Seminar Thesis

Learning Disentangled Representations with Semi-Supervised Deep Generative Models: A Review

Department of Statistics
Ludwig-Maximilians-Universität München

David B. Hoffmann

Munich, February 6th, 2026

Supervised by M. Sc. Simon Rittel

Abstract

The Semi-Supervised VAE (SSVAE) framework by Narayanaswamy et al. (2017) bridges the gap between supervised graphical models and unsupervised deep generative models. It allows the user to impose arbitrary dependence structures on the latent space of a Variational Auto-Encoder (VAE) through partial supervision. This work introduces the reader to the SSVAE method and critically assesses its limitations. We identify key works in the related literature which point out the expensive use of labels, problems with disentanglement, and the shift towards weak supervision. In our experiments, we address two limitations and identify a trade-off between disentanglement of the partially supervised latents and the remaining unsupervised variables. Furthermore, we find that the supervised disentanglement in SSVAEs only marginally degrades when introducing up to 20% noise to the labels on the MNIST dataset.

Contents

1 Introduction

The success of deep learning in discriminative tasks is largely attributed to its ability to automatically discover hierarchical feature representations from raw data. However, in the realm of generative modelling, simply approximating the data distribution is often insufficient. A central goal of modern representation learning is *disentanglement*, the ability to learn latent representations where distinct dimensions correspond to interpretable, real-world factors of variation (Bengio et al., 2013). While Variational Auto-Encoders (VAEs) (Kingma and Welling, 2014) provide a powerful probabilistic framework for learning latent variables, standard unsupervised approaches often yield entangled representations that are difficult to interpret or manipulate.

Recent theoretical work has suggested that learning disentangled representations from data alone, without any inductive bias or supervision, is fundamentally impossible (Locatello et al., 2019), motivating the use of supervised and structured approaches that leverage limited domain knowledge. A pivotal contribution in this direction is the Semi-Supervised VAE (SSVAE) framework proposed by Kingma et al. (2014), which unifies the flexibility of deep neural networks with the interpretability of probabilistic graphical models. Narayanaswamy et al. (2017) generalise the SSVAE framework to allow for the specification of arbitrary dependency structures, incorporating symbolic labels and unstructured style variables within a single stochastic computation graph.

In this work, we provide a comprehensive analysis of the generalised SSVAE framework shown in Figure 1. We aim to go beyond a mere summary by critically evaluating its theoretical foundations, its reception in the broader literature, and its empirical robustness. Specifically, we address a gap in the existing critiques by conducting experiments that quantify the model's sensitivity to two key practical constraints: the weight of the supervision signal and the presence of label noise.

The remainder of this thesis is organised as follows: Section 2 establishes the theoretical background of disentanglement in VAEs, semi-supervised learning, SSVAEs and importance sampling. Section 3 details the generalised framework by Narayanaswamy et al. (2017), explaining the importance-sampled objective and the stochastic computation graph. Section 4 provides a critical review of the literature, highlighting theoretical limitations such as the semantic conflation problem and discussing extensions into weak supervision. Section 5 presents our extension of the experiments of Narayanaswamy et al. (2017), specifically investigating the trade-off between supervision strength and disentanglement under label corruption. Finally, section 6 and section 7 synthesise our findings and discuss their implications for future research in structured representation learning.

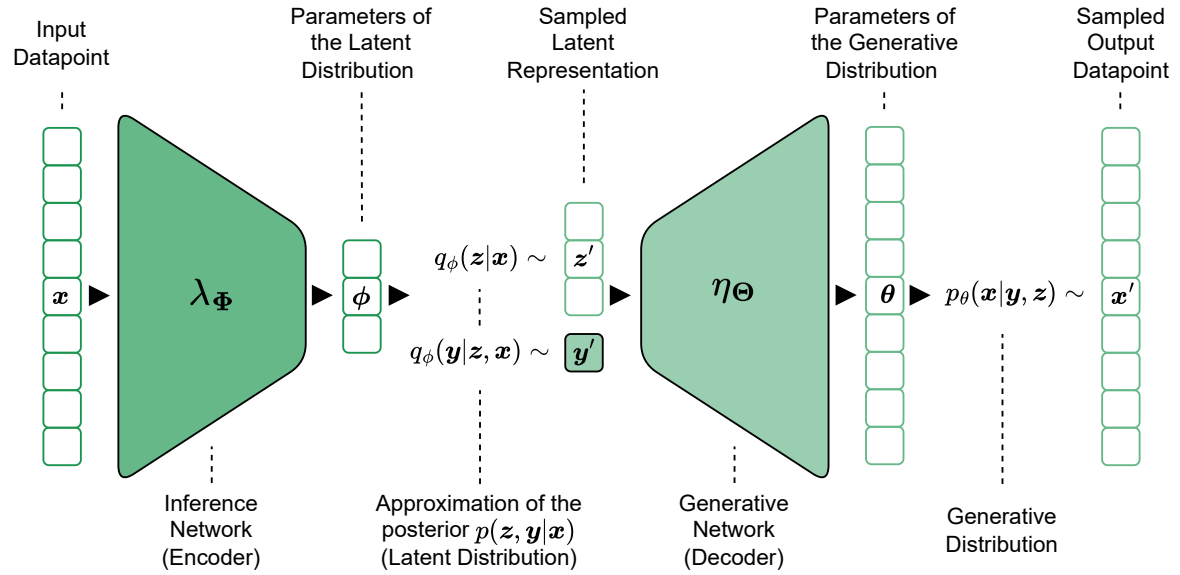


Figure 1: Formulation of the SSVAE framework. The graphical model partitions the latent space into explicitly supervised variables y (e.g., digit class in MNIST) and unsupervised variables z (e.g., style), governed by arbitrary dependency structures.

2 Background and Formulation

2.1 Disentanglement in Variational Auto-Encoders

VAEs are generative models that learn a deep latent variable model by maximizing a lower bound on the marginal likelihood of the data (Kingma and Welling, 2014). Given a dataset $\mathcal{D}_U = \{x^{(i)}\}_{i=1}^N$, we assume the data is generated by a random process involving an unobserved continuous random variable z. The generative process consists of a prior distribution $p_\theta(z)$ and a conditional likelihood $p_\theta(x|z)$, typically parametrised by a neural network (the decoder) with parameters θ.

Since the true posterior $p_\theta(z|x)$ is generally intractable, VAEs introduce an approximate posterior $q_\phi(z|x)$, parametrised by a separate neural network (the encoder) with parameters ϕ. The model is trained by maximizing the evidence lower bound (ELBO) on the marginal log-likelihood $\log p_\theta(x)$

$$\log p_\theta(x) \geq \mathcal{L}_{\text{ELBO}}(x;\theta,\phi) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)\|p(z)) \tag{1}$$

The first term is the reconstruction error, encouraging the decoder to recover the data from the latent code. The second term is the Kullback-Leibler (KL) divergence, which regularises the approximate posterior to be close to the prior, typically assumed to be a standard multivariate Gaussian $p(z) = \mathcal{N}(0, I)$.

To allow for backpropagation through the stochastic sampling process, VAEs utilise the reparametrisation trick. Instead of sampling z directly from $q_\phi(z|x) = \mathcal{N}(\mu, \sigma^2)$, we sample an auxiliary noise variable $\epsilon \sim \mathcal{N}(0, I)$ and compute

$$z = \mu_\phi(x) + \sigma_\phi(x) \odot \epsilon \tag{2}$$

where $\odot$ denotes the element-wise product.
A key goal in representation learning is disentanglement, where individual latent units are sensitive to changes in single generative factors of the data while being invariant to others (Bengio et al., 2013). Formally, a representation is disentangled if it factorises into independent subspaces corresponding to underlying factors of variation.
Standard VAEs often fail to learn disentangled representations due to the lack of explicit constraints on the latent structure. To address this, the β-VAE framework Higgins et al. (2017) introduces a hyperparameter β to weight the KL divergence term

$$\mathcal{L}_{\beta\text{-VAE}} = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \beta D_{KL}(q_\phi(z|x)\|p(z)) \tag{3}$$

Setting $\beta > 1$ imposes a stronger constraint on the latent bottleneck, encouraging the learned distribution $q_\phi(z|x)$ to align with the isotropic unit Gaussian prior. Since the prior has independent components, this pressure encourages the latent dimensions to become statistically independent, thereby promoting disentanglement, though often at the cost of reconstruction quality. Building on this, β-TCVAE explicitly penalises the total correlation of the aggregated posterior to encourage factorised latents and stronger disentanglement (Chen et al., 2018).

2.2 Degrees of Supervision

While standard supervised learning relies on a fully labelled dataset $\mathcal{D}_S = \{(x^{(i)}, y^{(i)})\}_{i=1}^M$, obtaining ground truth labels y is often expensive. Semi-supervised learning addresses this by leveraging a small set of labelled data $\mathcal{D}_S$ of size M alongside a much larger set of unlabelled data $\mathcal{D}_U = \{x^{(j)}\}_{j=1}^N$. The objective is to utilise the structural information inherent in $p(x)$ from $\mathcal{D}_U$ to improve the estimation of the conditional distribution $p(y|x)$, typically under the assumption that the decision boundary lies in low-density regions of the input space.
In contrast, weak supervision relaxes the requirement for exact ground truth labels, utilizing instead a dataset $\mathcal{D}_W = \{(x^{(i)}, \tilde{y}^{(i)})\}_{i=1}^L$ where $\tilde{y}$ represents a noisy, limited, or heuristic approximation of the true label y. A prominent subset is pair learning, where supervision is provided as pairwise constraints $\mathcal{D}_P = \{(x^{(i)}, x^{(j)}, r^{(ij)})\}$. Here, $r^{(ij)}$ denotes a binary relationship (e.g., must-link or cannot-link) between samples, guiding the model to learn a representation that respects these relative similarities rather than absolute class assignments.

2.3 Semi-Supervised Variational Auto-Encoders

To leverage the generalization benefits of generative modelling for classification tasks, Kingma et al. (2014) extend the VAE framework to the semi-supervised setting. They propose three approaches: the *M1* model, which separates feature learning from downstream classification; the *M2* model, which integrates the class label directly into the latent generative process; and a stacked *M1+M2* architecture. We focus here on the *M2*

model, where the data x is generated by both a latent class variable y and a continuous latent variable z.
The generative process factorises as

$$p_\theta(x, y, z) = p_\theta(x|y, z)p(y)p(z) \tag{4}$$

where $p(y) = \text{Cat}(y|\pi)$ and $p(z) = \mathcal{N}(z|0, I)$. In this prior specification, y and z are marginally independent. The likelihood $p_\theta(x|y, z)$ is parametrised by a deep neural network that takes both y and z as inputs to generate the observation x.
Crucially, the inference structure introduced to approximate the intractable posterior $p(z, y|x)$ utilises a specific factorization that introduces a fixed dependency between the latent variables. The recognition model is specified as

$$q_\phi(z, y|x) = q_\phi(z|x, y)q_\phi(y|x) \tag{5}$$

Here, $q_\phi(y|x)$ is modelled as a categorical distribution, which allows it to be used as a classifier, while $q_\phi(z|x, y)$ is a Gaussian distribution where the parameters (mean and variance) are functions of both the data x and the label y. This dependency structure in the inference network allows the model to learn a class-conditional latent distribution for z, enabling it to effectively separate style z from class y.
In the inference process, however, these variables are coupled. The approximate posterior for the continuous variable is defined as $q_\phi(z|x, y)$. By conditioning on y, the network encourages z to capture residual variations orthogonal to class identity.
For the supervised dataset $\mathcal{D}_S = \{(x^m, y^m)\}$, both x and y are observed. The supervised objective $\mathcal{L}_S(\theta, \phi; x, y)$ consists of two main components: a generative term, which is the ELBO on the joint distribution of x, y, and z, as well as a discriminative term weighted by α. By expanding the generative term using the factorization $p_\theta(x, y, z) = p_\theta(x|y, z)p(y)p(z)$, we obtain a reconstruction term which encourages the decoder output to closely match the encoder input, a KL regularization that encourages the style latent z to follow the prior, and the label prior $\log p(y)$

$$\mathcal{L}_S(\theta, \phi; x, y) = \underbrace{\mathbb{E}_{q_\phi(z|x,y)}\left[\log \frac{p_\theta(x, y, z)}{q_\phi(z|x, y)}\right]}_{\text{Generative (ELBO on joint } x,y)} + \underbrace{\alpha \log q_\phi(y|x)}_{\text{Discriminative}} \tag{6}$$

$$= \mathbb{E}_{q_\phi(z|x,y)}\Big[\log p_\theta(x|y, z) + \log p(y) + \log p(z) - \log q_\phi(z|x, y)\Big] + \alpha \log q_\phi(y|x) \tag{7}$$

$$= \underbrace{\mathbb{E}_{q_\phi(z|x,y)}[\log p_\theta(x|y, z)]}_{\text{Reconstruction}} - \underbrace{D_{KL}(q_\phi(z|x, y)\|p(z))}_{\text{Style Regularisation}} + \underbrace{\log p(y)}_{\text{Label Prior}} + \underbrace{\alpha \log q_\phi(y|x)}_{\text{Discriminative}} \tag{8}$$

For the unsupervised dataset $\mathcal{D}_U = \{x^n\}$, the label y is treated as a latent variable. We

marginalise over all classes to derive the unsupervised bound $\mathcal{L}_U(\theta, \phi; x)$

$$\mathcal{L}_U(\theta, \phi; x) = \sum_y q_\phi(y|x) \left(\mathbb{E}_{q_\phi(z|x,y)} \left[\log \frac{p_\theta(x, y, z)}{q_\phi(z|x, y)} \right] \right) + \mathcal{H}(q_\phi(y|x)) \tag{9}$$

where $q_\phi(y|x)$ is the recognition network acting as a discriminative classifier.
The final training objective combines these bounds over both datasets

$$\mathcal{L}(\theta, \phi; \mathcal{D}_U, \mathcal{D}_S) = \sum_{x \in \mathcal{D}_U} \mathcal{L}_U(\theta, \phi; x) + \gamma \sum_{(x,y) \in \mathcal{D}_S} \mathcal{L}_S(\theta, \phi; x, y) \tag{10}$$

where γ is used to balance the contribution of the supervised data. Note that this additional weighting parameter was not part of the original formulation from Kingma et al. (2014) which solely relied on α.

2.4 Importance Sampling

Importance sampling is a Monte Carlo method used to estimate properties of a target distribution $p(x)$ that is difficult to sample from directly, by utilizing a simpler proposal distribution $q(x)$.
Given a target density $p(x)$ and a function $f(x)$, we wish to estimate the expectation $\mathbb{E}_p[f(x)]$. We introduce a proposal distribution $q(x)$ such that $q(x) > 0$ whenever $p(x)f(x) \neq 0$ (i.e., the support of q covers the support of pf). By multiplying and dividing by $q(x)$, we rewrite the expectation as

$$\mathbb{E}_{x \sim p}[f(x)] = \int f(x)p(x)\, dx = \int f(x)\frac{p(x)}{q(x)}q(x)\, dx = \mathbb{E}_{x \sim q}\left[f(x)\frac{p(x)}{q(x)} \right] \tag{11}$$

The term $w(x) = \frac{p(x)}{q(x)}$ is known as the importance weight. In practice, we draw K samples $\{x^{(k)}\}_{k=1}^K$ from the proposal distribution $q(x)$ and approximate the expectation using the sample mean

$$\mathbb{E}_p[f(x)] \approx \frac{1}{K} \sum_{k=1}^{K} f(x^{(k)})w(x^{(k)}) \tag{12}$$

This standard estimator is unbiased, provided $p(x)$ is a normalised density. An example of importance sampling in the context of VAEs is the Importance Weighted Autoencoder (IWAE) (Burda et al., 2016) which employs importance sampling to derive a tighter lower bound on the marginal log-likelihood by averaging K latent samples drawn from the approximate posterior. Each sample is weighted by the importance ratio $w_i = p(x, z_i)/q(z_i|x)$, which reduces the bias of the variational approximation and improves the model's representative capacity.
However, in many applications, the target distribution is only known up to an intractable normalizing constant Z, i.e., $p(x) = \frac{\tilde{p}(x)}{Z}$. In this scenario, self-normalised importance sampling is used. We calculate unnormalised weights $\tilde{w}(x) = \frac{\tilde{p}(x)}{q(x)}$ and approximate the expectation using the ratio of estimators

$$\mathbb{E}_p[f(x)] \approx \frac{\sum_{k=1}^K \tilde{w}(x^{(k)}) f(x^{(k)})}{\sum_{k=1}^K \tilde{w}(x^{(k)})} \tag{13}$$

Unlike the standard estimator, the self-normalised estimator is biased for finite K (with bias of order $1/K$), but consistent, converging to the true expectation as $K \to \infty$.

