $D \leftarrow W^T$
$\mathbf{d}_f \leftarrow D[f, :]$ // a row of D corresponding to feature f
randomly initialize $\mathbf{d}_r$ with $\|\mathbf{d}_r\| = \|\mathbf{d}_f\|$
$\hat{\mathbf{z}} \leftarrow encoder(\hat{\mathbf{x}})$.
$\hat{\mathbf{z}}_f \leftarrow \hat{\mathbf{z}} + \lambda \mathbf{d}_f$
$\hat{\mathbf{z}}_r \leftarrow \hat{\mathbf{z}} + \lambda \mathbf{d}_r$
$\hat{\mathbf{x}}_f \leftarrow decoder(\hat{\mathbf{z}}_f)$
$\hat{\mathbf{x}}_r \leftarrow decoder(\hat{\mathbf{z}}_r)$
$visualize(\hat{\mathbf{x}}, \hat{\mathbf{x}}_f, \hat{\mathbf{x}}_r)$ to verify semantics in $\mathbf{d}_f$ //Fig. 4a
$\mathbf{x}_f \leftarrow decoder(\lambda \mathbf{d}_f)$
$\mathbf{d}'_f \leftarrow D[f', :]$ // another row of D corresponding to feature f'
$\mathbf{x}_{f'} \leftarrow decoder(\lambda \mathbf{d}_{f'})$
$visualize(\lambda \mathbf{d}_f, \lambda \mathbf{d}_{f'}), (\mathbf{x}_f, \mathbf{x}_{f'})$ to compare semantics //Fig. 4c

We cluster the 40 feature categories (40 semantics) into five groups using the K-means algorithm based on the cosine similarity between their corresponding semantic directions. Interestingly, the feature categories inside each group present similar semantics. For example, the feature "make up", "no beard", and "attractive" are in the same group, which are all "women-ish" features. With the similar logic, the other four semantic groups are named "man-ish" (e.g., "mustache", "five-o'clock shadow"), "weak-woman-ish" (e.g., "smile", "bangs"), "weak-man-ish"(e.g., "bushy eyebrows", "hat"), and "old-ish" (e.g., "chubby", "bald"). The five clusters can be easily identified from the symmetric pair-wise similarity matrix (Fig. 4b), and we can select any two semantics (i.e., one row and one column) for comparison. For example, Fig. 4c shows the negative correlation between the "rosy cheeks" and "male" feature,

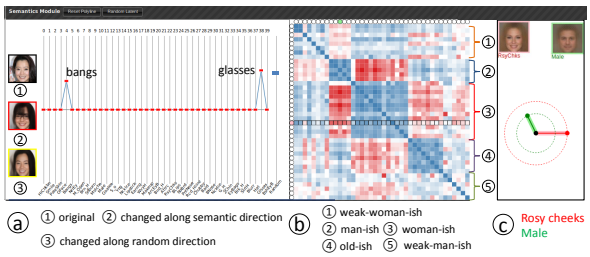

Figure 4: Semantic Module on DFC-VAE from the 197^{th} training stage: (a) detect, (b) cluster, and (c) compare semantic directions. The cell color from red over white to blue in the matrix (b) indicates the cosine similarity of two semantic directions from -1 over 0 to 1; (b1-b5) represent five clusters: "weak-woman-ish", "man-ish", "woman-ish", "old-ish", and "weak-man-ish".

which are from the "woman-ish" (Fig. 4-b3) and "man-ish" (Fig. 4-b2) groups respectively. The negative correlation is indicated by an obtuse angle between the two colored semantic vectors (i.e., $\lambda \mathbf{d}_f$ and $\lambda \mathbf{d}_{f'}$ in Algorithm 1). To visualize these two high-dimensional vectors $\lambda \mathbf{d}_f$ and $\lambda \mathbf{d}_{f'}$ in a 2D plot intuitively, $\lambda \mathbf{d}_f$ (corresponding to the selected feature) is always along the horizontal direction, and $\lambda \mathbf{d}_{f'}$ (corresponding to the compared feature) presents an angle with $\lambda \mathbf{d}_f$, calculated as the angle between them in the original HD latent space. The length of each colored segment reflects the norm of the corresponding semantic vector. Dragging the green/red point in Fig. 4c to the opposite direction (i.e., change λ to a negative value), we can also verify that the opposite direction indeed encodes the opposite feature, e.g., "pale skin" is the opposite of "dark skin". Interestingly, we found several such pairs, showing a similar way of how human understand these semantics, such as "smile" v.s. "scary face", "bangs" v.s. "high hairlines".

