



# Bayesian mixture variational autoencoders for multi-modal learning

Keng-Te Liao<sup>1</sup> · Bo-Wei Huang<sup>1</sup> · Chih-Chun Yang<sup>1</sup> · Shou-De Lin<sup>1</sup>

Received: 24 November 2021 / Revised: 25 August 2022 / Accepted: 16 October 2022 /  
Published online: 7 November 2022

© The Author(s), under exclusive licence to Springer Science+Business Media LLC, part of Springer Nature 2022

## Abstract

This paper provides an in-depth analysis on how to effectively acquire and generalize cross-modal knowledge for multi-modal learning. Mixture-of-Expert (MoE) and Product-of-Expert (PoE) are two popular directions in generalizing multi-modal information. Existing works based on MoE or PoE have shown notable improvement on data generation, while new challenges such as high training cost, overconfident experts, and encoding modal-specific features also emerge. In this work, we propose Bayesian mixture variational autoencoder (BMVAE) which learns to select or combine experts via Bayesian inference. We show that the proposed idea can naturally encourage models to learn modal-specific knowledge and avoid overconfident experts. Also, we show that the idea is compatible with both MoE and PoE frameworks. When being a MoE model, BMVAE can be optimized by a tight lower bound and is efficient to train. The PoE BMVAE has the same advantages and a theoretical connection to existing works. In the experiments, we show that BMVAE achieves state-of-the-art performance.

**Keywords** Multi-modal learning · Mixture-of-Gaussian · Product-of-Gaussian · Deep generative model · Disentangled representation

---

Editor: Pradeep Ravikumar.

---

✉ Keng-Te Liao  
d05922001@ntu.edu.tw

Bo-Wei Huang  
boewoei0123@gmail.com

Chih-Chun Yang  
ck20jimmy@gmail.com

Shou-De Lin  
sdlin@csie.ntu.edu.tw

<sup>1</sup> National Taiwan University, Taipei, Taiwan

# 1 Introduction

Objects or concepts in the real world can often be realized from different perspectives or modalities. For example, people can learn a new type of neural network by visually observing the architecture diagram or studying reports textually describing the model. To have comprehensive understanding, the ability to effectively generalize the knowledge across modalities is essential. While it is a trivial skill for human beings, how to make machines generalize heterogeneous information is still an open topic. In this work, we focus on learning generative models from multi-modal data without explicit supervision. The underlying challenges or criteria summarized by Shi et al. (2019) are as follows.

*Coherent joint generation* Given a randomly sampled latent vector, the model should be able to generate multi-modal data by transforming the vector and ensure the generated data describe the same objects or concepts. For example, a model can generate an arbitrary image and the associated texts describing the image content.

*Coherent cross generation* The model should be able to transfer modalities. For example, the model can generate text descriptions of a given image, and vice versa. It should also be applicable to data missing scenarios. Specifically, missing information is expected to be at least partially recovered by existing modalities.

*Latent factorization* The learned latent space can be decomposed into subspaces capturing shared and modality-specific features.

*Synergy* The quality of generation can be boosted when multiple modalities are observed.

To satisfy the criteria, one plausible solution is to build a variational encoder (Kingma & Welling, 2014) for each modality, and combine the encoders to obtain a joint posterior by Product-of-Experts (PoE) or Mixture-of-Experts (MoE) methods. MVAE (Wu & Goodman, 2018) is a representative PoE model, which has product of Gaussian experts as the joint posterior. Although MVAE does not focus on factorizing latent spaces by shared or private factors, it possesses a key benefit that the cross-modal generation can be effectively done without additional uni-modal encoders. In terms of MoE, the state-of-the-art model is MMVAE (Shi et al., 2019). MMVAE shows notable improvement over MVAE on cross-modality generation and satisfies the four proposed criteria. More importantly, experiments show that MMVAE can avoid over-confident experts which commonly exist in MVAE. However, MMVAE is relatively inefficient to train mainly due to the fact that the MoE posterior has no analytic form in most cases. To address the efficiency issue, a PoE model mmJSD (Sutter et al., 2020) has been proposed. However, the limitation being mmJSD accepts only the experts that follow Gaussian. Another recently proposed model, DMVAE (Daunhawer et al., 2020), shows remarkable performance on disentangling latent factors but is also constrained to employing Gaussian posteriors.

In general, PoE can be more efficient than MoE if all the experts are Gaussian, as there exists a closed-form joint posterior. However, for non-Gaussian experts, PoE can be intractable (Hinton, 2002). On the other hand, MoE is much easier to work with non-Gaussian experts through tractable training methods. The flexibility of assuming diverse distributions can potentially lead to better fit on the observed data.

In this work, we propose the Bayesian mixture variational autoencoder (BMVAE) for multi-modal learning. The idea comes from an assumption that uni-modal experts are not always equally reliable if modality-specific information exists. For example, an expert trained by image data is unlikely to learn sentence structures or tones of textual descriptions. Similarly, we may not expect an expert to learn all the details of images from textual

descriptions. Therefore, to achieve high-quality generation, it is necessary to rely on certain clever ways to select suitable experts. Following this idea, BMVAE is designed to select experts for each latent dimension via Bayesian inference during learning. We show that BMVAE can be implemented given both MoE and PoE frameworks. When implemented as MoE, denoted by  $\text{BMVAE}_M$ , it has a clear connection to *Bayesian Model Averaging* (Hoeting et al., 1999) and shows the following advantages over MMVAE:

- $\text{BMVAE}_M$  shows improvement over MMVAE on coherent joint generation, coherent cross generation and synergy. Regarding the latent factorization criteria,  $\text{BMVAE}_M$  naturally learns to disentangle and encode modality-specific features. Additionally, the degree of specificity is quantified for each latent dimension, making the representations more explainable.
- $\text{BMVAE}_M$  is more efficient to train. For data with  $M$  modalities, MMVAE requires  $M^2$  passes through decoders during training, while  $\text{BMVAE}_M$  only needs  $M$  passes.
- MMVAE needs to be learned by optimizing a looser lower bound, as the tighter bound empirically causes overconfident experts. We show that  $\text{BMVAE}_M$  does not need to sacrifice the theoretical tightness to avoid the overconfidence issue.

For PoE-based BMVAE, denoted by  $\text{BMVAE}_p$ , we present connections between the proposed and existing models. Specifically, we show that the model can be regarded as generalized mmJSD with stochastic weights and a different sampling strategy. We also show that  $\text{BMVAE}_p$  and  $\text{BMVAE}_M$  can have equivalent joint posteriors in specific settings.

## 2 Background

Section 2.1 is an overview of the recently proposed works. Section 2.2 briefly reviews the idea of joint posteriors factorization via MoE and PoE. Sections 2.3 and 2.4 are the fundamentals of MMVAE. Section 2.5 introduces the mmJSD objectives.

### 2.1 Overview of recent works

Variational autoencoders (VAE) have been shown to be effective for multi-modal learning. An example is JMVAE (Suzuki et al., 2017) which is designed to model the joint distribution of modalities by following the VAE framework. Also, the model has the ability to handle missing modality at test (or prediction) time. The cost of handling missing data is that it requires an additional uni-modal encoder for each modality, and the uni-modal encoders are optimized to approximate the joint distribution. The TELBO model proposed by Vedantam et al. (2018) can be learned by optimizing a different objective, which can handle partially observed features by PoE inference networks. To obtain more effective latent factorization, Hsu and Glass (2018) propose PVAE which specifically learns shared and modality-specific representations. Similar to JMVAE, PVAE requires additionally training uni-modal encoders to handle missing modalities. Another work, MFM (Tsai et al., 2019), factorizes latent representations into discriminative and modality-specific factors, where discriminative factors are learned from labeled data. Instead of employing VAE, the authors propose multi-modal Wasserstein autoencoder for inference.

The works introduced above need additional data information, learning objectives, or uni-modal networks, which is a less ideal setting as discussed by Shi et al. (2019).

By contrast, a more compact solution is MVAE proposed by Wu and Goodman (2018). MVAE is designed to model joint posterior via PoE architectures. Different from TELBO, an expert in MVAE is responsible for a whole modality instead of a single feature. As a result, MVAE shows competitive performance and does not need additional uni-modal encoders. Shi et al. (2019) report that MVAE could severely suffer from over-confident experts due to the product operations and propose MMVAE which models the joint posterior by MoE methods. Empirical results show that MMVAE can avoid aforementioned issues and outperform MVAE on data generation and modality transferring. Although MMVAE has several advantages over MVAE, it is less efficient to train. Since the joint posterior generated by MoE generally has no analytic form, the learning process would then rely on sampling-based methods having higher training cost. Besides, for data with  $M$  modalities, MMVAE requires  $M^2$  passes through decoders during training. More recently, Sutter et al. (2020) propose mmJSD to address the efficiency issue by focusing on PoE-based dynamic prior. Instead of employing KL divergence for regularization, mmJSD applies JS divergence for both regularization and learning multi-modal information. To achieve efficient learning, mmJSD requires the prior and uni-modal posteriors to be Gaussian. Another PoE model proposed recently is DMVAE (Daunhawer et al., 2020). DMVAE focuses on not only efficient learning but also disentangling latent factors. Different from typical multi-modal VAEs, DMVAE requires additional loss functions and adversarial training approaches to learn disentangled features.

## 2.2 Factorized joint posteriors

Here we introduce the idea of factorizing posteriors via MoE and PoE for multi-modal learning. Given  $M$ -modalities data  $\{x_1, \dots, x_M\}$  or  $x_{1:M}$  for training, the parameterized likelihood  $p_\theta$  and posterior  $q_\phi$  are commonly modelled by deep neural networks with  $\Theta = \{\theta_1, \dots, \theta_M\}$  and  $\Phi = \{\phi_1, \dots, \phi_M\}$ , respectively. In MMVAE, the joint posterior  $q_\phi$  is designed to be factorized by a uniform combination of uni-modal posteriors. Namely,  $q_\phi(z | x_{1:M}) = \sum_m \alpha_m \cdot q_{\phi_m}(z | x_m)$ , where  $\alpha_m = \frac{1}{M}$ . Each uni-modal posterior  $q_{\phi_m}$  is implemented by an encoder with corresponding modality as input.

Regarding PoE-based factorization, the most common form would be  $q_\phi(z | x_{1:M}) = \prod_m q_{\phi_m}(z | x_m)$ . In PoE settings, the experts are usually assumed to be Gaussian, as the joint distribution will also be Gaussian and thus can be learned efficiently. If the experts are non-Gaussian, training the PoE model generally becomes intractable (Hinton, 2002). On the contrary, we note that MoE models are more flexible on selecting distributions to fit data. If the weights  $\alpha_{1:M}$  are constants or learnable parameters, the MoE joint posterior  $q_\phi(z | x_{1:M})$  can be easily trained if sampling from  $q_{\phi_m}(z | x_m)$  is efficient. Specifically, the gradient for learning  $\phi_m$  can be estimated via Monte Carlo methods which can be applied to a wide range of distributions (Mohamed et al., 2020).

Another difference between MoE and PoE is the overconfidence issue. Shi et al. (2019) show that PoE could suffer from over-confident experts, which empirically leads to weak performance on modality transfer. Also, in order to let a PoE model be aware of missing modalities, training tricks involving artificial sub-sampling may be necessary (Wu & Goodman, 2018).

