DISENTANGLING LATENT GROUPS OF FACTORS

Nakamasa Inoue[1*], *Ryota Yamada*[1*], *Rei Kawakami*[1,2], *Ikuro Sato*[1,2]

[1]Tokyo Institute of Technology, [2]Denso IT Laboratory, Inc.

ABSTRACT

This paper proposes a framework for training variational autoencoders (VAEs) for image distributions that have latent groups of factors. Our key idea is to introduce a mechanism to predict the factor group an image belongs to while simultaneously disentangling factors in it. More specifically, we propose an architecture consisting of three components: an encoder, a decoder, and a factor-group prediction header. The first two components are trained with a VAE objective, and the last one is trained with the proposed algorithm using the loss of unsupervised contrastive learning. In experiments, we designed a task in which more than one group of factors were entangled by combining multiple datasets and demonstrated the effectiveness of the proposed framework. The Mutual Information Gap score was improved from 0.089 to 0.125 on a merged dataset of Color-dSprites, 3DShapes, and MPI3D.

***Index Terms*—** Variational autoencoders, Disentangling factors, Metric learning, Unsupervised contrastive learning.

1. INTRODUCTION

Learning disentangled image representations is a challenging subject, since a naive way of modeling data, say by an autoencoder, easily results in a highly complex semantic structure in the latent space, making human interpretation unfeasible. In the past few years, great progress has been achieved by variational autoencoders (VAEs), which learn the encoding and decoding distributions of images [12]. Researchers have shown that VAE and its variants have decent ability to disentangle factors of variations, each of which corresponds to an interpretable change, such as in the object size, in a given image dataset [10, 24, 11, 3]. There also have been many efforts to develop applications of VAEs for semantic segmentation [14], action recognition [25], and 3D morphing [22].

A real-world unlabeled dataset often contains multiple domains, and data in each domain share a group of latent factors. We refer to a set of factors shared in a given domain as a *factor group*. This work aims to solve the problem of isolating latent variables, each of which corresponds to a particular factor when the training set contains more than one factor group, in such a way that the disentanglement measure, called Mutual Information Gaps [4], averaged over all the factor groups is large.[1] The problem setting is depicted in Fig. 1. Although VAE is obviously a candidate method for solving this problem, it is still unclear how it performs beyond a single domain. Note that here we assume the domain labels are unavailable but the same set of domains are given at test time, making the problem setting different from those dealt with by domain agnostic learning [19] and by life-long disentangled representation learning [2].

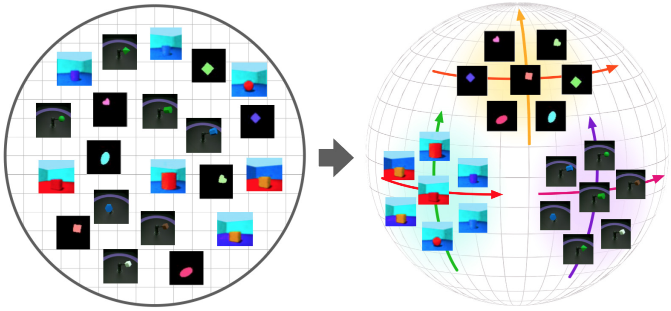


Fig. 1. The setting considered in this paper, where the distribution of images has latent groups of factors. The goal is to generate a latent space so that a latent variable corresponds to a particular factor for any given factor group.

The key idea of our framework is to predict the factor group an image belongs to while simultaneously disentangling factors in it. This is achieved by constraining the disentangled factors to better predict the factor groups. More specifically, an architecture consisting of three components is proposed: an encoder, a decoder, and a factor-group prediction (FGP) header. The first two are trained with a VAE objective, and the last one is trained with unsupervised contrastive learning. In experiments, we designed a task in which more than one group of factors are entangled by combining multiple datasets and demonstrated the effectiveness of the proposed framework. In summary, our contributions are twofold.

1. We propose an extended VAE architecture that contains a Factor-Group Prediction (FGP) header to encourage representation disentanglement for given groups of latent factors underlying the input variations.
2. We propose a learning algorithm for the extended architecture by leveraging unsupervised contrastive learning and the mini-batch k-means algorithms on top of the ordinary VAE learning.

