Identifiable Variational Autoencoders via Sparse Decoding

Gemma E. Moran∗1, Dhanya Sridhar1, Yixin Wang2, and David M. Blei1,3,4

1Data Science Institute, Columbia University
2Department of Statistics, University of Michigan
3Department of Statistics, Columbia University
4Department of Computer Science, Columbia University

October 22, 2021

Abstract

We develop the Sparse VAE, a deep generative model for unsupervised representation learning on high-dimensional data. Given a dataset of observations, the Sparse VAE learns a set of latent factors that captures its distribution. The model is sparse in the sense that each feature of the dataset (i.e., each dimension) depends on a small subset of the latent factors. As examples, in ratings data each movie is only described by a few genres; in text data each word is only applicable to a few topics; in genomics, each gene is active in only a few biological processes. We first show that the Sparse VAE is identifiable: given data drawn from the model, there exists a uniquely optimal set of factors. (In contrast, most VAE-based models are not identifiable.) The key assumption behind Sparse-VAE identifiability is the existence of “anchor features”, where for each factor there exists a feature that depends only on that factor. Importantly, the anchor features do not need to be known in advance. We then show how to fit the Sparse VAE with variational EM. Finally, we empirically study the Sparse VAE with both simulated and real data. We find that it recovers meaningful latent factors and has smaller heldout reconstruction error than related methods.

1 Introduction

We develop a new method for unsupervised representation learning on high-dimensional data. The goal is to find a low-dimensional representation of each datapoint—a vector of latent factors—that captures the variability in the distribution of the observed data (Bengio et al., 2013).

A deep generative model can help solve this problem (Kingma and Welling, 2014; Rezende et al., 2014). The model posits a low-dimensional vector of latent factors, passes it through a neural network, and probabilistically produces an observation. This model can be fit with a variational autoencoder. For each datapoint, the estimated factors are its representation.

Here, we consider the goal of finding a sparse solution to this representation learning problem, one where each feature depends on a subset of the latent factors (Bengio, 2017). Sparsity is an assumption that often reflects underlying patterns in data. In genomics, each gene is associated with a few biological processes; in text, each term is applicable to a few underlying topics; in movie ratings, each movie is associated with a few genres.

To this end, we develop the Sparse Variational Autoencoder (Sparse VAE). It posits a deep sparse generative model, where each of the features depends on a subset of the latent factors. (The latent

∗Email: gm2918@columbia.edu
factors themselves do not need to be sparse). We fit the Sparse VAE with amortized variational inference (Gershman and Goodman, 2014; Kingma and Welling, 2014), and we study it both theoretically and empirically.

Theoretically, we prove that the sparse deep generative model is identifiable. We assume that the true distribution satisfies what we refer to as the “anchor feature” assumption, that each factor has at least two features that depend only on that factor. Under this assumption, we then show that the Sparse VAE model is identified: given data drawn from the model, there is a uniquely optimal set of factors. Crucially, we do not require that the anchor features be known (only that they exist).

In contrast, the deep generative model fit by a standard VAE is not identifiable (Locatello et al., 2019a). There are many possible representations that fit the data equally well, and without further inductive biases it is impossible to discern which solution should be preferred based on the model likelihood alone (Yalcin and Amemiya, 2001; Locatello et al., 2019a).

We also study the Sparse VAE empirically. We show that (i) on synthetic data, the Sparse VAE recovers ground truth factors, including when the true factors are correlated; (ii) on text and movie-ratings data, the Sparse VAE achieves smaller heldout reconstruction error than comparable methods; (iii) on semi-synthetic data, the Sparse VAE has better heldout performance on test data that is distributed differently to training data; (iv) the Sparse VAE finds meaningful structure in text, movie-ratings and genomics data.

In summary, the contributions of this paper are as follows:

- We propose the Sparse VAE. It posits a deep sparse generative model, where each of the features depends on a subset of the latent factors.

- We prove the Sparse VAE is identifiable, where the key assumption we make is an anchor feature assumption. The anchor features do not need to be known a priori. The identifiability result also allows for the true generative factors to be correlated.

- We study the Sparse VAE empirically. It outperforms existing methods on synthetic, semi-synthetic, and real datasets of text, ratings, and genomics.

Related Work. The Sparse VAE is related to sparse methods in linear and nonlinear representation learning. The Sparse VAE places a sparsity-inducing prior on the factor-to-feature mapping (i.e., the decoder). This is similar to many linear factor analysis methods which also induce sparsity in the factor-to-feature mapping (i.e., the loadings matrix); see, for example Bernardo et al. (2003); Bhattacharya and Dunson (2011); Carvalho et al. (2008); Knowles et al. (2011); Ročková and George (2016).

Beyond linearity, Barello et al. (2018) consider a hybrid linear/nonlinear VAE model which uses a sparse linear decoder and a neural network encoder. In nonlinear representation learning, Tonolini et al. (2020) impose sparsity-inducing priors directly on the latent factors, instead of on the factor-to-feature mapping as in the Sparse VAE.

The Sparse VAE is most related to the OI-VAE (Ainsworth et al., 2018), a method for grouped features where the factor-to-group mapping is sparse. In contrast, the Sparse VAE induces sparsity in the factor-to-feature mapping. Moreover, we prove the Sparse VAE is identifiable, a property that Ainsworth et al. (2018) does not consider.

The Sparse VAE also contributes to the literature that uses the anchor feature assumption for identifiability. In linear models, the anchor feature assumption has led to identifiability results for topic models (Arora et al., 2013), non-negative matrix factorization (Donoho and Stodden, 2003), and linear factor analysis (Bing et al., 2020). The key idea is that the anchor assumption removes the rotational invariance of the latent factors. This idea also arises in identifiable factor analysis (Rohe and Zeng, 2020) and independent component analysis, where rotational invariance is removed by
Figure 1: In a VAE, a feature $x_{ij}$ depends on all factors, $z_{ik}$. A Sparse VAE is displayed where features $x_{i1}, x_{i2}$ depend only on $z_{i1}$; $x_{i3}$ depends on $z_{i1}$ and $z_{i2}$; and $x_{i4}, x_{i5}$ depend only on $z_{i2}$. The features are passed through the same neural network $f_\theta$.

assuming that the components are non-Gaussian (Eriksson and Koivunen, 2004).

Finally, the Sparse VAE is related to methods in identifiable nonlinear representation learning. These methods typically require auxiliary information or weak supervision for identifiability. Examples of weak supervision include the following strategies: using auxiliary information (Khemakhem et al., 2020; Locatello et al., 2019b); using paired samples (Locatello et al., 2020); using data augmentation (von Kügelgen et al., 2021); and leveraging temporal or spatial dependencies among the samples (Hälvä et al., 2021). In contrast, the Sparse VAE does not need auxiliary information or weak supervision for identifiability; instead, the anchor feature assumption is sufficient.

2 The Sparse Variational Autoencoder

The observed data is a vector of features, $x_i = (x_{i1}, \ldots, x_{iG}) \in \mathbb{R}^G$, for $i \in \{1, \ldots, N\}$ datapoints. The goal is to estimate a low-dimensional set of factors, $z_i = (z_{i1}, \ldots, z_{iK}) \in \mathbb{R}^K$, for each datapoint, where $K \ll G$.

In a standard VAE, the vector $z_i$ is a latent variable that is fed through a neural network to reconstruct the distribution of $x_i$. The Sparse VAE introduces an additional parameter $w_j \in \mathbb{R}^K$, a per-feature vector that selects which of the $K$ latent factors are used to produce the $j$th feature of $x_i$. In this work, we also allow for the true factors to be correlated. The Sparse VAE models the prior covariance between factors with the matrix $C \in \mathbb{S}^+_K$.

The Sparse VAE model is:

\begin{align*}
\theta &\sim \mathcal{N}(0, \tau) \quad (1) \\
w_{jk} &\sim \text{Spike-and-Slab Lasso}(\lambda_0, \lambda_1, a, b) \quad (2) \\
z_i &\sim \mathcal{N}_K(0, C) \quad (3) \\
x_{ij} &\sim \mathcal{N}((f_\theta(w_j \odot z_i))_j, \sigma_j^2) \quad (4)
\end{align*}

where $f_\theta : \mathbb{R}^K \to \mathbb{R}^G$ is a neural network with weights $\theta$, $\tau$ is a fixed regularization parameter, $\sigma_j^2$ is the per-feature noise variance and $\odot$ denotes element-wise multiplication. The Spike-and-Slab Lasso (Ročková and George, 2018) is a sparsity-inducing prior which we will describe below.

The parameter $w_{jk}$ controls whether the distribution of feature $j$ depends on factor $k$ for all the data. If $w_{jk} \neq 0$, then $z_{ik}$ contributes to $x_{ij}$, while if $w_{jk} = 0$, $z_{ik}$ cannot contribute to $x_{ij}$. This notion of sparsity is illustrated in Figure 1.

