arXiv:2006.10102v1 [cs.LG] 17 Jun 2020

Rethinking Semi–Supervised Learning in VAEs

Tom Joy
University of Oxford
tomjoy@robots.ox.ac.uk

Sebastian M. Schmon
University of Oxford
schmon@stats.ox.ac.uk

Philip H. S. Torr
University of Oxford
philip.torr@eng.ox.ac.uk

N. Siddharth*
University of Oxford
nsid@robots.ox.ac.uk

Tom Rainforth*
University of Oxford
rainforth@stats.ox.ac.uk

Abstract

We present an alternative approach to semi-supervision in variational autoencoders (VAEs) that incorporates labels through auxiliary variables rather than directly through the latent variables. Prior work has generally conflated the meaning of labels, i.e. the associated *characteristics* of interest, with the actual label values themselves—learning latent variables that directly correspond to the label values. We argue that to learn meaningful representations, semi-supervision should instead try to capture these richer characteristics and that the construction of latent variables as label values is not just unnecessary, but actively harmful. To this end, we develop a novel VAE model, the *reparameterized* VAE (REVAE), which "reparameterizes" supervision through auxiliary variables and a concomitant variational objective. Through judicious structuring of mappings between latent and auxiliary variables, we show that the REVAE can effectively learn meaningful representations of data. In particular, we demonstrate that the REVAE is able to match, and even improve on the classification accuracy of previous approaches, but more importantly, it also allows for more effective and more general interventions to be performed. We include a demo of REVAE at https://github.com/thwjoy/revae-demo.

1 Introduction

Learning the characteristic factors of perceptual observations has long been desired for effective machine intelligence [2, 1, 8, 26]. In particular, the ability to learn *meaningful* factors—capturing human-understandable characteristics from data—has been of interest from the perspective of human-like learning [27, 13] and improving decision making and generalization across tasks [1, 27].

At a fundamental level, learning meaningful representations of data allows one to a) make predictions and b) manipulate factors, for individual data points. Prediction provides a mechanism to *interpret* observations in terms of the different meaningful factors. Manipulation allows for the expression of *causal* effects between the meaning of factors and their corresponding realizations in the data. This can be further categorized into conditional generation—the ability to construct whole exemplar data instances with characteristics dictated by constraining relevant factors—and intervention—the ability to manipulate just particular factors for a given data point, and subsequently affect only the associated characteristics. Together, the prediction and manipulation tasks can be used to construct measures of fidelity and robustness of the learned representations, and of how meaningful they actually are.

A particularly flexible and powerful framework within which to explore the learning of meaningful representations are variational autoencoders (VAEs), a class of deep generative latent-variable models, where representations of data are captured in the *latent variables* of the underlying generative model. Learning meaningful factors in this framework typically manifests in the form of constraints, or *inductive biases*, [14, 17, 16, 11, 23, 28]—effected via either the model, objective, data, or the learning algorithm, or some combination thereof.

*equal contribution

Preprint. Under review.

One such constraint is semi-supervised learning, where labels are provided for some small subset of the observed data. In this setting, the expectation is for the learned factors to capture the *denotation*—the characteristic meaning—of the given labels, by establishing a correspondence between factors and labels. The label *values* themselves are simply a discrete set of *prototypes* [21, 5] within the category of characteristics denoted by the label, i.e. the label values are used to refer to the class meaning, and do not describe the class by themselves. For example, the label "female" can be seen as denoting a varied set of characteristics, whereas the label value itself is simply a prototypical element in that set, and not the entire set itself (cf. Figure 1).

This is particularly relevant in the VAE setting, as it directly affects the question of how latent variables capture meaningful representations from labels. While it might be tempting to directly define some set of latent variables as labels, as done by prior work [11, 23, 15], this conflates the denotation of the labels with its values—we want the representations to capture the characteristic meaning of the labels, not just the label values themselves.

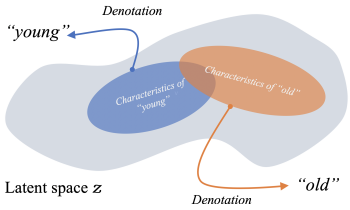

Figure 1: Conceptualizing the distinction between a label's *value* and its *denotation*.

Here, we develop a novel framework for semi-supervised learning in VAEs that respects the distinction between a label's denotation and its values. The framework, which we refer to as the *reparameterized* VAE (REVAE), employs a novel VAE formulation that "reparameterizes" latent variables corresponding to labels through the introduction of auxiliary variables. Along with a principled variational objective and careful structuring of the mappings between latent and auxiliary variables, we show that REVAEs successfully capture meaningful representations, while also enabling better performance on prediction and manipulation tasks. In particular, they permit certain manipulation tasks that cannot be performed with conventional approaches, such as manipulating denotations *without* changing the labels themselves and producing *multiple* distinct samples consistent with the desired intervention. We summarize our contributions as follows:

i) showing how semi-supervision can be used to disentangle rich *characteristic* information through careful structuring of the latent space and the introduction of auxiliary variables;
ii) formulating REVAEs, a novel model class and objective for semi-supervised learning in VAEs that respects the distinction between labels and their denotations;
iii) developing of a set of quantitative measures for manipulation, including both conditional generation and interventions, that highlight the required *invariances*; and
iv) demonstrating REVAEs' ability to successfully learn meaningful representations in practice.

2 Background

VAEs [10, 20] are a powerful and flexible class of model that combine the unsupervised representation-learning capabilities of deep autoencoders [9] with generative latent-variable models—a popular tool to capture factored low-dimensional representations of higher-dimensional observations. In contrast to deep autoencoders, generative models capture representations of data not as distinct values corresponding to observations, but rather as *distributions* of values. A generative model defines a joint distribution over observed data $\boldsymbol{x}$ and latent variables $\boldsymbol{z}$ as $p_\theta(\boldsymbol{x}, \boldsymbol{z}) = p(\boldsymbol{z})p_\theta(\boldsymbol{x} \mid \boldsymbol{z})$. Given a model, learning representations of data can be viewed as performing *inference*—learning the *posterior* distribution $p_\theta(\boldsymbol{z} \mid \boldsymbol{x})$ that constructs the distribution of latent values for a given observation.

VAEs employ amortized variational inference (VI) [29, 10] using the encoder and decoder of an autoencoder to transform this setup by i) taking the model likelihood $p_\theta(\boldsymbol{x} \mid \boldsymbol{z})$ to be parameterized by a neural network using the *decoder*, and ii) constructing an amortized variational approximation $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ to the (intractable) posterior $p_\theta(\boldsymbol{z} \mid \boldsymbol{x})$ using the *encoder*. The variational approximation of the posterior enables effective estimation of the objective—maximizing the marginal likelihood—through importance sampling, to derive the evidence lower bound (ELBO) of the model as

$$\log p_\theta(\boldsymbol{x}) = \log \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\frac{p_\theta(\boldsymbol{z}, \boldsymbol{x})}{q_\phi(\boldsymbol{z} \mid \boldsymbol{x})}\right] \geq \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p_\theta(\boldsymbol{z}, \boldsymbol{x})}{q_\phi(\boldsymbol{z} \mid \boldsymbol{x})}\right] \equiv \mathcal{L}(\boldsymbol{x}; \phi, \theta). \tag{1}$$

Given observations $\mathcal{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ taken to be realizations of random variables generated from an unknown distribution $p_{\mathcal{D}}(\boldsymbol{x})$, the overall objective is $\frac{1}{N}\sum_n \mathcal{L}(\boldsymbol{x}_n; \theta, \phi)$.

Semi-supervised VAEs (SSVAEs) [11] consider the setting where a subset of data $\mathcal{S} \subset \mathcal{D}$ is assumed to also have corresponding *labels* $\boldsymbol{y}$. Denoting the rest of the (unlabeled) data as $\mathcal{U} = \mathcal{D} \backslash \mathcal{S}$, the

log-marginal likelihood can be decomposed as

$$\log p\left(\mathcal{D}\right) = \sum\nolimits_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{S}} \log p_\theta(\boldsymbol{x},\boldsymbol{y}) + \sum\nolimits_{\boldsymbol{x}\in\mathcal{U}} \log p_\theta(\boldsymbol{x}).$$

In the most common SSVAE approach, known as M2 [11], the supervised term is estimated by taking label $\boldsymbol{y}$ to be an observed variable, yielding

$$\log p_\theta(\boldsymbol{x},\boldsymbol{y}) \geq \mathcal{L}(\boldsymbol{x},\boldsymbol{y};\theta,\phi) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})}\left[\log\left(\frac{p_\theta(\boldsymbol{x}\mid\boldsymbol{y},\boldsymbol{z})p(\boldsymbol{z})p(\boldsymbol{y})}{q(\boldsymbol{z}\mid\boldsymbol{x},\boldsymbol{y})}\right)\right].$$

The unsupervised term is then estimated by treating the label $\boldsymbol{y}$ as an unobserved latent to marginalize over. This has a caveat however—when given supervised data where $\boldsymbol{y}$ is observed, $\boldsymbol{z}$ is the only latent variable under consideration, and hence an additional classifier term $q_\phi(\boldsymbol{y}\mid\boldsymbol{x})$ is needed:

$$\mathcal{J}_{M2} = \mathbb{E}_{p_\mathcal{S}(\boldsymbol{x},\boldsymbol{y})}\left[\mathcal{L}(\boldsymbol{x},\boldsymbol{y};\theta,\phi) + \alpha\log q_\phi(\boldsymbol{y}\mid\boldsymbol{x})\right] + \mathbb{E}_{p_\mathcal{U}(\boldsymbol{x})}\left[\mathcal{L}(\boldsymbol{x};\theta,\phi)\right]. \tag{2}$$

Subsequent work [23] extended this setting with multiple labels, capturing *arbitrary* dependencies within labels, and between labels and latent variables.

In addition to M2, a simpler *generative classifier* model, called M1, is built in the spirit of dimensionality reduction. This simply learns a VAE over *all* data, and then training a classifier from the learned latent space. They also further construct a combined M1 + M2 model that leverages both.