3 Semi-Supervised Disentanglement with Arbitrary Dependencies

While the *M2* model by Kingma et al. (2014) provides a powerful framework for semi-supervised learning, it imposes a specific graphical structure where the latent variables y (label) and z (style) are assumed to be independent in the prior, and the recognition model is required to factorise as $q_\phi(y, z|x) = q_\phi(y|x)q_\phi(z|x, y)$. Narayanaswamy et al. (2017) generalise this approach to a broader class of partially-specified graphical models as visualised in Figure 2.

In this framework, the modeller specifies a graphical model where a subset of variables y are semantically meaningful (and partially observed), while the remaining variables z are left unstructured to capture nuisance variations. Unlike the *M2* model, this framework allows for arbitrary dependency structures in both the generative model $p_\theta(x, y, z)$ and the approximate posterior $q_\phi(y, z|x)$. This flexibility enables the modelling of complex relationships, such as hierarchical dependencies or causal structures between labels and style variables, which are often necessary for disentanglement in real-world datasets (Yang et al., 2019, Biswal et al., 2021).

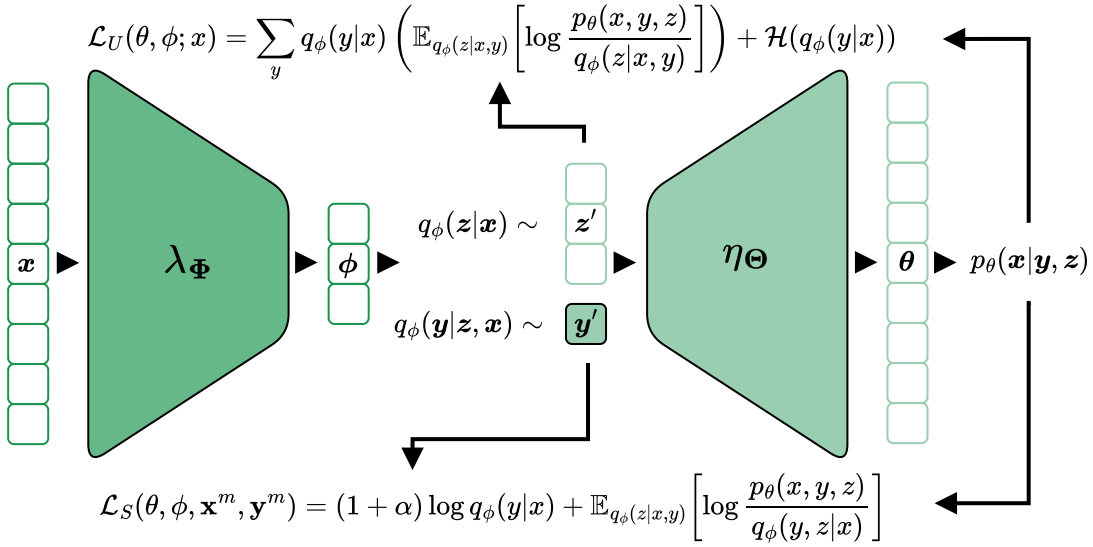


Figure 2: SSVAE framework with reformulated supervised (Equation 16) and unsupervised (Equation 9) loss functions.

3.1 The Generalised Variational Objective

The core challenge in training these generalised models lies in the supervised objective as defined in Equation 6. For the unsupervised term $\mathcal{L}_U$ (Equation 9), standard variational inference applies. However, for the supervised term $\mathcal{L}_S$, a difficulty arises when the recognition model $q_\phi(y, z|x)$ does not factorise conveniently. Specifically, evaluating the supervised lower bound requires sampling from the conditional posterior $q_\phi(z|x, y)$. For arbitrary graphical structures, this conditional distribution may not be available in closed form or easy to sample from directly.

To address this, Narayanaswamy et al. (2017) proposes an importance sampling estimator that alleviates the need to sample from $q_\phi(z|x,y)$ directly. Instead, samples are drawn from an unconditioned proposal distribution, the marginal encoder $q_\phi(z|x)$, and reweighted. They start by rewriting the supervised objective using the fact that $q_\phi(z|x,y)$ factorises to $\frac{q_\phi(y,z|x)}{q_\phi(y|x)}$ as follows

$$\mathcal{L}_S(\theta,\phi;x,y) = \alpha \log q_\phi(y|x) + \mathbb{E}_{q_\phi(z|x,y)}\left[\log \frac{p_\theta(x,y,z)}{q_\phi(z|x,y)}\right] \tag{14}$$

$$= \alpha \log q_\phi(y|x) + \mathbb{E}_{q_\phi(z|x,y)}\left[\log \frac{p_\theta(x,y,z)}{q_\phi(y,z|x)} + \log q_\phi(y|x)\right] \tag{15}$$

$$= (1+\alpha) \log q_\phi(y|x) + \mathbb{E}_{q_\phi(z|x,y)}\left[\log \frac{p_\theta(x,y,z)}{q_\phi(y,z|x)}\right] \tag{16}$$

This formulation removes the need to evaluate $q_\phi(z|x,y)$ directly. To approximate the expectation, they employ self-normalised importance sampling as introduced in subsection 2.4. Instead of sampling from the conditional $q_\phi(z|x,y)$, they sample proposals z^s from the unconditioned encoder distribution $q_\phi(z|x)$. The unnormalised importance weights w^s are defined as the ratio of the target joint distribution to the proposal

$$w^s = \frac{q_\phi(y,z^s|x)}{q_\phi(z^s|x)} \tag{17}$$

where $z^s \sim q_\phi(z|x)$. Using these weights, the expectation is approximated as

$$\mathbb{E}_{q_\phi(z|x,y)}\left[\log \frac{p_\theta(x,y,z)}{q_\phi(y,z|x)}\right] \approx \frac{1}{S}\sum_{s=1}^{S} \frac{w^s}{Z} \log \frac{p_\theta(x,y,z^s)}{q_\phi(y,z^s|x)} \tag{18}$$

where $Z = \frac{1}{S}\sum_{s=1}^{S} w^s$ is the normalisation constant.
Furthermore, the term $\log q_\phi(y|x)$ is itself approximated using a Monte Carlo estimator of the lower bound normally used in maximum likelihood estimation

$$\log q_\phi(y|x) \geq \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{q_\phi(y,z|x)}{q_\phi(z|x)}\right] \approx \frac{1}{S}\sum_{s=1}^{S} \log w^s \tag{19}$$

By combining the importance sampling estimator for the expectation (Equation 18) and the lower bound estimator for the log-likelihood (Equation 19), they obtain the final estimator for the supervised objective

$$\hat{\mathcal{L}}_S(\theta,\phi;x,y) := \frac{1}{S}\sum_{s=1}^{S} \frac{w^s}{Z} \log \frac{p_\theta(x,y,z^s)}{q_\phi(y,z^s|x)} + (1+\alpha)\log w^s \tag{20}$$

This formulation ensures that the discriminative power of the model is maximised using the same samples z^s and weights w^s for both the generative and discriminative components.

3.2 Relation to Kingma's M2 Model

The framework by Narayanaswamy et al. (2017) generalises the *M2* model. We can recover the exact *M2* objective by restricting the dependency structure. If we enforce the factorisation

$$q_\phi(y, z|x) = q_\phi(y|x)q_\phi(z|x, y) \tag{21}$$

then the importance weights simplify to constants with respect to z

$$w^s = \frac{q_\phi(y|x)q_\phi(z^s|x, y)}{q_\phi(z^s|x, y)} = q_\phi(y|x) \tag{22}$$

Substituting this back into the estimator yields the standard *M2* objective. However, the generalised importance sampling formulation allows for alternative factorisations, such as $q_\phi(y, z|x) = q_\phi(y|x, z)q_\phi(z|x)$, where the label is predicted from the latent style z.

3.3 Graphical Model and Stochastic Implementation

To perform gradient ascent, the generative and recognition models are mapped onto a stochastic computation graph where each node forms a sub-graph.
The implementation handles supervision through three specific node types. Firstly, for the fully supervised variable $\mathbf{x}$, Narayanaswamy et al. (2017) compute the likelihood $p_\theta(\mathbf{x}|\mathbf{y}, \mathbf{z}) = \mathcal{N}(\mathbf{x}; \eta_\theta(\mathbf{y}, \mathbf{z}))$, where η_θ is the encoder network returning the distribution parameters. Second, the unobserved latent variable $\mathbf{z}$ requires computing both the prior $p(\mathbf{z})$ and the conditional $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})$. This utilises the Gaussian reparametrisation trick $\mathbf{z} = g(\epsilon, \lambda_{\phi_z}(\mathbf{x}, \mathbf{y}))$ where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to ensure differentiability. Finally, the partially observed variable $\mathbf{y}$ is treated as observed when labels are available. When unavailable, it is sampled from $q_{\phi_y}(\mathbf{y}|\mathbf{x})$ using a Gumbel-Softmax relaxation to allow for gradient estimation on the discrete distribution.
Narayanaswamy et al. (2017) implement these different node types in a stochastic machine learning framework called ProbTorch which is available at: `https://github.com/probtorch/probtorch`.

3.4 Experiments and Findings

Narayanaswamy et al. (2017) evaluate the proposed framework across three distinct domains to demonstrate its ability to learn disentangled representations under partial supervision and varying structural complexity.
The authors do not provide full details about the implementation of their experiments. However, they do state that they use linear layers for Modified National Institute of Standards and Technology (MNIST), a convolutional network for Street View House Numbers (SVHN) and the Yale-B datasets, and a gated recurrent unit for the Multi-MNIST experiment. For the training process, they use Adam Kingma and Ba (2015) with the "default learning rate and momentum correction terms". Narayanaswamy et al. (2017) vaguely state that depending on the dataset a batch size between 100-700 is used. Experiment-specific information is mentioned in the following subsections. While further parameters are specified in their implementation on GitHub, these are unreliable as they deviate from the paper (e.g. the number of style variables).

3.4.1 Benchmark Classification: MNIST and SVHN

To validate the generalised objective against the standard *M2* model (Kingma et al., 2014), the authors first applied the framework to the MNIST (Deng, 2012) and SVHN (Netzer et al., 2011) datasets. The graphical model structure mirrored the *M2* setup, where the latent space consists of a partially observed class label y and an unobserved style variable z. They state that they use $\gamma = 1$ and introduce a supervision rate $\rho = \frac{\gamma M}{N+\gamma M}$. The α parameter is not specified.
On the MNIST dataset, classification performance is evaluated for supervised set sizes of 100, 600, 1000, and 3000 out of 50,000 total samples, where they marginally outperform the baseline *M2* model. In the more complex SVHN domain, the authors compare classification results for 1000 and 3000 labels out of the 70,000 unsupervised samples. They find that their SSVAE performs similarly compared to the two-stage *M1+M2* approach despite using a single-stage training process (Narayanaswamy et al., 2017). Qualitatively, the results demonstrated a clean separation of style and content. By fixing the style latent z and varying the label y, the model generated analogies where the same handwriting style was consistently applied to different digits. Conversely, fixing y and varying z produced variations in stroke width and slant while maintaining the digit identity.
Furthermore, the authors analysed the effect of the supervision weight γ and found that for sparsely labelled datasets ($M \ll N$), over-weighting the supervised term ($\gamma > 1$) significantly improved generalisation on the test set, although excessive weighting eventually led to overfitting. Here they offer conflicting statements about α, where the figure description states that $\alpha = 50$ for MNIST and $\alpha = 70$ for SVHN while the discussion of the results of the supervision rate mentions that they used $\alpha = 0.1/\rho$ in line with Kingma et al. (2014). For a more detailed description of experimental results, refer to Figure 3 of Narayanaswamy et al. (2017).

3.4.2 Intrinsic Faces: Disentangling Continuous Factors

Moving beyond categorical labels, the authors utilised the Yale-B dataset (Georghiades et al., 2001) to learn disentangled representations of *Identity* (categorical) and *Lighting* (continuous). The generative model defined latent variables for identity, lighting, shading, and reflectance, with partial supervision provided only for identity and lighting direction. The authors use a supervision rate of $\rho = 0.5$ with 6 supervised and 32 unsupervised data points, which implies $\gamma \approx 5.33$. α is set to 10. The authors demonstrate that the model disentangled identity from lighting conditions with qualitative samples where they manipulate lighting direction on a specific identity's face without altering facial features. Quantitatively, the framework outperformed the existing baseline from Jampani et al. (2015) on the identity classification task, achieving a semi-supervised error rate of 3.5% compared to approximately 30% for the baseline method. For the regression of the continuous lighting parameter, the semi-supervised model achieved an angular error of 17.6%, compared to approximately 10% for the fully supervised baseline from Jampani et al. (2015), demonstrating the efficacy of the objective for continuous latents even with limited supervision. For a more detailed description of experimental results, refer to Figure 4 of Narayanaswamy et al. (2017).

3.4.3 Multi-MNIST: Stochastic Dimensionality and Compositionality

The final experiment addressed a more complicated counting task using a custom Multi-MNIST dataset, where each image contained a varying number (1–3) of MNIST digits. This required a model with stochastic dimensionality, where the number of latent variables is itself a random variable K. The authors proposed a recursive generative model that sequentially samples digits and places them on a canvas using Spatial Transformer Networks. For this experiment, neither α, γ, nor ρ are reported by the authors. The model learned to reliably predict the number of digits K and successfully decomposed overlapping digits into their constituent parts and background. Additionally, the authors compared a flat model against a nested model that incorporated a pre-trained MNIST auto-encoder as a sub-component. This nested configuration demonstrated the framework's capacity for modular model design and transfer learning. While the flat model achieved higher raw counting accuracy, the nested model still effectively leveraged the pre-learned constituent representations to reconstruct complex multi-digit scenes and decompose inputs into coherent individual digits. For a more detailed description of experimental results, refer to Figure 6 of Narayanaswamy et al. (2017).

4 Extensions and Limitations

Several works build on the results from Narayanaswamy et al. (2017) and extend it to specific domains or in a general context. However, there are also some limitations and critiques in the subsequent literature. Both aspects are presented in the following sections. Relevant sources were systematically identified as outlined in Appendix A.

4.1 Own Criticism

Reviewing the work of Narayanaswamy et al. (2017) we identified several points of critique. For example, they claim a number of properties and findings that are not shown theoretically or empirically. One example of this in the introduction is that "a representation that has some factorisable structure, and consistent semantics associated to different parts, is more likely to generalise to a new task". This is not verified by their experiment or any cited source. In a later paper Locatello et al. (2019) investigated this claim and found no evidence to support it as further explained in subsubsection 4.2.1.
Further, the variable n is never defined. It comes up in the relationship

$$(n \cdot l) \times r + \epsilon \tag{23}$$

of the supervised variables in the Yale-B dataset (Georghiades et al., 2001) experiment we explain in subsubsection 3.4.2. The meaning of this variable only becomes clear after reading the cited work of Jampani et al. (2015) which declares it as the normal map.
Moreover, in their Yale-B dataset experiment Narayanaswamy et al. (2017) claim to learn the relationship between lighting, shading and reflectance from Equation 23. While they qualitatively demonstrate disentanglement as well as classification and regression performance, the learning of this relationship is not shown.
Another limitation of the current formulation is its reliance on a binary distinction between unsupervised data $\mathcal{D}_U$ and supervised data $\mathcal{D}_S$. This assumes a consistent set of available labels for the supervised portion. However, in complex real-world datasets involving K distinct supervised variables $\mathbf{y} = \{y_1, \ldots, y_K\}$, supervision is often heterogeneous; a data point $x^{(n)}$ may possess annotations for a random subset of variables $O^{(n)} \subseteq \mathbf{y}$, while remaining variables $U^{(n)} = \mathbf{y} \setminus O^{(n)}$ are missing.
Under the proposed framework, variables are treated as observed when available and sampled otherwise. Yet, implementing this dynamically in a batched stochastic gradient descent setting becomes non-trivial. The standard objective presented in Equation 10 breaks down because there is no single $\mathcal{L}_S$. Instead, the objective function effectively fractures into 2^K potential observation patterns. To handle this, the practitioner must manually specify unique loss components or intricate masking logic for each missingness pattern to correctly toggle between computing the likelihood $p(y_k)$ (for $y_k \in O^{(n)}$) and performing importance sampling using $q_\phi(y_k|x)$ (for $y_k \in U^{(n)}$). This combinatorial explosion necessitates a bespoke and brittle implementation of the stochastic computation graph, undermining the flexibility intended by the general framework.
Moreover, the architectural choice of a one-hot encoding for the discrete latent variables scales linearly with the number of classes. While allocating 10 neurons for 10 digits is functional for small datasets like MNIST, it fails to provide a compact representation for tasks

with a large number of classes. A distributed binary encoding (e.g., using independent Bernoulli distributions approximated via Gumbel-Softmax) would scale logarithmically, representing 2^n states with n neurons. The authors do not discuss the potential benefits of a denser, more scalable binary representation for learning disentangled factors.
Additionally, it is considered a good practice to reproduce experiments for methods presented in other papers instead of directly copying their results. This is not done by Narayanaswamy et al. (2017). For their MNIST/SVHN and Yale-B experiments (outlined in subsubsection 3.4.1 and subsubsection 3.4.2 respectively), they merely copy experimental result values from Kingma et al. (2014) and Jampani et al. (2015) instead of confirming the performance in their own experiments. Moreover, the definition of hyperparameters such as α and γ is not always clear, as alluded to in subsection 3.4. Another minor error is the wrong column description in figure 6 of Narayanaswamy et al. (2017) which states that the table on the bottom right reports count error, while the table actually contains accuracy values as correctly pointed out in the corresponding caption.
Lastly, in the same experiments, Narayanaswamy et al. (2017) do not mention which exact values they report. They state that they report error rates but not which metric they use. This complicates comparability to other methods and isolated interpretation of their results.

