A Background: Variational Autoencoder

VSAE is built upon the variational autoencoder (VAE) (Kingma and Welling, 2014; Rezende et al., 2014). Latent variable models attempt to model $p(\mathbf{x}, \mathbf{z})$ over observations of $\mathbf{x}$. However, the marginal likelihood $p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$ is computationally intractable. By introducing a parametric proposal distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$, a common strategy to alleviate the issue is to maximize an evidence lower bound (ELBO) of $p(\mathbf{x})$:

$$\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}) = \underbrace{\mathbb{E}_{\mathbf{z}\sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})]}_{\text{Conditional Log-Likelihood}} - \underbrace{D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))}_{\text{KL Regularizer}}$$

VAE realizes inference network (encoder) $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ and generative network (decoder) $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ with deep neural networks, and uses a standard Gaussian as the prior $p(\mathbf{z})$. Thus, $\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})$ is optimized over all training data w.r.t the parameters $\{\boldsymbol{\theta}, \boldsymbol{\phi}\}$ using backpropagation with reparameterization trick.

B Derivation: Expected ELBO on $\mathbf{x_u}$

The unobserved $\mathbf{x_u}$ makes the complete log-likelihood intractable and a common approach is to marginalize over $\mathbf{x_u}$ and directly model $\log p(\mathbf{x_o}, \mathbf{m})$. To model the dependence between $\mathbf{x_u}$ and $\mathbf{m}$, we do not ignore $\mathbf{x_u}$ using marginalization. Instead, we want to maximize $\log p(\mathbf{x_o}, \mathbf{m}|\boldsymbol{\theta}, \boldsymbol{\epsilon})$ through the lens of modeling complete log-likelihood $\log p(\mathbf{x_o}, \mathbf{x_u}, \mathbf{m}|\boldsymbol{\theta}, \boldsymbol{\epsilon})$, where $\boldsymbol{\theta}$ and $\boldsymbol{\epsilon}$ are parameters of data and mask generative models.

By introducing a distribution $q(\mathbf{x_u})$ defined over the unobserved attributes, for any choice of $q(\mathbf{x_u})$, we can decompose the observed log-likelihood:

$$\begin{aligned}&\log p(\mathbf{x_o}, \mathbf{m}|\boldsymbol{\theta}, \boldsymbol{\epsilon}) = \\ &\mathcal{L}''_{\boldsymbol{\theta},\boldsymbol{\epsilon}}(\mathbf{x_o}, \mathbf{x_u}, \mathbf{m}) + D_{\mathrm{KL}}(q(\mathbf{x_u})||p(\mathbf{x_u}|\mathbf{x_o}, \mathbf{m}, \boldsymbol{\theta}, \boldsymbol{\epsilon}))\end{aligned} \tag{11}$$

where the lower bound

$$\begin{aligned}\mathcal{L}''_{\boldsymbol{\theta},\boldsymbol{\epsilon}}(\mathbf{x_o}, \mathbf{x_u}, \mathbf{m}) = &\int q(\mathbf{x_u}) \log p(\mathbf{x_u}, \mathbf{x_o}, \mathbf{m}|\boldsymbol{\theta}, \boldsymbol{\epsilon}) d\mathbf{x_u} \\ &- \int q(\mathbf{x_u}) \log q(\mathbf{x_u}) d\mathbf{x_u}\end{aligned}$$

•E **step**: fix $\boldsymbol{\theta}, \boldsymbol{\epsilon}$ as $\boldsymbol{\theta}^*, \boldsymbol{\epsilon}^*$, so the lower bound

$$\begin{aligned}&\mathcal{L}''_{\boldsymbol{\theta}^*,\boldsymbol{\epsilon}^*}(\mathbf{x_o}, \mathbf{x_u}, \mathbf{m}) = \\ &\log p(\mathbf{x_o}, \mathbf{m}|\boldsymbol{\theta}^*, \boldsymbol{\epsilon}^*) - D_{\mathrm{KL}}(q(\mathbf{x_u})||p(\mathbf{x_u}|\mathbf{x_o}, \mathbf{m}, \boldsymbol{\theta}^*, \boldsymbol{\epsilon}^*))\end{aligned} \tag{12}$$