[1] Even if two or more domains have the same type of factor, we treat them as separate factors.

2. RELATED WORK

2.1. Variational Autoencoders

VAEs are versatile generative models. The first VAE was proposed in [12] for image generation. As VAEs have the ability to disentangle factors, they play an important role in research on latent semantics in images. Researchers have proposed various types of VAE, such as β-VAE [10], InfoVAE [24], FactorVAE [11], DIP-VAE [13], and β-TCVAE [3]. Among them, FactorVAE is known to be effective at disentangling factors of variation over images.

Some recent studies have focused on VAEs that work in different domains. Examples include VASE for life-long learning [2] and DADA [19] for domain adaptation. These methods assume that the division of domains is known in advance. In contrast, this paper focuses on the case where multiple groups of factors are latent.

2.2. Metric Learning and Self-Supervised Learning

Metric learning is a framework for learning a metric space and is typically implemented in a supervised manner. A recent trend for this is to introduce a metric-based loss in a neural network. For example, CosFace [23], ArcFace [6], and SphereFace [15] use the cosine similarity as their loss. These networks are effective for face verification and object recognition. Musgrave *et al.* [17] compared the performances of metric learning methods including losses from traditional contrastive loss [8] with the recent SoftTriplet loss [20]. The results show that although each loss has its advantages and disadvantages, CosFace and ArcFace are generally stable across various training conditions.

Self-supervised learning is a framework for learning data representations without the need for annotated information. In this type of learning, neural networks are trained on pretext tasks, such as Jigsaw [18]. Unsupervised contrastive learning has received attention for its high performance on many image-recognition tasks. Specifically, SimCLR [5] and MoCo [9] have demonstrated state-of-the-art performance.

3. PROPOSED METHOD

This section presents the proposed framework for disentangling latent groups of factors. We first give a preliminary review of FactorVAE [11] on the assumption that the distribution of images has a single set of factors F. We then propose a VAE architecture with an FGP header for cases where the distribution of images has multiple groups of factors $F_1, F_2, \cdots, F_K$ $(K > 1)$.

3.1. Preliminary: Disentangling Factors

Let X be a set of images, which has d factors of variation, and denote the set of factors as $F = \{c_1, c_2, \cdots, c_d\}$. The goal here is to disentangle the factors, *i.e.*, to find a mapping from X to $Z = \mathbb{R}^d$, where $\boldsymbol{z} \in Z$ is a representation of an image $\boldsymbol{x} \in X$.

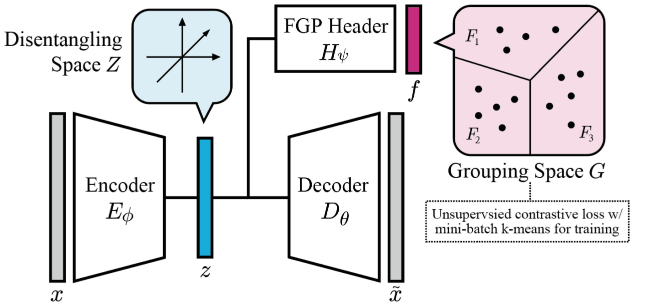


Fig. 2. Proposed VAE structure. The factor-group prediction (FGP) header is trained via supervised metric learning or unsupervised contrastive learning.

FactorVAE predisposes the distribution of representations $q(\boldsymbol{z})$ to be factorial by maximizing the following objective function:

$$\mathcal{O}_{\text{F-VAE}} = \mathbb{E}_{p(\boldsymbol{x})}\left[\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log p(\boldsymbol{x}|\boldsymbol{z})\right] - \mathrm{KL}(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z}))\right] + \gamma \mathrm{KL}(q(\boldsymbol{z})||\bar{q}(\boldsymbol{z})), \quad (1)$$

where $q(\boldsymbol{z}|\boldsymbol{x})$ is the encoding distribution, $p(\boldsymbol{x}|\boldsymbol{z})$ is the decoding distribution, $p(\boldsymbol{x})$ is the data distribution, $p(\boldsymbol{z})$ is a normal distribution, and KL is the Kullback–Leibler divergence. Note that the encoder $E_\phi : X \to Z$ and decoder $D_\theta : Z \to X$ have learnable parameters ϕ and θ. The last term in Eq. (1) is the total correlation, which encourages $\boldsymbol{z}$ to be factorial by maximizing the KL divergence of $q(\boldsymbol{z})$ from $\bar{q}(\boldsymbol{z}) = \prod_{j=1}^{d} q(z_j)$.

