Benchmarks, Algorithms, and Metrics for Hierarchical Disentanglement

Andrew Slavin Ross [1] **Finale Doshi-Velez** [1]

Abstract

In representation learning, there has been recent interest in developing algorithms to disentangle the ground-truth generative factors behind a dataset, and metrics to quantify how fully this occurs. However, these algorithms and metrics often assume that both representations and ground-truth factors are flat, continuous, and factorized, whereas many real-world generative processes involve rich hierarchical structure, mixtures of discrete and continuous variables with dependence between them, and even varying intrinsic dimensionality. In this work, we develop benchmarks, algorithms, and metrics for learning such hierarchical representations.

1. Introduction

Autoencoders aim to learn structure in data by compressing it to a lower-dimensional representation with minimal loss of information. Although they have proven useful in many applications (LeCun et al., 2015), the individual dimensions of their representations are often inscrutable, even when the underlying data is generated by simple processes. Motivated by needs for interpretability (Alvarez-Melis & Jaakkola, 2018; Marx et al., 2019), fairness (Creager et al., 2019), and generalizability (Bengio et al., 2013), as well as a basic intuition that representations should model the data correctly, a subfield has emerged which applies representation learning algorithms to synthetic datasets and checks how well representation dimensions "disentangle" the known ground-truth factors behind the dataset.

Perhaps the most common disentanglement approach has been to learn continuous vector representations whose dimensions are statistically independent (and evaluate them using metrics that assume ground-truth factors are also independent), reasoning that factorization is a useful proxy (Ridgeway, 2016; Higgins et al., 2017; Chen et al., 2018; Kim & Mnih, 2018). However, this problem is not identifiable (Locatello et al., 2018), and it seems unlikely that continuous, factorized, fixed-dimensional representations are the optimal choice for modeling many real-world generative processes, which are often highly structured, with nested parameters that only become active in particular cases.

As a concrete example, consider the problem of learning representations of medical phenotypes of patients with and without diabetes mellitus, a complex disease with multiple types and subtypes (American Diabetes Association, 2005). Some underlying factors of phenotype variation—as well as the intrinsic complexity of these variations—are likely specific to the disease, its types, or its subtypes (Ahlqvist et al., 2018). A representation that faithfully modeled the true factors of variation would need to be deeply hierarchical, with some dimensions only active for certain subtypes. Ideally, it also should also be possible to learn such representations even if these subtypes (and the number of dimensions needed to parameterize them) are unknown.

Our approach in this paper is ambitious: we introduce (1) a flexible framework for modeling deep hierarchical structure in datasets, (2) novel algorithms for learning both structure and structured autoencoders entirely from data, which we apply to (3) novel benchmark datasets, and evaluate with (4) novel hierarchical disentanglement metrics. Our framework is based on the idea that data may lie on multiple manifolds with different intrinsic dimensionalities, and that certain (nested groups of) dimensions may only be active for a subset of the data. Though at first glance this approach seems it should worsen, not improve, identifiability, our structure assumptions also serve as an inductive bias that empirically help us learn representations that more faithfully (and explicitly) model ground-truth generative processes.

2. Related Work

In this section, we review work related to our notion of "hierarchical disentangled representations." However, there are many notions of hierarchy that can be introduced into representations (or into definitions of disentanglement), some of which have little in common except a shift in focus away from flatness or factorization.

Still, the problem of learning a flat, factorized representation has received significant attention over the years. Much of the

[1]Harvard University, Cambridge, MA, USA. Correspondence to: Andrew Slavin Ross <andrew_ross@g.harvard.edu>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021.

initial work, e.g. from Schmidhuber (1992), Zemel (1994), and Comon (1994), was motivated by classic problems like source separation or biological and information-theoretic arguments about minimum description length (Barlow, 1961). More recently, Ridgeway (2016) argued that factorization was often a useful real-world proxy for disentanglement in the seminal sense of Bengio (2013), which motivated the development of a number of popular methods for training variational autoencoders (VAEs, Kingma & Welling (2013)) to reconstruct data from compressed flat vectors, but with minimal total correlation (TC) between their components (Higgins et al., 2017; Chen et al., 2018; Kim & Mnih, 2018; Dupont, 2018; Kim et al., 2019; Jeong & Song, 2019). We build on these approaches in our work, which we also tie back to some of their original motivating problems like minimum description length (see §8.1).

There are, however, a number of limitations to learning factorized representations. To begin with, the problem was actually shown by Locatello et al. (2018) to be non-identifiable, at least without weak supervision (Locatello et al., 2020a; Klindt et al., 2021). More pressingly, though, factorization sometimes *prevents* us from learning representations that disentangle independent causal mechanisms with nontrivial structure (Parascandolo et al., 2018; Träuble et al., 2020), which is actually how Bengio (2013) defined the challenge of disentanglement. Our goal in this work is to learn representations that can identify and explicitly model this kind of structure when it exists.

Still, there are a wide variety of ways to incorporate structure into representations or disentanglement. One is simply to change the disentanglement objective, e.g. to encourage different degrees of factorization within and across subgroups (Esmaeili et al., 2018). Another is to change the representation architecture such that "low-level" components are drawn conditionally on "high-level" components from some fixed hierarchy or graphical model (Sønderby et al., 2016; Siddharth et al., 2017; Singh et al., 2019). Others use mixed discrete-continuous representations where continuous representation components are either "global" (marginally independent) or "local" to a specific categorical value (conditionally independent, and sometimes inactive when the categorical takes other values) (Sorrenson et al., 2020; Choi et al., 2020). Typically, though, these approaches only support shallow hierarchies that must be specified by the user in advance, or require instance-level supervision (Yang et al., 2020). Our work is closest to the global-local approach of Choi et al. (2020), but we support arbitrarily deep hierarchies, and also learn them from data.