Suppose the ELBO is tight and we have the generative model $p(\mathbf{x_o}, \mathbf{x_u}, \mathbf{m}, \mathbf{z})$ that can model the dependency between $\mathbf{x_o}$, $\mathbf{x_u}$ and $\mathbf{m}$. As $D_{\mathrm{KL}} \geq 0$, the maximum is obtained if the proposal distribution $q(\mathbf{x_u}) = p(\mathbf{x_u}|\mathbf{x_o}, \mathbf{m}, \boldsymbol{\theta}^*, \boldsymbol{\epsilon}^*) = \int p(\mathbf{z}|\mathbf{x_o}, \mathbf{m}, \boldsymbol{\theta}^*, \boldsymbol{\epsilon}^*)p(\mathbf{x_u}|\mathbf{m}, \mathbf{z}, \boldsymbol{\theta}^*, \boldsymbol{\epsilon}^*)d\mathbf{z}$.

•M **step**: fix $q(\mathbf{x_u})$ as $q^*(\mathbf{x_u}) = p(\mathbf{x_u}|\mathbf{x_o}, \mathbf{m}, \boldsymbol{\theta}^*, \boldsymbol{\epsilon}^*)$, and maximize the lower bound w.r.t parameters:

$$\begin{aligned}\mathcal{L}''_{\boldsymbol{\theta},\boldsymbol{\epsilon}}(\mathbf{x_o}, \mathbf{x_u}, \mathbf{m}) = &\int q^*(\mathbf{x_u}) \log p(\mathbf{x_u}, \mathbf{x_o}, \mathbf{m}|\boldsymbol{\theta}, \boldsymbol{\epsilon}) d\mathbf{x_u} \\ &- \int q^*(\mathbf{x_u}) \log q^*(\mathbf{x_u}) d\mathbf{x_u}\end{aligned} \tag{13}$$

Let $\mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\theta},\boldsymbol{\epsilon}}(\mathbf{x_o}, \mathbf{x_u}, \mathbf{m})$ denote the ELBO derived in Eq. (4) that lower bounds the complete data log-likelihood, $\log p(\mathbf{x_u}, \mathbf{x_o}, \mathbf{m}|\boldsymbol{\theta}, \boldsymbol{\epsilon}) \geq \mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\theta},\boldsymbol{\epsilon}}(\mathbf{x_o}, \mathbf{x_u}, \mathbf{m})$ where $\boldsymbol{\phi}, \boldsymbol{\psi}$ are parameters of proposal distribution of variational inference. We have

$$\begin{aligned}&\int q^*(\mathbf{x_u}) \log p(\mathbf{x_u}, \mathbf{x_o}, \mathbf{m}|\boldsymbol{\theta}, \boldsymbol{\epsilon}) d\mathbf{x_u} \\ \geq &\int q^*(\mathbf{x_u}) \mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\theta},\boldsymbol{\epsilon}}(\mathbf{x_o}, \mathbf{x_u}, \mathbf{m}) d\mathbf{x_u} = \mathcal{L}'\end{aligned} \tag{14}$$

Therefore,

$$\log p(\mathbf{x_o}, \mathbf{m}|\boldsymbol{\theta}, \boldsymbol{\epsilon}) \geq \mathcal{L}'' \geq \mathcal{L}' - \text{const} \tag{15}$$

The objective is to maximize the expected lower bound $\mathcal{L}'$, which lower bounds the lower bound $\mathcal{L}'' + \text{const}$.

C Model Architecture

In all models, all the layers are modeled by multilayer perceptrons (MLPs). To fairly compare, we use author's public code and all other baselines are implemented with same backbone networks and at least as many parameters as our method. Basically, the attributive proposal networks take single attribute of data vector as input to infer the attributive proposal distribution; the collective proposal network takes the observed data vectors and mask vector (concatenation is used here) as input to infer the collective proposal distributions. The input vector to collective proposal network should have consistent dimension for the neural network, here we concatenate all attribute vectors and replace the unobserved attribute vectors with standard normal noise. Our implementation is based on PyTorch. All experiments were conducted on one Tesla P100 and one GeForce GTX 1080.