3.2. Disentangling Latent Groups of Factors

We will focus on the case where X has multiple groups of factors $F_1, F_2, \cdots, F_K$, under the assumption that each image $\boldsymbol{x} \in X$ corresponds to one of F_k. This setting requires us to solve the following two problems:

(P1) Finding F_k to which the image $\boldsymbol{x}$ belongs, and
(P2) Disentangling the factors in F_k.

To solve these two problems simultaneously, we introduce two vector spaces, namely a disentangling space $Z = \mathbb{R}^d$ and a grouping space $G = \mathbb{R}^{d'}$. The former is a latent space in which the factors of variation are disentangled. This is the same as the latent space Z for the vanilla VAE. The latter is a space to distinguish factor groups. This space has prototype vectors of factor groups, by which each factor group F_k is represented by a vector $\boldsymbol{f}_k \in G$. The input image $\boldsymbol{x}$ is also embedded into this space, and its factor group is estimated by finding the nearest-neighbor factor group in G as $k^* = \mathrm{argmin}_k\, d(\boldsymbol{h}, \boldsymbol{f}_k)$, where $\boldsymbol{h} \in G$ is an embedding of

$\boldsymbol{x}$, and $d(\cdot,\cdot)$ is the distance metric in G. Note that the set of prototype vectors $\{\boldsymbol{f}_k\}_{k=1}^K$ and the metric $d(\cdot,\cdot)$ are learned from training samples. In summary, (P1) and (P2) are solved in G and Z, respectively.

Figure 2 shows the proposed architecture to learn the two vector spaces. It consists of three components: an encoder $E_\phi : X \to Z$, a decoder $D_\theta : Z \to X$, and a factor-group prediction (FGP) header $H_\psi : Z \to G$. The first two components are from a VAE. The last component, the FGP header, is a network for mapping the encoder output $\boldsymbol{z} \in Z$ into the grouping space G. The total objective is given by

$$\mathcal{O} = \mathcal{O}_{\text{F-VAE}} - \lambda \mathcal{L}, \tag{2}$$

where $\mathcal{O}_{\text{F-VAE}}$ is the FactorVAE objective in Eq. (1), $\mathcal{L}$ is the loss for the FGP header, and λ is a hyper-parameter. In the following, we present two learning methods, each with its own definition of $\mathcal{L}$.

(i) Unsupervised Learning for the FGP header. In the proposed framework described above, two items must be learned from a training set of images: (i) the distance metric $d(\cdot,\cdot)$ in G, and (ii) the factor group prototypes $\{\boldsymbol{f}_k\}_{k=1}^K \subset G$. The proposed algorithm incorporates unsupervised contrastive learning into mini-batch k-means clustering. The algorithm consists of the following four steps.

1) Randomly initialize $\{\boldsymbol{f}_k\}_{k=1}^K$ using a normal distribution.
2) Draw a minibatch of images and update the parameters of the encoder E_ϕ, the decoder D_θ and the FGP header H_ψ by using a stochastic optimizer with the loss in Eq. (2). For the FGP header, the following loss is used:

$$\mathcal{L} = \mathbb{E}_{p(\boldsymbol{x})}\left[-\log \frac{e^{\tau \operatorname{sim}(\boldsymbol{h},\boldsymbol{h}')}}{\sum_{\boldsymbol{r}\in B\setminus\{\boldsymbol{h}\}} e^{\tau \operatorname{sim}(\boldsymbol{h},\boldsymbol{r})}}\right], \tag{3}$$