### 2.3 Importance weighted autoencoder for multi-modal learning

In the work of MMVAE, Shi et al. suggest that importance weighted autoencoder (IWAE) (Burda et al., 2016) could be more effective than vanilla VAE in multi-modal learning. Equation 1 is the objective function of IWAE for data with  $M$ -modalities.

$$\mathcal{L}_I = \mathbb{E}_{z^{1:K} \sim q_{\Phi}(z|x_{1:M})} \left[ \log \sum_{k=1}^K \frac{1}{K} \frac{p_{\Theta}(z^k, x_{1:M})}{q_{\Phi}(z^k | x_{1:M})} \right] \quad (1)$$

In Eq. 1, the hyper-parameter  $K$  is the number of samples from posterior  $q_{\Phi}$ . Burda et al. (2016) theoretically prove that higher  $K$  improves tightness of a lower bound for variational inference. It potentially enhances the model to learn more informative latent representations and achieve better performance on data generation. When  $K = 1$ , Eq. 1 is equivalent to the vanilla VAE.

In addition to improved performance over vanilla VAE in general cases, Shi et al. (2019) suggest that IWAE can be especially beneficial in multi-modal learning. Specifically, the estimated posteriors tend to have higher entropy, encouraging an uni-modal posterior (i.e.  $q_{\phi_m}$ ) to assign higher probability to regions of other modalities.

### 2.4 Learning objectives of MMVAE

Here we briefly introduce MMVAE. Equation 2 is the proposed objective function for training. The function  $\mathcal{L}_M$  has been shown to be a lower bound of log likelihood of observed data, i.e.  $\mathcal{L}_M \leq \log p(x_{1:M})$ .

$$\mathcal{L}_M = \frac{1}{M} \sum_{m=1}^M \mathbb{E}_{z_m^{1:K} \sim q_{\phi_m}} \left[ \log \frac{1}{K} \sum_{k=1}^K \frac{p_{\Theta}(z_m^k, x_{1:M})}{q_{\Phi}(z_m^k | x_{1:M})} \right] \quad (2)$$

Shi et al. (2019) also reveal that there exists a tighter lower bound as shown in Eq. 3, where  $L = K/M$  for having the same number of samples as Eq. 2.

$$\mathcal{L}_T = \mathbb{E}_{\substack{z_1^{1:L} \sim q_{\phi_1} \\ \vdots \\ z_M^{1:L} \sim q_{\phi_M}}} \left[ \log \frac{1}{M} \sum_{m=1}^M \frac{1}{L} \sum_{l=1}^L \frac{p_{\Theta}(z_m^l, x_{1:M})}{q_{\Phi}(z_m^l | x_{1:M})} \right] \quad (3)$$

Optimizing  $\mathcal{L}_T$  is more effective theoretically as  $\mathcal{L}_M \leq \mathcal{L}_T \leq \log p(x_{1:M})$ . However, empirical results show that optimizing  $\mathcal{L}_T$  can lead to modality collapse in MMVAE, significantly degrading performance on multi-modal data generation. Specifically, the joint posterior ignores most experts during training and is then reduced to a uni-modal posterior generally.

A possible reason for the collapse could be the weights for gradients in IWAE. For example, let  $z_1^{1:K}$  and  $z_2^{1:K}$  be latent vectors sampled from  $q_{\phi_1}$  and  $q_{\phi_2}$  respectively. In  $\mathcal{L}_T$ ,  $z_1^{1:K}$  and  $z_2^{1:K}$  can simultaneously exist inside the log summation. By the weight mechanism of IWAE,  $z_1^{1:K}$  and  $z_2^{1:K}$  receive gradients with different weights according to their contributions to the likelihood. If modality 1 has less contribution,  $q_{\phi_1}$  would be gradually ignored due to decreasing gradients. On the contrary, in  $\mathcal{L}_M$ , samples inside the log summation must come from the

same modality. The effect is that gradients from different modalities are forced to be equally weighed, preventing a modality from fading out due to weak update signals.

## 2.5 The mmJSD learning objectives

### 2.5.1 Standard mmJSD objective

Sutter et al. (2020) recently propose mmJSD as an objective for multi-modal learning. The differences between mmJSD and previous works are twofold. Firstly, the evidence lower bound (ELBO) is optimized via JS instead of KL divergence. Secondly, the joint posterior is combined with priors and serves as a so-called *dynamic prior*. Equation 4 is the objective, where  $\sum_{m=1}^P \pi_m = 1$  and function  $f_{\mathcal{M}}$  defines a mixture distribution averaging uni-modal  $q_{\phi_m}$  and parameterized prior  $p_{\theta}(z)$ .

$$\begin{aligned} & \mathbb{E}_{q_{\phi}} \left[ \log p_{\theta}(x_{1:M} | z) \right] - JS_{\pi}^{M+1}(\{q_{\phi_m}(z | x_m)\}_{m=1}^M, p_{\theta}(z)) \\ & JS_{\pi}^P(\{r_m(z)\}_{m=1}^P) = \sum_{m=1}^P \pi_m D_{KL}(r_m(z) | f_{\mathcal{M}}(\{r_v(z)\}_{v=1}^P)) \end{aligned} \tag{4}$$

For efficient training,  $f_{\mathcal{M}}$  is restricted to be product of Gaussian. Namely,  $f_{\mathcal{M}}(\{q_{\phi_m}(z | x_m)\}_{m=1}^M, p_{\theta}(z)) = \prod_{m=1}^M q_{\phi_m}^{\pi_m} p_{\theta}^{\pi_{M+1}}$ , where  $q_{\phi_m}$  and  $p_{\theta}$  are all Gaussian.

### 2.5.2 Modality-specific mmJSD objective

Sutter et al. (2020) also propose a variant of mmJSD focusing on learning shared and modality-specific latent factors. The idea is to let latent vectors  $z$  be concatenation of sub-vectors  $\{s_m\}_{m=1}^M$  and  $c$ , where  $\{s_m\}$  encodes features specific to the  $m$ -th modality, and  $c$  encodes modality-independent information. Equation 5 is the objective.

$$\begin{aligned} & \sum_{m=1}^M \mathbb{E}_{q_{\phi_c}(c|x_{1:M})} \left[ \mathbb{E}_{q_{\phi_{s_m}}} \left[ \log p_{\theta}(x_m | s_m, c) \right] \right] \\ & - \sum_{m=1}^M D_{KL}(q_{\phi_{s_m}}(s_m | x_m) || p_{\theta}(s_m)) - JS_{\pi}^{M+1}(\{q_{\phi_{c_m}}(c | x_m)\}_{m=1}^M, p_{\theta}(c)) \end{aligned} \tag{5}$$

Although the objective is in a form of mmJSD, Sutter et al. show that the idea can also work on MMVAE and MVAE.

A limitation of the objective is that  $\{s_m\}_{m=1}^M$  are constrained to have the same number of dimensions despite the fact that some modalities might be more complex than others. Besides, deciding the number of dimensions of  $\{s_m\}_{m=1}^M$  or  $c$  requires additional experiments for validation.

### 3 The MoE Bayesian mixture variational autoencoder

In this section, we introduce the learning algorithms of BMVAE<sub>M</sub>. The main ideas, dimension-wise mixture, stochastic weight inference and explicit regularization, are introduced in Sects. 3.1–3.3. The introductions are based on IWAE for generality and can easily fit VAE by setting  $K = 1$ .

#### 3.1 Dimension-wise MoE mixture

We first introduce the joint posterior in BMVAE<sub>M</sub>. We follow the MoE framework but propose a different algorithm from MMVAE for mixing uni-modal experts. We note that the proposed method can naturally fit a tight lower bound  $\mathcal{L}_T$  without modality collapse and is more computationally efficient.

In our method, the joint posterior  $q_\phi$  is factorized by not only modality but also latent factor. Equation 6 is the factorization, where  $D$  is the number of dimensions of a latent vector.

$$q_\phi(z | x_{1:M}) = \prod_{d=1}^D q_{\phi,d}(z_d | x_{1:M}) \quad (6)$$

$$q_{\phi,d}(z_d | x_{1:M}) = \sum_{m=1}^M \alpha_{m,d} \cdot q_{\phi_{m,d}}(z_d | x_m)$$

Conceptually, we create  $D$  expert sets where each set has  $M$  experts to decide the value of one latent factor. It differs from MMVAE in two ways. The first difference is each latent factor has its own mixture weights  $\alpha_{m,d}$  which can be  $\frac{1}{M}$  or learned from data. The algorithms for learning the weights are discussed in Sect. 3.2. The second difference lies in the individually sampled latent factor. Specifically, the mixture weights  $\alpha_{m,d}$  form a categorical distribution  $C_{\alpha_d}(m)$  for latent dimension  $d$ . When we want to obtain a sample  $z$  from joint posterior  $q_\phi$ , the sampling process repeats Eq. 7 for  $d = 1, \dots, D$ . Afterwards, we can concatenate the sampled  $z_1, \dots, z_D$  to get  $z$ .

$$m \sim C_{\alpha_d}(m)$$

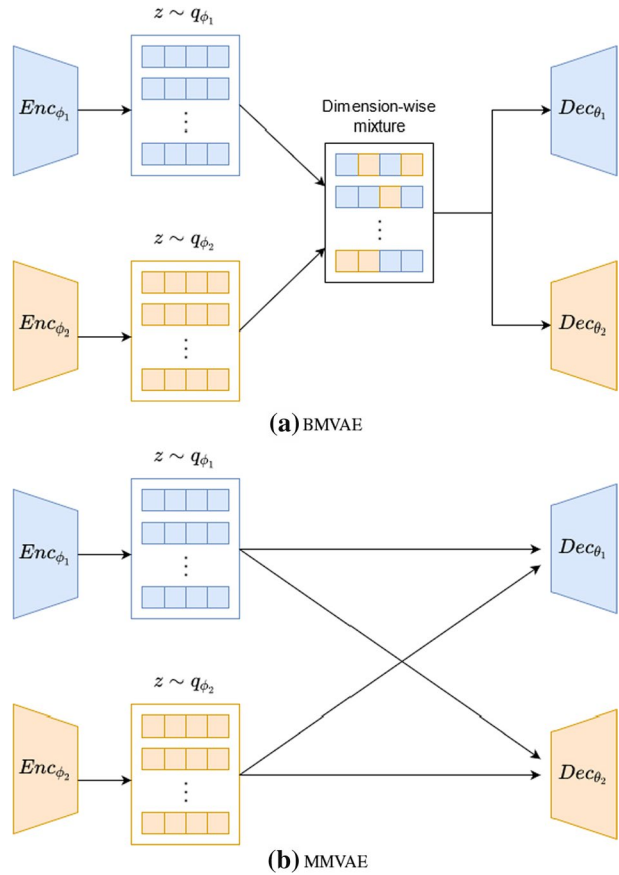
$$z_d \sim q_{\phi_{m,d}}(z_d | x_m) \quad (7)$$

To train the IWAE, we conduct standard IWAE objective (i.e.  $\mathcal{L}_I$ ) with  $z^{1:K}$  sampled by the proposed method and apply reparameterization. Note that the number of parameters of encoders and decoders is the same as it is in MMVAE, as each expert here is responsible for only one factor. The comparison of encoding-decoding between BMVAE and MMVAE is illustrated in Fig. 1.

From Fig. 1a, we can see that a sampled  $z$  in BMVAE<sub>M</sub> is composed of factors generated by randomly chosen encoders. It can be regarded as a simulation of modality missing in training time. Additionally, we note that this mechanism has two merits.

