Variational Hetero-Encoder Randomized Generative Adversarial Networks for Joint Image-Text Modeling

Anonymous Author(s)
Affiliation
Address
email

Abstract

For bidirectional joint image-text modeling, we develop variational hetero-encoder (VHE) randomized generative adversarial network (GAN) that integrates a probabilistic text decoder, probabilistic image encoder, and GAN into a coherent end-to-end multi-modality learning framework. VHE randomized GAN (VHE-GAN) encodes an image to decode its associated text, and feeds the variational posterior as the source of randomness into the GAN image generator. We plug three off-the-shelf modules, including a deep topic model, a ladder-structured image encoder, and StackGAN++, into VHE-GAN, which already achieves competitive performance. This further motivates the development of VHE-raster-scan-GAN that generates photo-realistic images in not only a multi-scale low-to-high-resolution manner, but also a hierarchical-semantic coarse-to-fine fashion. By capturing and relating hierarchical semantic and visual concepts with end-to-end training, VHE-raster-scan-GAN achieves state-of-the-art performance in a wide variety of image-text multi-modality learning and generation tasks. PyTorch code is provided.

1 Introduction

Images and texts commonly occur together in the real world. There exists a wide variety of deep neural network based unidirectional methods that model images (texts) given texts (images) [1–5]. There also exist probabilistic graphic model based bidirectional methods [6–8] that capture the joint distribution of images and texts. These bidirectional methods, however, often make restrictive parametric assumptions that limit their image generation ability. Exploiting recent progress on deep probabilistic models and variational inference [9–13], we propose an end-to-end learning framework to construct multi-modality deep generative models that can not only generate vivid image-text pairs, but also achieve state-of-the-art results on various unidirectional tasks [1, 5–8, 13–16], such as generating photo-realistic images given texts and performing text-based zero-shot learning.

To extract and relate semantic and visual concepts, we first introduce variational hetero-encoder (VHE) that encodes an image to decode its textual description ($e.g.$, tags, sentences, binary attributes, and long documents), where the probabilistic encoder and decoder are jointly optimized using variational inference [9, 17–20]. The latent representation of VHE can be sampled from either the variational posterior provided by the image encoder given an image input, or the posterior of the text decoder via MCMC given a text input. VHE by construction has the ability to generate texts given images. To further enhance its text generation performance and allow synthesizing photo-realistic images given an image, text, or random noise, we feed the variational posterior of VHE in lieu of random noise as the source of randomness into the image generator of a generative adversarial network (GAN) [12]. We refer to this new modeling framework as VHE randomized GAN (VHE-GAN).

Off-the-shelf text decoders, image encoders, and GANs can be directly plugged into the VHE-GAN framework for end-to-end multi-modality learning. To begin with, as shown in Figs. 1(a) and 1(b), we

Submitted to 33rd Conference on Neural Information Processing Systems (NeurIPS 2019). Do not distribute.

construct VHE-StackGAN++ by using the Poisson gamma belief network (PGBN) [10] as the VHE text decoder, using the Weibull upward-downward variational encoder [11] as the VHE image encoder, and feeding the concatenation of the multi-stochastic-layer latent representation of the VHE as the source of randomness into the image generator of StackGAN++ [13]. While VHE-StackGAN++ already achieves very attractive performance, we find that its performance can be clearly boosted by better exploiting the multi-stochastic-layer semantically meaningful hierarchical latent structure of the PGBN text decoder. To this end, as shown in Figs. 1(a) and 1(c), we develop VHE-raster-scan-GAN to perform image generation in not only a multi-scale low-to-high-resolution manner in each layer, as done by StackGAN++, but also a hierarchical-semantic coarse-to-fine fashion across layers, a unique feature distinguishing it from existing methods. Consequently, not only can VHE-raster-scan-GAN generate vivid high-resolution images with better details, but also build interpretable hierarchical semantic-visual relationships between the generated images and texts.

Our main contributions include: 1) VHE-GAN that provides a plug-and-play framework to integrate off-the-shelf probabilistic decoders, variational encoders, and GANs for end-to-end bidirectional multi-modality learning, and 2) VHE-raster-scan-GAN that captures and relates hierarchical semantic and visual concepts to achieve state-of-the-art results in various joint image-text modeling tasks.

2 Variational hetero-encoder randomized generative adversarial networks

VAEs and GANs are two distinct types of deep generative models. Consisting of a generator (decoder) $p(\boldsymbol{x} \mid \boldsymbol{z})$, a prior $p(\boldsymbol{z})$, and an inference network (encoder) $q(\boldsymbol{z} \mid \boldsymbol{x})$ that is used to approximate the posterior $p(\boldsymbol{z} \mid \boldsymbol{x})$, VAEs [9, 20] are optimized by maximizing the evidence lower bound (ELBO) as

$$\text{ELBO} = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\mathcal{L}(\boldsymbol{x})], \quad \mathcal{L}(\boldsymbol{x}) := \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}\mid\boldsymbol{x})}\left[\ln p(\boldsymbol{x}\mid\boldsymbol{z})\right] - \text{KL}\left[q(\boldsymbol{z}\mid\boldsymbol{x})||p(\boldsymbol{z})\right], \quad (1)$$

where $p_{\text{data}}(\boldsymbol{x}) = \sum_{i=1}^{N} \frac{1}{N}\delta_{\boldsymbol{x}_i}$ represents the empirical distribution of N data points. Distinct from VAEs that make parametric assumptions on the data distribution and perform posterior inference, GANs in general use implicit data distribution and do not come with meaningful latent representations [12]; they learn both a generator G and a discriminator D by optimizing a mini-max objective as

$$\min_G \max_D \{\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}\left[\ln D(\boldsymbol{x})\right] + \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}\left[\ln(1 - D(G(\boldsymbol{z})))\right]\}, \quad (2)$$

where $p(\boldsymbol{z})$ is a random noise distribution that acts as the source of randomness for data generation.

2.1 VHE-GAN objective function for end-to-end multi-modality learning

Below we show how to construct VHE-GAN to jointly model images $\boldsymbol{x}$ and their associated texts $\boldsymbol{t}$, capturing and relating hierarchical semantic and visual concepts. First, we modify the usual VAE into VHE, optimizing a lower bound of the text log-marginal-likelihood $\mathbb{E}_{\boldsymbol{t}\sim p_{\text{data}}(\boldsymbol{t})}[\ln p(\boldsymbol{t})]$ as

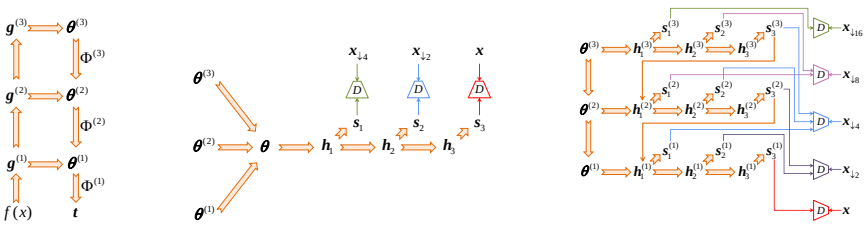
$$\text{ELBO}_{\text{vhe}} = \mathbb{E}_{p_{\text{data}}(\boldsymbol{t},\boldsymbol{x})}[\mathcal{L}_{\text{vhe}}(\boldsymbol{t},\boldsymbol{x})], \;\; \mathcal{L}_{\text{vhe}}(\boldsymbol{t},\boldsymbol{x}) := \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}\mid\boldsymbol{x})}\left[\ln p(\boldsymbol{t}\mid\boldsymbol{z})\right] - \text{KL}\left[q(\boldsymbol{z}\mid\boldsymbol{x})||p(\boldsymbol{z})\right], \quad (3)$$

where $p(\boldsymbol{t} \mid \boldsymbol{z})$ is the text decoder, $p(\boldsymbol{z})$ is the prior, $p(\boldsymbol{t}) = \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}[p(\boldsymbol{t} \mid \boldsymbol{z})]$, and $\mathcal{L}_{\text{vhe}}(\boldsymbol{t},\boldsymbol{x}) \le \ln \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}\mid\boldsymbol{x})}[\frac{p(\boldsymbol{t}\mid\boldsymbol{z})p(\boldsymbol{z})}{q(\boldsymbol{z}\mid\boldsymbol{x})}] = \ln p(\boldsymbol{t})$. Second, the image encoder $q(\boldsymbol{z} \mid \boldsymbol{x})$, which encodes image $\boldsymbol{x}$ into its latent representation $\boldsymbol{z}$, is used to approximate the posterior $p(\boldsymbol{z} \mid \boldsymbol{t}) = p(\boldsymbol{t} \mid \boldsymbol{z})p(\boldsymbol{z})/p(\boldsymbol{t})$. Third, variational posterior $q(\boldsymbol{z} \mid \boldsymbol{x})$ in lieu of random noise $p(\boldsymbol{z})$ is fed as the source of randomness into the GAN image generator. Combing these three steps, with the parameters of the image encoder $q(\boldsymbol{z} \mid \boldsymbol{x})$, text decoder $p(\boldsymbol{t} \mid \boldsymbol{z})$, and GAN generator denoted by E, G_{vae}, and G_{gan}, respectively, we express the objective function of VHE-GAN for joint image-text end-to-end learning as

$$\min_{E, G_{\text{vae}}, G_{\text{gan}}} \max_D \mathbb{E}_{p_{\text{data}}(\boldsymbol{t},\boldsymbol{x})}[\mathcal{L}(\boldsymbol{t},\boldsymbol{x})],$$

$$\mathcal{L}(\boldsymbol{t},\boldsymbol{x}) := \ln D(\boldsymbol{x}) + \text{KL}\left[q(\boldsymbol{z}\mid\boldsymbol{x})||p(\boldsymbol{z})\right] + \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}\mid\boldsymbol{x})}\big[\ln(1 - D(G_{\text{gan}}(\boldsymbol{z}))) - \ln p(\boldsymbol{t}\mid\boldsymbol{z})\big]. \quad (4)$$

Note the objective function in (4) implies a data-triple-reuse training strategy, which uses the same data mini-batch in each stochastic gradient update iteration to jointly train the VHE, GAN discriminator, and GAN generator; see a related objective function, shown in (8) of Appendix A, that is resulted from naively combining the VHE and GAN training objectives. In VHE-GAN, the optimization of the encoder parameter E is related to not only the VHE's ELBO, but also the GAN mini-max objective function, forcing the variational posterior $q(\boldsymbol{z} \mid \boldsymbol{x})$ to serve as a bridge between VHE and GAN, allowing them to help each other. This describes the basic idea of using VHE-GAN for modeling two different modalities. In Appendix A, we analyze the properties of the VHE-GAN objective function and discuss related work. In the following, we develop two different VHE-GANs.



(a) (b) (c)

Figure 1: Illustration of (a) VHE, (b) StackGAN++, and (c) raster-scan-GAN. VHE-StackGAN++ consists of (a) and (b). VHE-raster-scan-GAN consists of (a) and (c). $\boldsymbol{x}_{\downarrow d}$ is down-sampled from $\boldsymbol{x}$ with scaling factor d.

2.2 VHE-StackGAN++ with off-the-shelf modules

As shown in Figs. 1(a) and 1(b), we first construct VHE-StackGAN++ by plugging three off-the-shelf modules, including a deep topic model [10], a ladder-structured encoder [11], and StackGAN++ [13], into VHE-GAN. For text analysis, both sequence models and topic models are widely used. Sequence models [21] often represent each document as a sequence of word embedding vectors, capturing local dependency structures with some type of recurrent neural networks (RNNs), such as long short-term memory (LSTM) [22]. Topic models [23] often represent each document as a bag of words (BOW), capturing global word cooccurrence patterns into latent topics. Suitable for capturing local dependency structure, existing sequence models often have difficulty in capturing long-term word dependencies and hence macro-level information, such as global word cooccurrence patterns ($i.e.$, topics), especially for long documents. By contrast, while topic models ignore the word order information, they are very effective in capturing latent topics, which are often directly related to macro-level visual information [1, 24, 25]. Moreover, topic models can be applied to not only sequential texts, such as few sentences [26, 27] and long documents [10], but also non-sequential ones, such as textual tags [7, 8, 28] and binary attributes [29, 30]. For this reason, for the VHE text decoder, we choose PGBN [10], which is a state-of-the-art topic model that can also be represented as a multi-stochastic-layer deep generalization of latent Dirichlet allocation (LDA) [31]. We complete VHE-StackGAN++ by choosing the Weibull upward-downward variational encoder [11] as the VHE image encoder, and feeding the concatenation of all the hidden layers of PGBN as the source of randomness to the image generator of StackGAN++ [13].

