

# UNSUPERVISED MULTIPLE DOMAIN TRANSLATION THROUGH CONTROLLED DISENTANGLEMENT IN VARIATIONAL AUTOENCODER

Antonio Almudévar<sup>1</sup>, Théo Mariotte<sup>2</sup>, Alfonso Ortega<sup>1</sup>, Marie Tahon<sup>2</sup>

<sup>1</sup>ViVoLab, Aragón Institute for Engineering Research (I3A), University of Zaragoza, Spain

<sup>2</sup>LIUM, Le Mans Université, France

almudevar@unizar.es

## ABSTRACT

Unsupervised Multiple Domain Translation is the task of transforming data from one domain to other domains without having paired data to train the systems. Typically, methods based on Generative Adversarial Networks (GANs) are used to address this task. However, our proposal exclusively relies on a modified version of a Variational Autoencoder. This modification consists on the use of two latent variables disentangled in a controlled way by design. One of this latent variables is imposed to depend exclusively on the domain, while the other one must depend on the rest of the variability factors of the data. Additionally, the conditions imposed over the domain latent variable allow for better control and understanding of the latent space. We empirically demonstrate that our approach works on different vision datasets improving the performance of other well known methods. Finally, we prove that, indeed, one of the latent variables stores all the information related to the domain and the other one hardly contains any domain information.

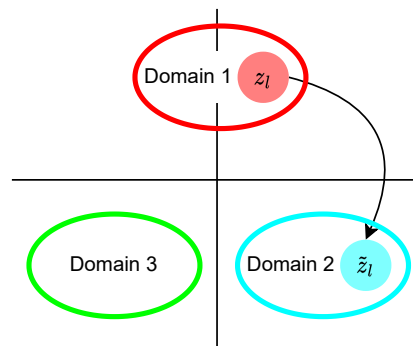
**Index Terms**— multiple domain translation, controlled disentanglement, variational autoencoder

## 1. INTRODUCTION

Domain translation involves the process of converting data from one domain to another while preserving the underlying information or structure. Multiple domain translation is a generalization of the previous task that implies translating between more than two domains simultaneously. In addition, when we do not use data pairs in the source and target domains to design and train the systems, they are said to be unsupervised. Typically, models based on Generative Adversarial Networks (GAN) [1] are used to solve the domain translation problem [2, 3, 4, 5, 6, 7, 8, 9, 10]. However, GANs are usually complex to train and tune. Moreover, they present potential problems such as mode collapse [1]. In this paper we present a method to solve the unsupervised multiple domain translation task using only a Variational Autoencoder (VAE) based model [11]. Thus, we avoid the problems derived from using GANs. To the best of our knowledge, there are no other works using only a VAE to solve this task.

This work was supported by MCIN/AEI/10.13039/501100011033 under Grants PDC2021-120846-C41 & PID2021-126061OB-C44, and in part by the Government of Aragón (Grant Group T36 23R). This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 101007666. This work was performed using HPC resources from GENCI-IDRIS (Grant 2022-AD011012565). The research reported here was conducted at the 2023 Frederick Jelinek Memorial Summer Workshop on Speech and Language Technologies, hosted at Le Mans University (France) and sponsored by Johns Hopkins University.

Specifically, in this paper, we propose to use a VAE that has two independent latent spaces. One of them should contain information only about the domain, while the other one should contain information about everything that is not the domain, which we call style. Equivalently, this means that we try to disentangle the information corresponding to the domain and the style. Thus, to modify the domain and keep only the style, we will modify the latent space corresponding to the domain and leave the one corresponding to the style intact. To achieve this, we impose during training that the latent distribution corresponding to the domain must be close to a target distribution that depends on the domain. Specifically, we locate these distributions in different regions of the space according to the domain. In this way, we perform a transformation of the latent variable by bringing it to the region of the space corresponding to the target domain. In this work, we impose for efficiency that these transformations are linear and, more specifically, rotations. Thus, to translate from one domain to another we perform a rotation in the latent space, as schematized in figure 1, and then decode once this rotation has been performed. The fact of imposing that these transformations are linear also allows us to have a better control and understanding of the latent space. Other transformations could be explored, but it is beyond the scope of this paper.



