Multimodal ELBO with Diffusion Decoders

Supplementary Material

6. Appendix

6.1. Multimodal ELBO with Diffusion Decoder

The marginal likelihood of the data $p(\mathbf{x})$ is given by:

$$\log p(\mathbf{x}) = \log \int p(\mathbf{x}, \mathbf{z})\, d\mathbf{z}, \tag{6}$$

where $\mathbf{z}$ is the shared latent variable. Using the definition of joint probability:

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z}), \tag{7}$$

the marginal likelihood becomes:

$$\log p(\mathbf{x}) = \log \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})\, d\mathbf{z}. \tag{8}$$

To make this integral tractable, we introduce a variational distribution $q(\mathbf{z}|\mathbf{x})$, which approximates the true posterior $p(\mathbf{z}|\mathbf{x})$. Using Jensen's inequality:

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\Big[\log p(\mathbf{x}|\mathbf{z})\Big] - \mathbb{D}_{\mathrm{KL}}\big(q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})\big). \tag{9}$$

This inequality defines the Evidence Lower Bound (ELBO):

$$\text{ELBO} = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\Big[\log p(\mathbf{x}|\mathbf{z})\Big] - \mathbb{D}_{\mathrm{KL}}\big(q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})\big). \tag{10}$$

In the multimodal setting, we assume that each modality $\mathbf{x}_i$ is conditionally independent given $\mathbf{z}$. Thus:

$$p(\mathbf{x}_M|\mathbf{z}) = \prod_{i \in M} p(\mathbf{x}_i|\mathbf{z}), \tag{11}$$

where $i \subseteq \{1, 2, \ldots, M\}$ represents the set of observed modalities. Substituting this into the ELBO:

$$\text{ELBO} = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\Big[\sum_{i \in M} \log p(\mathbf{x}_i|\mathbf{z})\Big] - \mathbb{D}_{\mathrm{KL}}\big(q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})\big). \tag{12}$$

Now, consider the mixture-based encoder $q(\mathbf{z}|\mathbf{x})$ defined as:

$$q(\mathbf{z}|\mathbf{x}) = \sum_{A \in \mathcal{S}} \omega_A q(\mathbf{z}|\mathbf{x}_A), \tag{13}$$

where $\mathcal{S}$ is the set of modalities, and ω_A is the weighting factor for each subset A and ≤ 1. The ELBO becomes:

$$\text{ELBO} = \mathbb{E}_{\sum_{A \in \mathcal{S}} \omega_A q(\mathbf{z}|\mathbf{x}_A)}\Big[\sum_{i \in M} \log p(\mathbf{x}_i|\mathbf{z})\Big] - \mathbb{E}_{\sum_{A \in \mathcal{S}} \omega_A q(\mathbf{z}|\mathbf{x}_A)}\Big[\log(q(\mathbf{z}|\mathbf{x})) - \log(p(\mathbf{z}))\Big] \tag{14}$$

Taking the sum outside the expectation and separating the second term:

$$\text{ELBO} = \sum_{A \in \mathcal{S}} \omega_A \mathbb{E}_{q(\mathbf{z}|\mathbf{x}_A)}\Big[\sum_{i \in M} \log p(\mathbf{x}_i|\mathbf{z})\Big] - \sum_{A \in \mathcal{S}} \omega_A \mathbb{E}_{q(\mathbf{z}|\mathbf{x}_A)} \log(q(\mathbf{z}|\mathbf{x})) + \sum_{A \in \mathcal{S}} \omega_A \mathbb{E}_{q(\mathbf{z}|\mathbf{x}_A)} \log(p(\mathbf{z})) \tag{15}$$

We can write the second term as:

$$\sum_{A\in\mathcal{S}} \omega_A \mathbb{E}_{q(\mathbf{z}|\mathbf{x}_A)} \log(q(\mathbf{z}|\mathbf{x})) = \sum_{A\in\mathcal{S}} \omega_A \mathbb{E}_{q(\mathbf{z}|\mathbf{x}_A)} \log(q(\mathbf{z}|\mathbf{x}_A)) - \sum_{A\in\mathcal{S}} \omega_A D_{\mathrm{KL}}\big(q_\phi(\mathbf{z}|\mathbf{x}_A)\big\|q(\mathbf{z}|\mathbf{x})\big) \tag{16}$$