As shown in Fig. 1, we use a VHE that encodes an image into a deterministic-upward–stochastic-downward ladder-structured latent representation, which is used to decode the corresponding text. More specifically, we represent each text document as a BOW high-dimensional sparse count vector $\boldsymbol{t}_n \in \mathbb{Z}^{K_0}$, where $\mathbb{Z} = \{0, 1, \cdots\}$ and K_0 is the vocabulary size. For the VHE text decoder, we choose to use PGBN to extract hierarchical latent representation from $\boldsymbol{t}_n$. PGBN consists of multiple gamma distributed stochastic hidden layers, generalizing the "shallow" Poisson factor analysis [32,33] into a deep setting. PGBN with L hidden layers, from top to bottom, is expressed as

$$\begin{aligned}
&\boldsymbol{\theta}_n^{(L)} \sim \text{Gam}\left(\boldsymbol{r}, 1/s_n^{(L+1)}\right),\ \boldsymbol{r} \sim \text{Gam}(\gamma_0/K_L, 1/s_0),\\
&\boldsymbol{\theta}_n^{(l)} \sim \text{Gam}\left(\boldsymbol{\Phi}^{(l+1)}\boldsymbol{\theta}_n^{(l+1)}, 1/s_n^{(l+1)}\right), l = L-1, \cdots, 2, 1, \quad \boldsymbol{t}_n \sim \text{Pois}\left(\boldsymbol{\Phi}^{(1)}\boldsymbol{\theta}_n^{(1)}\right),
\end{aligned} \tag{5}$$

where the hidden units $\boldsymbol{\theta}_n^{(l)} \in \mathbb{R}_+^{K_l}$ of layer l are factorized under the gamma likelihood into the product of the topics $\boldsymbol{\Phi}^{(l)} \in \mathbb{R}_+^{K_{l-1} \times K_l}$ and hidden units of the next layer, $\mathbb{R}_+ = \{x, x \geq 0\}$, $s_n^{(l)} > 0$, and K_l is the number of topics of layer l. If the texts are represented as binary attribute vectors $\boldsymbol{b}_n$, we can add a Bernoulli-Poisson link layer as $\boldsymbol{b}_n = \mathbf{1}(\boldsymbol{t}_n \geq 1)$ [10, 34]. We place a Dirichlet prior on each column of $\boldsymbol{\Phi}^{(l)}$. The topics can be organized into a directed acyclic graph (DAG), whose node k at layer l can be visualized with the top words of $\left[\prod_{t=1}^{l-1} \boldsymbol{\Phi}^{(t)}\right]\boldsymbol{\phi}_k^{(l)}$; the topics tend to be very general in the top layer and become increasingly more specific when moving downwards. This semantically meaningful latent hierarchy provides unique opportunities to build a better image generator by coupling the semantic hierarchical structures with visual ones.

Let us denote $\boldsymbol{\Phi} = \{\boldsymbol{\Phi}^{(1)}, \ldots, \boldsymbol{\Phi}^{(L)}, \boldsymbol{r}\}$ as the set of global parameters of PGBN shown in (5). Given $\boldsymbol{\Phi}$, we adopt the inference in Zhang et al. [11] to build an Weibull upward-downward variational image encoder as $\prod_{n=1}^{N}\prod_{l=1}^{L} q(\boldsymbol{\theta}_n^{(l)} \mid \boldsymbol{x}_n, \boldsymbol{\Phi}^{(l+1)}, \boldsymbol{\theta}_n^{(l+1)})$, where $\boldsymbol{\Phi}^{(L+1)} := \boldsymbol{r}$, $\boldsymbol{\theta}_n^{(L+1)} := \emptyset$, and

$$q(\boldsymbol{\theta}_n^{(l)} \mid \boldsymbol{x}_n, \boldsymbol{\Phi}^{(l+1)}, \boldsymbol{\theta}_n^{(l+1)}) = \text{Weibull}(\boldsymbol{k}_n^{(l)} + \boldsymbol{\Phi}^{(l+1)}\boldsymbol{\theta}_n^{(l+1)}, \boldsymbol{\lambda}_n^{(l)}). \tag{6}$$

The Weibull distribution is used to approximate the gamma distributed conditional posterior, and its parameters $\boldsymbol{k}_n^{(l)} \in \mathbb{R}^{K_l}$ and $\boldsymbol{\lambda}_n^{(l)} \in \mathbb{R}^{K_l}$ are both deterministically transformed from the convolutional neural network (CNN) image features $f(\boldsymbol{x}_n)$ [35], as illustrated in Fig. 1(a) and detailedly described in Appendix D.1. We denote $\boldsymbol{\Omega}$ as the set of encoder parameters. We refer to Zhang et al. [11] for more details about this deterministic-upward–stochastic-downward ladder-structured inference network, which is distinct from a usual VAE inference network that has a pure bottom-up structure and only interacts with the generative model via the ELBO [9, 36].

The multi-stochastic-layer latent representation $\boldsymbol{z} = \{\boldsymbol{\theta}^{(l)}\}_{l=1}^{L}$ is the bridge between two modalities. As shown in Fig. 1(b), VHE-StackGAN++ simply randomizes the image generator of StackGAN++ [13] with the concatenated vector $\boldsymbol{\theta} = \left[\boldsymbol{\theta}^{(1)}, \cdots, \boldsymbol{\theta}^{(L)}\right]$. We provide the overall objective function in (13) of Appendix D.2. We also note that bidirectional transforms between image $\boldsymbol{x}$ and text $\boldsymbol{t}$ require $\boldsymbol{z}$ to be inferred regardless of whether $\boldsymbol{x}$ or $\boldsymbol{t}$ is given. This is straightforward for the proposed model, as $\boldsymbol{z}$ can be either drawn from the image encoder $q(\boldsymbol{z} \mid \boldsymbol{x})$ in (6), or drawn with an upward-downward Gibbs sampler [10] from the conditional posteriors of the PBGN text decoder $p(\boldsymbol{t} \mid \boldsymbol{z})$ in (5). By contrast, many existing models can perform only unidirectional transforms [1, 5, 13–16].

2.3 VHE-raster-scan-GAN with a hierarchical-semantic multi-resolution image generator

While we find that VHE-StackGAN++ has already achieved impressive results, its simple concatenation of $\boldsymbol{\theta}^{(l)}$ does not fully exploit the semantically-meaningful hierarchical latent representation of the PGBN-based text decoder. For three DAG subnets inferred from three different datasets, as shown in Figs. 20 -22 of Appendix C.7, the higher-layer PGBN topics match general visual concepts, such as those on shapes, colors, and backgrounds, while the lower-layer ones provide finer details. This motivates us to develop an image generator to exploit the semantic structure, which matches coarse-to-fine visual concepts, to gradually refine its generation. To this end, as shown in Fig. 1(c), we develop "raster-scan" GAN that performs generation not only in a multi-scale low-to-high-resolution manner in each layer, but also a hierarchical-semantic coarse-to-fine fashion across layers.

Suppose we are building a three-layer raster-scan GAN to generate an image of size 256^2. We randomly select an image $\boldsymbol{x}_n$ and then sample $\{\boldsymbol{\theta}_n^{(l)}\}_{l=1}^{3}$ from the variational posterior $\prod_{l=1}^{3} q(\boldsymbol{\theta}_n^{(l)} \mid \boldsymbol{x}_n, \boldsymbol{\Phi}^{(l+1)}, \boldsymbol{\theta}_n^{(l+1)})$. First, the top-layer latent variable $\boldsymbol{\theta}^{(3)}$, often capturing general semantic information, is transformed to hidden features $h_i^{(3)}$ for the i^{th} branch: $h_1^{(3)} = F_1^{(3)}(\boldsymbol{\theta}^{(3)})$ and $h_i^{(3)} = F_i^{(3)}(h_{i-1}^{(3)}, \boldsymbol{\theta}^{(3)})$ for $i = 2, 3$, where $F_i^{(l)}$ is a CNN. Second, having obtained $\{h_i^{(3)}\}_{i=1}^{3}$, generators $\{G_i^{(3)}\}_{i=1}^{3}$ synthesize low-to-high-resolution image samples $\{\boldsymbol{s}_i^{(3)} = G_i^{(3)}(h_i^{(3)})\}_{i=1}^{3}$, where $\boldsymbol{s}_1^{(3)}$, $\boldsymbol{s}_2^{(3)}$, and $\boldsymbol{s}_3^{(3)}$ are of 16^2, 32^2, and 64^2, respectively. Third, $\boldsymbol{s}_3^{(3)}$ is down-sampled to $\hat{\boldsymbol{s}}_3^{(3)}$ of size 32^2 and combined with the information from $\boldsymbol{\theta}^{(2)}$ to provide the hidden features at layer two: $h_1^{(2)} = C(F_1^{(2)}(\boldsymbol{\theta}^{(2)}), \hat{\boldsymbol{s}}_3^{(3)})$ and $h_i^{(2)} = F_i^{(2)}(h_{i-1}^{(2)}, \boldsymbol{\theta}^{(2)})$ for $i = 2, 3$, where C denotes concatenation along the channel. Fourth, the generators synthesize image samples $\{\boldsymbol{s}_i^{(2)} = G_i^{(2)}(h_i^{(2)})\}_{i=1}^{3}$, where $\boldsymbol{s}_1^{(2)}$, $\boldsymbol{s}_2^{(2)}$, and $\boldsymbol{s}_3^{(2)}$ are of 32^2, 64^2, and 128^2, respectively. The same process is then replicated at layer one to generate $\{\boldsymbol{s}_i^{(1)} = G_i^{(1)}(h_i^{(1)})\}_{i=1}^{3}$, where $\boldsymbol{s}_1^{(1)}$, $\boldsymbol{s}_2^{(1)}$, and $\boldsymbol{s}_3^{(1)}$ are of size 64^2, 128^2, and 256^2, respectively, and $\boldsymbol{s}_3^{(1)}$ becomes a desired high-resolution synthesized image with fine details. The detailed structure of raster-scan-GAN is described in Fig. 25 of Appendix D.3. PyTorch code is included in the Supplementary Material to aid the understanding and help reproduce the results.

Different from many existing methods [1, 3, 13, 14] whose textual feature extraction is separated from the end task, VHE-raster-scan-GAN performs joint optimization. As detailedly described in the Algorithm in Appendix E, at each mini-batch based iteration, after updating $\boldsymbol{\Phi}$ by the topic-layer-adaptive stochastic gradient Riemannian (TLASGR) MCMC of [31], a Weibull distribution based reparameterization gradient [11] is used to end-to-end optimize the following objective:

$$\begin{aligned}\min_{\{G_i^{(l)}\}_{i,l},\, \boldsymbol{\Omega}} \max_{\{D_i^{(l)}\}_{i,l}} \mathbb{E}_{p_{\text{data}}(\boldsymbol{x}_n, \boldsymbol{t}_n)} \mathbb{E}_{\prod_{l=1}^{3} q(\boldsymbol{\theta}_n^{(l)} \mid \boldsymbol{x}_n, \boldsymbol{\Phi}^{(l+1)}, \boldsymbol{\theta}_n^{(l+1)})} \big\{ &- \log p(\boldsymbol{t}_n \mid \boldsymbol{\Phi}^{(1)}, \boldsymbol{\theta}_n^{(1)}) \\ &+ \textstyle\sum_{l=1}^{3} \mathrm{KL}[q(\boldsymbol{\theta}_n^{(l)} \mid \boldsymbol{x}_n, \boldsymbol{\Phi}^{(1+1)}, \boldsymbol{\theta}_n^{(l+1)}) \,||\, p(\boldsymbol{\theta}_n^{(l)} \mid \boldsymbol{\Phi}^{(1+1)}, \boldsymbol{\theta}_n^{(l+1)})] \\ &+ \textstyle\sum_{l=1}^{3} \sum_{i=1}^{3} [\log D_i^{(l)}(\boldsymbol{x}_{n,i}^{(l)}, \boldsymbol{\theta}_n^{(l)}) + \log(1 - D_i^{(l)}(G_i^{(l)}(\boldsymbol{\theta}_n^{(l)}), \boldsymbol{\theta}_n^{(l)}))] \big\}, \end{aligned} \tag{7}$$

where $\{\boldsymbol{x}_{n,i}^{(l)}\}_{i=1,l=1}^{3,3}$ denote different resolutions of $\boldsymbol{x}_n$ corresponding to $\{\boldsymbol{s}_{n,i}^{(l)}\}_{i=1,l=1}^{3,3}$.

Table 1: Inception score (IS) and Frechet inception distance (FID) of StackGAN++ [13], HDGAN [16], AttGAN [14], and the proposed models; the values labeled with * are calculated by the provided well-trained models and the others are quoted from the original publications; see Tab. 3 in Appendix C.1 for error bars of IS.