**Fig. 1:** Scheme of how the translation is performed in a two-dimensional latent space with three domains. In this case, the input belongs to domain 1 and is translated to domain 2 by rotating the latent variable the same angle that separates the regions of the prior distributions of these two domains.

We could summarize the main contributions of this work in the following points:

- We design a VAE in which the target distribution depends on the domain, but the encoder and decoder are common for every domain.

- We construct a VAE with two disentangled latent variables in which one of them depends only on the domain.
- The domain translation is performed by means of a linear transformation latent variable corresponding to the domain.

The code to implement and perform the experiments explained in this paper is available online <sup>1</sup>.

## 2. RELATED WORK

**Domain Translation.** Different GAN-based works have achieved excellent results for domain translation. For example, pix2pix [3] uses Conditional GANs [12] and combines adversarial and L1 losses to train them. However, this method requires paired datasets. Other known methods are unsupervised, i.e., they do not require paired datasets. One of these is CycleGAN [4], which introduces a cycle-consistency loss, which enforces that translated images can be converted back to their original domain. On the other hand, UNIT [6] makes a shared-latent space assumption and proposes an unsupervised image-to-image translation framework based on Coupled GANs. MUNIT [7] extends the capabilities of UNIT by introducing separate content and style latent spaces for each domain, allowing for fine-grained control over content and style. The above proposals, while providing impressive results, only allow translation between two domains and not from multiple domains to multiple domains. The main method in the literature to solve this limitation is StarGAN [9]. This uses a single generator and discriminator to translate between multiple domains. During training, the authors randomly generate a target domain label and train the model to flexibly translate an input image into the target domain. StarGANv2 [10] builds upon StarGAN by introducing a more advanced architecture and improved disentanglement of content and style.

**Disentanglement in VAEs.** Disentangled VAEs aim to learn representations of data where different factors of variation are separated and interpretable. To achieve this, different approaches are relevant.  $\beta$ -VAE [13] introduces a hyperparameter,  $\beta$ , that balances the reconstruction loss and a KL loss, which encourages the model to learn more disentangled representations. Additionally, Factor-VAE [14] extends  $\beta$ -VAE by introducing a total correlation term in the loss function. On the other side, InfoVAE [15] introduces an information-theoretic framework to disentangle factors of variation in data. Concretely, it uses mutual information between latent variables and data to encourage meaningful disentanglement. While these approaches successfully disentangle various sources of variability, they do not provide the ability to specify the characteristics we wish to separate in advance. Determining which dimensions correspond to specific factors must be conducted after training these models [16, 17, 18].

**Conditional VAEs.** In [19] the Conditional VAE (CVAE) is presented, which is a variant of VAE in which the encoder, decoder, and prior distribution of the latent space depend on a condition. In this work, the authors suggest two alternatives: one is to make use of a neural network that takes the condition as input to define the prior distribution. The other is to simplify the problem by maintaining a consistent prior distribution regardless of the condition. In the present work, as we explain below, we make the encoder and the decoder do not depend on the condition. In addition, we define the condition-dependent prior without using a neural network, which allows us to perform domain translation in a very efficient way.

## 3. PROPOSED APPROACH

### 3.1. Variational Autoencoder

In the classical version of the Variational Autoencoder [11], the aim is to generate new data that closely resembles the input data from a dataset. To achieve this goal, a latent variable  $z$  is used, which can be considered to contain relevant information from the input  $x$ . To facilitate this process, authors propose the use of an encoder  $q_\phi(z|x)$ , a decoder  $p_\theta(x|z)$ , and a prior  $p_\theta(z)$ . This prior is usually defined as  $\mathcal{N}(0, I)$ , which implies that  $p_\theta(z)$  is equivalent to  $p(z)$ . The ELBO to maximize in this case is:

$$\mathcal{L}_{VAE} = \mathbb{E}_{q_\phi(z|x)} [p_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p(z)) \quad (1)$$

Once the model has been trained to maximize this ELBO, one or more samples from  $p(z)$  are drawn and passed through the decoder to obtain  $p_\theta(x|z)$ , enabling the generation of new samples.