where $\boldsymbol{h}$ is the embedding of an image $\boldsymbol{x}$ obtained by $\boldsymbol{h} = H_\psi(E_\phi(\boldsymbol{x}))$, $\boldsymbol{h}' = H_\psi(E_\phi(\boldsymbol{x}'))$ is the embedding of the augmented image $\boldsymbol{x}' = t(\boldsymbol{x})$, t is an augmentation function, $\operatorname{sim}(\boldsymbol{a},\boldsymbol{b}) = \boldsymbol{a}^T\boldsymbol{b}/(\|\boldsymbol{a}\|\|\boldsymbol{b}\|)$, is the cosine similarity between two vectors in G, τ is a hyper-parameter, and B is the set of embeddings and augmented ones (*i.e.*, B consists of $\boldsymbol{h}$ and $\boldsymbol{h}'$ of all images in the minibatch).
3) Update $\{\boldsymbol{f}_k\}_{k=1}^K$ by applying mini-batch k-means clustering [21] to B.
4) Repeat 2 and 3 until the loss converges.

In Eq. (3), the similarity between $\boldsymbol{h}$ and the augmented $\boldsymbol{h}'$ is maximized. This works because if an image $\boldsymbol{x}$ belongs to the factor group F_k, the augmented image $\boldsymbol{x}'$ also belongs to the same factor group. As a result, the loss helps to learn a reasonable metric in G without using ground-truth labels. It is worth noting that this part can be viewed as being in a family of self-supervised contrastive learnings for image classification, *e.g.*, [5].

(ii) Supervised Metric Learning for the FGP header. We here describe how to learn the FGP header under the assumption that ground-truth factor-group labels are given for all training samples. This is a reference model to measure the upper bound performance.

Let $\mathcal{D}$ be a training dataset, which consists of pairs $(\boldsymbol{x}, y)$ of an image $\boldsymbol{x}$ and its factor-group label $y \in \{1, 2, \cdots, K\}$. By absorbing all $\boldsymbol{f}_k$ into learnable parameters, the FGP header can be learned by applying metric learning. Specifically, we employ CosFace [23] as the loss function, which is given by

$$\mathcal{L}{=}\mathbb{E}_{p(\boldsymbol{x})}\left[-\log \frac{e^{\tau(\operatorname{sim}(\boldsymbol{h},\boldsymbol{f}_y)-m)}}{e^{\tau(\operatorname{sim}(\boldsymbol{h},\boldsymbol{f}_y)-m)} + \sum_{k\neq y} e^{\tau \operatorname{sim}(\boldsymbol{h},\boldsymbol{f}_k)}}\right], \tag{4}$$

where $\boldsymbol{f}_k$ is a learnable vector of F_k, and τ, m are hyper-parameters. Note that Eq. (3) uses the cosine similarity instead of a distance metric, but this is equivalent to learning of the Euclidean distance metric $d(\cdot,\cdot)$ on a L_2 normalized vector subspace in G.

4. EXPERIMENTS

4.1. Datasets and Evaluation Measures

We demonstrated the effectiveness of the proposed framework in a setting with entangled latent groups of factors. This setting was created by merging multiple datasets, each of which had a single set of factors. Specifically, we used combinations of the following datasets.

MPI3D dataset [7][2] This dataset consists of 1,036,800 images of a robotic arm carrying an object. It has 7 ground-truth latent factors: *object-color, object-shape, object-size, camera-height, background-color, horizontal-axis,* and *vertical-axis*.

3DShapes dataset [11] This dataset consists of 480,000 images of 3D shapes generated from 6 ground-truth latent factors: *object-color, shape, floor-color, wall-color, orientation,* and *scale*.

Color-dSprites [10] This dataset consists of 737,280 images of 2D shapes, generated from 5 ground-truth latent factors: *shape, scale, rotation, x-position,* and *y-position*.

From these three datasets, we made four combined datasets: (1) dSprites+3DShapes, (2) 3DShapes+MPI3D, (3) MPI3D+ dSprites, and (4) All three. The original image size in all datasets is 64x64 pixels; thus, each combined dataset was made by simple dataset merging. The evaluation measure was the Mutual Information Gap (MIG) [4].