C.1 Encoders

Attributive proposal networks. In UCI repository experiment, the attributive encoders are modeled by

3-layer 16-dim MLPs for numerical data and 3-layer 64-dim MLPs for categorical data, all followed by Batch Normalization and Leaky ReLU nonlinear activations. In MNIST+MNIST experiment, the attributive encoders are modeled by 3-layer 128-dim MLPs followed by Leaky ReLU nonlinear activations. We set the latent dimension as 2-dim for every attributes in UCI repository experiments and 256-dim for every attribute in other experiments.

Collective proposal networks. In general, any inference network capable of domain fusion (Morency et al., 2011) can be naturally used here to map the observed data $\mathbf{x_o}$ and the mask $\mathbf{m}$ to the latent variables $\mathbf{z}$. One may also use techniques like in (Ma et al., 2019) to input a set of attributes. In this paper we simply use an MLP architecture, whose input is the complete data vector with unobserved attributes replaced with noise. So the collective encoder dimension is consistent for all data instances.

C.2 Decoders

We feed the aggregated latent variable to the mask decoder first. Then the output of mask decoder will be an extra condition for data decoders.

Mask Decoder.

UCI datasets: Linear(latent-dimension, 16)→ BatchNorm1d(16)→ LeakyReLU→ Linear(16, 16)→ LeakyReLU→ Linear(16, 16)→ LeakyReLU→ Linear(16, mask-dimension)→ Sigmoid;
multi-modal datasets: Linear(latent-dimension, 16)→ BatchNorm1d(16)→ LeakyReLU→ Linear(16,16)→ LeakyReLU→ Linear(16, 16)→ LeakyReLU→ Linear(16, mask-dimension)→ Sigmoid;

Data Decoder.

UCI data decoder: Linear(latent-dimension, 64)→ BatchNorm1d(64)→ LeakyReLU→ Linear(64)→ Linear(64, 64)→ Linear(64, data-dimension);
MNIST+MNIST data decoder: Linear(latent dim., 128)→ BatchNorm1d(128)→ LeakyReLU→ Linear(128,128)→ Linear(128, 128)→ Linear(128, data-dimension)

D Training and Baselines

Training. We use Adam optimizer for all models. For UCI numerical, mixed and CMU-MOSI experiment, learning rate is 1e-3 and use validation set to find a best model in 1000 epochs. For UCI categorical experiment, learning rate is 1e-2 and use validation set to find a best model in 2000 epochs. For MNIST+MNIST, FashionMNIST+label experiments, learning rate is 1e-4 and use validation set to find a best model in 1000 epochs. For evaluation, we evaluate numerical data by NRMSE, categorical data by PFC, images by MSE, labels by PFC, multimedia features by MSE. Initially we train the model for some (empirically choose 100) epochs without estimating the conditional log-likelihood of $\mathbf{x_u}$ to obtain a good approximated posterior. And then first feed the partially-observed data to the model and generate 100 samples of the unobserved attributes $\tilde{\mathbf{x}}_\mathbf{u}$; then feed the same batch for another pass and estimate the conditional log-likelihood with observed $\mathbf{x_o}, \mathbf{m}$ and generated $\mathbf{x_u}$.

Baselines. The baselines did not cover all types of experiments we consider. For example, HIVAE (Nazabal et al., 2020) reported experiments only on tabular data and MVAE (Wu and Goodman, 2018) reported experiments on multi-modal data. Therefore, we re-implement those models to fit our experiments if they do not have relevant experimental setups. All the re-implemented baselines use the same backbone architecture and the total number of parameters same as (or more than) our method. All the deep latent variable model baselines have same size of latent variables. In the setting of AE/VAE, the input is the whole data without the mask; In CVAE w/ mask, the encoder and decoder are both conditioned on the mask vector.