Substituting back:

$$\begin{aligned}\text{ELBO} = &\sum_{A\in\mathcal{S}} \omega_A \mathbb{E}_{q(\mathbf{z}|\mathbf{x}_A)}\Big[\sum_{i\in M} \log p(\mathbf{x}_i|\mathbf{z})\Big] + \sum_{A\in\mathcal{S}} \omega_A D_{\mathrm{KL}}\big(q_\phi(\mathbf{z}|\mathbf{x}_A)\big\|q(\mathbf{z}|\mathbf{x})\big) - \sum_{A\in\mathcal{S}} \omega_A \mathbb{E}_{q(\mathbf{z}|\mathbf{x}_A)} \log(q(\mathbf{z}|\mathbf{x}_A)) \\ &+ \sum_{A\in\mathcal{S}} \omega_A \mathbb{E}_{q(\mathbf{z}|\mathbf{x}_A)} \log(p(\mathbf{z}))\end{aligned}$$

Because the KL term is always positive, we can remove it and create a new ELBO:

$$\begin{aligned}&= \sum_{A\in\mathcal{S}} \omega_A \mathbb{E}_{q(\mathbf{z}|\mathbf{x}_A)}\Big[\sum_{i\in M} \log p(\mathbf{x}_i|\mathbf{z})\Big] - \sum_{A\in\mathcal{S}} \omega_A \mathbb{E}_{q(\mathbf{z}|\mathbf{x}_A)} \log(q(\mathbf{z}|\mathbf{x}_A)) + \sum_{A\in\mathcal{S}} \omega_A \mathbb{E}_{q(\mathbf{z}|\mathbf{x}_A)} \log(p(\mathbf{z})) \\ &= \sum_{A\in\mathcal{S}} \omega_A \mathbb{E}_{q(\mathbf{z}|\mathbf{x}_A)}\Big[\sum_{i\in M} \log p(\mathbf{x}_i|\mathbf{z})\Big] - \sum_{A\in\mathcal{S}} \omega_A D_{\mathrm{KL}}\big(q_\phi(\mathbf{z}|\mathbf{x}_A)\big\|p(\mathbf{z})\big) \\ &= \sum_{A\in\mathcal{S}} \omega_A \left\{\mathbb{E}_{q(\mathbf{z}|\mathbf{x}_A)}\Big[\sum_{i\in M} \log p(\mathbf{x}_i|\mathbf{z})\Big] - D_{\mathrm{KL}}\big(q_\phi(\mathbf{z}|\mathbf{x}_A)\big\|p(\mathbf{z})\big)\right\}\end{aligned}$$

Which gives us the ELBO shown in equation 3:

$$\log p(x) \geq \sum_{A\in\mathcal{S}} \omega_A \left\{\mathbb{E}_{q(\mathbf{z}|\mathbf{x}_A)}\Big[\sum_{i\in M} \log p(\mathbf{x}_i|\mathbf{z})\Big] - D_{\mathrm{KL}}\big(q_\phi(\mathbf{z}|\mathbf{x}_A)\big\|p(\mathbf{z})\big)\right\}$$

6.1.1. Restating Equation 5

Now, adding the diffusion decoder part, the objective in Equation 5 is given by:

$$\begin{aligned}\sum_{A\in\mathcal{S}} \omega_A \Bigg\{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}_A)}\Bigg[\sum_{i=1}^{M} \mathbb{I}_{\text{ff}(i)} \log p_\theta(\mathbf{x}_i|\mathbf{z}) + \mathbb{I}_{\text{diff}(i)} \mathbb{E}_{\mathbf{x}_{it}} \frac{\lambda(t)}{2} \Big\|\nabla_{\mathbf{x}_{it}} \log p_t(\mathbf{x}_{it}|\mathbf{x}_i, \mathbf{z}) - s_\theta(\mathbf{x}_{it}, \mathbf{z}, t)\Big\|^2\Bigg] \\ -\mathbb{D}_{\mathrm{KL}}\big(q_\phi(\mathbf{z}|\mathbf{x}_A)\|p(\mathbf{z})\big)\Bigg\}.\end{aligned} \tag{17}$$

where $\mathbb{I}_{\text{diff}(i)}$ as the indicator function for using a diffusion decoder for modality i, and $\mathbb{I}_{\text{ff}(i)}$ as the indicator function for using a standard feed-forward decoder for modality i