4.2. Implementation Details

We used the PyTorch VAE implementation in [1] with the evaluation metric implementation in [16], which provides implementations of recent models including β-VAE, FactorVAE, and β-TCVAE. The backbone architecture and all VAE hyper-parameters were the same as in [1]; *i.e.*, the encoder

[2] https://github.com/google-research/disentanglement_lib

Table 1. Comparison of MIG score on four combined datasets (higher is better). "Unsupervised" and "Supervised" use losses of unsupervised contrastive learning in Sec 3.2-i and supervised metric learning in Sec 3.2-ii, respectively.

Method	MPI3D + 3DShapes	MPI3D + dSprites	3DShapes + dSprites	Combination of all three
FactorVAE (baseline)	0.183 ± 0.043	0.069 ± 0.030	0.109 ± 0.030	0.089 ± 0.032
Ours (Unsupervised)	0.230 ± 0.075	0.086 ± 0.036	0.113 ± 0.068	0.125 ± 0.044
Ours (Supervised)	0.230 ± 0.062	0.099 ± 0.040	0.155 ± 0.062	0.090 ± 0.023

Table 2. Comparison with other methods. MIG scores on individual test datasets are reported. Models are trained jointly on three datasets. The last row shows scores obtained by FactorVAEs trained on individual datasets (*i.e.*, three FactorVAEs are independently trained).

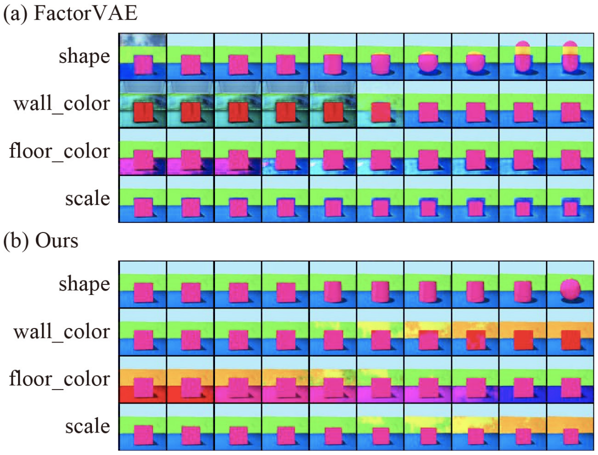
Method	MPI3D	3DShapes	dSprites	Average
β-TCVAE [3]	0.082	0.218	0.050	0.116
DIP-VAE [13]	0.025	0.218	0.022	0.088
InfoVAE [24]	0.043	0.161	0.080	0.090
FactorVAE [11]	0.044	0.194	0.029	0.089
Ours	0.061	0.239	0.075	0.125
FactorVAE (Individual)	0.256	0.422	0.045	-

consisted of 5 conv layers (with strides of 2 and kernel sizes of 3x3), and the decoder consisted of 6 transposed conv layers (with strides of 2 with kernel sizes of 4). ReLU activation was applied to all layers. The dimension of the latent variables was set to 20. The FGP header consisted of two blocks, each with a fully-connected layer (both input and output sizes are 20), batch normalization, and ReLU. All models were trained with the Adam optimizer for 90k iterations. Hyperparameters were as follows: the mini-batch size was 64, $\tau = 10.0$, $m = 0.1$, and $\lambda = 100.0$. For mini-batch k-means, the scikit-learn implementation was used.

4.3. Results

Table 1 compares the performance in the case of optimizing multiple domains jointly. It shows that the proposed unsupervised method outperforms the FactorVAE baseline for any combination of domains. Interestingly, our unsupervised method shows comparable or just slightly worse results compared to the one with supervision. This confirms the effectiveness of the proposed unsupervised learning algorithm.

Table 2 analyzes the MIG scores on each test dataset, using the models trained on 3 datasets jointly as well as in comparison with other VAEs. The results indicate that the proposed method performs the best, consistently outperforming FactorVAE. Compared with the FactorVAE trained on just a single dataset (the last row), the baseline (trained with 3 datasets jointly, appearing in the top row) shows totally degraded performance, whereas our model performs better on one of the datasets. What we have learnt here is that disentanglement of multiple domains is generally more difficult than that of a single domain; nevertheless, our method partly succeeds in overcoming this difficulty. These results indicate that the FGP header for distinguishing factor groups helps to disentangle latent factors.