We also include additional baselines outside of deep latent variable models as in Table 6.

E Additional Experimental Results

E.1 UCI datasets

Please refer to Table 6 for results on more baselines MissForest (Stekhoven and Bühlmann, 2011), MICE (Azur et al., 2011), GAIN (Yoon et al., 2018) and MisGAN (Li et al., 2019) on the datasets reported in main paper. We include more results on mixed datasets for the DLVM baselines (refer to Table **??**) and non-MCAR missing mechanisms (refer to Table 5). In Table 5, we also include MCAR baselines under non-MCAR setting to get a sense of quantitative improvement.

E.2 multi-modal MNIST dataset

We pair digits as {(0,9),(1,8),(2,7),(3,6),(4,5)}. The training/test/validation sets respectively contain 23257/4832/5814 samples. For more quantitative results, please refer to Table 7. Fig. 5 illustrates more qualitative imputation performance, and Fig. 6 illustrates generation from parameter-free prior. Fig. 7

	Method	MAR	NMAR
Yeast	VAE	0.485 ± 0.004	$0.462 \pm \Delta$
	CVAE w/ mask	0.483 ± 0.001	$0.443 \pm \Delta$
	HI-VAE	0.479 ± 0.002	$0.434 \pm \Delta$
	MIWAE	$\mathbf{0.475 \pm 0.005}$	0.456 ± 0.036
	VSAE (ours)	$\mathbf{0.472 \pm 0.006}$	$\mathbf{0.425 \pm 0.007}$
Whitewine	VAE	$0.3845 \pm \Delta$	$0.3727 \pm \Delta$
	CVAE w/ mask	$0.3841 \pm \Delta$	$0.3726 \pm \Delta$
	HI-VAE	$0.3840 \pm \Delta$	$0.3724 \pm \Delta$
	MIWAE	$0.3834 \pm \Delta$	$0.3723 \pm \Delta$
	VSAE (ours)	$\mathbf{0.3825 \pm \Delta}$	$\mathbf{0.3717 \pm \Delta}$

Table 5: **Addtional Non-MCAR Data Imputation.** We show mean and standard deviation of **NRMSE** over 3 independent trials, lower is better. MCAR baselines are included to get a sense of quantitative improvement. Single imputation of 1000 importance samplings is used by all models. $\Delta < 0.0005$.

shows imputation results from multiple independent samplings given observed attribute.

Figure 5: **Imputation on MNIST+MNIST.** Top, middle and bottom rows visualize observed attribute, unobserved attribute, and the imputation of unobserved attribute from VSAE, respectively.

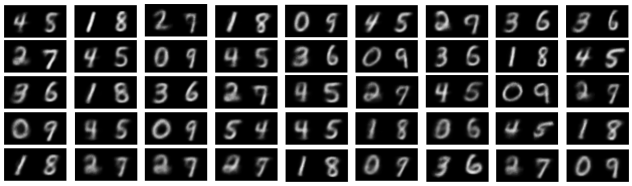

Figure 6: **Generation on MNIST+MNIST.** Generated Samples w/o conditional information. As shown, the correspondence between attributes (pre-defined pairs) are preserved while generation.

Another synthetic multi-modal dataset is to pair one digit in MNIST with a same digit in SVHN, which results in more heterogeneity. is The training/test/validation sets contain 44854/10000/11214 samples. We synthesize mask vectors over each modality by sampling from Bernoulli distribution and fixed after synthesis process. All original data points are only used once. Please refer to Table 8 for the attribute-wise imputation performance; Table 9 for the imputation performance under different missing ratios.

E.3 Image+label experiment

See Table 10 for more results.

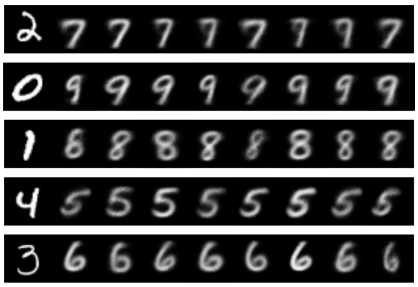

Figure 7: **Imputation from multiple independent sampling from selected latent space.** The leftmost digits are observed images in ground truth, and the right eight digits are imputations of corresponding unobserved digits.