4.2 General Critiques

In the following, the semi-supervised disentanglement approach from Narayanaswamy et al. (2017) is confirmed by results from Locatello et al. (2019), which demonstrate the impossibility of stable unsupervised disentanglement. Casale et al. (2018) offer a general critique of the independent and identically distributed (i.i.d.) assumption in VAEs directly mentioning Narayanaswamy et al. (2017). Lastly, Mattei and Frellsen (2018) discuss a bias in using continuous versus discrete latents.

4.2.1 Impossibility of Unsupervised Disentanglement

In their paper, "Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations", Locatello et al. (2019) presented a proof that unsupervised disentanglement is fundamentally impossible without inductive biases on both the model and the data.
The proof relies on the geometry of the latent space and the properties of the Gaussian prior used in VAEs. Assuming a ground-truth generative process $x = f(z)$ where $z \sim P(z)$, then if $P(z)$ is a standard multivariate Gaussian, it is rotationally symmetric. That is, for any orthogonal rotation matrix R, the distribution of Rz is identical to the distribution of z.
Given a new generative function $f'(z) = f(R^T z)$. The marginal distribution of observations $p(x)$ produced by this new model is identical to the original:

$$p(x) = \int p(x|z)p(z)dz = \int p(x|R^T z)p(z)dz \tag{24}$$

Since the unsupervised ELBO only depends on the marginal likelihood of the data $p(x)$, it cannot distinguish between the true disentangled model $f(z)$ and the entangled model

$f'(z)$.
In their experiments Locatello et al. (2019) "do not find any evidence that the considered models can be used to reliably learn disentangled representations in an unsupervised manner as random seeds and hyperparameters seem to matter more than the model choice. Furthermore, well-trained models seemingly cannot be identified without access to ground-truth labels even if we are allowed to transfer good hyperparameter values across datasets".
This result confirms the necessity of some form of inductive bias for stable latent representations as proposed by Narayanaswamy et al. (2017). Narayanaswamy et al. (2017) anchor the variable y with labels, however, the variable z is essentially learned in an unsupervised manner relative to the style factors. Locatello et al. (2019)'s proof implies that there is no guarantee that z will factorise into meaningful sub-components purely based on the VAE objective. The model is free to learn any rotation of these style factors that explains the data.
Locatello et al. (2019) further highlight that "increased disentanglement does not seem to lead to a decreased sample complexity of learning for downstream tasks", shifting the focus more towards interpretability and fairness. While Narayanaswamy et al. (2017) claim the contrary, namely that consistent and factorisable representations are more likely to generalise to a new task, they never formally prove or demonstrate this empirically in their paper as already pointed out in subsection 4.1.

4.2.2 The Limitation of Isotropic Priors

Casale et al. (2018) critique the use of overly simplistic i.i.d. Gaussian priors. While Narayanaswamy et al. (2017) move away from such priors by composing latent graphical models with deep likelihoods, Casale et al. (2018) argue that the framework still often relies on conditional independence assumptions that may induce excessive regularization. Casale et al. (2018) point out that for many important datasets—such as time-series of images, medical scans of the same patient, or rotated views of an object—the samples exhibit structured correlations that are better captured through the prior's covariance.
Using a prior that ignores these sample-to-sample correlations is a model misspecification that forces the encoder to discard the correlation structure. Casale et al. (2018) introduced the Gaussian Process Prior VAE (GPPVAE), which replaces the independent prior with a Gaussian Process:

$$\mathbf{z} \sim \mathcal{GP}(0, K(X, X')) \tag{25}$$

Here, the covariance kernel K explicitly models the correlation between samples (e.g., temporal proximity or identity). This allows the model to disentangle "object identity" from "view" by leveraging the kernel structure, a form of disentanglement that Narayanaswamy et al. (2017)'s graphical approach typically handles through semi-supervised labels and specific message-passing variational families.
Esmaeili et al. (2019) also address the limitations of the isotropic Gaussian prior, arguing that it fails to exert sufficient regularizing pressure to force effective disentanglement. To resolve the common problem of disentangling discrete factors of variation from continuous variables—a task that remains problematic for many contemporary approaches—they

propose a structured decomposition of the ELBO objective. This modification allows for explicit control over the relative levels of disentanglement within different groups of latent variables. However, their analysis also highlights a significant optimization hurdle: discrete variables tend to exhibit higher likelihood values than continuous variables, which can introduce a bias that skews the optimization process in semi-supervised generative models.

4.2.3 Unbounded Likelihoods and Mode Collapse

Mattei and Frellsen (2018) investigated the maximum likelihood estimation for Deep Latent Variable Models (DLVMs), the class of models to which Narayanaswamy et al. (2017) SSVAE belongs. They proved that for continuous data, if the variance of the decoder's output distribution is learned without constraints, the likelihood function is unbounded.
Consider a decoder that outputs a Gaussian distribution $\mathcal{N}(\mu_\theta(z), \sigma^2_\theta(z))$. If the model can map a specific latent point z_i to exactly match a data point x_i (i.e., $\mu_\theta(z_i) \approx x_i$) and simultaneously drive the variance $\sigma^2_\theta(z_i)$ toward zero, the likelihood density $p(x_i \mid z_i)$ approaches infinity:

$$p(x_i|z_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_i - \mu)^2}{2\sigma^2}\right) \to \infty \text{ as } \sigma \to 0 \tag{26}$$

This creates a "hole" in the optimization landscape where the model collapses to storing data points (overfitting) rather than learning a generalised generative process. This is a severe limitation for Narayanaswamy et al. (2017)'s framework when applied to continuous data, as it can lead to numerical instability and meaningless latent representations where the "style" z degenerates into a lookup table index rather than a semantic factor. Mattei and Frellsen (2018) suggest constraints on σ or specific architectural choices to ensure the existence of generalised maximum likelihood estimates.

4.3 Disentanglement Implications and Limitations

4.3.1 Semantic Conflation Problem

Narayanaswamy et al. (2017)'s model feeds the label y and the latent z into the decoder: $p(x \mid y, z)$. The assumption is that y handles the class-specific information and z handles the rest. Joy et al. (2021) argue that this rigid separation is flawed because labels often imply continuous characteristics. Specifically, they state that "Originally motivated by the desiderata of semi–supervised classification, each label is given a corresponding latent variable of the same type (e.g. categorical), whose value is fixed to that of the label when the label is observed and imputed by the encoder when it is not. Though natural, we argue that this assumption is not just unnecessary but actively harmful from a representation-learning perspective, particularly in the context of performing manipulations. To allow manipulations, we want to learn latent factors that capture the characteristic information associated with a label, which is typically much richer than just the label value itself."
They provide an example pertaining to a dataset of faces. Consider a face with the label "Young". Joy et al. (2021) argue that age is not inherently discrete and that youth corre-

lates with a number of continuous features such as smooth skin, specific facial structures, and hair density. In a standard VAE, the decoder sees $y = 1$ ("Young") and learns to generate these features. Consequently, the latent z is absolved of the responsibility to encode the degree of youth or the specific way youth manifests in that image. This is semantic conflation: the discrete label "steals" the semantic content from the continuous latent space.
They state that this leads to a failure in disentanglement: if we want to manipulate the "age" of a face continuously, we cannot do so easily because the age information is locked inside the discrete variable y. Conversely, if we change the label y from "Young" to "Old" while keeping z fixed, the style z (which might encode "smiling") might be ignored or misinterpreted by the decoder because, so Joy et al. (2021) argue, the correlations between style and age are broken.
To resolve this, they proposed the Characteristic Capturing VAE (CCVAE) which builds on the SSVAE proposed by Narayanaswamy et al. (2017). They radically altered the graphical model. Instead of conditioning the generation on the label, they condition the inference of specific latents on the label. Specifically, they split the latent space into: (1) characteristic latents (z_c) which capture information correlated with the label and (2) salient latents (z_s) which capture the residual information.
The generative model is $p(x \mid z_c, z_s)$. Note that y is not an input to the decoder. Instead, an auxiliary classifier ensures that z_c is predictive of y: $\mathbb{E}[\log p(y \mid z_c)]$. By forcing z_c to predict y but feeding z_c (not y) to the decoder, the model must encode the semantic content of the label into the continuous space z_c. This avoids conflation. The label y acts as a supervisor for the structure of the latent space rather than a crutch for the decoder. Joy et al. (2021) demonstrated that this leads to superior performance in "attribute traversal." For example, dealing with the attribute "Young," the CCVAE could generate smooth transitions from young to old faces by interpolating in z_c, whereas the baseline models could only flip the binary switch y, resulting in discontinuous and often lower-quality transitions.
Nie et al. (2020) underlines this critique and states that "it still remains unclear how the use of supervision impacts the disentanglement learning". This is, in part, also due to a lack of relevant metrics (Adel et al., 2018), making it hard to quantify how well disentanglement works and how the semi-supervised framework of Narayanaswamy et al. (2017) impacts it.

4.3.2 Breaking the ELBO Bottleneck

Feng et al. (2021) identified a specific optimization pathology in loss functions of SSVAEs (Narayanaswamy et al., 2017), termed the *ELBO Bottleneck* and proposed SmootH-ELBO Optimal InTerpolation VAE (SHOT-VAE) as an extension of SSVAE to mitigate this limitation.
In SSVAEs, the objective combines the ELBO and a classification loss. Feng et al. (2021) observed that as training progresses, the ELBO term often plateaus before the inference accuracy is maximised. The standard ELBO does not strictly penalise misclassifications in the latent space as long as the reconstruction is good. This leads to a disconnect: the model might reconstruct the image of a "9" perfectly well even if the latent code z and label y are slightly misaligned or ambiguous.

In the SSVAEs framework proposed by Narayanaswamy et al. (2017), the latent space is explicitly partitioned into a discrete label y and a continuous style variable z. While this separation is designed to "anchor" semantics, Feng et al. (2021) identify a fundamental optimization pathology in this joint objective: the *ELBO Bottleneck*. Because the generative term $p(x|y,z)$ can achieve high likelihood by absorbing label-relevant information into the "unspecified" style variable z, the model often reaches an ELBO plateau where reconstruction is near-perfect despite poor classification accuracy. This indicates that the inductive bias intended by Narayanaswamy et al. (2017) is often bypassed during training, as the standard ELBO lacks the gradient pressure to prevent z from "leaking" the information that should be exclusive to y.
Feng et al. (2021) introduce SHOT-VAE with two key innovations (1) Smooth-ELBO, which is an approximation that integrates the label predictive loss directly into the ELBO derivation, rather than treating it as an auxiliary loss. This aligns the generative and discriminative objectives more tightly. (2) Optimal Interpolation, for which they utilised data augmentation in the latent space (mixup) to fill the gaps between class clusters. By enforcing linearity in the latent space (interpolating between a "1" and a "7" should yield a semantic blend), they break the ELBO bottleneck and force the encoder to learn a more robust, disentangled structure.
Empirically, SHOT-VAE achieved significant error rate reductions on CIFAR-10 (6.11% vs baseline) compared to standard SSVAEs from Kingma et al. (2014), demonstrating that optimization dynamics are as critical as model architecture for disentanglement.

4.3.3 Differentiating Consistency and Restrictiveness

Shu et al. (2020) introduces a refined definition of disentanglement that is made up of consistency and restrictiveness. Here, consistency is the degree to which a representation is deterministic with respect to the ground-truth factors. If we fix the ground-truth factor (e.g., color), the latent code should ideally be constant. On the other hand, restrictiveness is the degree to which a single dimension of the representation encodes only one ground-truth factor. This prevents a single latent dimension from encoding both color and shape.
Shu et al. (2020) state that Narayanaswamy et al. (2017) falsely claim that their method leads to disentangled results through semi-supervision. Instead, it merely creates consistent representations, which are not necessarily restrictive and hence not guaranteed to be disentangled. However, they relativise this by conceding that on real-world data consistency and restrictiveness are often strongly correlated.
These definitions provide a finer granularity than the mutual information metrics used previously. Critically, observational metrics (like the Mutual Information Gap (MIG) (Chen et al., 2018) used in Locatello et al. (2019)) often fail to capture these causal properties accurately. The differentiation of disentanglement introduced by Shu et al. (2020) paves the way for researchers to formally prove which types of weak supervision (e.g., restricted labeling, match-pairing) guarantee consistency or restrictiveness. This represents a theoretical maturation from Narayanaswamy et al. (2017)'s reliance on empirical validation via reconstruction.

4.4 Extensions in Supervision: From Semi to Weak

In response to the impossibility results from Locatello et al. (2019), the research community pivoted. If pure unsupervised disentanglement is impossible, and full supervision is expensive, what is the minimal signal required? Locatello et al. (2020) proposed Weakly-Supervised Disentanglement as a robust extension to the semi-supervised paradigm. Joy et al. (2022) further extend the concept of weakly-supervised VAEs to multimodal learning and inference.

4.4.1 The Weak Supervision Paradigm

While Locatello et al. (2019) highlights the need for an inductive bias to create stable representations, several works (Lin et al., 2020, Bouchacourt et al., 2018, Ke et al., 2024, Kim and Mnih, 2018) also critique the use of labels in the framework of Narayanaswamy et al. (2017). These criticisms are summarised well by Kim and Mnih (2018) who note that "semi-supervised approaches that require implicit or explicit knowledge about the true underlying factors of the data have excelled at disentangling" referring to Narayanaswamy et al. (2017). However, they also point out that "ideally we would like to learn these in an unsupervised manner, due to the following reasons: 1. Humans are able to learn factors of variation unsupervised [Perry et al. (2010)]. 2. Labels are costly as obtaining them requires a human in the loop. 3. Labels assigned by humans might be inconsistent or leave out the factors that are difficult for humans to identify".
Following these critiques and moving away from semi-supervision a number of weak-supervision frameworks emerged.
In "Weakly-Supervised Disentanglement Without Compromises," Locatello et al. (2020) model observations not as independent samples, but as pairs (x_1, x_2) that share at least one underlying factor of variation, even if the label of that factor is unknown.
For example, in a video of a moving arm, two adjacent frames might share the same "background color" and "object identity" but differ in "arm position." The weak label here is simply the knowledge that some factors are shared, without specifying what they are (unlike Narayanaswamy et al. (2017) who require the explicit label $y =$ "Digit 7").
They proved theoretically that knowing how many factors have changed (or stayed the same) between pairs is sufficient to guarantee disentanglement. This relaxes the requirement of Narayanaswamy et al. (2017) for expensive annotations for parts of the data while maintaining the capability of supervised disentanglement, only requiring a data collection process that provides paired samples (e.g., temporal coherence in video, or multi-camera setups).
Chen and Batmanghelich (2020) also critique the necessity for labels Narayanaswamy et al. (2017) and take a similar approach to Locatello et al. (2020), but focus more on the geometric structure of the latent space, utilizing a ranking-based loss to ensure that pairwise similarities in the data are preserved as proximity in specific latent dimensions. While Locatello et al. (2020) provide formal identifiability guarantees using *rank-1* pairs, where exactly one factor is varied, Chen and Batmanghelich (2020) rely on more flexible binary same/different judgments. This allows the model to learn from supervision that indicates whether any factor is shared, without requiring the explicit knowledge of which specific factor differs between the pair.

Yang and Yao (2019) extend these principles to hand pose estimation and image synthesis, claiming the ability to learn interpretable disentangled representations without the necessity for additional weak labels. However, their framework still fundamentally relies on explicit pose annotations to supervise the partitioning of the latent space into pose and appearance components. This highlights a recurring theme in the literature where, despite claims of minimizing supervision, the underlying disentanglement mechanism remains dependent on the primary task's labels to maintain structural integrity bringing us back to the impossibility theorem from Locatello et al. (2019).