We place a Spike-and-Slab Lasso prior on $w_{jk}$, which involves additional latent variables (Ročková
Algorithm 1: The Sparse VAE

\textbf{input:} data $X$, hyperparameters $\lambda_0, \lambda_1, a, b, C$
\textbf{output:} factor distributions $q_\phi(z|x)$, selector matrix $W$, parameters $\theta$
\textbf{while not converged do}
\hspace{1em} For $j = 1, \ldots, G$; $k = 1, \ldots, K$, update:
\hspace{2em} $\mathbb{E}[\gamma_{jk} | w_{jk}, \eta_k] = [1 + (1 - \eta_k)/\eta_k \psi_0(w_{jk})/\psi_1(w_{jk})]^{-1}$.
\hspace{1em} For $k = 1, \ldots, K$, update:
\hspace{2em} $\eta_k = \left( \sum_{j=1}^{G} \mathbb{E}[\gamma_{jk} | w_{jk}, \eta_k] + a - 1 \right) / (a + b + G - 2)$.
\hspace{1em} Update $\theta, \phi, W$ with stochastic gradient ascent according to Eq. 9.
\textbf{end}

and George, 2018):

$$\eta_k \sim \text{Beta}(a, b),$$  \hspace{1em} (5)
$$\gamma_{jk} \sim \text{Bernoulli}(\eta_k),$$  \hspace{1em} (6)
$$w_{jk} \sim \gamma_{jk} \psi_1(w_{jk}) + (1 - \gamma_{jk}) \psi_0(w_{jk}).$$  \hspace{1em} (7)

In this prior, $\psi_s(w) = \frac{\lambda_s}{\lambda_s - 1} \exp(-\lambda_s |w|)$ is the Laplace density and $\lambda_0 \gg \lambda_1$. The variable $w_{jk}$ is drawn \textit{a priori} from either a Laplacian “spike” parameterized by $\lambda_0$ and is consequentially negligible, or a Laplacian “slab” parameterized by $\lambda_1$, and can be large.

The variable $\gamma_{jk}$ is a binary indicator variable that determines whether $w_{jk}$ is negligible. The Beta-Bernoulli prior on $\gamma_{jk}$ allows for uncertainty in determining which factors contribute to each feature.

The parameter $\eta_k \in [0, 1]$ controls the proportion of features that depend on factor $k$. By allowing $\eta_k$ to vary, the Sparse VAE allows each factor to contribute to different numbers of features. In movie-ratings data, for example, a factor corresponding to the action/adventure genre may be associated with more movies than a more esoteric genre.

Moreover, the prior on $\eta_k$ helps to “zero out” extraneous factor dimensions and consequently estimate the number of factors $K$. Specifically, if the hyperparameters are set to $a \propto 1/K$ and $b = 1$, the Beta-Bernoulli prior corresponds to the finite Indian Buffet Process prior (Griffiths and Ghahramani, 2005).

2.1 Inference

We fit the Sparse VAE with approximate maximum \textit{a posteriori} (MAP) estimation. We use a combination of coordinate ascent and gradient ascent on a variational bound of the MAP objective. In optimizing this bound, we fit an amortized variational approximation for the posterior of the representation $z_i$ (Kingma and Welling, 2014; Rezende et al., 2014) and the exact posterior for $\Gamma = \{\gamma_{jk}\}_{j,k=1}^{G,K}$.

The posterior of the factors, $p(z_i|x_i)$, is approximated by the variational family:

$$q_\phi(z_i|x_i) = N_K(\mu_\phi(x_i), \sigma^2_\phi(x_i)),$$  \hspace{1em} (8)

where $\mu_\phi(\cdot)$ and $\sigma^2_\phi(\cdot)$ are neural networks with weights $\phi$.  

4
The approximate MAP objective function is:

\[
\mathcal{L}(\theta, \phi, W, \eta) = \sum_{i=1}^{N} \left\{ \mathbb{E}_{q_{\phi}(z_i|x_i)} \left[ \log p_\theta(x_i|z_i, W) \right] - D_{KL}(q_{\phi}(z_i|x_i)||p(z_i)) \right\} \\
+ \mathbb{E}_{\Gamma|W^{(i)}, \eta^{(i)}} \left[ \log [p(W|\Gamma)p(\Gamma|\eta)p(\eta)] \right],
\]

where \( W = [w_1, \ldots, w_G]^T \in \mathbb{R}^{G \times K} \). To optimize \( \mathcal{L}(\theta, \phi, W, \eta) \), we alternate between an expectation step and a maximization step. In the E-step, we calculate the complete conditional posterior of \( \Gamma \). In the M-step, we take gradient steps in model parameters and variational parameters with the help of reparameterization gradients (Kingma and Welling, 2014; Rezende et al., 2014). The procedure is in Algorithm 1. Additional details of the derivation of Eq. 9 are available in Appendix B.

We note that the Sparse VAE requires \( G \) forward passes through the network at each gradient step, unlike a VAE which requires only one. This additional computation is because the Sparse VAE considers a different mapping for each of the \( G \) observed features.

### 2.2 Synthetic example

We now consider a synthetic dataset to provide intuition for the Sparse VAE. We have \( N = 1000 \) samples each with \( G = 7 \) features generated from \( K = 2 \) latent factors: for \( i = 1, \ldots, N \),

\[
x_i \sim \mathcal{N}(f(z_i), \sigma^2 I_G) \quad \text{with} \quad \sigma^2 = 0.5,
\]

where the first three features of the data depend on factor 1 as \( f(z_i)_1 = z_{i1} \), \( f(z_i)_2 = 2z_{i1} \), \( f(z_i)_3 = 3z_{i1}^2 \); features four through six depend on factor 2 as: \( f(z_i)_4 = 4z_{i2} \), \( f(z_i)_5 = 5z_{i2} \), \( f(z_i)_6 = 6 \sin(z_{i2}) \); and the seventh feature depends on both factors 1 and 2 as \( f(z_i)_7 = 7z_{i1} \cdot z_{i2} \). The true \( W \) matrix corresponding to this dependency is displayed in Figure 2c.

We take the true factors to be correlated and generate them as \( z_i \sim \mathcal{N}(0, C) \) where \( C \) has diagonal entries \( C_{kk} = 1 \) and off-diagonal entries \( C_{kk'} = 0.6 \) for \( k \neq k' \).

We initialize the Sparse VAE with an overestimate of the true number of factors (\( K_{\text{init}} = 5 \)). The Sparse VAE finds factors that reflect the true generative model (Figure 2a). Moreover, the Sparse VAE correctly finds the true number of factors by setting three columns of \( W \) to zero (Figure 2d). In contrast, the VAE recovers the second factor reasonably well, but doesn’t recover the first factor (Figure 2b).
3 Identifiability of the Sparse VAE

An identifiable model is one which has a unique set of factors and parameters which maximize the likelihood. In both linear and nonlinear factor analysis, the factors are usually unidentifiable without additional constraints on the model (Yalcin and Amemiya, 2001; Locatello et al., 2019a). This lack of identifiability means that the true factors $z_i$ can have the same likelihood as another solution $\tilde{z}_i$. We cannot choose between these representations based on the likelihood alone, and different representations may have different interpretations. Further inductive biases are necessary to narrow down the set of equally likely solutions.

In this section, we prove that the Sparse VAE model yields identifiable factors, up to coordinate-wise transformation. Specifically, we prove that for any solutions $z$ and $\tilde{z}$ with equal likelihood, we have for all factors $k = 1, \ldots, K$ (up to permutation):

$$z_{ik} = g_k(\tilde{z}_{ik}), \quad i = 1, \ldots, N,$$

for an invertible and differentiable function $g_k : \mathbb{R} \rightarrow \mathbb{R}$. This definition of identifiability is weaker than the canonical definition, for which we would need the two solutions $z$ and $\tilde{z}$ to be exactly equal.

We use this weaker notion of identifiability because our goal is to isolate the dimensions of $z_i$ which drive the observed response, and not necessarily find their exact value—i.e. we want to avoid mixing the dimensions of $z_i$. For example, we want to be able to distinguish between solutions $z$ and $\tilde{z} = Pz$, for arbitrary rotation matrices $P$.

Starting point. This paper builds on a body of work in identifiable linear unsupervised models (Arora et al., 2013; Bing et al., 2020; Donoho and Stodden, 2003). These works assume the existence of anchor features, features which depend on only one factor. The existence of anchor features helps with identifiability because they leave a detectable imprint in the covariance of the observed data.

We extend these results to the nonlinear setting. The key insight is that if two anchor features have the same nonlinear dependence on a factor, then we can apply results from the linear setting. We now formally state the anchor feature assumption.