3 Rethinking Semi–Supervision

The de facto assumption for most approaches to semi-supervision in VAEs is that the labels correspond directly to discrete latent variables $\boldsymbol{z}_c = \boldsymbol{y}$. However, this can cause a number of significant issues if we want the latent space to encapsulate not just the labels themselves, but also their *denotations*—the underlying characteristics that relate a datapoint to its label. For example, encapsulating the masculine characteristics of a face, not just the fact that it is a man's face.

The first issue is that the information represented by a denotation (which is typically continuous) is more than can be stored through a single discrete variable. That is not to say this denotational information is not present in approaches like M2, but here it is *entangled* within the continuous latent variables, $\boldsymbol{z}_{\setminus c}$, which simultaneously contain the denotations for all the labels, rather than having the information *disentangled* to distinct latents associated with each label. Relatedly, it can be difficult to ensure that the VAE actually makes use of $\boldsymbol{z}_c$, rather than just storing all information relevant to reconstruction in the (higher capacity) $\boldsymbol{z}_{\setminus c}$. Overcoming this is challenging and generally requires additional heuristics and hyperparameters, such as the need to tune α in (2).

Second, we may wish to manipulate aspects of the denotation without fully changing the label itself. For example, making somebody look more or less feminine without fully changing their gender. Here we do not know how to manipulate $\boldsymbol{z}_{\setminus c}$ to achieve this desired effect: we can only do the binary operation of changing the relevant variable in $\boldsymbol{z}_c$. Relatedly, we often wish to keep a level of diversity when carrying out conditional generation and, in particular, interventions. For example, if we change somebody's gender then there is no single correct answer for how they would then look, but taking $\boldsymbol{z}_c = \boldsymbol{y}$ only allows for a single point estimate for the change.

Finally, taking the labels to be explicit latent variables can cause a mismatch between the VAE prior $p(\boldsymbol{z})$ and the pushforward distribution of the data to the latent space $q(\boldsymbol{z}) = \mathbb{E}_{p_\mathcal{D}(\boldsymbol{x})}[q_\phi(\boldsymbol{z}\mid\boldsymbol{x})]$. During training, latents are effectively generated according to $q(\boldsymbol{z})$, but once learned, $p(\boldsymbol{z})$ is used to make generations; variations between the two effectively corresponds to a train-test mismatch. As there is a ground truth data distribution over the labels (which are typically not independent), taking $\boldsymbol{z}_c = \boldsymbol{y}$ as the labels themselves implies that there will be a ground truth $q(\boldsymbol{z}_c)$. However, as this is not generally known a priori, we will inevitably end up with a mismatch.

What do we want from semi-supervision? Given these issues, it is natural to ask whether $\boldsymbol{z}_c = \boldsymbol{y}$ is actually necessary? To answer this, we need to think about exactly what it is we are hoping to achieve through the semi-supervision itself. Along with uses of VAEs more generally, the three most common tasks SSVAEs are used for are: **a) Classification**, we try to predict the labels of inputs where these are not known a priori; **b) Conditional Generation**, we generate new examples conditioned on those examples conforming to certain desired labels; and **c) Intervention**, we manipulate certain desired characteristics of a data point before reconstructing it.

Previous work has often implicitly assumed that carrying out these tasks requires the labels to correspond directly to latents. However, on close inspection we see that this is not actually the case. For classification, we need only some classifier going from $\boldsymbol{z}$ to $\boldsymbol{y}$. For conditional generation, we need a mechanism for sampling $\boldsymbol{z}$ given $\boldsymbol{y}$. For interventions, we need to know which latent variables relate to each label and a mechanism for manipulating, or in some cases resampling, them.

4 Reparameterized Variational Autoencoders

To demonstrate how one might apply the insights of the last section, we first introduce a simplistic approach: *jointly* learning a classifier with the VAE. Specifically, reflecting the fact that label information only captures particular aspects of data, we partition the latent space into disjoint subsets: $\boldsymbol{z}_c$, to encapsulate the label denotations, and $\boldsymbol{z}_{\setminus c}$ for the rest. We then construct the objective

$$\mathcal{J}_{M3} = \sum_{\boldsymbol{x}\in\mathcal{U}} \mathcal{L}(\boldsymbol{x}) + \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{S}} \left(\mathcal{L}(\boldsymbol{x},\boldsymbol{y}) + \alpha \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log q_\varphi(\boldsymbol{y} \mid \boldsymbol{z}_c)\right]\right), \tag{3}$$

where α is a hyperparameter that controls the trade-off between the ELBO and the classifier. In this formulation, which we refer to as M3, $q_\varphi(\boldsymbol{y} \mid \boldsymbol{z}_c)$ will implicitly influence the latent embedding; we can interpret $\boldsymbol{z}_c$ as a stochastic layer in the overall classification. Critically, as $\boldsymbol{z}_c$ is also used in the reconstructions, it will also contain the information required for this, including the denotations.

However, in general, the denotations of different class labels will be *entangled* within $\boldsymbol{z}_c$: though it will contain the required information, the latents will typically not be interpretable, and it is unclear how we could perform conditional generation or interventions. To *disentangle* the denotations of different labels, we further partition the latent space, such that the classification of particular labels y^i only has access to particular latents $\boldsymbol{z}_c^i$ and thus $\log q_\varphi(\boldsymbol{y} \mid \boldsymbol{z}_c) = \sum_i \log q_{\varphi^i}(y^i \mid \boldsymbol{z}_c^i)$.

This has the critical effect of forcing the denotational information needed to classify y^i to be stored in the corresponding $\boldsymbol{z}_c^i$, providing a means to encapsulate the characteristic information of each label separately. We further see that it addresses many of the prior issues: there are no measure-theoretic issues as $\boldsymbol{z}_c^i$ is not discrete, diversity in interventions is achieved by sampling different $\boldsymbol{z}_c^i$ for a given label, $\boldsymbol{z}_c^i$ can be manipulated while remaining within class decision boundaries, and a mismatch between $p(\boldsymbol{z}_c)$ and $q(\boldsymbol{z}_c)$ does not manifest as there is no longer ground truth for $q(\boldsymbol{z}_c)$.

However, how to conditionally generate or intervene with M3 is not immediately obvious. Critically, the classifier *implicitly* contains the requisite information to do this via *inference* in an implied Bayesian model. For example, conditional generation needs samples from $p(\boldsymbol{z}_c)$ that classify to the desired labels, e.g. through rejection sampling. See Appendix A for further details.

4.1 The Reparameterized Variational Autoencoder

One way to address the need for inference at test time is to introduce a conditional generative model $p_\psi(\boldsymbol{z}_c \mid \boldsymbol{y})$, simultaneously learnt alongside existing components of M3, along with a prior $p(\boldsymbol{y})$ that is either learnt or held fixed. This approach, which we term the ReVAE, allows the required sampling for conditional generations and interventions directly. Further, by persisting with the latent partitioning above, we can introduce a factorized set of generative models $p(\boldsymbol{z}_c \mid \boldsymbol{y}) = \prod_i p(\boldsymbol{z}_c^i \mid y^i)$, enabling easy generation and manipulation of $\boldsymbol{z}_c^i$ individually. This approach has the advantage of ensuring that the labels remain a part of the model for unlabelled datapoints, which transpires to be an important component for effective learning in practice.

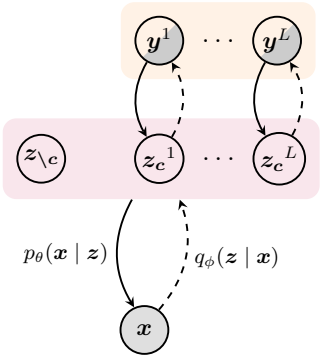

Figure 2: ReVAE graphical model.

To address the issue of learning and semi-supervision, we perform variational inference, treating $\boldsymbol{y}$ as an observation in the supervised case, and an additional variable in the unsupervised case. The final graphical model associated with our variational model is illustrated in Figure 2. The ReVAE can be seen as a way of combining top-down and bottom-up information to obtain a structured latent representation. By enforcing different auxiliary variables to link to a single latent dimension, we are able to align the labelled generative factors with the axes in the latent space.

4.2 Model Objective

We now construct an objective function that encapsulates the model described above, by deriving a lower bound on the full model log-likelihood which factors over the supervised and unsupervised

subsets as discussed in § 2. The supervised objective can be defined as

$$\log p_\theta(\boldsymbol{x}, \boldsymbol{y}) \geq \mathbb{E}_{q_{\varphi,\phi}(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})}\left[\log \frac{p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) p_\psi(\boldsymbol{z} \mid \boldsymbol{y}) p(\boldsymbol{y})}{q_{\varphi,\phi}(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{y})}\right] \equiv \mathcal{L}_{\text{REVAE}}(\boldsymbol{x}, \boldsymbol{y}), \tag{4}$$

with $p_\psi(\boldsymbol{z} \mid \boldsymbol{y}) = p(\boldsymbol{z}_{\setminus \boldsymbol{c}}) p_\psi(\boldsymbol{z}_{\boldsymbol{c}} \mid \boldsymbol{y})$. Here, we avoid directly modelling $q_{\varphi,\phi}(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{y})$; instead leveraging the conditional independence $\boldsymbol{x} \perp\!\!\!\perp \boldsymbol{y} \mid \boldsymbol{z}$, along with Bayes rule, to give

$$q_{\varphi,\phi}(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{y}) = \frac{q_\varphi(\boldsymbol{y} \mid \boldsymbol{z}_c) q_\phi(\boldsymbol{z} \mid \boldsymbol{x})}{q_{\varphi,\phi}(\boldsymbol{y} \mid \boldsymbol{x})}, \quad \text{where} \quad q_{\varphi,\phi}(\boldsymbol{y} \mid \boldsymbol{x}) = \int q_\varphi(\boldsymbol{y} \mid \boldsymbol{z}_c) q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \mathrm{d}\boldsymbol{z}.$$

Using this equivalence in (4) yields (see Appendix B.1 for a derivation and numerical details)

$$\mathcal{L}_{\text{REVAE}}(\boldsymbol{x}, \boldsymbol{y}) = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\frac{q_\varphi(\boldsymbol{y} \mid \boldsymbol{z})}{q_{\varphi,\phi}(\boldsymbol{y} \mid \boldsymbol{x})} \log \frac{p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) p_\psi(\boldsymbol{z} \mid \boldsymbol{y})}{q_\varphi(\boldsymbol{y} \mid \boldsymbol{z}_c) q_\phi(\boldsymbol{z} \mid \boldsymbol{x})}\right] + \log q_{\varphi,\phi}(\boldsymbol{y} \mid \boldsymbol{x}) + \log p(\boldsymbol{y}). \tag{5}$$

Note that unlike M2, M3, and similar models, a classifier term $\log q_{\varphi,\phi}(\boldsymbol{y} \mid \boldsymbol{x})$ falls out naturally from the derivation. Reparametrising labels as auxiliary variables rather than directly as latent variables is crucial for this feature. When defining latents directly to be labels (as in M2), observing both $\boldsymbol{x}$ and $\boldsymbol{y}$ *detaches* the mapping $q_{\varphi,\phi}(\boldsymbol{y} \mid \boldsymbol{x})$ between them, resulting in the parameters (φ, ϕ) not being learned—motivating addition of an explicit (weighted) classifier. Here however, observing both data $\boldsymbol{x}$ and label $\boldsymbol{y}$ does not detach any mapping, since they are always connected via an unobserved random variable $\boldsymbol{z}_c$, and hence do not need additional terms.