### 3.2. Conditional Variational Autoencoder

In [19], the authors suggest conditioning data generation on specific conditions denoted as  $c$ . Typically, these conditions align with the class of the input  $x$  in a labeled dataset. In this case we have an encoder  $q_\phi(z|x, c)$ , a decoder  $p_\theta(x|z, c)$  and a prior  $p_\theta(z|c)$ . In the original proposal, this prior is proposed to be a neural network that takes the conditions  $c$  as input. Different architectures for this networks can be used [20]. However, in the majority of applications the relaxation  $p_\theta(z|c) = p(z) = \mathcal{N}(0, I)$  is used [21, 22, 23]. The ELBO in the CVAE is:

$$\mathcal{L}_{CVAE} = \mathbb{E}_{q_\phi(z|x, c)} [p_\theta(x|z, c)] - D_{KL}(q_\phi(z|x, c)||p_\theta(z|c)) \quad (2)$$

### 3.3. Conditioning only the Prior Distribution

One of the novelties we propose in this work is to condition the prior distribution, but not the encoder and decoder. Moreover, instead of using a neural network to condition this prior, we propose to define it by imposing conditions that will allow us to perform domain translation, as explained in the next two sections. In this case, therefore, we have an encoder  $q_\phi(z|x)$ , a decoder  $p_\theta(x|z)$  and a prior  $p(z|c)$  and the ELBO is:

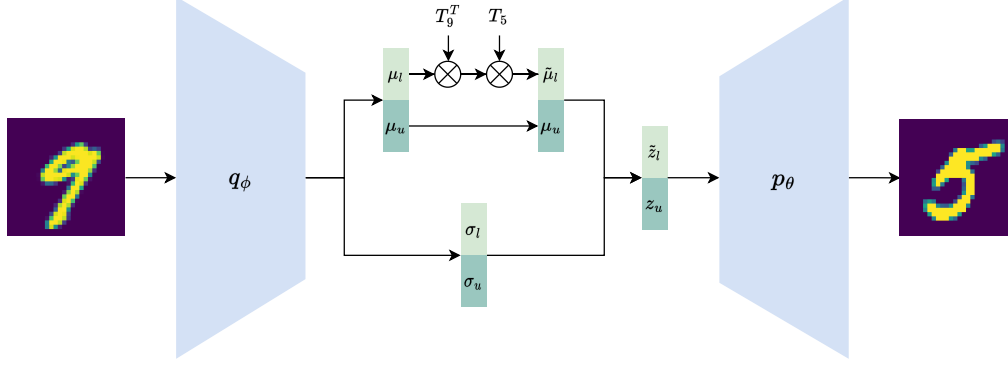
$$\mathcal{L}_{CPVAE} = \mathbb{E}_{q_\phi(z|x)} [p_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p(z|c)) \quad (3)$$

### 3.4. Disentanglement of Labeled and Unlabeled information

The second fundamental novelty that we incorporate in this work is to include two conditionally independent latent variables with respect to the class  $c$ . One of those, denoted  $z_l$ , models the information corresponding to the known information depends on the class. The other one corresponds to all the information that is unlabeled and we denote  $z_u$ . Moreover, since they are conditionally independent, we have that  $p(z_l, z_u|c) = p(z_l|c)p(z_u|c)$ . Also, having imposed that  $z_u$  does not depend on  $c$ , then it remains that  $p(z_l, z_u|c) = p(z_l|c)p(z_u)$ . Finally, we impose that  $q_\phi(z_l, z_u|x) = q_\phi(z_l|x)q_\phi(z_u|x)$ . Under these conditions, we are left with ELBO being:

$$\begin{aligned} \mathcal{L}_{CDVAE} = & \mathbb{E}_{q_\phi(z_l, z_u|x)} [p_\theta(x|z_l, z_u)] \\ & - D_{KL}(q_\phi(z_l|x)||p(z_l|c)) \\ & - D_{KL}(q_\phi(z_u|x)||p(z_u)) \end{aligned} \quad (4)$$