A1. (Anchor feature assumption) For every factor, $z_k$, there are at least two features $x_{ij}, x_{ij'}$ which depend only on that factor. Moreover, the two features have the same mapping, $f_j$, from the factors, $z_k$; that is, for all $i = 1, \ldots, N$:

$$\mathbb{E}[x_{ij}|z_i] = f_j(z_{ik}), \quad \mathbb{E}[x_{ij'}|z_i] = f_j(z_{ik}).$$

We refer to such features as “anchor features.”

Roadmap. We prove the identifiability of the Sparse VAE model in two parts. First, we prove that the latent factors are identifiable up to coordinate-wise transformation when the anchor features are known (Theorem 1). Second, we prove that the anchor features can be detected from the covariance matrix of the data (Theorem 2). Together, Theorem 1 and Theorem 2 give the identifiability of the Sparse VAE model.

Known anchor features. The first result, Theorem 1, proves that the Sparse VAE factors are identifiable if we are given the anchor features. If feature $j$ is an anchor feature for factor $k$, we set $w_{jk} = 1$ and $w_{jk'} = 0$ for all $k' \neq k$.

Theorem 1. Assume the model in Eq. 4 with A1 holds. Assume we are given the rows of $W$ corresponding to the anchor features. Suppose we have two solutions with equal likelihood: $\{\theta, \tilde{z}\}$ and $\{\tilde{\theta}, \bar{z}\}$, with

$$p_{\tilde{\theta}}(x|\bar{z}, W) = p_{\tilde{\theta}}(x|\bar{z}, W).$$

Starting point. This paper builds on a body of work in identifiable linear unsupervised models (Arora et al., 2013; Bing et al., 2020; Donoho and Stodden, 2003). These works assume the existence of anchor features, features which depend on only one factor. The existence of anchor features helps with identifiability because they leave a detectable imprint in the covariance of the observed data.

We extend these results to the nonlinear setting. The key insight is that if two anchor features have the same nonlinear dependence on a factor, then we can apply results from the linear setting. We now formally state the anchor feature assumption.

A1. (Anchor feature assumption) For every factor, $z_k$, there are at least two features $x_{ij}, x_{ij'}$ which depend only on that factor. Moreover, the two features have the same mapping, $f_j$, from the factors, $z_k$; that is, for all $i = 1, \ldots, N$:

$$\mathbb{E}[x_{ij}|z_i] = f_j(z_{ik}), \quad \mathbb{E}[x_{ij'}|z_i] = f_j(z_{ik}).$$

We refer to such features as “anchor features.”

Roadmap. We prove the identifiability of the Sparse VAE model in two parts. First, we prove that the latent factors are identifiable up to coordinate-wise transformation when the anchor features are known (Theorem 1). Second, we prove that the anchor features can be detected from the covariance matrix of the data (Theorem 2). Together, Theorem 1 and Theorem 2 give the identifiability of the Sparse VAE model.

Known anchor features. The first result, Theorem 1, proves that the Sparse VAE factors are identifiable if we are given the anchor features. If feature $j$ is an anchor feature for factor $k$, we set $w_{jk} = 1$ and $w_{jk'} = 0$ for all $k' \neq k$.

Theorem 1. Assume the model in Eq. 4 with A1 holds. Assume we are given the rows of $W$ corresponding to the anchor features. Suppose we have two solutions with equal likelihood: $\{\theta, \tilde{z}\}$ and $\{\tilde{\theta}, \bar{z}\}$, with

$$p_{\tilde{\theta}}(x|\bar{z}, W) = p_{\tilde{\theta}}(x|\bar{z}, W).$$

Starting point. This paper builds on a body of work in identifiable linear unsupervised models (Arora et al., 2013; Bing et al., 2020; Donoho and Stodden, 2003). These works assume the existence of anchor features, features which depend on only one factor. The existence of anchor features helps with identifiability because they leave a detectable imprint in the covariance of the observed data.

We extend these results to the nonlinear setting. The key insight is that if two anchor features have the same nonlinear dependence on a factor, then we can apply results from the linear setting. We now formally state the anchor feature assumption.

A1. (Anchor feature assumption) For every factor, $z_k$, there are at least two features $x_{ij}, x_{ij'}$ which depend only on that factor. Moreover, the two features have the same mapping, $f_j$, from the factors, $z_k$; that is, for all $i = 1, \ldots, N$:

$$\mathbb{E}[x_{ij}|z_i] = f_j(z_{ik}), \quad \mathbb{E}[x_{ij'}|z_i] = f_j(z_{ik}).$$

We refer to such features as “anchor features.”

Roadmap. We prove the identifiability of the Sparse VAE model in two parts. First, we prove that the latent factors are identifiable up to coordinate-wise transformation when the anchor features are known (Theorem 1). Second, we prove that the anchor features can be detected from the covariance matrix of the data (Theorem 2). Together, Theorem 1 and Theorem 2 give the identifiability of the Sparse VAE model.

Known anchor features. The first result, Theorem 1, proves that the Sparse VAE factors are identifiable if we are given the anchor features. If feature $j$ is an anchor feature for factor $k$, we set $w_{jk} = 1$ and $w_{jk'} = 0$ for all $k' \neq k$.

Theorem 1. Assume the model in Eq. 4 with A1 holds. Assume we are given the rows of $W$ corresponding to the anchor features. Suppose we have two solutions with equal likelihood: $\{\theta, \tilde{z}\}$ and $\{\tilde{\theta}, \bar{z}\}$, with

$$p_{\tilde{\theta}}(x|\bar{z}, W) = p_{\tilde{\theta}}(x|\bar{z}, W).$$
Then, the factors \( \tilde{z} \) and \( \bar{z} \) are equal up to coordinate wise transformations. For \( g_k : \mathbb{R} \rightarrow \mathbb{R} \) and \( i = 1, \ldots, N \),

\[
(\tilde{z}_i, \ldots, \tilde{z}_K) = (g_1(\tilde{z}_i), \ldots, g_K(\tilde{z}_i)).
\] (14)

The proof of Theorem 1 is in Appendix A.1. The main idea of the proof is that if the anchor features are known, then we are given noisy transformations of the coordinates of \( z \). As we only need to identify \( z \) up to coordinate-wise transform, this essentially solves the problem.

**Unknown anchor features.** The second result proves that the anchor features can be determined from an approximation of the covariance matrix of the data \( X \), under additional assumptions.

The next assumption concerns the weights of the neural network, \( f_\theta \). For this assumption, we require some additional notation. After applying the chain rule to the first layer, we rewrite the derivative of mapping \( f_\theta \) as follows:

\[
\frac{\partial (f_\theta(w_j \odot z_i))_j}{\partial z_{ik}} = \sum_{d=1}^{D_1} \frac{\partial g_\theta(u_{ijd})_j}{\partial u_{ijd}} H^{(1)}_{dk} w_{jk},
\] (15)

where \( (H^{(1)})_{d,k} \) are the weights of the first neural network layer and \( D_1 \) is the dimension of the first layer, \( u_{ijd} = \sum_{k=1}^{K} H_{dk} w_{jk} z_{ik} \) is the first layer before the activation function, and \( g_\theta : \mathbb{R}^{D_1} \rightarrow \mathbb{R}^{G} \) is the rest of the neural network which takes as input the first layer \( (u_{ijd})_{d=1}^{D_1} \). Let \( B_{ijk} = \sum_{d=1}^{D_1} \frac{\partial g_\theta(u_{ijd})_j}{\partial u_{ijd}} H^{(1)}_{dk} \).

**A2.** Suppose \( j \) is an anchor feature for factor \( k \). For another feature \( l \), we assume that

\[
|w_{jk}| \sum_{i=1}^{N} |B_{ijk}|^2 \geq \sum_{i=1}^{N} \sum_{k'=1}^{K} |w_{ik'}| |B_{ijk}| |B_{ik'}|,
\] (16)

with equality when \( l \) is also an anchor feature for \( k \) and inequality otherwise.

Essentially, this assumption ensures that the covariance between two anchor features is larger than the covariance between an anchor feature and a non-anchor feature. This fact then allows us to pinpoint the anchor features in the covariance matrix of the data. In Appendix A.3, we show this assumption holds for a neural network with ReLU activations and independent weights.

Finally, we also require an assumption regarding the covariance of the factors: this is the same as assumption (iii) of Bing et al. (2020).

**A3.** Denote the covariance matrix of the factors as \( \text{Cov}(z_i) = C \). We assume \( \min\{C_{kk'}, C_{kk''}\} > |C_{kk'}| \), for all \( k, k' = 1, \ldots, K \).

Assumption A3 requires that the variance of each factor be greater than its covariance with any other factor. To gain some intuition, consider a dataset of news articles where the latent factors are topics, sports and politics. Assumption A3 would be violated if every article about sports also discussed politics (and vice versa). In this case, the anchor word for sports will be equally as correlated with an anchor word for politics as it is with another anchor word for sports. That is, there is no discernible difference in the data that helps us distinguish the sports and politics factors.