The unsupervised part of the objective, $\mathcal{L}_{\text{REVAE}}(\boldsymbol{x})$, derives as the standard (unsupervised) ELBO. However, it requires marginalising over labels as $p(\boldsymbol{z}) = p(\boldsymbol{z}_c) p(\boldsymbol{z}_{\setminus c}) = p(\boldsymbol{z}_{\setminus c}) \sum_{\boldsymbol{y}} p(\boldsymbol{z}_c|\boldsymbol{y}) p(\boldsymbol{y})$. This can be computed exactly, but doing so can be prohibitively expensive if the number of possible label combinations is large. Here, we apply Jensen's inequality a second time to the expectation over $\boldsymbol{y}$ to produce a looser, but cheaper to calculate, ELBO. See Appendix B.2 for details.

Putting this together, we get the complete REVAE objective as

$$\log p_\theta\left(\mathcal{D}\right) \geq \sum\nolimits_{(\boldsymbol{x},\boldsymbol{y}) \in \mathcal{S}} \mathcal{L}_{\text{REVAE}}(\boldsymbol{x}, \boldsymbol{y}) + \sum\nolimits_{\boldsymbol{x} \in \mathcal{U}} \mathcal{L}_{\text{REVAE}}(\boldsymbol{x}), \tag{6}$$

which, unlike some prior approaches, is a valid lower bound on the evidence. It is interesting to note that explicitly modelling the connection between labels and their corresponding latent variables yields such a markedly different objective. As we shall see in (§ 6), this enables a range of capabilities and behaviors that encapsulate the distinction between a label's value and its denotation.

5 Related Work

Beyond the immediate connections to prior work described in § 2, there are a number of other existing approaches that are related to our proposed framework. An auxiliary-variable approach [15] more related to the M2 model augments the encoding distribution with an additional, unobserved latent variable, that enables better semi-supervised classification accuracies. From a pure modeling perspective, there also exists prior work on *hieararchical* VAEs [19, 31] that involve hierarchies of latent variables, exploring richer higher-order inference and issues with redundancy among latent variables in an *unsupervised* setting. Unlike our approach, these auxiliary or hierarchical variables do not have a direct interpretation, but exist merely to improve the flexibility of the encoder. Regarding the disparity between continuous and discrete latent variables in the typical semi-supervised VAEs, [3] provide an approach to enable effective *unsupervised* learning in this setting.

From a more conceptual standpoint, there are two approaches that also incorporate the separation of labels and latent variables. The first of these [18] introduces interventions (called revisions) on VAEs for text data, regressing to auxiliary sentiment scores as a means of influencing the latent variables. This formulation is similar to M3 (3) in spirit, although in practice they employ a range of additional factoring and regularizations particular to their domain of interest, in addition to training models in stages, involving different objective terms. Nonetheless, we share the desire to enforce meaningfulness in the latent representations through auxiliary supervision.

The other approach involves explicitly treating labels as another data *modality* [28, 25, 30, 22]. This work is motivated by the need to learn latent representations that *jointly encode* data from different modalities. Looking back to (4), by refactoring $p(\boldsymbol{z} \mid \boldsymbol{y}) p(\boldsymbol{y})$ as $p(\boldsymbol{y} \mid \boldsymbol{z}) p(\boldsymbol{z})$, and taking $q(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{y}) = \mathcal{G}(q(\boldsymbol{z} \mid \boldsymbol{x}), q(\boldsymbol{z} \mid \boldsymbol{y}))$, one derives *multi-modal* VAEs, where $\mathcal{G}$ can construct a

product [30] or mixture [22] of experts. Of these, the MVAE [30] is more closely related to our setup here, as it explicitly targets cases where alternate data modalities are labels—only ever containing information that is a subset of the information contained in the data. However, they differ in that the latent representations are not structured explicitly to map to distinct classifiers, and do not explore the question of explicitly capturing the denotations of the labels in the latent representations.

6 Experiments

Following our reasoning in § 3 we now showcase the efficacy of REVAE for the three broad aims of *(a) classification*, *(b) conditional generation* and *(c) intervention* on the FashionMNIST and CelebA dataset (restricing ourselves to the 18 labels which are distinguishable in reconstructions for the former, see Appendix C.1 for details). For our encoder and decoder we use MLPs for FashionMNIST and standard architectures [7] for CelebA. The label-predictive distribution $q_\varphi(\boldsymbol{y} \mid \boldsymbol{z}_c)$ is defined as $\text{Cat}(\boldsymbol{y}|\boldsymbol{\pi}_\varphi(\boldsymbol{z_c}))$ with MLP $\boldsymbol{\pi}_\varphi(\cdot)$ for FashionMNIST, and as $\text{Ber}(\boldsymbol{y} \mid \boldsymbol{\pi}_\varphi(\boldsymbol{z_c}))$ with a diagonal transformation $\boldsymbol{\pi}_\varphi(\cdot)$ enforcing $q_\varphi(\boldsymbol{y} \mid \boldsymbol{z}_c) = \prod_i q_{\psi^i}(y_i \mid \boldsymbol{z_c}^i)$ for CelebA. The conditional prior is defined as $p_\psi(\boldsymbol{z}_c \mid \boldsymbol{y}) = \mathcal{N}(\boldsymbol{z_c}|\boldsymbol{\mu}_\psi(\boldsymbol{y}), \text{diag}(\boldsymbol{\sigma}^2_\psi(\boldsymbol{y})))$, with appropriate factorization for CelebA, and has its parameters also derived through MLPs. For FashionMNIST we calculate $p(\boldsymbol{z})$ analytically as $p(\boldsymbol{z}_{\setminus c}) \sum_{\boldsymbol{y}} p(\boldsymbol{z}_c|\boldsymbol{y})p(\boldsymbol{y})$. For CelebA this is not feasible so we employ the additional variational bounding described in Section 4.2. See Appendix C.2 for further details.

Classification We first inspect the predictive ability of REVAE for classification across a range of supervision rates. Table 1 shows the classification accuracies of the label predictive distribution $q_{\varphi,\phi}(\boldsymbol{y} \mid \boldsymbol{x})$ learned by M2, M3 and REVAE. It can be observed that REVAE generally obtains prediction accuracies equivalent or slightly superior to M2 (except in the case of very low supervision on FashionMNIST) and much better than M3. We emphasize here that REVAE is not setup up to achieve better classification accuracies; we are simply checking that it does not harm them.

Table 1: Classification accuracies. Boldface denotes the best performing model and models whose performance is not statistically significantly different to it according to a non-parametric Mann–Whitney U test.

	FashionMNIST			CelebA		
Model	$f = 0.004$	$f = 0.06$	$f = 0.2$	$f = 0.004$	$f = 0.06$	$f = 0.2$
REVAE	0.68±0.02	**0.82±0.01**	**0.85±0.01**	**0.80±0.00**	**0.88±0.00**	**0.88±0.00**
M2	**0.73±0.03**	**0.83±0.00**	**0.85±0.00**	0.79±0.01	0.86±0.00	0.87±0.00
M3	0.22±0.07	0.34±0.14	0.31±0.06	0.52±0.02	0.64±0.02	0.73±0.00

Conditional Generation To asses conditional generation, we first train an independent classifier for both datasets. We then conditionally generate samples given labels and evaluate them using this pre-trained classifier. To further give an indication of how "real" the generations are, i.e. whether they are overall *out-of-distribution*, we also measure the mutual information (MI) between the parameters of the classifier and the labels as per [24]. This provides a measure the *epistemic* uncertainty of the classifier, with lower MI values being preferable. See Appendix C.3 for more details. Results are shown in Table 2. We can see that in all cases for FashionMNIST REVAE outperforms M2, indicating that having a diverse set of generations actually improves performance. For CelebA, REVAE also outperforms M2 for the accuracy of the generations. To measure visual fidelity, rather than MI we instead report the FID [6], both M2 and REVAE perform comparably, unlike M3 which is significantly worse, possible due to more weight being applied to the classifier than the VAE.

Single Interventions To assess the fidelity of interventions, we first consider intervening on a single label. For REVAE, this corresponds to sampling from $p_{\psi^i}(\boldsymbol{z}^i_c \mid \boldsymbol{y}_i)$ in the dimension of the

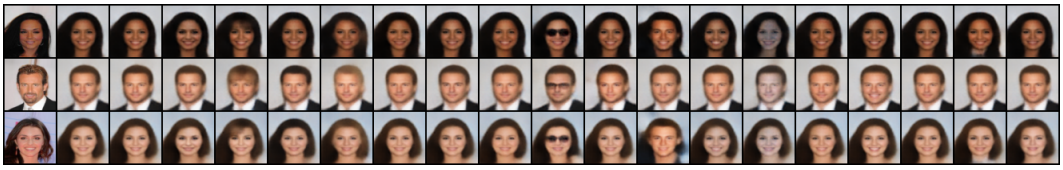

Figure 3: From left to right: original, reconstruction, then interventions from switching on the following labels: `arched eyebrows`, `bags under eyes`, `bangs`, `black hair`, `blond hair`, `brown hair`, `bushy eyebrows`, `chubby`, `eyeglasses`, `heavy makeup`, `male`, `no beard`, `pale skin`, `receding hairline`, `smiling`, `wavy hair`, `wearing necktie`, `young`.