<sup>1</sup>[https://github.com/antonioalmudevar/variational\\_translation](https://github.com/antonioalmudevar/variational_translation)



**Fig. 2:** Scheme of how an input image  $x$  whose class is  $c$  is translated to class  $t$ . The original input is first encoded obtaining  $p(z_l)$  and  $q_\phi(z_u)$ . Then, the mean of  $q_\phi(z_l)$  is multiplied by  $T_c^T$  and  $T_t$  to obtain the mean of  $q_\phi(\tilde{z}_l)$ . Finally, a sample from this last distribution and another from  $q_\phi(z_u)$  are decoded to obtain the translated version of the input. In this figure the input class  $c$  is 9 and the target class  $t$  is 5.

It's worth noting that during training, we maximize a modified version of the ELBO, in which both KL divergence terms are multiplied by a  $\beta$  factor. In addition, multiple labeled latent variables could be defined, each conditionally independent of the others, in order to modify multiple input attributes. However, this is beyond the scope of the present work.

Following the prevalent approach in VAEs, we define  $q_\phi(z_l|x) = \mathcal{N}(\mu_\phi^l, \text{diag}(\sigma_\phi^l))$  and  $q_\phi(z_u|x) = \mathcal{N}(\mu_\phi^u, \text{diag}(\sigma_\phi^u))$ . This formulation ensures that  $z_l$  and  $z_u$  exhibit conditional independence with respect to  $x$ .

### 3.5. Defining prior distributions

The last novelty is the way we define the priors, explained in the previous section. Defining these priors properly will allow us to perform domain translation. To do so, the first step is to define a vector  $\mu_0$ . We do this randomly so that each of its elements follows a  $\mathcal{U}(0, 1)$  distribution. Subsequently, we define a rotation matrix  $T_c$  for each class  $c = 1, 2, \dots, C$ . Each of these rotation matrices  $T_c$  is the normalized version of the Q matrix of a QR factorization of a random matrix in which each element follows a uniform distribution  $\mathcal{U}(0, 1)$ . Finally, we define  $\mu_c = T_c \cdot \mu_0$ ,  $p(z_l|c) = \mathcal{N}(\mu_c, I)$ ,  $c = 1, 2, \dots, C$  and  $p(z_u) = \mathcal{N}(0, I)$ . We should note that  $\mu_0 = T_c^{-1} \cdot \mu_c = T_c^T \cdot \mu_c$ , since  $T_c$  is a rotation matrix and therefore orthogonal.

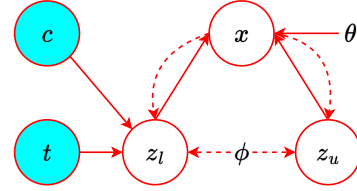
### 3.6. Translating the domain

Once the model is trained, we can perform the domain translation. For this, we have an input  $x$ , its class  $c$ , and a target class  $t$ . First, we obtain  $z_l \sim q_\phi(z_l|x)$  and  $z_u \sim q_\phi(z_u|x)$ . Subsequently, we obtain the rotated version of  $z_l$  as  $\tilde{z}_l = T_t \cdot \tilde{z}_0$ , where  $\tilde{z}_0 = T_c^T \cdot z_l$ . Finally, we obtain the translated version of the input as  $\tilde{x} \sim p_\theta(x|\tilde{z}_l, z_u)$ . We show the scheme of the proposed method and the graphical model of this domain translation in figures 2 and 3, respectively.