**Theorem 2.** Assume the model Eq. 4 with A1 – 3 holds. Then, the set of anchor features can be determined uniquely from \( \frac{1}{N} \sum_{i=1}^{N} \text{Cov}(z_i) \) as \( N \rightarrow \infty \), (given additional regularity conditions detailed in Appendix A.2).

The proof of Theorem 2 is in Appendix A.2.
Proof idea: We adapt the proof technique of Bing et al. (2020), which considers linear factor analysis. The proof idea of that paper is that for any two features \( x_j \) and \( x_{j'} \) that are anchors for the same factor, their covariance \( \text{Cov}(x_j, x_{j'}) \) will be greater than their covariance with any other non-anchor feature (under some conditions on the loadings matrix, analogous to A2). Then, the anchor features can be pinpointed from the observed covariance matrix. In the nonlinear case, we can apply a similar strategy: even if the mapping is nonlinear, if the mapping is the same for the anchor features, we can pinpoint the two anchors in the covariance matrix.

To summarize, Theorem 2 proves that the anchor features can be determined from the covariance of the data. Given the anchor features, Theorem 1 then proves that the Sparse VAE model is identifiable.

Connection to the Sparse VAE algorithm. Theorem 2 proves that the existence of anchor features implies a particular structure in the covariance of the data. Consequently, an appropriate estimation procedure can then pinpoint the anchor features. In the Sparse VAE algorithm, we do not directly consider the covariance matrix as it would involve laborious hyperparameter tuning to determine which covariance entries are “close enough” to be anchor features. Instead, we estimate \( W \) with a sparsity-inducing prior, motivated by the sparsity in the decoder specifically for the anchor features. Due to the Gaussian likelihood assumption in Eq. 4, the Sparse VAE algorithm indirectly fits the covariance matrix of the data.

4 Experiments

We empirically study the Sparse VAE using a mix of synthetic, semi-synthetic and real data.\(^1\) We consider the following questions: 1) How well does the Sparse VAE recover ground truth factors when (i) the factors are uncorrelated and (ii) the factors are correlated? 2) How does the heldout reconstruction of the Sparse VAE compare to related methods? 3) How does the Sparse VAE perform compared to related methods when the correlation between factors is different in the training and test data? 4) Does the Sparse VAE find meaningful structure in data?

We compare the Sparse VAE to three other deep generative models: the VAE (Kingma and Welling, 2014), \( \beta \)-VAE (Higgins et al., 2017), and Variational Sparse Coding (VSC, Tonolini et al., 2020). None of these methods are identifiable. We find that: 1) on synthetic data, the Sparse VAE achieves the best recovery of ground truth factors both when the factors are correlated and uncorrelated; 2) the Sparse VAE achieves the best predictive performance on both synthetic and real datasets; and 3) the Sparse VAE achieves the best predictive performance on heldout data drawn from factors that are correlated differently than in the training data.

Further, we find that the Sparse VAE finds meaningful factors in both text and movie-ratings data. On a single-cell genomics dataset labeled by cell type, the Sparse VAE finds clusters of genes which correspond to the different cell types.

Datasets. We consider the following datasets; further details about the data are available in Appendix C.1.

- **Synthetic data.** We consider again the synthetic dataset generated as Eq. 10. We vary correlation between the true factors from \( \rho = 0, 0.2, 0.4, 0.6, 0.8 \).

- **PeerRead** (Kang et al., 2018). Dataset of word counts for paper abstracts (\( N \approx 10,000, G = 500 \)).

- **MovieLens** (Harper and Konstan, 2015). Dataset of binary user-movie ratings (\( N = 100,000, G = 300 \)).

- **Semi-synthetic PeerRead.** A semi-synthetic version of the PeerRead dataset generated so that the correlations between data-generating factors differ across the test and training data, inducing

\(^1\)Code for the Sparse VAE is available at [https://github.com/gemoran/sparse-vae-code](https://github.com/gemoran/sparse-vae-code).
Figure 3: Synthetic data. (a) Sparse VAE has better heldout predictive performance than the VAE over a range of factor correlation levels. (b) Sparse VAE recovers the true factors better than the VAE. ($\beta$-VAE performed similarly to VAE). Scores are shown for 25 datasets per correlation setting.

Different training and test distributions.

* Zeisel (Zeisel et al., 2015). Dataset of RNA molecule counts in mouse cortex cells ($N = 3005$, $G = 558$).

Implementation details. Empirically, we found that setting the prior covariance of the factors to $C = I_K$ worked well, even when the true generative factors were correlated. The complete implementation details for all methods are in Appendix C.

Recovering the ground truth factors. How well does the Sparse VAE recover the factors when the ground truth factors are known? We assess this question with simulated data. We use the DCI disentanglement score to measure the fidelity between estimated factors and true factors (Eastwood and Williams, 2018). The DCI score is an average measure of how relevant each estimated factor is for the true factors. This score also penalizes estimated factors that are equally informative for multiple true factors.

We create synthetic datasets with the model given in Eq. 10 and evaluate the DCI metric between estimated and true factors as the true factors are increasingly correlated. As the correlation increases, we expect a standard VAE to conflate the factors while the Sparse VAE recovers the two true underlying factors.

We see this phenomenon in Figure 3b; the Sparse VAE has higher DCI scores than the VAE in all settings. The Sparse VAE is robust to factor correlations up to $\rho = 0.4$, with decreasing performance as $\rho$ is further increased. Here, VSC performs worse than both the Sparse VAE and VAE (the MSE scores for VSC in were too large for visualization in Figure 3a). The poor performance of VSC is likely due to the true generative factors in Eq. 10 not being sparse; only the mapping between the factors and features is sparse.

Heldout reconstruction. We compare the log loss on heldout data achieved by the Sparse VAE and related methods. Figure 3a shows that the Sparse VAE performs better than the VAE and VSC on synthetic data. On MovieLens and PeerRead datasets, Table 1 shows that the Sparse VAE achieves the lowest heldout log loss among the compared methods. For the MovieLens dataset, Table 1 additionally shows that the Sparse VAE has the highest heldout Recall@5 and normalized discounted cumulative gain (NDCG@10), which compare the predicted rank of heldout items to their true rank (Liang et al., 2018).

Different training and test distributions. How does the Sparse VAE perform when the factors that generate data are correlated differently across training and test distributions? This particular type of shift in distribution affects many real world settings. For example, we may estimate document representations from scientific papers where articles about machine learning also often discuss genomics, and want to use the representations to analyze new collections of papers where articles about machine learning rarely involve genomics. We hypothesize that because the Sparse VAE associates each latent factor with only a few features (e.g., words) even when the factors are correlated, it will reconstruct differently distributed test data better than the related methods.
Table 1: The Sparse VAE achieves the best heldout performance on text and movie-ratings datasets. Results are averaged over five splits of the data, with standard deviation in parentheses. For the $\beta$-VAE, we show the result with best performing $\beta$.

| Method     | MovieLens | | | | | PeerRead | | | |
|------------|-----------| | Log loss | Recall@5 | NDCG@10 | Log Loss | | | |
| Sparse VAE | 170.9 (2.1) | 0.98 (0.002) | 0.98 (0.003) | Sparse VAE | 245.0 (2.0) |
| VAE        | 175.9 (2.4) | 0.97 (0.001) | 0.96 (0.001) | VAE | 252.6 (1.4) |
| $\beta$-VAE ($\beta = 2$) | 178.2 (2.4) | 0.95 (0.002) | 0.93 (0.002) | $\beta$-VAE ($\beta = 2$) | 254.5 (3.0) |
| VSC        | 192.2 (2.3) | 0.79 (0.008) | 0.77 (0.009) | VSC | 252.9 (2.0) |

Table 2: In the semi-synthetic PeerRead data where the training and test data are generated from underlying factors with different correlations, the Sparse VAE achieves the lowest test log loss. We create three settings where the difference between training and test data distributions range from high (hardest) to low (easiest). We report the average log loss across five simulated datasets for each setting.

| Method     | Difference between train and test | | | | | | | |
|------------|----------------------------------| | High | Medium | Low | | | | |
| Sparse VAE | 52.4 (0.4) | 49.2 (0.3) | 48.6 (0.1) | Sparse VAE | 245.0 (2.0) |
| VAE        | 54.6 (0.5) | 52.3 (0.2) | 50.8 (0.2) | VAE | 252.6 (1.4) |
| $\beta$-VAE ($\beta = 2$) | 54.7 (0.3) | 52.1 (0.2) | 51 (0.4) | $\beta$-VAE ($\beta = 2$) | 254.5 (3.0) |
| VSC        | 58.7 (0.6) | 56.1 (0.3) | 55.4 (0.2) | VSC | 252.9 (2.0) |

We assess this question using the semi-synthetic PeerRead dataset, where the train and test data were generated by factors with different correlations. Table 2 summarizes the results from three settings where the difference between training and test data distributions range from high (hardest) to low (easiest). We report the average log loss across five semi-simulated datasets. The Sparse VAE performs the best, highlighting its ability to estimate models which generalize better to test data where the factors are distributed differently.