4.4.2 Multimodal VAEs and Mutual Supervision

Joy et al. (2022), in "Learning Multimodal VAEs through Mutual Supervision," extended the semi-supervised framework to scenarios with multiple high-dimensional modalities (e.g., Image + Text).
In Narayanaswamy et al. (2017), supervision comes from a low-dimensional label y. In Multimodal VAEs, supervision comes from another rich modality. Joy et al. (2022) utilised a Product-of-Experts (PoE) aggregation. The joint posterior is approximated as the product of individual posteriors:

$$q(z|x_{\text{img}}, x_{\text{text}}) \propto q(z|x_{\text{img}}) \cdot q(z|x_{\text{text}}) \tag{27}$$

This structure enforces a form of inter-modality disentanglement. Similar to Locatello et al. (2020), the shared information (semantics) must be encoded in the intersection of the posteriors, while modality-specific noise is filtered out. This mutual supervision allows the model to learn robust representations even when one modality is missing, extending the partial-specification idea of Narayanaswamy et al. (2017) to complex, unstructured labels.

4.5 Applications and Integrations in the Literature

While multiple papers simply use SSVAE as a baseline in their experiments (Gordon et al., 2019, Kulinski and Inouye, 2023), others apply various aspects of Narayanaswamy et al. (2017)'s framework to their method. Vaze et al. (2023) critiques the prevalent use of synthetic data in the literature; several papers apply SSVAE to real-world problems. This section discusses several works that apply the concepts of SSVAEs or use it as a baseline in their benchmark, which provides perspective into the utility of the framework proposed by Narayanaswamy et al. (2017) in real-world conditions.

4.5.1 Domain Adaptation in the Medical Domain

Yang et al. (2019) applies the framework of SSVAEs to the critical problem of unsupervised domain adaptation in medical imaging (e.g., segmenting livers in CT vs. MRI scans). Building on the latent decomposition proposed by Narayanaswamy et al. (2017), Yang et al. (2019)'s framework utilises separate encoders (E_c content code and E_s style code) to partition representations into a shared content manifold for anatomical preservation and a domain-specific style subspace for modality features. This architecture facilitates many-to-many cross-domain translation via style-code swapping and adversarial

alignment, forcing the content space to converge on a modality-agnostic representation suitable for unsupervised liver segmentation.
By enforcing that the content code is shared while the style code is domain-specific, they could train a segmenter on labeled CT scans and apply it to unlabeled MRI scans by swapping the style codes. This effectively treats the "domain" (CT/MRI) as the specified factor y in Narayanaswamy et al. (2017)'s framework, but extends the mechanism to allow for image-to-image translation via the disentangled latent space.
Biswal et al. (2021) extend the semi-supervised generative framework to the medical domain through the development of EVA, a model designed for the generation of longitudinal electronic health records. Their approach builds upon the SSVAE architecture by introducing hierarchically factorised latent variables to better capture the complex dependencies inherent in temporal patient data. In this hierarchical structure, the upper-level latent variables are shared across the entire population to represent common clinical characteristics and global trends, while the lower-level variables remain patient-specific to capture individual disease trajectories and variations. By incorporating this multi-level factorization, the model ensures semantic consistency across the population while simultaneously allowing for the fine-grained representation of individual patient histories.
Finally, in the field of drug discovery, Li et al. (2018) utilise conditional graph generative models for multi-objective de novo design, identifying the SSVAE framework as a promising extension to further enhance their generative capabilities.

4.5.2 Causal Extension and Robustness

Zhang et al. (2020) provide a causal extension to the graphical latent variable framework, applying it to the study of neural network robustness. By adopting a setup similar to the semi-supervised generative models in Narayanaswamy et al. (2017), they frame the relationship between latent factors and observed data through a structural causal model (SCM). This causal lens allows for a formal investigation into how perturbations and distribution shifts propagate through the model's representations. Their work demonstrates that leveraging a graphical latent model is not only useful for disentanglement but is also critical for analyzing and improving the robustness of neural networks against adversarial or environmental interventions.

4.5.3 Catastrophic Forgetting and Generative Replay

To address catastrophic forgetting in deep learning models trained sequentially, Ye and Bors (2020) introduced the Lifelong VAEGAN, which combines the inference capabilities of VAEs with the high-quality generation of Generative Adversarial Networks (GANs). Instead of a standard teacher-student setup, the model employs a generative replay mechanism and a two-step "wake-dreaming" optimization. In the "wake" phase, the generator is updated to approximate the distribution of both current data and replayed samples from previous tasks. In the "dreaming" phase, the model maximises the log-likelihood of these samples to learn shared, disentangled latent representations across multiple domains.
Building on Narayanaswamy et al. (2017), Ye and Bors (2020) extend their Lifelong VAEGAN framework to the supervised setting and directly compare to SSVAE in their

experiments where it has the second-lowest classification error after the stacked generative semi-supervised model (M1+M2) model from Kingma et al. (2014).
In a later paper, Ye and Bors (2022) extend the SSVAE framework into the domain of continual learning by integrating it within a lifelong teacher-student network. Their method builds upon the generative principles established by Narayanaswamy et al. (2017) to facilitate knowledge retention and transfer across sequential learning tasks, utilizing the original model as a primary baseline for experimental comparison. Notably, while the authors formally cite Narayanaswamy et al. (2017), they attribute the architectural foundations to the Kingma et al. (2014) model in their technical descriptions, reflecting the shared heritage of these semi-supervised generative approaches. This extension demonstrates how the structured latent spaces of SSVAEs can be leveraged to mitigate catastrophic forgetting in dynamic data environments. Further, they find that the SSVAE architecture performs well in an experiment comparing the semi-supervised classification results on MNIST data, when considering MNIST to MNIST-Fashion lifelong learning. Specifically, they compare ten architectures with SSVAE as the second-lowest error.

5 Experiments

5.1 Implementation Details

Following previous work, our experiments were conducted with the MNIST dataset Deng (2012) using the digit label for partial supervision. We use the same fully connected feed-forward SSVAE architecture as Narayanaswamy et al. (2017). The specific setup used in the following experiments is detailed in Figure 3.

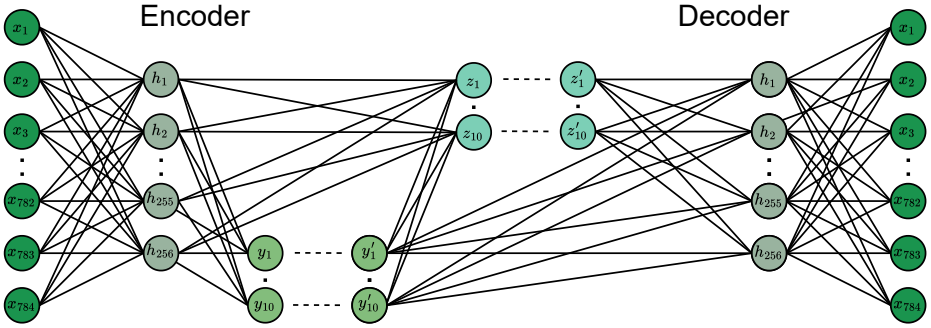


Figure 3: SSVAE architecture used throughout all our experiments. The encoder has $28 \times 28 = 784$ inputs $\{x_i\}_{i=1}^{784}$ matching the MNIST input and the decoder has the same number of output neurons with sigmoid activations and a binary cross-entropy loss. Both encoder and decoder have 256 hidden neurons $\{h_i\}_{i=1}^{256}$ with Rectified Linear Units (ReLU) (Nair and Hinton, 2010) activations. The latent space has 10 style variables $\{z_i\}_{i=1}^{10}$ and 10 digit neurons $\{y_i\}_{i=1}^{10}$. The decoder uses sampled $\{z'_i\}_{i=1}^{10}$ from a Gaussian and $\{y'_i\}_{i=1}^{10}$ from a Gumbel-Softmax distribution as inputs. Note that the dots between the neurons of a layer symbolise the remaining not shown neurons. Dashed lines symbolise the reparametrization trick and sampling process.

Following Narayanaswamy et al. (2017) we use the Adam optimiser (Kingma and Ba, 2015) with default parameters ($\beta_1 = 0.9$ and $\beta_2 = 0.999$), a learning rate of 10^{-3} and a batch size of 128. Unlike the original paper we train for 40 instead of 200 epochs due to computational constraints. As shown in Appendix B this is sufficient for all trained models to converge. The γ parameter from Equation 10 is set to the default value of 1. For the importance sampling, the default value of 8 samples is used. Each experiment is repeated with 10 different random seeds and for different supervised set sizes. Following Narayanaswamy et al. (2017) we use supervised set sizes M of 100 ($0.00\bar{16}\%$), 600 (0.01%), 1000 ($0.0\bar{16}\%$) and 3000 (0.05%) out of the 50,000 MNIST samples used for training. Experiments were run on an Nvidia Tesla P40 GPU and implemented with the ProbTorch package introduced by Narayanaswamy et al. (2017). To reproduce the experiments presented in this paper, see the code at: `https://github.com/davidbhofmann/ssvae`.

For the evaluation, we use accuracy to assess whether the semi-supervised latent y is correctly learned. To evaluate the unsupervised latents, we utilise three metrics that assess disentanglement through different information-theoretic lenses. The β-VAE Higgins et al. (2017) and Factor-VAE Kim and Mnih (2018) metrics provide a measure of how effectively the latent units isolate individual generative factors, specifically by penalizing statistical

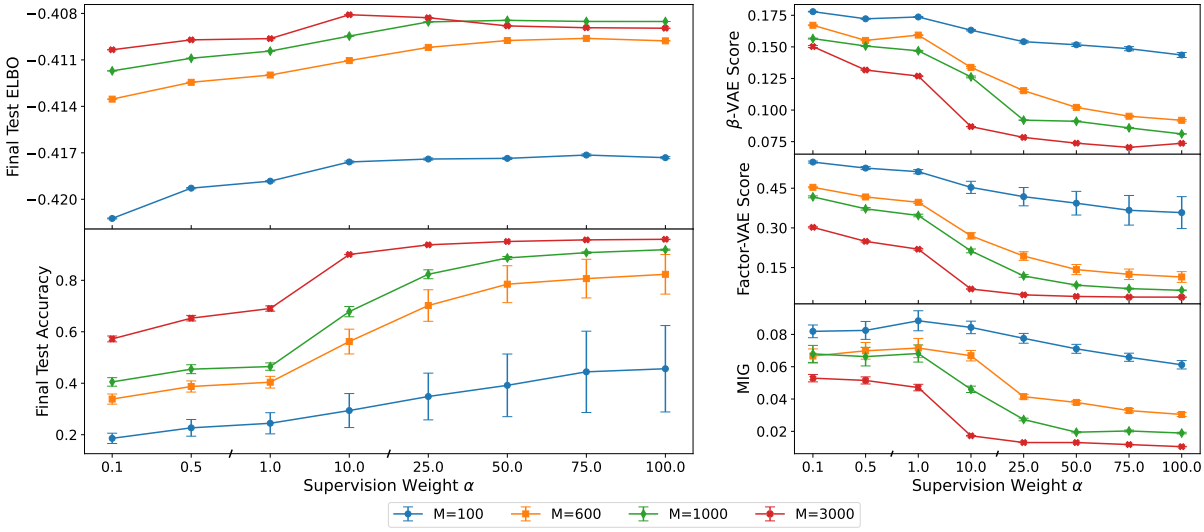


Figure 4: **Supervision Weight** experiment results with varying α for supervised set sizes M of 100 ($0.001\bar{6}\%$), 600 (0.01%), 1000 ($0.01\bar{6}\%$) and 3000 (0.05%). Error bars indicate variance across 10 random seeds. Note that the y-scales vary and do not start at zero. **Top Left:** final test ELBO for varying α remains relatively stable. **Bottom Left:** final test accuracy of y improves for higher values of α. **Right:** disentanglement metrics decrease with increasing α.

dependencies between dimensions to promote axis-alignment with the underlying data structure. Complementing these, the MIG offers a robust, scale-invariant measure of restrictiveness by calculating the difference in mutual information between the two latent variables that are most predictive of a given ground-truth factor. Together, these metrics ensure that the representation is not only consistent but also that information is concentrated within a single, interpretable latent dimension rather than being redundantly encoded across the bottleneck. Furthermore, the ELBO training objective is reported.

5.2 Supervision Weight

While Narayanaswamy et al. (2017) explore the supervision rate ρ and therefore indirectly γ, the supervision weight α is not explored by the authors. Furthermore, the authors make conflicting statements about the value of α in the MNIST experiments as critiqued in subsection 4.1. In the Yale-B experiment α is set to a different value without justification and in the Multi-MNIST experiments the value of α is never mentioned.
To address this, we explore how robust the SSVAE framework is with respect to the supervision weight α which we evaluate for eight different values: 0.1, 0.5, 1, 10, 25, 50, 75 and 100. The remaining configurations follow the setup explained in subsection 5.1.
Figure 4 shows that classification accuracy of the semi-supervised y variable is sensitive to the supervision weight α. As expected, increasing α leads to better classification performance, although the effect of increasing the value beyond 10 only leads to marginal improvements. We did not report final training accuracy, making it hard to asses overfitting in the experiments. However, we can still observe that there is no overfitting for

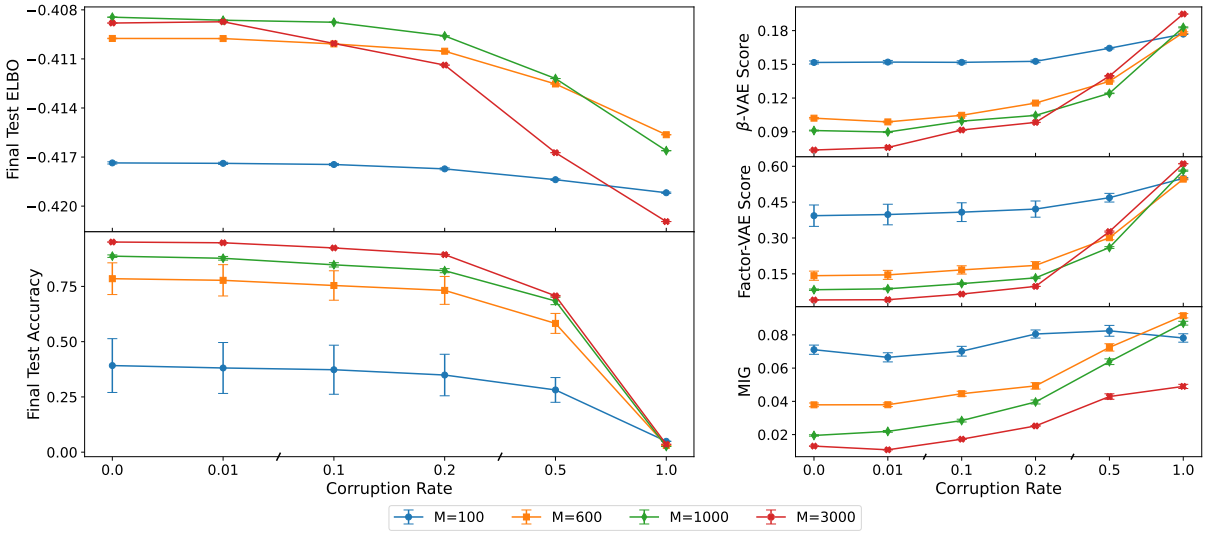


Figure 5: **Label Corruption** experiment results with varying label corruption rate for supervised set sizes M of 100 ($0.001\bar{6}\%$), 600 (0.01%), 1000 ($0.01\bar{6}\%$) and 3000 (0.05%). Error bars indicate variance across 10 random seeds. Note that the y-scales vary and do not start at zero. **Top Left:** final test ELBO for varying noise levels remains relatively stable. **Bottom Left:** while the final test accuracy of y decreases for higher corruption rates, it remains relatively high for up to 20% noise. **Right:** disentanglement increases with a higher corruption rate across all three metrics.

high α values to the point of decreasing test accuracy.
The final test ELBO remains relatively constant with respect to α, which can also be seen in Appendix B. A minor increase of the ELBO can be observed with increasing α. The results further show that for the unsupervised z, stronger supervision decreases disentanglement. A table with detailed results can be found in Appendix C.

5.3 Label Corruption

The most frequent critique identified in our review process was the use of labels (Lin et al., 2020, Bouchacourt et al., 2018, Ke et al., 2024, Kim and Mnih, 2018) as explained in subsubsection 4.4.1. One aspect of these critiques is the potential for incorrect human annotations.
To investigate the sensitivity of SSVAEs to potentially noisy labels, we use the setup as explained above and introduce randomly corrupted labels into the training process. Specifically, in each epoch a specified number of labels is flipped to a random class. We evaluate the SSVAEs for corruption rates of 0.0, 0.01, 0.1, 0.2, 0.5 and 1. Although it is not clear which α Narayanaswamy et al. (2017) use in their MNIST experiment, we set it to 50 coinciding with the value noted in the caption of Figure 3 of their paper. The remaining configurations follow the setup explained in subsection 5.1.
The result of this experiment is shown in Figure 5. As expected, we find that classification accuracy of the latent y variable decreases as the corruption rate of the labels increases. However, the accuracy does remain relatively stable up to a corruption rate of 20%.