Other related approaches not directly in this line of research include relational autoencoders (Wang et al., 2014), which model structure between non-iid flat data, and graph neural networks (Defferrard et al., 2016), which learn flat representations of structured data. In contrast, we model structure *within* flat inputs. Also relevant are advances in object representations, such as slot attention (Locatello et al., 2020b). While this area has generally not focused on hierarchically nested objects, it does learn structure and seamlessly handles sets; we view our method as complementary. Finally, our hierarchy detection method is closely related to work in manifold learning. We build on work in multiple- and robust manifold learning (Mahapatra & Chandola, 2017; Mahler, 2020), contributing new innovations on top of them.

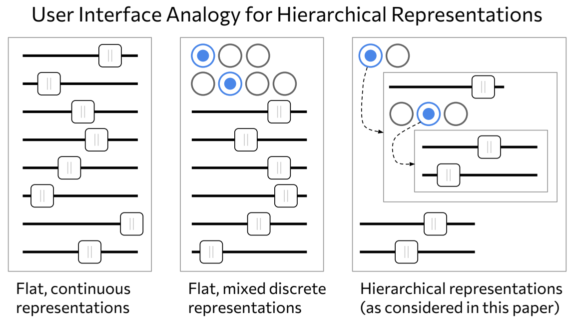

Figure 1. User interface analogy for our model of representation hierarchy. In traditional flat representations, all dimensions are active simultaneously, no matter whether they are continuous (shown as sliders) or discrete (shown as radio buttons). In our model, groups of dimensions may only be active for specific values of discrete dimensions, and groups can be nested.

3. Hierarchical Disentanglement Framework

In this section, we outline our framework for modeling hierarchical structure in representations. In our framework, we associate individual data points with paths down a *dimension hierarchy* (examples in Fig. 2). Dimension hierarchies consist of dimension group nodes (shown as boxes), each of which can have any number of continuous dimensions (shown as ovals) and an optional categorical variable (diamonds) that leads to other groups based on its value. For any data point, we "activate" only the dimensions along its corresponding path. Notation-wise, $\texttt{root}(Z)$ denotes the group at the root of a hierarchy, and $\texttt{children}(Z_j)$ denotes the child groups of a categorical dimension Z_j. In the context of a dataset, for a dimension Z_j or a dimension group g, $\texttt{on}(Z_j)$ or $\texttt{on}(g)$ denotes the subset of the dataset where that Z_j or g is active.

As a potentially more intuitive analogy (as well as a visualization method), we can also understand hierarchical representations in terms of user interfaces with nested groups of sliders and radio buttons (Fig. 1). While traditional representations might consist of a single group of constantly visible sliders (or a mixture of sliders and radio buttons), hierarchical representations contain subgroups that only ap-

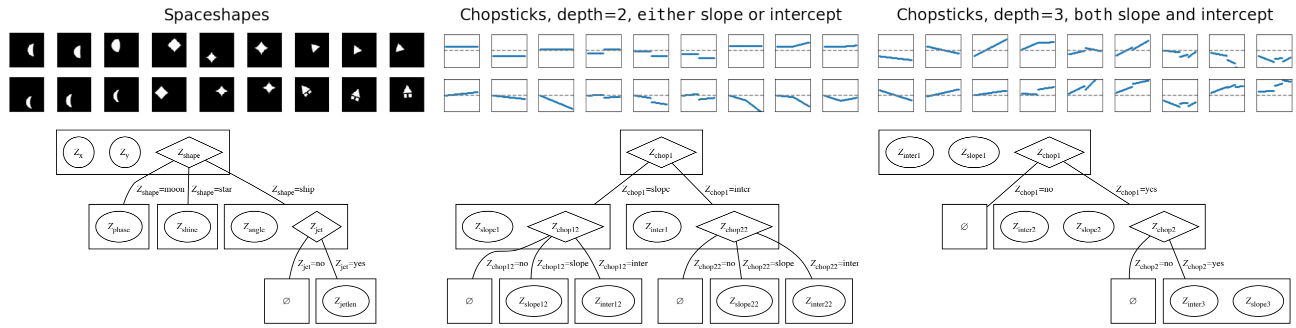

Figure 2. Examples and ground-truth variable hierarchies for Spaceshapes and two different variants of Chopsticks. Continuous variables are shown as circles and discrete variables are shown as diamonds. Discrete variables have subhierarchies of additional variables that are only active for particular discrete values. See also Fig. 1-style interactive visualizations of hierarchical representations explicitly trained to match ground-truth for each dataset respectively.

pear when parent radio buttons take particular values. For such representations, only a subset of dimensions need to be visible to users at any given time, even if many are required to model the dataset—which could significantly improve interpretability (Ross et al., 2021), if the hierarchy itself is comprehensible.

Although in this work we only consider tree-structured hierarchies, our framework could be extended to support multiple categoricals per node or even DAGs, such that instances can be associated with directed flows down multiple paths.

4. Hierarchical Disentanglement Benchmarks

In this section, both to clarify our framework and enable testing of our algorithms, we introduce several synthetic benchmark datasets with ground-truth hierarchical structure (see Fig. 2 for instances and dimension hierarchies).

4.1. Spaceshapes

Our first benchmark dataset is Spaceshapes, a binary 64x64 image dataset meant to hierarchically extend dSprites, a shape dataset common in the disentanglement literature (Matthey et al., 2017). Like dSprites, Spaceshapes images have location variables `x` and `y`, as well as a categorical `shape` with three options (in our case, `moon`, `star`, and `ship`). However, depending on shape, we add other continuous variables with qualitatively different effects: `moons` have a `phase`; `stars` have a sharpness to their `shine`; and `ships` have an `angle`. Finally, `ships` can optionally have a `jet`, which has a length (`jetlen`), but this is only defined at the deepest level of the hierarchy. The presence of `jetlen` alters the intrinsic dimensionality of the representation; it can be either 3D or 4D depending on the path. As in dSprites, variables are sampled from continuous or discrete uniforms. An interactive visualization of a representation trained to model this ground-truth hierarchy can be viewed here.