Interpretability. We now examine whether the Sparse VAE finds meaningful structure in the data. For each factor in the MovieLens dataset, we consider the four movies with the largest selector variable $\tilde{w}_{jk}$ values (Table 3). The Sparse VAE finds clear patterns: for example, the top movies in first factor are all science fiction movies; the second factor contains animated children’s movies; the third factor contains three Star Wars movies and an Indiana Jones movie, all blockbuster science fiction movies from the 1980s. We perform the same analysis for the PeerRead dataset (Table 3). The Sparse VAE finds some themes of computer science papers: the first factor contains words related to reinforcement learning; the second factor contains words about information theory; the third factor involves words about neural networks.

Table 3: On MovieLens and PeerReads, the Sparse VAE finds meaningful topics via the matrix $W$.

| Topic | MovieLens | PeerReads | |
|-------|-----------|-----------||
| A     | The Fifth Element; Alien; Gattaca; Aliens | task; policy; planning; heuristic; decision | |
| B     | A Bug’s Life; Monsters, Inc.; Toy Story 3 & 2 | information; feature; complex; sparse; probability | |
| C     | Star Wars V, IV & IV; Indiana Jones 1 | given; network; method; bayesian; neural | |
Finally, we consider the single-cell genomics dataset of (Zeisel et al., 2015). Each of the $N = 3005$ cells in this dataset are labeled with their cell type. We examine how well the factors found by the Sparse VAE capture this cell-type information. We visualize the estimated factors using tSNE (Van der Maaten and Hinton, 2008). We see that the Sparse VAE factors more clearly separate the cell types than VSC, a VAE or non-negative matrix factorization (NMF) (Figure 4).

5 Discussion

We proposed the Sparse VAE, a deep generative model with priors that induce sparsity in the mapping between factors and features. Under an anchor feature assumption, we prove that the Sparse VAE model yields identifiable factors. On real and synthetic data, we show the Sparse VAE performs well. (i) It has good heldout predictive performance. (ii) It generalizes to out of distribution data. (iii) It is interpretable.

There are a few limitations of this work. First, the Sparse VAE is designed for tabular data, where each feature $j$ has a consistent meaning across samples. Image data does not have this property as a specific pixel has no consistent meaning across samples. Second, the Sparse VAE is more computationally intensive than a standard VAE. This expense is because it requires $G$ forward passes through the network at each gradient step. Finally, we proved that the Sparse VAE model is identifiable, but we did not theoretically consider the estimation properties of the Sparse VAE algorithm. Similarly to other deep generative models, estimation guarantees for the Sparse VAE algorithm are difficult to obtain due to the nonconvexity of the objective function. The majority of works which consider identifiability in VAEs also do not provide estimation guarantees for their algorithms, including Locatello et al. (2020); Khemakhem et al. (2020). We leave the theoretical study of the convergence of the Sparse VAE algorithm to future work.

Bibliography

Ainsworth, S. K., Foti, N. J., Lee, A. K., and Fox, E. B. (2018). oi-VAE: Output interpretable VAEs for nonlinear group factor analysis. In International Conference on Machine Learning.

Arora, S., Ge, R., Halperrn, Y., Mimno, D., Moitra, A., Sontag, D., Wu, Y., and Zhu, M. (2013). A practical algorithm for topic modeling with provable guarantees. In International Conference on Machine Learning.

Barello, G., Charles, A. S., and Pillow, J. W. (2018). Sparse-coding variational auto-encoders. bioRxiv, page 399246.

Bengio, Y. (2017). The consciousness prior. arXiv preprint arXiv:1709.08568.

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.

Bernardo, J., Bayarri, M., Berger, J., Dawid, A., Heckerman, D., Smith, A., and West, M. (2003). Bayesian factor regression models in the “large p, small n” paradigm. Bayesian Statistics, 7:733–742.
Bhattacharya, A. and Dunson, D. B. (2011). Sparse Bayesian infinite factor models. *Biometrika*, pages 291–306.

Bing, X., Bunea, F., Ning, Y., Wegkamp, M., et al. (2020). Adaptive estimation in structured factor models with applications to overlapping clustering. *Annals of Statistics*, 48(4):2055–2081.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Bolstad, B. (2018). *preprocessCore: A collection of pre-processing functions*. R package version 1.44.0.

Bolstad, B., Irizarry, R., Åstrand, M., and Speed, T. (2003). A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185–193.

Carvalho, C. M., Chang, J., Lucas, J. E., Nevins, J. R., Wang, Q., and West, M. (2008). High-dimensional sparse factor modeling: applications in gene expression genomics. *Journal of the American Statistical Association*, 103(484):1438–1456.

Donoho, D. L. and Stodden, V. (2003). When does non-negative matrix factorization give a correct decomposition into parts? In *Neural Information Processing Systems*.

Eastwood, C. and Williams, C. K. (2018). A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*.

Eriksson, J. and Koivunen, V. (2004). Identifiability, separability, and uniqueness of linear ica models. *IEEE Signal Processing Letters*, 11(7):601–604.

Gershman, S. and Goodman, N. (2014). Amortized inference in probabilistic reasoning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 36.

Griffiths, T. L. and Ghahramani, Z. (2005). Infinite latent feature models and the Indian buffet process. In *Neural Information Processing Systems*.

Hälvä, H., Corff, S. L., Lehéricy, L., So, J., Zhu, Y., Gassiat, E., and Hyvarinen, A. (2021). Disentangling identifiable features from noisy data with structured nonlinear ica. *arXiv preprint arXiv:2106.09620*.

Harper, F. M. and Konstan, J. A. (2015). The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 5(4):1–19.

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

Kang, D., Ammar, W., Dalvi, B., van Zuylen, M., Kohlmeier, S., Hovy, E., and Schwartz, R. (2018). A dataset of peer reviews (PeerRead): Collection, insights and NLP applications. *arXiv preprint arXiv:1804.09635*.

Khemakhem, I., Kingma, D., Monti, R., and Hyvarinen, A. (2020). Variational autoencoders and nonlinear ICA: A unifying framework. In *Artificial Intelligence and Statistics*.

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

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

Knowles, D., Ghahramani, Z., et al. (2011). Nonparametric Bayesian sparse factor models with application to gene expression modeling. *Annals of Applied Statistics*, 5(2B):1534–1552.
Appendix

A Proofs

A.1 Proof of Theorem 1

We consider the case of known anchor features, where we have at least two anchor features per factor. We consider the case where the likelihood is Gaussian. We have two solutions, $(\theta, \tilde{z}, \tilde{\sigma}^2)$ and $(\hat{\theta}, \tilde{z}, \tilde{\sigma}^2)$, with equal likelihood. We show that $\tilde{z}$ must be a coordinate-wise transform of $\tilde{z}$. 

Liang, D., Krishnan, R. G., Hoffman, M. D., and Jebara, T. (2018). Variational autoencoders for collaborative filtering. In World Wide Web Conference.

Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., and Bachem, O. (2019a). Challenging common assumptions in the unsupervised learning of disentangled representations. In International Conference on Machine Learning.

Locatello, F., Poole, B., Rätsch, G., Schölkopf, B., Bachem, O., and Tschannen, M. (2020). Weakly-supervised disentanglement without compromises. In International Conference on Machine Learning.

Locatello, F., Tschannen, M., Bauer, S., Rätsch, G., Schölkopf, B., and Bachem, O. (2019b). Disentangling factors of variations using few labels. In International Conference on Learning Representations.

Lopez, R., Regier, J., Cole, M. B., Jordan, M. I., and Yosef, N. (2018). Deep generative modeling for single-cell transcriptomics. Nature methods, 15(12):1053–1058.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In International Conference on Machine Learning.

Ročková, V. and George, E. I. (2016). Fast Bayesian factor analysis via automatic rotations to sparsity. Journal of the American Statistical Association, 111(516):1608–1622.

Ročková, V. and George, E. I. (2018). The spike-and-slab lasso. Journal of the American Statistical Association, 113(521):431–444.

Rohe, K. and Zeng, M. (2020). Vintage factor analysis with varimax performs statistical inference. arXiv preprint arXiv:2004.05387.

Tonolini, F., Jensen, B. S., and Murray-Smith, R. (2020). Variational sparse coding. In Uncertainty in Artificial Intelligence, pages 690–700. PMLR.

Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-SNE. Journal of Machine Learning Research, 9(11).

von Kügelgen, J., Sharma, Y., Gresele, L., Brendel, W., Schölkopf, B., Besserve, M., and Locatello, F. (2021). Self-supervised learning with data augmentations provably isolates content from style. arXiv preprint arXiv:2106.04619.