## 4. RESULTS

### 4.1. Datasets and Experiments Description

To test the performance of our proposal we used three datasets. The well-known MNIST [24] and SVHN [25] and, in addition, Cars3D [26]. The latter contains images of 199 cars rendered at 24 equispaced angles. We randomly chose 796 images for the test and the



**Fig. 3:** Graphical model of the translation method. Dashed lines correspond to the encoding processes  $q_\phi(z_l|x)$  and  $q_\phi(z_u|x)$ . Solid lines correspond to the decoding process in which the target class  $t$  is used to modify the latent variable  $z_l$  before getting  $p_\theta(x|z_l, z_u)$ .

remaining 3980 for the train. As for the architectures, we use convolutional layers with a kernel size of 3 and stride of 2. In Table 1 we show the number of channels, layers, and dimensions of the latent variables for each dataset. To train all of them, we used the Adam Optimizer with a batch size of 128 and a learning rate of 0.001 for 100 epochs for MNIST and 500 epochs for SVHN and Cars3D. In all cases,  $\beta = 0.001$  in the modified ELBO that must be maximized.

**Table 1:** Architecture details for each dataset where  $d(z_l)$  and  $d(z_u)$  are the dimensions for  $z_l$  and  $z_u$ , respectively.

	$d(z_l)$	$d(z_u)$	Channels
MNIST	512	512	{16,32,64,128}
SVHN	1024	1024	{32,64,128,256}
Cars3D	1024	1024	{128,128,256,256,512,512}

### 4.2. Domain Translation Results

In the following, we show visually the results provided by our method for each of the datasets. In Figure 4, we randomly select some test samples from the MNIST dataset and compare the outcomes generated by our system with those produced by StarGAN [9], which is the most representative baseline for the task of the present work. In it, we see that the numbers generated by our system are much more realistic, and also that the background is less noisy, as in the original images. Likewise, in Figure 5 we show the results for SVHN. In this case, we see that the style and the color of the digit as well as the background are preserved. In addition, in the