Our aim is to show that this objective is a valid lower bound on the marginal likelihood of the data $p(\mathbf{x})$. Since we can only have one decoder for one modality, we only need to show that when using each type of decoder, we are optimizing a lower bound.

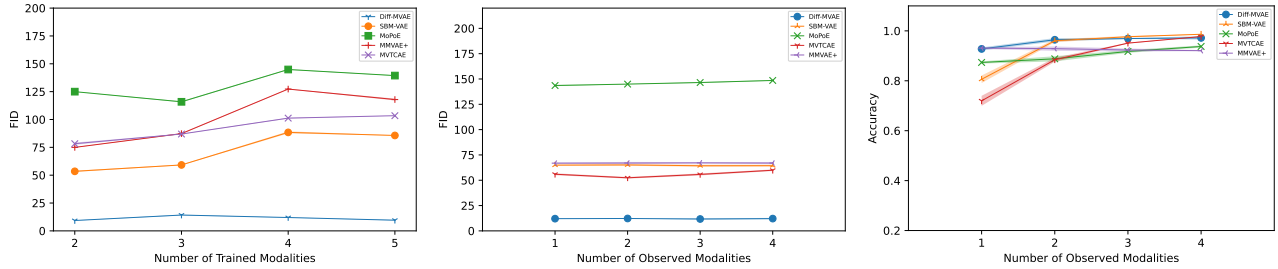
For a feed-forward decoder of a single modality, this directly corresponds to the Evidence Lower Bound (ELBO):

$$\text{ELBO}_{\text{ff}} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\Big[\log p_\theta(\mathbf{x}|\mathbf{z})\Big] - \mathbb{D}_{\mathrm{KL}}\big(q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})\big). \tag{18}$$

For a diffusion decoder of a single modality, we model $p_\theta(\mathbf{x}|\mathbf{z})$ using the score-based framework. Instead of directly optimizing $\log p_\theta(\mathbf{x}|\mathbf{z})$, we minimize the score-matching loss:

$$\mathbb{E}_{\mathbf{x}_{it}} \frac{\lambda(t)}{2} \Big\|\nabla_{\mathbf{x}_{it}} \log p_t(\mathbf{x}_{it}|\mathbf{x}, \mathbf{z}) - s_\theta(\mathbf{x}_{it}, \mathbf{z}, t)\Big\|^2. \tag{19}$$

To train a model to maximize a probability distribution $p(\mathbf{y})$, we can optimize a model with parameters θ and minimize the KL divergence between $p(\mathbf{y})$ and $p_\theta(\mathbf{y})$ which is $D_{\mathrm{KL}}(p(\mathbf{y}) \parallel p_\theta(\mathbf{y}))$. When we use the SDE described in this work, Song et al. [32] shows that: $D_{\mathrm{KL}}(p \parallel p_\theta^{\mathrm{SDE}}) \leq \mathcal{J}_{\mathrm{SM}}(\theta; g(\cdot)^2) + D_{\mathrm{KL}}(p_T \parallel \pi)$. By design of the forward SDE process, the second term goes to zero and is not dependent on the model parameters. $\mathcal{J}_{\mathrm{SM}}$ is the exact score-matching loss, which is