**Optimization with tight lower bound** It can be seen that latent vectors sampled by our methods naturally contain outputs from multiple modalities. By taking derivative of the log summation term in  $\mathcal{L}_I$ , gradients through the outputs can be weighed differently. Compared with MMVAE optimized by  $\mathcal{L}_T$ , we find BMVAE<sub>M</sub> optimized by  $\mathcal{L}_I$  does not suffer from modality collapse. The difference could come from the stochastic selection of

**Fig. 1** Comparison of encoding and decoding between BMVAE and MMVAE



experts. In  $\mathcal{L}_T$ , samples from all the experts simultaneously exist for decoding. If one of the experts, say  $q_{\phi_1}$ , is relatively powerful, the model could choose to rely on  $q_{\phi_1}$  and ignore  $q_{\phi_{2:M}}$ . On the other hand, in  $BMVAE_M$ , experts are selected by binary indicators sampled from categorical distribution  $C_{\alpha_d}(m)$ . If  $\alpha_d$  are all close to  $\frac{1}{M}$ , relying on a single modality could be risky, as there is a chance the corresponding expert is not selected. Therefore, the model does have motivation to make all experts similarly capable.

**Reduced computational cost** As illustrated in Fig. 1a,  $BMVAE_M$  is similar to a multi-task autoencoder generating heterogeneous data by decoding a given latent code. Specifically, decoding during training  $BMVAE_M$  is done by computing  $p_{\theta_1}(x | z)$  and  $p_{\theta_2}(x | z)$ , where  $z$  is one of the sampled latent vectors. In MMVAE, decoding is done by computing  $p_{\theta_1}(x | z_1)$ ,  $p_{\theta_1}(x | z_2)$ ,  $p_{\theta_2}(x | z_1)$  and  $p_{\theta_2}(x | z_2)$ , where  $z_1$  and  $z_2$  are sampled from modality 1 and 2, respectively. In summary, MMVAE requires  $M^2$  passes through decoders while  $BMVAE_M$  needs  $M$  passes.



### 3.2 Stochastic inference on mixture weights

In MMVAE, the mixture weights are constant  $\frac{1}{M}$ , which is reasonable as favoring a specific modality without concrete evidence could result in overconfident experts. In BMVAE<sub>M</sub>, we note that using multiple sets of experts allows imbalanced mixture weights. In reality, the imbalance could have a connection with finding shared and private latent subspaces, which is beneficial to multi-modal learning. For example, assuming latent factor  $d$  encodes private features of modality 1, it is intuitive to expect  $q_{\phi_{1,d}}$  to be able to make more reliable predictions and should have higher credibility, namely,  $\alpha_{1,d} > \frac{1}{M}$ .

To determine parameters  $\alpha_{m,d}$ , we propose learning them jointly with  $\Theta$  and  $\Phi$  by following the same IWAE objective. We assume  $C_\alpha = \prod_{d=1}^D C_{\alpha_d}$ , and denote a collection of  $D$  binary indicators sampled from  $C_{\alpha_{1:D}}$  respectively as a symbol  $m^*$ . The objective function after incorporating modality selection is shown in Eq. 8.

$$\begin{aligned} \mathbb{E}_{\substack{z^{1:K} \sim q_\Phi \\ m^{*1:K} \sim C_\alpha}} & \left[ \log \sum_{k=1}^K \frac{1}{K} \frac{p_\Theta(z^k, m^{*k}, x_{1:M})}{q_\Phi(z^k, m^{*k} | x_{1:M})} \right] \\ & = \mathbb{E}_{\substack{z^{1:K} \sim q_\Phi \\ m^{*1:K} \sim C_\alpha}} \left[ \log \sum_{k=1}^K \frac{1}{K} \frac{p_\Theta(x_{1:M} | z^k, m^{*k}) p(z^k) C(m^{*k})}{q_\Phi(z^k | x_{1:M}, m^{*k}) C_\alpha(m^{*k})} \right] \end{aligned} \tag{8}$$

In Eq. 8, there are two additional assumptions. Firstly, we assume the prior  $p(z)$  is independent to modalities. Hence,  $p(z | m^*) = p(z)$ . Secondly, the categorical prior  $C$  can be factorized as  $C = \prod_{d=1}^D C_d$ . The parameters (i.e. mixture weights) of  $C_d$  are all constants  $\frac{1}{M}$ . The reason for choosing  $\frac{1}{M}$  is to provide uninformative prior. Also, it could encourage experts from different modalities to find shared features, which can improve performance on cross-modality generation.

To learn  $C_\alpha$ , we employ differentiable Gumbel-Softmax (Jang et al., 2017; Maddison et al., 2017) to approximate discrete samples generated by categorical distributions. In this work, we allocate  $M$  learnable parameters (i.e.  $\alpha_d$ ) for each latent dimension. An indicator,  $m$ , for selecting modality in dimension  $d$  is sampled by Eq. 9 for  $i = 1, \dots, M$  with temperature  $\tau_d$ . The sampled  $M$ -dimensional vector is followed by straight-through trick (Jang et al., 2017) for being discrete. The general idea of the trick is to create a constant vector  $m_c$  having the same size and values as  $m_{1:M}$ . A discrete indicator is then obtained via  $\text{one\_hot}(m_c) + m_{1:M} - m_c$ . The result is one-hot encoded but the gradient will only pass through continuous  $m_{1:M}$  since  $m_c$  is constant.

$$\begin{aligned} m_i &= \frac{\exp((\log(\alpha_{i,d}) + g_i)/\tau_d)}{\sum_{j=1}^M \exp((\log(\alpha_{j,d}) + g_j)/\tau_d)} \\ g &= -\log(-\log(u)) \\ u &\sim \text{Uniform}(0, 1) \end{aligned} \tag{9}$$

Another issue of learning with Eq. 9 is deciding the temperature  $\tau_d$ . High temperature can make the Gumbel-Softmax distribution more continuous and uniform. It helps finding shared information across modalities; however, it also increases the difficulty of encoding private features. On the contrary, low temperature leads to a more discrete distribution, which discourages the model from finding shared information. In order to find both

shared and private information, we propose *dimension-wise temperature*, which includes both high and low  $\tau_d$ . We first decide lower and higher temperature bounds  $l$  and  $u$ . Then, we let  $\tau_d = l + \frac{(u-l)(d-1)}{D-1}$  to ensure both high and low  $\tau_d$  are included. As a result, the model can thereby find appropriate dimensions for encoding shared and private features.

### 3.3 Explicit regularization for inference

We find explicitly controlling optimization of mixture weights can make training easier and help strengthen shared features. To do this, we approximately decompose Eq. 8 into two terms as shown in Eq. 10.

$$\begin{aligned}
 F &= \mathbb{E}_{\substack{z^{1:K} \sim q_\Phi \\ m^{*1:K} \sim C_\alpha}} \left[ \log \sum_{k=1}^K \frac{1}{K} \frac{p_\Theta(x_{1:M} | z^k, m^{*k}) p(z^k)}{q_\Phi(z^k | x_{1:M}, m^{*k})} \right] \\
 G &= \mathbb{E}_{m^{*1:K} \sim C_\alpha} \left[ \log \sum_{k=1}^K \frac{1}{K} \frac{C(m^{*k})}{C_\alpha(m^{*k})} \right]
 \end{aligned}
 \tag{10}$$

It can be observed that  $G$  is negative KL divergence  $D_{KL}(C_\alpha \| C)$  if  $K = 1$ . In practice, we find optimizing  $D_{KL}(C_\alpha \| C)$  where  $C$  and  $C_\alpha$  are both categorical distributions is more numerically stable, as there is an analytic form. The final objective of BMVAE<sub>M</sub> is Eq. 11, where the hyper-parameter  $\lambda_c$  reweighs strength of the divergence. If  $\lambda_c = 1$ ,  $\mathcal{L}_B$  is approximately the derived result in Eq. 8.

$$\mathcal{L}_B = F - \lambda_c \cdot D_{KL}(C_\alpha \| C)
 \tag{11}$$

In practice, we note that models trained with  $\lambda_c = 1$  do not necessarily realize multi-modal learning. For example, a model may learn to set  $\alpha_{m,d}$  to either 1 or 0 for all  $d$ . Given  $\sum_{m=1}^M \alpha_{m,d} = 1$ , the binary weights turn BMVAE<sub>M</sub> into a concatenation of multiple uni-modal IWAE. To fix this, we simply raise the strength of  $D_{KL}(C_\alpha \| C)$  to encourage  $\alpha_{m,d} \approx \frac{1}{M}$ . We find  $\lambda_c$  in [10, 15] helps the model learn both shared and private features from data.

## 4 The PoE Bayesian mixture variational autoencoder

In this section, we present our PoE-based BMVAE, namely BMVAE<sub>p</sub>. The proposed model can be regarded as a generalized mmJSD with stochastic weights and a different sampling strategy. In Sect. 4.1, we first reveal an alternative form of mmJSD. The learning algorithm and objective of BMVAE<sub>p</sub> are introduced in Sect. 4.2. Finally, we show that there exists an equivalence between our MoE and PoE-based posteriors in Sect. 4.3.

### 4.1 The alternative form of mmJSD

Our PoE-based BMVAE is derived from the standard mmJSD objective but in a different form. Specifically, we find Eq. 4 is equivalent to the common ELBO with additional objectives.

Let the PoE joint posterior  $\prod_{m=1}^M q_{\phi_m}$  be  $\bar{q}$ ,  $\pi_q = \sum_{m=1}^M \pi_m$ , and  $\pi_z = 1 - \pi_q$ . The JS divergence term in Eq. 4 can be reorganized as follows.

$$\begin{aligned}
 JS_{\pi}^{M+1}(\{q_{\phi_m}\}_{m=1}^M, p_{\theta}(z)) &= \sum_{m=1}^M \pi_m D_{KL}(q_{\phi_m} || \bar{q}^{\pi_q} p_{\theta}^{\pi_z}(z)) + \pi_z D_{KL}(p_{\theta}(z) || \bar{q}^{\pi_q} p_{\theta}^{\pi_z}(z)) \\
 &= \sum_{m=1}^M \pi_m \int_z q_{\phi_m} \log \frac{q_{\phi_m}^{\pi_q} q_{\phi_m}^{\pi_z}}{\bar{q}^{\pi_q} p_{\theta}^{\pi_z}(z)} dz + \pi_z \int_z p_{\theta}(z) \log \frac{p_{\theta}^{\pi_q}(z) p_{\theta}^{\pi_z}(z)}{\bar{q}^{\pi_q} p_{\theta}^{\pi_z}(z)} dz \\
 &= \sum_{m=1}^M \pi_m (\pi_q D_{KL}(q_{\phi_m} || \bar{q}) + \pi_z D_{KL}(q_{\phi_m} || p_{\theta}(z))) + \pi_z \pi_q D_{KL}(p_{\theta}(z) || \bar{q})
 \end{aligned}
 \tag{12}$$

By combining Eq. 12 with Eq. 4, we obtain the following objective equivalent to the standard mmJSD.

$$\begin{aligned}
 &\underbrace{\mathbb{E}_{z \sim q_{\phi}} [\log p_{\theta}(x_{1:M} | z)] - \sum_{m=1}^M \pi_m \pi_z D_{KL}(q_{\phi_m} || p_{\theta}(z))}_{\text{Multi-modal ELBO}} \\
 &\quad - \underbrace{\sum_{m=1}^M \pi_m \pi_q D_{KL}(q_{\phi_m} || \bar{q}) - \pi_z \pi_q D_{KL}(p_{\theta}(z) || \bar{q})}_{\text{Additional objectives}}
 \end{aligned}
 \tag{13}$$

Given that  $p_{\theta}(z)$  mainly encodes prior knowledge of data, the first part of Eq. 13 simply combines  $M$  uni-modal VAEs without explicit cross-modalities alignment. Therefore, it can be seen that multi-modal learning is mostly realized by the additional objectives. The realization is also intuitive. The first KL divergence encourages uni-modal posteriors to learn cross-modality information from the joint posterior  $\bar{q}$ , and the second one regularizes the learned  $\bar{q}$ .