4.2. Chopsticks

Our second benchmark, Chopsticks, is actually a family of arbitrarily deep timeseries datasets. Chopsticks samples are 64D linear segments, each of which can have a uniform-sampled `slope` and/or `intercept`; different dataset variants can have one, the other, `both`, or `either` but not both. For all variants, segments initially span the whole interval. However, we then flip a coin to determine whether to `chop` the segment, in which case we add a uniform offset to the slope and/or intercept of the right half. We repeat this process recursively up to a configurable maximum `depth`, biasing probabilities so that we have equal probability of stopping at each level; each chop requires increasing local dimensionality to track additional slopes and intercepts. Although the underlying process is quite simple, the structure can be made arbitrarily deep, making it a useful benchmark for testing structure learning. We provide more details in §A.2, and interactive visualizations are also available for the depth-2 `either` and depth-3 `both` variants.

Although these datasets are designed to have clear hierarchical structure, there are certain ambiguities in how to structure aspects of the dimension hierarchies, which we discuss in §6.1.

5. Hierarchical Disentanglement Algorithms

We next present a method for learning hierarchical disentangled representations from data alone. We split the problem into two brunch-themed algorithms, MIMOSA (which infers hierarchies) and COFHAE (which trains autoencoders).

5.1. Learning Hierarchies with MIMOSA

The goal of our first algorithm, MIMOSA (**M**ulti-manifold **I**so**M**ap **O**n **S**mooth **A**utoencoder), is to learn a hierarchy $\hat{H}$ from data, as well as an assignment vector $\hat{A}_n$ of data points

Algorithm 1 MIMOSA(X)

1: Encode the data X using a smooth autoencoder to reduce the initial dimensionality. Store as Z.
2: Construct a neighborhood graph on Z using a Ball Tree (Omohundro, 1989).
3: Run LocalSVD (Algorithm 3) on each point in Z and its neighbors to identify local manifold directions.
4: Run BuildComponent (Algorithm 5) to successively merge neighboring points with similar local manifold directions.
5: Run MergeComponents (Algorithm 6) to combine similar components over longer distances and discard outliers.
6: Run ConstructHierarchy (Algorithm 7) to create a dimension hierarchy based on which components enclose others.
7: **return** the hierarchy and component assignments.

to hierarchy leaves. MIMOSA consists of the following steps (see Appendix for Algorithms 3-7 and complexity, and Fig. 3 for a detailed example):

Dimensionality Reduction (Algorithm 1, line 1): We start by performing an initial reduction of X to Z using a flat autoencoder. While we could start with $Z = X$, performing this reduction saves computation as later steps (e.g. finding neighbors) scale linearly with $|Z|$. Although this requires choosing $|Z|$, we find the exact value is not critical as long as it exceeds the (max) intrinsic dimensionality of the data. We also find it important to use differentiable activation functions (e.g. Softplus rather than ReLU) to keep latent manifolds smooth; see Fig. A.4 for more.

Manifold Decomposition (Algorithms 3-6): We next decompose Z into a set of manifold "components" by computing SVDs locally around each point and merging neighboring points with sufficiently similar subspaces. We then perform a second merging step over longer lengthscales, combining equal-dimensional components with similar local SVDs along their nearest boundary points and discarding small outliers, which we found was necessary to handle interstitial gaps when two manifolds intersect. The core of this step is based on a multi-manifold learning method (Mahapatra & Chandola, 2017), but we make efficiency as well as robustness improvements by combining ideas from RANSAC (Fischler & Bolles, 1981) and contagion dynamics (Mahler, 2020). The merging step is a new contribution.

It bears emphasis that manifold decomposition, which groups points based on the similarity of local principal components, is distinct from clustering, which groups points based on proximity. In the examples we consider, even hierarchical iterative clustering methods like OPTICS (Ankerst et al., 1999) will not suffice, as nearby points may lie on different manifolds.

Hierarchy Identification (Algorithm 7): Finally, we construct a tree by drawing edges from low-dimensional components to the higher-dimensional components that best "enclose" them, which we define using a ratio of inter-component to intra-component nearest neighbor distances; we believe this is novel. We use this tree and the component dimensionalities to construct a dimension hierarchy and a set of assignments from points to paths, which we output.

Hyperparameters: Each of these steps has several hyperparameters, and we provide a full listing and sensitivity study in §A.5. The one we found most critical was the minimum SVD similarity to merge neighboring points.

5.2. Training Autoencoders with COFHAE

Algorithm 2 COFHAE(X)

hierarchy, assignments = MIMOSA(X) # Algorithm 1
$\text{HAE}_\theta = \text{init\_hierarchical\_autoencoder}(\text{hierarchy})$
$D_\psi = \text{init\_discriminator}()$
for $x, a \sim \text{minibatch}(X, \text{assignments})$ **do**
  $a', z = \text{HAE}_\theta.\text{encode}(x; \tau)$ # Algorithm 8
  $x' = \text{HAE}_\theta.\text{decode}(\text{concat}(a', z))$ # normal NN
  $z' = \text{copy}(z)$
  for $i = 1..|z_0|$ **do**
    shuffle $z'_{:,i}$ over minibatch indices where $\text{on}(z_{:,i})$
  end for
  $\mathcal{L}_\theta = \mathcal{L}_x(x', x) + \lambda_1 \mathcal{L}_a(a', a) - \lambda_2 \log \frac{D_\psi(z)}{1-D_\psi(z)}$
  $\mathcal{L}_\psi = -\log D_\psi(z') - \log(1 - D_\psi(z))$
  $\theta = \text{descent\_step}(\theta, \mathcal{L}_\theta)$
  $\psi = \text{descent\_step}(\psi, \mathcal{L}_\psi)$
end for
return HAE_θ

Our first stage, MIMOSA, gives us the hierarchy and assignments of data down it. In the second stage, COFHAE (**CO**nditionally **F**actorized **H**ierarchical **A**uto**E**ncoder, Algorithms 2 and 8), we learn an autoencoder that respects this hierarchy via (differentiable) masking operations that impose structure on flat representations.