Method	StackGAN++		HDGAN		AttnGAN		PGBN+StackGAN++		VHE-StackGAN++		VHE-raster-scan-GAN	
Criterion	IS	FID	IS	FID	IS	FID	IS	FID	IS	FID	IS	FID
Flower	3.26	48.68	3.45	40.12*	–	–	3.29	41.04	3.56	38.66	**3.72**	**35.13**
CUB	3.84	15.30	4.15	13.48*	4.36	13.02*	3.92	13.79	4.20	12.93	**4.41**	**12.02**
COCO	8.30	81.59	11.86	78.16*	25.89	77.01*	10.63	79.65	12.63	78.02	**27.16**	**75.88**

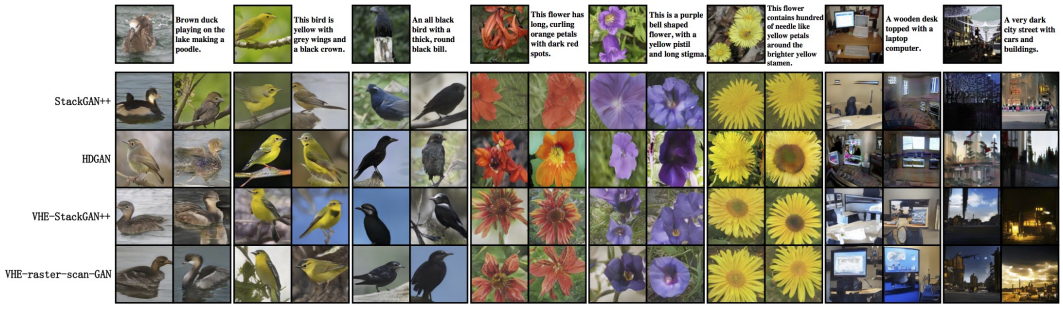


Figure 2: Comparison on image generation given texts from CUB, Flower, and COCO. Shown in the top row are the textual descriptions and their associated real images; see Appendix C.2 for higher-resolution images.

2.4 Related work on joint image-text learning

Gomez et al. [1] develop a CNN to learn a transformation from images to textual features pre-extracted by LDA [23]. Outstanding in image generation, GANs have been exploited to model images given pre-learned textual features extracted by RNNs [3, 5, 14, 16, 37]. All these work need a pre-trained linguistic model based on large-scale extra text data and the transformations between the images and texts are only unidirectional. On the other hand, probabilistic graphical model based methods [6–8] are proposed to learn a joint latent space for images and texts to realize bidirectional transformations, but their image generators are often limited to generating low-level image features. By contrast, VHE-raster-scan-GAN performs bidirectional end-to-end learning to capture and relate hierarchical visual and semantic concepts across multiple stochastic layers, capable of a wide variety of joint image-text learning and generation tasks, as described below.

3 Experimental results

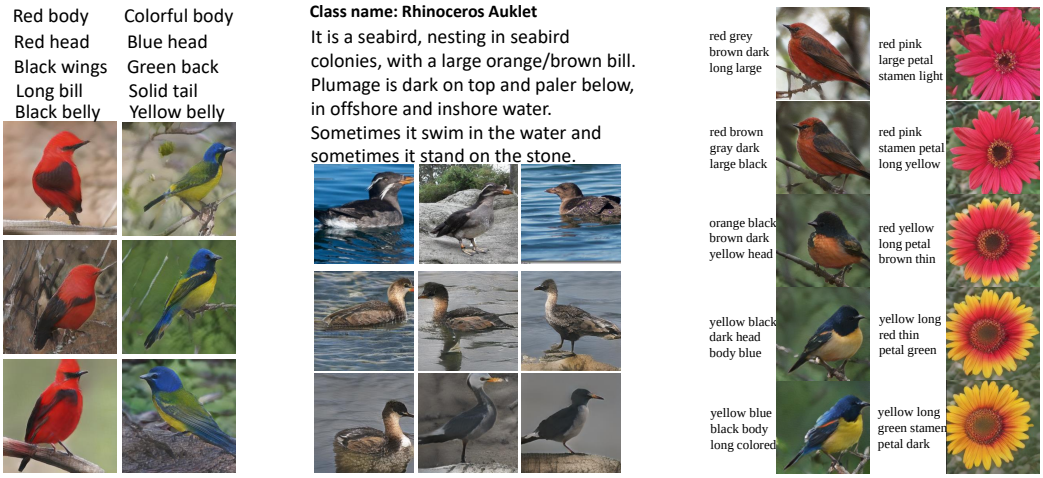
For joint image-text multimodal learning, following previous work, we evaluate the proposed VHE-StackGAN++ and VHE-raster-scan-GAN on three datasets: CUB [38], Flower [39], and COCO [40], as described in Appendix F. Besides the usual text-to-image generation task, due to the distinct bidirectional inference capability of the proposed models, we can perform a rich set of additional tasks such as image-to-text, image-to-image, and noise-to-image-text-pair generations. Due to space constraint, we present below some representative results, and defer additional ones to the Appendix. We provide the details of our experimental settings in Appendix F.

3.1 Text-to-image learning

Although the proposed VHE-GANs do not have a text encoder to directly project a document to the shared latent space, given a document and a set of topics inferred during training, we use the upward-downward Gibbs sampler of Zhou et al. [10] to draw $\{\boldsymbol{\theta}^{(l)}\}_{l=1}^{L}$ from its conditional posterior under PGBN, which are then fed into the GAN image generator to synthesize random images.

Text-to-image generation: In Tab. 1, with inception score (IS) [41] and Frechet inception distance (FID) [42], we compare our models with three state-of-the-art GANs in text-to-image generation. For visualization, we show in the top row of Fig. 2 different test textual descriptions and the real images associated with them, and in the other rows random images generated conditioning on these textual descriptions by different algorithms. Higher-resolution images are shown in Appendix C.2. We also provide example results on COCO, a much more challenging dataset, in Fig. 12 of Appendix C.3.

It is clear from Fig. 2 that although both StackGAN++ [13] and HDGAN [16] generate photo-realistic images nicely matched to the given texts, they often misrepresent or ignore some key textual information, such as "black crown" for the 2nd test text, "yellow pistil" for the 5th text, "yellow



(a) (b) (c)

Figure 3: Example results of VHE-raster-scan-GAN on three different tasks: (a) image generation given five textual attributes; (b) image generation given a long class-specific document (showing three representative sentences for brevity) from CUB; and (c) latent space interpolation for joint image-text generation on CUB (left column) and Flower (right column), where the texts in the first and last row are given.

stamen" for the 6th text, and "computer" for the 7th text. By contrast, both the proposed VHE-StackGAN++ and VHE-raster-scan-GAN do a better job in capturing and faithfully representing these key textual information into their generated images. Fig. 12 for COCO further shows the advantages of VHE-raster-scan-GAN in better representing the given textual information in its generated images.

Note VHE-StackGAN++ has the same structured image generator as both StackGAN++ and HDGAN do. We attribute its performance gain to 1) its PGBN deep topic model helps better capture key semantic information from the textual descriptions; and 2) it performs end-to-end joint image-text learning via the VHE-GAN framework, rather than separating the extraction of textual features from text-to-image generation. Furthermore, VHE-raster-scan-GAN outperforms VHE-StackGAN++ by better utilizing the hierarchically structured text latent representation for image generation.

We also consider an ablation study for text-to-image generation, where we modify the original StackGAN++ [13], using the text features extracted by PGBN to replace the original ones by RNN, referred to as PGBN+StackGAN++. It is clear from Tab. 1 that PGBN+StackGAN++ outperforms the original StackGAN++, but underperforms VHE-StackGAN++, which can be explained by that 1) the PGBN deep topic model is more effective in extracting macro-level textual content, such as key words, than RNNs; and 2) jointly training the textual feature extractor and image encoder, discriminator, and generator in an end-to-end manner helps better capture and relate the visual and semantical concepts. Below we focus on illustrating the outstanding performance of VHE-raster-scan-GAN.

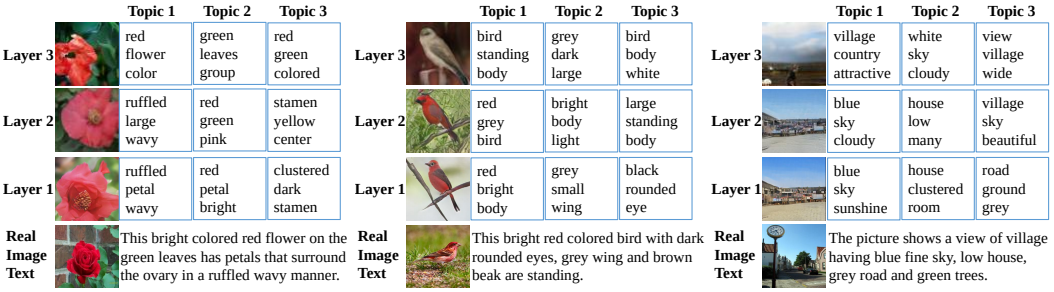
As discussed in Section 2.2, compared with sequence models, topic models can be applied to more diverse textual descriptions, including textual attributes and long documents. For illustration, we show in Figs. 3(a) and 3(b) example images generated conditioning on a set of textual attributes and an encyclopedia document, respectively. These synthesized images are photo-realistic and their visual contents well match the semantics of the given texts. See Appendix B for more illustrations.

Latent space interpolation: In order to understand the jointly learned image and text manifolds, given texts $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$, we draw $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ and use the interpolated variables between them to generate both images via the GAN's image generator and texts via the PGBN text decoder. As in Fig. 3(c), the first row shows the true texts $\boldsymbol{t}_1$ and images generated with $\boldsymbol{\theta}_1$, the last row shows $\boldsymbol{t}_2$ and images generated with $\boldsymbol{\theta}_2$, and the second to fourth rows show the generated texts and images with the interpolations from $\boldsymbol{\theta}_1$ to $\boldsymbol{\theta}_2$. The strong correspondences between the generated images and texts, with smooth changes in colors, object positions, and backgrounds between adjacent rows, suggest that the latent space of VHE-raster-scan-GAN is both visually and semantically meaningful. Additional more fine-gridded latent space interpolation results are shown in Figs. 14-17 of Appendix C.4.

Visualization of captured semantic and visual concepts: Zhou et al. [10] shows that the semantic concepts extracted by PGBN and their hierarchical relationships can be represented as a DAG,

(a) (b)

Figure 4: Example results of using VHE-raster-scan-GAN for (a) image-to-textual-tags generation, where the generated tags highlighted in red are included in the original ones; (b) image-text-pair generations (columns from left to right are based on Flower, CUB, and COCO, respectively).



(a) (b) (c)

Figure 5: Visualization of example semantic and visual concepts captured by a three-stochastic-hidden-layer VHE-raster-scan-GAN from (a) Flower, (b) Bird, and (c) COCO. In each subplot, given the real text $\boldsymbol{t}_n$ shown at the bottom, we draw $\{\boldsymbol{\theta}_n^{(l)}\}_{l=1}^3$ via Gibbs sampling; we show the three most active topics in $\boldsymbol{\Phi}^{(l)}$ (ranked by the weights of $\boldsymbol{\theta}_n^{(l)}$) at layer $l = 3, 2, 1$, where each topic is visualized by its top three words; and we feed $\{\boldsymbol{\theta}_n^{(l)}\}_{l=1}^3$ into raster-scan-GAN to generate three random images (one per layer, coarse to fine from layers 3 to 1).

only a subnet of which will be activated given a specific text input. In each subplot of Fig. 5, we visualize example topic nodes of the DAG subnet activated by the given text input, and show the corresponding images generated at different hidden layers. There is a good match at each layer between the visual contents of the generated images and semantics of the top activated topics, which are mainly about general shapes, colors, or backgrounds at the top layer, and become more and more fine-grained when moving downward. In Fig. 6, for the DAG learned on COCO, we show a representative subnet that is rooted at a top-layer node about "rooms and objects at home," and provide both semantic and visual representations for each node. Being able to capturing and relating hierarchical semantic and visual concepts helps explain the state-of-the-art performance of VHE-raster-scan-GAN.

Figure 6: An example topic hierarchy learned on COCO and its visual representation. We sample $\boldsymbol{\theta}_n^{(1:3)} \sim q(\boldsymbol{\theta}_n^{(1:3)} \mid \boldsymbol{\Phi}, \boldsymbol{x}_n)$ for all n; for topic node k of layer l, we show both its top words and the top two images ranked by their activations $\theta_{nk}^{(l)}$.

3.2 Image-to-text learning

VHE-raster-scan-GAN can perform a wide variety of extra tasks, such as image-to-text generation, text-based zero-shot learning (ZSL), and image retrieval given a text query. In particular, given image $\boldsymbol{x}_n$, we draw $\hat{\boldsymbol{t}}_n$ as $\hat{\boldsymbol{t}}_n \mid \boldsymbol{\theta}_n \sim p(\boldsymbol{t} \mid \boldsymbol{\Phi}, \boldsymbol{\theta}_n)$, $\boldsymbol{\theta}_n \mid \boldsymbol{x}_n \sim q_{\boldsymbol{\Omega}}(\boldsymbol{\theta} \mid \boldsymbol{\Phi}, \boldsymbol{x}_n)$ and use it for downstream tasks.