From the above explorations and visual evidence, we feel confident to believe the following hypothesis on the semantic structure of the latent space: (1) latent space tends to encode semantics along unique directions (i.e., semantic directions); (2) smaller angles between semantic directions denote similar semantics and opposite semantic directions encode opposite semantics.

4.2. Comparing Semantics across VAEs

The Comparison Module compares two VAE models and facilitates model diagnosis using their latent spaces. The comparison is across different training stages, trainings (with randomly initialized neural network parameters), and VAE models. For each pair of compared

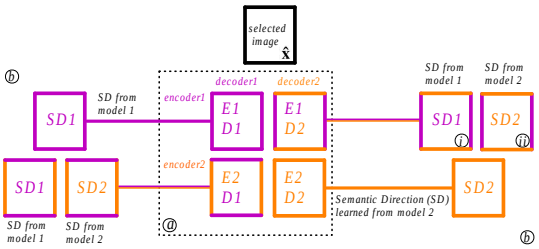

Figure 5: The mapping between images in the Comparison Module. (a) Reconstructed images from different combinations of encoders and decoders of two compared VAEs; (b) reconstructed images when changing along the semantic direction (SD) learned by different VAEs.

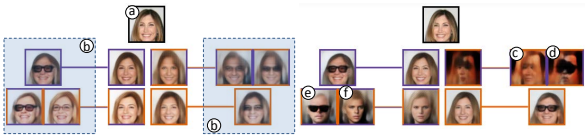

Figure 6: The Comparison Module on the " glasses" feature. Left: the orange and purple color represent the DFC-VAE model from the 3^{rd} and the 197^{th} training stages. Right: the orange and purple color represent two separate trainings of the same DFC-VAE model from the 197^{th} training stage.

models, we run Algorithm 1 to obtain their individual semantic directions and generate the corresponding reconstructed images. Given any selected image, Algorithm 1 also outputs its latent variable, from which, we can regenerate the image with a VAE's decoder. We can use the same VAE's decoder and a different VAE's decoder to reconstruct two images and compare them. From the comparison, we can track the work division between an encoder and a decoder, and also diagnose which of these two networks is more responsible for certain model functions.

Since we have two encoders and two decoders, there are four possible combinations between them. Each rectangle in Fig. 5a represents one combination. We use purple and orange color to denote model 1 and model 2. The left and right borders' color of a rectangle reflects which model's encoder is in use, whereas the top and bottom borders' color reflects which model's decoder is in use. For example, the top right rectangle in Fig. 5a means the reconstructed image uses encoder 1 (the left and right borders of the rectangle are in purple) and decoder 2 (the top and bottom borders of the rectangle are in orange), i.e., the pair of E1-D2.

When dragging the horizontal control point (i.e., change λ) in the top of the Comparison Module, these four images will be modified along the semantic directions (SD) for the focused semantics learned from the two models. Fig. 5b shows the mappings, i.e., which image is changing along which semantic direction. For examples, Fig. 5i and Fig. 5ii show the reconstructed images when changing the top right image of Fig. 5a along the semantic directions learned from model 1 and model 2, respectively.

Comparing Semantic Directions Across-Time. We take the DFC-VAE in an early training stage and a well-trained stage to perform the comparison. Fig. 6 investigates the DFC-VAE model with parameters from the 3^{rd} (orange) and the 197^{th} (purple) training stage. When moving the latent variable of the image in Fig. 6a along the semantic direction learned in those two stages, all the six reconstructed images generated the "glasses" feature (as shown in Fig. 6b), regardless of the swapping of the encoders and decoders from the two training stages. This observation indicates that different stages of the training encode the semantic direction of the same feature in a consistent way. It also implies another insight on the semantic structure of the latent space, i.e., the semantic directions may have a tolerable range, within which, the learned semantics is evolving over the training process.

Comparing Semantic Directions Across-Training. Focusing on a well-trained stage, we compared the DFC-VAE from two separate trainings (where the model parameters are randomly initialized in each training). Our goal is to explore whether the semantics is encoded in the same way over the two trainings. We used the same training hyperparameters and trained the same model twice with enough epochs. Fig. 6c investigates the semantic directions of the "glasses" feature from the two trainings. We found the "glasses" feature can only be generated when using the matched encoder-decoder pairs. For example, Fig. 6c reconstructs an image using the decoder from the second training, but the latent variable is moved along the semantic direction learned from the first training. As a result, the "glasses" feature was not generated. Conversely, the "glasses" feature could be generated when moving along the semantic direction learned from the second training, as shown in Fig. 6d. The results shown in Fig. 6e (image with "glasses"), 6f (image without "glasses") further verify this. The observation indicates that different trainings of the same VAE may encode the semantic direction of the same feature differently.