Hierarchical Encoding (Algorithm 8): Instances x pass through a neural network encoder to an initial vector z_{pre}, whose dimensions correspond to both continuous and categorical dimensions. We then pass each set of categoricals through a softmax with temperature τ, and use them to recursively mask the entirety of z_{pre} based on the hierarchy. We finally split this masked vector into a continuous vector z and a list of estimated assignments a', outputting both.

Supervising Assignments: Although we lack ground-truth during training, we do have assignments a from MIMOSA (for at least a subset of the dataset). We add a penalty

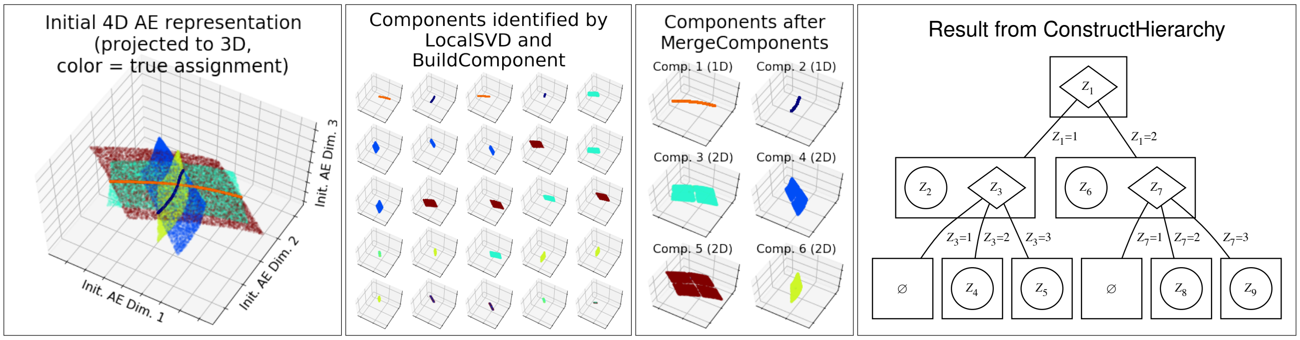

Figure 3. Breakdown of MIMOSA for the depth-2 `either` version of Chopsticks, colored by ground-truth assignments. MIMOSA learns an initial 4D softplus AE representation (left), decomposes it into lower-dimensional components by grouping together neighboring points with similar local SVDs (second from left), merges them over longer distances while discarding outliers (second from right), and finally uses enclosure relationships to infer a hierarchy (right). In this case, correspondence between the assignment of points to learned components vs. ground-truth is very close (99.8% purity, covering 93.7% of the training set, and with no H-error—see §6.2 for definitions of these metrics). Similar examples are shown for other datasets in Figs. A.11-A.16 of the Appendix.

$\mathcal{L}_a(a', a)$, weighted by λ_1, to make encoded a' match a.

Conditional Factorization: Kim & Mnih (2018) penalize the total correlation (TC) between dimensions of flat continuous representations z with two tricks. First, noting that TC is the KL divergence between $q(z)$ (the joint distribution of the encoded z) and $\bar{q}(z) \equiv \prod_{j=1}^{|z|} q(z_j)$ (the product of its marginals), they approximate samples from $\bar{q}(z)$ by randomly permuting the values of each z_i across batches (Arcones & Gine, 1992). Second, they approximate the KL divergence between the two distributions using the density ratio trick (Sugiyama et al., 2012) on an auxiliary discriminator $D_\psi(z)$, where $KL(q(z)||\bar{q}(z)) \approx \log \frac{D_\psi(z)}{1-D_\psi(z)}$ if $D_\psi(z)$ outputs accurate probabilities of z having been sampled from $\bar{q}$. We adopt a similar approach, except instead of permuting each z_i across the full batch $\mathcal{B}$, we only permute it where it is *active*, i.e. $\mathcal{B} \cap \texttt{on}(z_i)$ (defined using the hardened version of the mask). This approximates a hierarchical version of $\bar{q}(z)$ where each dimension distribution is a mixture of 0 and the product of its *active* marginals. $D_\psi(z)$ then lets us estimate the KL between this distribution and $q(z)$, which we penalize and weight with λ_2.

This approach presumes ground-truth continuous variables should be conditionally independent given categorical values, which is a major assumption. However, it is less strict than the assumption taken by many disentanglement methods, i.e. that continuous variables are independent marginally, and it may remain useful as an inductive bias.

6. Hierarchical Disentanglement Metrics

In this section, we develop metrics for quantifying how well learned representations and hierarchies match ground-truth.

6.1. Desiderata and Invariances

Our goal in designing metrics is to measure whether we have learned the "right representation," both in terms of global structure and specific variable correspondences. In an ideal world, we would measure whether a learned representation Z is identically equal to a ground-truth V. However, most existing disentanglement metrics are invariant to permutations, so that dimensions V_i can be reordered to match different Z_j, as well as univariate transformations, so that the values of Z_j do not need to be identical to V_i. In the case of methods like the SAP score (Kumar et al., 2017), these univariate transformations must be linear, but as the uniformity of scaling can be arbitrary, we permit general nonlinear transformations, as long as they are 1:1, or invertible.