Image-to-text generation: Given an image, we may generate some key words, as shown in Fig. 4(a), where the true and generated ones are displayed on the left and right of the input image, respectively. It is clear that VHE-raster-scan-GAN successfully captures the object colors, shapes, locations, and backgrounds to predict relevant key words.

Text-based ZSL: Text-based ZSL is a specific task that learns a relationship between images and texts on the seen classes and transfer it to the unseen ones [43]. We follow the the same settings on CUB and Flower as existing text-based ZSL methods summarized in Tab. 2. There are two default splits for CUB—the hard (CUB-H) and easy one (CUB-E)—and one split setting for Flower, as described in Appendix F. Note that except for our models that infer a shared semantically meaningful latent space between two modalities, none of the other methods have generative models for both modalities, regardless of whether they learn a classifier or a distance metric in a latent space for ZSL.

Tab. 2 shows that VHE-raster-scan-GAN clearly outperforms the state-of-the-art in terms of the Top-1 accuracy on both the CUB-H and Flower, and is comparable to the second best on CUB-E (it is the best among all methods that have reported their Top-5 accuracies on CUB-E). Note for CUB-E, every unseen class has some corresponding seen classes under the same super-category, which makes the classification of surface or distance metric learned on the seen classes easier to generalize to the unseen ones. We also note that both GAZSL and ZSLPP rely on visual part detection to extract image features, making their performance sensitive to the quality of the visual part detector that often has to be elaborately tuned for different classes and hence limiting their generalization ability, for example, the visual part detector for birds is not suitable for flowers. Tab. 2 also includes the results of ZSL using VHE, which show that given the same structured text decoder and image encoder, VHE consistently underperforms both VHE-StackGAN++ and VHE-raster-scan-GAN. This suggests 1) the advantage of a joint generation of two modalities, and 2) the ability of GAN in helping VHE achieve better data representation. The results in Tab. 2 also show that the ZSL performance of VHE-raster-scan-GAN has a clear trend of improvement as PGBN becomes deeper, suggesting the advantage of having a multi-stochastic-hidden-layer deep topic model for text generation.

Table 2: Accuracy (%) of ZSL on CUB and Flower. Note that some of them are attribute-based methods but applicable in our setting by replacing attribute vectors with text features (labeled by *), as discussed in [29]; see Tab. 4 in Appendix C.1 for error bars.

Text-ZSL dataset	CUB-H	CUB-E		Flower
Accuracy criterion	top-1	top-1	top-5	top-1
WAC-Kernel [44]	7.7	33.5	–	9.1
ZSLNS [45]	7.3	29.1	61.8	8.7
ESZSL* [46]	7.4	28.5	59.9	8.6
SynC* [47]	8.6	28.0	61.3	8.2
ZSLPP [29]	9.7	37.2	–	–
GAZSL [30]	10.3	**43.7**	67.61	–
VHE-L3	14.0	34.6	64.6	8.9
VHE-StackGAN++-L3	16.1	38.5	68.2	10.6
VHE-raster-scan-GAN-L1	11.7	32.1	62.6	9.4
VHE-raster-scan-GAN-L2	14.9	37.1	64.6	11.0
VHE-raster-scan-GAN-L3	**16.7**	39.6	**70.3**	**12.1**

3.3 Generation of random text-image pairs

Below we show how to generate data samples that contain both modalities. After training a three-stochastic-hidden-layer VHE-raster-scan-GAN, following the data generation process of the PGBN text decoder, given $\{\mathbf{\Phi}^{(l)}\}_{l=1}^{3}$ and $\boldsymbol{r}$, we first generate $\boldsymbol{\theta}^{(3)} \sim \text{Gam}\left(\boldsymbol{r}, 1/s^{(4)}\right)$ and then downward propagate it through the PGBN as in (5) to calculate the Poisson rates for all words using $\mathbf{\Phi}^{(1)}\boldsymbol{\theta}^{(1)}$. Given a random draw, $\{\boldsymbol{\theta}^{(l)}\}_{l=1}^{3}$ is fed into the raster-scan-GAN image generator to generate a corresponding image. Shown in Fig. 4(b) are six random draws, for each of which we show its top seven words and generated image, whose relationships are clearly interpretable, suggesting that VHE-raster-scan-GAN is able to recode the key information of both modalities and the relationships between them. In addition to the tasks shown above, VHE-raster-scan-GAN can also be used to perform image retrieval given a text query, and image regeneration; see Appendices C.5 and C.6 for example results on these additional tasks.

4 Conclusion

We develop variational hetero-encoder randomized generative adversarial network (VHE-GAN) to provide a plug-and-play joint image-text modeling framework. VHE-GAN integrates off-the-shelf image encoders, text decoders, and GAN image discriminators and generators into a coherent end-to-end learning objective. It couples its VHE and GAN components by feeding the VHE variational posterior in lieu of noise as the source of randomness of the GAN generator. We show VHE-StackGAN++ that combines the Poisson gamma belief network, a deep topic model, and StackGAN++ achieves competitive performance, and VHE-raster-scan-GAN, which further improves VHE-StackGAN++ by exploiting the semantically-meaningful hierarchical structure of the deep topic model, generates photo-realistic images not only in a multi-scale low-to-high-resolution manner, but also in a hierarchical-semantic coarse-to-fine fashion, achieving outstanding results in many challenging image-to-text, text-to-image, and joint text-image learning and generation tasks.

References

[1] L. Gomez, Y. Patel, M. Rusinol, D. Karatzas, and C. V. Jawahar, "Self-supervised learning of visual features through embedding images into text topic spaces," in *CVPR*, 2017, pp. 2017–2026.

[2] R. Kiros and C. Szepesvari, "Deep representations and codes for image auto-annotation," pp. 908–916, 2012.

[3] S. E. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *ICML*, 2016, pp. 1060–1069.

[4] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He, "AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks," in *CVPR*, 2018.

[5] H. Zhang, T. Xu, and H. Li, "StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *CVPR*, 2017.

[6] N. Srivastava and R. Salakhutdinov, "Learning representations for multimodal data with deep belief nets," in *NIPS workshop*, 2012, pp. 2222–2230.

[7] ——, "Multimodal learning with deep Boltzmann machines," in *NIPS*, 2012, pp. 2222–2230.

[8] C. Wang, B. Chen, and M. Zhou, "Multimodal Poisson gamma belief network," in *AAAI*, 2018.

[9] D. P. Kingma and M. Welling, "Stochastic gradient VB and the variational auto-encoder," in *ICLR*, 2014.

[10] M. Zhou, Y. Cong, and B. Chen, "Augmentable gamma belief networks," *Journal of Machine Learning Research*, vol. 17, no. 163, pp. 1–44, 2016.

[11] H. Zhang, B. Chen, D. Guo, and M. Zhou, "WHAI: Weibull hybrid autoencoding inference for deep topic modeling," in *ICLR*, 2018.

[12] I. J. Goodfellow, J. Pougetabadie, M. Mirza, B. Xu, D. Wardefarley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014, pp. 2672–2680.

[13] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas, "StackGAN++: Realistic image synthesis with stacked generative adversarial networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2017.

[14] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He, "AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks," in *CVPR*, 2018, pp. 1316–1324.

[15] V. K. Verma, G. Arora, A. K. Mishra, and P. Rai, "Generalized zero-shot learning via synthesized examples," in *CVPR*, 2018, pp. 4281–4289.

[16] Z. Zhang, Y. Xie, and L. Yang, "Photographic text-to-image synthesis with a hierarchically-nested adversarial network," in *CVPR*, 2018.

[17] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Machine learning*, vol. 37, no. 2, pp. 183–233, 1999.

[18] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American Statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.

[19] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 1303–1347, 2013.

[20] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *ICML*, 2014, pp. 1278–1286.

[21] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, no. 6, pp. 1137–1155, 2003.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[24] A. B. Dieng, C. Wang, J. Gao, and J. Paisley, "TopicRNN: A recurrent neural network with long-range semantic dependency," in *ICLR*, 2017.

[25] J. H. Lau, T. Baldwin, and T. Cohn, "Topically driven neural language model." in *ACL*, 2017, pp. 355–365.

[26] D. Wang, S. Zhu, T. Li, and Y. Gong, "Multi-document summarization using sentence-based topic models," in *ACL*, 2009, pp. 297–300.

[27] J. Jin, K. Fu, R. Cui, F. Sha, and C. Zhang, "Aligning where to see and what to tell: image caption with region-based attention and scene factorization," in *CVPR*, 2015.

[28] N. Srivastava and R. Salakhutdinov, "Multimodal learning with deep Boltzmann machines," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 2949–2980, 2014.

[29] M. Elhoseiny, Y. Zhu, H. Zhang, and A. M. Elgammal, "Link the head to the "beak": Zero shot learning from noisy text description at part precision," in *CVPR*, 2017, pp. 6288–6297.

[30] Y. Zhu, M. Elhoseiny, B. Liu, X. Peng, and A. M. Elgammal, "A generative adversarial approach for zero-shot learning from noisy texts," in *CVPR*, 2018.

[31] Y. Cong, B. Chen, H. Liu, and M. Zhou, "Deep latent Dirichlet allocation with topic-layer-adaptive stochastic gradient Riemannian MCMC," in *ICML*, 2017.

[32] M. Zhou, L. Hannah, D. Dunson, and L. Carin, "Beta-negative binomial process and Poisson factor analysis," in *AISTATS*, 2012, pp. 1462–1471.

[33] M. Zhou and L. Carin, "Negative binomial process count and mixture modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 2, pp. 307–320, 2015.

[34] M. Zhou, "Infinite edge partition models for overlapping community detection and link prediction," in *AISTATS*, 2015, pp. 1135–1143.

[35] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *CVPR*, 2016, pp. 2818–2826.

[36] I. Gulrajani, K. Kumar, F. Ahmed, A. A. Taiga, F. Visin, D. Vazquez, and A. C. Courville, "PixelVAE: A latent variable model for natural images," in *ICLR*, 2017.

[37] E. L. Denton, S. Chintala, A. Szlam, and R. Fergus, "Deep generative image models using a laplacian pyramid of adversarial networks," in *NIPS*, 2015, pp. 1486–1494.

[38] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," Tech. Rep., 2011.

[39] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on*. IEEE, 2008, pp. 722–729.

[40] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014, pp. 740–755.

[41] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *NIPS*, 2016, pp. 2234–2242.

[42] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local nash equilibrium," in *NIPS*, 2017, pp. 6626–6637.

[43] Y. Fu, T. Xiang, Y. G. Jiang, X. Xue, L. Sigal, and S. Gong, "Recent advances in zero-shot recognition," *IEEE Signal Processing Magazine*, vol. 35, 2018.

[44] M. Elhoseiny, A. M. Elgammal, and B. Saleh, "Write a classifier: Predicting visual classifiers from unstructured text," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2539–2553, 2017.

[45] R. Qiao, L. Liu, C. Shen, and A. V. Den Hengel, "Less is more: Zero-shot learning from online textual documents with noise suppression," in *CVPR*, 2016, pp. 2249–2257.

[46] B. Romeraparedes and P. H. S. Torr, "An embarrassingly simple approach to zero-shot learning," in *ICML*, 2015, pp. 2152–2161.

[47] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha, "Synthesized classifiers for zero-shot learning," in *CVPR*, 2016, pp. 5327–5336.

[48] M. D. Hoffman and M. J. Johnson, "ELBO surgery: Yet another way to carve up the variational evidence lower bound," in *Workshop in Advances in Approximate Bayesian Inference, NIPS*, 2016.

[49] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," *arXiv preprint arXiv:1511.05644*, 2015.

[50] L. Mescheder, S. Nowozin, and A. Geiger, "Adversarial variational Bayes: Unifying variational autoencoders and generative adversarial networks," in *ICML*. PMLR, 2017, pp. 2391–2400.

[51] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf, "Wasserstein auto-encoders," in *ICLR*, 2018.

[52] V. Dumoulin, I. Belghazi, B. Poole, A. Lamb, M. Arjovsky, O. Mastropietro, and A. C. Courville, "Adversarially learned inference," in *ICLR*, 2017.

[53] J. Donahue, P. Krahenbuhl, and T. Darrell, "Adversarial feature learning," in *ICLR*, 2017.

[54] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li, "Mode regularized generative adversarial networks," in *ICLR*, 2017.

[55] A. Srivastava, L. Valkoz, C. Russell, M. U. Gutmann, and C. A. Sutton, "VEEGAN: Reducing mode collapse in GANs using implicit variational learning," in *NIPS*, 2017, pp. 3308–3318.

[56] A. Grover, M. Dhar, and S. Ermon, "Flow-GAN: Combining maximum likelihood and adversarial learning in generative models," in *AAAI*, 2018, pp. 3069–3076.

[57] A. B. L. Larsen, S. K. Sonderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *ICML*, 2016, pp. 1558–1566.