### 4.2 Stochastic and dimension-wise weights

Similar to the idea presented in Sect. 3, we propose inferring weights by latent dimension. To do this, we let the non-negative weights  $\pi_{1:D} = \{\pi_1, \dots, \pi_D\}$  be  $D$   $M$ -dimensional random variables instead of constant vectors. The weight  $\pi_z$  remains a hyper-parameter. For dimension  $d$ , there is a constraint  $\sum_{m=1}^M \pi_{m,d} = 1$ . Note that this constraint seems to violate the mmJSD objective since  $\sum_{m=1}^M \pi_{m,d} + \pi_z > 1$ . However, as shown in Eq. 13, the violation simply increases the strength of regularization and additional objectives, which can be easily fixed by re-scaling the KL divergence terms.

To define a PoE-based joint posterior, we follow Eq. 6 but employ geometric rather than arithmetic mean. Given that  $\pi_{1:M}$  are not necessarily discrete, we propose sampling  $\pi_{1:M}$  from parameterized Dirichlet distributions. Equation 14 is the defined posterior in dimension  $d$ , where  $\beta_d \in \mathbb{R}_{>0}^M$  are learnable parameters and inferred via reparameterization gradient for Dirichlet (Figurnov et al., 2018).

$$\begin{aligned}
 q_{\phi,d}(z_d | x_{1:M}) &= \prod_{m=1}^M q_{\phi_{m,d}}^{\pi_{m,d}}(z_d | x_m) \\
 \pi_{1:M,d} &\sim \text{Dir}(\beta_d)
 \end{aligned}
 \tag{14}$$

With the defined joint posterior, we present the objective  $\mathcal{L}_P$  for learning BMVAE<sub>p</sub> in Eq. 15, where  $\bar{q}$  is constructed by Eq. 14,  $\lambda_{ab} = \pi_a \pi_b$ ,  $\eta$  and  $\beta_{prior}$  are hyper-parameters.

$$\begin{aligned}
 \mathcal{L}_P &= \mathbb{E}_{\pi \sim \text{Dir}} \underbrace{\left[ \mathbb{E}_{z \sim \bar{q}} \left[ \log p_{\theta}(x_{1:M}|z) \right] - \lambda_{zq} D_{KL}(\bar{q} || p_{\theta}(z)) \right]}_{\text{Multi-modal ELBO}} \\
 &\quad - \sum_{m=1}^M \lambda_{mq} D_{KL}(q_{\phi_m} || \bar{q}) \\
 &\quad - \sum_{m=1}^M \lambda_{mz} D_{KL}(q_{\phi_m} || p_{\theta}(z)) \\
 &\quad - \eta \sum_{d=1}^D D_{KL}(\text{Dir}(\beta_d) || \text{Dir}(\beta_{prior}))
 \end{aligned}
 \tag{15}$$

There are two differences between  $\mathcal{L}_P$  and the alternative form of mmJSD (i.e. Eq. 13). The first one is the newly included KL divergence for inferring Dirichlet distributions. The second difference is we introduce weighted product of posteriors. Here we also reverse  $D_{KL}(p_{\theta}(z) || \bar{q})$  to be consistent with the common ELBO form. The reason for shifting to multi-modal ELBO is that we find the learned  $\pi$  are not meaningful when training with uni-modal ELBO. To see this, in uni-modal ELBO, the gradient propagated from reconstruction loss is not through  $\pi$ . Therefore, the learned  $\pi$  only involves  $D_{KL}(q_{\phi_m} || \bar{q})$  and  $D_{KL}(\text{Dir}(\beta_d) || \beta_{prior})$ , which includes insufficient data information.

### 4.3 On the equivalence of MoE and PoE-based joint posteriors

We show that there exists an equivalence between MoE and PoE joint posteriors in BMVAE. The equivalence originates from parameter settings of Dirichlet distributions. Let the weights  $\pi_d \sim \text{Dir}(\beta_d)$  and  $\beta_{m,d}$  for all  $m \in \{1, \dots, M\}$  have similar values for simplicity. When  $\beta_{m,d} > 1$ ,  $\pi_d$  are likely to be close to the centre of the  $(M - 1)$ -simplex in a sense of continuous weights (i.e.  $\pi_{m,d} \approx \frac{1}{M}$ ). On the contrary,  $\pi_d$  sampled with  $\beta_{m,d} < 1$  tend to be in corners of the simplex, resulting in nearly discrete weights. Specifically, if  $\beta_{m,d} \rightarrow 0$ ,  $\pi_{m,d}$  is either 1 or 0.

When  $\pi_{m,d}$  are discrete, the MoE and PoE mixture methods of BMVAE become similar, as they both learn to stochastically select one expert for each latent dimension. Therefore, training BMVAE<sub>p</sub> with a constraint  $\beta_d \ll 1$  would be close to training BMVAE<sub>M</sub> with additional objectives  $D_{KL}(q_{\phi_m} || \bar{q})$  and  $D_{KL}(q_{\phi_m} || p_{\theta}(z))$ . It first implies that BMVAE<sub>p</sub> can approximate BMVAE<sub>M</sub> by constraining  $\beta_d \ll 1$  and  $\lambda_{mq} = \lambda_{mz} = 0$ . Secondly, it may raise a question that if we can equip BMVAE<sub>M</sub> with the additional objectives to improve MoE-based multi-modal learning. However, we find the two additional objectives empirically do not improve BMVAE<sub>M</sub> on the evaluation tasks we reveal in Sect. 6. Also, different from PoE, the joint posterior  $\bar{q}$  has no analytic form, increasing computational cost of  $D_{KL}(q_{\phi_m} || \bar{q})$  in the training stage.

**Table 1** Time complexity of multi-modal VAEs

	MMVAE	BMVAE <sub>M</sub>	mmJSD	BMVAE <sub>p</sub>
Encoding + decoding	$(M^2K + M)\mathcal{C}_T$	$M(K + 1)\mathcal{C}_T + O(D)$	$2M\mathcal{C}_T$	$2M\mathcal{C}_T + O(D)$
Joint posteriors	0	$O(D)$	$(M + 1) \cdot O(D)$	$M \cdot O(D)$
Sampling	$MK \cdot O(D)$	$(M + 1)K \cdot O(D)$	$O(D)$	$O(D)$
KL divergence	$MK \cdot O(D)$	$(M + 1)K \cdot O(D)$	$M \cdot O(D)$	$(2M + 1) \cdot O(D)$
Total cost	$\approx M^2K\mathcal{C}_T$	$\approx M(K + 1)\mathcal{C}_T$	$\approx 2M\mathcal{C}_T$	$\approx 2M\mathcal{C}_T$

## 5 Analysis of time and space complexity

We analyze complexity of training MMVAE, mmJSD, BMVAE<sub>M</sub> and BMVAE<sub>p</sub> in this section. We first discuss time complexity. Training the VAE-based models involves encoding, decoding, obtaining joint posteriors, sampling from posteriors, and KL divergence minimization. We define the upper bound of the cost of training an encoder or decoder as  $\mathcal{C}_T$  including forward and backward passes. As an example, the cost of training a single-modality auto-encoder without variational inference is bounded by  $\mathcal{C}_T + \mathcal{C}_T = 2\mathcal{C}_T$ . As latent dimensions are assumed to be mutually independent in the discussed models, we note that the training procedures excluding encoding and decoding can be decomposed into dimension-wise operations with complexity  $O(D)$ . The operations are enumerated as follows.

- Obtaining product of 2 Gaussian posteriors.
- Obtaining a categorical or Dirichlet posterior where the underlying encoder is a single-layer neural network.
- Sampling from a Gaussian or categorical posterior.
- Sampling from a mixture of posteriors.
- Estimating KL divergence between 2 Gaussian or categorical distributions.

With the enumerated upper bounds, we analyze training costs of the discussed models and summarize the results in Table 1.

**MMVAE** The cost of obtaining  $M$  posteriors is  $M\mathcal{C}_T$ . As MMVAE does not sample embeddings from a joint posterior but  $M$  posteriors, the cost of sampling is  $MK \cdot O(D)$  where  $K$  is the sample size of IWAE. The  $MK$  embeddings are then decoded by  $M$  decoders, costing  $M^2K\mathcal{C}_T$ . Finally, the cost of estimating the sampling-based KL divergence for the  $MK$  embeddings is  $MK \cdot O(D)$ .

**BMVAE<sub>M</sub>** The cost of obtaining  $M$  posteriors and 1 categorical posterior is  $M\mathcal{C}_T + O(D)$ . Sampling from a mixture of posteriors of an IWAE-based BMVAE<sub>M</sub> costs  $(M + 1)K \cdot O(D)$ . Different from MMVAE, the number of generated embeddings is  $K$  instead of  $MK$ . Therefore, the cost of decoding reduces to  $MK\mathcal{C}_T$ . Finally, the cost of the sampling-based KL divergence is  $(M + 1)K \cdot O(D)$  where the additional cost comes from the categorical distributions.

**mmJSD** The cost of encoding and decoding is  $2M\mathcal{C}_T$ . Obtaining the product of  $M$  Gaussian densities and a prior costs  $(M + 1) \cdot O(D)$ . We assume the product of Gaussian is both used for the joint posterior and dynamic prior. Sampling from the joint posterior costs  $O(D)$  and estimating KL divergence between uni-modal posteriors and dynamic prior costs  $M \cdot O(D)$  in total.

**Table 2** Space complexity of multi-modal VAEs

	MMVAE	BMVAE <sub>M</sub>	mmJSD	BMVAE <sub>p</sub>
Encoders and decoders	$2M\mathcal{C}_S$	$2M\mathcal{C}_S + O(D)$	$2M\mathcal{C}_S$	$2M\mathcal{C}_S + O(D)$
Sampled embeddings	$MK \cdot O(D)$	$MK \cdot O(D)$	$O(D)$	$O(D)$
Total cost	$\approx 2M\mathcal{C}_S$	$\approx 2M\mathcal{C}_S$	$\approx 2M\mathcal{C}_S$	$\approx 2M\mathcal{C}_S$

**BMVAE<sub>p</sub>** The cost of encoding and decoding is  $2M\mathcal{C}_T + O(D)$  where  $O(D)$  comes from the Dirichlet distribution. The joint posterior is the weighted product of  $M$  Gaussian densities costing  $M \cdot O(D)$ . The embeddings for decoding are sampled from the joint posterior costing  $O(D)$ . The cost of the KL divergence losses is  $(M + M + 1) \cdot O(D)$  which can be observed from Eq. 15.

For the analysis of space complexity, we define the upper bound of the cost of an encoder or decoder as  $\mathcal{C}_S$ . Other costs coming from the following components are assumed to have space complexity  $O(D)$ .

- A single-layer neural network with  $D$  neurons.
- A  $D$ -dimensional embedding sampled from a posterior.

The costs of the discussed multi-modal VAEs are summarized in Table 2.

## 6 Experiments

In Sects. 6.1 and 6.2, we compare BMVAE<sub>M</sub> with MMVAE following protocols and datasets proposed by Shi et al. (2019). The autoencoders are IWAE and the posteriors are Laplace distributions. In each evaluation task, the sample size  $K$  of IWAE is the value suggested by Shi et al. and applied to both MMVAE and BMVAE<sub>M</sub>. In Sect. 6.3, we evaluate both BMVAE<sub>M</sub> and BMVAE<sub>p</sub> on a 3-modal dataset proposed by Sutter et al. (2020). For fair comparison, the posteriors are all Gaussian and the autoencoders used in testing include both VAE and IWAE. In Sect. 6.4, we evaluate models via MultiBench (Liang et al., 2021) measuring cross-modality generalization, training speed, and robustness. The tested datasets are more challenging, and the models for evaluations are not limited to MoE and PoE methods.