Hierarchical representations have an additional ambiguity about the right "vertical" placement of continuous variables. For example, on Spaceshapes, the `phase`, `shine`, and `angle` variables could all be "merged up" to a single top-level variable whose effect changes based on `shape`. Alternatively, `x` and `y` position could be "pushed down" and duplicated for each shape despite their analogous effects (see Fig. A.10 for an illustration). In terms of our user interface analogy from Fig. 1 (or our specific implementation), "merge up" and "push down" transformations correspond to moving sliders into or out of outlined groups, but keeping their effects on the outputs the same, as well as preserving the structure of nested radio buttons. To a user interacting with such representations, they would appear almost identical, except some slider labels might change with radio button settings. Because of this functional near-equivalence, we defer the problem of deciding the most natural vertical placement of continuous variables to future work, and make our main metrics invariant to them.

6.2. MIMOSA Metrics: H-error, Purity, Coverage

The first metric we use to evaluate MIMOSA is **H-error**, which measures whether learned hierarchy $\hat{H}$ has the same essential structure as the ground-truth hierarchy H. We define H-error in terms of the tree edit distance of Zhang & Shasha (1989) (i.e. minimum number of insertions, edits, or deletions to transform H into $\hat{H}$), but between normalized "merged up" representations of each hierarchy; details are in §A.3. This metric is 0 if and only if both hierarchies are the same up to the transformations described in §6.1.

The second MIMOSA metric is **purity**, which measures whether the assignments output by MIMOSA match ground-truth. To compute it, we iterate over each leaf in $\hat{H}$, find the leaf in H to which most of its assigned points belong, and then compute the fraction that belong to the majority. Then we average these scores across $\hat{H}$, weighting by the number of points in each leaf. This metric only falls below 1 if leaves contain points with different ground-truth assignments.

The final metric we use to evaluate MIMOSA is **coverage**. Since MIMOSA discards small outlier components, it is possible that the final set of assignments will not cover the full training set. If almost all points are discarded this way, the other metrics may not be meaningful. As such, we measure coverage as the fraction of the training set which is not discarded. We note that hyperparameters can be tuned to ensure high coverage without knowing ground-truth assignments.

6.3. COFHAE Metrics: R^4 and R^4_c Scores

Per our desiderata, we seek to check whether every ground-truth variable V_i can be mapped invertibly to some learned dimension Z_j. As a preliminary definition, we say that a learned Z_j *corresponds* to a ground-truth V_i over some set $\mathcal{S} \subseteq \mathbb{R}$ if a bijection between them exists; that is,

$$\exists\, f(\cdot) : \mathcal{S} \to \mathbb{R} \text{ s.t. } f(V_i) = Z_j \text{ and } f^{-1}(Z_j) = V_i \quad (1)$$

We say that Z *disentangles* V if all V_i have a corresponding Z_j. To measure the extent to which bijections exist, we can simply try to learn them (over random splits of many paired samples of V_i and Z_j). Concretely, for each pair of learned and true dimensions, we train univariate models to map in both directions, compute their coefficients of determination (R^2), and take their geometric mean:

$$\begin{aligned} f &\equiv \min_{f \in \mathcal{F}} \mathbb{E}_{\text{train}} \left[(f(X) - Y)^2 \right] \\ R^2(X \to Y) &\equiv \mathbb{E}_{\text{test}} \left[1 - \frac{\sum (f(X) - Y)^2}{\sum (\mathbb{E}[Y] - Y)^2} \right] \\ R^2(X \leftrightarrow Y) &\equiv \sqrt{\lfloor R^2(X \to Y) \rfloor_+ \lfloor R^2(Y \to X) \rfloor_+}, \end{aligned} \quad (2)$$

where we average over train/test splits (we use 5), assume $\mathcal{F}$ is sufficiently flexible to contain the optimal bijection (we use gradient-boosted decision trees), and assume our dataset is large enough to reliably identify $f \in \mathcal{F}$. In the limit, $R^2(X \leftrightarrow Y)$ can only be 1 if a bijection exists, as any region of non-zero mass in the joint distribution of X and Y where this is false would imply $\mathbb{E}[(f(X) - Y)^2] > 0$ or $\mathbb{E}[(f(Y) - X)^2] > 0$. In the special case that Y is discrete rather than continuous, we use classifiers for f and accuracy instead of R^2, but the same argument holds.

To measure whether a *set* of variables Z disentangles another *set* of variables V, we check if, for each V_i, there is at least one Z_j for which $R^2(V_i \leftrightarrow Z_j) = 1$:

$$R^4(V, Z) \equiv \frac{1}{|V|} \sum_i \max_j R^2(V_i \leftrightarrow Z_j), \quad (3)$$

We call this the "right-representation" R^2, or R^4 score. Note that this metric is related to the existing SAP score (Kumar et al., 2017), except we allow for nonlinearity, require high R^2 in both directions, and take the maximum over each score column rather than the difference between the top two entries (which avoids assuming ground-truth is factorized).