Similar to the supervision weight experiments, the final test ELBO remains relatively constant. However, a marginal ELBO drop with increased corruption rates is noticeable across configurations. The results further show that for the unsupervised z, more label corruption in y increases disentanglement. A table with detailed results can be found in Appendix D.

6 Discussion

In this work, we reviewed the SSVAE framework by Narayanaswamy et al. (2017), which unifies deep generative models with partially-specified probabilistic graphical models. Unlike standard variational autoencoders that typically assume a flat latent space, this approach allows for the injection of domain knowledge through structured dependencies while retaining the flexibility of neural networks for function approximation. By incorporating arbitrary dependency structures, they extend previous work on semi-supervised variational autoencoders by Kingma et al. (2014).

While the majority of the literature builds upon the original SSVAE formulation by Kingma et al. (2014), there is a considerable body of work deriving from Narayanaswamy et al. (2017). However, few works critically evaluate the specific limitations of this method. Locatello et al. (2019) validate the necessity of inductive biases for disentanglement but critique the reliance on labels. This reliance is the most frequent critique in the broader literature (Lin et al., 2020, Bouchacourt et al., 2018, Ke et al., 2024, Kim and Mnih, 2018). Other works question the effect of supervision on disentanglement generally, rather than specifically addressing the SSVAE architecture. Notably, we find that these critiques often lack specific empirical evidence substantiating the failure modes of SSVAEs under realistic conditions.

Our experiments addressed this gap by investigating the robustness of SSVAEs to label noise and supervision intensity. We investigated the critique that noisy labels might harm the disentanglement process. While we observed that noisy labels decrease the classification accuracy of the semi-supervised latent y, the model demonstrated surprising robustness with a label corruption rate of up to 20% only marginally affected performance. We further analysed the effect of varying the supervision weight α. Increasing α improved the classification of y without leading to overfitting, even for values up to 100, and did not negatively impact the final test ELBO. Throughout our experiments, the ELBO remained relatively stable, which can be interpreted as evidence for the ELBO Bottleneck theory proposed by Feng et al. (2021) outlined in subsubsection 4.3.2. Here, they claim that the ELBO often plateaus before inference performance is maximised. Generally, we observed a trade-off: altering the strength of supervision, either directly through α and supervised set size or indirectly through label corruption, impacted the disentanglement of the unsupervised latent z. Specifically, z became more disentangled as supervision strength decreased. A potential explanation for this is the semantic conflation problem discussed in subsubsection 4.3.1. We hypothesise that as the model is forced to prioritise the y objective (due to high α), the optimization landscape becomes dominated by the classification loss, potentially starving the z variables of the gradient signal required to capture rich, independent semantic information. Future research could investigate whether the CCVAE proposed by Joy et al. (2021) can mitigate this trade-off.

These findings demonstrate the robustness of SSVAEs to noisy labels but also highlight their sensitivity to hyperparameter tuning. These conclusions are currently limited to the MNIST dataset; future research should verify these effects on more complex data. Additionally, this study explored a limited range for α and did not extensively track training accuracy, making it difficult to definitively rule out overfitting in all regimes. Future work could investigate the joint effects of varying α, the supervision ratio γ, and

label corruption simultaneously. Another limitation is the training for only 40 epochs. Although models converge in terms of the ELBO objective, the critique from Feng et al. (2021) suggests that this might not directly correspond to fully optimised inference and that further training could improve the latent representation.

7 Conclusion

In this seminar thesis, we provided a comprehensive review and empirical evaluation of the SSVAE framework proposed by Narayanaswamy et al. (2017). By unifying probabilistic graphical models with deep generative networks, this framework offers a principled method for injecting domain knowledge through partial supervision into the representation learning process. This approach addresses the fundamental limitations of purely unsupervised disentanglement, which Locatello et al. (2019) later proved to be theoretically impossible without such inductive biases.

Through our systematic literature review, we identified critiques such as the semantic conflation problem (Joy et al., 2022), which highlights that rigid partitioning of latent spaces can inadvertently starve continuous variables of semantic content. Furthermore, the field has largely evolved from the semi-supervised paradigm toward weak supervision (Locatello et al., 2020) and causal disentanglement (Zhang et al., 2020) to overcome the reliance on expensive explicit labels.

Our own empirical experiments extended the original analysis of Narayanaswamy et al. (2017) by investigating the model's sensitivity to supervision quality and intensity. We demonstrated that the SSVAE is remarkably robust to label corruption, maintaining high classification accuracy even when 20% of the supervision signal is randomised.

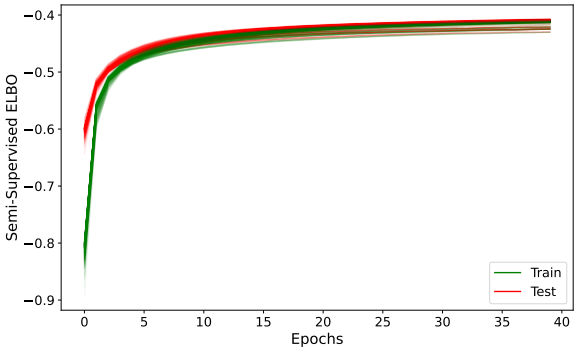
Ultimately, while newer methods have refined the mechanisms for disentanglement, the SSVAE remains a foundational framework. It successfully demonstrates that combining structured probabilistic modelling with the flexibility of deep neural networks is a viable path for learning interpretable, data-efficient representations.

A Systematic Review Process

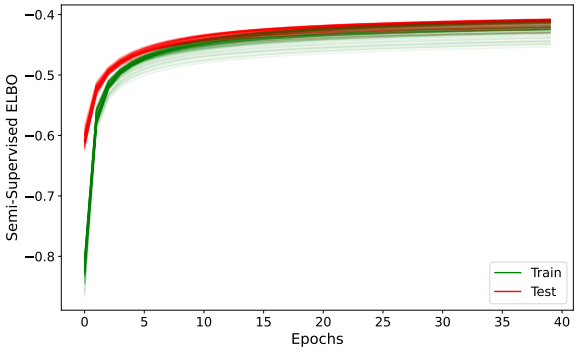
To identify relevant extensions and limitations of Narayanaswamy et al. (2017) in the literature, the results of several retrieval methods were synthesised. Note that the papers identified through the different methods had some overlap. For any paper to be considered, it had to fulfil one of the following selection criteria: (1) directly critique Narayanaswamy et al. (2017), (2) cite the paper in the context of a more general critique, (3) extend the SSVAE framework or (4) use it in an experiment.
Firstly, Connected Papers (`https://www.connectedpapers.com`) led to a list of 10 derivative works, of which only one matched the selection criteria.
Secondly, Google Scholar (`https://scholar.google.com`) was used to find relevant literature in the list of 445 papers that cite Narayanaswamy et al. (2017). The first 150 citing papers sorted by relevance were checked for the selection criteria. For the first 80 results, all paragraphs that mentioned Narayanaswamy et al. (2017) were reviewed in detail and for the remaining 70 papers, only papers that mentioned one of the following keywords in the title were considered: *semi-supervised*, *weak-supervision* and *label*. With this method, 21 relevant papers were identified.
Additionally, the search functionality of Google Scholar was used to find results for the search terms *semi-supervised disentanglement* and *supervised disentanglement*. For each term, the first 10 results were reviewed in detail, leading to a total of 4 relevant papers.
Lastly, Gemini-3-Pro was used to generate a report with relevant extensions and critiques which led to 5 relevant papers.
Note that during the review process we found that most derivative works build on the original SSVAE of Kingma et al. (2014) with 4094 citations on Google Scholar while Narayanaswamy et al. (2017) is cited 445 times.

B SSVAE Training

Figure 6a and Figure 6b show the training curve plots of all runs from both the supervision weight (α) and corruption rate experiments. The figures show that all runs converge without overfitting.



(a) All training runs of the **supervision weight experiment** in subsection 5.2



(b) All training runs of the **corruption rate experiment** in subsection 5.3

C Supervision Weight Results

The following section contains tables with aggregated results for the supervision weight experiment discussed in Table 1. Detailed results for each individual run can be found in Table 2.

Table 1: Aggregated results for the supervision weight experiment.

		β-VAE		Factor-VAE		MIG		Test ELBO		Test Accuracy	
		Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
M	α										
100	0.1	0.18	0.02	0.55	0.06	0.08	0.06	-0.42	0.01	0.19	0.14
	0.5	0.17	0.02	0.53	0.08	0.08	0.07	-0.42	0.01	0.23	0.18
	1.0	0.17	0.02	0.51	0.09	0.09	0.08	-0.42	0.01	0.24	0.20
	10.0	0.16	0.02	0.45	0.15	0.08	0.06	-0.42	0.01	0.29	0.26
	25.0	0.15	0.03	0.42	0.19	0.08	0.06	-0.42	0.01	0.35	0.30
	50.0	0.15	0.04	0.39	0.21	0.07	0.05	-0.42	0.01	0.39	0.35
	75.0	0.15	0.04	0.37	0.24	0.07	0.05	-0.42	0.01	0.44	0.40
	100.0	0.14	0.04	0.36	0.25	0.06	0.05	-0.42	0.01	0.46	0.41
600	0.1	0.17	0.02	0.45	0.05	0.07	0.07	-0.41	0.00	0.34	0.14
	0.5	0.16	0.02	0.42	0.06	0.07	0.07	-0.41	0.00	0.39	0.15
	1.0	0.16	0.02	0.40	0.07	0.07	0.08	-0.41	0.00	0.40	0.15
	10.0	0.13	0.03	0.27	0.11	0.07	0.06	-0.41	0.00	0.56	0.22
	25.0	0.12	0.03	0.19	0.13	0.04	0.04	-0.41	0.00	0.70	0.25
	50.0	0.10	0.03	0.14	0.14	0.04	0.03	-0.41	0.00	0.78	0.27
	75.0	0.10	0.03	0.12	0.14	0.03	0.04	-0.41	0.00	0.81	0.27
	100.0	0.09	0.03	0.11	0.14	0.03	0.04	-0.41	0.00	0.82	0.28
1000	0.1	0.16	0.02	0.42	0.06	0.07	0.07	-0.41	0.00	0.41	0.13
	0.5	0.15	0.02	0.37	0.07	0.07	0.08	-0.41	0.00	0.45	0.13
	1.0	0.15	0.02	0.35	0.07	0.07	0.07	-0.41	0.00	0.46	0.12
	10.0	0.13	0.03	0.21	0.08	0.05	0.04	-0.41	0.00	0.68	0.14
	25.0	0.09	0.01	0.12	0.07	0.03	0.03	-0.41	0.00	0.82	0.13
	50.0	0.09	0.02	0.08	0.05	0.02	0.02	-0.41	0.00	0.89	0.08
	75.0	0.09	0.02	0.07	0.04	0.02	0.02	-0.41	0.00	0.91	0.06
	100.0	0.08	0.01	0.06	0.04	0.02	0.02	-0.41	0.00	0.92	0.05
3000	0.1	0.15	0.03	0.30	0.05	0.05	0.05	-0.41	0.00	0.57	0.11
	0.5	0.13	0.01	0.25	0.05	0.05	0.05	-0.41	0.00	0.65	0.10
	1.0	0.13	0.02	0.22	0.06	0.05	0.04	-0.41	0.00	0.69	0.10
	10.0	0.09	0.01	0.07	0.01	0.02	0.01	-0.41	0.00	0.90	0.02
	25.0	0.08	0.01	0.05	0.01	0.01	0.01	-0.41	0.00	0.94	0.01
	50.0	0.07	0.00	0.04	0.00	0.01	0.01	-0.41	0.00	0.95	0.01
	75.0	0.07	0.01	0.04	0.00	0.01	0.01	-0.41	0.00	0.96	0.00
	100.0	0.07	0.01	0.04	0.00	0.01	0.01	-0.41	0.00	0.96	0.00

Table 2: Detailed results for the supervision weight experiment.

α	M	Seed	β-VAE	Factor-VAE	MIG	Test ELBO	Test Accuracy
0.10	100	42	0.19	0.54	0.02	-0.42	0.33
0.10	100	43	0.19	0.57	0.05	-0.42	0.03
0.10	100	44	0.17	0.64	0.15	-0.43	0.10

Continued on next page

Table 2 – continued from previous page

α	M	Seed	β-VAE	Factor-VAE	MIG	Test ELBO	Test Accuracy
0.10	100	45	0.16	0.52	0.10	-0.42	0.04
0.10	100	46	0.18	0.43	0.20	-0.41	0.38
0.10	100	47	0.17	0.52	0.03	-0.42	0.26
0.10	100	48	0.17	0.50	0.11	-0.42	0.15
0.10	100	49	0.18	0.60	0.02	-0.42	0.08
0.10	100	50	0.16	0.56	0.03	-0.42	0.38
0.10	100	51	0.21	0.61	0.12	-0.42	0.10
0.10	600	42	0.17	0.46	0.04	-0.41	0.49
0.10	600	43	0.17	0.48	0.04	-0.41	0.31
0.10	600	44	0.15	0.37	0.01	-0.41	0.47
0.10	600	45	0.16	0.52	0.10	-0.42	0.04
0.10	600	46	0.18	0.43	0.20	-0.41	0.38
0.10	600	47	0.16	0.45	0.01	-0.41	0.35
0.10	600	48	0.16	0.47	0.16	-0.41	0.25
0.10	600	49	0.20	0.52	0.08	-0.42	0.22
0.10	600	50	0.13	0.42	0.02	-0.41	0.45
0.10	600	51	0.18	0.41	0.03	-0.41	0.42
0.10	1000	42	0.16	0.38	0.03	-0.41	0.59
0.10	1000	43	0.15	0.45	0.05	-0.41	0.35
0.10	1000	44	0.14	0.30	0.02	-0.41	0.57
0.10	1000	45	0.15	0.51	0.07	-0.41	0.15
0.10	1000	46	0.18	0.43	0.19	-0.41	0.36
0.10	1000	47	0.16	0.45	0.01	-0.41	0.35
0.10	1000	48	0.16	0.44	0.20	-0.41	0.36
0.10	1000	49	0.19	0.45	0.09	-0.41	0.39
0.10	1000	50	0.12	0.37	0.01	-0.41	0.52
0.10	1000	51	0.16	0.38	0.01	-0.41	0.41
0.10	3000	42	0.14	0.28	0.00	-0.41	0.61
0.10	3000	43	0.14	0.29	0.07	-0.41	0.58
0.10	3000	44	0.12	0.23	0.00	-0.41	0.73
0.10	3000	45	0.19	0.38	0.04	-0.41	0.38
0.10	3000	46	0.15	0.29	0.11	-0.41	0.61
0.10	3000	47	0.13	0.29	0.03	-0.41	0.55
0.10	3000	48	0.18	0.37	0.13	-0.41	0.42
0.10	3000	49	0.18	0.33	0.10	-0.41	0.59
0.10	3000	50	0.11	0.25	0.01	-0.41	0.68
0.10	3000	51	0.16	0.31	0.02	-0.41	0.57
0.50	100	42	0.19	0.47	0.01	-0.42	0.50
0.50	100	43	0.19	0.57	0.05	-0.42	0.03
0.50	100	44	0.17	0.64	0.15	-0.43	0.10
0.50	100	45	0.16	0.52	0.10	-0.42	0.04
0.50	100	46	0.15	0.41	0.19	-0.41	0.38
0.50	100	47	0.17	0.46	0.00	-0.41	0.34
0.50	100	48	0.16	0.48	0.18	-0.41	0.24
0.50	100	49	0.18	0.60	0.02	-0.42	0.08
0.50	100	50	0.14	0.50	0.00	-0.42	0.46
0.50	100	51	0.21	0.61	0.12	-0.42	0.10
0.50	600	42	0.18	0.40	0.01	-0.41	0.54
0.50	600	43	0.14	0.39	0.06	-0.41	0.40
0.50	600	44	0.13	0.32	0.01	-0.41	0.48
0.50	600	45	0.16	0.52	0.10	-0.42	0.04
0.50	600	46	0.15	0.41	0.19	-0.41	0.38