Table 2: Pre-trained clasifier accuracies and MI for FashionMNIST (top), and pre-trained clasifier accuracies and FID for CelebA (bottom). Boldface denotes the best performing model and models whose performance is not statistically significantly different to it according to a non-parametric Mann–Whitney U test.

	Model	$f = 0.004$		$f = 0.06$		$f = 0.2$	
		Acc	MI	Acc	MI	Acc	MI
FMNIST	REVAE	**0.61±0.02**	**0.03±0.00**	**0.72±0.02**	**0.03±0.00**	**0.76±0.00**	**0.02±0.00**
	M2	0.55±0.01	0.06±0.00	0.68±0.01	0.05±0.00	0.69± 0.00	0.05±0.00
	M3	0.19±0.06	0.04±0.01	0.27±0.11	0.04±0.01	0.21±0.04	0.08±0.02
		Acc	FID	Acc	FID	Acc	FID
CelebA	REVAE	**0.49±0.00**	**130.0±2.10**	**0.55±0.04**	**122.6±1.10**	**0.59±0.00**	**121.0±1.60**
	M2	0.43±0.00	**129.2±1.79**	**0.53±0.00**	**121.4±1.69**	0.53±0.00	**118.4±2.00**
	M3	**0.49±0.01**	190.0±13.5	0.49±0.01	159.3±13.2	0.49±0.01	301.3±68.7

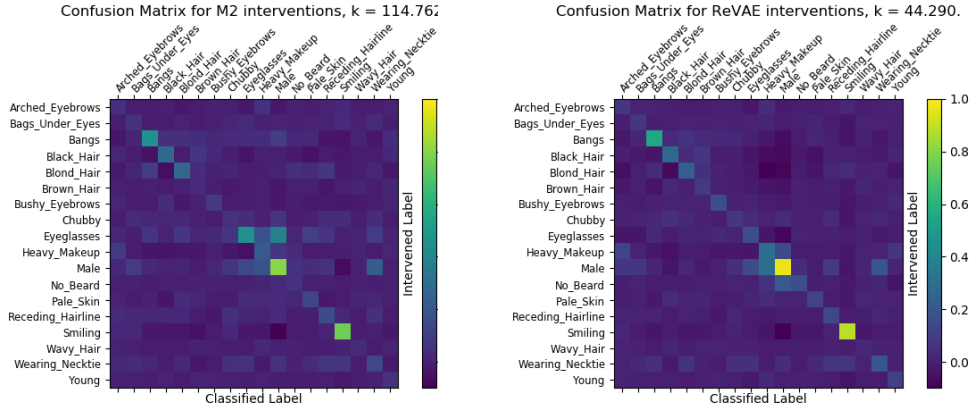

Figure 4: Intervention confusion matrices for M2 (left) and REVAE (right). Here we intervene on the label of a column an report the probability change for the class given by the row. Condition number is given in the title.

class we want to intervene on. We demonstrate the qualitative results for REVAE in Figure 3, which shows only a single attribute changing in each column. Equivalent plots for M2 and M3 are given in the Appendix. We further quantitatively assess these intervention by constructing a confusion matrix on how interventions affect the prediction probability of *each* class, the result of which are given in Figure 4. Here, perfect interventions should produce an identity matrix and we see that REVAE outperforms M2 as reflected in its lower matrix condition number.

Full Interventions We next consider making wholesale interventions through "denotation swaps." Namely, we encode an image, replace its $\boldsymbol{z}_c$ with the $\boldsymbol{z}_c$ of another image, and then reconstruct the image while holding $\boldsymbol{z}_{\setminus c}$ constant. For REVAE, this has the effect of applying the denotations of the second image to the first. By comparison, M2 only allows for the labels to be transferred and not the more subtle denotation information. The results, shown in Figure 5, reveal that this is indeed the case. The differences are perhaps most obvious in how for REVAE, but not M2, the bangs are preserved in the fifth row and the grinning nature of the smile is preserved in the third row. The figure further provides a quantitative measure for these interventions: we report the difference in log probabilities of the classes when evaluated on a pre-trained classifier of the reconstruction, averaging these values over a large number of such interventions. Here we see REVAE outperforms M2 on *every class*, providing a quantitative confirmation of its superior intervention performance.

Intervention Diversity As outlined in Section 3, there are typically multiple different interventions that are consistent with manipulating a label. For example, if we add glasses to a person, there are multiple styles of glasses we might add. We demonstrate REVAEs' ability to encapsulate this *diversity* in Figure 6 where we draw multiple sample interventions $\boldsymbol{z}_c^i \sim p_{\psi^i}(y_i \mid \boldsymbol{z}_c^i)$ and look at the different reconstructions it produces. We see that REVAE produces a diverse set of label specific realizations. By comparison, such a plot cannot be generated at all for M2 as it only allows single point estimate interventions.

Latent Walk Interventions By manipulating a particular $\boldsymbol{z}_c^i$, REVAE is able to smoothly manipulate the characteristics of an image relating to a given label, without severely affecting others. This

Figure 5: [Left, Middle] "Denotation swaps" for M2 and REVAE respectively. Here, the denotations of the leftmost column are applied to the top image, such that rows should contain the same characteristics. [Right] quantitative results produced by averaging the differences in log probabilities of each class from performing 20 randomly chosen such interventions for each datapoint over the whole dataset (lower is better).

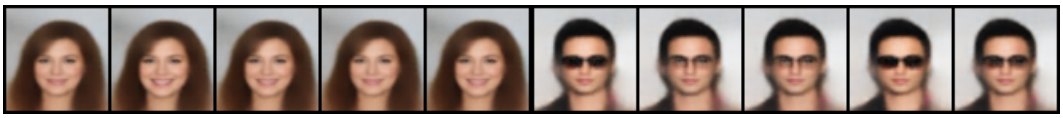

Figure 6: Diverse interventions on `smiling` and `eyeglasses` for CelebA.

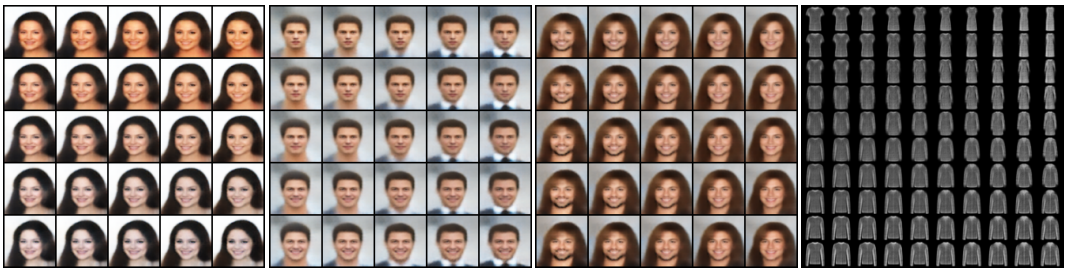

Figure 7: Continuous interventions through traversal of $\boldsymbol{z}_c$. From left to second from the right: CelebA latent walks between `pale skin` and `young`; `smiling` and `necktie`; `male` and `beard`. Far right: interpolation on surface in the latent space between the four FashionMNIST classes: `t-shirt`, `pullover`, `coat` and `dress`.

allows for a fine control during intervention, unlike M2 which can only make binary changes. To demonstrate this, we traverse two dimensions of the latent space and display the reconstructions in Figure 7. These examples indicate that REVAE is able to smoothly manipulate characteristics through its encapsulation and disentangling of denotations; no such traversals are possible for M2. For example, in the leftmost set of images we are able to induce varying skin tones rather than have this be a binary intervention on `pale skin`. In the second set, we find that the $\boldsymbol{z}_c^i$ associated with the `necktie` label has also managed to encapsulate information about whether someone is wearing a shirt or is completely bare-necked. In the third, we are able to separately encapsulate the length of beard and gender as continuous variables that have separate impacts on the image. Finally, in the last set, we see that we are able to smoothly interpolate between classes in FashionMNIST, e.g. going from a `t-shirt` to a dress involves a steady elongation of the torso, as one would expect.

7 Discussion

We have presented a novel mechanism for performing semi-supervised learning in deep generative models, the *reparameterized* VAE (REVAE), wherein we avoid a direct correspondence between labels and latents, and instead treat the labels as auxiliary variables. This has allowed us to encapsulate and disentangle the *denotations* associated with labels, rather than just the label values. We are able to do so without affecting the ability to perform the tasks one typically does in the semi-supervised setting—namely classification, conditional generation, and intervention. In particular, we have shown that, not only does this lead to more effective conventional label-switch interventions, it also allows for more fine-grained interventions to be performed, such as producing diverse sets of samples consistent with an intervened label value, or performing continuous traversals of the denotation space both within and across class labels.

Broader Impact

Manipulating generative factors of data comes with obvious advantage such as the ability to manipulate certain characteristics without affecting others, e.g. seeing what someone will look like with a different hair color, or when wearing glasses. However, the ability to do so on such personal and potentially sensitive features leads to serious thought into the ethical considerations about how such approaches should be used. Moreover with the ever pressing issue of Deep-Fakes[12] undermining the confidence of images representing a true scene, this work has the potential to stoke this problem even further. While this concern is very real, it is purely application based and limited to photographs of people—a domain which ReVAE certainly is not exclusively tied to. Moreover, these issues are common to all of semi–supervised learning in deep generative models, rather than being specific to our work. The conceptual and methodological ideas presented in the paper draw attention to how representations of denotations should be stored in latent variable models, which is potentially very useful in certain domains. Therefore, though the aforementioned concerns should not be forgotten, we do not feel like they are a basis to avoid pursuing such avenues of research.