Although R^4 is useful for measuring correspondence between sets of variables that are both always active, it does not immediately apply to hierarchical representations unless inactive variables are represented somehow, e.g. as 0 (an arbitrary implementation decision that affects R^2 by changing $\mathbb{E}[Y]$). It also lacks invariance to merge-up and push-down operations. Instead, we seek *conditional correspondence* between V_i and a set of dimensions in Z, defined as

$$\begin{aligned} &\text{for all } V_i \in \mathtt{on}(V_i)\, \exists\, \mathcal{Z}_i = \{Z_j, Z_k, \ldots\} \text{ s.t.} \\ &\quad (a)\ V_i \text{ corresponds to } Z_j \text{ over } \mathtt{on}(V_i) \cap \mathtt{on}(Z_j), \\ &\quad (b)\ \mathtt{on}(Z_j) \cap \mathtt{on}(Z_k) = \emptyset \text{ for all } j \neq k, \text{ and} \\ &\quad (c)\ \textstyle\bigcup_{z \in \mathcal{Z}_i} \mathtt{on}(z) = \mathtt{on}(V_i), \end{aligned} \quad (4)$$

or rather that we can find some tiling of $\mathtt{on}(V_i)$ into regions where it corresponds 1:1 with different Z_j which are never active simultaneously. This allows for one Z_j to correspond to non-overlapping elements of V (e.g. merging up), as well as for one V_i to be modeled by non-overlapping elements of Z (e.g. pushing down).

We can then formulate a conditional R^4_c score which quantifies how closely conditional correspondence holds:

$$\begin{aligned} R^2_c(V_i, g) \equiv \max \Bigg(&\max_{j \in g} \Big(R^2\big(V_i \leftrightarrow Z_j \big| \mathtt{on}(V_i) \cap \mathtt{on}(g)\big), \\ &\sum_{g' \in \mathtt{children}(Z_j)} R^2_c(V_i, g') \frac{|\mathtt{on}(V_i) \cap \mathtt{on}(g')|}{|\mathtt{on}(V_i)|} \Big) \Bigg), \end{aligned}$$

for a dimension group g; the overall disentanglement is:

$$R^4_c(V \leftrightarrow Z) \equiv \frac{1}{|V|} \sum_{i=1}^{|V|} R^2_c(V_i, \mathtt{root}(Z)). \quad (5)$$

In the special case that V and Z are flat, R^4_c reduces to R^4. We note that even for flat representations, the R^4 score may be a useful measure of disentanglement when ground-truth variables are not factorized.

7. Experimental Setup

Benchmarks: We ran experiments on nine benchmark datasets: Spaceshapes, and eight variants of Chopsticks (varying `slope`, `intercept`, `both`, and `either` at recursion depths of 2 and 3). See §4 for more details, and Fig. A.1 for preliminary experiments on noisy data.

Algorithms: In addition to COFHAE with MIMOSA, we trained a number of flat baselines. As fully continuous baselines, we trained autoencoders (AE), variational autoencoders (Kingma & Welling, 2013) (VAE), the β-total correlation autoencoder (Chen et al., 2018) (TCVAE), and FactorVAE (Kim & Mnih, 2018). As mixed discrete-continuous baselines, we trained JointVAE (Dupont, 2018) and CascadeVAE (Jeong & Song, 2019), providing them with the ground-truth structure of discrete variables.[1] Finally, we ran COFHAE ablations using the ground-truth hierarchy and assignments, testing all possible combinations of loss terms and comparing conditional vs. marginal TC penalties; results are in Fig. 5. See §A.1 for training and model details.

Metrics: To evaluate hierarchies, we computed purity, coverage, and H-error (§6.2). Results are in Table 1. To measure disentanglement, we primarily use R^4_c (§6.3); results for all datasets and models are in Fig. 4. We also compute the following baseline metrics: the SAP score (Kumar et al., 2017) (SAP), the mutual information gap (Chen et al., 2018) (MIG, estimated using 2D histograms), the FactorVAE score (Kim & Mnih, 2018) (FVAE), and the DCI disentanglement score (Eastwood & Williams, 2018) (DCI). Most implementations were adapted from `disentanglement_lib` (Locatello et al., 2018). We also compute our marginal R^4 score. Results across metrics are shown for a subset of datasets and models in Fig. 6.

Hyperparameters: COFHAE is only given instances X, which complicates cross-validation. However, we can still tune its hyperparameters to ensure assignments a' match MIMOSA outputs a and reconstruction loss for x is low (which fail to can happen if the adversarial term dominates). Over a grid of τ in $\{\frac{1}{2}, \frac{2}{3}, 1\}$, λ_1 in $\{10, 100, 1000\}$, and λ_2 in $\{1, 10, 100\}$, we select the model with the lowest *training* reconstruction loss $\mathcal{L}_x$ from the $\frac{1}{3}$ with the lowest assignment loss $\mathcal{L}_a$. For MIMOSA, hyperparameters can be tuned to ensure high coverage (purity and H-error require side-information); see §A.5 for more details.

[1]Note that CascadeVAE only supports a single categorical variable, but we give it cardinality equal to the total number of paths down the true hierarchy.

For our baselines, we show results at β=5 for TCVAE, γ=10 for FactorVAE, β=1, C_z=C_c=10 for JointVAE, and β_ℓ=2 for CascadeVAE (with other hyperparameters set to the same settings as the original paper). However, we tested each method across a variety of strength and capacity hyperparameters, and show more complete results in Fig. A.7.

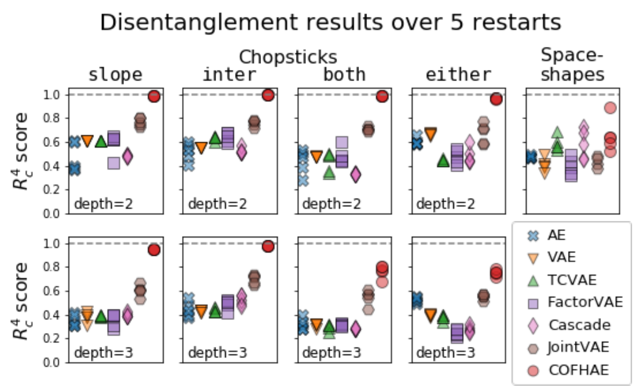

Figure 4. Hierarchical disentanglement results for representation learning methods (baselines and COFHAE + MIMOSA) over all nine datasets. COFHAE almost perfectly disentangles ground-truth on the six simplest versions of Chopsticks, with some degradations on the two most complex versions (with very deep hierarches) and on Spaceshapes (with a shallower hierarchy, but higher-dimensional inputs). Baseline methods were generally much more entangled, though JointVAE, β-TCVAE, and CascadeVAE are competitive in certain cases.