In each experiment, the models for comparison have the same uni-modal encoders and decoders. Our models are all trained by Adam optimizer (Kingma & Ba, 2015) with learning rate 0.001. The parameters  $\lambda_c$  and  $\eta$  for controlling mixture weights in BMVAE<sub>M</sub> and BMVAE<sub>p</sub> are decided via validation data. The parameter search space of  $\lambda_c$  and  $\eta$  is  $\{0.01h, 0.1h, 0.5h\}$ , where  $h$  is the average size of data. For example, if the training data are all  $32 \times 32$  images,  $h = 32 \times 32$ . The remaining parameters of BMVAE<sub>p</sub>,  $\lambda_{zq}$ ,  $\lambda_{mz}$  and  $\lambda_{mq}$ , are set to be  $1, \frac{1}{M}$  and  $\frac{1}{M}$  respectively. The parameters  $\beta_{prior}$  of the Dirichlet prior is 0.5 in Sects. 6.1, 6.2, and 6.3. In Sect. 6.4, it is chosen from  $\{0.5, 1, 10\}$  via validation. The training epochs are reported in respective subsections. We note that although BMVAE seems to

**Fig. 2** Examples of MNIST-SVHN data**Table 3** MNIST-SVHN encoders and decoders. Args: (kernel size, stride, padding)

	Layer	Type	In	Out	Args
MNIST encoder	1	Linear, RELU	784	400	
	2a	Linear	400	20	
	2b	Linear	400	20	
MNIST decoder	1	Linear, RELU	20	400	
	2	Linear, Sigmoid	400	784	
SVHN encoder	1	Conv2d, RELU	3	32	(4,2,1)
	2	Conv2d, RELU	32	64	(4,2,1)
	3	Conv2d, RELU	64	128	(4,2,1)
	4a	Conv2d	128	20	(4,1,0)
	4b	Conv2d	128	20	(4,1,0)
SVHN decoder	1	Conv <sup>T</sup> 2d, RELU	20	128	(4,1,0)
	2	Conv <sup>T</sup> 2d, RELU	128	64	(4,2,1)
	3	Conv <sup>T</sup> 2d, RELU	64	32	(4,2,1)
	4	Conv <sup>T</sup> 2d, Sigmoid	32	3	(4,2,1)

be more complex than MMVAE in the aspect of mixture method, BMVAE can normally converge in fewer epochs.

## 6.1 MNIST-SVHN

The first dataset used for evaluation is MNIST-SVHN. It is constructed by pairing images depicting the same digit class from MNIST and SVHN. Examples are shown in Fig. 2. As can be seen that the modalities are both images but with distinct and more complicated styles.

Following the settings in the previous work, the encoders for MNIST and SVHN data are multi-layer perceptron (MLP) and convolutional neural network (CNN) respectively. The number of dimensions of a latent vector is 20. Likelihoods, prior and posteriors are all Laplace. The likelihoods are weighted to balance reconstruction errors. In particular, given SVHN are 3-channel 32×32 images and MNIST are 1-channel 28×28 images, the log likelihood of MNIST is multiplied by  $\frac{32 \times 32 \times 3}{28 \times 28} = 3.92$ . *Latent classification and coherence* of generations are used for evaluation as suggested by Shi et al. (2019). BMVAE and MMVAE are both trained for 30 epochs. The structures of uni-modal encoders and decoders are shown in Table 3.

### 6.1.1 Latent classification

The goal here is to examine whether the shared information (i.e. digit class) can be effectively learned by models by checking if the digits can be successfully recognized from

**Table 4** Digit classification accuracy

	BMVAE <sub>M</sub>	MMVAE	Uni-modal VAE
MNIST	<b>0.94</b>	0.91	0.85
SVHN	<b>0.76</b>	0.68	0.21

The bold values indicate the best performance in average

latent vectors by linear classifiers. The assumption is that accurate recognition implies informative representations regarding shared information. Second, implication is that the information is encoded separately across latent dimensions as shallow classifiers are sufficient for the recognition task. The classification accuracy is listed in Table 4. To show the advantages of multi-modal architectures, performance of uni-modal VAE is also included.

### 6.1.2 Coherence

Another evaluation metric is to examine coherence of joint and cross-modal generations. In general, coherence checks both shared and private information by observing the generated data. When evaluating random generation via coherence, the decoders are required to generate images given the same vector which is randomly sampled from prior. To achieve high scores, the generated images need to depict the same digit class. Also, styles of the images need to be consistent with the corresponding modality. In cross-modal coherence, the generation conditions on distinct modality rather than random noises. For example, given an SVHN image depicting digit “3”, we first input the image into SVHN encoder and let it generate a sampled latent vector. Afterwards, we let the MNIST decoder generate an image given the sampled latent vector and check if it also depicts “3” in MNIST-style. To recognize digits and styles from generated images, two CNN-based classifiers are trained by SVHN and MNIST datasets respectively. The score of coherence is estimated by the probability of correct digit matching. Evaluation results are shown in Table 5.

## 6.2 CUB image-captions

The other dataset used in the experiment is Caltech-UCSD Birds (CUB). It contains 11,788 photos of birds with captions describing birds’ visual characteristics. Examples are shown in Fig. 3. CUB dataset provides more challenges than MNIST-SVHN due to more complex data and heterogeneous modalities.

For evaluation on CUB, we follow the methods proposed by Shi et al. (2019) which focuses on coherence as CUB has no clear label information. Notably, the coherence discussed here is measured on a vector space. The details of the proposed evaluation are as follows.

**Table 5** Random and cross-modal coherence. M and S are abbreviations of MNIST and SVHN. S M and M S are generations given MNIST and SVHN, respectively

	Random	S M	M S
BMVAE <sub>M</sub>	<b>0.50</b>	<b>0.89</b>	<b>0.71</b>
MMVAE	0.42	0.86	0.69

The bold values indicate the best performance in average





**Fig. 3** Examples of CUB data

**Table 6** CUB encoders and decoders. CN: Conv2d. BN: BatchNorm2d. RE: RELU. Args: (kernel size, stride, padding)

	Layer	Type	In	Out	Args
Image encoder	1	Linear, ELU	2048	1024	
	2a	Linear, ELU	1024	512	
	2b	Linear, ELU	512	256	
	4a	Linear	256	128	
	4b	Linear	256	128	
Image decoder	1	Linear, ELU	128	256	
	2	Linear, ELU	256	512	
	3	Linear, ELU	512	1024	
	4	Linear	1024	2048	
Text encoder	1	CN, BN, RE	1	32	(4,2,1)
	2	CN, BN, RE	32	64	(4,2,1)
	3	CN, BN, RE	64	128	(4,2,1)
	4	CN, BN, RE	128	256	((1,4),(1,2),(0,1))
	5	CN, BN, RE	256	512	((1,4),(1,2),(0,1))
	6a	CN	512	128	(4,1,0)
Text decoder	6b	CN	512	128	(4,1,0)
	1	$C^T$ N, BN, RE	128	512	(4,1,0)
	2	$C^T$ N, BN, RE	512	256	((1,4),(1,2),(0,1))
	3	$C^T$ N, BN, RE	256	128	((1,4),(1,2),(0,1))
	4	$C^T$ N, BN, RE	128	64	(4,2,1)
	5	$C^T$ N, BN, RE	64	32	(4,2,1)
6	$C^T$ N, RE	32	1	(4,2,1)	

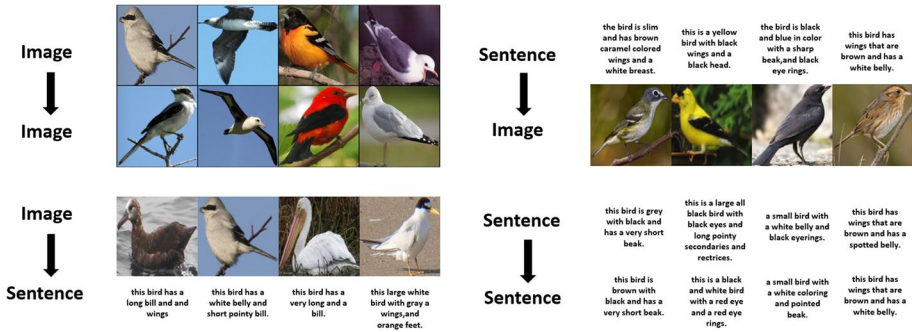
**Data generation** For image data, the decoder actually outputs vectors in feature space of a pre-trained ResNet-101 He et al. (2016) instead of real images. The motivation is to avoid generating blurry images. The method of generation and reconstruction then becomes finding the most similar vector of a real photo on the feature space by Euclidean distance. Additionally, likelihood of the image decoder is Laplace distribution. For the caption data, the encoder and decoder are based on CNN. The likelihood of decoder is Categorical distribution. For other training details, the priors and posteriors are Laplace where the number of dimensions of latent vectors is 128. The training epochs for BMVAE and MMVAE are 30 and 50. The structures of uni-modal encoders and decoders are shown in Table 6.

**Coherence** To check whether a pair of generated image and caption matches, the proposed idea is to map generated images and captions into a common vector space. To do so, Shi et al. (2019) suggested employing Canonical Correlation Analysis (CCA). Specifically,

**Table 7** Correlation of images (I) and sentences (S) generations. The ground truth of random coherence is 0.273

	Random	I S	S I
BMVAE <sub>M</sub>	<b>0.270</b>	<b>0.171</b>	<b>0.210</b>
MMVAE	0.263	0.104	0.135
Uni-modal VAE	–	0.028	0.003

The bold values indicate the best performance in average



**Fig. 4** CUB generation results

generated images are converted into 2048-dimensional feature vectors by pre-trained ResNet-101. Generated captions are converted into 300-dimensional vectors by averaging word vectors trained by FastText Bojanowski et al. (2017). To do the mapping, two projection matrices  $W_1 \in \mathbb{R}^{2048 \times 40}$  and  $W_2 \in \mathbb{R}^{300 \times 40}$  are trained by maximizing correlations between  $W_1^T y_1$  and  $W_2^T y_2$ , where  $y_1$  and  $y_2$  are the feature vectors. After training, when there are new pairs of feature vectors, we can do the projection by  $W_1$  and  $W_2$  and compute the correlation as the performance measurement. The random and cross-modal coherence results are shown in Table 7. Examples of cross-modal generation are also provided in Fig. 4.

### 6.3 MNIST-SVHN-Text

Here we evaluate our models with data having 3 modalities. The dataset for evaluation is MNIST-SVHN-Text proposed by Sutter et al. (2020). It is constructed by adding text modality on the MNIST-SVHN dataset, where the text data are character-level strings of digit names (e.g. ‘o’, ‘n’, ‘e’). In order to increase complexity of the text data, random numbers of space characters are dynamically inserted in front of the digit names.

In addition to datasets, a notable difference to the settings in Sect. 6.1 is that we conduct Gaussian instead of Laplace posteriors for fair comparisons with mmJSD and MVAE (Wu & Goodman, 2018). Models trained by the modality-specific framework reviewed in Sect. 2.5.2 are also included for comparisons. BMVAE, MVAE, MMVAE, mmJSD are trained for 50 epochs. The modality-specific versions of MVAE, MMVAE and mmJSD are trained for 100 epochs. The uni-modal encoders and decoders are shown in Table 8.