E.4 Multi-modal experiment

See Table 11, we include additional experiments on multi-modal datasets to demonstrate the general effectiveness of our model. We choose the datasets following MVAE (Wu and Goodman, 2018) and MFM (Tsai et al., 2019). We choose CMU-MOSI and ICT-MMMO (Tsai et al., 2019) and use the publicly released features of each modality. All the numbers are calculated on the feature level. CMU-MOSI (Zadeh et al., 2019) is a collection of 2199 monologue opinion video clips annotated with sentiment. ICT-MMMO consists of 340 online social review videos annotated for sentiment. We train all the models using Adam optimizer with learning rate of 1e-3.

	Phishing	**Mushroom**	**Yeast**	**Whitewine**	**Heart (mixed)**	
Attribute type	categorical	categorical	numerical	numerical	categorical	numerical
MissForest	$0.478 \pm \Delta$	$0.419 \pm \Delta$	$0.424 \pm \Delta$	$0.3762 \pm \Delta$	$\mathbf{0.487 \pm 0.026}$	0.634 ± 0.005
MICE	$0.396 \pm \Delta$	$0.574 \pm \Delta$	$0.521 \pm \Delta$	$0.4280 \pm \Delta$	0.616 ± 0.026	0.749 ± 0.005
GAIN	0.301 ± 0.001	0.541 ± 0.006	0.583 ± 0.008	$0.3730 \pm \Delta$	0.708 ± 0.021	$0.661 \pm \Delta$
MisGAN	0.321 ± 0.005	0.533 ± 0.009	0.483 ± 0.007	$0.3725 \pm \Delta$	0.538 ± 0.017	0.643 ± 0.019
VSAE (ours)	$\mathbf{0.237 \pm \Delta}$	$\mathbf{0.416 \pm 0.009}$	$\mathbf{0.419 \pm 0.008}$	$\mathbf{0.3719 \pm \Delta}$	$\mathbf{0.482 \pm 0.014}$	$\mathbf{0.579 \pm 0.015}$

Table 6: **Addtional baselines of MCAR Data Imputation on UCI datasets.** We consider three types—categorical, numerical and mixed tabular datasets. Missing ratio is 0.5 on all datasets.We use the public code of GAIN/MisGAN, package missingpy for MissForest, and package fancyimpute for MICE. $\Delta < 0.0005$.

	0.3	0.5	0.7
AE	0.2124 ± 0.0012	0.2147 ± 0.0008	0.2180 ± 0.0008
VAE	0.1396 ± 0.0002	0.1416 ± 0.0001	0.1435 ± 0.0006
CVAE w/ mask	0.1393 ± 0.0002	0.1412 ± 0.0006	0.1425 ± 0.0012
MVAE	0.1547 ± 0.0012	0.1562 ± 0.0003	0.1579 ± 0.0006
HI-VAE	0.1464 ± 0.0024	0.1482 ± 0.0013	0.1497 ± 0.0016
VSAE	$\mathbf{0.1371 \pm 0.0001}$	$\mathbf{0.1376 \pm 0.0002}$	$\mathbf{0.1379 \pm 0.0001}$

Table 7: **Data Imputation on MNIST+MNIST under different missing ratios.** Missing ratio is 0.3, 0.5 and 0.7. Evaluated by sum error of two attributes. We show mean and standard deviation over 3 independent runs. Lower is better.