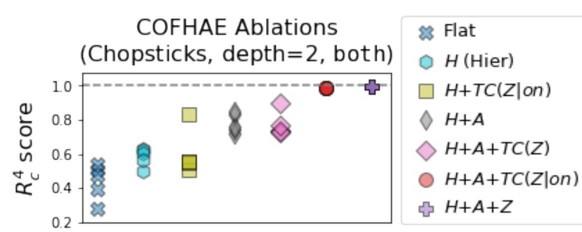

Figure 5. Ablation study for COFHAE on the depth-2 `both` version of Chopsticks (over 5 restarts). Hierarchical disentanglement is low for flat AEs (Flat); adding the ground-truth hierarchy H improves it (Hier H), as does also adding supervision for ground-truth assignments A (H+A). Adding a FactorVAE-style marginal TC penalty (H+A+$TC(Z)$) does not appear to help disentanglement, but making that TC penalty conditional (H+A+$TC(Z|on)$, i.e. COFHAE) brings it close to the near-optimal disentanglement of a hierarchical model whose latent representation is fully supervised (H+A+Z). However, the hierarchical conditional TC penalty fails to produce this same disentanglement without any supervision over assignments (H+$TC(Z)$).

MIMOSA Metric	Chopsticks, depth=2				Chopsticks, depth=3				Space-shapes
	`inter`	`slope`	`both`	`either`	`inter`	`slope`	`both`	`either`	
Purity	1.0±0.0	1.0±0.0	1.0±0.0	1.0±0.0	.98±0.0	.95±0.0	.94±0.0	.93±0.0	1.0±0.0
Coverage	.99±0.0	.99±0.0	.96±0.0	.94±0.0	.98±0.0	.98±0.0	.82±0.01	.75±0.01	1.0±0.0
H-error	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	2.40±0.89	0.0±0.0

Table 1. MIMOSA results across all datasets, with means and standard deviations across 5 restarts. In general, MIMOSA components contained points only from single ground-truth sets of paths (purity), were inclusive of most points in the training set (coverage), and resulting in perfectly accurate hierarchies (H errors), with the greatest or only exception being the Chopsticks depth-3 `either` dataset (where we tended to miss 2-3 of the 8 deepest 3D components).

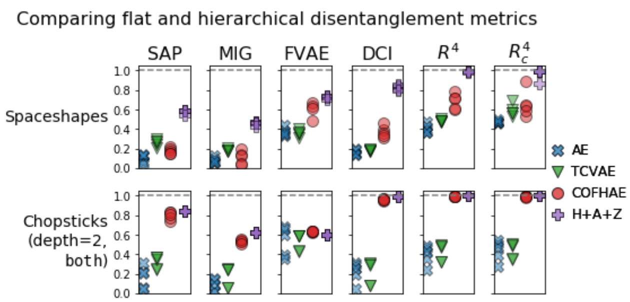

Figure 6. Comparison of disentanglement metrics across two datasets and four models. Only R^4 and R^4_c correctly and consistently award near-optimal scores to the supervised H+A+Z model.

8. Results and Discussion

MIMOSA consistently recovered the right hierarchies. Per Table 1, we consistently found the right hierarchy for all datasets except depth-3 `either`-Chopsticks, but even there results were close, generally recovering 12 out of 14 nodes (see Fig. A.16 for more details). Purity and coverage were also high, often near perfect as in Spaceshapes or depth-2 Chopsticks.

COFHAE significantly outperformed baselines. Per Fig. 4, COFHAE R^4_c scores were near-perfect for 6 out of 9 datasets, and were highest on all (both in terms of mean and maximum). On Spaceshapes and the depth-3 `either` and `both` versions of Chopsticks, scores were slightly worse. Part of this suboptimality could be due to non-identifiability. For Spaceshapes and the `both` versions of Chopsticks, dimension group nodes contain multiple continuous variables, which even conditionally can be modeled by multiple factorized distributions (Locatello et al., 2018). However, optimization issues could also be at fault, as we do not see suboptimal R^4_c on Chopsticks until a depth of 3, and even supervised $H{+}A{+}Z$ models occasionally fail to converge on Spaceshapes. Kim & Mnih (2018) note that the relatively low-dimensional discriminator used by FactorVAE is easier to optimize than the generally high-dimensional discriminators used in GANs, which are notoriously tricky to train (Mescheder et al., 2018). In our case, flattened hierarchy vectors can be high-dimensional (e.g. Fig. A.17), and in any given batch, instances corresponding to different paths down the hierarchy may have different numbers of samples (potentially requiring larger batch sizes or stratified sampling to ensure sufficient coverage). Finally, alongside non-identifiability and optimization issues, MIMOSA errors (e.g. merge-up/push-down differences for Spaceshapes and suboptimal purity and coverage for Chopsticks) also may play a role, as evidenced by performance improvements in our full COFHAE ablations in Fig. A.6. Despite all of these issues, COFHAE is still closer to optimal, at best and on average, than any of our baseline algorithms (even on Spaceshapes, where it is possible for a flat representation to disentangle all features except jet length). We note also that our baselines often performed *worse* with increasing disentanglement penalty strength (Fig. A.7), with the closest COFHAE competitor, JointVAE, achieving its best results at its minimal tested value $\gamma{=}1$ (i.e. equivalent to a normal VAE). These results are consistent with the fact that minimizing marginal rather than conditional TC on these datasets *prevents* models from learning the right representation.

