Pattern Recognition 156 (2024) 110792

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

EdVAE: Mitigating codebook collapse with evidential discrete variational autoencoders

Gulcin Baykal [a,b,*], Melih Kandemir [b], Gozde Unal [c]

[a] Department of Computer Engineering, Istanbul Technical University, Istanbul, Turkiye
[b] Department of Mathematics and Computer Science, University of Southern Denmark, Odense, Denmark
[c] Department of AI and Data Engineering, Istanbul Technical University, Istanbul, Turkiye

ARTICLE INFO

Keywords:
Vector quantized variational autoencoders
Discrete variational autoencoders
Evidential deep learning
Codebook collapse

ABSTRACT

Codebook collapse is a common problem in training deep generative models with discrete representation spaces like Vector Quantized Variational Autoencoders (VQ-VAEs). We observe that the same problem arises for the alternatively designed discrete variational autoencoders (dVAEs) whose encoder directly learns a distribution over the codebook embeddings to represent the data. We hypothesize that using the softmax function to obtain a probability distribution causes the codebook collapse by assigning overconfident probabilities to the best matching codebook elements. In this paper, we propose a novel way to incorporate evidential deep learning (EDL) through a hierarchical Bayesian modeling instead of softmax to combat the codebook collapse problem of dVAE. We evidentially monitor the significance of attaining the probability distribution over the codebook embeddings, in contrast to softmax usage. Our experiments using various datasets show that our model, called EdVAE, mitigates codebook collapse while improving the reconstruction performance, and enhances the codebook usage compared to dVAE and VQ-VAE based models. Our code can be found at https://github.com/ituvisionlab/EdVAE.

1. Introduction

Generative modeling of images has been one of the popular research themes that aided in advancement of deep neural networks, particularly in enhancement of unsupervised learning models like Variational Autoencoders (VAEs) [1], Generative Adversarial Networks (GANs) [2], and Diffusion models [3]. VAEs [1] and their variant models have shown promise as solutions to numerous problems in generative modeling such as disentanglement of the representations [4], discretization of the representations [5,6], and high-quality image generation [7,8]. Although most of the VAE models assume a continuous latent space to represent the data, discrete representations are more suitable to express categories that modulate the observation space [5]. Supporting this rationale, recent celebrated large generative models like VQGAN [9], LDM [10], and DALL-E 1 [6] also rely on discrete latent spaces learned by discrete VAEs to describe the image data.

It is customary to form the latent space as a *codebook* consisting of discrete latent embeddings, where those embeddings are learned to represent the data. VQ-VAE [5,11] and its variants [9,12] are discrete VAEs that quantize the encoded representation of the data by an encoder with the nearest latent embedding in the learnable codebook in a deterministic way. VQ-VAEs achieve considerably high reconstruction and generation performances. However, they are observed to suffer from the *codebook collapse* problem defined as the under-usage of the codebook embeddings, causing a redundancy in the codebook and limiting the expressive power of the generative model. As the deterministic quantization is the most likely cause of the codebook collapse problem in VQ-VAEs [13], probabilistic approaches [13], optimization changes [14] as well as codebook reset [15,16] and hyperparameter tuning [16] are employed in VQ-VAEs to combat the codebook collapse problem.

Unlike VQ-VAEs, the encoder of another discrete VAE, which is called dVAE [6], learns a *distribution* over the codebook embeddings for each latent in the representation. That means, instead of quantizing a latent with a single, deterministically selected codebook embedding, the encoder of dVAE incorporates stochasticity to the selection of the embeddings where the learned distribution is modeled as a *Categorical* distribution. We find out that dVAE also suffers from the codebook collapse problem even though stochasticity is involved. One of our hypotheses is that attaining the probability masses for codebook embeddings using a softmax function induces a codebook collapse in dVAE, as we demonstrate in this paper.

* Corresponding author at: Department of Computer Engineering, Istanbul Technical University, Istanbul, Turkiye.
E-mail addresses: baykalg@itu.edu.tr (G. Baykal), kandemir@imada.sdu.dk (M. Kandemir), gozde.unal@itu.edu.tr (G. Unal).

https://doi.org/10.1016/j.patcog.2024.110792
Received 8 December 2023; Received in revised form 3 July 2024; Accepted 12 July 2024
Available online 24 July 2024

Softmax notoriously overestimates the probability mass of the prediction, which in turn led to exploration of different alternatives to softmax, especially in classification tasks [17,18]. Among those approaches, the widely adopted EDL [18] places a Dirichlet distribution whose concentration parameters over the class probabilities are learned by the encoder. In EDL, the class predictions are considered as the subjective opinions, and the evidences leading to those opinions are collected from the data, which are explicitly used as the concentration parameters of the Dirichlet distribution. To define such a framework, the softmax layer of the encoder is removed, and the logits of the encoder are used as the concentration parameters whose mean values are used as the predicted class probabilities. EDL can be also viewed as a generative model where the class labels follow a normal distribution whose mean is set by the uninformative Dirichlet prior over the class probabilities [19].

In this work, we find out that the root cause of the codebook collapse in dVAE can be framed as the artificial intelligence counterpart of the cognitive psychological phenomenon called the *confirmation bias* [20]. This bias is developed cumulatively during the whole training process as the model overconfidently relates new observations to those already seen ones. We demonstrate by way of experiments that the spiky softmax probabilities lead to the confirmation bias problem. In order to mitigate the latter, we introduce an uncertainty-aware mechanism to map the inputs to the codebook embeddings by virtue of an evidential formulation. To that end, dVAE encoder collects evidences from the data, and the codebook embeddings that represent the data are selected based on those evidences. While the highest evidence increases the probability of the corresponding embedding to be selected, the collected evidences lead to a smoother probability distribution compared to softmax probabilities which leads to a diversified codebook usage. We reformulate the original EDL framework, and build a hierarchical Bayesian extension of dVAE. We summarize our contributions as follows: (1) We introduce an original extension of dVAE that is a hierarchical Bayesian model using Dirichlet-Categorical distributions by virtue of EDL incorporation. (2) We report evidence of the confirmation bias problem caused by the softmax probabilities used in dVAE. (3) We observe that the EDL integration improves the codebook usage of dVAE, which is measured by the perplexity.

In our work, we set the baseline results of dVAE for various datasets, and surpass the baseline measures with our model called Evidential dVAE (EdVAE) in terms of reconstruction performance, codebook usage, and image generation performance in most of the settings. We also compare EdVAE with state-of-the-art VQ-VAE based models using various experimental settings to demonstrate that our model performs close to or better than the VQ-VAE based methods.

2. Related work

Codebook Collapse on VQ-VAEs: Vector quantization, which is useful for various tasks including image compression [21,22], is the backbone of the VQ-VAE [5]. While deterministically trained VQ-VAEs show favorable performance on image reconstruction and generation [9,11], text decoding [23], music generation [16], and motion generation [24], some of the VQ-VAE variants use stochasticity and other tricks during the training, especially to alleviate the codebook collapse problem.

Codebook reset trick [15] replaces the unused codebook embeddings with the perturbed version of the most used codebook embedding during the training, and increases the number of embeddings used from the codebook. New perspectives on comprehending VQ-VAEs such as affine re-parameterization of the codebook embeddings, alternated optimization during the training, and synchronized update rule for the quantized representation are proposed by Huh et al. [14] to address the problems of VQ-VAEs including the codebook collapse.

To incorporate stochasticity, a soft expectation maximization (EM) algorithm is reformulated based on the hard EM modeling of the vector quantization [25]. GS-VQ-VAE [26] uses the Euclidean distance between the encoder's output and the codebook as the parameters of a Categorical distribution, and the codebook embeddings are selected by sampling. SQ-VAE [13] defines stochastic quantization and dequantization processes which are parameterized by probability distributions. Those stochastic processes enable codebook usage implicitly without needing additional tricks such as codebook resetting.

Although the codebook collapse problem of the VQ-VAEs is studied in detail with observed success, it is still an open question for dVAEs. The dVAEs are employed instead of VQ-VAEs in [6] to obtain an image representation for the text-to-image generation problem. As the dVAEs have shown great potential in such complicated tasks, specifying existing problems of dVAE and providing relevant solutions are poised to bring both methodological and practical benefits.

To that end, our work proposes a novel way to combat the codebook collapse problem of the dVAEs. While the proposed methods for VQ-VAEs have the same objective of mitigating the codebook collapse problem like our method, our intuition differs since we focus on the properties of the distribution learned over the codebook, and try to attain a better distribution. On the other hand, other methods focus on the internal dynamics of the VQ-VAE model and its training.

Hierarchical Bayesian Models: Hierarchical Bayesian models have diverse variants like belief networks [27], Bayesian neural networks [28], applied in healthcare, finance, and machine learning. Hierarchical VAEs are the realization of hierarchical Bayesian modeling in the context of deep generative models. Some hierarchical VAEs, exemplified by models like the Ladder VAE [29], leverage a layered structure to capture hierarchical dependencies in data. In contrast, models like VQ-VAE-2 [11] adopt a different approach to hierarchical VAEs. VQ-VAE-2 focuses on discrete latent codes organized hierarchically within a codebook. However, our model exhibits a hierarchical structure by modeling uncertainty in the parameters of the Categorical distribution, and it involves sampling latent variables from these distributions.