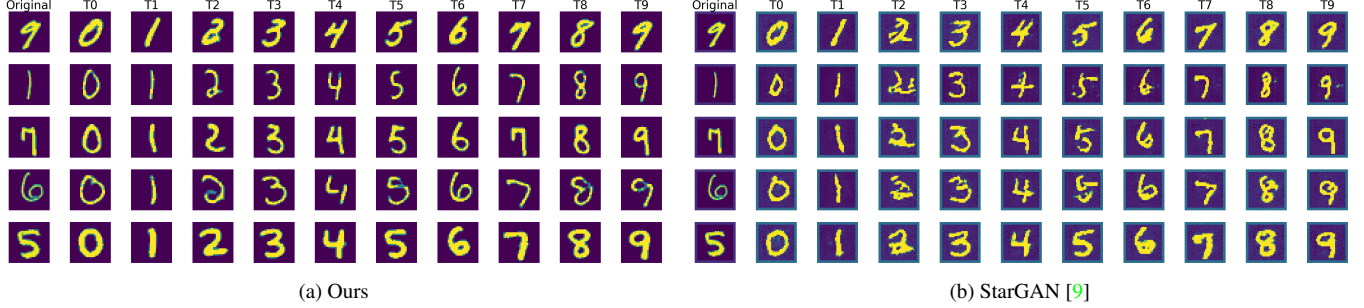


Fig. 4: Comparison between our method and StarGAN for MNIST

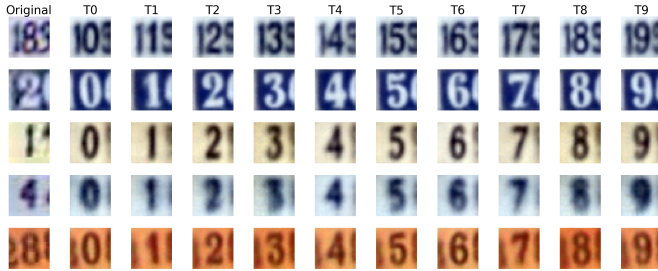


Fig. 5: Results for SVHN

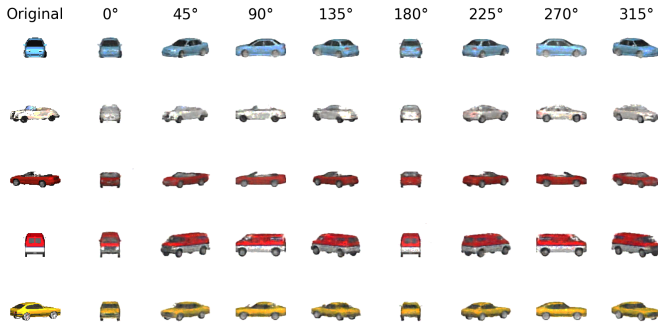


Fig. 6: Results for Cars3D

cases where extra digits appear in the images, these are maintained, modifying only the middle one, which is the one that corresponds to the label during training. Finally, in Figure 6, we present the outcomes for the Cars3D dataset. Here, we observe that the model and color of the car remain consistent, with only the car’s angle being modified. However, in the second-row car, the model fails to generate the correct profile, possibly because it belongs to an uncommon car style in the dataset. For the last two datasets, we refrain from showing results obtained with StarGAN, as the generated images are not realistic.

#### 4.3. Verifying the Disentanglement

Finally, to assess if all the relevant information to identify the class is modeled by  $z_l$  and if  $z_u$  is independent of class information, we build two classifiers, both with a linear layer followed by a softmax. The first one,  $C_l$ , takes as input  $z_l \sim q_\phi(z_l|x)$  and the second one,  $C_u$ , takes as input  $z_u \sim q_\phi(z_u|x)$ . Both aim to determine the class to which  $x$  belongs. The accuracy of  $C_l$  is expected to be very high,

whereas that of  $C_u$  should be nearly equivalent to random classification. In table 2 we show the results of the experiment. We see that, indeed, the accuracy is very high in all cases in  $C_l$ . However, although the accuracy in  $C_u$  is very low, it is not equivalent to a completely random classification for the first two datasets (it should be around 10% in both cases). This may be due either to the fact that in  $z_u$  there is still information about the class, or that there is some correlation between class and style in these datasets. In the Cars3D dataset, we have certainty that this correlation does not exist since all the cars appear at all the angles. Here, the results are similar to a random classification. Therefore, we could think that there is some correlation between class and style in the MNIST and SVHN samples, although it is difficult to verify and remains beyond the scope of this paper.

Table 2: Accuracy (%) for  $C_l$  and  $C_u$  in the different datasets

	MNIST	SVHN	CARS3D
$C_l$	99.29	90.78	95.58
$C_u$	25.29	19.68	3.88

## 5. CONCLUSION

In this paper, we have presented a method to solve the unsupervised multiple-domain translation task using exclusively a modified Variational Autoencoder. This model incorporates two distinct latent variables. One is intentionally designed to be influenced by the domain, while the other is meant to capture dependencies related to all factors except the domain. Another peculiarity of our proposal is that, although one of the latent variables depends on the domain, the encoder and decoder are common to all domains. Therefore, we only need to make transformations in the latent variable to perform domain translation. We define these transformations ourselves and impose that they are linear. This, in addition to simplifying the domain translation, allows us to have control and understanding over the latent space. Concretely, we design this latent space such that to carry out the domain translation we must perform a rotation of the domain latent variable, while the other latent variable must remain intact so that properties not associated with the domain are not modified.

We have verified that our proposal satisfactorily solves the objective task for different datasets outperforming other well-known works which, in addition, have higher complexity. Finally, we have verified that the two latent variables have a high degree of disentanglement, so that the one associated with the domain contains all the relevant information about the domain, while the other one hardly contains any information about the domain.

