Supplementary material for: Discovering Interpretable Representations for Both Deep Generative and Discriminative Models

8. Main steps of the ILVM Algorithm

Algorithm 1 Interpretable Lens Variable Model (ILVM)

Parameters: *generative:* $\boldsymbol{\theta}$ *and* ψ*, variational:* $\boldsymbol{\phi}$.
$\mathbb{J}(\mathbf{z})$ *: the lower bound.*
repeat
  $\mathbf{z} \leftarrow$ *a random minibatch.*
  $\mathbf{z_0} \leftarrow \mathbf{q_0}(\mathbf{z_0}|\mathbf{z}, \mathbf{s})$
  $\mathbf{z}^* \leftarrow \mathbf{f_T} \circ \mathbf{f_{T-1}} \circ \ldots \circ \mathbf{f_1}(\mathbf{z})$
  Compute $\mathbb{J}(\mathbf{z})$
  $\Delta\boldsymbol{\theta} \propto -\nabla_{\boldsymbol{\theta}}\mathbb{J}(\mathbf{z})$
  $\Delta\psi \propto -\nabla_{\psi}\mathbb{J}(\mathbf{z})$
  $\Delta\boldsymbol{\phi} \propto -\nabla_{\boldsymbol{\phi}}\mathbb{J}(\mathbf{z})$
until $\boldsymbol{\theta}, \psi, \boldsymbol{\phi}$ *do not change*

9. Related Work

Along with the interpretability algorithms cited in the main document, we briefly point out some of the relevant state-of-the-art algorithms. One of the most notable interpretability frameworks at the moment is an extension of generative adversarial networks (GANs) (Goodfellow et al., 2014) referred to as InfoGAN (Chen et al., 2016). In Chen et al. (2016), a recognition network covering a subset of the variables is established in a GAN such that the mutual information between the recognition network and a set of prespecified variables representing salient attributes is maximized. The claim that InfoGAN is unsupervised is not fully precise due to the required knowledge of such variables. Performance of InfoGAN is also affected by the instability of GANs. Another issue directly stemming from the reliance on a GAN framework is the lack of a comprehensive inference network like those guaranteed in VAE-based frameworks.

Another recently proposed framework is a modification to variational auto-encoders (VAEs), referred to as β-VAE (Higgins et al., 2017), which makes fewer assumptions about the data than InfoGAN. The main idea of β-VAE is to augment the standard VAE formulation with a hyperparameter that emphasizes on learning statistically independent latent factors. The β-VAE is an interesting step in a promising direction but it relies on rather strong assumptions like the assumption that the interpretable factors are always statistically independent, and the hypothesis that higher values of the hyperparameter should encourage learning a disentangled and therefore interpretable representation. Also, as mentioned in the paper, the impact of optimizing the parameter for interpretability on the reconstruction fidelity can be negative since having a more disentangled representation sometimes comes at the expense of blurrier reconstructions (Higgins et al., 2017).

There are several other recent algorithms in the literature including Kulkarni et al. (2015); Desjardins et al. (2012); Hsu et al. (2017); Siddharth et al. (2017); Mathieu et al. (2016); Vedantam et al. (2018); Donahue et al. (2018), but we believe the two aforementioned frameworks are the most related to ours. Other methods such as LIME (Ribeiro et al., 2016) have a similar motivation to ILVM in that they can be used to explain an existing model that has been optimized for performance. Others, such as generalized additive models (GAM) (Larsen, 2015), instead restrict the model learned in order to improve interpretability, which is a similar theme to JLVM.

10. Datasets

10.1. MNIST

MNIST (standing for Mixed National Institute of Standards and Technology) is a 28 $\times$ 28 pixel image dataset. The MNIST dataset (LeCun et al., 1998) is a handwritten digit database consisting of a training set of 60,000 instances and a test set of 10,000 instances.