Evidential Deep Learning: The foundational work by Sensoy et al. [18] has been pivotal in advancing uncertainty quantification in deep learning, and its methodology for modeling uncertainty inspires subsequent research. Soleimany et al. [30] applies EDL to molecular property prediction, Bao et al. [31] explores its use in open-set action recognition, and Wang et al. [32] focuses on uncertainty estimation for stereo matching. These works collectively demonstrate the versatility and impact of the EDL framework, showcasing its influence across diverse domains. In our work, we employ EDL within a VAE-based framework for the first time. This innovative incorporation leads to a hierarchical Bayesian model using Dirichlet-Categorical distributions.

3. Background

3.1. Discrete variational autoencoders

Discrete VAEs aim to model the high-dimensional data x with the low-dimensional and discrete latent representation z by maximizing the ELBO of the log-likelihood of the data:

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q(z|x)}[\log p(x|z)] - \text{KL}[q(z|x)\|p(z)], \tag{1}$$

where $p(x|z)$ is the generative model which is designed as a decoder, $q(z|x)$ is the approximated posterior which is designed as an encoder, and $p(z)$ is a prior.

In discrete VAEs, d dimensional z is sampled from a Categorical distribution over the possible values of z as $z \sim Cat(z|\pi)$. Encoder returns unnormalized log probabilities $l = [l_1, \dots, l_d]$ and $\pi = softmax(l)$ is used to obtain the probability masses of this Categorical distribution. Then, z is fed into the decoder to reconstruct x.

As the discrete variables sampled from a Categorical distribution do not permit an end-to-end training, methods for continuous relaxation of the discrete variables are proposed, introducing the Gumbel-Softmax trick [33,34]. Gumbel-Softmax trick briefly introduces Gumbel noises

$g = [g_1, \dots, g_d], g_i \sim Gumbel(0, 1)$ into l to introduce randomness so that:

$$z = argmax_d(l + g) \sim Cat(z|\pi), \tag{2}$$

holds. While Eq. (2) is the first step to take a differentiable sample from a Categorical distribution, taking the *argmax* is also non-differentiable. Therefore, softmax operation with a temperature parameter τ:

$$z_m = \frac{e^{\frac{l_m+g_m}{\tau}}}{\sum_{j=1}^{k} e^{\frac{l_j+g_j}{\tau}}}, \tag{3}$$

where $m \in \{1, \dots, d\}$ can be used to approximate *argmax* operation, and τ controls how closely the Gumbel-Softmax distribution approximates the Categorical distribution, resulting in a differentiable z.

VQ-VAE is a discrete VAE variant where a learnable codebook $\mathcal{M} \in \mathbb{R}^{K \times D}$ consisting K number of D dimensional embeddings is trained to represent a dataset. In VQ-VAE, discrete latent variables are used to retrieve embeddings from $\mathcal{M}$ so that these embeddings can represent x. These discrete latent variables are obtained as follows: an encoder constructs a continuous latent representation $z_e(x) \in \mathbb{R}^{N \times N \times D}$ spanned by $N \times N$ matrices, and the discrete latent $z \in \mathbb{R}^{N \times N \times K}$ is *deterministically* obtained using the one-hot representations of the closest embeddings' indices $\mathcal{M}$ for each $N \times N$ spatial location using Euclidean distance as follows:

$$q(z_i = \text{one-hot(k)}|x) = \begin{cases} 1 & \text{if } k = \text{argmin}_j \|z_e(x)_i - e_j\|_2 \\ 0 & \text{otherwise} \end{cases}$$

where $i \in \{1, \dots, N \times N\}$, $k \in \{1, \dots, K\}$, $z_e(x)_i \in \mathbb{R}^D$ is the ith vector in $z_e(x)$, and $e_j \in \mathbb{R}^D$ is a codebook embedding. Then, a quantized representation $z_q(x) = z * \mathcal{M}$ is obtained using the indices zs to retrieve the corresponding codebook embeddings. The operator * performs tensor-matrix multiplication where each z_i for each spatial position will retrieve the related codebook vectors from codebook matrix $\mathcal{M}$. Lastly, the quantized representation $z_q(x) \in \mathbb{R}^{N \times N \times D}$ is fed into the decoder to reconstruct x. The value of N changes depending on the downsampling amount of the encoder (see Appendix C for encoder design).

Unlike VQ-VAEs, dVAE [6] compresses x into $z_e(x) \in \mathbb{R}^{N \times N \times K}$ which is taken as the unnormalized log probabilities of a Categorical distribution over K different codebook embeddings for each $N \times N$ spatial location. Then, the quantizing discrete codebook indices are sampled as follows:

$$\pi = softmax(z_e(x)), \qquad z \sim Cat(z|\pi), \qquad z_q(x) = z * \mathcal{M}. \tag{4}$$

While VQ-VAE obtains the discrete latent z deterministically, dVAE samples them as in Eq. (4). dVAE also incorporates Gumbel-Softmax relaxation for differentiability of the sampled discrete variables zs in Eqs. (2) and (3).

3.2. Hierarchical Bayesian models

A Hierarchical Bayesian model is defined as a chain of random variables following a hierarchical structure, where each level represents a different level of abstraction or variability in the data. Given a data set $X = \{x_i | i = 1, \dots, L\}$ of L observations, assume the generation of this data set is governed by two groups of local latent variables U and Z that are independent across data points. The joint distribution reads:

$$p(X, Z, U) = p(X|Z, U)p(Z|U)p(U) = \prod_i^L p(x_i|z_i, u_i)p(z_i|u_i)p(u_i),$$

where the first equality follows from the assumed hierarchy starting from the root node U going down through Z to X, and the second from the independence assumption of the latent variables. The resulting data generating process can then be written as:

$$u_i \sim p(u_i), \qquad z_i|u_i \sim p(z_i|u_i), \qquad x_i|z_i, u_i \sim p(x_i|z_i, u_i).$$

z_i could be a high level conceptual representation of x_i, e.g. determinant features of an object and u_i an even higher level representation, such as properties of object groups. The fact that these variables are linked via probability distributions also accounts for random factors in a way that accumulates downward in the hierarchy.

A hierarchical deep generative network represents the conditional relationships in this data generating process as Neural Networks (NNs). For instance, Hierarchical VAEs do that by introducing hierarchical layers of latent variables and mapping the conditional distributions $p(x_i|z_i, u_i), p(z_i|u_i)$, and $p(u_i)$ to the parameters of NNs. Specifically, the encoder network maps the observed data x_i to latent variables z_i and u_i, while the decoder network reconstructs the data from sampled latent variables. The hierarchical structure facilitates the learning of increasingly abstract representations of the data.

3.3. Evidential deep learning as a hierarchical Bayesian model

EDL enhances the NNs in order to allow for probabilistic uncertainty quantification for various tasks such as classification [18] or regression [35]. As the deterministic NNs output a point estimate for the given input which does not comprise uncertainty over the decision, the EDL changes this behavior such that the output of the NN is a set of probabilities that represent the likelihood of each possible outcome, by using the evidential logic. EDL implements a hierarchical Bayesian model for a feed-forward classification task as:

$$u_i|x_i \sim p(u_i|x_i), \qquad z_i|u_i \sim p(z_i|u_i), \qquad y_i|z_i \sim p(y_i|z_i) \tag{5}$$

where the hyperprior u_i is conditioned on an input observation x_i, and y_i is the class label of x_i. EDL's loss function can be cast as a variational inference problem where $q(u, z|x) = p(z|u)p(u|x)$.

4. Method

In this work, we enhance the dVAE model with an evidential perspective to mitigate the codebook collapse problem. To establish a stronger connection between the motivation and the proposed approach, it is essential to delve deeper into the root cause of codebook collapse. This issue arises when the model consistently relies on previously used codebook embeddings throughout training. Conceptually, the phenomenon of confirmation bias aligns with the codebook collapse problem. Over the course of training, the model gradually develops a bias, excessively associating new observations with previously encountered ones due to **overconfidence**. Given that the softmax function has been acknowledged to exhibit overconfidence issues [17,18], our motivation stems from the need to address the overconfidence challenge. Leveraging EDL, a framework demonstrated to alleviate the overconfidence problem associated with the softmax function, becomes a compelling solution. To sum up, overconfidence is the key connection between the evidential perspective and the codebook collapse problem, and EdVAE strategically leverages EDL's capabilities to counteract the codebook collapse.