[58] H. Huang, Z. Li, R. He, Z. Sun, and T. Tan, "IntroVAE: Introspective variational autoencoders for photographic image synthesis," in *NeurIPS*, 2018.

[59] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele, "Evaluation of output embeddings for fine-grained image classification," in *CVPR*, 2015, pp. 2927–2936.

[60] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

Variational Hetero-Encoder Randomized Generative Adversarial Networks for Joint Image-Text Modeling: Supplementary Material

A Model property of VHE-GAN and related work

Let us denote $q(\boldsymbol{z}) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[q(\boldsymbol{z} \mid \boldsymbol{x})] = \frac{1}{N}\sum_{n=1}^{N} q(\boldsymbol{z} \mid \boldsymbol{x}_n)$ as the aggregated posterior [48, 49]. Removing the triple-data-reuse training strategy, we can re-express the VHE-GAN objective in (4) as

$$\min_{E,G_{\text{vae}},G_{\text{gan}}} \max_{D} [-\text{ELBO}_{\text{vhe}} + \mathcal{L}_{\text{gan}}], \;\; \mathcal{L}_{\text{gan}} := \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})} \ln D(\boldsymbol{x}) + \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z})} \ln(1 - D(G_{\text{gan}}(\boldsymbol{z}))), \tag{8}$$

which corresponds to a naive combination of the VHE and GAN training objectives, where the data samples used to train the VHE, GAN generator, and GAN discriminator in each gradient update iteration are not imposed to be the same. While the naive objective in (8) differs from (4) that is used to train VHE-GAN, it simplifies the analysis of its theoretical property, as described below.

Let us denote $q(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{t}) := q(\boldsymbol{z} \mid \boldsymbol{x}) p_{data}(\boldsymbol{x}, \boldsymbol{t})$ as the joint distribution of $(\boldsymbol{x}, \boldsymbol{t})$ and $\boldsymbol{z}$ under the VHE variational posterior $q(\boldsymbol{z} \mid \boldsymbol{x})$, $I_q(\boldsymbol{x}, \boldsymbol{z}) := \mathbb{E}_{q(\boldsymbol{z},\boldsymbol{x})}\big[\ln \frac{q(\boldsymbol{z},\boldsymbol{x})}{q(\boldsymbol{z})p_{data}(\boldsymbol{x})}\big]$ as the mutual information between $\boldsymbol{x} \sim p_{data}(\boldsymbol{x})$ and $\boldsymbol{z} \sim q(\boldsymbol{z})$, and $\text{JDS}(p_1||p_2) := \frac{1}{2}\text{KL}[p_1||(p_1+p_2)/2] + \frac{1}{2}\text{KL}[p_2||(p_1+p_2)/2]$ as the Jensen–Shannon divergence between distributions p_1 and p_2. Similar to the analysis in Hoffman and Johnson [48], the VHE's ELBO can be rewritten as $\text{ELBO}_{\text{vhe}} = \mathbb{E}_{q(\boldsymbol{z},\boldsymbol{x},\boldsymbol{t})}[\log p(\boldsymbol{t} \mid \boldsymbol{z})] - I_q(\boldsymbol{x}, \boldsymbol{z}) - \text{KL}[q(\boldsymbol{z})||p(\boldsymbol{z})]$, where the mutual information term can also be expressed as $I_q(\boldsymbol{x}, \boldsymbol{z}) = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}\text{KL}[q(\boldsymbol{z} \mid \boldsymbol{x})||q(\boldsymbol{z})]$. Thus maximizing the ELBO encourages the mutual information term $I_q(\boldsymbol{x}, \boldsymbol{z})$ to be minimized, which means while the data reconstruction term $\mathbb{E}_{q(\boldsymbol{z},\boldsymbol{x},\boldsymbol{t})}[\log p(\boldsymbol{t} \mid \boldsymbol{z})]$ needs to be maximized, part of the VHE optimization objective penalizes a $\boldsymbol{z}$ from carrying the information of the $\boldsymbol{x}$ that it is encoded from. This mechanism helps provide necessary regularization to prevent overfitting. As in Goodfellow et al. [12], with an optimal discriminator D_G^* for generator G, we have $\min \mathcal{L}_{\text{GAN}}(D_G^*, G) = \ln 4 + 2\text{JSD}(p_{data}(\boldsymbol{x})||p_{G_{\boldsymbol{z}}}(\boldsymbol{x}))$, where $p_{G_{\boldsymbol{z}}(\boldsymbol{x})}$ denotes the distribution of the generated data $G(\boldsymbol{z})$ that use $\boldsymbol{z} \sim q(\boldsymbol{z})$ as the random source fed into the GAN generator. The JSD term is minimized when $p_{G_{\boldsymbol{z}}}(\boldsymbol{x}) = p_{data}(\boldsymbol{x})$.

With these analyses, given an optimal GAN discriminator, the naive VHE-GAN objective function in (8) reduces to

$$\min_{E,G_{\text{gan}},G_{\text{vae}}} -\mathbb{E}_{q(\boldsymbol{z},\boldsymbol{x},\boldsymbol{t})}[\log p(\boldsymbol{t} \mid \boldsymbol{z})] + \text{KL}[q(\boldsymbol{z})||p(\boldsymbol{z})] + I_q(\boldsymbol{x}, \boldsymbol{z}) + 2\text{JSD}(p_{data}(\boldsymbol{x})||p_{G_{\boldsymbol{z}}}(\boldsymbol{x})). \tag{9}$$

From the VHEs' point of view, examining (9) shows that it alleviates the inherent conflict in VHE of maximizing the ELBO and maximizing the mutual information $I_q(\boldsymbol{x}, \boldsymbol{z})$. This is because while the VHE part of VHE-GAN still relies on minimizing $I_q(\boldsymbol{x}, \boldsymbol{z})$ to regularize the learning, the GAN part tries to transform $q(\boldsymbol{z})$ through the GAN generator to match the true data distribution $p_{data}(\boldsymbol{x})$. In other words, while its VHE part penalizes a $\boldsymbol{z}$ from carrying the information about the $\boldsymbol{x}$ that it is encoded from, its GAN part encourages a $\boldsymbol{z}$ to carry information about the true data distribution $p_{data}(\boldsymbol{x})$, but not necessarily the observed $\boldsymbol{x}$ that it is encoded from.

From the GANs' point of view, examining (9) shows that it provides GAN with a meaningful latent space, necessary for performing inference and data reconstruction (with the aid of the data-triple-use training strategy). More specifically, this latent representation is also used by the VHE to maximize the data log-likelihood, a training procedure that tries to cover all modes of the empirical data distribution rather than dropping modes. For VHE-GAN (4), the source distribution is $q(\boldsymbol{z} \mid \boldsymbol{x})$, not only allowing GANs to participate in posterior inference and data reconstruction, but also helping GANs resist mode collapse. Below we discuss some related work on combining VAEs and GANs.

A.1 Related work on combining VAEs and GANs

Examples in improving VAEs with adversarial learning include Mescheder et al. [50], which allows the VAEs to take implicit encoder distribution, and adversarial auto-encoder [49] and Wasserstein auto-encoder [51], which drop the mutual information term from the ELBO and use adversarial learning to match the aggregated posterior and prior. Examples in allowing GANs to perform inference include Dumoulin et al. [52] and Donahue et al. [53], which use GANs to match the

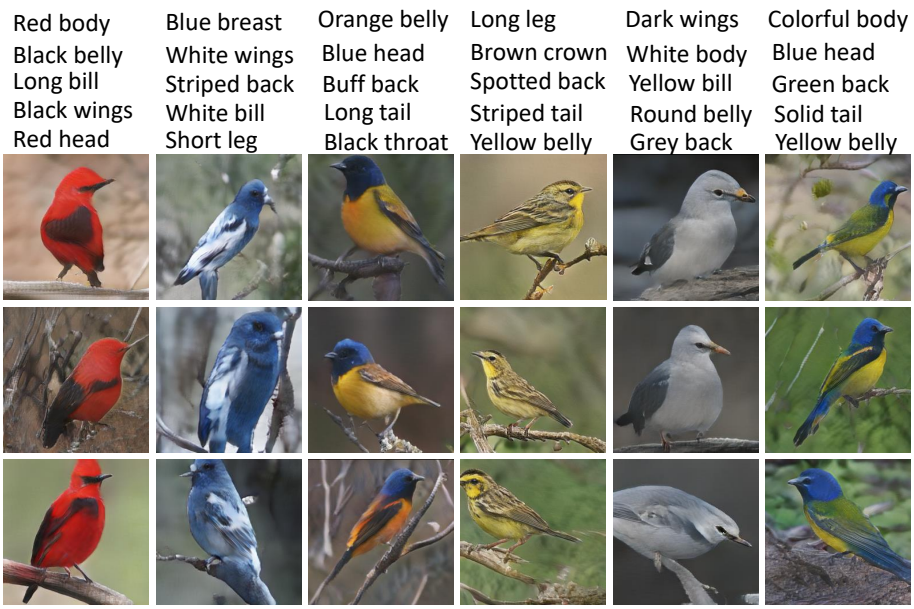


Figure 7: Generated random images by VHE-raster-scan-GAN conditioning on five binary attributes.

joint distribution $q(\boldsymbol{z} \mid \boldsymbol{x})p_{data}(\boldsymbol{x})$ defined by the encoder and the one $p(\boldsymbol{x} \mid \boldsymbol{z})p(\boldsymbol{z})$ defined by the generator. However, they often do not provide good data reconstruction. Examples in using VAEs or maximum likelihood to help GANs resist mode collapse include [54–56]. Another example is VAEGAN [57] that combines unit-wise likelihood at hidden layer and adversarial loss at original space, but its update of the encoder is separated from the GAN mini-max objective. On the contrary, IntroVAE [58] retains the pixel-wise likelihood with an adversarial regularization on the latent space. Sharing network between the VAE decoder and GAN generator in VAEGAN and IntroVAE, however, limit them to model a single modality.

B More discussion on sequence models and topic models and additional example results of text-to-image generation

In Section 3.1, we have discussed two models to represent the text: sequence models and topic models. Considering the versatility of topic models [7, 8, 10, 26–30] in dealing with different types of textual information, and its effectiveness in capturing latent topics that are often directly related to macro-level visual information [1, 24, 25], we choose a state-of-the-art deep topic model, PGBN, to model the textual descriptions in VHE. Due to space constraint, we only provide simple illustrations in Figs. 3(a) and 3(b). In this section, more insights and discussions are provided.

As discussed before, topic models are able to model non-sequential texts such as binary attributes. The CUB dataset provides 312 binary attributes [38] for each images, such as whether "crown color is blue" and whether "tail shape is solid" to define the color or shape of different body parts of a bird. We first transform these binary attributes for the nth image to a 312-dimensional binary vector $\boldsymbol{t}_n$, whose ith element is 1 or 0 depending on whether the bird in this image owns the ith attribute or not. The binary attribute vectors $\boldsymbol{t}_n$ are used together with the corresponding bird images $\boldsymbol{x}_n$ to train VHE-raster-scan-GAN. As shown in Fig. 7, we generate images given five binary attributes, which are formed into a 312-dimensional binary vector $\boldsymbol{t}$ (with five non-zero elements at these five attributes) that becomes the input to the PGBN text decoder. Clearly, these generated images are photo-realistic and faithfully represent the five provided attributes.

The proposed VHE-GANs can also well model long documents. In text-based ZSL discussed in Section 3.2, each class (not each image) is represented as a long encyclopedia document, whose global semantic structure is hard to captured by existing sequence models. Besides a good ZSL performance achieved by VHE-raster-scan-GAN, illustrating its advantages of text generation given images, we show Fig. 8 example results of image generation conditioning on long encyclopedia documents on the unseen classes of CUB-E [45, 59] and Flower [44].

Class name: Rhinoceros Auklet

It is a seabird, nesting in seabird colonies, with a large orange/brown bill. Plumage is dark on top and paler below, in offshore and inshore water.
Sometimes it swim in the water and sometimes it stand on the strong.

Class name: Yellow Bellied Flycatcher

Brownish-olive upperparts, darker on the wings and tail, yellowish underparts.
Have small bill short tail, on a perch low or in the middle of a tree.
Its eyes are dark and round with radiating vigor, like looking for food or insects.

(a)

Class name: Ball Moss

It tends to form a spheroid shape ranging in size from a golf ball to a soccer ball.
It may hinder tree growth.
Its petals are stripe-like yellow ones and its stamen is also round dark brown or yellow.

Class name: Barberton Daisy

It bear a large capitulum with striking, two-lipped ray floret in yellow or orange.
Colors include white, yellow, and pink.
Its petals are medium, and each of them is round and the number is about six.

(b)

Figure 8: Image generation conditioning on long encyclopedia documents using VHE-raster-scan-GAN trained on (a) CUB-E and (b) Flower. Shown in the top part of each subplot are representative sentences taken from the long document that describes an unseen class; for the three rows of images shown in the bottom part, the first row includes three real images from the corresponding unseen class, and the other two rows include a total of six randomly generated images conditioning on the long encyclopedia document of the corresponding unseen class.