(a) FactorVAE

(b) Ours

Fig. 3. Qualitative results of disentanglement performance. Models are trained on MPI3D+3DShapes jointly. 3DShapes images are shown. Each row corresponds to the latent variable that most represents the factor, and the x-axis represents the changes in the latent variables.

Figure 3 shows a qualitative comparison of latent traversals. The proposed framework mostly (but not perfectly) disentangles the factors of *shape, wall_color, floor_color* and *scale* on 3DShapes. The baseline exhibits somewhat similar or slightly worse behavior; however, it shows sign of entanglement with other datasets, unlike the proposed method.

5. CONCLUSION

This paper proposed a framework for disentangling latent groups of factors, which introduces an FGP header into VAEs. The FGP header is trained in an unsupervised manner using contrastive learning and mini-batch k-means clustering. In experiments, the effectiveness of the proposed framework was demonstrated on combinations of Color-dSprites, 3DShapes, and MPI3D datasets. Our future work will involve applying this technology to image retrieval on a realistic dataset.

Acknowledgement. This work was supported by DENSO IT LAB Recognition and Learning Algorithm Collaborative Research Chair (Tokyo Tech.), JST CREST Grant Number JPMJCR1687, and JST ACT-X Grant Number JPMJAX1905, Japan.

6. REFERENCES

[1] A. H. Abdi, P. Abolmaesumi, and S. Fels. Variational learning with disentanglement-pytorch. In *NeurIPS Disentanglement Challenge*, 2019.

[2] A. Achille, T. Eccles, L. Matthey, C. P. Burgess, N. Watters, A. Lerchner, and I. Higgins. Life-long disentangled representation learning with cross-domain latent homologies. In *NeurIPS*, 2018.

[3] R. T. Q. Chen, X. Li, R. Grosse, and D. Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *NeurIPS*, 2018.

[4] R. T. Q. Chen, X. Li, R. Grosse, and D. Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *NeurIPS*, 2018.

[5] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[6] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, 2019.

[7] M. W. Gondal, M. Wüthrich, D. Miladinović, F. Locatello, M. Breidt, V. Volchkov, J. Akpo, O. Bachem, B. Schölkopf, and S. Bauer. On the transfer of inductive bias from simulation to the real world: a new disentanglement dataset. In *NeurIPS*, 2019.

[8] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006.

[9] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[10] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. Beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

[11] H. Kim and A. Mnih. Disentangling by factorising. In *ICML*, 2018.

[12] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[13] A. Kumar, P. Sattigeri, and A. BalakrishnanRicky. Variational inference of disentangled latent concepts from unlabeled observations. In *ICLR*, 2018.

[14] Z. Li, R. Togo, T. Ogawa, and M. Haseyama. Variational autoencoder based unsupervised domain adaptation for semantic segmentation. In *ICIP*, 2020.

[15] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*, 2017.

[16] F. Locatello, S. Bauer, M. Lucic, G. Rätsch, S. Gelly, B. Schölkopf, and O. Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *ICML*, 2019.

[17] K. Musgrave, S. Belongie, and S.-N. Lim. A metric learning reality check. In *ECCV*, 2020.

[18] M. Noroozi and P. Favaro. Unsupervised learning of visual represenions by solving jigsaw puzzles. In *ECCV*, 2016.

[19] X. Peng, Z. Huang, X. Sun, and K. Saenko. Domain agnostic learning with disentangled representations. In *ICML*, 2019.

[20] Q. Qian, L. Shang, B. Sun, J. Hu, H. Li, and R. Jin. Softtriple loss: Deep metric learning without triplet sampling. In *ICCV*, 2019.

[21] D. Sculley. Web-scale k-means clustering. In *WWW*, 2010.

[22] N. Vercheval, H. D. Bie, and A. Pižurica. Variational auto-encoders without graph coarsening for fine mesh learning. In *ICIP*, 2020.

[23] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu. Cosface: Large margin cosine loss for deep face recognition. In *CVPR*, 2018.

[24] S. Zhao, J. Song, and S. Ermon. Infovae: Balancing learning and inference in variational autoencoders. In *AAAI*, 2019.

[25] J. Zhou and T. Komuro. Recognizing fall actions from videos using reconstruction error of variational autoencoder. In *ICIP*, 2019.