Overview of the proposed method is shown in Fig. 1. The encoder $\mathcal{E}_\theta$'s continuous latent space $z_e(x) \in R^{N \times N \times K}$ spanned by $N \times N$ matrices as explained in Section 3.1 is in agreement with the goal of learning a distribution over K number of embeddings. Originally, $z_e(x)$ in Fig. 1 is passed to a softmax activation to obtain the parameters πs of the Categorical distribution that models the codebook embedding assignment in dVAE as explained in Section 3.1. In EdVAE, we define a higher-level Dirichlet prior over πs, and treat $z_e(x)$ as the concentration parameters α_θ to be learned of this distribution. Therefore, EDL approach builds a hierarchical Bayesian model of dVAE. In Fig. 1, the areas highlighted by blue and gray show the hierarchy between the distributions. Unlike dVAE which has a single level of Categorical distribution as highlighted by gray, we employ a hierarchical level in EdVAE highlighted by blue, in order to incorporate randomness with well quantified uncertainty over the selection process of the codebook embeddings by virtue of

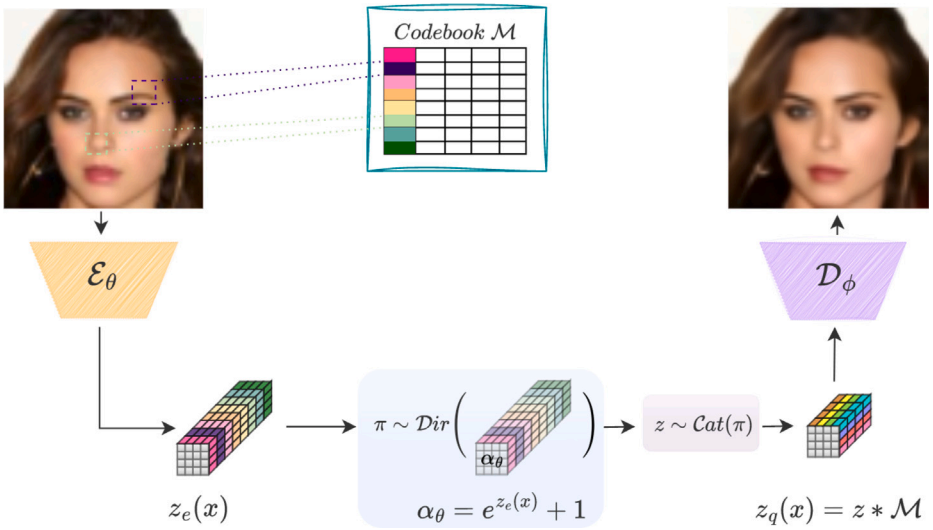


Fig. 1. Overview of the method. An illustrative codebook is defined as $\mathcal{M} \in R^{8\times4}$ where 8 is the number of the codebook embeddings, 4 is the dimensionality of each embedding. For each 16 spatial positions in $z_e(x)$ where N is 4, we define a Dirichlet prior over the parameters of the Categorical distributions which models the codebook embedding assignment to each spatial position. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Dirichlet prior. EDL incorporation models a second-order uncertainty in EdVAE such that α_θ represents how confident this prediction is, and π predicts which codebook element can best reconstruct the input. The incorporated uncertainty awareness is expected to increase the codebook usage in EdVAE which is highly limited in dVAE due to the softmax operation. The intuition behind this expectation originates from the definition of the codebook collapse. New codebook elements are employed whenever the existing ones cannot explain the newcoming observation. Codebook collapse occurs when the model's prediction is wrong on whether the learned representations are capable of reconstructing the new observation or not. Therefore, employing uncertainty awareness over the codebook embedding selection enables the model to use unused codebook embeddings when it is uncertain about the codebook embedding selection. To validate this, we conduct an experiment revealing a correlation between uncertainty values and perplexity in CIFAR10, and this correlation supports our intuition (see Section 5.2.2).

Algorithm 1 Training algorithm of EdVAE

Input: Dataset $\mathbf{x}_{\text{train}}$
Output: Reconstructed $\mathbf{x}_{\text{train}}$
Initialize the encoder $\mathcal{E}_\theta^{[0]}$, the decoder $D_\phi^{[0]}$, the codebook $\mathcal{M}^{[0]}$, and the temperature parameter $\tau^{[0]} = 1.0$
for $t = 1, 2, \dots, T$ **do**
  $x \leftarrow$ Random minibatch from $\mathbf{x}_{\text{train}}$
  $z_e(x) \leftarrow \mathcal{E}_\theta^{[t-1]}(x)$
  $\alpha_\theta \leftarrow e^{z_e(x)} + 1$
  $\pi \sim Dir(\pi|\alpha_\theta)$
  $z \sim$ RelaxedOneHotCategorical(temperature $= \tau^{[t-1]}$, probs $= \pi$)
  $\hat{x} \leftarrow D_\phi^{[t-1]}(\mathcal{M}, z)$
  $g \leftarrow \nabla_{\mathcal{M},\theta,\phi} \mathcal{L}(\mathcal{M}^{[t-1]}, \theta^{[t-1]}, \phi^{[t-1]})$
    with sampled x and $\hat{x}$
  $\mathcal{M}^{[t]}, \theta^{[t]}, \phi^{[t]} \leftarrow$ Update parameters using g
  $\tau^{[t]} \leftarrow$ CosineAnneal$(\tau^{[t-1]}, t)$
end for

4.1. EdVAE design

In order to incorporate an evidential mechanism into the dVAE model, the forward model and the design choices should be properly arranged. When we design our forward model, we follow a similar form to the latent variable modeling of EDL as described in Section 3.3 where $p(u_i|x)$ in Eq. (5) is modeled as a Dirichlet distribution over the Categorical distribution $p(z_i|u_i)$. The concentration parameters, α_θ, of the Dirichlet distribution are defined to be greater than or equal to 1. Therefore, $z_e(x)$ is passed through an exp(.) operation to obtain *evidences*, and 1 is added to obtain the concentration parameters as follows:

$$\alpha_\theta(x) = \exp(z_e(x)) + 1. \tag{6}$$

We define our forward model to be:

$$p(\pi) = Dir(\pi|1, \dots, 1), \tag{7}$$

$$Pr(z|\pi) = Cat(z|\pi), \tag{8}$$

$$p(x|\mathcal{M}, z = k) = \mathcal{N}(x|D_\phi(\mathcal{M}, z), \sigma^2 I). \tag{9}$$

In dVAE, the prior is defined as a uniform distribution over the codebook embeddings. Eq. (7) demonstrates our prior design as a Dirichlet distribution that generates uniform distributions over the codebook embeddings on average, and $\pi = [\pi_1, \dots, \pi_K]$. Eq. (8) shows that we model the embedding selection as a Categorical distribution where z is the index of the sampled codebook embedding from $\mathcal{M}$, and the parameters πs of the Categorical distribution are sampled from the Dirichlet prior. During the training, we obtain samples from the Categorical distribution using Gumbel-Softmax relaxation to backpropagate gradients to the encoder. We decay the temperature parameter of the Gumbel-Softmax to 0 as described in [33] so that the soft quantization operation turns into the hard quantization. Therefore, there may be multiple embedding dimensions that are chosen independently due to the temperature value at train-time. At test-time, we perform hard quantization and simply take a single sample from Eq. (8). The algorithms of the training and the inference of EdVAE are given in Algorithms 1 and 2, respectively. t denotes the index of the training iterations, and b denotes the batch index in the inference. RelaxedOneHotCategorical(.) distribution is differentiable, and the samples from this distribution are soft one-hot vectors. On the other hand, Categorical(.) distribution is not differentiable which is not required during the inference. The samples from this distribution are hard one-hot vectors which indicate one-to-one quantizations.

In Eq. (9), we define our likelihood $p(x|\mathcal{M}, z = k)$ as a Normal distribution, and D_ϕ denotes the decoder network. The input of the decoder can be formed $z_q(x) = z * \mathcal{M}$ as explained in Section 3.1.

Algorithm 2 Inference algorithm of EdVAE

Input: Dataset $\mathbf{x}_{\text{test}}$
Output: Reconstructed $\mathbf{x}_{\text{test}}$
Freeze the parameters of the trained encoder $\mathcal{E}_\theta$, the trained decoder $\mathcal{D}_\phi$,
and the trained codebook $\mathcal{M}$
for $b = 1, 2, \dots, B$ **do**
  $x_b \leftarrow$ Minibatch from $\mathbf{x}_{\text{test}}$
  $z_e(x) \leftarrow \mathcal{E}_\theta(x)$
  $\alpha_\theta \leftarrow e^{z_e(x)} + 1$
  $\pi \sim Dir(\pi|\alpha_\theta)$
  $z \sim$ Categorical(probs $= \pi$)
  $\hat{x} \leftarrow \mathcal{D}_\phi(\mathcal{M}, z)$
end for

We approximate the intractable true posterior $p(\pi, z|x)$ by structured variational inference. Thanks to the Dirichlet-Categorical conjugacy, we use the following approximate distribution that accounts for the full factorization of the codebook elements and the uncertainty on their probabilities:

$$q(\pi, z|x) = Cat(z|\pi)Dir(\pi|\alpha_\theta^1(x), \dots, \alpha_\theta^K(x)). \tag{10}$$

where $\alpha_\theta(x) = \left[\alpha_\theta^1(x), \dots, \alpha_\theta^K(x)\right]$. Our hierarchical Bayesian model has a Normal distributed likelihood function the mean of which takes codebook elements chosen by the Categorical distribution as the input and maps them to the image space. It follows the Bayesian structure explained in Section 3.2. Even though the Dirichlet-Categorical conjugacy does not grant us an analytical solution for the posterior distribution, it simplifies the computations of the ELBO detailed in Appendix B.

As a result of our derivations, ELBO to be maximized during the training is:

$$\mathcal{L}(\mathcal{M}, \theta, \phi) = \mathbb{E}_{Pr(z|\pi)}\left[\mathbb{E}_{q(\pi|x)}[\log p(x|\mathcal{M}, z)]\right] - \mathcal{D}_{\text{KL}}(q(\pi|x)\|p(\pi)). \tag{11}$$

The second term in Eq. (11) is the Kullback–Leibler divergence between two Dirichlet distributions, hence has the analytical solution as derived in Appendix A. After further derivations given in Appendix B over the first term in Eq. (11), our loss function is derived in Eq. (12):

$$\mathcal{L}(\mathcal{M}, \theta, \phi) = \mathbb{E}_{q(\pi|x)}\left[(x - \mathcal{D}_\phi(\mathcal{M}, z))^2\right] - \beta\mathcal{D}_{\text{KL}}(Dir(\pi|\alpha_\theta(x))\|Dir(\pi|1, \dots, 1)) \tag{12}$$

which is optimized to increase the likelihood, while decreasing the KL distance regularized with β coefficient [4] between the amortized posterior and the prior. We can perform the optimization in an end-to-end manner as we turn the only non-differentiable part in our hierarchical model that is sampling from a Categorical distribution into a differentiable operation RelaxedOneHotCategorical with Gumbel-Softmax as explained in Section 3.1 and shown in Algorithm 1. Therefore, we use Adam optimizer with an initial learning rate $1e^{-3}$ to optimize our model.