C More experimental results on joint image-text learning

C.1 Table 1 and Table 2 with error bars.

For text-to-image generation tasks, we use the official pre-defined training/testing split (illustrated in Appendix F) to train and test all the models. Following the definition of error bar of IS in StackGAN++ [13], HDGAN [16], and AttnGAN [14], we provide the IS results with error bars for various methods in Table 3, where the results of the StackGAN++ , HDGAN, and AttnGAN are quoted from the published papers. The FID error bar is not included as it has not been clearly defined.

Table 3: Inception score (IS) results in Table 1 with error bars.

Method	StackGAN++	HDGAN	AttnGAN	PGBN+StackGAN++	VHE-StackGAN++	VHE-raster-scan-GAN
Flower	3.26 ± .01	3.45 ± .07	–	3.29 ± .02	3.56 ± .03	**3.72 ± .01**
CUB	3.84 ± .06	4.15 ± .05	4.36 ± .03	3.92 ± .06	4.20 ± .04	**4.41 ± .03**
COCO	8.30 ± .10	11.86 ± .18	25.89 ± .47	10.63 ± .10	12.63 ± .15	**27.16 ± .23**

For text-based ZSL tasks, we also use the official pre-defined training/testing splits. We collect the ZSL results of the last 1000 mini-batch based stochastic gradient update iterations to calculate the error bars. For existing methods, since there are no error bars provided in published paper, we only provide the text error bars of the methods that have publicly accessible code.

Table 4: Test errors and error bars of each models in text-based ZSL.

Text-ZSL dataset	CUB-H	CUB-E		Flower
Accuracy criterion	top-1	top-1	top-5	top-1
WAC-Kernel [44]	7.7 ± 0.28	33.5 ± 0.22	64.3 ± 0.20	9.1 ± 2.77
ZSLNS [45]	7.3 ± 0.36	29.1 ± 0.28	61.8 ± 0.22	8.7 ± 2.46
ESZSL* [46]	7.4 ± 0.31	28.5 ± 0.26	59.9 ± 0.20	8.6 ± 2.53
SynC* [47]	8.6	28.0	61.3	8.2
ZSLPP [29]	9.7	37.2	–	–
GAZSL [30]	10.3 ± 0.26	**43.7** ± 0.28	67.61 ± 0.24	–
VHE-L3	14.0 ± 0.24	34.6 ± 0.25	64.6 ± 0.20	8.9 ± 1.57
VHE-StackGAN++-L3	16.1	38.5	68.2	10.6
VHE-raster-scan-GAN-L1	11.7 ± 0.31	32.1 ± 0.32	62.6 ± 0.33	9.4 ± 1.68
VHE-raster-scan-GAN-L2	14.9 ± 0.26	37.1 ± 0.24	64.6 ± 0.25	11.0 ± 1.54
VHE-raster-scan-GAN-L3	**16.7 ± 0.24**	39.6 ± 0.20	**70.3 ± 0.18**	**12.1 ± 1.47**

C.2 Larger-size replot of Figure 2

Due to space constraint, we provide relative small-size images in Fig. 2. Below we show the corresponding images with larger sizes.

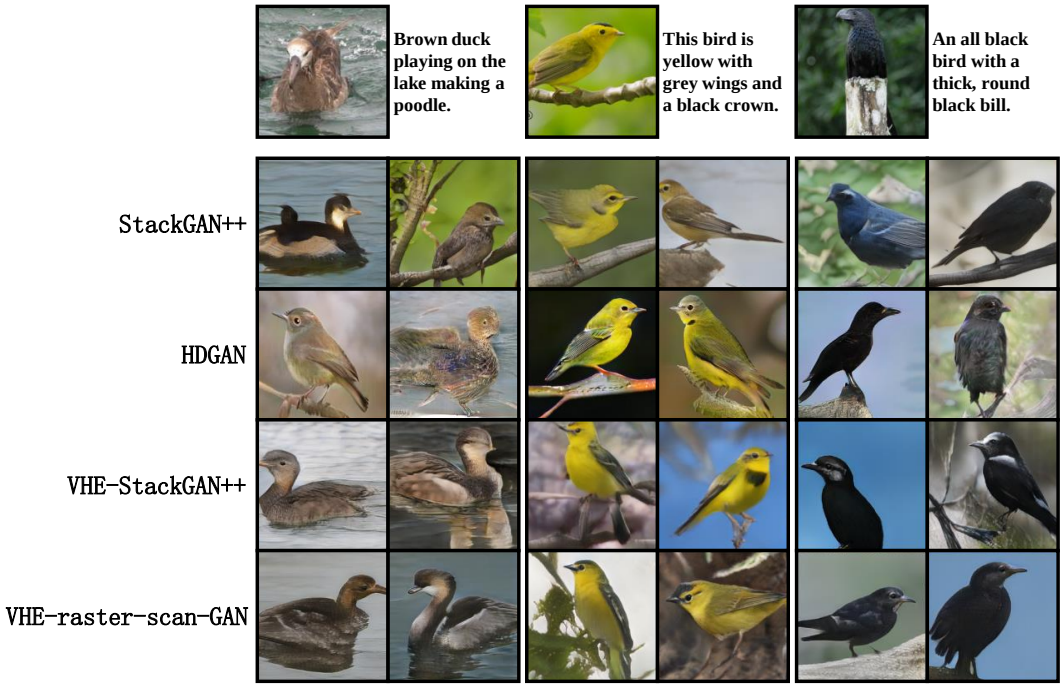


Figure 9: Larger-size replots of CUB Bird images in Figure 2.

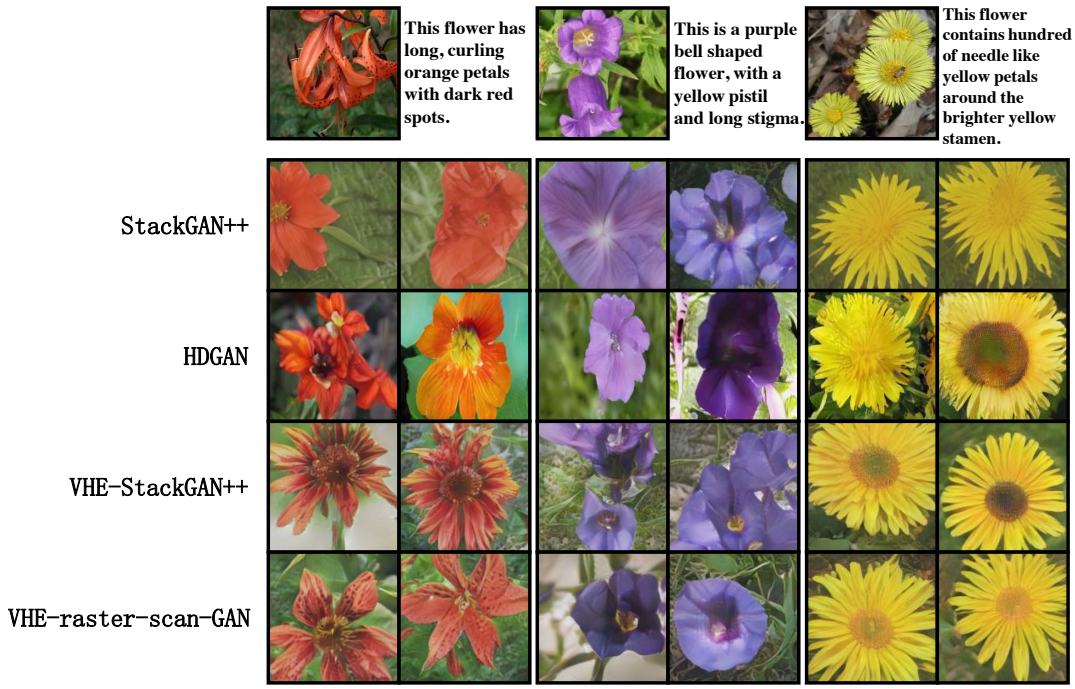


Figure 10: Larger-size replots of the Flower images in Figure 2.

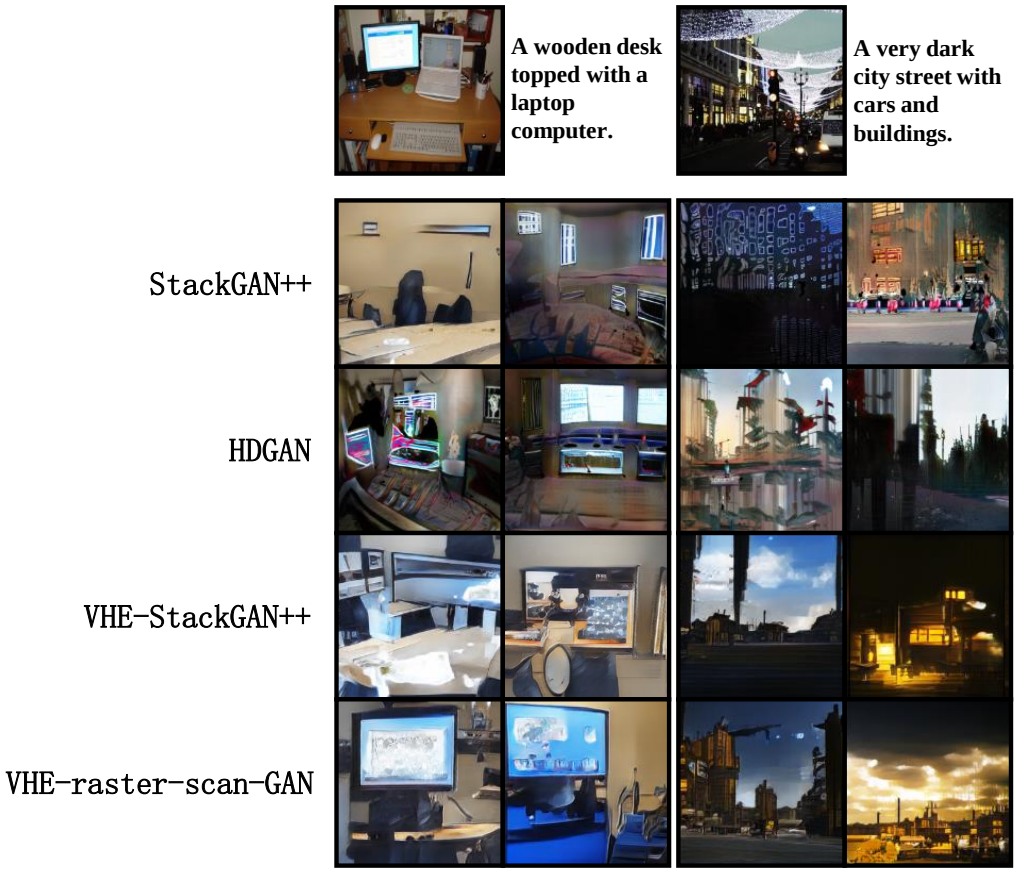


Figure 11: Larger-size replots of the COCO images in Figure 2.

C.3 More text-to-image generation results on COCO

COCO is a more challenging dataset than CUB and Flower, as it contains very diverse objects and scenes. We show in Fig. 12 more samples conditioned on different textural descriptions.