Yalcin, I. and Amemiya, Y. (2001). Nonlinear factor analysis as a statistical method. Statistical science, pages 275–294.

Zeisel, A., Muñoz-Manchado, A. B., Codeluppi, S., Lönnerberg, P., La Manno, G., Juréus, A., Marques, S., Munguba, H., He, L., Betsholtz, C., et al. (2015). Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. Science, 347(6226):1138–1142.

Appendix

A Proofs

A.1 Proof of Theorem 1

We consider the case of known anchor features, where we have at least two anchor features per factor. We consider the case where the likelihood is Gaussian. We have two solutions, $(\theta, \tilde{z}, \tilde{\sigma}^2)$ and $(\hat{\theta}, \tilde{z}, \tilde{\sigma}^2)$, with equal likelihood. We show that $\tilde{z}$ must be a coordinate-wise transform of $\tilde{z}$.
We have

\[
\sum_{i=1}^{N} \sum_{j=1}^{G} \frac{1}{\sigma_j^2} [x_{ij} - f_{\theta_j}(\tilde{w}_{jk} \odot \tilde{z}_{ik})]^2 = \sum_{i=1}^{N} \sum_{j=1}^{G} \frac{1}{\sigma_j^2} [x_{ij} - f_{\theta_j}(\tilde{w}_{jk} \odot \tilde{z}_{ik})]^2,
\]

(17)

where \( f_{\theta_j}(\cdot) \) denotes \( f_\theta(\cdot) \), the \( j \)th output of \( f_\theta \).

Suppose \( j \) is an anchor feature for \( k \). That is, \( w_{jl} = 0 \) for all \( l \neq k \). We have:

\[
\sum_{i=1}^{N} \frac{1}{\sigma_j^2} [x_{ij} - f_{\theta_j}(\tilde{w}_{jk} \tilde{z}_{ik})]^2 = \sum_{i=1}^{N} \frac{1}{\sigma_j^2} [x_{ij} - f_{\theta_j}(\tilde{w}_{jk} \tilde{z}_{ik})]^2 \quad (18)
\]

\[
\left( \frac{1}{\sigma_j^2} - \frac{1}{\sigma_j^2} \right) \sum_{i=1}^{N} x_{ij}^2 - 2 \sum_{i=1}^{N} x_{ij} \left( \frac{f_{\theta_j}(\tilde{w}_{jk} \tilde{z}_{ik})}{\sigma_j^2} - \frac{f_{\theta_j}(\tilde{w}_{jk} \tilde{z}_{ik})}{\sigma_j^2} \right) + \sum_{i=1}^{N} \left( \frac{f_{\theta_j}(\tilde{w}_{jk} \tilde{z}_{ik})}{\sigma_j^2} - \frac{f_{\theta_j}(\tilde{w}_{jk} \tilde{z}_{ik})}{\sigma_j^2} \right) = 0. \quad (19)
\]

For Eq. 19 to hold for all \( x_{ij} \), we must have the coefficients equal to zero:

\[
\frac{1}{\sigma_j^2} - \frac{1}{\sigma_j^2} = 0 \implies \sigma_j^2 = \sigma_j^2 \quad (20)
\]

and

\[
\frac{f_{\theta_j}(\tilde{w}_{jk} \tilde{z}_{ik})}{\sigma_j^2} = \frac{f_{\theta_j}(\tilde{w}_{jk} \tilde{z}_{ik})}{\sigma_j^2} \implies f_{\theta_j}(\tilde{w}_{jk} \tilde{z}_{ik}) = f_{\theta_j}(\tilde{w}_{jk} \tilde{z}_{ik}). \quad (21)
\]

If \( f_{\theta_j} : \mathbb{R} \to \mathbb{R} \) is invertible, then we have

\[
\tilde{z}_{ik} = \tilde{w}_{jk}^{-1} f_{\theta_j}^{-1}(f_{\theta_j}(\tilde{w}_{jk} \tilde{z}_{ik})). \quad (22)
\]

That is, \( \tilde{z}_{ik} \) is identifiable up to coordinate-wise transformation.

If \( f_{\theta_j} \) is not invertible, then there exists bijective functions \( g : \mathbb{R} \to \mathbb{R}, h : \mathbb{R} \to \mathbb{R} \) such that \( \tilde{w}_{jk} \tilde{z}_{ik} = g(\tilde{w}_{jk} \tilde{z}_{ik}) \) and the equality is satisfied

\[
f_{\theta_j}(\tilde{w}_{jk} \tilde{z}_{ik}) = f_{\theta_j}(h(g(\tilde{w}_{jk} \tilde{z}_{ik}))) \quad (as \ f_{\theta_j} \ is \ not \ bijective) \quad (23)
\]

\[
f_{\theta_j}(\tilde{w}_{jk} \tilde{z}_{ik}) = f_{\theta_j}(\tilde{w}_{jk} \tilde{z}_{ik}). \quad (24)
\]

As we have \( \tilde{z}_{ik} = \tilde{w}_{jk}^{-1} g(\tilde{w}_{jk} \tilde{z}_{ik}) \), \( \tilde{z}_{ik} \) is identifiable up to coordinate-wise transformation.

### A.2 Proof of Theorem 2

To prove the result, we:

1. Approximate the covariance \( \frac{1}{N} \sum_{i=1}^{N} \text{Cov}(x_i) \) using a first order Taylor approximation.

2. Show that an anchor feature for factor \( k \) always has larger covariance with another anchor feature for \( k \) than other non-anchor features. Specifically: for any \( k \in \{1, \ldots, K\} \) and \( j = \text{anchor for } k \), we have

   * \( \sum_{i=1}^{N} \text{Cov}(x_{ij}, x_{il}) = C_{kk} w_{jk}^2 \sum_{l=1}^{N} B_{ljk}^2 \) for all \( l = \text{anchor for } k \),

   * \( \sum_{i=1}^{N} \text{Cov}(x_{ij}, x_{il}) < C_{kk} w_{jk}^2 \sum_{l=1}^{N} B_{ljk}^2 \) for all \( l \neq \text{anchor for } k \).
3. Apply Theorem 1 of (Bing et al., 2020) to prove that the anchor features can be identified from the covariance of the data.

**Regularity condition.** We assume: for all \( z_i \) in a neighborhood of \( \hat{z}_i = E[z_i] \),

\[
\frac{1}{N} \sum_{i=1}^{N} \mu(z_i) = \frac{1}{N} \sum_{i=1}^{N} \hat{z}_i + \left. \frac{\partial \mu(z_i)}{\partial z_i} \right|_{z_i=\hat{z}_i} (z_i - \hat{z}_i) + o_P(1)
\]

(25)

where \( o_P(1) \) denotes a random variable converging to 0 in probability.

**Step 1.** Here is the marginal covariance of the Sparse VAE:

\[
\text{Var}(x_i) = E[\text{Var}(x_i|z_i)] + \text{Var}(E[x_i|z_i]) = \Sigma + \text{Var}(\mu(z_i))
\]

(26)

(27)

Take the first order Taylor expansion \( \mu(z_i) \approx \hat{z}_i + \left. \frac{\partial \mu(z_i)}{\partial z_i} \right|_{z_i=\hat{z}_i} (z_i - \hat{z}_i) \).

Then

\[
\text{Var}(\mu(z_i)) \approx \left. \frac{\partial \mu(z_i)}{\partial z_i} \right|_{z_i=\hat{z}_i} \text{Var}(z_i) \left( \left. \frac{\partial \mu(z_i)}{\partial z_i} \right|_{z_i=\hat{z}_i} \right)^T.
\]

(28)

Now, \( \mu(z_i) = \{ f_{\theta_j}(w_j \odot z_i) \}^{G_j}_{j=1} \). Then, for any two features, \( j \) and \( l \), we have

\[
\text{Cov}(x_{ij}, x_{il}) \approx \sum_{k=1}^{K} \sum_{k'=1}^{K} \left. \frac{\partial f_{\theta_j}(w_j \odot z_i)}{\partial z_{ik}} \right|_{z_i=\hat{z}_i} \left. \frac{\partial f_{\theta_l}(w_l \odot z_i)}{\partial z_{ik'}} \right|_{z_i=\hat{z}_i} C_{kk'}
\]

(29)

where \( \text{Var}(z_i) = C \).

**Step 2.** We re-write \( f_{\theta_j}(\cdot) \) to expose the first layer of the neural network (before the activation function is applied):

\[
f_{\theta_j}(w_j \odot z_i) = g_{\theta_j}(u_{ij})
\]

(30)

where \( u_{ij} = \{ u_{ijd} \}_{d=1}^{D_j} \), with

\[
u_{ijd} = \sum_{k=1}^{K} H_{dk}^{(1)} w_{jk} z_{ik},
\]

(31)

where \( \{ H_{dk}^{(1)} \}_{d,k=1}^{D_j,K} \) are the weights of the first neural network layer with dimension \( D \).