Acknowledgments and Disclosure of Funding

TJ, PHST, and NS were supported by the ERC grant ERC-2012-AdG 321162-HELIOS, EPSRC grant Seebibyte EP/M013774/1 and EPSRC/MURI grant EP/N019474/1. Toshiba Research Europe also support TJ. PHST would also like to acknowledge the Royal Academy of Engineering and FiveAI. SMS was partially supported by the Engineering and Physical Sciences Research Council (EPSRC) grant EP/K503113/1. TR's research leading to these results has received funding from a Christ Church Oxford Junior Research Fellowship and from Tencent AI Labs.

References

[1] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, August 2013.

[2] Rodney A Brooks. Intelligence without representation. *Artificial intelligence*, 47(1-3):139–159, 1991.

[3] Emilien Dupont. Learning disentangled joint continuous and discrete representations. In *Advances in Neural Information Processing Systems*, pages 710–720, 2018.

[4] Yarin Gal. *Uncertainty in deep learning*. PhD thesis, University of Cambridge, 2016. Unpublished doctoral dissertation.

[5] Peter Gärdenfors. *Conceptual spaces: The geometry of thought*. MIT press, 2004.

[6] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626–6637, 2017.

[7] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *Proceedings of the International Conference on Learning Representations*, 2016.

[8] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[9] Geoffrey E Hinton and Richard S Zemel. Autoencoders, minimum description length and helmholtz free energy. In *Advances in neural information processing systems*, pages 3–10, 1994.

[10] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[11] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pages 3581–3589, 2014.

[12] Pavel Korshunov and Sébastien Marcel. Deepfakes: a new threat to face recognition? assessment and detection. *arXiv preprint arXiv:1812.08685*, 2018.

[13] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[14] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, pages 4114–4124, 2019.

[15] Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. Auxiliary deep generative models. *arXiv preprint arXiv:1602.05473*, 2016.

[16] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B Tenenbaum, and Jiajun Wu. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. *arXiv preprint arXiv:1904.12584*, 2019.

[17] Emile Mathieu, Tom Rainforth, N Siddharth, and Yee Whye Teh. Disentangling disentanglement in variational autoencoders. In *International Conference on Machine Learning*, pages 4402–4412, 2019.

[18] Jonas Mueller, David Gifford, and Tommi Jaakkola. Sequence to better sequence: continuous revision of combinatorial structures. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2536–2544. JMLR. org, 2017.

[19] Rajesh Ranganath, Dustin Tran, and David Blei. Hierarchical variational models. In *International Conference on Machine Learning*, pages 324–333, 2016.

[20] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286, 2014.

[21] Eleanor H Rosch. Natural categories. *Cognitive psychology*, 4(3):328–350, 1973.

[22] Yuge Shi*, N. Siddharth*, Brooks Paige, and Philip H. S. Torr. Variational mixture-of-experts autoencoders for multi-modal deep generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 15692–15703, December 2019.

[23] N. Siddharth, T Brooks Paige, Jan-Willem Van de Meent, Alban Desmaison, Noah Goodman, Pushmeet Kohli, Frank Wood, and Philip Torr. Learning disentangled representations with semi-supervised deep generative models. In *Advances in Neural Information Processing Systems*, pages 5925–5935, 2017.

[24] Lewis Smith and Yarin Gal. Understanding measures of uncertainty for adversarial example detection. *arXiv preprint arXiv:1803.08533*, 2018.

[25] Masahiro Suzuki, Kotaro Nakayama, and Yutaka Matsuo. Joint multimodal learning with deep generative models. In *International Conference on Learning Representations Workshop*, 2017.

[26] Joshua B Tenenbaum. Mapping a manifold of perceptual observations. In *Advances in neural information processing systems*, pages 682–688, 1998.

[27] Joshua B Tenenbaum and William T Freeman. Separating style and content with bilinear models. *Neural computation*, 12(6):1247–1283, 2000.

[28] Ramakrishna Vedantam, Ian Fischer, Jonathan Huang, and Kevin Murphy. Generative models of visually grounded imagination. In *Proceedings of the International Conference on Learning Representations*, 2018.

[29] Martin J Wainwright and Michael I Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.

[30] Mike Wu and Noah Goodman. Multimodal generative models for scalable weakly-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5580–5590, 2018.

[31] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Learning hierarchical features from deep generative models. In *International Conference on Machine Learning*, pages 4091–4099, 2017.

A Conditional Generation and Intervention for M3

For the M3 model to be usable, we must consider whether it can carry out the classification, conditional generation, and intervention tasks outlined in the last section. Of these, classification is obviously trivial, but it is less immediately apparent how the others could be performed. The key here is to realize that the classifier itself *implicitly* contains the information required to perform these tasks.

Consider first conditional generation and note that we still have access to the prior $p(\boldsymbol{z})$ as per a standard VAE. One simple way of performing conditional generation would be to conduct a rejection sampling where we draw samples $\hat{\boldsymbol{z}} \sim p(\boldsymbol{z})$ and then accept these if and only if they lead to the classifier predicting the desired labels up to a desired level of confidence, i.e. $q_\phi(\boldsymbol{y} \mid \hat{\boldsymbol{z}}_c) > \lambda$ where $0 < \lambda < 1$ is some chosen confidence threshold. Though such an approach is likely to be highly inefficient for any general $p(\boldsymbol{z})$ due to the curse of dimensionality, in the standard setting where each dimension of $\boldsymbol{z}$ is independent, this rejection sampling can be performed separately for each $\boldsymbol{z}_c^i$, for which it which actually often be relatively painless. More generally, we have that conditional generation becomes an inference problem where we wish to draw samples from

$$p\left(\boldsymbol{z} \mid \{q_\phi(\boldsymbol{y} \mid \boldsymbol{z}_c) > \lambda\}\right) \propto p(\boldsymbol{z}) \mathbb{I}\left(q_\phi(\boldsymbol{y} \mid \boldsymbol{z}_c) > \lambda\right).$$

Interventions can also be performed in an analogous manner. Namely, for a conventional intervention where we change one or more labels, we can simply resample the $\boldsymbol{z}_c^i$ associated we those labels, thereby sampling new denotations to match the new labels. Further, unlike in previous approaches, there are alternative interventions we can perform as well. For example, we might attempt to find the closest $\boldsymbol{z}_c^i$ to the original that leads to the class label changing; this can be done in a manner akin to how adversarial attacks are performed. Alternatively, we might look to manipulate the $\boldsymbol{z}_c^i$ without actually changing the class itself to see what other denotations are consistent with the labels.

To summarize, this M4 model provides a mechanism of learning semi-supervised VAEs that avoid the pitfalls of directly fixing the latents to correspond to labels. It still allows us to perform all the tasks usually associated with semi-supervised VAEs and in fact allows a more general form of interventions to be performed. However, this comes at the cost of requiring inference to be performed to perform conditional generation or interventions. Further, as the auxiliary variables $\boldsymbol{y}$ are not present when the labels are unobserved, there may be empirical complications with forcing all the denotational information to be encoded to the appropriate $\boldsymbol{z}_c^i$. In particular, we still have a hyperparameter α that must be carefully tuned to ensure the appropriate balance between classification and reconstruction.

B Model Formulation

B.1 Variational Lower Bound

In this section we provide the mathematical details of our objective functions. We show how to derive it as a lower bound to the marginal model likelihood and show how we estimate the model components.

The variational lower bound for the generative model in Figure 2, is given as

$$\begin{aligned}
\mathcal{L}_{\text{ReVAE}} &= \sum_{\boldsymbol{x} \in \mathcal{U}} \mathcal{L}_{\text{ReVAE}}(\boldsymbol{x}) + \sum_{(\boldsymbol{x},\boldsymbol{y}) \in \mathcal{S}} \mathcal{L}_{\text{ReVAE}}(\boldsymbol{x}, \boldsymbol{y}) \\
\mathcal{L}_{\text{ReVAE}}(\boldsymbol{x}, \boldsymbol{y}) &= E_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\frac{q_\varphi(\boldsymbol{y} \mid \boldsymbol{z}_c)}{q_{\varphi,\phi}(\boldsymbol{y} \mid \boldsymbol{x})} \log\left(\frac{p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) p_\psi(\boldsymbol{z} \mid \boldsymbol{y})}{q_\varphi(\boldsymbol{y} \mid \boldsymbol{z}_c) q_\phi(\boldsymbol{z} \mid \boldsymbol{x})}\right)\right] + \log q_{\varphi,\phi}(\boldsymbol{y} \mid \boldsymbol{x}) + \log p(\boldsymbol{y}), \\
\mathcal{L}_{\text{ReVAE}}(\boldsymbol{x}) &= E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}) q_\varphi(\boldsymbol{y}|\boldsymbol{z}_c)}\left[\log\left(\frac{p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) p_\psi(\boldsymbol{z}_c \mid \boldsymbol{y}) p(\boldsymbol{y})}{q_\varphi(\boldsymbol{y} \mid \boldsymbol{z}_c) q_\phi(\boldsymbol{z} \mid \boldsymbol{x})}\right)\right].
\end{aligned}$$

The overall likelihood in the semi-supervised case is given as

$$p_\theta(\boldsymbol{x}, \boldsymbol{y}) = \prod_{(\boldsymbol{x},\boldsymbol{y}) \in \mathcal{S}} p_\theta(\boldsymbol{x}, \boldsymbol{y}) \prod_{\boldsymbol{x} \in \mathcal{U}} p_\theta(\boldsymbol{x}),$$