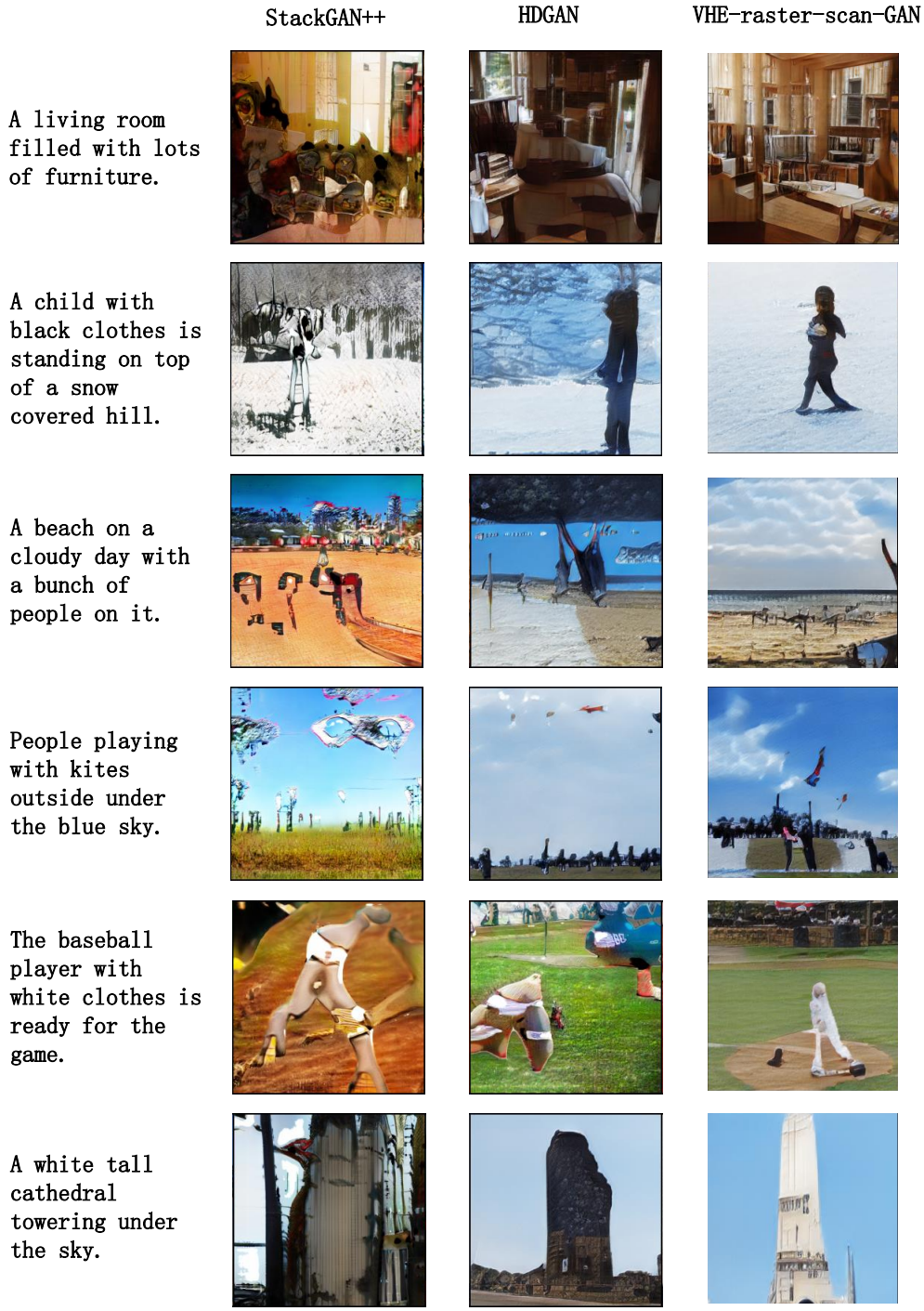


Figure 12: Example text-to-image generation results on COCO.

C.4 Latent space interpolation

In addition to the latent space interpolation results of VHE-raster-scan-GAN in Fig. 3(c) of Section 3.1, below we provide more fine-gridded latent space interpolation in Figs. 14-17.

Figure 13: Example of latent space interpolation on CUB.

Figure 14: Example of latent space interpolation on CUB.

Figure 15: Example of latent space interpolation on CUB.

Figure 16: Example of latent space interpolation on Flower.

Figure 17: Example of latent space interpolation on Flower.

C.5 Image retrieval given a text query

For image $\boldsymbol{x}_n$, we draw its BOW textual description $\hat{\boldsymbol{t}}_n$ as $\hat{\boldsymbol{t}}_n \,|\, \boldsymbol{\theta}_n \sim p(\boldsymbol{t} \,|\, \boldsymbol{\Phi}, \boldsymbol{\theta}_n)$, $\boldsymbol{\theta}_n \,|\, \boldsymbol{x}_n \sim q_{\boldsymbol{\Omega}}(\boldsymbol{\theta} \,|\, \boldsymbol{\Phi}, \boldsymbol{x}_n)$. Given the BOW textual description $\boldsymbol{t}$ as a text query, we retrieve the top five images ranked by the cosine distances between $\boldsymbol{t}$ and $\hat{\boldsymbol{t}}_n$'s. Shown in Fig. 18 are three example image retrieval results, which suggest that the retrieved images are semantically related to their text queries in colors, shapes, and locations.

Figure 18: Top-5 retrieved images given a text query. Rows 1 to 3 are for Flower, CUB, and COCO, respectively.

C.6 Image regeneration

We note for VHE-GAN, its image encoder and GAN component together can also be viewed as an "autoencoding" GAN for images. More specifically, given image $\boldsymbol{x}$, VHE-GAN can provide random regenerations using $G\left(q_{\boldsymbol{\Omega}}\left(\boldsymbol{\theta} \,|\, \boldsymbol{\Phi}, \boldsymbol{x}\right)\right)$. We show example image regeneration results by both VHE-StackGAN++ and VHE-raster-scan-GAN in Fig. 19. These example results suggest that the regenerated random images by the proposed VHE-GANs more of less resemble the original real image fed into the VHE image encoder.

Figure 19: Example results of image regeneration using VHE-StackGAN++ and VHE-raster-scan-GAN. An original image is fed into the VHE image encoder, whose latent representation is then fed into the GAN image generator to generate a corresponding random image. The models in columns 1-4 are trained on Flower, columns 5-8 on CUB, and columns 9-12 on COCO.

C.7 Learned hierarchical topics in VHE

The inferred topics at different layers and the inferred sparse connection weights between the topics of adjacent layers are found to be highly interpretable. In particular, we can understand the meaning of each topic by projecting it back to the original data space via $\left[\prod_{t=1}^{l-1} \boldsymbol{\Phi}^{(t)}\right] \boldsymbol{\phi}_k^{(l)}$ and understand the

relationship between the topics by arranging them into a directed acyclic graph (DAG) and choose its subnets to visualize. We show in Figs. 20, 21, and 22 example subnets taken from the DAGs inferred by the three-layer VHE-raster-scan-GAN of size 256-128-64 on Flower, CUB, and COCO, respectively. The semantic meaning of each topic and the connection weights between the topics of adjacent layers are highly interpretable. For example, in Figs. 20, the topics describe very specific flower characteristics, such as special colors, textures, shapes, and parts, at the bottom layer, and become increasingly more general when moving upwards.

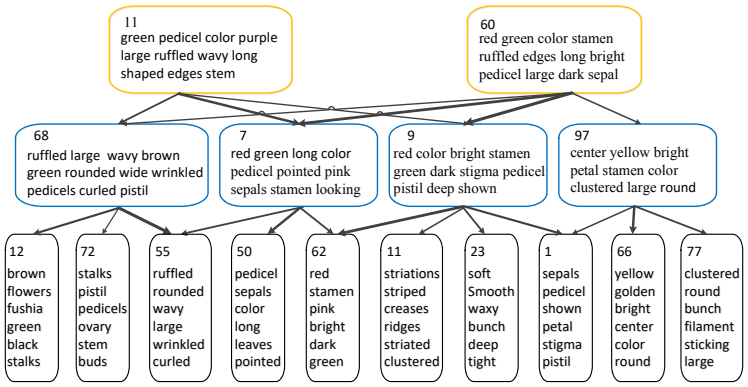


Figure 20: An example topic hierarchy taken from the directed acyclic graph learned by a three-layer VHE-raster-scan-GAN of size 256-128-64 on Flower.

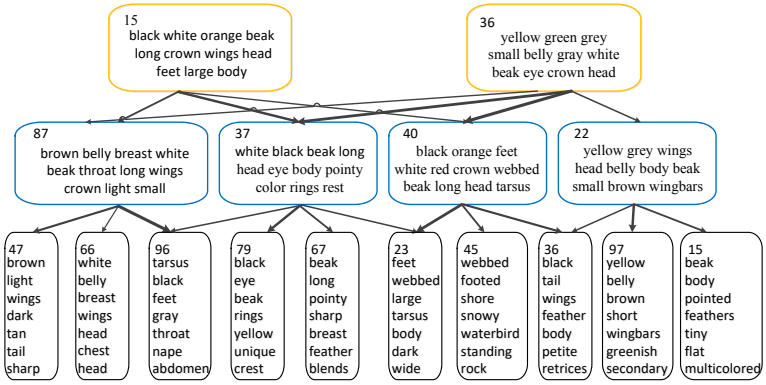


Figure 21: Analogous plot to Fig. 20 on CUB.

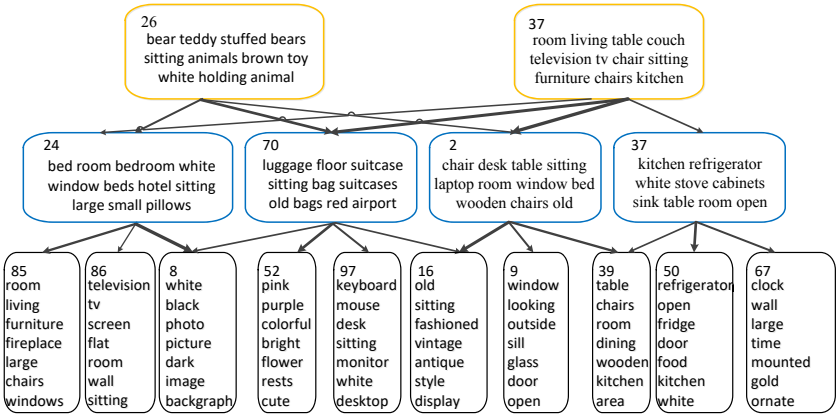


Figure 22: Analogous plot to Fig. 20 on COCO.

D Specific model structure in VHE-StackGAN++ and VHE-raster-scan-GAN

D.1 Model structure of VHE

In Fig. 23, we give the structure of VHE used in VHE-StackGAN++ and VHE-raster-scan-GAN, where $f(\boldsymbol{x})$ is the image features extracted by Inception v3 network and $\boldsymbol{\varepsilon}^{(l)} \sim \prod_{k=1}^{K_l} \text{Uniform}(\varepsilon_k^{(l)}; 0, 1)$. With the definition of $\boldsymbol{g}^{(0)} = f(\boldsymbol{x})$, we have

$$\boldsymbol{k}^{(l)} = \exp(\mathbf{W}_1^{(l)} \boldsymbol{g}^{(l)} + \boldsymbol{b}_1^{(l)}), \tag{10}$$

$$\boldsymbol{\lambda}^{(l)} = \exp(\mathbf{W}_2^{(l)} \boldsymbol{g}^{(l)} + \boldsymbol{b}_2^{(l)}), \tag{11}$$

$$\boldsymbol{g}^{(l)} = \ln[1 + \exp(\mathbf{W}_3^{(l)} \boldsymbol{g}^{(l-1)} + \boldsymbol{b}_3^{(l)})], \tag{12}$$

where $\mathbf{W}_1^{(l)} \in \mathbb{R}^{K_l \times K_l}$, $\mathbf{W}_2^{(l)} \in \mathbb{R}^{K_l \times K_l}$, $\mathbf{W}_3^{(l)} \in \mathbb{R}^{K_l \times K_{l-1}}$, $\boldsymbol{b}_1^{(l)} \in \mathbb{R}^{K_l}$, $\boldsymbol{b}_2^{(l)} \in \mathbb{R}^{K_l}$, and $\boldsymbol{b}_3^{(l)} \in \mathbb{R}^{K_l}$.

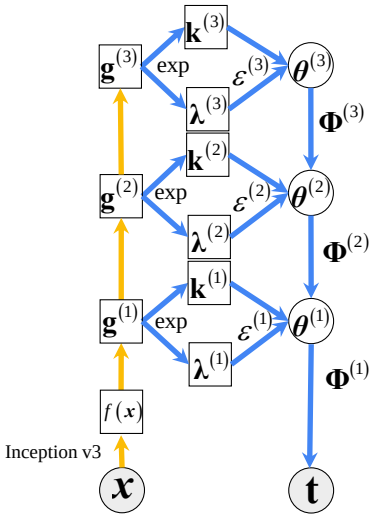


Figure 23: The architecture of VHE in VHE-StackGAN++ and VHE-raster-scan-GAN.

D.2 Model of VHE-StackGAN++

In Section 2.2, we first introduce the VHE-StackGAN++, where the multi-layer textual representation $\{\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \cdots, \boldsymbol{\theta}^{(L)}\}$ is concatenated as $\boldsymbol{\theta} = \left[\boldsymbol{\theta}^{(1)}, \cdots, \boldsymbol{\theta}^{(L)}\right]$ and then fed into StackGAN++ [13]. In Figs. 1 (a) and (b), we provide the model structure of VHE-StackGAN++. We also provide a detailed plot of the structure of StackGAN++ used in VHE-StackGAN++ in Fig. 24, where JCU is a specific type of discriminator; see Zhang et al. [13] for more details.