Then,

\[
\text{Cov}(x_{ij}, x_{il}) = \sum_{k=1}^{K} \sum_{k'=1}^{K} w_{jk} w_{lk} \left( \sum_{d=1}^{D_j} H_{dk}^{(1)} \frac{\partial g_{\theta_j}(u_{ijd})}{\partial u_{ijd}} \right) \left( \sum_{d'=1}^{D_l} H_{dk'}^{(1)} \frac{\partial g_{\theta_l}(u_{ild})}{\partial u_{ild}} \right) C_{kk'}
\]

(32)

\[= \sum_{k=1}^{K} \sum_{k'=1}^{K} w_{jk} w_{lk} B_{ijk} B_{lk'} C_{kk'},
\]

(33)

where \( B_{ijk} = \sum_{d=1}^{D_j} H_{dk}^{(1)} \frac{\partial g_{\theta_j}(u_{ijd})}{\partial u_{ijd}} \).
Suppose $j$ is an anchor feature for $k$. Then the absolute covariance of feature $j$ and any other feature $l$ is:

$$|\text{Cov}(x_{ij}, x_{il})| = |w_{jk}B_{ijk} \sum_{k'=1}^{K} w_{lkk'}C_{kk'}|$$  \hspace{1cm} (34)

$$\leq |C_{kk}w_{jk}B_{ijk} \sum_{k'=1}^{K} w_{lkk'}| \text{ by A3.}$$  \hspace{1cm} (35)

$$\leq C_{kk}w_{jk}^2B_{ijk}^2 \text{ by A2.}$$  \hspace{1cm} (36)

**Step 3.** In Step 2, we proved the equivalent of Lemma 2 of (Bing et al., 2020), adapted for the Sparse VAE. This allows us to apply Theorem 1 of (Bing et al., 2020), which proves that the anchor features can be determined uniquely from the approximate covariance matrix, $\frac{1}{N} \sum_{i=1}^{N} \text{Cov}(x_i)$, as $N \to \infty$.

### A.3 Discussion of identifiability assumptions

In this section, we examine the suitability of assumption A2. We do so by showing A2 holds for a three-layer neural network with ReLU activation functions, independently distributed Gaussian weights and no bias terms. Specifically:

$$\mathbb{E} [x_{ij} | w_j, z_l] = H_{j}^{(3)} \text{ReLU}(H_{j}^{(2)} \text{ReLU}(H_{j}^{(1)}(w_j \odot z_l)))$$  \hspace{1cm} (37)

where $H_{j}^{(1)} \in \mathbb{R}^{D \times K}$ are the weights for layer 1 with $D$ hidden units, $H_{j}^{(2)} \in \mathbb{R}^{D \times D}$ are the weights for layer 2, and $H_{j}^{(3)}$ denotes the $j$th row of the weights for layer 3, $H_{j}^{(3)} \in \mathbb{R}^{G \times D}$.

We have

$$\frac{\partial \mu(z_i)}{\partial z_{ik}} = \sum_{d_1=1}^{D} \sum_{d_2=1}^{D} H_{j,d_2}^{(3)} H_{d_2,d_1}^{(2)} H_{d_1,k}^{(1)} \mathbb{1}[H_{d_2}^{(2)} \text{ReLU}(H_{j}^{(1)}(w_j \odot z_l)) > 0] \mathbb{1}[\sum_{k=1}^{K} H_{d_1,k}^{(1)} w_{jk} z_{ik} > 0].$$  \hspace{1cm} (38)

where $\mathbb{1}(\cdot)$ is the indicator function.

Assume $\{w_{jk}\}_{j,k=1}^{G,K}$ are independent and symmetric around zero. Assume all weights are independent and distributed as: $H_{j,d_1,d_2}^{(m)} \overset{i.i.d.}{\sim} N(0, \tau)$ for all layers $m = 1, 2, 3$.

Taking the first order Taylor approximation, for any two features, $j$ and $l$, we have

$$\text{Cov}(x_{ij}, x_{il}) \approx \sum_{k=1}^{K} \sum_{k'=1}^{K} C_{kk'} \sum_{d_1=1}^{D} \sum_{d_2=1}^{D} \sum_{d_1'=1}^{D} \sum_{d_2'=1}^{D} (H_{j,d_2}^{(3)} H_{d_2,d_1}^{(2)} H_{d_1,k}^{(1)} w_{jk} (H_{d_1,d_2'}^{(3)} H_{d_2',d_1'}^{(2)} H_{d_1',k'}^{(1)} W_{lk'}))$$  \hspace{1cm} (39)

$$\times \mathbb{1}[H_{d_2}^{(2)} \text{ReLU}(H_{j}^{(1)}(w_j \odot z_l)) > 0] \mathbb{1}[\sum_{k=1}^{K} H_{d_1,k}^{(1)} w_{jk} z_{ik} > 0].$$  \hspace{1cm} (40)

$$\times \mathbb{1}[H_{d_2'}^{(2)} \text{ReLU}(H_{l}^{(1)}(w_l \odot z_l)) > 0] \mathbb{1}[\sum_{k=1}^{K} H_{d_1',k}^{(1)} w_{lk} z_{ik} > 0].$$  \hspace{1cm} (41)

As the neural network weights are independent Gaussians, we keep only the terms where $d_1 = d_1'$.
and \( d_2 = d_2' \):

\[
\text{Cov}(x_{ij}, x_{il}) \approx \sum_{d_2=1}^{D} H_{j,d_2}^{(3)} H_{l,d_2}^{(3)} \sum_{d_1=1}^{D} (H_{d_2,d_1}^{(2)})^2 \sum_{k=1}^{K} \sum_{k'=1}^{K} C_{kk'} H_{d_1,k}^{(1)} H_{d_1,k'}^{(1)} w_{jk} w_{lk} \tag{42}
\]

\[
\times \left[ H_{d_2}^{(2)} \text{ReLU}(H^{(1)}(w_j \odot \bar{z}_i)) > 0 \right] \left[ \sum_{k=1}^{K} H_{d_1,k}^{(1)} w_{jk} \bar{z}_{ik} > 0 \right] \quad (43)
\]

\[
\times \left[ H_{d_2}^{(2)} \text{ReLU}(H^{(1)}(w_l \odot \bar{z}_i)) > 0 \right] \left[ \sum_{k=1}^{K} H_{d_1,k}^{(1)} w_{lk} \bar{z}_{ik} > 0 \right]. \tag{44}
\]

If \( j \) and \( l \) are both anchor features for factor \( k \), we have:

\[
\text{Cov}(x_{ij}, x_{il}) \approx \sum_{d_2=1}^{D} (H_{j,d_2}^{(3)})^2 \sum_{d_1=1}^{D} (H_{d_2,d_1}^{(2)})^2 C_{kk} (H_{d_1,k}^{(1)})^2 w_{jk}^2 \tag{45}
\]

\[
\times \left[ H_{d_2}^{(2)} \text{ReLU}(H^{(1)}(w_j \odot \bar{z}_i)) > 0 \right] \left[ \sum_{k=1}^{K} H_{d_1,k}^{(1)} w_{jk} \bar{z}_{ik} > 0 \right], \tag{46}
\]

as \( H_{j,d_2}^{(3)} = H_{l,d_2}^{(3)} \) and \( w_{jk} = w_{lk} \) by definition of an anchor feature.

Meanwhile, if \( j \) and \( l \) are not anchor features for the same factor, we have

\[
\text{Cov}(x_{ij}, x_{il}) \approx \sum_{d_2=1}^{D} H_{j,d_2}^{(3)} H_{l,d_2}^{(3)} A_{d_2} \tag{47}
\]

where

\[
A_{d_2} = \sum_{d_1=1}^{D} (H_{d_2,d_1}^{(2)})^2 \sum_{k=1}^{K} \sum_{k'=1}^{K} C_{kk'} H_{d_1,k}^{(1)} H_{d_1,k'}^{(1)} w_{jk} w_{lk} \left[ H_{d_2}^{(2)} \text{ReLU}(H^{(1)}(w_j \odot \bar{z}_i)) > 0 \right] \tag{48}
\]

\[
\times \left[ \sum_{k=1}^{K} H_{d_1,k}^{(1)} w_{jk} \bar{z}_{ik} > 0 \right] \left[ H_{d_2}^{(2)} \text{ReLU}(H^{(1)}(w_l \odot \bar{z}_i)) > 0 \right] \left[ \sum_{k=1}^{K} H_{d_1,k}^{(1)} w_{lk} \bar{z}_{ik} > 0 \right]. \tag{49}
\]

As the weights are independent, we have that \( H_{j,d_2}^{(3)}, H_{l,d_2}^{(3)}, A_{d_2} \) are independent. For large \( D \), we then have

\[
\sum_{d_2=1}^{D} H_{j,d_2}^{(3)} H_{l,d_2}^{(3)} A_{d_2} \to 0. \tag{50}
\]

Hence, for two anchor features \( j \) and \( l \) and non-anchor feature \( m \), we have \( \text{Cov}(x_{ij}, x_{il}) > \text{Cov}(x_{ij}, x_{im}) \).