The first term in Eq. (12) indicates the reconstruction error in terms of mean squared error (MSE) between the input and the reconstruction. The objective goal of the model is to obtain a lower reconstruction error presuming that a well representing latent space is constructed. The second term indicates that the Dirichlet distribution which defines a distribution over the Categorical distributions should converge to a generated distribution in a uniform shape on average in order to represent the codebook embedding selection in a diverse way. When the distribution over the codebook embeddings is uniform, codebook usage is maximized as the probabilities of each codebook embedding to be selected become similar in value. Therefore, while the first term in Eq. (12) aids directly in building the representation capacity of the latent space, the second term in Eq. (12) indirectly supports that goal via a diversified codebook usage. We use a β coefficient to balance the reconstruction performance and the codebook usage.

EdVAE introduces a hierarchical structure by learning a Dirichlet distribution over the parameters of a Categorical distribution, with latent variables sampled from this distribution. This approach suits the hierarchical Bayesian modeling basics we present in Section 3.2 while it differs from other hierarchical VAEs in several aspects. Firstly, the hierarchy is expressed through probabilistic modeling of the parameters, rather than through discrete latent codes or layered representations, in order to increase the codebook entropy. Secondly, while the model captures uncertainty in the parameters, it does not explicitly learn hierarchical representations of the data. Lastly, variational inference involves approximating the posterior distribution over latent variables through the learned Dirichlet-Categorical relationship. Overall, EdVAE introduces a unique approach to hierarchical VAEs, emphasizing probabilistic modeling of parameters and differing from others in terms of representation learning and variational inference strategies.

5. Experiments

5.1. Experimental settings

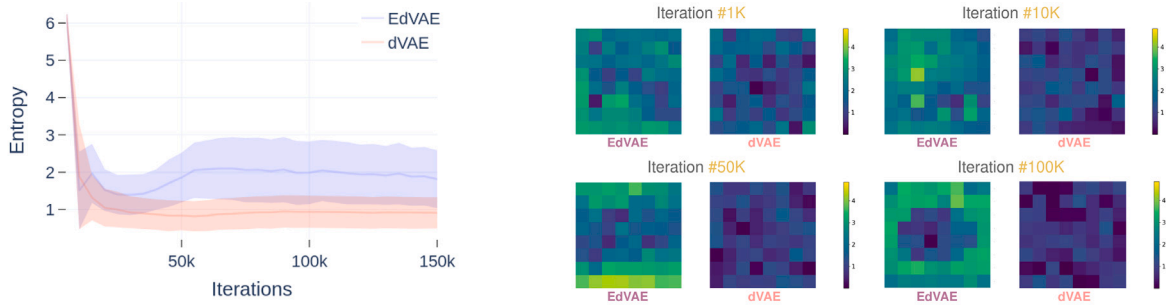
We perform experiments on CIFAR10 [36], CelebA [37], and LSUN Church [38] datasets to demonstrate the performance of EdVAE compared to the baseline dVAE and VQ-VAE based methods. We use VQ-VAE-EMA which updates the codebook embeddings with exponential moving averages, and GS-VQ-VAE as the basic VQ-VAE models along with the state-of-the-art VQ-VAE based methods including SQ-VAE and VQ-STE++ mitigating codebook collapse problem. We use the same architectures and hyperparameters described in the original papers and official implementations for a fair comparison. We repeat all of our experiments using three different seeds. Hyperparameter choices and architectural designs are further detailed in Appendix C.

5.2. Evaluations

5.2.1. Effects of the softmax distribution

Our hypothesis is that the spiky softmax distribution over the codebook embeddings develops confirmation bias, and the confirmation bias causes a codebook collapse. We test our hypothesis by comparing the average entropy of the probability distributions learned by the encoders of dVAE and EdVAE during the training using CIFAR10 dataset. Low entropy indicates a spiky distribution while high entropy represents a distribution closer to a uniform shape, which is the desired case for a diverse codebook usage. A spiky probability distribution yields a confirmation bias because the codebook embedding with the highest probability mass is favorably selected. On the other hand, when the Categorical distribution has a flatter probability distribution, a variety of codebook embeddings might be sampled to represent the same data for its different details, which leads to an enriched codebook and an enhanced codebook usage.

Fig. 2(a) visualizes the average entropy of the probabilities during the training. We measure the entropy of each probability distribution over the codebook embeddings sample-wise, meaning that each sample consists of $N \times N$ number of entropy values calculated for each spatial position. We gather all $L \times N \times N$ entropy values at on the entire dataset where L is the number of training samples, and plot the average entropy of the probabilities with mean and standard deviation. We find out that EdVAE's mean values of the entropy are higher than those of dVAE's, and this performance gain in terms of entropy is preserved during the training. Furthermore, we observe higher standard deviation for EdVAE in contrast to dVAE, which indicates that dVAE squeezes the probability masses into a smaller interval for all positions while the entropy values of EdVAE have a wider range aiding a relatively liberal codebook usage.



(a) Average entropy of the probabilities during the training.

(b) Entropy of the probabilities for each spatial position in the same sample during the training.

Fig. 2. Entropy visualization of the probability distributions for CIFAR10. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 3. EdVAE training on CIFAR10: perplexity values increase during the training due to the increase in uncertainty values.

Fig. 2(b) visualizes the entropy changes of the probabilities for each spatial position in the same sample during the training. All iterations have the same color bar for both models to observe the change effectively. Heat map visualization is useful to monitor the entropy of the probabilities within a single sample, and EdVAE obtains higher entropy values for most of the spatial positions compared to those of dVAE during the training.

We note that while the prior works [13,15] induce high entropy via regularizers, the feature introduced by our ELBO formulation inherently achieves the same effect as a result of our modeling assumptions that harness the power of the Dirichlet distribution.

While the spiky softmax distribution is the main problem, one might think to increase the stochasticity of sampling from the Categorical distribution with a higher temperature value to reduce the effects of overestimated probabilities. In order to test this assumption, we use higher temperature values with dVAE on CIFAR10 and CelebA datasets (see Appendix D.1 for details). We show that the perplexity does not increase with a high temperature, and including randomness insensibly does not always affect the training in a good way. Our method incorporates stochasticity such that we learn how to perturb probabilities from the input. Learning a distribution over Categorical distributions from the input directly makes the model more reliable and resistant to hyperparameter change.

5.2.2. Uncertainty vs. Perplexity

We anticipate that introducing uncertainty awareness will enhance codebook usage, addressing limitations in dVAE caused by the softmax operation. Our intuition is rooted in the definition of codebook collapse—new elements are introduced when existing ones fail to explain observations. In order to validate our intuition, we monitor the training of CIFAR10 and present an interval of the training until saturation in Fig. 3.

Table 1
Comparison of the models in terms of perplexity (↑) using a codebook $\mathcal{M}^{512\times16}$.

Method	CIFAR10	CelebA	LSUN Church
VQ-VAE-EMA [5]	412.67 ± 2.05	405.33 ± 5.88	379.67 ± 3.09
GS-VQ-VAE [26]	208.33 ± 6.03	193.33 ± 10.68	189.67 ± 5.02
SQ-VAE [13]	407.33 ± 7.32	**409.33 ± 2.05**	374.00 ± 2.16
VQ-STE++ [14]	414.33 ± 9.10	370.33 ± 4.11	375.67 ± 5.58
dVAE [6]	190.33 ± 13.02	254.67 ± 11.08	363.33 ± 4.07
EdVAE	**420.33 ± 4.49**	371.33 ± 2.86	**385.67 ± 5.63**

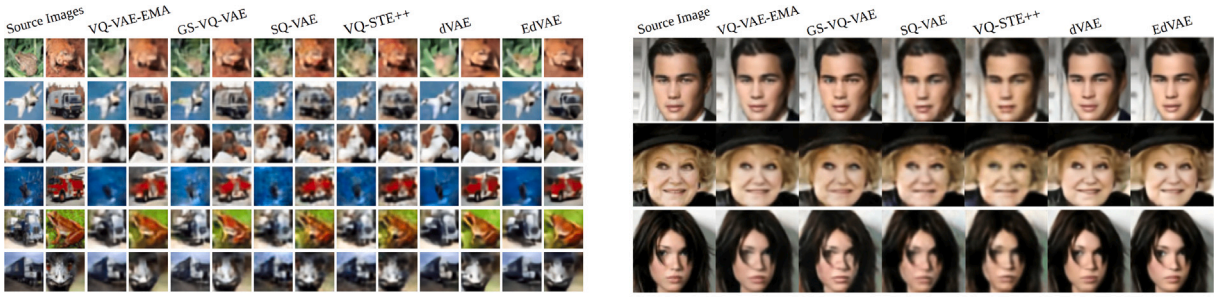
Table 2
Comparison of the models in terms of MSE ($\times10^3$, ↓) using a codebook $\mathcal{M}^{512\times16}$.