## 6. REFERENCES

- [1] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, “Generative adversarial nets,” *Advances in neural information processing systems*, vol. 27, pp. 2672–2680, 2014.
- [2] Michael F Mathieu, Junbo Jake Zhao, Junbo Zhao, Aditya Ramesh, Pablo Sprechmann, and Yann LeCun, “Disentangling factors of variation in deep representation using adversarial training,” *Advances in neural information processing systems*, vol. 29, 2016.
- [3] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [4] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.
- [5] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong, “Dualgan: Unsupervised dual learning for image-to-image translation,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2849–2857.
- [6] Ming-Yu Liu, Thomas Breuel, and Jan Kautz, “Unsupervised image-to-image translation networks,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- [7] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz, “Multimodal unsupervised image-to-image translation,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.
- [8] Abel Gonzalez-Garcia, Joost Van De Weijer, and Yoshua Bengio, “Image-to-image translation for cross-domain disentanglement,” *Advances in neural information processing systems*, vol. 31, 2018.
- [9] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo, “Stargan: Unified generative adversarial networks for multi-domain image-to-image translation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [10] Yunjey Choi, Youngjung Kim, Minjae Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo, “Stargan v2: Diverse image synthesis for multiple domains,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [11] Diederik P. Kingma and Max Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.
- [12] Mehdi Mirza and Simon Osindero, “Conditional generative adversarial nets,” *arXiv preprint arXiv:1411.1784*, 2014.
- [13] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, and Shakir Mohamed, “Understanding disentangling in  $\beta$ -vae,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.
- [14] Hyunjik Kim and Andriy Mnih, “Disentangling by factorising,” in *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.
- [15] Shengjia Zhao, Jiaming Song, and Stefano Ermon, “Variational inference of disentangled latent concepts from unlabeled observations,” in *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.
- [16] Diederik P Kingma, Danilo J Rezende, Shakir Mohamed, and Max Welling, “Semi-supervised learning with deep generative models,” *Advances in Neural Information Processing Systems*, 2014.
- [17] Hyunjik Kim and Andriy Mnih, “Disentangling by factorising,” in *International Conference on Machine Learning*. PMLR, 2018, pp. 2649–2658.
- [18] Javier Antoran and Antonio Miguel, “Disentangling and learning robust representations with natural clustering,” in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. IEEE, 2019, pp. 694–699.
- [19] Kihyuk Sohn, Honglak Lee, and Xinchen Yan, “Learning structured output representation using deep conditional generative models,” *Advances in neural information processing systems*, vol. 28, 2015.
- [20] Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi, “Learning discourse-level diversity for neural dialog models using conditional variational autoencoders,” *arXiv preprint arXiv:1703.10960*, 2017.
- [21] Jaechang Lim, Seongok Ryu, Jin Woo Kim, and Woo Youn Kim, “Molecular generative model based on conditional variational autoencoder for de novo molecular design,” *Journal of cheminformatics*, vol. 10, no. 1, pp. 1–9, 2018.
- [22] Adrian Alan Pol, Victor Berger, Cecile Germain, Gianluca Cermignani, and Maurizio Pierini, “Anomaly detection with conditional variational autoencoders,” in *2019 18th IEEE international conference on machine learning and applications (ICMLA)*. IEEE, 2019, pp. 1651–1657.
- [23] Mathis Petrovich, Michael J Black, and Gül Varol, “Action-conditioned 3d human motion synthesis with transformer vae,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10985–10995.
- [24] Yann LeCun, Corinna Cortes, and Christopher J. Burges, “The mnist database of handwritten digits,” *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 237–257, 1998.
- [25] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng, “The street view house numbers (svhn) dataset,” *arXiv preprint arXiv:1202.2745*, 2013.
- [26] Scott Reed, Yi Zhang, Yuting Zhang, and Honglak Lee, “Deep visual analogy-making,” in *NIPS*, 2015.