**B Inference**

The ELBO is derived from the marginal likelihood as:

\[
\log p_\theta(x_i \mid W) = \log \int p_\phi(x_i \mid z_i, W) p(z_i) p(W \mid \Gamma) p(\Gamma \mid \eta) \, dz_i \, d\Gamma \, d\eta
\]

\[
= \log \int \left( \int p(W \mid \Gamma) p(\Gamma \mid \eta) \, d\Gamma \, d\eta \right) \frac{p_\phi(x_i \mid z_i, W) p(z_i) q(z_i \mid x)}{q(z_i \mid x)} \, dz
\]

\[
\geq \mathbb{E}_{q_\phi(z_i \mid x_i)} \left[ \log p_\theta(x_i \mid z_i, W) \right] - D_{KL}(q_\phi(z_i \mid x_i) \mid \mid p(z_i)) + \mathbb{E}_{\Gamma \mid W^{(i)}, \eta^{(i)}} \left[ \log \left[ p(W \mid \Gamma) p(\Gamma \mid \eta) p(\eta) \right] \right]
\]

\[
= \mathcal{L}_i(\theta, \phi, W, \eta). \tag{51}
\]
The final term of the ELBO in Eq. 51 is:

\[
\mathbb{E}_{\Gamma|W^{(t)},\eta^{(t)}} \left[ \log [p(W|\Gamma)p(\Gamma)p(\eta)] \right] = \sum_{k=1}^{K} \sum_{j=1}^{G} \lambda^*(w_{jk}, \eta_k) |w_{jk}| + \sum_{k=1}^{K} \left[ \sum_{j=1}^{G} \mathbb{E}[y_{jk}|w_{jk}^{(t)}, \eta_k^{(t)}] + a - 1 \right] \log \eta_k
\]

\[
+ \left[ G - \sum_{j=1}^{G} \mathbb{E}[y_{jk}|w_{jk}^{(t)}, \eta_k^{(t)}] + b - 1 \right] \log (1 - \eta_k),
\]

where

\[
\mathbb{E}[y_{jk}|w_{jk}, \eta_k] = \frac{\eta_k \psi_1(w_{jk})}{\eta_k \psi_1(w_{jk}) + (1 - \eta_k) \psi_0(w_{jk})}
\]

\[
\lambda^*(w_{jk}, \eta_k) = \lambda_1 \mathbb{E}[y_{jk}|w_{jk}, \eta_k] + \lambda_0 (1 - \mathbb{E}[y_{jk}|w_{jk}, \eta_k]).
\]

As described in Algorithm 1, we alternate between updating \( \mathbb{E}[y_{jk}|w_{jk}, \eta_k] \) and \( \eta_k \), and taking gradient steps for \( \theta, \phi \) and \( W \).

C Details of empirical studies

C.1 Dataset details and preprocessing

**PeerRead** (Kang et al., 2018). We discard any word tokens that appear in fewer than about 0.1% of the abstracts and in more than 90% of the abstracts. From the remaining word tokens, we consider the \( G = 500 \) most used tokens as features. The observations are counts of each feature across \( N \approx 11,000 \) abstracts.

**Semi-synthetic PeerRead** The training and testing data are distributed differently because we vary the correlations among the latent factors that generate the features across the two datasets. This data was generated as follows: (i) we applied Latent Dirichlet Allocation (LDA, Blei et al., 2003) to the PeerRead dataset using \( K = 20 \) components to obtain factors \( \theta \in \mathbb{R}^{N \times K} \) and topics (loadings) \( \beta \in \mathbb{R}^{K \times G} \). (ii) We created train set factors, \( \theta_{tr} \), by dropping the last \( K/2 \) columns of \( \theta \) and replacing them with columns calculated as: \( \log \theta_{k} = \log \theta_{k} + N(0, \sigma^2) \) for each of the first \( K/2 \) latent dimension \( k \). We fix the test set factors as \( \theta_{te} = \theta \).

**MovieLens** (Harper and Konstan, 2015). We consider the MovieLens 25M dataset. Following Liang et al. (2018), we code ratings four or higher as one and the remaining ratings as zero. We retain users who have rated more than 20 movies. We keep the top \( G = 300 \) most rated movies. Finally, we randomly subsample \( N = 100,000 \) users.

**Zeisel** (Zeisel et al., 2015). We first processed the data following Zeisel et al. (2015). Next, we normalized the gene counts using quantile normalization (Bolstad et al., 2003; Bolstad, 2018). Finally, following Lopez et al. (2018), we retained the top \( G = 558 \) genes with the greatest variability over cells.

C.2 Sparse VAE settings

- For all experiments, the Sparse VAE takes the \( \eta_k \) prior hyperparameters to be \( a = 1, b = G \), where \( G \) is the number of observed features.

- For experiments with Gaussian loss, the prior on the error variance is:

\[
\sigma_j^2 \sim \text{Inverse-Gamma}(\nu/2, \nu \xi/2).
\]
We set $\nu = 3$. The hyperparameter $\xi$ is set to a data-dependent value. Specifically, we first calculate the sample variance of each feature, $x_j$. Then, we set $\xi$ such that the 5% quantile of the sample variances is the 90% quantile of the Inverse-Gamma prior. The idea is: the sample variance is an overestimate of the true noise variance. The smaller sample variances are assumed to correspond to mostly noise and not signal – these variances are used to calibrate the prior on the noise.

C.3 Experimental settings

- For stochastic optimization, we use automatic differentiation in PyTorch, with optimization using Adam (Kingma and Ba, 2015) with default settings (beta1=0.9, beta2=0.999)
- For LDA and NMF, we used Python’s sklearn package with default settings.
- The dataset-specific experimental settings are in Table 4.

C.4 Compute

- GPU: NVIDIA TITAN Xp graphics card (24GB).
- CPU: Intel E4-2620 v4 processor (64GB).
Table 4: Settings for each experiment.

|                        | Synthetic data                              | MovieLens                                |
|------------------------|---------------------------------------------|------------------------------------------|
| **Settings**           | **Value**                                   | **Value**                                |
| # hidden layers        | 3                                           | 3                                        |
| # layer dimension      | 50                                          | 300                                      |
| Latent space dimension | 5                                           | 30                                      |
| Learning rate          | 0.01                                        | 0.0001                                   |
| Epochs                 | 200                                         | 100                                     |
| Batch size             | 100                                         | 100                                     |
| Loss function          | Gaussian                                    | Softmax                                  |
| Sparse VAE             | $\lambda_1 = 1, \lambda_0 = 10$            | $\lambda_1 = 1, \lambda_0 = 10$        |
| $\beta$-VAE            | [2, 4, 6, 8, 16]                             | [2, 4, 6, 8, 16]                         |
| VSC                    | $\alpha = 0.01$                             | $\alpha = 0.01$                         |
| Runtime per split      | CPU, 2 mins                                 | GPU, 1 hour                              |

|                        | PeerRead                                    | Semi-synthetic PeerRead                  |
|------------------------|---------------------------------------------|------------------------------------------|
| **Settings**           | **Value**                                   | **Value**                                |
| # hidden layers        | 3                                           | 3                                        |
| # layer dimension      | 100                                         | 50                                       |
| Latent space dimension | 20                                          | 20                                       |
| Learning rate          | 0.01                                        | 0.01                                     |
| Epochs                 | 40                                          | 50                                       |
| Batch size             | 128                                         | 128                                      |
| Loss function          | Softmax                                     | Softmax                                  |
| Sparse VAE             | $\lambda_1 = 0.001, \lambda_0 = 5$        | $\lambda_1 = 0.01, \lambda_0 = 5$      |
| $\beta$-VAE            | [2, 4, 6, 8, 16]                             | [2, 4, 6, 8, 16]                         |
| VSC                    | $\alpha = 0.01$                             | $\alpha = 0.01$                         |
| Runtime per split      | GPU, 20 mins                                | GPU, 30 mins                             |

|                        | Zeisel                                      |
|------------------------|---------------------------------------------|
| **Settings**           | **Value**                                   |
| # hidden layers        | 3                                           |
| # layer dimension      | 100                                         |
| Latent space dimension | 15                                          |
| Learning rate          | 0.01                                        |
| Epochs                 | 100                                         |
| Batch size             | 512                                         |
| Loss function          | Gaussian                                    |
| Sparse VAE             | $\lambda_1 = 1, \lambda_0 = 10$            |
| VSC                    | $\alpha = 0.01$                             |
| Runtime per split      | CPU, 15 mins                                |