Method	CIFAR10	CelebA	LSUN Church
VQ-VAE-EMA [5]	3.21 ± 0.05	1.07 ± 0.06	1.71 ± 0.05
GS-VQ-VAE [26]	3.63 ± 0.01	1.32 ± 0.02	1.84 ± 0.06
SQ-VAE [13]	4.01 ± 0.03	1.05 ± 0.02	1.79 ± 0.03
VQ-STE++ [14]	3.82 ± 0.1	1.11 ± 0.08	1.83 ± 0.03
dVAE [6]	3.42 ± 0.08	1.01 ± 0.08	1.60 ± 0.01
EdVAE	**2.99 ± 0.04**	**0.89 ± 0.01**	**1.58 ± 0.01**

We observe a correlation between the perplexity values and the uncertainty values during the training. The trend of perplexity values perfectly matches the trend of uncertainty values. This correlation emphasizes that our model dynamically adjusts codebook usage based on its uncertainty, preventing codebook collapse by utilizing embeddings effectively.

5.2.3. Perplexity and reconstruction performance

In order to evaluate the codebook usage of all models, we leverage on the perplexity metric whose upper bound is equal to the number of the codebook embeddings. Therefore, we directly compare the number of used embeddings for all models. Perplexity results of all models



(a) CIFAR10. **(b)** CelebA.

Fig. 4. Reconstructions from (a) CIFAR10, (b) CelebA.

Fig. 5. Reconstructions from LSUN Church.

Table 3
Comparison of the models in terms of FID (↓).

Method	CIFAR10	CelebA	LSUN Church
VQ-VAE-EMA [5]	57.04 ± 2.32	34.30 ± 2.41	71.22 ± 2.72
GS-VQ-VAE [26]	56.35 ± 2.17	33.12 ± 1.30	72.52 ± 3.25
SQ-VAE [13]	54.17 ± 2.85	33.03 ± 1.04	**63.41 ± 2.36**
VQ-STE++ [14]	55.53 ± 1.97	32.98 ± 2.27	71.03 ± 1.95
dVAE [6]	58.85 ± 0.93	37.29 ± 3.14	71.32 ± 0.71
EdVAE	**51.82 ± 1.58**	**32.51 ± 1.13**	69.63 ± 1.29

are presented in Table 1. Additionally, since the reconstruction performance should not be lost while increasing the diversity of the codebook embeddings used in the latent representation, we also evaluate all models in terms of MSE. Numerical results are presented in Table 2. While we obtain the lowest MSE values for all datasets, we outperform the other methods for CIFAR10 and LSUN Church datasets in terms of perplexity. Whereas SQ-VAE achieves the highest perplexity result for CelebA dataset, EdVAE performs close to SQ-VAE's perplexity, while obtaining a better reconstruction performance than SQ-VAE and the other methods. Moreover, EdVAE outperforms dVAE substantially in perplexity. It is important to note that EdVAE not only mitigates the codebook collapse problem of dVAE, but also outperforms the VQ-VAE based methods.

We evaluate our model and the other models visually for a qualitative evaluation assessment. Figs. 4(a), 4(b), and 5 compare all of the models' reconstruction performance on CIFAR10, CelebA, and LSUN Church datasets, respectively. We observe that while our model reconstructs most of the finer details such as the shape of eye or gaze direction better than the other models in CelebA, we also include some examples where some of the finer details such as the shape of the nose and the mouth are depicted by one of the opponent models better than our model. For LSUN Church examples, we observe that our model's reconstructions depict the colors and the shapes in the source images better than the other models.

5.2.4. *Effects of codebook design*

We emphasize the critical role of codebook design using CIFAR10 and CelebA datasets in Fig. 6. It is important to have a model that uses most of the codebook embeddings even with a larger codebook. Therefore, we evaluate EdVAE's and other methods' performance using various codebook sizes and dimensionalities. In order to observe the effects of size and dimensionality separately, we fix the dimensionality to 16 while we use different codebook sizes ranging from 128 to 2048. Then, we fix the size to 512 while we use different codebook dimensionalities ranging from 8 to 256.

Fig. 6 shows that EdVAE outperforms the other methods in most of the setting where we use different codebook sizes. EdVAE's perplexity is not affected negatively when the codebook size increases, and it obtains the lowest MSE values in most of the settings for both of the datasets. Therefore, EdVAE is suitable to work with a larger codebook. When the codebook dimensionality changes, we observe that EdVAE outperforms the other methods or obtains compatible results. While the other methods seem sensitive to the codebook design, EdVAE does not need a carefully designed codebook which eases the process of model construction.

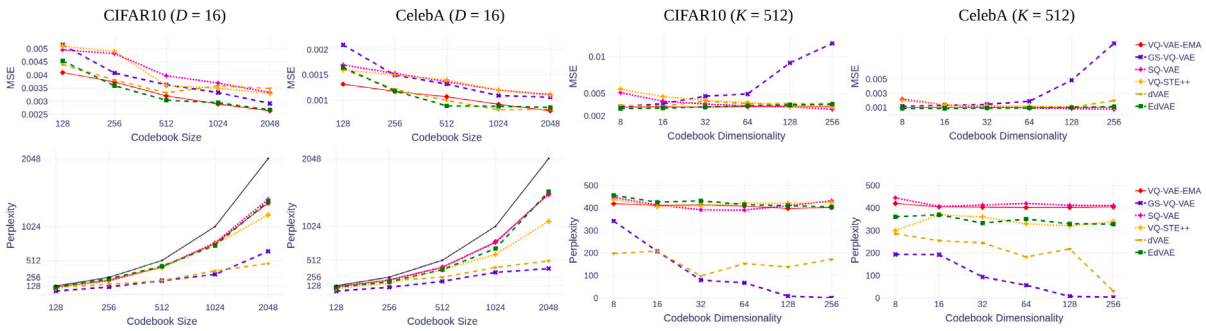


Fig. 6. Impact of codebook design on perplexity and MSE, using CIFAR10 and CelebA datasets. The black "codebook size" line indicates the upper bound for the perplexity.

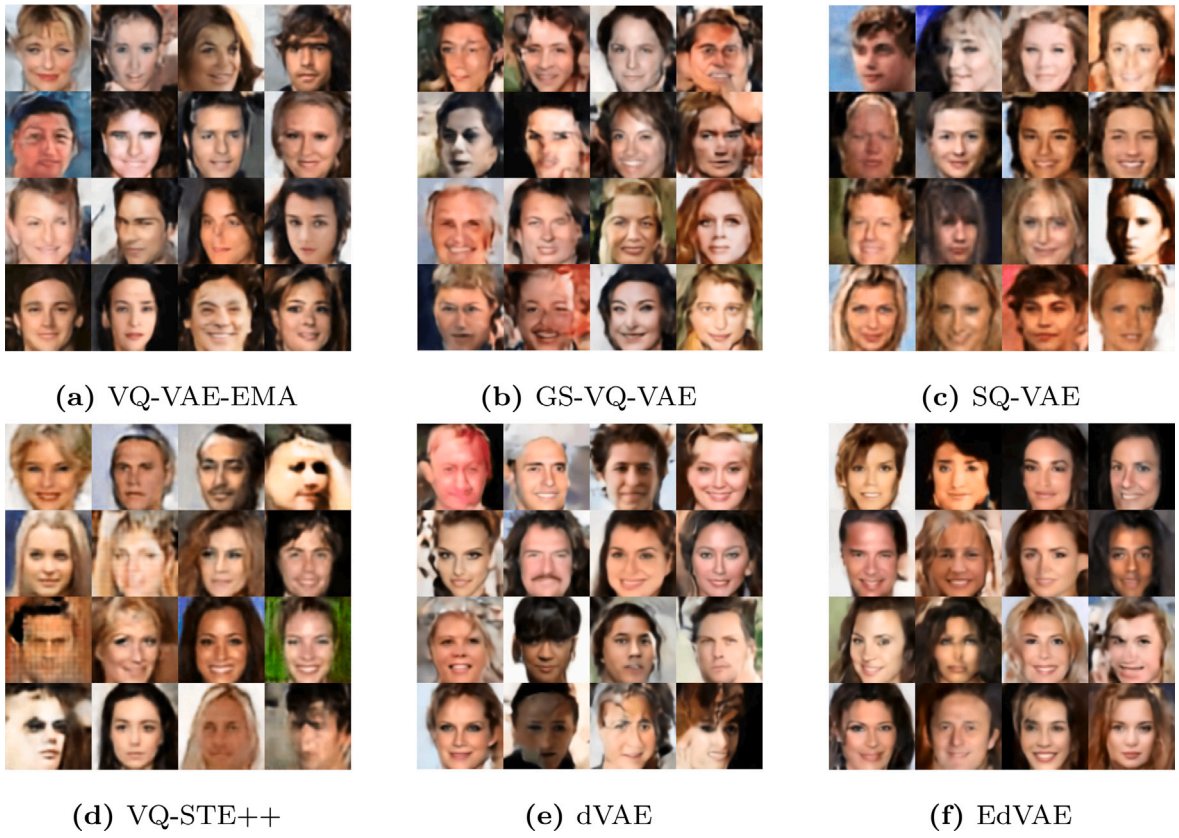


Fig. 7. Generated samples from CelebA dataset.