(a) Unconditional FID (b) Conditional FID 2 (c) Conditional Accuracy

Figure 7. Performance on 5 modalities of Ext-PolyMnist dataset

unknown but can be trained by approximating the score function using denoising score-matching loss shown in equation 2 which is also a lower bound as $D_{\text{KL}}(p \parallel p_\theta^{\text{SDE}}) \leq \mathcal{J}_{\text{DSM}}(\theta; g(\cdot)^2)$ [32]. This loss ensures that the model approximates the gradient of the log-likelihood $\nabla_{\mathbf{x}} \log p(\mathbf{x}|\mathbf{z}) \approx s_\theta(\mathbf{x}, \mathbf{z})$ and indirectly maximizes the likelihood $p(\mathbf{x}|\mathbf{z})$ as the loss is its lower bound. Thus, the score-matching loss serves as a surrogate for the likelihood term in the ELBO. This can be seen as another way of optimizing $p(\mathbf{x}|\mathbf{z})$ using the denoising score matching loss which will optimize the lower bound on the likelihood, $\mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \log(p(\mathbf{x}|\mathbf{z}) \geq \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \log(p_\theta^{\text{SDE}}(\mathbf{x}|\mathbf{z})$.

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\Big[\log p(\mathbf{x}|\mathbf{z})\Big] - \mathbb{D}_{\text{KL}}\big(q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})\big) \geq \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\Big[\log(p_\theta^{\text{SDE}}(\mathbf{x}|\mathbf{z})\Big] - \mathbb{D}_{\text{KL}}\big(q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})\big). \tag{20}$$

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\Big[\log(p_\theta^{\text{SDE}}(\mathbf{x}|\mathbf{z})\Big] - \mathbb{D}_{\text{KL}}\big(q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})\big). \tag{21}$$

So, the ELBO for using diffusion decoder will be:

$$\text{ELBO}_{\text{dd}} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\Big[\mathbb{E}_{\mathbf{x}_{it}} \frac{\lambda(t)}{2} \Big\|\nabla_{\mathbf{x}_{it}} \log p_t(\mathbf{x}_{it}|\mathbf{x}_i, \mathbf{z}) - s_\theta(\mathbf{x}_{it}, \mathbf{z}, t)\Big\|^2\Big] - \mathbb{D}_{\text{KL}}\big(q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})\big). \tag{22}$$

Combining the Objectives

The combined objective, which will give us Equation 5, incorporates both feed-forward and diffusion decoders. Only one of the decoders will be used for each modality, and both terms contribute to maximizing the marginal likelihood of the data $p(\mathbf{x})$, as the score-matching loss provides a valid surrogate for the likelihood term $\log p_\theta(\mathbf{x}|\mathbf{z})$ which makes the overall equation a lower bound on the probability of the data distribution.

$$\begin{aligned}\log p(x) \geq \sum_{A \in \mathcal{S}} \omega_A \Bigg\{ \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}_A)} \Bigg[\sum_{i=1}^{M_F} \log p_\theta(\mathbf{x}_i|\mathbf{z}) + \sum_{j=1}^{M_D} \mathbb{E}_{\mathbf{x}_{jt}} \frac{1}{2} \lambda(t) \Big\|\nabla_{\mathbf{x}_{jt}} \log p_t(\mathbf{x}_{jt} \mid \mathbf{x}_j, \mathbf{z}) - \\ s_\theta(\mathbf{x}_{jt}, \mathbf{z}, t)\Big\|_2^2 \Bigg] - D_{\text{KL}}\big(q_\phi(\mathbf{z}|\mathbf{x}_A)\big\|p(\mathbf{z})\big) \Bigg\}.\end{aligned} \tag{23}$$

6.2. PolyMnist Experiment

In this section, we add 5 modalities from the Extended PolyMnist dataset [42]. This dataset is mainly used in many of the baselines because it is easier to study the properties of the model under different number of modalities without being affected by computation power. We show the performance of the models in unconditional and conditional settings. First, figure 7a shows the unconditional performance of the models when the number of modalities the model is trained is increased from 2 to 5 modalities. Daunhawer et al. [5] showed that mixture-based multimodal VAEs experience performance degradation as the number of modalities the model is trained on increases. Diff-MVAE* shows almost a horizontal line, showing that the model doesn't lose performance even when the modalities are high. Figure 7b and 7c show models trained on 5 modalities and how the models perform conditionally. The number of observed modalities to generate the last modality is slowly increased and FID and the accuracy are calculated. As shown in the figures, Diff-MVAE* has superior FID and very good accuracy. Overall, our proposed model has better conditional coherence while having the best image generation quality.