4.3. Diagnosing VAEs via Semantics Comparison

Learning Process Comparison between Encoders and Decoders. To interpret the learning work division be-

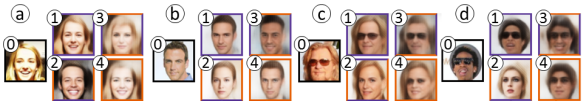

Figure 7: Images reconstructed from model 1 (DFC-VAE, 197^{th} stage, in purple) and model 2 (DFC-VAE, 3^{rd} stage, in orange) with matched and swapped encoder-decoder pairs. The numbers 1, 2, 3, 4 represent the combinations of E_1-D_1, E_2-D_1, E_1-D_2, E_2-D_2, respectively.

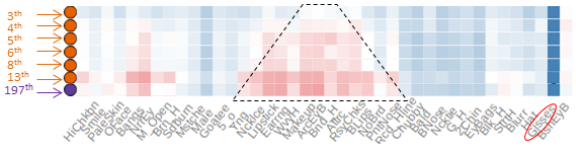

Figure 8: Comparing the cosine similarity between the "glasses" semantic direction and other semantic directions across seven training stages; red over white to blue denotes values from -1 over 0 to 1.

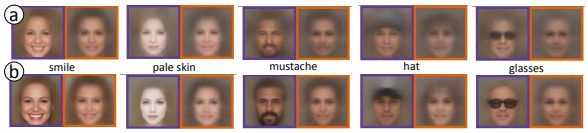

Figure 9: Comparing the reconstructed images along semantic directions at λ = 4.5 from DFC-VAE (purple borders) and VAE (orange borders) in (a) the 3^{rd} and (b) 197^{th} stage.

Figure 10: Comparing the semantic correlations learned by DFC-VAE (top row) and VAE (bottom row) between the five interested semantic directions and other semantic directions in the the (a) 3^{rd} and (b) 197^{th} stage.

tween the encoder and decoder, we explored the DFC-VAE model from a well-trained stage and an early training stage. We swapped the pairing between the two encoders and two decoders to investigate their respective responsibilities, i.e., the well-trained encoder is paired with the early-stage decoder, and the early-stage encoder is paired with the well-trained decoder. By comparing the reconstructed images from them, we discovered that a well-trained encoder is responsible for controlling semantics, while a well-trained decoder is responsible for generating clear images. For example, the image in Fig. 7-a2 reconstructs the image in Fig. 7-a0 using the early-stage encoder (E_2) and the well-trained decoder (D_1). Although the reconstruction did not catch the features of the original image (e.g., gender, hair style and color), the generated image is clear. On the contrary, the image in Fig. 7-a3 is reconstructed using the well-trained encoder (E_1) and the early-stage decoder (D_2). The image captures most of the features in the original image, but it is blurry. Similar observations can also be found in Fig. 7b, 7c, and 7d.

We believe a well-trained encoder can better control semantics because it better captures the correlation between different semantics. In other words, better semantic correlations make the semantic directions more accurate in the latent space generated from a well-trained encoder. For example, Fig. 8 compares the correlations between the "glasses" feature and other features in the 3^{rd} – 6^{th}, 8^{th}, 13^{th}, and 197^{th} training stages. It is obvious that the negative correlations of different semantics (i.e., the region in the black dashed lines) were evolving gradually over the training. Comparing the trend of negative to positive correlations between semantics (i.e., red to blue cells), we can see the negative correlations are acquired in later stages.

Comparing VAE and DFC-VAE. Although the VAE and DFC-VAE shared a similar network structure, the feature perceptual loss used in DFC-VAE dramatically improved the semantics learning. The image features generated from DFC-VAE tend to be less blurry and more recognizable than those from VAE. For example, the left and right image in the five pairs of images in Fig. 9a compare five image features generated from DFC-VAE and VAE respectively from the 3^{rd} training stage. Fig. 9b shows the same comparison but using the parameters of DFC-VAE and VAE from the 197^{th} stage. Comparing Fig. 9a and 9b vertically, i.e., across time, we can see that DFC-VAE enhanced the features in the reconstructed images more than VAE.

Comparing the semantic correlations learned from DFC-VAE and VAE in those two training stages, we found that both models captured the semantic correlations at a similar pace. For example, the top and bottom row of the 10 row-pairs in Fig. 10 show the semantic correlations from the DFC-VAE and VAE respectively, in stage 3 and 197. As highlighted by the rectangles, the evolutions of the correlations between the five semantics and other semantics are similar in both models, from the 3^{rd} stage (left) to the 197^{th} stage (right).