**Table 8** MNIST-SVHN-Text encoders and decoders. Args: (kernel size, stride, padding)

	Layer	Type	In	Out	Args
MNIST encoder	1	Linear, RELU	784	400	
	2a	Linear	400	20	
	2b	Linear	400	20	
MNIST decoder	1	Linear, RELU	20	400	
	2	Linear, Sigmoid	400	784	
SVHN encoder	1	Conv2d, RELU	3	32	(4,2,1)
	2	Conv2d, RELU	32	64	(4,2,1)
	3	Conv2d, RELU	64	64	(4,2,1)
	4	Conv2d, RELU	64	128	(4,2,1)
	5a	Conv2d	128	20	(4,1,0)
	5b	Conv2d	128	20	(4,1,0)
SVHN decoder	1	Conv <sup>T</sup> 2d, RELU	20	128	(4,1,0)
	2	Conv <sup>T</sup> 2d, RELU	128	64	(4,2,1)
	3	Conv <sup>T</sup> 2d, RELU	64	64	(4,2,1)
	4	Conv <sup>T</sup> 2d, RELU	64	32	(4,2,1)
	5	Conv <sup>T</sup> 2d, Sigmoid	32	3	(4,2,1)
Text encoder	1	Conv1d, RELU	71	128	(1,1,0)
	2	Conv1d, RELU	128	128	(4,2,1)
	3	Conv1d, RELU	128	128	(4,2,0)
	4a	Linear	128	20	
	4b	Linear	128	20	
Text decoder	1	Linear, RELU	20	128	
	2	Conv <sup>T</sup> 1d, RELU	128	128	(4,1,0)
	3	Conv <sup>T</sup> 1d, RELU	128	128	(4,2,1)
	4	Conv <sup>T</sup> 1d, Softmax	128	71	(1,1,0)

### 6.3.1 Latent classification

The classification task is the same as the one introduced in Sect. 6.1.1. However, we can now examine the generated samples given data with multiple modalities. To obtain samples from  $\text{BMVAE}_M$ , we first construct a categorical distribution by normalizing  $\alpha_{m,d}$  for each latent dimension. For example, if modality 1 and 3 are given, the parameters of the categorical distribution in dimension  $d$  are  $\frac{\alpha_{1,d}}{\alpha_{1,d} + \alpha_{3,d}}$  and  $\frac{\alpha_{3,d}}{\alpha_{1,d} + \alpha_{3,d}}$ . Then, we can conduct ancestral sampling to obtain samples from  $\text{BMVAE}_M$ .

To obtain samples from  $\text{BMVAE}_p$ , we first sample mixture weights from the learned Dirichlet distributions. Then, as weighted product-of-Gaussian has an analytic form, we directly construct the joint posteriors to generate samples for  $\text{BMVAE}_p$ . The evaluation results are shown in Table 9.

We first compare  $\text{BMVAE}_M$  with other MoE models, and see that  $\text{BMVAE}_M$  has a clear advantage when more modalities are given. Similar results can also be found from the comparisons between  $\text{BMVAE}_p$  and other PoE models. The advantages could indicate effectiveness of the learned weights. More concretely, the ideal experts are correctly selected or properly weighted via the learned  $\alpha$ .

**Table 9** Classification accuracy of sampled latent vectors. The provided modalities include MNIST (M), SVHN (S) and Text (T). The modality specific (MS) objectives improve performance of MVAE, MMVAE, and mmJSD. BMVAE has particular advantages when multiple modalities are available

	M	S	T	M,S	M,T	S,T	M,S,T
MVAE	0.85	0.20	0.58	0.80	0.92	0.46	0.90
MMVAE	0.96	0.81	<b>0.99</b>	0.89	0.97	0.90	0.93
mmJSD	0.97	0.82	<b>0.99</b>	0.93	<b>0.99</b>	0.92	0.98
MVAE (MS)	0.86	0.28	0.78	0.82	0.94	0.64	0.92
MMVAE (MS)	0.96	0.81	<b>0.99</b>	0.89	0.98	0.91	0.92
mmJSD (MS)	<b>0.98</b>	<b>0.85</b>	<b>0.99</b>	0.94	0.98	0.94	<b>0.99</b>
BMVAE <sub>M(K=1)</sub>	0.96	0.80	<b>0.99</b>	0.94	0.98	0.98	0.96
BMVAE <sub>M(K=10)</sub>	0.96	0.74	<b>0.99</b>	0.92	0.98	0.95	0.95
BMVAE <sub>p</sub>	0.97	0.82	<b>0.99</b>	<b>0.96</b>	<b>0.99</b>	<b>0.98</b>	0.98

The bold values indicate the best performance in average

**Table 10** Random (R) and cross-modal coherence with modalities MNIST (M), SVHN (S) and Text (T). The modality specific (MS) objectives does not always improve performance of MVAE, MMVAE, and mmJSD but is able to prevent a significant performance drop (e.g., the S M column). BMVAE does not suffer from the issue and has advantages on performance when multiple modalities are available

	R	M S	M T	M S,T	S M	S T	S M,T	T M	T S	T M,S
MVAE	0.72	0.17	0.14	0.22	0.37	0.30	0.86	0.20	0.12	0.22
MMVAE	0.54	<b>0.82</b>	<b>0.99</b>	0.91	0.32	0.30	0.31	0.96	<b>0.83</b>	0.90
mmJSD	0.60	<b>0.82</b>	<b>0.99</b>	0.95	0.37	0.36	0.48	<b>0.97</b>	<b>0.83</b>	0.92
MVAE (MS)	<b>0.74</b>	0.16	0.17	0.25	0.35	0.37	0.85	0.24	0.14	0.26
MMVAE (MS)	0.67	0.77	0.97	0.86	0.88	0.93	0.90	0.82	0.70	0.76
mmJSD (MS)	0.66	0.80	0.97	0.93	0.89	0.93	0.92	0.92	0.79	0.86
BMVAE <sub>M(K=1)</sub>	0.66	0.76	<b>0.99</b>	<b>0.97</b>	0.88	<b>0.99</b>	0.92	0.95	0.77	0.93
BMVAE <sub>M(K=10)</sub>	0.70	0.72	<b>0.99</b>	0.92	<b>0.91</b>	0.96	<b>0.94</b>	0.91	0.70	0.88
BMVAE <sub>p</sub>	0.72	0.80	<b>0.99</b>	<b>0.97</b>	0.90	0.94	0.92	<b>0.97</b>	0.80	<b>0.96</b>

The bold values indicate the best performance in average

### 6.3.2 Coherence

The experiment settings are the same as Sect. 6.1.2 with an exception that we apply a different method for random generation for BMVAE. The motivation is that we observe the joint posteriors  $q_{\phi}(z)$  are less similar to the prior  $p(z)$  after learning the data. To have more effective generation, we follow an idea proposed by Daunhawer et al. (2020) where *ex-post estimation* (Ghosh et al., 2020) is employed. Specifically, we find  $\hat{p}(z) \approx q_{\phi}(z)$  via density estimation and draw samples from  $\hat{p}(z)$  instead of  $p(z)$  for evaluating random coherence. In the experiments, we let  $\hat{p}(z)$  be a 10-component Gaussian mixture model with diagonal covariance matrices. The results are shown in Table 10.

The results in Table 10 also lead to the conclusions we made in Sect. 6.3.1 with an interesting observation on the performance of MMVAE and mmJSD. As can be seen, the two models achieve the best performance in MNIST and Text generation; however, the accuracies significantly drop to 30%–48% when generating SVHN. Therefore, the MS learning framework of the two models becomes essential. On the other hand, BMVAE<sub>M</sub> and BMVAE<sub>p</sub> both avoid this issue and do not need additional learning algorithms.

## 6.4 MultiBench

In this subsection, we consider a broader class of models, modalities, and metrics for evaluations. We select MultiBench (Liang et al., 2021), which is a benchmark scoring generalization across modalities, training or testing complexity, and robustness against noisy or missing data. The benchmark was proposed with 15 datasets covering 10 modalities and 6 research areas. With a unified pipeline handling multi-modal data processing, models based on distinct paradigms are possible to be jointly evaluated. In our experiments, we select 5 publicly available datasets officially supporting generative models for training, and we compare BMVAE with 6 competitors including but not limited to MoE and PoE methods.

### 6.4.1 Datasets and models

We select datasets that do not have restricted access issues and have been officially tested by MultiBench MVAE. The selected datasets are CMU-MOSI (Zadeh et al., 2016), UR-FUNNY (Hasan et al., 2019), CMU-MOSEI (Bagher Zadeh et al., 2018), MUSTARD (Castro et al., 2019) and AV-MNIST (Vielzeuf et al., 2018). Modalities covered by the 5 datasets are language, image, video, and audio.

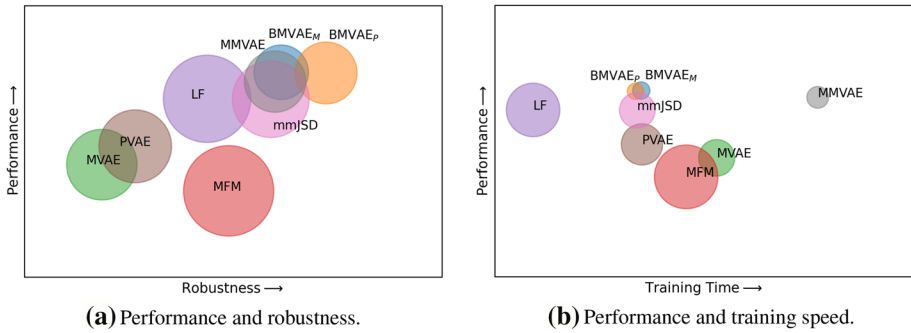
Besides MoE and PoE models, we additionally test MFM (Tsai et al., 2019), PVAE (Hsu & Glass, 2018) and late fusion (LF). As introduced in Sect. 2.1, MFM factorizes latent representations into modality-specific factors and discriminative factors encoding shared information, where the discriminative factors are learned from labels. Another model, PVAE, also learns to find modality-specific and shared factors but does not follow the MoE or PoE framework. Instead, PVAE transforms concatenated uni-modal embeddings into joint representations, requiring an additional hidden layer and separately learning uni-modal and multi-modal encoders. The third method, LF, is a baseline method adopted by MultiBench. It directly takes concatenation of uni-modal embeddings as the multi-modal representations. Despite being a simple method, it shows remarkable performance in MultiBench evaluation.

In the experiments, the architectures and parameter sizes of uni-modal encoders and decoders are the same across the tested methods, except that VAE-based methods require additional layers for reparameterization, and LF does not need decoders. The the architectures and parameters follow the released code.<sup>1</sup> The likelihoods we select are all Laplace. Regarding importance sampling, we do not set the sample size  $K > 1$  as no improved performance is observed. To obtain mean and variance of model performance, all the experiments are repeated 10 times, and the built-in early-stopping mechanism is always turned on.

### 6.4.2 Evaluation results

We follow the presentation proposed by MultiBench where the evaluation results are visualized to reveal trade-offs and provide deeper insights. The visualization results are plotted in Fig. 5a, b.

<sup>1</sup> <https://github.com/pliang279/MultiBench>.