The same with VHE-raster-scan-GAN, VHE-StackGAN++ is also able to jointly optimize all components by merging the expectation in VHE and GAN to define its loss function as

$$\begin{aligned}\min_{\boldsymbol{\Omega}, \{G_i\}_{i=1}^3} \max_{\{D_i\}_{i=1}^3} \mathbb{E}_{p_{\text{data}}(\boldsymbol{x}_n, \boldsymbol{t}_n)} \mathbb{E}_{\prod_{l=1}^L q(\boldsymbol{\theta}_n^{(l)} \mid \boldsymbol{x}_n, \boldsymbol{\Phi}^{(1+1)}, \boldsymbol{\theta}_n^{(l+1)})} \big\{ &- \log p(\boldsymbol{t}_n \mid \boldsymbol{\Phi}^{(1)}, \boldsymbol{\theta}_n^{(1)}) \\ &+ \textstyle\sum_{l=1}^L \text{KL}[q(\boldsymbol{\theta}_n^{(l)} \mid \boldsymbol{x}_n, \boldsymbol{\Phi}^{(1+1)}, \boldsymbol{\theta}_n^{(l+1)}) \,||\, p(\boldsymbol{\theta}_n^{(l)} \mid \boldsymbol{\Phi}^{(1+1)}, \boldsymbol{\theta}_n^{(l+1)})] \\ &+ \textstyle\sum_{i=1}^3 [\log D_i(\boldsymbol{x}_{n,i}, \boldsymbol{\theta}_n) + \log(1 - D_i(G_i(\boldsymbol{\theta}_n), \boldsymbol{\theta}_n))] \big\}. \end{aligned} \tag{13}$$

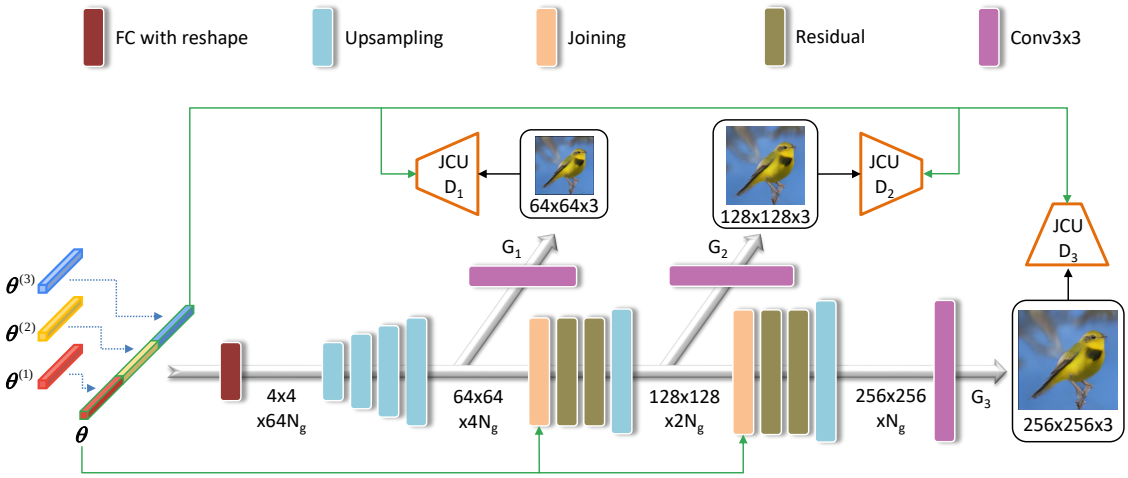


Figure 24: The structure of Stack-GAN++ in VHE-StackGAN++, where JCU is a type of discriminator proposed in Zhang et al. [13].

D.3 Structure of raster-scan-GAN

In Fig. 25, we provide a detailed plot of the structure of the proposed raster-scan-GAN.

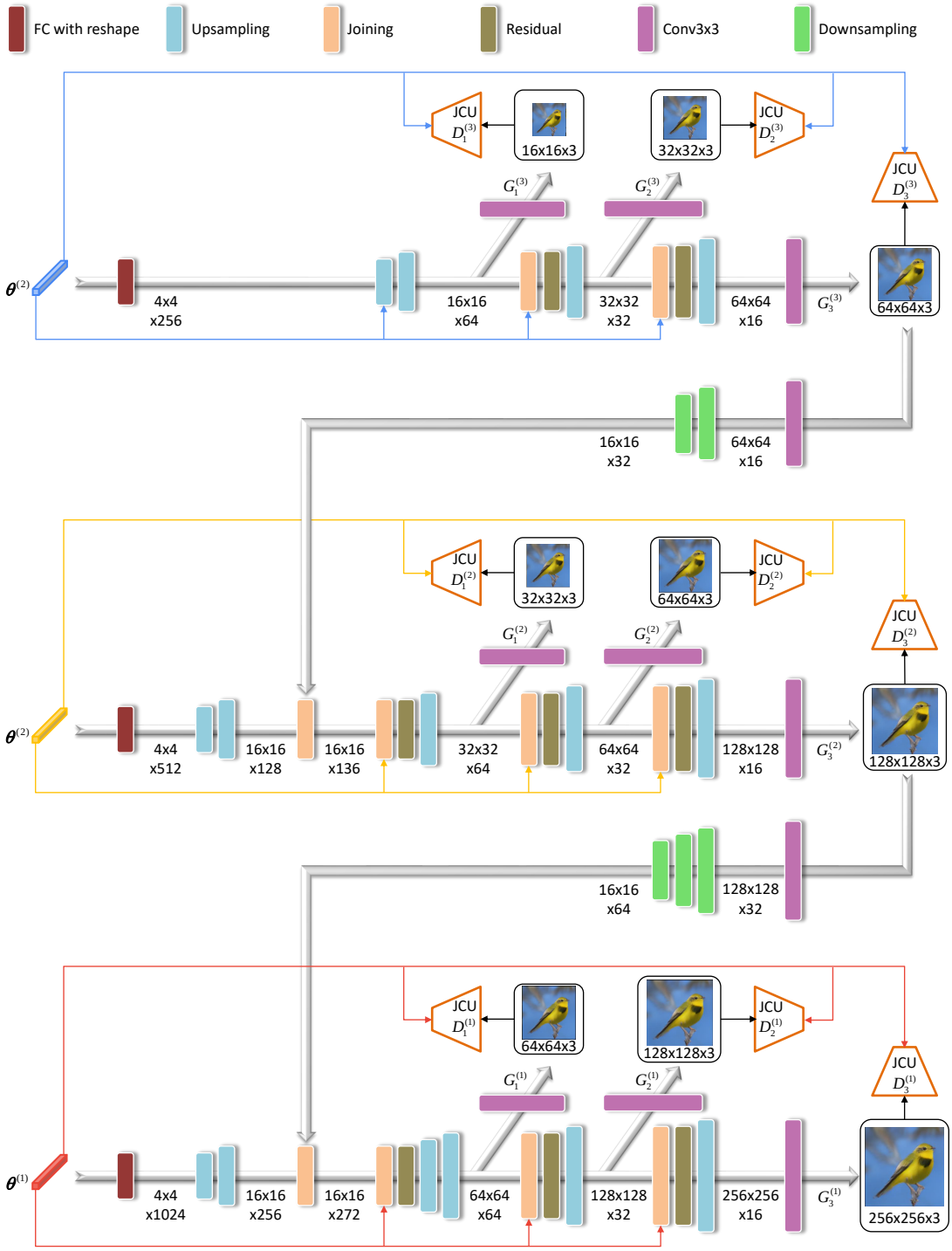


Figure 25: The structure of raster-scan-GAN in VHE-raster-scan-GAN, where JCU is a type of discriminator proposed in Zhang et al. [13].

E Joint optimization for VHE-raster-scan-GAN

Based on the loss function of VHE-raster-scan-GAN (7), with TLASGR-MCMC [31] and WHAI [11], we describe in Algorithm 1 how to perform mini-batch based joint update of all model parameters.

Algorithm 1 Hybrid TLASGR-MCMC/VHE inference algorithm for VHE-raster-scan-GAN.

Initialize encoder parameters $\boldsymbol{\Omega}$, topic parameters of PGBN $\{\boldsymbol{\Phi}^{(l)}\}_{1,L}$, generator G, and discriminator D.
for $iter = 1, 2, \cdots$ **do**
  Randomly select a mini-batch containing N image-text pairs $\boldsymbol{d} = \{\boldsymbol{x}_n, \boldsymbol{t}_n\}_{n=1}^{N}$;
  Draw random noise $\left\{\boldsymbol{\varepsilon}_n^{(l)}\right\}_{n=1,l=1}^{N,L}$ from uniform distribution;
  Calculate $\nabla_D \mathcal{L}\left(D, G, \boldsymbol{\Omega} \,|\, \boldsymbol{x}\right)$;
  Calculate $\nabla_G \mathcal{L}\left(D, G, \boldsymbol{\Omega} \,|\, \boldsymbol{x}\right)$;
  Calculate $\nabla_{\boldsymbol{\Omega}} L$ by the aid of $\left\{\boldsymbol{\varepsilon}_n^{(l)}\right\}_{n=1,l=1}^{N,L}$;
  Update D as $D = D + \nabla_D \mathcal{L}\left(D, G, \boldsymbol{\Omega} \,|\, \boldsymbol{x}\right)$;
  Update G as $G = G - \nabla_G \mathcal{L}\left(D, G, \boldsymbol{\Omega} \,|\, \boldsymbol{x}\right)$;
  Update $\boldsymbol{\Omega}$ as $\boldsymbol{\Omega} = \boldsymbol{\Omega} - \nabla_{\boldsymbol{\Omega}} L$;
  Sample $\{\boldsymbol{\theta}_n^{(l)}\}_{l=1}^{L}$ from (6) given $\boldsymbol{\Omega}$ and $\{\boldsymbol{\Phi}^{(l)}\}_{l=1}^{L}$, and use $\{\boldsymbol{t}\}_{n=1}^{N}$ to update topics $\{\boldsymbol{\Phi}^{(l)}\}_{l=1}^{L}$ according to TLASGR-MCMC;
end for

F Data description on CUB, Flower, and COCO with training details

In image-text multi-modality learning, CUB [38], Flower [39] and COCO [40] are widely used datasets.

CUB (**`http://www.vision.caltech.edu/visipedia/CUB-200-2011.html`**): CUB contains 200 bird species with 11,788 images. Since 80% of birds in this dataset have object-image size ratios of less than 0.5 [38], as a preprocessing step, we crop all images to ensure that bounding boxes of birds have greater-than-0.75 object-image size ratios, which is the same with all related work. For textual description, Wah et al. [38] provide ten sentences for each image and we collect them together to form BOW vectors. Besides, for each species, Elhoseiny et al. [44] provide its encyclopedia document for text-based ZSL, which is also used in our text-based ZSL experiments.

For CUB, there are two split settings: the hard one and the easy one. The hard one ensures that the bird subspecies belonging to the same super-category should belong to either the training split or test one without overlapping, referred to as CUB-hard (CUB-H in our manuscript). A recently used split setting [45, 59] is super-category split, where for each super-category, except for one subspecies that is left as unseen, all the other are used for training, referred to as CUB-easy (CUB-E in our manuscript). For CUB-H, there are 150 species containing 9410 samples for training and 50 species containing 2378 samples for testing. For CUB-E, there are 150 species containing 8855 samples for training and 50 species containing 2933 samples to testing. We use both of them the for the text-based ZSL, and only CUB-E for all the other experiments as usual.

Flower **`http://www.robots.ox.ac.uk/~vgg/data/flowers/102/index.html`**: Oxford-102, commonly referred to as Flower, contains 8,189 images of flowers from 102 different categories. For textual description, Nilsback and Zisserman [39] provide ten sentences for each image and we collect them together to form BOW vectors. Besides, for each species, Elhoseiny et al. [44] provide its encyclopedia document for text-based ZSL, which is also used in our text-based ZSL experiments in section 4.2.2. There are 82 species containing 7034 samples for training and 20 species containing 1155 samples for testing.

For text-based ZSL, we follow the same way in Elhoseiny et al. [44] to split the data. Specifically, five random splits are performed, in each of which $4/5$ of the classes are considered as "seen classes" for training and $1/5$ of the classes as "unseen classes" for testing. For other experiments, we follow Zhang et al. [13] to split the data.

COCO `http://cocodataset.org/#download`: Compared with Flower and CUB, COCO is a more challenging dataset, since it contains images with multiple objects and diverse backgrounds. To show the generalization capability of the proposed VHE-GANs, we also utilize COCO for evaluation. Following the standard experimental setup for COCO [3, 13], we directly use the pre-split training and test sets to train and evaluate our proposed models. There are 82081 samples for training and 40137 samples for testing.

Training details: we train VHE-rater-scan-GAN in four Nvidia GeForce RTX2080 TI GPUs. The experiments are performed with mini-batch size 32 and about 30.2G GPU memory space. We run 600 epochs to train the models on CUB and Flower, taking about 797 seconds for CUB-E and 713 seconds for Flower for each epoch. We run 100 epochs to train the models on COCO, taking about 6315 seconds for each epoch. We use the Adam optimizer [60] with learning rate $2e-4$, $\beta_1 = 0.5$, and $\beta_2 = 0.999$ to optimize the parameters of the GAN generator and discriminator, and use Adam with learning rate $1e-4$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$ to optimize the VHE parameters. The hyper-parameters to update the topics $\mathbf{\Phi}$ with TLASGR-MCMC are the same with those in Cong et al. [31]