10.2. SVHN

The Street View House Numbers (SVHN) (Netzer et al., 2011) dataset is a digit classification dataset consisting of 32 $\times$ 32 color (32 $\times$ 32 $\times$ 3) images where each instance consists of one, two or three digits. The SVHN dataset contains 73,257 training digits (instances) and 26,032 test digits. Each image is 64 $\times$ 64.

10.3. Chairs

The 3D Chairs dataset (Aubry et al., 2014) contains renderings of 1,393 chair models. Each model is rendered from 62 viewpoints: 31 azimuth angles (with a step of 11°) and 2 elevation angles (20° and 30°), with a fixed distance to the chair (Dosovitskiy et al., 2017).

11. Derivation of the JLVM Lower Bound

In order to be compressive of $\mathbf{x}$ and expressive about $\mathbf{s}$, the notion of information bottleneck among $\mathbf{x}$, $\mathbf{s}$ and $\mathbf{z}^*$ can be defined as follows:

$$\mathbf{IB}(\mathbf{z}^*, \mathbf{x}, \mathbf{s}) = \mathbf{I}(\mathbf{z}^*, \mathbf{s}) - \beta \mathbf{I}(\mathbf{z}^*, \mathbf{x}) \tag{1}$$

We therefore consider maximizing (1) to be a proxy for having an interpretable $\mathbf{z}^*$. First, assume that $\mathbf{x}$ is generated by $\mathbf{z}^*$ and that $\mathbf{z}^*$ is dependent on $\mathbf{s}$, i.e. $\mathbf{p}(\mathbf{x}, \mathbf{s}, \mathbf{z}^*) = \mathbf{p}(\mathbf{s})\mathbf{p}(\mathbf{z}^*|\mathbf{s})\mathbf{p}(\mathbf{x}|\mathbf{z}^*)$. Recall that $\mathbf{z}^*$ should as well fit the data and for that we use a VAE consisting of a recognition model and a generative model. Let's refer to the parameters of the generative and recognition models as $\boldsymbol{\omega}$ and $\boldsymbol{\alpha}$, respectively. Also, note that an interpretable model will have the probability distribution of the latent space $\mathbf{z}^*$ given the side information, $\mathbf{s}$, as a simple (e.g. linear) distribution. The objective in (1) is the first out of two objectives that this framework needs to satisfy. In addition to the interpretability objective in (1) and for the sake of data fitting, the latent space $\mathbf{z}^*$ is also constrained by the variational objective described as follows:

$$\begin{aligned}
\log \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{x}) &= \log \int_{\mathbf{z}^*} \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{x}, \mathbf{z}^*)\, d\mathbf{z}^* = \log \int_{\mathbf{z}^*} \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{x}, \mathbf{z}^*) \frac{\mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{z}^*|\mathbf{x})}{\mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{z}^*|\mathbf{x})}\, d\mathbf{z}^* \\
&\geq \mathbb{E}_{\mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{z}^*|\mathbf{x})}[\log \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{x}, \mathbf{z}^*) - \log \mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{z}^*|\mathbf{x})] \\
&= \mathbb{E}_{\mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{z}^*|\mathbf{x})}[\log \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{x}|\mathbf{z}^*)] - \mathbb{KL}(\mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{z}^*|\mathbf{x}) \| \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*))
\end{aligned} \tag{2}$$

Note that the objective in (2) is based on the marginal likelihood of an individual data point. The variational lower bound objective over $\mathbf{N}$ data points is therefore obtained by composing a sum over the marginal likelihoods of the individual data points:

$$\begin{aligned}
&\log\, \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{x}^{\mathbf{1}}, \cdots, \mathbf{x}^{\mathbf{N}}) = \frac{1}{\mathbf{N}} \sum_{\mathbf{i}=1}^{\mathbf{N}} \log \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{x}^{\mathbf{i}}) \geq \\
&\frac{1}{\mathbf{N}} \sum_{\mathbf{i}=1}^{\mathbf{N}} [\mathbb{E}_{\mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{z}^*|\mathbf{x}^{\mathbf{i}})}[\log \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{x}^{\mathbf{i}}|\mathbf{z}^*)] - \mathbb{KL}(\mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{z}^*|\mathbf{x}^{\mathbf{i}}) \| \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*))]
\end{aligned} \tag{3}$$