**Fig. 5** Trade-off between performance and robustness or between performance and training speed. The size of circles in Fig. 5a represents variance of robustness. The size of circles in Fig. 5b represents variance of performance

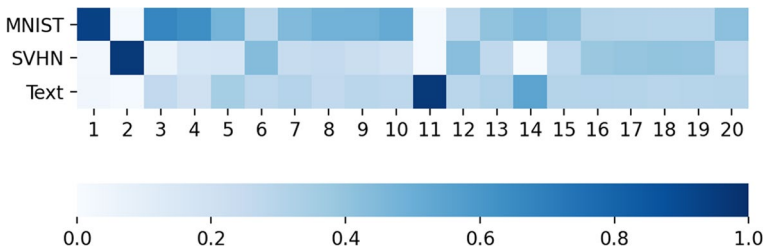
**Table 11** Log likelihood of VAE-based models, where  $M=10^6$  and  $K=10^3$ . The tested data are the 5 selected datasets and the covered modalities

		MVAE	MMVAE	mmJSD	BMVAE <sub>M</sub>	BMVAE <sub>P</sub>	PVAE
CMU-MOSI	Video	-1882	-1860	-1884	-1850	-1841	<b>-1709</b>
	Audio	-6052	-6177	-6064	-6042	-6052	<b>-5806</b>
	Language	-7828	-7914	-7838	-7914	-7895	<b>-7314</b>
UR-FUNNY	Video	-9138	-9218	-9126	-9228	-9155	<b>-8226</b>
	Audio	-2602	-2600	-2497	-2568	-2580	<b>-2055</b>
	Language	-7822	-7899	-7795	-7904	-7853	<b>-7163</b>
CMU-MOSEI	Video	-2596	-2647	-2627	-2651	-2604	<b>-2397</b>
	Audio	-9081	-9093	-9193	-9138	-9184	<b>-8844</b>
	Language	-9088	-9095	-9063	-9143	-9106	<b>-9049</b>
MUSTARD	Video	-10.3 M	-10.3 M	-10.3 M	-10.3 M	-10.3 M	-10.3 M
	Audio	-0.24 M	-0.24 M	-0.24 M	-0.24 M	-0.24 M	-0.24 M
	Language	-7801	-7789	-7812	-7935	-7799	<b>-7126</b>
AV-MNIST	Image	-25.9K	-25.8K	-26.0K	-25.1K	-25.2K	<b>-24.9K</b>
	Audio	-0.57 M	-0.57 M	-0.57 M	-0.57 M	-0.58 M	<b>-0.42 M</b>

The bold values indicate the best performance in average

Figure 5a is the result summarizing model performance and robustness. The performance scores correspond classification accuracies of predicting labels given latent representations. Robustness is tested by measuring model performance with increasingly noisy data. The robustness can be quantified via computing *relative robustness* and *effective robustness* proposed by MultiBench. Given noisy data, relative robustness directly measures model performance while effective robustness measures the rate of performance drops. We average relative and effective robustness scores as the final result, and represent variance of robustness via circle size. Figure 5b is the visualization presenting trade-off between performance and training speed. The circle size corresponds to variance of performance.

From Fig. 5a, b, we confirm that LF is a strong baseline as it shows decent performance and robustness in average, while we also note a potential weakness that the variances are relatively large. We can also observe trade-offs between LF and MFM in terms



**Fig. 6** Mixture weights learned from MNIST-SVHN-Text

of performance, robustness, and training time. The MoE models we discussed in this work show good and similar performance and robustness. We also note that the MoE models have lower variances of robustness when compared to the PoE models, mmJSD and BMVAE<sub>p</sub>. Finally, we confirm that although MMVAE can reach good performance and robustness, it requires a significantly long training time.

Finally, we report log-likelihood of the VAE-based models in Table 11 to examine performance of generation. We note that the differences are marginal except for PVAE. The reason for achieving higher log-likelihood could be the additional modality-specific representations. For example, to generate data with 3 modalities, PVAE takes joint and 3 sets of modality-specific representations for decoding, while other models only use joint representations. It can be observed that the additional representations effectively improve log-likelihood. while a potential issue would be handling unexpectedly noisy or missing data.

## 7 Analysis of mixture weights

In this section, we analyze the learned mixture weights from multiple aspects. In Sect. 7.1, we provide observations and visualizations of the learning results. Sections 7.2 and 7.3 are quantitative and qualitative analyses of the weights. The weights we select for analyses in Sects. 7.1–7.3 are learned by BMVAE<sub>M(k=1)</sub>; however, weights learned by BMVAE<sub>M(k>1)</sub> and BMVAE<sub>p</sub> also show similar characteristics and do not contradict conclusions we reach. In Sect. 7.4, we verify if BMVAE can be trained with fixed and uniform mixture weights. Takeaway messages summarizing our observations are provided in respective subsections.

### 7.1 Observations and visualizations

We take the weights learned from MNIST-SVHN-Text for analyses, since the data contain explainable attributes such as digit class, style or color scheme. The learned mixture weights are visualized in Fig. 6.

In the figure, the learned  $\alpha_{m,d}$  for  $d = 1, \dots, 20$  are presented. Note that the values are normalized probabilities, namely,  $\sum_{m=1}^3 \alpha_{m,d} = 1$ . We can then judge whether a dimension tends to encode private or public information by observing the normalized values. For example, we may conclude that dimension 1, 2 and 11 encode information specific to MNIST, SVHN and Text respectively. In contrast, dimension 12 may be more like a shared dimension with a slight preference for SVHN. We note that the preference can also

**Table 12** Digit classification accuracy using sub-vectors. The results show that the digit information is aligned and encoded in dimension  $D_{\bar{S}}$ 

	$Z_{D_{\bar{S}}}^M$	$Z_{D_{\bar{S}}}^S$	$Z_{D_{\bar{S}}}^T$	$Z_{D_S}^M$	$Z_{D_S}^S$	$Z_{D_S}^T$
Accuracy	0.95	0.78	0.99	0.25	0.15	0.30

be meaningful. Let a dimension set  $D_S$  be  $\{2, 6, 12, 16, 17, 18, 19\}$ . It can be observed that the 7 dimensions all prefer SVHN. Also, MNIST and Text are equally weak in  $D_S$ . With these observations we make two assumptions:

**Assumption 1**  $D_S$  is mainly responsible for encoding image styles specific to SVHN, which can explain why MNIST and Text are equally uninformative here. It may also imply that the specific styles are irrelevant to digit classes.

**Assumption 2** Given that dimensions in  $D_S$  are occupied, the information relevant to digit recognition tends to be encoded in the remaining 13 dimensions.

We denote the set of 13 dimensions as  $D_{\bar{S}}$ , and compare it with  $D_S$  in Sects. 7.2 and 7.3 to verify the two assumptions.

## 7.2 Quantitative analysis

Here we conduct 10-class digit classification to check if digit information is mostly encoded in  $D_{\bar{S}}$ . The idea is that if  $D_{\bar{S}}$  are more relevant to digit recognition, the values in  $D_{\bar{S}}$  serve as better features for the classification task. To do this, we split vectors sampled from learned posteriors into sub-vectors by  $D_{\bar{S}}$  and  $D_S$ . A set containing 13-dimensional vectors corresponding to  $D_{\bar{S}}$  and sampled from MNIST, SVHN or Text encoders are denoted as  $Z_{D_{\bar{S}}}^M$ ,  $Z_{D_{\bar{S}}}^S$ , or  $Z_{D_{\bar{S}}}^T$  respectively. Likewise, we can define sub-vector sets  $Z_{D_S}^M$ ,  $Z_{D_S}^S$ , and  $Z_{D_S}^T$ . Afterwards, we train a linear classifier and measure the accuracy for each sub-vector set. The results are shown in Table 12.

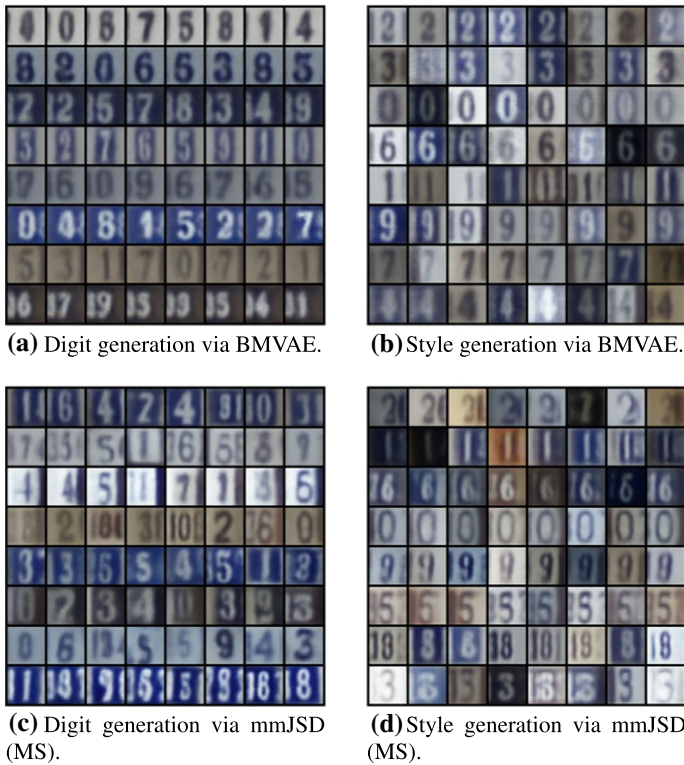
The observations we have are as follows:

- $Z_{D_{\bar{S}}} = \{Z_{D_{\bar{S}}}^M, Z_{D_{\bar{S}}}^S, Z_{D_{\bar{S}}}^T\}$  significantly outperforms  $Z_{D_S}$  on the classification task. Moreover,  $Z_{D_{\bar{S}}}$  generally reproduces the accuracies reported in Table 9. It could indicate that digit classes are mainly encoded in  $D_{\bar{S}}$ .
- We can see that the three encoders all learn to encode digit information in  $Z_{D_{\bar{S}}}$  instead of  $Z_{D_S}$ . It indicates that the three posteriors are successfully aligned with each other.

The second observation could be an expected result for  $\text{BMVAE}_p$  as the  $D_{KL}(q_{\phi_m} || \bar{q})$  objective encourages unifying posteriors. On the other hand, why does  $\text{BMVAE}_M$  can still align the posteriors without the additional objective may be unclear.

To see the reason for alignment, let us tentatively assume  $D_S$  indeed encodes SVHN-specific styles which are irrelevant to digit classes. It means that the SVHN decoder will learn to focus on  $D_S$  for image reconstruction, while other decoders tend to ignore  $D_S$ . If





**Fig. 7** Results of conditional generation. Both BMVAE and mmJSD (MS) learns to disentangle digits and styles, while BMVAE is able to decide appropriate dimensions for encoding information without supervision

the learned weights are positive in  $D_S$ , at some moments  $Z_{D_S}^M$  or  $Z_{D_S}^T$  will be selected to reconstruct SVHN images during training. Given that SVHN-specific styles are meaningless to MNIST and Text, the propagated gradients could be noise for the encoders. It may be similar to variational dropout which injects random noise to neurons to achieve sparsification. In our case, the MNIST and Text encoders learn to compress information to  $D_{\bar{S}}$ . It also explains why SVHN encoder learns digit classes well in  $D_{\bar{S}}$ . Since the digit classes are also meaningful to SVHN, the propagated gradients from MNIST or Text decoders would not be noise but beneficial information for learning.

In summary, the classification results confirm Assumption 2 that  $D_{\bar{S}}$  encodes information relevant to digit recognition. The results could also lead to the following takeaway message.

**Message 1** Despite no explicit objectives constraining alignment, BMVAE can learn to recognize and encode shared information in specific latent dimensions.