Table 4
Comparison of the models in terms of Precision & Recall (↑).

Method	CIFAR10	CelebA	LSUN Church
VQ-VAE-EMA [5]	0.47, 0.32	0.44, 0.35	0.19, 0.15
GS-VQ-VAE [26]	0.45, 0.34	0.39, 0.32	0.20, 0.14
SQ-VAE [13]	0.52, 0.30	**0.52**, **0.37**	0.23, 0.17
VQ-STE++ [14]	0.51, 0.32	0.47, 0.36	0.24, 0.15
dVAE [6]	0.43, 0.34	0.41, 0.30	0.20, 0.15
EdVAE	**0.54**, **0.35**	0.48, 0.35	**0.28**, **0.18**

5.2.5. Approximated prior

Image generation is a downstream task over which we evaluate the performance of the discrete latent spaces learned by the discrete VAEs. As the prior used during the training of $\mathcal{E}_\theta$, $\mathcal{D}_\phi$, and $\mathcal{M}$ is a uniform distribution, i.e. an uninformative prior, it should be updated to accurately reflect the true distribution over the discrete latents. Therefore, we fit an autoregressive distribution over the discrete latents of the training samples, for which we follow PixelSNAIL [39]. PixelSNAIL is also used in VQ-VAE-2 instead of PixelCNN [40] that is used in VQ-VAE and SQ-VAE.

We report FID and Precision-Recall in Tables 3 and 4 to evaluate the quality of the generated images that are created autoregressively after a training over the discrete codebook indices attained from a given model, respectively. The FID results imply that EdVAE performs better than the other models in all datasets except LSUN Church, while it outperforms other models in terms of Precision-Recall in all datasets except CelebA. These results elucidate the point that the discrete latent spaces learned by EdVAE helps to attain a reinforced representation capacity for the latent space, which is a desirable goal for the image generation task. Figs. 7 and 8 present generated samples using the discrete latents of all models for CelebA and LSUN Church datasets, respectively. While all of the models manage to generate realistic images, structural consistency and semantic diversity in EdVAE's results are noticeable and in line with the numerical results. Fig. 9 present additional EdVAE samples to advocate EdVAE's performance.

Fig. 8. Generated samples from LSUN Church dataset.

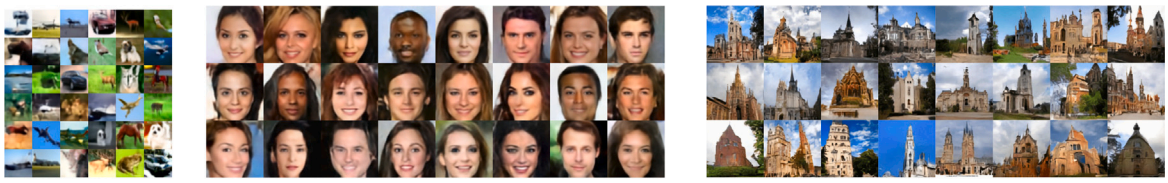


Fig. 9. Generated samples using the learned prior over the discrete indices produced by EdVAE.

6. Conclusion

The proposed EdVAE extends dVAE with a hierarchical Bayesian modeling to mitigate the codebook collapse problem of the latter. We demonstrate the essence of the problem, that is the confirmation bias caused by the spiky softmax distribution, and reformulate the optimization with an evidential view in order to acquire a hierarchy between the probability distributions aiding uncertainty-aware codebook usage. We show that our method outperforms the former methods in most of the settings in terms of reconstruction and codebook usage metrics.

Although the proposed method improves the codebook usage for dVAE family, there is still room for improvements both numerically and experimentally. We evaluate our method over small to medium size datasets compared to datasets like ImageNet [41]. As our work is the first to state the codebook collapse problem of dVAE and to propose a solution to it, it sets a baseline that can be extended to obtain improved outcomes in different models and datasets. We mainly provide evidence for the utility of our method for relatively structured datasets like CelebA or LSUN Church, and exploring the best parameterization for more diverse datasets like ImageNet would be a possible direction of our future work.

CRediT authorship contribution statement

Gulcin Baykal: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Melih Kandemir:** Supervision. **Gozde Unal:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

I have shared a link to my codes in my Manuscript.

Acknowledgments

In this work, Gulcin Baykal was supported by TÜBİTAK 2214-A International Research Fellowship Programme for PhD Students

and Google DeepMind Scholarship Program at ITU. Computing resources used during this research were provided by the National Center for High Performance Computing of Turkey (UHeM) [grant number 1007422020].

Appendix A. Kullback–Leibler divergence between two Dirichlet distribution

For two Dirichlet distributions $Dir(\pi|\alpha_\theta(x))$ and $Dir(\pi|1,\dots,1)$ over K-dimensional probability π, the following equality holds

$$\mathcal{D}_{\mathrm{KL}}(Dir(\pi|\alpha_\theta(x))\|Dir(\pi|1,\dots,1)) = \log\left(\frac{\Gamma\left(\sum_k \alpha_\theta^k(x)\right)}{\Gamma(K)\prod_k \Gamma(\alpha_\theta^k(x))}\right) + \sum_{k=1}^{K}(\alpha_\theta^k(x)-1)\left(\psi(\alpha_\theta^k(x)) - \psi\left(\sum_k \alpha_\theta^k(x)\right)\right).$$

where $\Gamma(.)$ and $\psi(\alpha) := \frac{\mathrm{d}}{\mathrm{d}\alpha}\log\Gamma(\alpha)$ are the *gamma* and *digamma* functions, respectively.

Appendix B. Derivation details

As we define in Section 4, our forward model is:

$$p(\pi) = Dir(\pi|1,\dots,1), \tag{B.1}$$

$$Pr(z|\pi) = Cat(z|\pi), \tag{B.2}$$

$$p(x|\mathcal{M}, z=k) = \mathcal{N}(x|\mathcal{D}_\phi(\mathcal{M},z),\sigma^2 I). \tag{B.3}$$

and the approximate posterior is:

$$q(\pi,z|x) = Cat(z|\pi)Dir(\pi|\alpha_\theta^1(x),\dots,\alpha_\theta^K(x)). \tag{B.4}$$

We derive the ELBO to be maximized during the training as:

$$\log p(x|\mathcal{M},\theta,\phi) = \log \mathbb{E}_{q(\pi,z|x)}\left[\frac{p(x|\mathcal{M},z)Pr(z|\pi)p(\pi)}{q(\pi,z|x)}\right] \tag{B.5}$$

$$= \log \mathbb{E}_{q(\pi,z|x)}\left[\frac{p(x|\mathcal{M},z)\cancel{Pr(z|\pi)}p(\pi)}{\cancel{Pr(z|\pi)}q(\pi|x)}\right] \tag{B.6}$$

$$\geq \mathbb{E}_{Pr(z|\pi)}\left[\mathbb{E}_{q(\pi|x)}[\log p(x|\mathcal{M},z)] - \mathbb{E}_{q(\pi|x)}[\log q(\pi|x)/p(\pi)]\right] \tag{B.7}$$

$$= \mathbb{E}_{Pr(z|\pi)}\left[\mathbb{E}_{q(\pi|x)}[\log p(x|\mathcal{M},z)]\right] - \mathcal{D}_{\mathrm{KL}}(q(\pi|x)\|p(\pi)). \tag{B.8}$$

The codebook can be viewed as $\mathcal{M} = [m_1,\dots,m_K]$ where m_ks are the codebook embeddings. The input of the decoder $z_q(x) = \mathcal{M}_k = z * \mathcal{M}$ consists of the codebook embeddings m_ks as shown in Fig. 1. $\mathcal{M}_k$ is retrieved using the indices zs sampled as $z \sim Cat(z|\pi)$ via * operator which performs tensor-matrix multiplication. Therefore, we can use $\mathcal{D}_\phi(\mathcal{M}_k)$ instead of $\mathcal{D}_\phi(\mathcal{M},z)$ for the remaining of the derivations.

The first term in Eq. (B.8) can be further derived as:

$$\mathbb{E}_{Pr(z|\pi)}\left[\mathbb{E}_{q(\pi|x)}[\log p(x|\mathcal{M},z)]\right] = -\frac{1}{2}\log\sigma^2 - \frac{1}{2\sigma^2}\sum_k q(\pi_k|x)(x-\mathcal{D}_\phi(\mathcal{M}_k))^2 \tag{B.9}$$

$$\propto \mathbb{E}_{q(\pi|x)}\left[\sum_k \pi_k (x-\mathcal{D}_\phi(\mathcal{M}_k))^2\right] \tag{B.10}$$

$$= \mathbb{E}_{q(\pi|x)}\left[\sum_k \pi_k (x^Tx - 2x^T\mathcal{D}_\phi(\mathcal{M}_k) + \mathcal{D}_\phi^T(\mathcal{M}_k)\mathcal{D}_\phi(\mathcal{M}_k))\right] \tag{B.11}$$

$$= x^Tx - 2x^T\mathbb{E}_{q(\pi|x)}\left[\sum_k \pi_k \mathcal{D}_\phi(\mathcal{M}_k)\right] + \mathbb{E}_{q(\pi|x)}\left[\sum_k \pi_k \mathcal{D}_\phi^T(\mathcal{M}_k)\mathcal{D}_\phi(\mathcal{M}_k)\right] \tag{B.12}$$