	MNIST-MSE/784	SVHN-MSE/3072	Sum error
AE	0.0867 ± 0.0001	0.1475 ± 0.0006	0.2342 ± 0.0007
VAE	0.0714 ± 0.0001	0.0559 ± 0.0002	0.1273 ± 0.0003
CVAE w/ mask	0.0692 ± 0.0001	0.0558 ± 0.0003	0.1251 ± 0.0005
MVAE	0.0707 ± 0.0003	0.0602 ± 0.0001	0.1309 ± 0.0005
HI-VAE	0.0733 ± 0.0013	0.0611 ± 0.0001	0.1344 ± 0.0015
VSAE	$\mathbf{0.0682 \pm 0.0001}$	$\mathbf{0.0516 \pm 0.0001}$	$\mathbf{0.1198 \pm 0.0001}$

Table 8: **Data Imputation on MNIST+SVHN.** Missing ratio is 0.5. Evaluated by MSE. We show mean and standard deviation over 3 independent runs. Lower is better.

	0.3	0.5	0.7
AE	0.1941 ± 0.0006	0.2342 ± 0.0007	0.2678 ± 0.0012
VAE	0.1264 ± 0.0001	0.1273 ± 0.0003	0.1322 ± 0.0005
CVAE w/ mask	0.1255 ± 0.0002	0.1251 ± 0.0005	0.1295 ± 0.0006
MVAE	$\mathbf{0.1227 \pm 0.0019}$	$\mathbf{0.1202 \pm 0.0006}$	0.1213 ± 0.0003
HI-VAE	0.1267 ± 0.0018	0.1279 ± 0.0003	0.1233 ± 0.0004
VSAE	$\mathbf{0.1217 \pm 0.0002}$	$\mathbf{0.1198 \pm 0.0001}$	$\mathbf{0.1202 \pm 0.0002}$

Table 9: **Data Imputation on MNIST+SVHN under different missing ratios.** Missing ratio is 0.3, 0.5 and 0.7. Evaluated by sum error of two modalities. We show mean and standard deviation over 3 independent runs. Lower is better.

	FashionMNIST		MNIST	
	image (MSE)	label (PFC)	image (MSE)	label (PFC)
AE	0.1104 ± 0.001	$0.366 \pm \Delta$	$0.0700 \pm \Delta$	$0.406 \pm \Delta$
VAE	$0.0885 \pm \Delta$	$0.411 \pm \Delta$	$0.0686 \pm \Delta$	0.406 ± 0.01
CVAE w/ mask	$0.0887 \pm \Delta$	$0.412 \pm \Delta$	$0.0686 \pm \Delta$	$0.419 \pm \Delta$
MVAE	0.1402 ± 0.002	0.374 ± 0.07	0.2276 ± 0.002	$0.448 \pm \Delta$
HI-VAE	$\mathbf{0.0875 \pm \Delta}$	$0.365 \pm \Delta$	0.0788 ± 0.004	$0.409 \pm \Delta$
VSAE	$\mathbf{0.0874 \pm \Delta}$	$\mathbf{0.356 \pm \Delta}$	$\mathbf{0.0681 \pm \Delta}$	$\mathbf{0.397 \pm 0.01}$

Table 10: **Data Imputation on Image+label datasets.**. Missing ratio is 0.5. Image and label attribute are evaluated by MSE and PFC respectively. We show mean and standard deviation over 3 independent runs (lower is better). $\Delta < 0.01$.

	Acoustic-MSE	Visual-MSE	Textual-MSE
AE	0.1211 ± 0.0013	$0.00502 \pm \Delta$	0.366 ± 0.001
VAE	0.0407 ± 0.0005	$0.00500 \pm \Delta$	0.293 ± 0.001
CVAE w/ mask	$\mathbf{0.0396 \pm 0.0042}$	$0.00492 \pm \Delta$	0.295 ± 0.001
MVAE	0.0836 ± 0.0357	$\mathbf{0.00485 \pm \Delta}$	0.405 ± 0.002
HI-VAE	0.0644 ± 0.0024	$0.00571 \pm \Delta$	0.385 ± 0.005
VSAE	$\mathbf{0.0381 \pm 0.0027}$	$\mathbf{0.00485 \pm \Delta}$	$\mathbf{0.243 \pm \Delta}$

Table 11: **Data Imputation on ICT-MMMO.** Missing ratio is 0.5. Evaluated by MSE of each attribute. We show mean and standard deviation over 3 independent runs (lower is better). $\Delta < 0.0001$.