**Table 13** Classification accuracy of sampled latent vectors. Models with superscript  $u$  are trained by constant and uniform mixture weights. The dropped performance in the M,S and S,T columns could imply that models trained with uniform weights do not effectively preserve multi-modal information after merging

	M	S	T	M,S	M,T	S,T	M,S,T
BMVAE $_M^u$	0.96	0.80	0.99	0.88	0.98	0.95	0.94
BMVAE $_M$	0.96	0.80	0.99	0.94	0.98	0.98	0.96
BMVAE $_p^u$	0.96	0.79	0.99	0.93	0.99	0.97	0.98
BMVAE $_p$	0.97	0.82	0.99	0.96	0.99	0.98	0.98

**Table 14** Coherence evaluation for verifying the impact of learnable weights. The models with uniform weights have slight modality collapse issues such as BMVAE $_M^u$  on S M and BMVAE $_p^u$  on M S

	R	M S	M T	M S,T	S M	S T	S M,T	T M	T S	T M,S
BMVAE $_M^u$	0.47	0.76	0.99	0.69	0.55	0.80	0.81	0.94	0.78	0.87
BMVAE $_M$	0.66	0.76	0.99	0.97	0.88	0.99	0.92	0.95	0.77	0.93
BMVAE $_p^u$	0.73	0.69	0.99	0.94	0.86	0.94	0.92	0.92	0.73	0.90
BMVAE $_p$	0.72	0.80	0.99	0.97	0.90	0.94	0.92	0.97	0.80	0.96

### 7.3 Qualitative analysis

Here we analyze the learned weights by image generation. In Sect. 7.2, we show  $D_{\bar{S}}$  is responsible for the digit class. Here we attempt to show  $D_S$  is relevant to SVHN-specific styles.

The idea is that given a latent vector sampled from the SVHN posterior, we explicitly replace the values in  $D_{\bar{S}}$  with random values, then let the SVHN decoder generate an image based on the modified vector. Ideally, the generated image would depict a different digit while preserving the original style. Conversely, if we choose to replace values in  $D_S$  with random noise, the expected result would be depicting the same digit with a different SVHN style. The results are shown in Fig. 7a, b.

In Fig. 7a, each row is generated by replacing values in  $D_S$  with random noise. The noise is sampled from the prior. As can be observed, digits in each row vary but image styles are well-preserved. In Fig. 7b, we change to replace values in  $D_{\bar{S}}$ . As expected, a sampled digit class can now be depicted in diverse styles.

The results of generation confirm Assumption 1 that information encoded in  $D_S$  is relevant to style but not the digit labels. It implies that BMVAE can naturally disentangle modality-specific features without the need for explicitly designed algorithms such as mmJSD (MS). The takeaway messages are as follows.

**Message 2** BMVAE can learn to disentangle shared and modality-specific information without supervision.

**Message 3** The inferred mixture weights effectively indicates how BMVAE distributes learned features among latent dimensions.

## 7.4 BMVAE with uniform weights

We investigate whether the learnable mixture weights can be replaced by uniform weights when training BMVAE. To do this, we train  $\text{BMVAE}_M$  and  $\text{BMVAE}_p$  with constant weights  $\frac{1}{M}$ , denoted by  $\text{BMVAE}_M^u$  and  $\text{BMVAE}_p^u$ , on the MNIST-SVHN-Text dataset. We conduct the latent classification and coherence evaluation tasks to examine the trained models.

From Table 13, it can be observed that classification performance is degraded if models are trained with uniform weights. The degradation is more obvious when multiple modalities are available, implying that the mixture method is not effective enough to preserve modality-specific information. From Table 14, it can be observed that  $\text{BMVAE}_M^u$  has a significant drop in performance and may have the modality collapse issue. The PoE model  $\text{BMVAE}_p^u$  is also weaker than  $\text{BMVAE}_p$ , while the degradation is relatively small.

We summarize the observations and provide the takeaway messages as follows.

**Message 4** Adopting learnable and uneven weights improves model performance in general. The advantage is particularly significant on  $\text{BMVAE}_M$ .

**Message 5** Forcing mixture weights to be uniform has negative effects on learning cross-modality information. In practice, one can observe less informative multi-modal latent representations and degraded performance in modality transferring.

**Message 6** Learnable and uneven weights can help eliminate the modality collapse problem.

## 8 Conclusion

In this work, we propose dimension-wise Bayesian inference for multi-modal learning. We demonstrate the idea works on both MoE and PoE frameworks. The proposed MoE model,  $\text{BMVAE}_M$ , achieves state-of-the-art performance and is efficient to train. Therefore, computational cost is no longer an issue when we employ MoE methods. Regarding the proposed PoE model,  $\text{BMVAE}_p$ , we demonstrate that it achieves state-of-the-art performance as well and has theoretical connections to mmJSD and  $\text{BMVAE}_M$ .

In addition to data generation, we reveal that BMVAE does not need a specially-designed objective such as mmJSD (MS) to encode modal-specific information. Moreover, the private and shared features can be described by the inferred mixture weights. We also show that the inferred weights are beneficial to expert selection, improving data generation when multiple modalities are provided. Finally, we demonstrate that diverse mixture weights not only disentangle latent factors but also prevent degraded generation.

**Author contributions** Keng-Te Liao and Shou-De Lin contributed to the study conception and design. Data collection and analysis were performed by Keng-Te Liao, Bo-Wei Huang and Chi-Chun Yang. The first draft of the manuscript was written by Keng-Te Liao and Shou-De Lin. All the authors read and approved the final manuscript.

**Funding** This material is based upon work supported by Taiwan Ministry of Science and Technology (MOST) under Grant Number 110-2634-F-002-050-.

**Availability of data and material** The data for the experiments are publicly available.

**Code availability** The code can be obtained by contacting the first author.

## Declarations

**Conflict of interest** All the authors declare no conflicts of interest.

**Ethical approval** Not applicable.

**Consent to participate** Not applicable.

**Consent to publication** Not applicable.

## References

- Bagher Zadeh, A., Liang, P. P., Poria, S., Cambria, E., & Morency, L. P. (2018). Multimodal language analysis in the wild: CMU-MOSEI dataset and interpretable dynamic fusion graph. In *Proceedings of the 56th annual meeting of the association for computational linguistics (Volume 1: Long Papers)*, association for computational linguistics (pp. 2236–2246).
- Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5, 135–146.
- Burda, Y., Grosse, R. B., & Salakhutdinov, R. (2016). Importance weighted autoencoders. In *4th international conference on learning representations, ICLR 2016*, San Juan, Puerto Rico, May 2–4, 2016, conference track proceedings.
- Castro, S., Hazarika, D., Pérez-Rosas, V., Zimmermann, R., Mihalcea, R., & Poria, S. (2019). Towards multimodal sarcasm detection (an \_obviously\_ perfect paper). CoRR, [arXiv:1906.01815](https://arxiv.org/abs/1906.01815).
- Daunhawer, I., Sutter, T. M., Marcinkevičius, R., & Vogt, J. (2020). Self-supervised disentanglement of modality-specific and shared factors improves multimodal generative models. In *GCPR*.
- Figurnov, M., Mohamed, S., & Mnih, A. (2018). Implicit reparameterization gradients. *Advances in Neural Information Processing Systems*, 31, 441–452.
- Ghosh, P., Sajjadi, M. S. M., Vergari, A., Black, M. J., Schölkopf, B. (2020). From variational to deterministic autoencoders. In *8th international conference on learning representations, ICLR 2020*, Addis Ababa, Ethiopia, April 26–30, 2020.
- Hasan, M. K., Rahman, W., Bagher Zadeh, A., Zhong, J., Tanveer, M. I., Morency, L. P., & Hoque, M. E. (2019). UR-FUNNY: A multimodal language dataset for understanding humor. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China (pp. 2046–2056). <https://doi.org/10.18653/v1/D19-1211>, <https://www.aclweb.org/anthology/D19-1211>
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE conference on computer vision and pattern recognition, CVPR 2016*, Las Vegas, NV, USA, June 27–30, 2016 (pp. 770–778).
- Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8), 1771–1800.
- Hoeting, J. A., Madigan, D., Raftery, A. E., & Volinsky, C. T. (1999). Bayesian model averaging: A tutorial. *Statistical Science*, 14(4), 382–401.
- Hsu, W. N., & Glass, J. (2018). Disentangling by partitioning: A representation learning framework for multimodal sensory data.
- Jang, E., Gu, S., & Poole, B. (2017). Categorical reparameterization with gumbel-softmax. In *5th international conference on learning representations, ICLR 2017*, Toulon, France, April 24–26, 2017, Conference Track Proceedings.
- Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *3rd international conference on learning representations, ICLR 2015*, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings.
- Kingma, D. P., & Welling, M. (2014). Auto-encoding variational bayes. In *2nd international conference on learning representations, ICLR 2014*, Banff, AB, Canada, April 14–16, 2014, Conference Track Proceedings.
- Liang, P. P., Lyu, Y., Fan, X., Wu, Z., Cheng, Y., Wu, J., Chen, L., Wu, P., Lee, M. A., Zhu, Y., Salakhutdinov, R., & Morency, L. (2021). Multibench: Multiscale benchmarks for multimodal representation

- learning. In *Proceedings of the neural information processing systems track on datasets and benchmarks 1*, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual.
- Maddison, C. J., Mnih, A., & Teh, Y. W. (2017). The concrete distribution: A continuous relaxation of discrete random variables. In *5th international conference on learning representations, ICLR 2017*, Toulon, France, April 24–26, 2017, Conference Track Proceedings.
- Mohamed, S., Rosca, M., Figurnov, M., & Mnih, A. (2020). Monte Carlo gradient estimation in machine learning. *JMLR*, 21, 132:1–132:62.
- Shi, Y., Paige, B., & Torr, P. (2019). Variational mixture-of-experts autoencoders for multi-modal deep generative models. *Advances in Neural Information Processing Systems*, 32, 15718–15729.
- Sutter, T. M., Daunhawer, I., & Vogt, J. E. (2020). Multimodal generative learning utilizing jensen-shannon-divergence. In *Advances in neural information processing systems 33: Annual conference on neural information processing systems 2020*, NeurIPS 2020, December 6–12, 2020, virtual.
- Suzuki, M., Nakayama, K., & Matsuo, Y. (2017). Joint multimodal learning with deep generative models. In *5th international conference on learning representations, ICLR 2017*, Toulon, France, April 24–26, 2017, Workshop Track Proceedings.
- Tsai, Y. H., Liang, P. P., Zadeh, A., Morency, L., & Salakhutdinov, R. (2019). Learning factorized multimodal representations. In *ICLR*.
- Vedantam, R., Fischer, I., Huang, J., & Murphy, K. (2018). Generative models of visually grounded imagination. In *6th international conference on learning representations, ICLR 2018*, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings.
- Vielzeuf, V., Lechervy, A., Pateux, S., & Jurie, F. (2018). Centralnet: A multilayer approach for multimodal fusion. CoRR, [arxiv:1808.07275](https://arxiv.org/abs/1808.07275)
- Wu, M., & Goodman, N. D. (2018). Multimodal generative models for scalable weakly-supervised learning. In *Advances in neural information processing systems 31: Annual conference on neural information processing systems 2018*, NeurIPS 2018, December 3–8, 2018, Montréal, Canada (pp. 5580–5590).
- Zadeh, A., Zellers, R., Pincus, E., & Morency, L. (2016). MOSI: Multimodal corpus of sentiment intensity and subjectivity analysis in online opinion videos. CoRR, [arXiv:1606.06259](https://arxiv.org/abs/1606.06259)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.