Maximizing (3) satisfies the data fitting objective. Now we additionally need to technically inspect how to optimize the objective in (1). Let's begin with analyzing the first term in (1), $\mathbf{I}(\mathbf{z}^*, \mathbf{s})$. Let $\mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{s}|\mathbf{z}^*)$ be the estimated approximation to the ground truth $\mathbf{p}(\mathbf{s}|\mathbf{z}^*)$. The KL-divergence between both is:

$$\begin{aligned}
&\mathbb{KL}[\mathbf{p}_{\boldsymbol{\omega}}(\mathbf{s}|\mathbf{z}^*) \| \mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{s}|\mathbf{z}^*)] \geq 0 \\
&\int \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{s}|\mathbf{z}^*) \log \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{s}|\mathbf{z}^*) d\mathbf{s} \geq \int \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{s}|\mathbf{z}^*) \log \mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{s}|\mathbf{z}^*) d\mathbf{s}
\end{aligned} \tag{4}$$

Therefore:

$$\begin{aligned}
\mathbf{I}(\mathbf{z}^*, \mathbf{s}) &= \int\int \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{s}, \mathbf{z}^*) \log \frac{\mathbf{p}_{\boldsymbol{\omega}}(\mathbf{s}, \mathbf{z}^*)}{\mathbf{p}(\mathbf{s})\mathbf{p}(\mathbf{z}^*)} \; d\mathbf{s} \; d\mathbf{z}^* \\
&= \int\int \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{s}, \mathbf{z}^*) \log \frac{\mathbf{p}(\mathbf{s}|\mathbf{z}^*)\mathbf{p}(\mathbf{z}^*)}{\mathbf{p}(\mathbf{s})\mathbf{p}(\mathbf{z}^*)} \; d\mathbf{s} \; d\mathbf{z}^* \\
&\geq \int\int \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{s}, \mathbf{z}^*) \log \frac{\mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{s}|\mathbf{z}^*)}{\mathbf{p}(\mathbf{s})} \; d\mathbf{s} \; d\mathbf{z}^* \\
&= \int\int \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{s}, \mathbf{z}^*) \log \mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{s}|\mathbf{z}^*) \; d\mathbf{s} \; d\mathbf{z}^* - \int \mathbf{p}(\mathbf{s}) \log \mathbf{p}(\mathbf{s}) \\
&= \int \mathbf{p}(\mathbf{s}) \int \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*|\mathbf{s}) \log \mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{s}|\mathbf{z}^*) \; d\mathbf{s} \; d\mathbf{z}^* + \mathbf{H}(\mathbf{s})
\end{aligned} \tag{5}$$

Note that the entropy term $\mathbf{H}(\mathbf{s})$ in (5) does not affect our optimization objective. Let's now move on to the second term in (1), $\mathbf{I}(\mathbf{z}^*, \mathbf{x})$:

$$\begin{aligned}
\mathbf{I}(\mathbf{z}^*, \mathbf{x}) &= \int\int \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{x}, \mathbf{z}^*) \log \frac{\mathbf{p}_{\boldsymbol{\omega}}(\mathbf{x}, \mathbf{z}^*)}{\mathbf{p}(\mathbf{x})\mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*)} \; d\mathbf{x} \; d\mathbf{z}^* \\
&= \int\int \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{x}, \mathbf{z}^*) \log \frac{\mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*|\mathbf{x})}{\mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*)} \, d\mathbf{x} \, d\mathbf{z}^* \\
&= \int\int [\mathbf{p}_{\boldsymbol{\omega}}(\mathbf{x}, \mathbf{z}^*) \log \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*|\mathbf{x}) - \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{x}, \mathbf{z}^*) \log \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*)] \, d\mathbf{x} \, d\mathbf{z}^*
\end{aligned} \tag{6}$$