Combining our observations from Fig. 9 and Fig. 10, we get a better understanding on how the perceptual loss (from the pre-trained VGG19) was affecting the model, i.e., compared to DFC-VAE, VAE captured the correlations between different semantics but it still could not generate clear features. We suspect that the per-

ceptual loss contributed more to improving the decoder in better reconstructing image features.

5. Limitations and Future Work

LatentVis can be easily adapted to analyze other VAE models, as it is a model-agnostic approach and does not use any model-specific information (e.g., network architectures). The required data are the input images, the reconstructed images, and the learned latent variables at different training stages. The labels for different feature categories are also demanded to train our supervised linear model. One interesting question here is whether finer granularity labels can further improve the accuracy of the derived semantic directions. For example, the current "glasses" feature includes both "sunglasses" and "normal glasses". Differentiating them as two features may help in more accurately extracting the semantic directions. Additionally, we can also verify the existence of the class hierarchy of features in the latent space. These are interesting research directions for us to explore in the future. However, similar to the current limitation of our work, these future works also heavily depend on the availability of the labeled datasets.

Moreover, it is also possible to extend *LatentVis* to VAEs trained on other data types, e.g., texts or audios. Compared to images, those types of data may not be able to be visually interpreted. However, through different visual encodings used in existing works, we believe they can be intuitively presented as well. We plan to investigate more from the literature and spend more efforts this direction in the future.

It is worth mentioning that our current explorations in this work are heuristic and based only on one dataset, through which, we hope to shed some light on how the latent space of VAEs captured the semantics of images. More thorough experimental studies on more datasets would be needed to further validate our findings, which is another planned future work for us.

6. Conclusion

In this paper, we propose *LatentVis*, a visual analytics system to interpret and compare the semantics encoded in the latent space of image-based VAEs. The system trains a supervised linear model to bridge the machine learned latent space with the human understandable semantic space. From this bridging, we found that data semantics is usually expressed along a fixed direction in the latent space (i.e., semantic direction), and human interpreted similar/different semantics tend to have smaller/larger angles between semantic directions. Also, *LatentVis* can be used to examine and compare VAEs from three different perspectives: (1) different training stages, (2) separate trainings with randomly initialized neural network parameters, and (3) different VAEs. Several interesting points on VAEs are discovered and summarized as follows:

- Different stages of one training encode the semantic direction of the same feature in a consistent way.
- Different trainings of the same VAE model may result in the VAE encoding the semantic direction for the same feature in a different way.
- For a well-trained VAE, its encoder tends to be responsible for controlling semantics, while its decoder tends to be responsible for generating clear images.
- For the specific dataset we worked on, the perceptual loss of DFC-VAE contributes more to the training of the decoder in better reconstructing image features. Without using the perceptual loss, VAE is still able to accurately capture the semantics correlations.

These explorations and comparisons demonstrate how the latent spaces can be used to interpret and compare the corresponding VAEs. With the promising results demonstrated in the paper, we are confident in extending *LatentVis* to other latent variable models or other data types in the future.

References

[1] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.

[2] D. Erhan, C. Szegedy, A. Toshev, D. Anguelov, Scalable object detection using deep neural networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 2147–2154.

[3] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical image computing and computer-assisted intervention, Springer, 2015, pp. 234–241.

[4] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional

generative adversarial networks, arXiv preprint arXiv:1511.06434 (2015).

[5] T. Karras, T. Aila, S. Laine, J. Lehtinen, Progressive growing of gans for improved quality, stability, and variation, arXiv preprint arXiv:1710.10196 (2017).

[6] H. Sak, A. Senior, K. Rao, F. Beaufays, Fast and accurate recurrent neural network acoustic models for speech recognition, arXiv preprint arXiv:1507.06947 (2015).

[7] H. Shao, A. Kumar, P. Thomas Fletcher, The riemannian geometry of deep generative models, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 315–323.

[8] G. Arvanitidis, L. K. Hansen, S. Hauberg, Latent space oddity: on the curvature of deep generative models, arXiv preprint arXiv:1710.11379 (2017).

[9] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781 (2013).

[10] S. Liu, P.-T. Bremer, J. J. Thiagarajan, V. Srikumar, B. Wang, Y. Livnat, V. Pascucci, Visual exploration of semantic relationships in neural word embeddings, IEEE transactions on visualization and computer graphics 24 (2018) 553–562.