To derive a lower bound for the overall objective, we need to obtain lower bounds on $\log p_\theta(\boldsymbol{x})$ and $\log p_\theta(\boldsymbol{x}, \boldsymbol{y})$. When the labels are unobserved the latent state will consist of $\boldsymbol{z}$ and $\boldsymbol{y}$. Using the

factorization according to the graph in Figure 2 yields

$$\log p_\theta(\boldsymbol{x}) \geq E_{q_\phi(\boldsymbol{z}|\boldsymbol{x})q_\varphi(\boldsymbol{y}|\boldsymbol{z}_c)}\left[\log\left(\frac{p_\theta(\boldsymbol{x} \mid \boldsymbol{z})p_\psi(\boldsymbol{z} \mid \boldsymbol{y})p(\boldsymbol{y})}{q_\varphi(\boldsymbol{y} \mid \boldsymbol{z}_c)q_\phi(\boldsymbol{z} \mid \boldsymbol{x})}\right)\right],$$

where $p_\psi(\boldsymbol{z} \mid \boldsymbol{y}) = p(\boldsymbol{z}_{\setminus \boldsymbol{c}})p_\psi(\boldsymbol{z}_{\boldsymbol{c}} \mid \boldsymbol{y})$. For supervised data points we consider a lower bound on the likelihood $p_\theta(\boldsymbol{x}, \boldsymbol{y})$,

$$\log p_\theta(\boldsymbol{x}, \boldsymbol{y}) \geq \int \log \frac{p_\theta(\boldsymbol{x} \mid \boldsymbol{z})p_\psi(\boldsymbol{z} \mid \boldsymbol{y})p(\boldsymbol{y})}{q_{\varphi,\phi}(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{y})} q_{\varphi,\phi}(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{y})\mathrm{d}\boldsymbol{z},$$

in order to make sense of the term $q_{\varphi,\phi}(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{y})$, which is usually different from $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ we consider the inference model

$$q_{\varphi,\phi}(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{y}) = \frac{q_\varphi(\boldsymbol{y} \mid \boldsymbol{z}_c)q_\phi(\boldsymbol{z} \mid \boldsymbol{x})}{q_{\varphi,\phi}(\boldsymbol{y} \mid \boldsymbol{x})}, \quad \text{where} \quad q_{\varphi,\phi}(\boldsymbol{y} \mid \boldsymbol{x}) = \int q_\varphi(\boldsymbol{y} \mid \boldsymbol{z}_c)q_\phi(\boldsymbol{z} \mid \boldsymbol{x})\mathrm{d}\boldsymbol{z}.$$

Returning to the lower bound on $\log p_\theta(\boldsymbol{x}, \boldsymbol{y})$ we obtain

$$\begin{aligned}
\log p_\theta(\boldsymbol{x}, \boldsymbol{y}) &\geq \int \log \frac{p_\theta(\boldsymbol{x} \mid \boldsymbol{z})p_\psi(\boldsymbol{z} \mid \boldsymbol{y})p(\boldsymbol{y})}{q(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{y})} q(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{y})\mathrm{d}\boldsymbol{z} \\
&= \int \log\left(\frac{p_\theta(\boldsymbol{x} \mid \boldsymbol{z})p_\psi(\boldsymbol{z} \mid \boldsymbol{y})p(\boldsymbol{y})q_{\varphi,\phi}(\boldsymbol{y} \mid \boldsymbol{x})}{q_\varphi(\boldsymbol{y} \mid \boldsymbol{z}_c)q_\phi(\boldsymbol{z} \mid \boldsymbol{x})}\right) \frac{q_\varphi(\boldsymbol{y} \mid \boldsymbol{z}_c)q_\phi(\boldsymbol{z} \mid \boldsymbol{x})}{q_{\varphi,\phi}(\boldsymbol{y} \mid \boldsymbol{x})}\mathrm{d}\boldsymbol{z} \\
&= E_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\frac{q_\varphi(\boldsymbol{y} \mid \boldsymbol{z}_c)}{q_{\varphi,\phi}(\boldsymbol{y} \mid \boldsymbol{x})} \log\left(\frac{p(\boldsymbol{x} \mid \boldsymbol{z})p_\psi(\boldsymbol{z}_c \mid \boldsymbol{y})}{q_\varphi(\boldsymbol{y} \mid \boldsymbol{z}_c)q_\phi(\boldsymbol{z} \mid \boldsymbol{x})}\right)\right] + \log q_{\varphi,\phi}(\boldsymbol{y} \mid \boldsymbol{x}) + \log p(\boldsymbol{y})
\end{aligned}$$

where $q_\varphi(\boldsymbol{y} \mid \boldsymbol{z}_c)/q_{\varphi,\phi}(\boldsymbol{y} \mid \boldsymbol{x})$ denotes the Radon-Nikodym derivative of $q_{\varphi,\phi}(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{y})$ with respect to $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$.

B.2 Alternative Derivation of Unsupervised Bound

The bound for the unsupervised case can alternatively be derived by applying Jensen's inequality twice. First, use the standard (unsupervised) ELBO

$$\log p_\theta(\boldsymbol{x}) \geq \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p_\theta(\boldsymbol{x} \mid \boldsymbol{z})p(\boldsymbol{z})}{q_\phi(\boldsymbol{z} \mid \boldsymbol{x})}\right].$$

Now, since calculating $p(\boldsymbol{z}) = p(\boldsymbol{z}_c)p(\boldsymbol{z}_{\setminus c}) = p(\boldsymbol{z}_{\setminus c})\sum_{\boldsymbol{y}} p(\boldsymbol{z}_c \mid \boldsymbol{y})p(\boldsymbol{y})$ can be expensive we can apply Jensen's inequality a second time to the expectation over $\boldsymbol{z}_c$ to obtain

$$\log p(\boldsymbol{z}_c) \geq \mathbb{E}_{q_\varphi(\boldsymbol{y}|\boldsymbol{z}_c)}\left[\log \frac{p_\psi(\boldsymbol{z}_s \mid \boldsymbol{y})p(\boldsymbol{y})}{q_\varphi(\boldsymbol{y} \mid \boldsymbol{z}_s)}\right].$$

Substituting this bound into the unsupervised ELBO yields again our bound

$$\log p(\boldsymbol{x}) \geq \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})q_\varphi(\boldsymbol{y}|\boldsymbol{z}_c)}\left[\log \frac{p_\theta(\boldsymbol{x} \mid \boldsymbol{z})p(\boldsymbol{z} \mid \boldsymbol{y})}{q_\phi(\boldsymbol{z} \mid \boldsymbol{x})q_\varphi(\boldsymbol{y} \mid \boldsymbol{z}_c)}\right] + \log p(\boldsymbol{y}) \tag{7}$$

C Implementation

C.1 CelebA

We chose to use only a subset of the labels present in CelebA. The reason for this is two-fold: firstly, not all attributes are visually distinguishable in the reconstructions e.g. (`earrings`); secondly, some of the attributes are potentially offensive e.g (`attractive`). As such we limited ourselves to the following labels: `arched eyebrows`, `bags under eyes`, `bangs`, `black hair`, `blond hair`, `brown hair`, `bushy eyebrows`, `chubby`, `eyeglasses`, `heavy makeup`, `male`, `no beard`, `pale skin`, `receding hairline`, `smiling`, `wavy hair`, `wearing necktie`, `young`. No images were omitted or cropped, the only modifications were keeping the aforementioned labels and resizing the images to be 64×64 in dimension.

C.2 Implementation Details

For our experiments we define the generative and inference networks as follows. The approximate posterior is represented as $q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) = \mathcal{N}(\boldsymbol{z_c}, \boldsymbol{z}_{\setminus \boldsymbol{c}} \mid \boldsymbol{\mu}_\phi(\boldsymbol{x}), \text{diag}(\boldsymbol{\sigma}^2_\phi(\boldsymbol{x})))$ with $\boldsymbol{\mu}_\phi(\boldsymbol{x})$ and $\text{diag}(\boldsymbol{\sigma}^2_\phi(\boldsymbol{x}))$ being an MLP for FashionMNIST and the architecture from [7] for CelebA. The generative model $p_\theta(\boldsymbol{x} \mid \boldsymbol{z})$ is represented by a Bernoulli distribution and also parametrised by an MLP for FashionMNIST and a Laplace distribution, again parametrised using the architecture from [7] for CelebA. The label predictive distribution $q_\varphi(\boldsymbol{y} \mid \boldsymbol{z}_c)$ is represented as $\text{Cat}(\boldsymbol{y} \mid \boldsymbol{\pi}_\varphi(\boldsymbol{z_c}))$ with $\boldsymbol{\pi}_\varphi(\boldsymbol{z_c})$ being an MLP for FashionMNIST, or as $\text{Ber}(\boldsymbol{y} \mid \boldsymbol{\pi}_\varphi(\boldsymbol{z_c}))$ with $\boldsymbol{\pi}_\varphi(\boldsymbol{z_c})$ being a diagonal transformation forcing the factorisation $q_\varphi(\boldsymbol{y} \mid \boldsymbol{z}_c) = \prod_i q_{\psi^i}(y_i \mid \boldsymbol{z_c}^i)$ for CelebA. The conditional prior is given as $p_\psi(\boldsymbol{z}_c \mid \boldsymbol{y}) = \mathcal{N}(\boldsymbol{z_c} \mid \boldsymbol{\mu}_\psi(\boldsymbol{y}), \text{diag}(\boldsymbol{\sigma}^2_\psi(\boldsymbol{y})))$ (with the appropriate factorisation for CelebA) where the parameters are represented by an MLP. Finally, the prior placed on the portion of the latent space reserved for unlabelled latent variables is $p(\boldsymbol{z}_{\setminus \boldsymbol{c}}) = \mathcal{N}(\boldsymbol{z}_{\setminus \boldsymbol{c}} \mid \mathbf{0}, \mathbf{I}))$. For the latent space $\boldsymbol{z_c} \in \mathbb{R}^{m_c}$ and $\boldsymbol{z}_{\setminus \boldsymbol{c}} \in \mathbb{R}^{m_{\setminus c}}$, where $m = m_c + m_{\setminus c}$ with $m_c = 4$ and $m_{\setminus c} = 10$ for FashionMNIST and $m_c = 18$ and $m_{\setminus c} = 27$ for CelebA. The architectures are given in Table 3 and Table 4.

Encoder	**Decoder**
Input 1 x 28 x 28 image	Input $\in \mathbb{R}^m$
784×600 Linear layer & ReLU	600×600 Linear layer & ReLU
600×600 Linear layer & ReLU	600×600 Linear layer & ReLU
$600 \times (2{\times}m)$ Linear layer	600×784 Linear layer & Sigmoid

Classifier	**Conditional Prior**
Input $\in \mathbb{R}^{m_c}$	Input $\in \mathbb{R}^{10}$
50×50 Linear layer & ReLU	10×50 Linear layer & ReLU
50×10 Linear layer	$50 \times (2 \times m_c)$ Linear layer

Table 3: Architectures for FashionMNIST and MNIST.