Again, since:

$$\begin{aligned}
&\mathbb{KL}[\mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*) \| \mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{z}^*)] \geq 0 \\
&\int \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*) \log \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*) d\mathbf{z}^* \geq \int \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*) \log \mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{z}^*) d\mathbf{z}^*
\end{aligned} \tag{7}$$

Therefore:

$$\begin{aligned}
&\int\int \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{x}, \mathbf{z}^*) \log \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*) \, d\mathbf{x} \, d\mathbf{z}^* = \\
&\int \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*) \log \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*) \, d\mathbf{z}^* \geq \int \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*) \log \mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{z}^*) \, d\mathbf{z}^*
\end{aligned} \tag{8}$$

Using (8) back into (6):

$$\begin{aligned}
&\mathbf{I}(\mathbf{z}^*, \mathbf{x}) \leq \\
&\int\int [\mathbf{p}_{\boldsymbol{\omega}}(\mathbf{x}, \mathbf{z}^*) \log \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*|\mathbf{x}) - \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{x}, \mathbf{z}^*) \log \mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{z}^*)] \, d\mathbf{x} \, d\mathbf{z}^* = \\
&\int\int \mathbf{p}(\mathbf{x}) [\mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*|\mathbf{x}) \log \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*|\mathbf{x}) - \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*|\mathbf{x}) \log \mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{z}^*)] \, d\mathbf{x} \, d\mathbf{z}^* \\
&= \int \mathbf{p}(\mathbf{x}) \, \mathbb{KL}[\mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*|\mathbf{x}) \| \mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{z}^*)] \, d\mathbf{x}
\end{aligned} \tag{9}$$

From (5) and (9) into (1):

$$\begin{aligned}
\mathbf{IB}(\mathbf{z}^*, \mathbf{x}, \mathbf{s}) \geq &\int \mathbf{p}(\mathbf{s}) \int \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*|\mathbf{s}) \log \mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{s}|\mathbf{z}^*) \; d\mathbf{s} \; d\mathbf{z}^* \\
&- \beta \int \mathbf{p}(\mathbf{x}) \, \mathbb{KL}[\mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*|\mathbf{x}) \| \mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{z}^*)] \, d\mathbf{x}
\end{aligned} \tag{10}$$

We can now aggregate the expressions denoting both objectives by concatenating the lower bounds in (3) and (10):

$$
\begin{aligned}
& max_{\boldsymbol{\omega},\boldsymbol{\alpha}} \frac{1}{\mathbf{N}} \sum_{1}^{\mathbf{N}} [E_{\mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{z}^*|\mathbf{x})}[\log \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{x}|\mathbf{z}^*)] - \mathbb{KL}(\mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{z}^*|\mathbf{x}) \| \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*))] \\
& + \int \mathbf{p}(\mathbf{s}) \int \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*|\mathbf{s}) \log \mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{s}|\mathbf{z}^*) \; d\mathbf{s} \; d\mathbf{z}^* \\
& - \beta \int \mathbf{p}(\mathbf{x}) \, \mathbb{KL}[\mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*|\mathbf{x}) \| \mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{z}^*)] \, d\mathbf{x}
\end{aligned} \tag{11}
$$

The original objective of the VAE in the first line of (11) aims at a high reconstruction fidelity, i.e. at being able to fit and reproduce $\mathbf{x}$ with high fidelity. Meanwhile the second term in (1), i.e. the last line of (11), aims at providing a highly interpretable representation by compressing the non-interpretable factors in $\mathbf{x}$. Therefore, these two terms are expected to potentially be in discord. Aggregation of the different terms of the proposed `JLVM` model provides further clarification of that. Note that the integration over $\mathbf{x}$ in the last line of (11) will ultimately be approximated by samples from the observed data. Therefore, we can replace it with a summation over the $\mathbf{N}$ data points. The overall objective of the `JLVM` model can therefore be lower bounded by:

$$
\begin{aligned}
& max_{\boldsymbol{\omega},\boldsymbol{\alpha}} \frac{1}{\mathbf{N}} (\sum_{1}^{\mathbf{N}} E_{\mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{z}^*|\mathbf{x})}[\log \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{x}|\mathbf{z}^*)] - \sum_{1}^{\mathbf{N}} \mathbb{KL}(\mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{z}^*|\mathbf{x}) \| \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*))) \\
& + \int \mathbf{p}(\mathbf{s}) \int \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*|\mathbf{s}) \log \mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{s}|\mathbf{z}^*) \; d\mathbf{s} \; d\mathbf{z}^* \\
& - \frac{\beta}{\mathbf{N}} \sum_{1}^{\mathbf{N}} \mathbb{KL}[\mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*|\mathbf{x}) \| \mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{z}^*)]
\end{aligned} \tag{12}
$$

$$
\begin{aligned}
& \equiv max_{\boldsymbol{\omega},\boldsymbol{\alpha}} \frac{1}{\mathbf{N}} \sum_{1}^{\mathbf{N}} E_{\mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{z}^*|\mathbf{x})}[\log \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{x}|\mathbf{z}^*)] \\
& - \frac{1}{\mathbf{N}} \sum_{1}^{\mathbf{N}} [\mathbb{KL}(\mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{z}^*|\mathbf{x}) \| \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*)) + \beta \mathbb{KL}[\mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*|\mathbf{x}) \| \mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{z}^*)]] \\
& + \int \mathbf{p}(\mathbf{s}) \int \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*|\mathbf{s}) \log \mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{s}|\mathbf{z}^*) \; d\mathbf{s} \; d\mathbf{z}^*
\end{aligned} \tag{13}
$$

In order to compute the lower bound in (13), we resort to the reparameterization trick (Kingma & Welling, 2014) where the variable $\mathbf{z}^*$ is expressed as: $\mathbf{z}^* = \mathbf{f}_{\boldsymbol{\alpha}}(\mathbf{x}, \epsilon)$. The function $\mathbf{f}_{\boldsymbol{\alpha}}$ is a deterministic function of $\mathbf{x}$ and the Gaussian random variable ϵ. Refer to the lower bound as $\mathbb{L}$, and assume a Gaussian $\mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{z}^*|\mathbf{x})$ and a Gaussian $\log \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{x}|\mathbf{z}^*)$ where the parameters (mean and standard deviation) of each are obtained via a neural network. Also assume a Gaussian prior $\mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*)$. The first KL-divergence can be analytically computed. Also, assume that the density choices of $\mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{z}^*)$ and of $\mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*|\mathbf{x}) \propto \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{x}|\mathbf{z}^*)\mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*)$ also allow for an analytical computation of the second KL-divergence. Our assumptions make this term tractable to compute, and are commonly used in this type of optimization. More specifically -for the second KL-divergence in (13)-, we assume a Gaussian likelihood term $\mathbf{p}_{\boldsymbol{\omega}}(\mathbf{x}|\mathbf{z}^*)$, which, along with a Gaussian prior $\mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*)$, leads to a Gaussian $\mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*|\mathbf{x})$. Then with a Gaussian $\mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{z}^*)$ as well, this makes it possible to analytically compute the second KL term. These assumptions are common for VAEs and it is feasible to extend beyond them in future work provided that the corresponding approximations can be performed. The gradients can now be computed:

$$
\begin{aligned}
& \nabla_{\{\boldsymbol{\omega},\boldsymbol{\alpha}\}} \mathbb{L} = \frac{1}{\mathbf{N}} \sum_{1}^{\mathbf{N}} E_{\mathbb{N} \in (0,\mathbf{I})}[\nabla_{\{\boldsymbol{\omega},\boldsymbol{\alpha}\}} \log \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{x}|\mathbf{f}_{\boldsymbol{\alpha}}(\mathbf{x}, \epsilon))] - \\
& \frac{1}{\mathbf{N}} \nabla_{\{\boldsymbol{\omega},\boldsymbol{\alpha}\}} \sum_{1}^{\mathbf{N}} [\mathbb{KL}(\mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{z}^*|\mathbf{x}) \| \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*)) + \beta \mathbb{KL}[\mathbf{p}_{\boldsymbol{\omega}}(\mathbf{z}^*|\mathbf{x}) \| \mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{z}^*)]] \\
& + \nabla_{\{\boldsymbol{\omega},\boldsymbol{\alpha}\}} E_{\mathbf{p}(\mathbf{s})}[\int \mathbf{p}_{\boldsymbol{\omega}}(\mathbf{f}_{\boldsymbol{\alpha}}(\mathbf{x}, \epsilon)|\mathbf{s}) \log \mathbf{q}_{\boldsymbol{\alpha}}(\mathbf{s}|\mathbf{f}_{\boldsymbol{\alpha}}(\mathbf{x}, \epsilon)) \; d\mathbf{z}^*]
\end{aligned} \tag{14}
$$

We use Adam (Kingma & Ba, 2015) to compute the gradients.

Figure 1. Another experiment comparing ILVM and JLVM to InfoGAN and β-VAE applied to SVHN. Each row represents an experiment where the saturation level of an SVHN image is varied while the other latent dimensions are kept fixed. Results in this figure as well as in Figure 5 of the main document demonstrate that the ranges of lighting and saturation generated by JLVM are considerably higher and clearer than those generated by InfoGAN and β-VAE, and also slightly higher than those generated by ILVM. Better viewed in color.

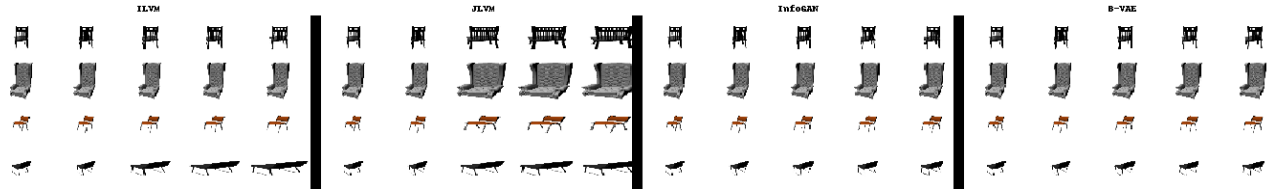

Figure 2. Another comparison between ILVM, JLVM, InfoGAN and β-VAE on 3D Chairs. Each row represents an experiment where the width level of an SVHN image is varied. This is equivalent to a trial to widen (or narrow) the chair. The JLVM, and to a lesser extent ILVM, models manage to efficiently represent a wider range of chair widths.

12. Other Details about the Experiments

We display two more figures (Figures 1 and 2) related to the experiments on SVHN and the 3D Chairs data in Section 6.1.1 of the main document.

In order to compute the metric, values of the side information $\mathbf{s}$ are categorized in groups and a linear classifier is used to express the dependence between $\mathbf{s}$ and $\mathbf{z}^*$. A linear SVM with a hard margin is used on top of which Platt scaling (Platt, 1999) is used to compute the probabilities. The JLVM parameter β is tuned using cross-validation.

Details of the model architectures are listed in Table 1. Adam (Kingma & Ba, 2015) is the optimizer used to compute the gradients.

Table 1. Architectures of the models in use.