6.3. CUB experiment setup

As discussed in the main paper, the CUB dataset consists of two modalities: image and text. The text VAE was initialized from the pre-trained weight of Li et al. [19] that uses a BERT encoder and GPT-2 decoder for all models. The images are resized to 64x64. We used a latent size of 768. The encoder and decoder architecture for the image modality are almost identical to that of Daniel and Tamar [4] except for the diffusion decoder, which uses a UNET architecture. The architecture with all the details can be referred in the attached code. The models were trained for 500 epochs using the Adam optimizer [13]. The baseline models are trained using different β values from 0.1, 0.5, 1, and 5, and the best model was selected by using the average of the conditional and unconditional FID. For MoPoE, β value of 5.0 is selected, while for the MVTCAE, β value of 1.0 is selected. We used a λ of $1e-5$ for Diff-MVAE variants.

The unconditional auxiliary score model also uses a 1D UNET architecture and accepts input similar to the size of the latent dimension (768). We use DDIM sampling for both the Diff-MVAE* and the auxiliary score-based model with 50 sampling steps. For Diff-MVAE, we use Euler-Maruyama sampling for 1000 steps. For training, we use continuous score-based diffusion for Diff-MVAE, and discrete timesteps for Diff-MVAE* similar to most DDPM training setup. For calculating FID and Clip-Score, 10000 samples were used from the test set. The main Diff-MVAE variants are trained on 2 A100 GPUs. The training time it takes is approximately about 150 hours. The auxiliary model is trained on 1 A100 GPU for approximately about 40 hours.

6.4. CelebAMask-HQ experiment setup

The CelebMaskHQ dataset is taken from Lee et al. [17] where the three modalities are images, masks, and attributes. All face part masks were combined into a single black-and-white image except the skin mask. Out of the 40 attributes, 18 were taken from it similar to the setup of [43]. The encoder and decoder architectures are similar to Daniel and Tamar [4] except MMVAE+ which uses an image encoder and decoder similar to the one presented in their work. A latent size of 256 was used for Diff-MVAE and MMVAE+, and a latent size of 1024 was used for MVTCAE and MoPoE. For MMVAE+, modality-specific and shared latent sizes are each 128 trained with IWAE estimator with K=1. We select the best β for the baselines from [0.1,0.5,1,2.5,5]. MoPoE use β of 1e-5 with the loss averaged over the dimensions of a data as that gave better result and to make comparison similar to Diff-MVAE. MVTCAE 0.5 and MMVAE+ 5. The diffusion decoder and the auxiliary score-based model use a UNET architecture. The architecture with all the details can be referred to in the attached code. The training and sampling of the Diff-MVAE variants are similar to the ones used in the CUB dataset 6.3. For calculating FID and F1-Score, the test set samples were used. The main Diff-MVAE model is trained on 4 A100 GPUs. The training time it takes is approximately about 120 hours. The auxiliary model is trained on 1 A100 GPU for approximately about 2 to 3 hours.

6.5. More Ablation

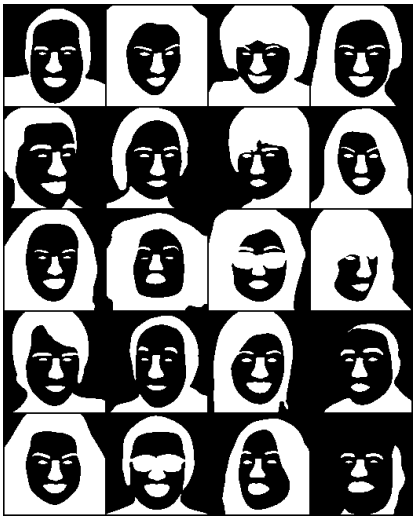
In this section, we study the effect of using the auxiliary score-based model on the baselines. Specifically, we select the MoPoE baseline and train a score-based diffusion prior on the product of experts of the posterior. After training the score-based prior, we generate unconditional z from the trained prior instead of a standard normal distribution to perform unconditional generation. We show the result in Table 4. The FID score is much better when using the unconditional auxiliary score-based prior and the approach can be applied to not only our models but also to other baselines.