Continued on next page

Table 2 – continued from previous page

α	M	Seed	β-VAE	Factor-VAE	MIG	Test ELBO	Test Accuracy
0.50	600	47	0.14	0.45	0.00	-0.41	0.35
0.50	600	48	0.16	0.45	0.20	-0.41	0.33
0.50	600	49	0.19	0.50	0.07	-0.41	0.33
0.50	600	50	0.14	0.35	0.04	-0.41	0.55
0.50	600	51	0.15	0.38	0.02	-0.41	0.47
0.50	1000	42	0.15	0.33	0.00	-0.41	0.62
0.50	1000	43	0.13	0.36	0.05	-0.41	0.42
0.50	1000	44	0.12	0.26	0.02	-0.41	0.61
0.50	1000	45	0.15	0.49	0.07	-0.41	0.20
0.50	1000	46	0.18	0.37	0.19	-0.41	0.46
0.50	1000	47	0.14	0.45	0.00	-0.41	0.35
0.50	1000	48	0.15	0.41	0.18	-0.41	0.38
0.50	1000	49	0.18	0.41	0.14	-0.41	0.41
0.50	1000	50	0.14	0.31	0.01	-0.41	0.58
0.50	1000	51	0.15	0.33	0.00	-0.41	0.52
0.50	3000	42	0.12	0.24	0.03	-0.41	0.68
0.50	3000	43	0.13	0.23	0.05	-0.41	0.74
0.50	3000	44	0.11	0.18	0.04	-0.41	0.78
0.50	3000	45	0.15	0.32	0.00	-0.41	0.45
0.50	3000	46	0.15	0.22	0.10	-0.41	0.69
0.50	3000	47	0.11	0.22	0.03	-0.41	0.68
0.50	3000	48	0.15	0.35	0.14	-0.41	0.48
0.50	3000	49	0.14	0.26	0.10	-0.41	0.68
0.50	3000	50	0.13	0.22	0.02	-0.41	0.69
0.50	3000	51	0.13	0.25	0.01	-0.41	0.65
1.00	100	42	0.18	0.43	0.02	-0.42	0.56
1.00	100	43	0.19	0.57	0.05	-0.42	0.03
1.00	100	44	0.17	0.64	0.15	-0.43	0.10
1.00	100	45	0.16	0.52	0.10	-0.42	0.04
1.00	100	46	0.18	0.37	0.21	-0.41	0.45
1.00	100	47	0.16	0.45	0.01	-0.41	0.35
1.00	100	48	0.17	0.47	0.20	-0.41	0.25
1.00	100	49	0.18	0.60	0.02	-0.42	0.08
1.00	100	50	0.14	0.46	0.01	-0.42	0.49
1.00	100	51	0.21	0.61	0.12	-0.42	0.10
1.00	600	42	0.17	0.37	0.02	-0.41	0.57
1.00	600	43	0.16	0.39	0.02	-0.41	0.42
1.00	600	44	0.14	0.30	0.01	-0.41	0.51
1.00	600	45	0.16	0.52	0.10	-0.42	0.04
1.00	600	46	0.18	0.37	0.21	-0.41	0.45
1.00	600	47	0.15	0.42	0.00	-0.41	0.36
1.00	600	48	0.15	0.44	0.20	-0.41	0.33
1.00	600	49	0.19	0.48	0.09	-0.41	0.36
1.00	600	50	0.16	0.35	0.05	-0.41	0.49
1.00	600	51	0.13	0.34	0.02	-0.41	0.52
1.00	1000	42	0.15	0.31	0.02	-0.41	0.50
1.00	1000	43	0.13	0.35	0.02	-0.41	0.44
1.00	1000	44	0.11	0.23	0.01	-0.41	0.65
1.00	1000	45	0.14	0.46	0.10	-0.41	0.26
1.00	1000	46	0.17	0.32	0.21	-0.41	0.52
1.00	1000	47	0.15	0.42	0.00	-0.41	0.36
1.00	1000	48	0.15	0.40	0.18	-0.41	0.40

Continued on next page

Table 2 – continued from previous page

α	M	Seed	β-VAE	Factor-VAE	MIG	Test ELBO	Test Accuracy
1.00	1000	49	0.18	0.39	0.09	-0.41	0.44
1.00	1000	50	0.15	0.34	0.04	-0.41	0.45
1.00	1000	51	0.14	0.26	0.02	-0.41	0.63
1.00	3000	42	0.13	0.19	0.04	-0.41	0.73
1.00	3000	43	0.11	0.16	0.09	-0.41	0.81
1.00	3000	44	0.10	0.15	0.04	-0.41	0.81
1.00	3000	45	0.15	0.28	0.01	-0.41	0.52
1.00	3000	46	0.13	0.25	0.05	-0.41	0.66
1.00	3000	47	0.11	0.21	0.01	-0.41	0.69
1.00	3000	48	0.15	0.33	0.13	-0.41	0.52
1.00	3000	49	0.14	0.24	0.10	-0.41	0.71
1.00	3000	50	0.12	0.17	0.00	-0.41	0.76
1.00	3000	51	0.12	0.22	0.00	-0.41	0.70
10.00	100	42	0.13	0.24	0.03	-0.41	0.72
10.00	100	43	0.19	0.57	0.05	-0.42	0.03
10.00	100	44	0.17	0.64	0.15	-0.43	0.10
10.00	100	45	0.16	0.52	0.10	-0.42	0.04
10.00	100	46	0.16	0.25	0.15	-0.41	0.62
10.00	100	47	0.14	0.36	0.01	-0.41	0.44
10.00	100	48	0.15	0.38	0.17	-0.41	0.38
10.00	100	49	0.18	0.60	0.02	-0.42	0.08
10.00	100	50	0.14	0.36	0.04	-0.41	0.43
10.00	100	51	0.21	0.61	0.12	-0.42	0.10
10.00	600	42	0.13	0.31	0.01	-0.41	0.48
10.00	600	43	0.10	0.26	0.06	-0.41	0.57
10.00	600	44	0.09	0.12	0.02	-0.41	0.82
10.00	600	45	0.16	0.52	0.10	-0.42	0.04
10.00	600	46	0.16	0.25	0.15	-0.41	0.62
10.00	600	47	0.13	0.29	0.02	-0.41	0.53
10.00	600	48	0.16	0.29	0.17	-0.41	0.58
10.00	600	49	0.16	0.30	0.08	-0.41	0.48
10.00	600	50	0.13	0.16	0.05	-0.41	0.76
10.00	600	51	0.12	0.19	0.01	-0.41	0.75
10.00	1000	42	0.12	0.23	0.03	-0.41	0.63
10.00	1000	43	0.15	0.27	0.08	-0.41	0.62
10.00	1000	44	0.08	0.07	0.00	-0.41	0.89
10.00	1000	45	0.15	0.35	0.13	-0.41	0.42
10.00	1000	46	0.13	0.19	0.10	-0.41	0.70
10.00	1000	47	0.13	0.29	0.02	-0.41	0.53
10.00	1000	48	0.15	0.24	0.00	-0.41	0.65
10.00	1000	49	0.16	0.23	0.04	-0.41	0.71
10.00	1000	50	0.11	0.15	0.02	-0.41	0.78
10.00	1000	51	0.10	0.10	0.03	-0.41	0.85
10.00	3000	42	0.08	0.09	0.02	-0.41	0.85
10.00	3000	43	0.09	0.06	0.01	-0.41	0.91
10.00	3000	44	0.09	0.06	0.03	-0.41	0.93
10.00	3000	45	0.07	0.08	0.04	-0.41	0.89
10.00	3000	46	0.08	0.06	0.01	-0.41	0.91
10.00	3000	47	0.10	0.06	0.03	-0.41	0.91
10.00	3000	48	0.09	0.08	0.00	-0.41	0.90
10.00	3000	49	0.10	0.07	0.01	-0.41	0.89
10.00	3000	50	0.09	0.06	0.00	-0.41	0.91

Continued on next page

Table 2 – continued from previous page

α	M	Seed	β-VAE	Factor-VAE	MIG	Test ELBO	Test Accuracy
10.00	3000	51	0.09	0.06	0.01	-0.41	0.91
25.00	100	42	0.12	0.25	0.04	-0.41	0.60
25.00	100	43	0.19	0.57	0.05	-0.42	0.03
25.00	100	44	0.17	0.64	0.15	-0.43	0.10
25.00	100	45	0.16	0.52	0.10	-0.42	0.04
25.00	100	46	0.12	0.15	0.11	-0.41	0.79
25.00	100	47	0.12	0.29	0.01	-0.41	0.56
25.00	100	48	0.14	0.30	0.15	-0.41	0.62
25.00	100	49	0.18	0.60	0.02	-0.42	0.08
25.00	100	50	0.13	0.26	0.02	-0.41	0.57
25.00	100	51	0.21	0.61	0.12	-0.42	0.10
25.00	600	42	0.12	0.15	0.02	-0.41	0.80
25.00	600	43	0.11	0.19	0.03	-0.41	0.68
25.00	600	44	0.07	0.06	0.00	-0.41	0.90
25.00	600	45	0.16	0.52	0.10	-0.42	0.04
25.00	600	46	0.12	0.15	0.11	-0.41	0.79
25.00	600	47	0.10	0.20	0.01	-0.41	0.71
25.00	600	48	0.11	0.18	0.07	-0.41	0.78
25.00	600	49	0.17	0.27	0.04	-0.41	0.63
25.00	600	50	0.09	0.10	0.02	-0.41	0.85
25.00	600	51	0.10	0.11	0.01	-0.41	0.85
25.00	1000	42	0.09	0.13	0.01	-0.41	0.81
25.00	1000	43	0.08	0.10	0.04	-0.41	0.87
25.00	1000	44	0.07	0.05	0.00	-0.41	0.93
25.00	1000	45	0.11	0.28	0.06	-0.41	0.50
25.00	1000	46	0.09	0.09	0.09	-0.41	0.88
25.00	1000	47	0.10	0.20	0.01	-0.41	0.71
25.00	1000	48	0.09	0.09	0.01	-0.41	0.88
25.00	1000	49	0.10	0.10	0.02	-0.41	0.87
25.00	1000	50	0.10	0.09	0.02	-0.41	0.88
25.00	1000	51	0.09	0.06	0.01	-0.41	0.92
25.00	3000	42	0.08	0.05	0.02	-0.41	0.93
25.00	3000	43	0.08	0.05	0.01	-0.41	0.94
25.00	3000	44	0.08	0.05	0.01	-0.41	0.94
25.00	3000	45	0.08	0.05	0.00	-0.41	0.93
25.00	3000	46	0.08	0.04	0.02	-0.41	0.95
25.00	3000	47	0.07	0.04	0.00	-0.41	0.95
25.00	3000	48	0.07	0.06	0.03	-0.41	0.93
25.00	3000	49	0.09	0.05	0.02	-0.41	0.93
25.00	3000	50	0.08	0.05	0.01	-0.41	0.94
25.00	3000	51	0.08	0.05	0.01	-0.41	0.94
50.00	100	42	0.13	0.20	0.02	-0.41	0.71
50.00	100	43	0.19	0.57	0.05	-0.42	0.03
50.00	100	44	0.17	0.64	0.15	-0.43	0.10
50.00	100	45	0.16	0.52	0.10	-0.42	0.04
50.00	100	46	0.09	0.09	0.09	-0.41	0.87
50.00	100	47	0.12	0.26	0.01	-0.41	0.56
50.00	100	48	0.14	0.23	0.12	-0.41	0.73
50.00	100	49	0.18	0.60	0.02	-0.42	0.08
50.00	100	50	0.13	0.22	0.02	-0.41	0.70
50.00	100	51	0.21	0.61	0.12	-0.42	0.10
50.00	600	42	0.09	0.09	0.02	-0.41	0.86

Continued on next page

Table 2 – continued from previous page

α	M	Seed	β-VAE	Factor-VAE	MIG	Test ELBO	Test Accuracy
50.00	600	43	0.07	0.09	0.01	-0.41	0.89
50.00	600	44	0.08	0.05	0.01	-0.41	0.92
50.00	600	45	0.16	0.52	0.10	-0.42	0.04
50.00	600	46	0.09	0.09	0.09	-0.41	0.87
50.00	600	47	0.10	0.11	0.03	-0.41	0.84
50.00	600	48	0.10	0.10	0.05	-0.41	0.88
50.00	600	49	0.12	0.22	0.04	-0.41	0.73
50.00	600	50	0.11	0.08	0.03	-0.41	0.89
50.00	600	51	0.10	0.07	0.00	-0.41	0.92
50.00	1000	42	0.09	0.07	0.01	-0.41	0.90
50.00	1000	43	0.07	0.07	0.01	-0.41	0.91
50.00	1000	44	0.07	0.04	0.01	-0.41	0.94
50.00	1000	45	0.14	0.23	0.03	-0.41	0.69
50.00	1000	46	0.09	0.06	0.07	-0.41	0.91
50.00	1000	47	0.10	0.11	0.03	-0.41	0.84
50.00	1000	48	0.09	0.07	0.00	-0.41	0.92
50.00	1000	49	0.10	0.07	0.03	-0.41	0.91
50.00	1000	50	0.09	0.07	0.01	-0.41	0.91
50.00	1000	51	0.09	0.05	0.00	-0.41	0.94
50.00	3000	42	0.07	0.04	0.02	-0.41	0.94
50.00	3000	43	0.08	0.04	0.03	-0.41	0.95
50.00	3000	44	0.08	0.04	0.00	-0.41	0.96
50.00	3000	45	0.07	0.04	0.00	-0.41	0.95
50.00	3000	46	0.08	0.04	0.00	-0.41	0.95
50.00	3000	47	0.08	0.04	0.02	-0.41	0.96
50.00	3000	48	0.07	0.05	0.00	-0.41	0.94
50.00	3000	49	0.07	0.04	0.02	-0.41	0.95
50.00	3000	50	0.07	0.04	0.02	-0.41	0.95
50.00	3000	51	0.08	0.04	0.01	-0.41	0.95
75.00	100	42	0.13	0.15	0.01	-0.41	0.81
75.00	100	43	0.19	0.57	0.05	-0.42	0.03
75.00	100	44	0.17	0.64	0.15	-0.43	0.10
75.00	100	45	0.16	0.52	0.10	-0.42	0.04
75.00	100	46	0.08	0.08	0.10	-0.41	0.90
75.00	100	47	0.13	0.18	0.01	-0.41	0.75
75.00	100	48	0.12	0.16	0.07	-0.41	0.82
75.00	100	49	0.18	0.60	0.02	-0.42	0.08
75.00	100	50	0.12	0.16	0.02	-0.41	0.83
75.00	100	51	0.21	0.61	0.12	-0.42	0.10
75.00	600	42	0.10	0.08	0.02	-0.41	0.89
75.00	600	43	0.09	0.07	0.01	-0.41	0.91
75.00	600	44	0.08	0.05	0.01	-0.41	0.94
75.00	600	45	0.16	0.52	0.10	-0.42	0.04
75.00	600	46	0.08	0.08	0.10	-0.41	0.90
75.00	600	47	0.09	0.09	0.02	-0.41	0.88
75.00	600	48	0.08	0.08	0.00	-0.41	0.89
75.00	600	49	0.10	0.16	0.05	-0.41	0.77
75.00	600	50	0.09	0.06	0.02	-0.41	0.92
75.00	600	51	0.09	0.06	0.01	-0.41	0.93
75.00	1000	42	0.08	0.06	0.02	-0.41	0.92
75.00	1000	43	0.06	0.05	0.01	-0.41	0.93
75.00	1000	44	0.07	0.04	0.01	-0.41	0.95