Dataset	Architecture
MNIST	Encoder: Conv. 64 4 × 4 stride 2 ReLU, Conv. 128 4 × 4 stride 2 ReLU, FC 1024 ReLU, FC output Decoder: Deconv. reverse of the encoder. ReLU.
SVHN	Encoder: Conv. 64 4 × 4 stride 2 ReLU, Conv. 128 4 × 4 stride 2 ReLU, Conv. 256 4 × 4 stride 2 ReLU, FC output Decoder: Deconv. reverse of the encoder. ReLU.
3D Chairs	Encoder: Conv. 64 4 × 4 ReLU, Conv. 128 4 × 4 ReLU, Conv. 256 4 × 4 ReLU, Conv. 256 4 × 4 ReLU, Conv. 256 4 × 4 ReLU, FC 1024 ReLU, FC output Decoder: Deconv. reverse of the encoder. ReLU.

12.1. Test Log-Likelihood (Test LL)

We display the empirical test LL results of both ILVM and JLVM. Empirically, the loss in test LL resulting from the joint optimization of JLVM is not huge. As can be seen in Table 2, the loss resulting from JLVM is higher with SVHN and Chairs (datasets where JLVM has outperformed the other methods), and very small with MNIST.

References

Aubry, M., Maturana, D., Efros, A., Russell, B., and Sivic, J. Seeing 3D chairs: Exemplar part-based 2D-3D alignment using a large dataset of CAD models. pp. 3762–3769, 2014.

Table 2. Test log-likelihood (test LL) values resulting from `ILVM` and `JLVM` on the MNIST, SVHN and 3D Chairs datasets.

Algorithm / Dataset	MNIST	SVHN	3D Chairs
`ILVM`	-110.4	-189.5	-177
`JLVM`	-111.1	-194.8	-183.2

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems (NIPS)*, pp. 2172–2180, 2016.

Desjardins, G., Courville, A., and Bengio, Y. Disentangling factors of variation via generative entangling. *arXiv:1210.5474*, 2012.

Donahue, C., Balsubramani, A., McAuley, J., and Lipton, Z. Semantically decomposing the latent spaces of generative adversarial networks. *International Conference on Learning Representations (ICLR)*, 2018.

Dosovitskiy, A., Springenberg, J., Tatarchenko, M., and Brox, T. Learning to generate chairs, tables and cars with convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):692–705, 2017.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. *Advances in neural information processing systems (NIPS)*, pp. 2672–2680, 2014.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. Beta-VAE: Learning basic visual concepts with a constrained variational framework. *International Conference on Learning Representations (ICLR)*, 2017.

Hsu, W., Zhang, Y., and Glass, J. Unsupervised learning of disentangled and interpretable representations from sequential data. *Advances in neural information processing systems (NIPS)*, pp. 1876–1887, 2017.

Kingma, D. and Ba, J. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations (ICLR)*, 2015.

Kingma, D. and Welling, M. Auto-encoding variational Bayes. *International Conference on Learning Representations (ICLR)*, 2014.

Kulkarni, T., Whitney, W., Kohli, P., and Tenenbaum, J. Deep convolutional inverse graphics network. *Advances in neural information processing systems (NIPS)*, pp. 2539–2547, 2015.

Larsen, K. GAM: The predictive modeling silver bullet. *Multithreaded. Stitch Fix*, 30, 2015.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *In Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Mathieu, M., Zhao, J., Sprechmann, P., Ramesh, A., and LeCun, Y. Disentangling factors of variation in deep representations using adversarial training. *Advances in neural information processing systems (NIPS)*, 2016.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Reading digits in natural images with unsupervised feature learning. *In NIPS workshop on deep learning and unsupervised feature learning*, 2011.

Platt, J. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

Ribeiro, M., Singh, S., and Guestrin, C. Why should I trust you? Explaining the predictions of any classifier. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 22:1135–1144, 2016.

Siddharth, N., Paige, B., van den Meent, J., Demaison, A., Goodman, N., Kohli, P., Wood, F., and Torr, P. Learning disentangled representations with semi-supervised deep generative models. *Advances in neural information processing systems (NIPS)*, 2017.

Vedantam, R., Fischer, I., Huang, J., and Murphy, K. Generative models of visually grounded imagination. *International Conference on Learning Representations (ICLR)*, 2018.