Table 4. Auxiliary Score Model Prior for MoPoE

	Unc (Gaussion Prior)	Unc (Score Prior)
MoPoE	139.8	95.2

6.6. Additional Samples

(a) Diff-MVAE* unc image

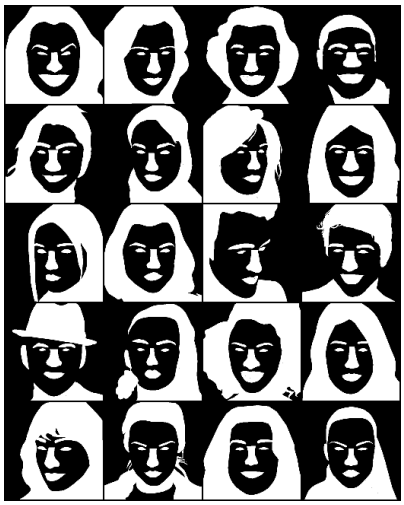


(b) Diff-MVAE* unc mask

(c) Diff-MVAE* unc attribute

Figure 8. Unconditional generation using Diff-MVAE

(a) Input attribute



(b) Input mask

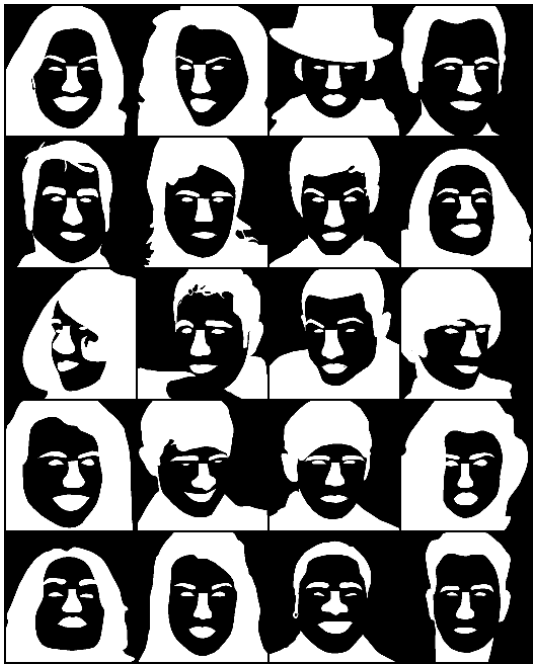
(c) Diff-MVAE* conditional image

Figure 9. Conditional generation using Diff-MVAE* given mask and attribute

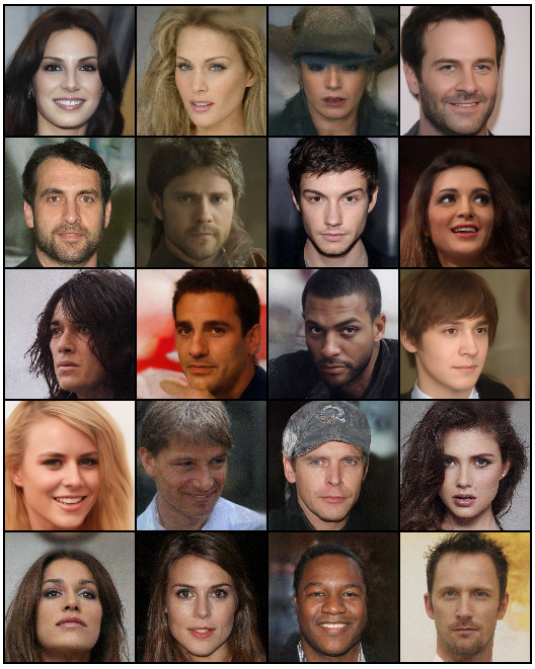
(a) Input attribute

(b) Diff-MVAE* given attribute

Figure 10. Conditional generation using Diff-MVAE* given attribute



(a) Input mask



(b) Diff-MVAE* given mask

Figure 11. Conditional generation using Diff-MVAE* given mask