$$= x^Tx - 2x^T\sum_k \mathbb{E}_{q(\pi|x)}\left[\pi_k\right]\mathcal{D}_\phi(\mathcal{M}_k) + \mathbb{E}_{q(\pi|x)}\left[\sum_k \pi_k \mathcal{D}_\phi^T(\mathcal{M}_k)\mathcal{D}_\phi(\mathcal{M}_k)\right] \tag{B.13}$$

$$= x^Tx - 2x^T\sum_k \frac{\alpha_\theta^k(x)}{S}\mathcal{D}_\phi(\mathcal{M}_k) + \mathbb{E}_{q(\pi|x)}\left[\sum_k \pi_k \mathcal{D}_\phi^T(\mathcal{M}_k)\mathcal{D}_\phi(\mathcal{M}_k)\right] \tag{B.14}$$

$$= x^Tx - 2x^T\sum_k \frac{\alpha_\theta^k(x)}{S}\mathcal{D}_\phi(\mathcal{M}_k) + \sum_k \mathbb{E}_{q(\pi|x)}\left[\pi_k\right]\mathcal{D}_\phi^T(\mathcal{M}_k)\mathcal{D}_\phi(\mathcal{M}_k) \tag{B.15}$$

$$= x^Tx - 2x^T\sum_k \frac{\alpha_\theta^k(x)}{S}\mathcal{D}_\phi(\mathcal{M}_k) + \sum_k \frac{\alpha_\theta^k(x)}{S}\mathcal{D}_\phi^T(\mathcal{M}_k)\mathcal{D}_\phi(\mathcal{M}_k) \tag{B.16}$$

$$= \mathbb{E}_{z\sim\pi}\left[(x-\mathcal{D}_\phi(\mathcal{M}_k))^T(x-\mathcal{D}_\phi(\mathcal{M}_k))\right] \tag{B.17}$$

$$= \mathbb{E}_{z\sim\pi}\left[\|x-\mathcal{D}_\phi(\mathcal{M}_k)\|_2^2\right]. \tag{B.18}$$

We define $S = \sum_k \alpha_\theta^k(x)$, and $\pi = [\pi_1,\dots,\pi_K]$. $\mathbb{E}_{q(\pi|x)}[\pi]$ in Eq. (B.15) of the ELBO derivation is equal to $\alpha_\theta^k(x)/S$ in Eq. (B.16) using the properties of the Dirichlet distribution. Therefore, $\pi_k = \alpha_\theta^k(x)/S$ holds. After we obtain the probabilities πs, we can get samples $z \sim Cat(z|\pi)$ using the Gumbel-Softmax trick that is explained in Section 3.1.

Appendix C. Experimental details

We use PyTorch framework in our implementation. We train all of the models for 150K iterations on all datasets, using a single NVIDIA A100 GPU. Our computation time for training EdVAE for 150K iterations varies within 12–21 h based on the image size of the data, and training PixelSNAIL over the latents for 500 epochs is 9 h. We use 128 as the batch size, and the Adam optimizer with an initial learning rate $1e^{-3}$, and follow the cosine annealing schedule to anneal the learning rate from $1e^{-3}$ to $1.25e^{-6}$ over the first 50K iterations. We rerun all of our experiments using the seed values of *42*, *1773*, and *1*.

We anneal the β coefficient starting from 0 over the first 5K iterations with cosine annealing schedule. Based on the model and the dataset, the upper bound for the β coefficient varies as the KL terms are different in different models, and the β coefficient decides the reconstruction vs. KL term tradeoff. We also follow the temperature annealing schedule $\tau = \exp(-10^{-5}.t)$ for the Gumbel-Softmax where τ denotes the temperature, and t denotes the global training step. We initialize the codebook embeddings of these models using a Gaussian normal distribution. In EdVAE, we clamp the encoder's output $z_e(x)$ to be maximum 20 before converting it to the α_θ parameters for the training stability. We observe that after we obtain the training stability, we clamp too few variables which does not affect the integrity of the latent variables. We provide detailed analysis for the effects of logits clamping in Appendix D.2.

As datasets, we use CIFAR10, CelebA, and LSUN Church. CIFAR10 consists of 60,000 32×32 RGB images in 10 classes. Each class consists of the same amount of images. We use the default train/test split of the dataset. CelebA dataset consists of more than 200,000 celebrity images. We use the default train/val/test split of the dataset. As preprocessing, we perform center cropping of 140×140, and resize the cropped images to 64×64 using bilinear interpolation. LSUN Church consists of 126,000 256×256 RGB images of various churches. We use the default train/test split of the dataset. We resize the images to 128×128 resolution using bilinear interpolation to use.

For VQ-VAE-EMA and GS-VQ-VAE, we use the same architecture and hyperparameters as suggested in [5]. We set the β coefficient for VQ-VAE-EMA's loss to 0.25, and the weight decay parameter for the EMA to 0.99. We initialize the codebook embeddings using a uniform distribution as in [5]. For GS-VQ-VAE, the upper bound for the annealed β coefficient of the KL divergence is set to $5e^{-6}$ for all datasets.

For SQ-VAE and VQ-STE++, we use the proposed architectures and hyperparameters, and replicate their results.

As dVAE is the baseline model for EdVAE, we describe the architecture of dVAE and EdVAE in detail. The common building blocks

Table C.5
Notations of network layers used on all models.

Notation	Description
$\text{Conv}_n^{(7\times7)}$	2D Conv layer (out_ch = n, kernel = 7, stride = 1, padding = 3)
$\text{Conv}_n^{(4\times4)}$	2D Conv layer (out_ch = n, kernel = 4, stride = 2, padding = 1)
$\text{Conv}_n^{(3\times3)}$	2D Conv layer (out_ch = n, kernel = 3, stride = 1, padding = 1)
$\text{Conv}_n^{(1\times1)}$	2D Convl layer (out_ch = n, kernel = 1, stride = 1, padding = 1)
MaxPool	2D Max pooling layer (kernel_size = 2)
Upsample	2D upsampling layer (scale_factor = 2)
EncResBlock_n	$3\times(\text{ReLU}\rightarrow\text{Conv}_n^{(3\times3)})\rightarrow\text{ReLU}\rightarrow\text{Conv}_n^{(1\times1)}$ + identity
DecResBlock_n	$\text{ReLU}\rightarrow\text{Conv}_n^{(1\times1)}\rightarrow 3\times(\text{ReLU}\rightarrow\text{Conv}_n^{(3\times3)})$ + identity

used in the encoders and the decoders are given in Table C.5. For the following architectures, w and h denote the *width* and the *height* of the images. For CIFAR10 $w = h = 32$, for CelebA $w = h = 64$, and for LSUN Church $w = h = 128$. We use a codebook $\mathcal{M} \in R^{512\times16}$ for all of our experiments.

As the encoder of dVAE and EdVAE return a distribution over the codebook, the last dimensions of the encoders' outputs $z_e(x)$s are all equal to 512 for all datasets. After the quantization of $z_e(x)$s, the last dimensions of the decoders' inputs $z_q(x)$s are all equal to 16 for all datasets.

Encoder: $x \in \mathbb{R}^{w\times h\times3} \rightarrow \text{Conv}_n^{(\text{kw}\times\text{kw})} \rightarrow \left[\text{EncResBlock}_n\right]_2 \rightarrow \text{MaxPool} \rightarrow \left[\text{EncResBlock}_{2*n}\right]_2 \rightarrow \text{MaxPool} \rightarrow \left[\text{EncResBlock}_{4*n}\right]_2 \rightarrow \text{Conv}_{4*n}^{(1\times1)} \rightarrow z_e(x) \in \mathbb{R}^{w/4\times h/4\times4*n}$

Decoder: $z_q(x) \in \mathbb{R}^{w/4\times h/4\times16} \rightarrow \left[\text{DecResBlock}_{4*n}\right]_2 \rightarrow \text{UpSample} \rightarrow \left[\text{DecResBlock}_{2*n}\right]_2 \rightarrow \text{UpSample} \rightarrow \left[\text{DecResBlock}_n\right]_2 \rightarrow \text{ReLU} \rightarrow \text{Conv}_3^{(1\times1)} \rightarrow \hat{x} \in \mathbb{R}^{w\times h\times3}$

where n is equal to 128 for all datasets, and $4 * n$ is equal to the number of the codebook embeddings. *kw* denotes the kernel size of the convolution layer. For CIFAR10 and CelebA kw = 3, for LSUN Church kw = 7.

Lastly, the upper bound for the annealed β coefficient of the KL divergence in dVAE is set to $5e^{-5}$ for all datasets. On the other hand, we set the upper bound for the annealed β coefficient of the KL divergence to $5e^{-7}$ for CIFAR10 while we use $1e^{-7}$ for the remaining datasets in EdVAE. We discuss the effects of the β coefficient in Appendix D.3.

Appendix D. Additional experiments

D.1. Higher temperature usage with dVAE

We use temperature values of 2 and 5 in our current experiments, and observe that perplexity value obtained as 190 for CIFAR10 dataset slightly decreases to 170 and 180, respectively. Similarly, for CelebA dataset, perplexity value obtained as 255 decreases to 217 using temperature 2, and increases to 296 using temperature 5. The performance of dVAE is sensitive to temperature hyperparameter, and perplexity does not always increase with a high temperature. Therefore, using higher temperature is not an appropriate solution.

Table D.6
Test perplexities using various max clamping values.