Continued on next page

Table 2 – continued from previous page

α	M	Seed	β-VAE	Factor-VAE	MIG	Test ELBO	Test Accuracy
75.00	1000	45	0.12	0.19	0.01	-0.41	0.75
75.00	1000	46	0.09	0.06	0.09	-0.41	0.92
75.00	1000	47	0.09	0.09	0.02	-0.41	0.88
75.00	1000	48	0.07	0.06	0.01	-0.41	0.93
75.00	1000	49	0.09	0.06	0.02	-0.41	0.92
75.00	1000	50	0.11	0.06	0.01	-0.41	0.92
75.00	1000	51	0.08	0.05	0.01	-0.41	0.94
75.00	3000	42	0.08	0.04	0.02	-0.41	0.95
75.00	3000	43	0.07	0.04	0.02	-0.41	0.96
75.00	3000	44	0.07	0.04	0.00	-0.41	0.96
75.00	3000	45	0.07	0.04	0.01	-0.41	0.95
75.00	3000	46	0.08	0.04	0.00	-0.41	0.96
75.00	3000	47	0.07	0.04	0.04	-0.41	0.96
75.00	3000	48	0.07	0.04	0.00	-0.41	0.95
75.00	3000	49	0.06	0.04	0.02	-0.41	0.96
75.00	3000	50	0.07	0.04	0.01	-0.41	0.95
75.00	3000	51	0.08	0.04	0.00	-0.41	0.96
100.00	100	42	0.12	0.13	0.00	-0.41	0.85
100.00	100	43	0.19	0.57	0.05	-0.42	0.03
100.00	100	44	0.17	0.64	0.15	-0.43	0.10
100.00	100	45	0.16	0.52	0.10	-0.42	0.04
100.00	100	46	0.07	0.06	0.09	-0.41	0.92
100.00	100	47	0.11	0.14	0.02	-0.41	0.80
100.00	100	48	0.11	0.14	0.03	-0.41	0.82
100.00	100	49	0.18	0.60	0.02	-0.42	0.08
100.00	100	50	0.11	0.16	0.02	-0.41	0.82
100.00	100	51	0.21	0.61	0.12	-0.42	0.10
100.00	600	42	0.08	0.06	0.03	-0.41	0.91
100.00	600	43	0.08	0.07	0.01	-0.41	0.92
100.00	600	44	0.08	0.05	0.01	-0.41	0.94
100.00	600	45	0.16	0.52	0.10	-0.42	0.04
100.00	600	46	0.07	0.06	0.09	-0.41	0.92
100.00	600	47	0.08	0.08	0.03	-0.41	0.89
100.00	600	48	0.09	0.08	0.01	-0.41	0.91
100.00	600	49	0.10	0.12	0.02	-0.41	0.84
100.00	600	50	0.09	0.06	0.01	-0.41	0.93
100.00	600	51	0.09	0.05	0.00	-0.41	0.94
100.00	1000	42	0.08	0.05	0.01	-0.41	0.93
100.00	1000	43	0.07	0.05	0.00	-0.41	0.93
100.00	1000	44	0.07	0.04	0.02	-0.41	0.95
100.00	1000	45	0.10	0.16	0.02	-0.41	0.79
100.00	1000	46	0.08	0.05	0.08	-0.41	0.94
100.00	1000	47	0.08	0.08	0.03	-0.41	0.89
100.00	1000	48	0.07	0.05	0.01	-0.41	0.94
100.00	1000	49	0.08	0.05	0.01	-0.41	0.93
100.00	1000	50	0.11	0.06	0.00	-0.41	0.93
100.00	1000	51	0.08	0.04	0.00	-0.41	0.95
100.00	3000	42	0.07	0.04	0.02	-0.41	0.96
100.00	3000	43	0.07	0.04	0.03	-0.41	0.96
100.00	3000	44	0.06	0.04	0.00	-0.41	0.97
100.00	3000	45	0.06	0.03	0.00	-0.41	0.96
100.00	3000	46	0.09	0.03	0.01	-0.41	0.96

Continued on next page

Table 2 – continued from previous page

α	M	Seed	β-VAE	Factor-VAE	MIG	Test ELBO	Test Accuracy
100.00	3000	47	0.08	0.04	0.03	-0.41	0.96
100.00	3000	48	0.06	0.04	0.00	-0.41	0.96
100.00	3000	49	0.08	0.04	0.00	-0.41	0.96
100.00	3000	50	0.07	0.04	0.00	-0.41	0.96
100.00	3000	51	0.10	0.04	0.00	-0.41	0.96

D Corruption Rate Results

The following section contains tables with aggregated results for the corruption rate experiment discussed in Table 3. Detailed results for each individual run can be found in Table 4.

Table 3: Aggregated results for the corruption rate experiment.

		β-VAE		Factor-VAE		MIG		Test ELBO		Test Accuracy	
		Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
M	Noise										
100	0.0	0.15	0.04	0.39	0.21	0.07	0.05	-0.42	0.01	0.39	0.35
	0.0	0.15	0.04	0.40	0.21	0.07	0.05	-0.42	0.01	0.38	0.34
	0.1	0.15	0.04	0.41	0.20	0.07	0.05	-0.42	0.01	0.37	0.33
	0.2	0.15	0.03	0.42	0.18	0.08	0.05	-0.42	0.01	0.35	0.31
	0.5	0.16	0.02	0.47	0.13	0.08	0.06	-0.42	0.01	0.28	0.24
	1.0	0.18	0.02	0.55	0.06	0.08	0.05	-0.42	0.01	0.05	0.03
600	0.0	0.10	0.03	0.14	0.14	0.04	0.03	-0.41	0.00	0.78	0.27
	0.0	0.10	0.03	0.15	0.14	0.04	0.03	-0.41	0.00	0.78	0.27
	0.1	0.10	0.03	0.17	0.13	0.04	0.04	-0.41	0.00	0.75	0.26
	0.2	0.12	0.03	0.19	0.13	0.05	0.04	-0.41	0.00	0.73	0.25
	0.5	0.13	0.02	0.30	0.09	0.07	0.05	-0.41	0.00	0.58	0.21
	1.0	0.18	0.03	0.55	0.06	0.09	0.04	-0.42	0.00	0.03	0.01
1000	0.0	0.09	0.02	0.08	0.05	0.02	0.02	-0.41	0.00	0.89	0.08
	0.0	0.09	0.02	0.09	0.06	0.02	0.02	-0.41	0.00	0.88	0.09
	0.1	0.10	0.02	0.11	0.06	0.03	0.03	-0.41	0.00	0.85	0.10
	0.2	0.10	0.02	0.13	0.06	0.04	0.03	-0.41	0.00	0.82	0.10
	0.5	0.12	0.02	0.26	0.06	0.06	0.04	-0.41	0.00	0.68	0.13
	1.0	0.18	0.02	0.58	0.05	0.09	0.03	-0.42	0.00	0.03	0.01
3000	0.0	0.07	0.00	0.04	0.00	0.01	0.01	-0.41	0.00	0.95	0.01
	0.0	0.08	0.01	0.04	0.00	0.01	0.01	-0.41	0.00	0.95	0.01
	0.1	0.09	0.01	0.07	0.01	0.02	0.01	-0.41	0.00	0.92	0.01
	0.2	0.10	0.01	0.10	0.01	0.03	0.01	-0.41	0.00	0.89	0.01
	0.5	0.14	0.02	0.33	0.05	0.04	0.04	-0.42	0.00	0.71	0.04
	1.0	0.19	0.02	0.61	0.02	0.05	0.04	-0.42	0.00	0.04	0.00

Table 4: Detailed results for the corruption rate experiment.

% Noise	M	Seed	β-VAE	Factor-VAE	MIG	Test ELBO	Test Accuracy
0.00	100	42	0.13	0.20	0.02	-0.41	0.71
0.00	100	43	0.19	0.57	0.05	-0.42	0.03
0.00	100	44	0.17	0.64	0.15	-0.43	0.10
0.00	100	45	0.16	0.52	0.10	-0.42	0.04
0.00	100	46	0.09	0.09	0.09	-0.41	0.87
0.00	100	47	0.12	0.26	0.01	-0.41	0.56
0.00	100	48	0.14	0.23	0.12	-0.41	0.73
0.00	100	49	0.18	0.60	0.02	-0.42	0.08
0.00	100	50	0.13	0.22	0.02	-0.41	0.70
0.00	100	51	0.21	0.61	0.12	-0.42	0.10
0.00	600	42	0.09	0.09	0.02	-0.41	0.86

Continued on next page

Table 4 – continued from previous page

% Noise	M	Seed	β-VAE	Factor-VAE	MIG	Test ELBO	Test Accuracy
0.00	600	43	0.07	0.09	0.01	-0.41	0.89
0.00	600	44	0.08	0.05	0.01	-0.41	0.92
0.00	600	45	0.16	0.52	0.10	-0.42	0.04
0.00	600	46	0.09	0.09	0.09	-0.41	0.87
0.00	600	47	0.10	0.11	0.03	-0.41	0.84
0.00	600	48	0.10	0.10	0.05	-0.41	0.88
0.00	600	49	0.12	0.22	0.04	-0.41	0.73
0.00	600	50	0.11	0.08	0.03	-0.41	0.89
0.00	600	51	0.10	0.07	0.00	-0.41	0.92
0.00	1000	42	0.09	0.07	0.01	-0.41	0.90
0.00	1000	43	0.07	0.07	0.01	-0.41	0.91
0.00	1000	44	0.07	0.04	0.01	-0.41	0.94
0.00	1000	45	0.14	0.23	0.03	-0.41	0.69
0.00	1000	46	0.09	0.06	0.07	-0.41	0.91
0.00	1000	47	0.10	0.11	0.03	-0.41	0.84
0.00	1000	48	0.09	0.07	0.00	-0.41	0.92
0.00	1000	49	0.10	0.07	0.03	-0.41	0.91
0.00	1000	50	0.09	0.07	0.01	-0.41	0.91
0.00	1000	51	0.09	0.05	0.00	-0.41	0.94
0.00	3000	42	0.07	0.04	0.02	-0.41	0.94
0.00	3000	43	0.08	0.04	0.03	-0.41	0.95
0.00	3000	44	0.08	0.04	0.00	-0.41	0.96
0.00	3000	45	0.07	0.04	0.00	-0.41	0.95
0.00	3000	46	0.08	0.04	0.00	-0.41	0.95
0.00	3000	47	0.08	0.04	0.02	-0.41	0.96
0.00	3000	48	0.07	0.05	0.00	-0.41	0.94
0.00	3000	49	0.07	0.04	0.02	-0.41	0.95
0.00	3000	50	0.07	0.04	0.02	-0.41	0.95
0.00	3000	51	0.08	0.04	0.01	-0.41	0.95
0.01	100	42	0.12	0.22	0.00	-0.41	0.66
0.01	100	43	0.19	0.57	0.05	-0.42	0.03
0.01	100	44	0.17	0.64	0.15	-0.43	0.10
0.01	100	45	0.16	0.52	0.10	-0.42	0.04
0.01	100	46	0.08	0.09	0.09	-0.41	0.88
0.01	100	47	0.12	0.26	0.02	-0.41	0.54
0.01	100	48	0.15	0.25	0.10	-0.41	0.69
0.01	100	49	0.18	0.60	0.02	-0.42	0.08
0.01	100	50	0.14	0.23	0.01	-0.41	0.69
0.01	100	51	0.21	0.61	0.12	-0.42	0.10
0.01	600	42	0.09	0.12	0.01	-0.41	0.82
0.01	600	43	0.08	0.10	0.02	-0.41	0.88
0.01	600	44	0.07	0.05	0.01	-0.41	0.92
0.01	600	45	0.16	0.52	0.10	-0.42	0.04
0.01	600	46	0.08	0.09	0.09	-0.41	0.88
0.01	600	47	0.09	0.11	0.03	-0.41	0.84
0.01	600	48	0.10	0.11	0.05	-0.41	0.87
0.01	600	49	0.12	0.22	0.04	-0.41	0.72
0.01	600	50	0.10	0.08	0.04	-0.41	0.89
0.01	600	51	0.10	0.07	0.00	-0.41	0.91
0.01	1000	42	0.09	0.07	0.01	-0.41	0.89
0.01	1000	43	0.07	0.07	0.02	-0.41	0.91
0.01	1000	44	0.06	0.05	0.01	-0.41	0.93

Continued on next page

Table 4 – continued from previous page

% Noise	M	Seed	β-VAE	Factor-VAE	MIG	Test ELBO	Test Accuracy
0.01	1000	45	0.15	0.25	0.03	-0.41	0.63
0.01	1000	46	0.10	0.07	0.08	-0.41	0.90
0.01	1000	47	0.09	0.11	0.03	-0.41	0.84
0.01	1000	48	0.09	0.06	0.01	-0.41	0.92
0.01	1000	49	0.10	0.07	0.02	-0.41	0.91
0.01	1000	50	0.09	0.07	0.01	-0.41	0.91
0.01	1000	51	0.08	0.05	0.01	-0.41	0.93
0.01	3000	42	0.07	0.04	0.01	-0.41	0.94
0.01	3000	43	0.08	0.04	0.02	-0.41	0.95
0.01	3000	44	0.07	0.04	0.01	-0.41	0.95
0.01	3000	45	0.06	0.04	0.01	-0.41	0.95
0.01	3000	46	0.08	0.04	0.01	-0.41	0.95
0.01	3000	47	0.08	0.04	0.02	-0.41	0.95
0.01	3000	48	0.07	0.05	0.01	-0.41	0.94
0.01	3000	49	0.08	0.04	0.00	-0.41	0.95
0.01	3000	50	0.08	0.04	0.02	-0.41	0.95
0.01	3000	51	0.09	0.05	0.00	-0.41	0.95
0.10	100	42	0.11	0.22	0.01	-0.41	0.70
0.10	100	43	0.19	0.57	0.05	-0.42	0.03
0.10	100	44	0.17	0.64	0.15	-0.43	0.10
0.10	100	45	0.16	0.52	0.10	-0.42	0.04
0.10	100	46	0.09	0.11	0.10	-0.41	0.86
0.10	100	47	0.12	0.30	0.01	-0.41	0.48
0.10	100	48	0.15	0.28	0.12	-0.41	0.65
0.10	100	49	0.18	0.60	0.02	-0.42	0.08
0.10	100	50	0.13	0.23	0.01	-0.41	0.70
0.10	100	51	0.21	0.61	0.12	-0.42	0.10
0.10	600	42	0.09	0.14	0.01	-0.41	0.81
0.10	600	43	0.07	0.11	0.01	-0.41	0.86
0.10	600	44	0.08	0.07	0.01	-0.41	0.90
0.10	600	45	0.16	0.52	0.10	-0.42	0.04
0.10	600	46	0.09	0.11	0.10	-0.41	0.86
0.10	600	47	0.10	0.15	0.02	-0.41	0.79
0.10	600	48	0.11	0.13	0.06	-0.41	0.83
0.10	600	49	0.13	0.25	0.09	-0.41	0.70
0.10	600	50	0.10	0.10	0.03	-0.41	0.86
0.10	600	51	0.11	0.09	0.02	-0.41	0.89
0.10	1000	42	0.10	0.09	0.01	-0.41	0.86
0.10	1000	43	0.09	0.09	0.03	-0.41	0.89
0.10	1000	44	0.08	0.06	0.01	-0.41	0.91
0.10	1000	45	0.15	0.26	0.03	-0.41	0.59
0.10	1000	46	0.10	0.09	0.10	-0.41	0.87
0.10	1000	47	0.10	0.15	0.02	-0.41	0.79
0.10	1000	48	0.10	0.08	0.03	-0.41	0.89
0.10	1000	49	0.10	0.08	0.01	-0.41	0.89
0.10	1000	50	0.09	0.10	0.00	-0.41	0.87
0.10	1000	51	0.09	0.07	0.03	-0.41	0.91
0.10	3000	42	0.08	0.07	0.01	-0.41	0.91
0.10	3000	43	0.09	0.07	0.04	-0.41	0.92
0.10	3000	44	0.11	0.06	0.02	-0.41	0.93
0.10	3000	45	0.08	0.06	0.01	-0.41	0.92
0.10	3000	46	0.10	0.06	0.01	-0.41	0.92

Continued on next page

Table 4 – continued from previous page

% Noise	M	Seed	β-VAE	Factor-VAE	MIG	Test ELBO	Test Accuracy
0.10	3000	47	0.09	0.06	0.03	-0.41	0.94
0.10	3000	48	0.08	0.07	0.01	-0.41	0.93
0.10	3000	49	0.09	0.07	0.03	-0.41	0.93
0.10	3000	50	0.10	0.07	0.01	-0.41	0.92
0.10	3000	51	0.11	0.07	0.02	-0.41	0.92
0.20	100	42	0.12	0.24	0.02	-0.41	0.62
0.20	100	43	0.19	0.57	0.05	-0.42	0.03
0.20	100	44	0.17	0.64	0.15	-0.43	0.10
0.20	100	45	0.16	0.52	0.10	-0.42	0.04
0.20	100	46	0.10	0.15	0.14	-0.41	0.82
0.20	100	47	0.13	0.31	0.05	-0.41	0.50
0.20	100	48	0.13	0.31	0.11	-0.41	0.57
0.20	100	49	0.18	0.60	0.02	-0.42	0.08
0.20	100	50	0.13	0.26	0.04	-0.41	0.64
0.20	100	51	0.21	0.61	0.12	-0.42	0.10
0.20	600	42	0.13	0.13	0.01	-0.41	0.82
0.20	600	43	0.08	0.13	0.02	-0.41	0.83
0.20	600	44	0.10	0.09	0.00	-0.41	0.87
0.20	600	45	0.16	0.52	0.10	-0.42	0.04
0.20	600	46	0.10	0.15	0.14	-0.41	0.82
0.20	600	47	0.10	0.17	0.02	-0.41	0.76
0.20	600	48	0.10	0.14	0.06	-0.41	0.83
0.20	600	49	0.15	0.27	0.07	-0.41	0.66
0.20	600	50	0.10	0.11	0.04	-0.41	0.85
0.20	600	51	0.12	0.13	0.04	-0.41	0.84
0.20	1000	42	0.10	0.11	0.02	-0.41	0.84
0.20	1000	43	0.08	0.10	0.04	-0.41	0.86
0.20	1000	44	0.09	0.09	0.01	-0.41	0.90
0.20	1000	45	0.14	0.28	0.09	-0.41	0.57
0.20	1000	46	0.10	0.12	0.10	-0.41	0.84
0.20	1000	47	0.10	0.17	0.02	-0.41	0.76
0.20	1000	48	0.10	0.11	0.04	-0.41	0.87
0.20	1000	49	0.10	0.11	0.01	-0.41	0.87
0.20	1000	50	0.10	0.13	0.01	-0.41	0.83
0.20	1000	51	0.12	0.11	0.06	-0.41	0.87
0.20	3000	42	0.12	0.07	0.02	-0.41	0.91
0.20	3000	43	0.08	0.10	0.01	-0.41	0.88
0.20	3000	44	0.10	0.10	0.04	-0.41	0.89
0.20	3000	45	0.09	0.09	0.02	-0.41	0.89
0.20	3000	46	0.09	0.09	0.02	-0.41	0.90
0.20	3000	47	0.11	0.09	0.01	-0.41	0.91
0.20	3000	48	0.10	0.09	0.01	-0.41	0.90
0.20	3000	49	0.10	0.10	0.04	-0.41	0.89
0.20	3000	50	0.10	0.12	0.04	-0.41	0.88
0.20	3000	51	0.11	0.12	0.04	-0.41	0.88
0.50	100	42	0.12	0.31	0.00	-0.41	0.42
0.50	100	43	0.19	0.57	0.05	-0.42	0.03
0.50	100	44	0.17	0.64	0.15	-0.43	0.10
0.50	100	45	0.16	0.52	0.10	-0.42	0.04
0.50	100	46	0.14	0.28	0.14	-0.41	0.66
0.50	100	47	0.16	0.39	0.07	-0.41	0.40
0.50	100	48	0.15	0.42	0.15	-0.41	0.44