Encoder	**Decoder**
Input 32 x 32 x 3 channel image	Input $\in \mathbb{R}^m$
$32 \times 3 \times 4 \times 4$ Conv2d stride 2 & ReLU	$(2{\times}m) \times 256$ Linear layer
$32 \times 32 \times 4 \times 4$ Conv2d stride 2 & ReLU	$128 \times 256 \times 4 \times 4$ ConvTranspose2d stride 1 & ReLU
$64 \times 32 \times 4 \times 4$ Conv2d stride 2 & ReLU	$64 \times 128 \times 4 \times 4$ ConvTranspose2d stride 2 & ReLU
$128 \times 64 \times 4 \times 4$ Conv2d stride 2 & ReLU	$32 \times 64 \times 4 \times 4$ ConvTranspose2d stride 2 & ReLU
$256 \times 128 \times 4 \times 4$ Conv2d stride 1 & ReLU	$32 \times 32 \times 4 \times 4$ ConvTranspose2d stride 2 & ReLU
$256 \times (2{\times}m)$ Linear layer	$3 \times 32 \times 4 \times 4$ ConvTranspose2d stride 2 & Sigmoid

(a) Fashion-MNIST dataset.

Classifier	**Conditional Prior**
Input $\in \mathbb{R}^{m_c}$	Input $\in \mathbb{R}^{m_c}$
$m_c \times m_c$ Diagonal layer	$m_c \times m_c$ Diagonal layer

Table 4: Architectures for CelebA.

Optimization To perform the optimization, we trained the models on a GeForce GTX Titan GPU. Training consumed $\sim$ 1Gb memory for FashionMNIST and $\sim$ 2Gb for CelebA, taking around 20 minutes and 4 hours to complete 100 epochs respectively. Both models were optimized using Adam with a learning rate of 5×10^{-4} and 2×10^{-4} for FashionMNIST and CelebA respectively.

C.3 Classifier Uncertainty and Mutual Information

We use classifier uncertainty as an *out-of-distribution* measure on generated or intervened data. In order to estimate the uncertainty, we transform a fixed pre-trained classifier into a Bayesian predictive classifier that integrates over the posterior distribution of parameters ω as $p(\boldsymbol{y} \mid \boldsymbol{x}, \mathcal{D}) = \int p(\boldsymbol{y} \mid \boldsymbol{x}, \omega) p(\omega \mid \mathcal{D}) \mathrm{d}\omega$. The utility of classifier uncertainties for out-of-distribution detection has previously been explored [24], where dropout is also used at test time to estimate the mutual information (MI) between the predicted label $\boldsymbol{y}$ and parameters ω [4, 24] as.

$$I(\boldsymbol{y}, \omega \mid \boldsymbol{x}, \mathcal{D}) = H[p(\boldsymbol{y} \mid \boldsymbol{x}, \mathcal{D})] - \mathbb{E}_{p(\omega \mid \mathcal{D})} \left[H[p(\boldsymbol{y} \mid \boldsymbol{x}, \omega)] \right].$$

However, the Monte-Carlo (MC) dropout approach has the disadvantage of requiring *ensembling* over multiple instances of the classifier for a robust estimate and repeated forward passes through the classifier to estimate MI. To mitigate this, we instead employ a sparse variational GP (with 200

inducing points) as a replacement for the last linear layer of the classifier, fitting just the GP to the data and labels while holding the rest of the classifier fixed. This, in our experience, provides a more robust and cheaper alternative to MC-dropout for estimating MI.

D Additional Results

D.1 Classification and Generation

For the case where classification and generation is the primary goal, we can improve the resulting accuracies by setting $\boldsymbol{z}_{\setminus \boldsymbol{c}} = \emptyset$, that is, forcing the classifier to use the whole of the latent space $\boldsymbol{z_c} = \boldsymbol{z}$. The results for classification and generation are given in Table 5 and Table 6, Subscript indicates the size of m.

The results indicate that when the primary goal is purely to perform classification or conditional generation, by not splitting the latent space, REVAE is able to obtain superior results, particularly for the case of conditional generation. We posit that this is due to M2 having to sample from the continuous and the discrete latent space, which as we showed in the main paper, entangles class specific denotations. As such, situations could arise where $\boldsymbol{z}$ and $\boldsymbol{y}$ potentially provide conflicting information. Conversely, REVAE learns to structure the latent space such that certain regions correspond to certain classes which implicitly contain the style information.

Table 5: Additional classification accuracies.

	FashionMNIST			MNIST		
Model	$f = 0.004$	$f = 0.06$	$f = 0.2$	$f = 0.004$	$f = 0.06$	$f = 0.2$
REVAE$_{20}$	**0.75±0.02**	**0.83±0.01**	**0.84±0.01**	**0.93±0.01**	**0.97±0.00**	**0.97±0.02**
REVAE$_{10}$	**0.75±0.01**	**0.83±0.01**	**0.86±0.00**	**0.93±0.00**	**0.96±0.00**	**0.97±0.00**
M2	**0.73±0.03**	**0.83±0.00**	**0.85±0.00**	0.90±0.01	0.94±0.00	0.96±0.00

Table 6: Pre-trained classifier accuracies and MI for FashionMNIST (top) and MNIST (bottom).

	Model	$f = 0.004$		$f = 0.06$		$f = 0.2$	
		Acc	MI	Acc	MI	Acc	MI
FMNIST	REVAE$_{20}$	**0.70±0.02**	**0.03±0.00**	**0.76±0.02**	**0.03±0.00**	**0.78±0.02**	**0.03±0.00**
	REVAE$_{10}$	**0.71±0.02**	**0.03±0.00**	**0.76±0.04**	**0.03±0.00**	**0.80±0.01**	**0.02±0.00**
	M2	0.55±0.01	0.06±0.00	0.68±0.02	0.05±0.00	0.69± 0.01	0.05±0.00
MNIST	REVAE$_{20}$	**0.90±0.01**	**0.02±0.00**	**0.94±0.00**	0.02±0.00	0.94±0.03	**0.02±0.00**
	REVAE$_{10}$	**0.91±0.01**	**0.02±0.00**	**0.95±0.00**	**0.01±0.00**	**0.96±0.00**	**0.01±0.00**
	M2	0.79±0.02	0.06±0.00	0.89±0.01	0.04±0.00	0.90±0.00	0.03±0.00

D.2 Supervision using only a single label

When performing semi-supervision, the question of 'how low can we go?' naturally arises. Here we show that we can drop the supervision rate to the lowest possible level, that is, only having one label for a single instance of each class. This is achieved through an additional term, which makes use of the fact we can sample from $p_\psi(\boldsymbol{z}_c \mid \boldsymbol{y})$ and evaluate the likelihood on $q_\varphi(\boldsymbol{y} \mid \boldsymbol{z}_c)$.

A naïve way to increase the performance of the classifier, is to sample $\boldsymbol{z_c} \sim p_\psi(\boldsymbol{z}_c \mid \boldsymbol{y})$ and then maximize $\log q_\varphi(\boldsymbol{y} \mid \boldsymbol{z}_c)$ using $\boldsymbol{y}$ as the labels. Thus increasing the strength of the gradients to the classifier. A simple regularizer incorporating this objective yields:

$$\max_{\varphi} \mathbb{E}_{\boldsymbol{z_c} \sim p_\psi(\boldsymbol{z}_c \mid \boldsymbol{y})} \log q_\varphi(\boldsymbol{y} \mid \boldsymbol{z}_c)$$

However, there is no guarantee that a sample $\boldsymbol{z_c}^{(1)} \sim p_\psi(\boldsymbol{z}_c \mid \boldsymbol{y})$, will fall in the same decision boundary as $\boldsymbol{z_c}^{(2)} \sim q(\boldsymbol{z_c} \mid \boldsymbol{x_y})$, where $\boldsymbol{y}$ is the label for $\boldsymbol{x}$, thus providing adverse gradients. This is particularly common early on in training where the encoder distribution and the conditional prior do not have significant overlap. To counter this, rather than using $\boldsymbol{y}$ from $p_\psi(\boldsymbol{z}_c \mid \boldsymbol{y})$ to evaluate $\log q_\varphi(\boldsymbol{y} \mid \boldsymbol{z}_c)$, we instead use $\tilde{\boldsymbol{y}} = \arg\max_y q(\boldsymbol{z_c}|\boldsymbol{x}_y)$, where $q(\boldsymbol{z_c}|\boldsymbol{x})$ is stored from the single supervised example at a very small cost. This effectively acts as a NN classifier to obtain the appropriate label, but removes the dependence on a match between $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ and $p_\psi(\boldsymbol{z}_c \mid \boldsymbol{y})$. With

this approach, the final additional term is given as:

$$\bar{\mathcal{L}}_{\text{REVAE}}(\boldsymbol{x}) = \mathcal{L}_{\text{REVAE}}(\boldsymbol{x}) + \mu \mathbb{E}_{\boldsymbol{z_c} \sim p_\psi(\boldsymbol{z_c}|\boldsymbol{y})} \log q_\varphi(\tilde{\boldsymbol{y}} \mid \boldsymbol{z_c})$$
$$\tilde{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} \log q_\phi(\boldsymbol{z_c} \mid \boldsymbol{x}_y) \quad \forall \boldsymbol{x}_y \in \mathcal{S}$$

With $\mu = 5$, the resulting objective allows us to achieve 76.7% accuracy on MNIST where supervision is present for only a single random image for each class.

D.3 CelebA Interventions

We would like to highlight the addition of a demo application included in the supplementary (`'./demo/main.ipynb'`). The demo provides a user interface to alter characteristics on a chosen image. The demo performs latent walk interventions along the latent dimension for the corresponding chosen characteristic, thus providing a way alter multiple attributes as apposed to a pair in main paper (Figure 5). A screenshot is given in Figure 8, where the original image has been altered to add a smile and sunglasses.