Max clamping value	10	15	20	25	30
CIFAR10	351	421	**425**	411	1
CelebA	319	376	**386**	1	1
LSUN Church	363	375	**393**	1	1

D.2. Effects of logits clamping

To obtain the parameters of the Dirichlet distribution, αs, we follow a common approach of logits clamping to stabilize the training since the exponential of logits might be really large. We conduct an ablation study to observe the effects of logits clamping, and present our findings in Table D.6.

We observe that clamping the logits with smaller max values clamps some of the values in logits, and limits the range of positive values logits can have. This situation limits the representativeness of the logits, and leads to lower perplexities. On the other hand, using larger max values for clamping causes divergence in the training as the exponential of logits gets large, and the model cannot be trained. Therefore, the logits should be clamped eventually with proper values. If a proper max value can be selected, clamping acts as a regularizer at the beginning of the training, and the encoder naturally outputs logits with no values greater than the max clamping value after a few iterations. If the training is already stabilized, the max clamping value does not affect the performance dramatically as both 15 and 20 lead to similar results. Therefore, using 20 as the max value can be a mutual design choice.

D.3. Effects of β coefficient

We conduct additional experiments to observe β coefficient's effects on the performance. We perform several experiments by changing the β coefficient within [1e−7, 1e−4]. We repeat our experiments for dVAE and EdVAE using all of the datasets, and present our findings in Fig. D.10.

We observe that our method is more sensitive to β coefficient than dVAE, and EdVAE diverges when the β coefficient increases. We think that the key factor to this sensitivity is the complexity introduced by the KL distance between our newly introduced posterior and prior, compared to the KL distance in dVAE. Therefore, fine-tuning β emerges. Even though our original KL term brings some sensitivity to training and it requires a hyper-parameter tuning like most of the AI models, its contribution to the performance is non-negligible and essential.

Besides, the best performing β coefficient for CIFAR10 dataset is slightly higher than the best performing β coefficient of CelebA and LSUN Church datasets. Our intuition for this difference is that, reconstructing images with lower resolution as in CIFAR10 is less challenging than reconstructing images with higher resolution as in CelebA and LSUN Church. Therefore, increasing the β coefficient from 1e−7 to 5e−7 improves the performance in CIFAR10 without hurting the reconstruction vs. KL term tradeoff. On the other hand, 1e−7 to 5e−7 conversion slightly decreases the performance in CelebA and LSUN Church datasets since the reconstruction of the higher resolution images affects the reconstruction vs. KL term tradeoff.

(a) EdVAE. **(b)** dVAE.

Fig. D.10. Effects of β coefficient to the performance.

References

[1] D.P. Kingma, M. Welling, Auto-Encoding Variational Bayes, in: ICLR, 2014.
[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, NIPS (2014).
[3] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, in: NeurIPS, 2020.
[4] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, A. Lerchner, Beta-VAE: Learning basic visual concepts with a constrained variational framework, in: ICLR, 2017.
[5] A.v.d. Oord, O. Vinyals, K. Kavukcuoglu, Neural discrete representation learning, in: NIPS, 2017.
[6] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, I. Sutskever, Zero-shot text-to-image generation, in: ICML, 2021.
[7] A. Vahdat, J. Kautz, NVAE: A deep hierarchical variational autoencoder, in: NeurIPS, 2020.
[8] R. Child, Very deep VAEs generalize autoregressive models and can outperform them on images, 2021.
[9] P. Esser, R. Rombach, B. Ommer, Taming transformers for high-resolution image synthesis, CVPR (2020).
[10] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, High-resolution image synthesis with latent diffusion models, CVPR (2021).
[11] A. Razavi, A.v.d. Oord, O. Vinyals, Generating diverse high-fidelity images with VQ-VAE-2, in: NeurIPS, 2019.
[12] M. Sun, W. Wang, X. Zhu, J. Liu, Reparameterizing and dynamically quantizing image features for image generation, Pattern Recognit. 146 (2024) 109962.
[13] Y. Takida, T. Shibuya, W. Liao, C.-H. Lai, J. Ohmura, T. Uesaka, N. Murata, S. Takahashi, T. Kumakura, Y. Mitsufuji, SQ-VAE: Variational Bayes on discrete representation with self-annealed stochastic quantization, in: ICML, 2022.
[14] M. Huh, B. Cheung, P. Agrawal, P. Isola, Straightening out the straight-through estimator: Overcoming optimization challenges in vector quantized networks, in: ICML, 2023.
[15] W. Williams, S. Ringer, J. Hughes, T. Ash, D. MacLeod, J. Dougherty, Hierarchical quantized autoencoders, in: NeurIPS, 2020.
[16] P. Dhariwal, H. Jun, C. Payne, J.W. Kim, A. Radford, I. Sutskever, Jukebox: A generative model for music, 2020, arXiv preprint arXiv:2005.00341.
[17] T. Joo, U. Chung, M.-G. Seo, Being Bayesian about categorical probability, in: ICML, 2020.
[18] M. Sensoy, L. Kaplan, M. Kandemir, Evidential deep learning to quantify classification uncertainty, in: NeurIPS, 2018.
[19] M. Kandemir, A. Akgül, M. Haussmann, G. Unal, Evidential turing processes, in: ICLR, 2022.
[20] D. Kahneman, Thinking, Fast and Slow, 2011.
[21] L. Theis, W. Shi, A. Cunningham, F. Huszár, Lossy image compression with compressive autoencoders, in: ICLR, 2017.
[22] E. Agustsson, F. Mentzer, M. Tschannen, L. Cavigelli, R. Timofte, L. Benini, L. Van Gool, Soft-to-hard vector quantization for end-to-end learning compressible representations, in: NIPS, 2017.
[23] L. Kaiser, S. Bengio, A. Roy, A. Vaswani, N. Parmar, J. Uszkoreit, N. Shazeer, Fast decoding in sequence models using discrete latent variables, in: ICML, 2018.
[24] L. Siyao, W. Yu, T. Gu, C. Lin, Q. Wang, C. Qian, C.C. Loy, Z. Liu, Bailando: 3D dance generation via actor-critic GPT with choreographic memory, in: CVPR, 2022.
[25] A. Roy, A. Vaswani, A. Neelakantan, N. Parmar, Theory and experiments on vector quantized autoencoders, 2018, arXiv:1805.11063.
[26] C.K. Sønderby, B. Poole, A. Mnih, Continuous relaxation training of discrete latent variable image models, in: Bayesian Deep Learning Workshop, NIPS, 2017.
[27] J. Pearl, Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference, 2009.
[28] R.M. Neal, Bayesian Learning for Neural Networks, Vol. 118, 2012.
[29] C.K. Sønderby, T. Raiko, L. Maaløe, S.K. Sønderby, O. Winther, Ladder variational autoencoders, in: NIPS, 2016.
[30] A.P. Soleimany, A. Amini, S. Goldman, D. Rus, S.N. Bhatia, C.W. Coley, Evidential deep learning for guided molecular property prediction and discovery, 2021.
[31] W. Bao, Q. Yu, Y. Kong, Evidential deep learning for open set action recognition, in: ICCV, 2021.
[32] C. Wang, X. Wang, J. Zhang, L. Zhang, X. Bai, X. Ning, J. Zhou, E. Hancock, Uncertainty estimation for stereo matching based on evidential deep learning, Pattern Recognit. 124 (2022) 108498.
[33] E. Jang, S. Gu, B. Poole, Categorical reparameterization with gumbel-softmax, in: ICLR, 2017.
[34] C.J. Maddison, A. Mnih, Y.W. Teh, The concrete distribution: A continuous relaxation of discrete random variables, in: ICLR, 2017.
[35] A. Amini, W. Schwarting, A. Soleimany, D. Rus, Deep evidential regression, NeurIPS (2020).
[36] A. Krizhevsky, G. Hinton, Learning Multiple Layers of Features from Tiny Images, Master's thesis, Department of Computer Science, University of Toronto, 2009.
[37] Z. Liu, P. Luo, X. Wang, X. Tang, Deep learning face attributes in the wild, in: ICCV, 2015.
[38] F. Yu, Y. Zhang, S. Song, A. Seff, J. Xiao, LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop, 2015, arXiv preprint arXiv:1506.03365.
[39] X. Chen, N. Mishra, M. Rohaninejad, P. Abbeel, Pixelsnail: An improved autoregressive generative model, in: ICML, 2018.
[40] A.v.d. Oord, N. Kalchbrenner, K. Kavukcuoglu, Pixel recurrent neural networks, in: ICLR, 2016.
[41] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: CVPR, 2009.

Gulcin Baykal received her B.Sc. and M.Sc. degrees of Computer Engineering from Istanbul Technical University (ITU) in 2018 and 2020, respectively. Gulcin is currently pursuing her Ph.D. as a Google DeepMind scholar in Computer Engineering at ITU. Her main research interests focus on representation learning, deep learning and generative models.

Melih Kandemir is an Associate Professor of Machine Learning at the University of Southern Denmark, Department of Mathematics and Computer Science. He is the founder and PI of the SDU Adaptive Intelligence Lab. His main research interests focus on Bayesian inference and stochastic process modeling of reinforcement learning and continual learning applications.

Gozde Unal is a Professor of AI and Data Engineering at Istanbul Technical University. She is the founder of ITU-AI Center. She is the recipient of the Marie Curie Alumni Association Career Award 2016 of European Commission. Her main research interests are in computer vision and deep learning, particularly representation learning and generative models.