Continued on next page

Table 4 – continued from previous page

% Noise	M	Seed	β-VAE	Factor-VAE	MIG	Test ELBO	Test Accuracy
0.50	100	49	0.18	0.60	0.02	-0.42	0.08
0.50	100	50	0.16	0.35	0.02	-0.41	0.55
0.50	100	51	0.21	0.61	0.12	-0.42	0.10
0.50	600	42	0.12	0.24	0.01	-0.41	0.66
0.50	600	43	0.12	0.26	0.08	-0.41	0.65
0.50	600	44	0.11	0.24	0.04	-0.41	0.71
0.50	600	45	0.16	0.52	0.10	-0.42	0.04
0.50	600	46	0.14	0.28	0.14	-0.41	0.66
0.50	600	47	0.13	0.30	0.02	-0.41	0.60
0.50	600	48	0.13	0.30	0.12	-0.41	0.63
0.50	600	49	0.17	0.41	0.11	-0.41	0.41
0.50	600	50	0.10	0.21	0.02	-0.41	0.74
0.50	600	51	0.16	0.25	0.08	-0.41	0.72
0.50	1000	42	0.11	0.21	0.02	-0.41	0.74
0.50	1000	43	0.11	0.19	0.08	-0.41	0.78
0.50	1000	44	0.13	0.27	0.03	-0.42	0.71
0.50	1000	45	0.14	0.41	0.11	-0.41	0.34
0.50	1000	46	0.13	0.26	0.12	-0.41	0.71
0.50	1000	47	0.13	0.30	0.02	-0.41	0.60
0.50	1000	48	0.12	0.24	0.08	-0.41	0.73
0.50	1000	49	0.11	0.22	0.07	-0.41	0.77
0.50	1000	50	0.11	0.23	0.01	-0.41	0.74
0.50	1000	51	0.16	0.28	0.10	-0.41	0.72
0.50	3000	42	0.11	0.23	0.07	-0.41	0.78
0.50	3000	43	0.12	0.31	0.02	-0.42	0.72
0.50	3000	44	0.13	0.38	0.00	-0.42	0.68
0.50	3000	45	0.14	0.29	0.02	-0.42	0.71
0.50	3000	46	0.14	0.32	0.03	-0.42	0.73
0.50	3000	47	0.16	0.35	0.01	-0.42	0.67
0.50	3000	48	0.12	0.29	0.06	-0.41	0.76
0.50	3000	49	0.14	0.36	0.08	-0.42	0.68
0.50	3000	50	0.13	0.37	0.00	-0.42	0.68
0.50	3000	51	0.19	0.36	0.12	-0.42	0.67
1.00	100	42	0.18	0.44	0.06	-0.41	0.04
1.00	100	43	0.19	0.57	0.05	-0.42	0.03
1.00	100	44	0.17	0.64	0.15	-0.43	0.10
1.00	100	45	0.16	0.52	0.10	-0.42	0.04
1.00	100	46	0.19	0.57	0.05	-0.41	0.02
1.00	100	47	0.16	0.48	0.05	-0.41	0.02
1.00	100	48	0.16	0.52	0.16	-0.41	0.03
1.00	100	49	0.18	0.60	0.02	-0.42	0.08
1.00	100	50	0.18	0.55	0.02	-0.41	0.03
1.00	100	51	0.21	0.61	0.12	-0.42	0.10
1.00	600	42	0.17	0.48	0.06	-0.41	0.05
1.00	600	43	0.14	0.47	0.03	-0.41	0.01
1.00	600	44	0.22	0.61	0.14	-0.42	0.03
1.00	600	45	0.16	0.52	0.10	-0.42	0.04
1.00	600	46	0.19	0.57	0.05	-0.41	0.02
1.00	600	47	0.16	0.49	0.07	-0.41	0.02
1.00	600	48	0.14	0.52	0.14	-0.41	0.02
1.00	600	49	0.20	0.59	0.11	-0.41	0.06
1.00	600	50	0.18	0.59	0.09	-0.42	0.03

Continued on next page

Table 4 – continued from previous page

% Noise	M	Seed	β-VAE	Factor-VAE	MIG	Test ELBO	Test Accuracy
1.00	600	51	0.22	0.62	0.12	-0.42	0.03
1.00	1000	42	0.18	0.57	0.11	-0.41	0.02
1.00	1000	43	0.18	0.56	0.06	-0.42	0.02
1.00	1000	44	0.22	0.62	0.12	-0.42	0.04
1.00	1000	45	0.15	0.51	0.13	-0.41	0.03
1.00	1000	46	0.19	0.63	0.03	-0.42	0.03
1.00	1000	47	0.16	0.49	0.07	-0.41	0.02
1.00	1000	48	0.18	0.61	0.12	-0.42	0.02
1.00	1000	49	0.20	0.63	0.06	-0.41	0.03
1.00	1000	50	0.17	0.60	0.08	-0.42	0.03
1.00	1000	51	0.20	0.59	0.08	-0.42	0.02
1.00	3000	42	0.18	0.57	0.00	-0.42	0.03
1.00	3000	43	0.20	0.61	0.06	-0.42	0.03
1.00	3000	44	0.21	0.60	0.07	-0.42	0.03
1.00	3000	45	0.20	0.61	0.03	-0.42	0.04
1.00	3000	46	0.19	0.62	0.02	-0.42	0.04
1.00	3000	47	0.18	0.62	0.06	-0.42	0.04
1.00	3000	48	0.18	0.64	0.07	-0.42	0.04
1.00	3000	49	0.20	0.60	0.01	-0.42	0.03
1.00	3000	50	0.18	0.60	0.06	-0.42	0.03
1.00	3000	51	0.22	0.64	0.12	-0.42	0.04

References

Adel, T., Ghahramani, Z. and Weller, A. (2018). Discovering interpretable representations for both deep generative and discriminative models, *in* J. Dy and A. Krause (eds), *Proceedings of the 35th International Conference on Machine Learning*, Vol. 80 of *Proceedings of Machine Learning Research*, PMLR, pp. 50–59.

Bengio, Y., Courville, A. and Vincent, P. (2013). Representation learning: A review and new perspectives, *IEEE transactions on pattern analysis and machine intelligence* **35**(8): 1798–1828.

Biswal, S., Ghosh, S., Duke, J., Malin, B., Stewart, W., Xiao, C. and Sun, J. (2021). Eva: Generating longitudinal electronic health records using conditional variational autoencoders, *Machine Learning for Healthcare Conference*, PMLR, pp. 260–282.

Bouchacourt, D., Tomioka, R. and Nowozin, S. (2018). Multi-level variational autoencoder: Learning disentangled representations from grouped observations, *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

Burda, Y., Grosse, R. and Salakhutdinov, R. (2016). Importance weighted autoencoders, *International Conference on Learning Representations*.

Casale, F. P., Dalca, A. V., Saglietti, L., Listgarten, J. and Fusi, N. (2018). Gaussian process prior variational autoencoders, *Advances in Neural Information Processing Systems (NeurIPS 2018)*.

Chen, J. and Batmanghelich, K. (2020). Weakly supervised disentanglement by pairwise similarities, *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, pp. 3495–3502.

Chen, R. T., Li, X., Grosse, R. B. and Duvenaud, D. K. (2018). Isolating sources of disentanglement in variational autoencoders, *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 31.

Deng, L. (2012). The mnist database of handwritten digit images for machine learning research, *IEEE Signal Processing Magazine* **29**(6): 141–142.

Esmaeili, B., Wu, H., Jain, S., Bozkurt, A., Siddharth, N., Paige, B., Brooks, D. H., Dy, J. and Meent, J.-W. (2019). Structured disentangled representations, *The 22nd International Conference on Artificial Intelligence and Statistics*, PMLR, pp. 2525–2534.

Feng, H.-Z., Kong, K., Chen, M., Zhang, T., Zhu, M. and Chen, W. (2021). Shot-vae: Semi-supervised deep generative models with label-aware elbo approximations, *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35, pp. 7413–7421.

Georghiades, A., Belhumeur, P. and Kriegman, D. (2001). From few to many: Illumination cone models for face recognition under variable lighting and pose, *IEEE Trans. Pattern Anal. Mach. Intelligence* **23**(6): 643–660.

Gordon, J., Bronskill, J., Bauer, M., Nowozin, S. and Turner, R. E. (2019). Meta-learning probabilistic inference for prediction, *International Conference on Learning Representations (ICLR)*.

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

Jampani, V., Eslami, S. M. A., Tarlow, D., Kohli, P. and Winn, J. (2015). Consensus Message Passing for Layered Graphical Models, *in* G. Lebanon and S. V. N. Vishwanathan (eds), *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, Vol. 38 of *Proceedings of Machine Learning Research*, PMLR, San Diego, California, USA, pp. 425–433.
URL: *https://proceedings.mlr.press/v38/jampani15.html*

Joy, T., Schmon, S. M., Torr, P. H. S., Siddharth, N. and Rainforth, T. (2021). Capturing label characteristics in vaes, *International Conference on Learning Representations (ICLR)*.

Joy, T., Shi, Y., Torr, P. H., Rainforth, T., Schmon, S. M. and Siddharth, N. (2022). Learning multimodal VAEs through mutual supervision, *International Conference on Learning Representations (ICLR)*.

Ke, Q., Jing, X., Woźniak, M., Xu, S., Liang, Y. and Zheng, J. (2024). Apgvae: Adaptive disentangled representation learning with the graph-based structure information, *Inf. Sci.* **657**(C).

Kim, H. and Mnih, A. (2018). Disentangling by factorising, *International Conference on Machine Learning*.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization, *International Conference on Learning Representations (ICLR)*.

Kingma, D. P., Rezende, D. J., Mohamed, S. and Welling, M. (2014). Semi-supervised Learning with Deep Generative Models, *Advances in Neural Information Processing Systems*, Vol. 27, Curran Associates, Inc.

Kingma, D. P. and Welling, M. (2014). Auto-Encoding Variational Bayes, *2nd International Conference on Learning Representations*.

Kulinski, S. and Inouye, D. I. (2023). Towards explaining distribution shifts, *Proceedings of the 40th International Conference on Machine Learning (ICML)*.

Li, Y., Zhang, L. and Liu, Z. (2018). Multi-objective de novo drug design with conditional graph generative model, *Journal of Cheminformatics* **10**(1): 33.

Lin, Z., Thekumparampil, K., Fanti, G. and Oh, S. (2020). InfoGAN-CR and ModelCentrality: Self-supervised model training and selection for disentangling GANs, *in*

H. D. III and A. Singh (eds), *Proceedings of the 37th International Conference on Machine Learning*, Vol. 119 of *Proceedings of Machine Learning Research*, PMLR, pp. 6127–6139.

Locatello, F., Bauer, S., Lucic, M., Gelly, S., Schoelkopf, B. and Bachem, O. (2019). Challenging common assumptions in the unsupervised learning of disentangled representations, *Proceedings of the 36th International Conference on Machine Learning (ICML)*.

Locatello, F., Poole, B., Raetsch, G., Schoelkopf, B., Bachem, O. and Tschannen, M. (2020). Weakly-supervised disentanglement without compromises, *Proceedings of the 37th International Conference on Machine Learning (ICML)*.

Mattei, P.-A. and Frellsen, J. (2018). Leveraging the exact likelihood of deep latent variable models, *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 31.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines, *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814.

Narayanaswamy, S., Paige, B., van de Meent, J.-W., Desmaison, A., Goodman, N., Kohli, P., Wood, F. and Torr, P. (2017). Learning Disentangled Representations with Semi-Supervised Deep Generative Models, *Advances in Neural Information Processing Systems*, Vol. 30, Curran Associates, Inc.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A. Y. et al. (2011). Reading digits in natural images with unsupervised feature learning, *NIPS workshop on deep learning and unsupervised feature learning*, Granada, p. 7.

Nie, W., Karras, T., Garg, A., Debnath, S., Patney, A., Patel, A. and Anandkumar, A. (2020). Semi-supervised StyleGAN for disentanglement learning, *in* H. D. III and A. Singh (eds), *Proceedings of the 37th International Conference on Machine Learning*, Vol. 119 of *Proceedings of Machine Learning Research*, PMLR, pp. 7360–7369.

Perry, G., Rolls, E. T. and Stringer, S. M. (2010). Continuous transformation learning of translation invariant representations, *Experimental Brain Research* **204**(2): 255–270.

Shu, R., Chen, Y., Kumar, A., Ermon, S. and Poole, B. (2020). Weakly supervised disentanglement with guarantees, *International Conference on Learning Representations (ICLR)*.

Vaze, S., Vedaldi, A. and Zisserman, A. (2023). No representation rules them all in category discovery, *Advances in Neural Information Processing Systems (NeurIPS)*.

Yang, J., Dvornek, N. C., Zhang, F., Chapiro, J., Lin, M. and Duncan, J. S. (2019). Unsupervised domain adaptation via disentangled representations: Application to cross-modality liver segmentation, *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, pp. 255–263.

Yang, L. and Yao, A. (2019). Disentangling latent hands for image synthesis and pose estimation, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ye, F. and Bors, A. G. (2020). Learning latent representations across multiple data domains using lifelong vaegan, *European Conference on Computer Vision (ECCV)*.

Ye, F. and Bors, A. G. (2022). Lifelong teacher-student network learning, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **44**(10): 6280–6296.

Zhang, C., Zhang, K. and Li, Y. (2020). A causal view on robustness of neural networks, *Advances in Neural Information Processing Systems (NeurIPS)*.

AI Tools

I hereby confirm that I have written all parts of this work myself. I used Gemini-3-Pro for linguistic corrections and to improve reading flow throughout all sections. Moreover, the model was used to generate initial formulation suggestions for the background section, as a critical reviewer offering feedback, and as a tutor to discuss the work of Narayanaswamy et al. (2017). All AI suggestions were manually reviewed, adapted, and reformulated; no generated text was adopted verbatim. Additionally, the Python code for Figure 4 and Figure 5 was generated using Gemini-3-Pro and adopted only after my manual verification of correctness. The model was also used to assist in the discovery and explanation of sources for the literature review, as detailed in Appendix A. Furthermore, Microsoft Copilot was used for code optimization and refactoring. Specifically, I used Copilot to implement multi-GPU support and to modularise the model implementation from Narayanaswamy et al. (2017). All resulting code changes were tested, comprehended, and validated by me to ensure full accuracy.

Declaration of authorship

I hereby declare that the report submitted is my own unaided work. All direct or indirect sources used are acknowledged as references. I am aware that the Thesis in digital form can be examined for the use of unauthorised aid and in order to determine whether the report as a whole or parts incorporated in it may be deemed as plagiarism. For the comparison of my work with existing sources I agree that it shall be entered in a database where it shall also remain after examination, to enable comparison with future theses submitted. Further rights of reproduction and usage, however, are not granted here. This paper was not previously presented to another examination board and has not been published.

Munich, 06.02.2026

David B. Hoffmann