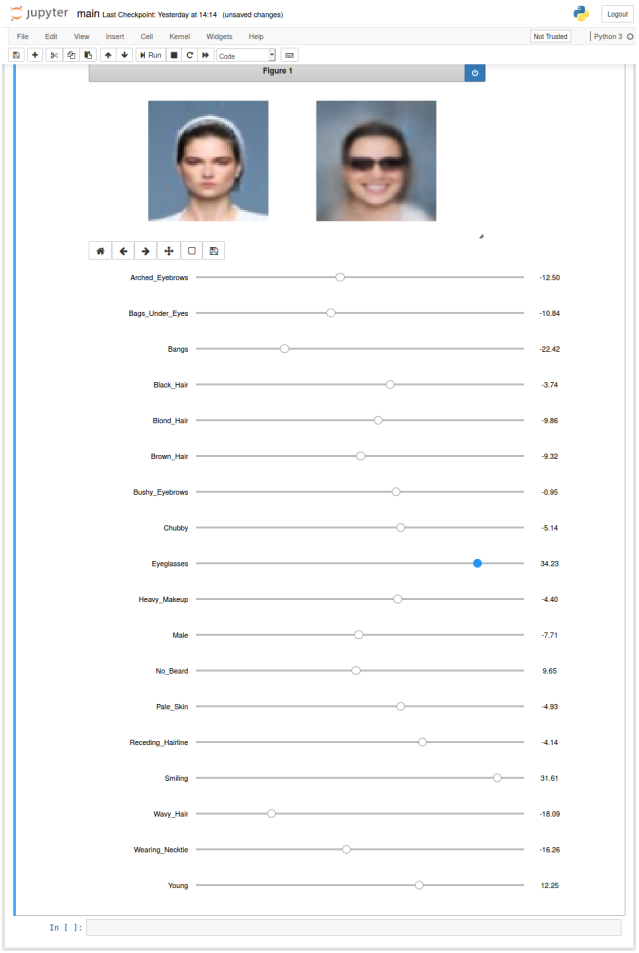

Figure 8: Screenshot of demo included in the supplementary.

We present additional qualitative results for single interventions on REVAE (Figure 9) and M2 (Figure 10).

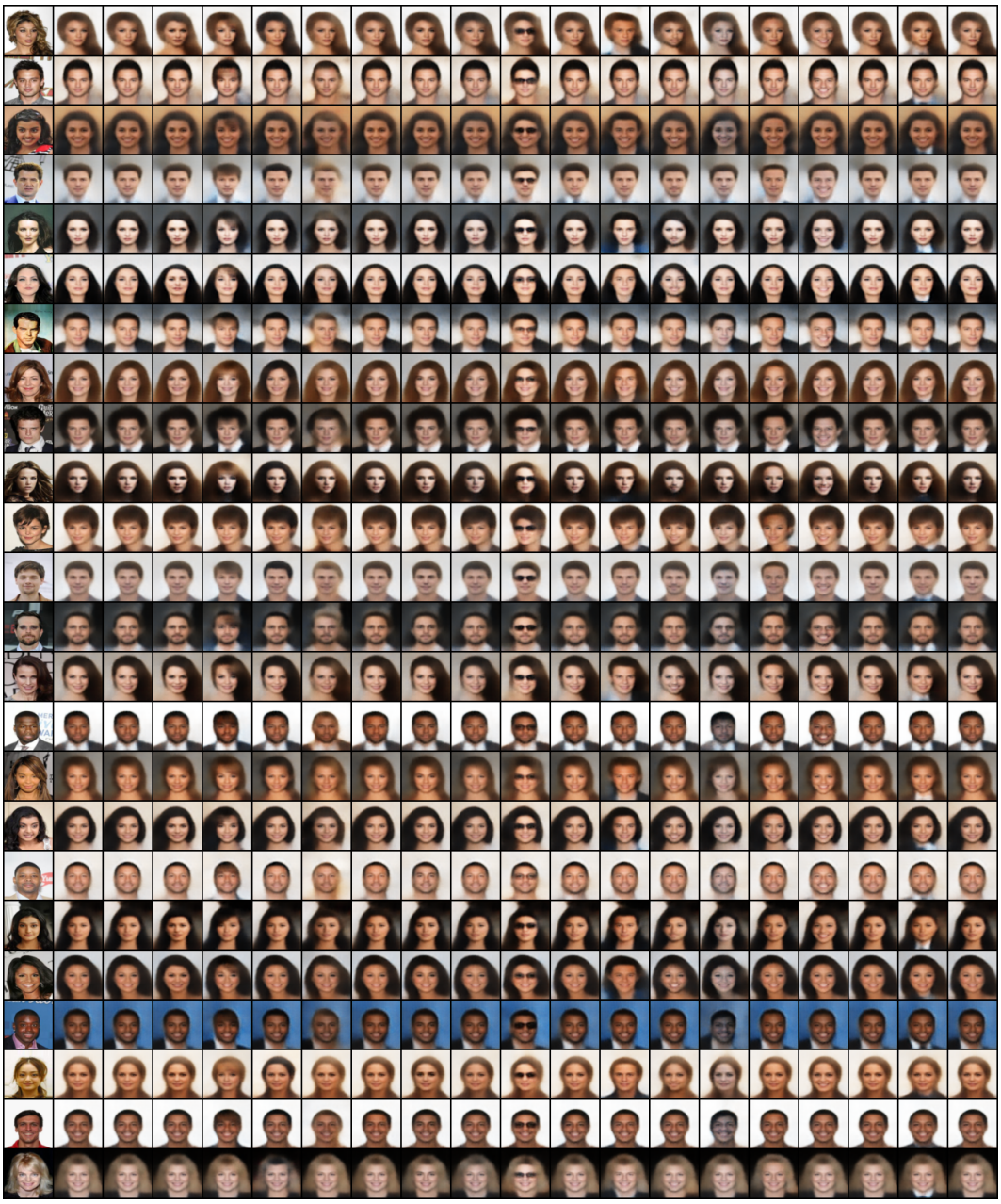

Figure 9: REVAE. From left to right: original, reconstruction, then interventions from switching on the following labels: `arched eyebrows`, `bags under eyes`, `bangs`, `black hair`, `blond hair`, `brown hair`, `bushy eyebrows`, `chubby`, `eyeglasses`, `heavy makeup`, `male`, `no beard`, `pale skin`, `receding hairline`, `smiling`, `wavy hair`, `wearing necktie`, `young`.

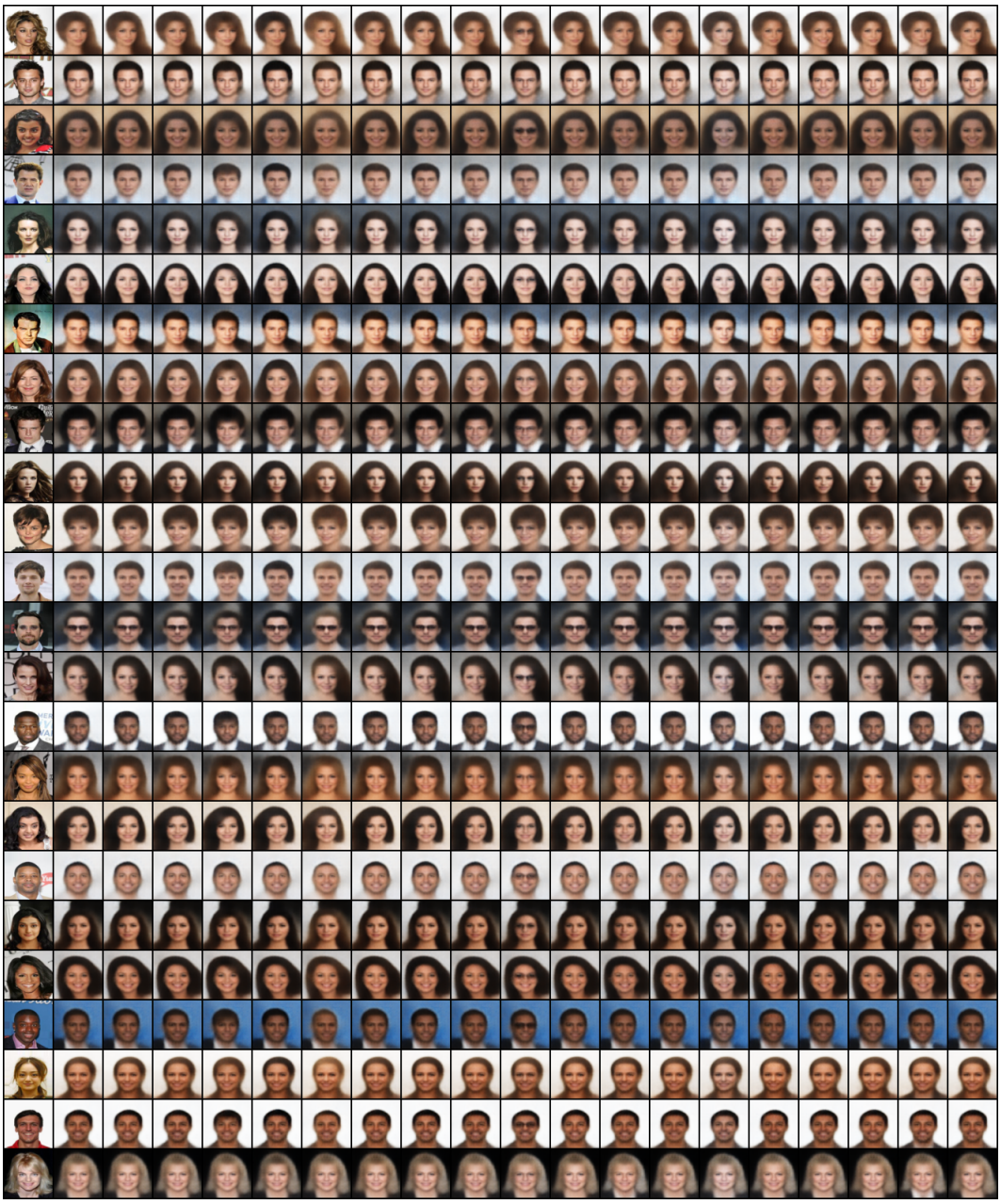

Figure 10: M2. From left to right: original, reconstruction, then interventions from switching on the following labels: `arched eyebrows`, `bags under eyes`, `bangs`, `black hair`, `blond hair`, `brown hair`, `bushy eyebrows`, `chubby`, `eyeglasses`, `heavy makeup`, `male`, `no beard`, `pale skin`, `receding hairline`, `smiling`, `wavy hair`, `wearing necktie`, `young`.