[11] L. Kuhnel, T. Fletcher, S. Joshi, S. Sommer, Latent space non-linear statistics, arXiv preprint arXiv:1805.07632 (2018).

[12] J. Wang, W. Zhang, H. Yang, Scanviz: Interpreting the symbol-concept association captured by deep neural networks through visual analytics, in: 2020 IEEE Pacific Visualization Symposium (PacificVis), IEEE, 2020, pp. 51–60.

[13] P. Bojanowski, A. Joulin, D. Lopez-Paz, A. Szlam, Optimizing the latent space of generative networks, arXiv preprint arXiv:1707.05776 (2017).

[14] A. A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, R. Hadsell, Meta-learning with latent embedding optimization, arXiv preprint arXiv:1807.05960 (2018).

[15] Y. Hoshen, J. Malik, Non-adversarial image synthesis with generative latent nearest neighbors, arXiv preprint arXiv:1812.08985 (2018).

[16] Q. Le, T. Mikolov, Distributed representations of sentences and documents, in: International conference on machine learning, 2014, pp. 1188–1196.

[17] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, S. Liu, Towards Better Analysis of Deep Convolutional Neural Networks, arXiv e-prints (2016) arXiv:1604.07043. arXiv:1604.07043.

[18] J. Wang, L. Gou, H. Yang, H.-W. Shen, Ganviz: A visual analytics approach to understand the adversarial game, IEEE transactions on visualization and computer graphics 24 (2018) 1905–1917.

[19] M. Kahng, P. Y. Andrews, A. Kalro, D. H. Chau, ActiVis: Visual Exploration of Industry-Scale Deep Neural Network Models, arXiv e-prints (2017) arXiv:1704.01942. arXiv:1704.01942.

[20] D. Bau, B. Zhou, A. Khosla, A. Oliva, A. Torralba, Network dissection: Quantifying interpretability of deep visual representations, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 6541–6549.

[21] M. T. Ribeiro, S. Singh, C. Guestrin, Why should i trust you?: Explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, ACM, 2016, pp. 1135–1144.

[22] Y. Ming, H. Qu, E. Bertini, Rulematrix: Visualizing and understanding classifiers with rules, IEEE transactions on visualization and computer graphics 25 (2019) 342–352.

[23] J. Zhang, Y. Wang, P. Molino, L. Li, D. S. Ebert, Manifold: A model-agnostic framework for interpretation and diagnosis of machine learning models, IEEE transactions on visualization and computer graphics 25 (2019) 364–373.

[24] J. Wang, L. Gou, W. Zhang, H. Yang, H.-W. Shen, Deepvid: Deep visual interpretation and diagnosis for image classifiers via knowledge distillation, IEEE transactions on visualization and computer graphics 25 (2019) 2168–2180.

[25] C. Zhang, S. Bengio, M. Hardt, B. Recht, O. Vinyals, Understanding deep learning requires rethinking generalization, arXiv preprint arXiv:1611.03530 (2016).

[26] E. D. Cubuk, B. Zoph, S. S. Schoenholz, Q. V. Le, Intriguing properties of adversarial examples, arXiv preprint arXiv:1711.02846 (2017).

[27] X. Hou, L. Shen, K. Sun, G. Qiu, Deep feature consistent variational autoencoder, in: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2017, pp. 1133–1141.

[28] D. P. Kingma, M. Welling, Auto-encoding variational bayes, arXiv preprint arXiv:1312.6114 (2013).

[29] J. Johnson, A. Alahi, L. Fei-Fei, Perceptual losses for real-time style transfer and super-resolution, in: European conference on computer vision, Springer, 2016, pp. 694–711.

[30] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, arXiv e-prints (2014) arXiv:1409.1556.

arXiv:1409.1556.
[31] Z. Liu, P. Luo, X. Wang, X. Tang, Deep learning face attributes in the wild, in: Proceedings of International Conference on Computer Vision (ICCV), 2015.

A. Appendix

The Data Module contains two linked visualization views. The first view presents a statistical summary of all training images. Each bubble in this view represents one feature category (the color of the bubble corresponds to different clusters in Fig. 4b), and the distances among bubbles reflect the Euclidean distances between those feature categories in the latent space. These distances are calculated via the Multi-Dimensional Scaling (MDS) algorithm, whose input is the average latent variable of images belong to the same feature category. Clicking on any bubble in this view will trigger the second view to display images from the corresponding category. The second view displays numerous randomly selected images from the selected image category, so that users can check the features of those images and select interested ones for further exploration. To save the screen space, images are scaled down to 32×32. Clicking on any image in this view will trigger further updates in other views.