R^4_c provides more insight into disentanglement than baselines. One way to evaluate an evaluation metric is to test it against a precisely known quantity. In this case, we know the $H{+}A{+}Z$ model, whose encoder is supervised to match ground-truth, should receive a near-perfect score. The only metrics to do this consistently are R^4 and R^4_c. Note that the DCI disentanglement score, based on the entropy of normalized feature importances from an estimator predicting single ground-truth factors from all learned dimensions, comes close. Intuitively, this metric could behave similarly to R^4 if its estimator was trained to be sparse (placing importance on as few dimensions as possible). However, using R^2s of univariate estimators is more direct, and also incorporates information from the DCI informativeness score.

Another way to evaluate an evaluation metric is to test whether quantitative differences capture salient qualitative differences. To this point, specifically to compare R^4 and R^4_c, we consider several examples in Fig. A.8 and Fig. A.9. First, we see that for the Spaceshapes COFHAE model in Fig. A.8c (or here), its R^4_c score (0.89) is higher than its R^4 (0.79). This increase is due to the fact that R^4 penalizes "push-down" differences (§6.1) between the learned and true factors representing x and y position, while R^4_c is invariant to them. However, the overall increase is less dramatic than one might expect due to modest decreases in correspondence scores for other dimensions (e.g. $0.98 \to 0.89$ for jetlen), which occur because R^4_c is not biased by spurious equality between dimensions which are both inactive. Another example of a difference between R^4 and R^4_c (illustrating invariance to "merging up" rather than "pushing down") is for the Spaceshapes β-TCVAE in Fig. A.8b. In this case, histograms show that one β-TCVAE variable (Z_3) corresponds closely to both moon phase and star shine (and to a lesser extent, jetlen), only one of which is active at a time. The R^4 score (0.47) assigns low scores to these correspondences, but R^4_c (0.69) properly factors them in.

COFHAE and MIMOSA subcomponents improve performance. Though COFHAE contains many moving parts, results in Fig. 5 and Fig. A.6 suggest they all count. Autoencoders only achieve optimal disentanglement if provided with the hierarchy, assignments, and a conditional (not marginal) penalty on the TC of continuous variables; no partial subset suffices. In the Appendix, Fig. A.5 shows ablations and sensitivity analyses for MIMOSA that validate its subcomponents are important as well.

8.1. Remark on Identifiability and Parsimony

From Locatello et al. (2018), we know all forms of TC minimization permit multiple solutions (though they often improve disentanglement empirically, especially when ground-truth factors are non-Gaussian). However, what about the other components of our method, such as MIMOSA?

MIMOSA does not minimize an objective function, so questions of identifiability might seem moot. However, we could reformulate it as trying to find a *small* set of *low-dimensional* and *bounded-curvature* manifolds that *approximately contain* a *large fraction* of the data. More concretely, we could place penalties or constraints on, e.g., the cardinality, dimensionality, and mean or percentiles of error and principal curvature magnitudes over the set. Such a problem might well be identifiable (up to the transformations discussed in §6.1), though analyzing it is beyond the scope of this work.

However, perhaps a better-motivated formulation that covers both MIMOSA and COFHAE would be to return to minimum description length (MDL)—the same problem that motivated much of the initial research into factorized representations (Barlow, 1961; Zemel, 1994). As an example, assume we are given a dataset of N instances, $\frac{7}{8}$ of which lie on a 1D manifold, and $\frac{1}{8}$ of which lie on an 8D manifold. If we must encode instances as flat vectors of 32-bit floats, those vectors will need to be at least 8D for accurate reconstruction, meaning the dataset's description length will be $8 * 32 * N = 256N$ bits (plus the size of the model, which is negligible for sufficiently large N). However, if we use a disentangled hierarchical representation, we need either 1 or 8 floats to represent each instance (plus a single bit to distinguish between them). In that case, the description length would be $(\frac{1}{8} * 8 * 32) + (\frac{7}{8} * 1 * 32) + 1)N = 61N$ bits, which is minimal (assuming the model is not much larger). The problem of learning factorized representations *within* each manifold might remain non-identifiable, but the MDL argument for doing so remains the same as in Zemel (1994). This example suggests that (disentangled) hierarchical representations might spontaneously emerge as the (partially identifiable) solution to MDL objectives, at least for datasets that lie on multiple manifolds.

9. Conclusion

In this work, we introduced a novel formulation of hierarchical disentanglement, where ground-truth representation dimensions are organized into a tree and activated or deactivated based on the values of categorical dimensions. We presented benchmarks, algorithms, and metrics for learning and evaluating such hierarchical representations.

There are a number of promising avenues for future work. One is extending our methods to handle a wider variety of underlying structures, e.g. dimension DAGs, or integrating our methods with object representation techniques to better model generative processes involving ordinal variables or unordered sets (Locatello et al., 2020b). Another is to better solve or understand hierarchical disentanglement as we have already formulated it, e.g. by improving robustness to noise (Fig. A.1) or providing a better theoretical understanding of identifiability, perhaps through the lens of description length. Finally, there are ample opportunities to apply these techniques to real-world data that we expect to have hierarchical multiple-manifold structure, such as patient phenotype or population genetics datasets.

More generally, we feel it is important for representation learning to move beyond flat vectors, and work towards explicitly modeling the rich structure contained in the real world. Symbolic AI and cognitive science researchers have made compelling arguments that future AI progress should be evaluated not by improvements in accuracy or reconstruction error, but by how well models build their own interpretable models of the world (Lake et al., 2017). Our work takes steps in this direction.

Acknowledgements

The authors thank members of the Harvard DtAK lab for helpful discussions and insights. ASR acknowledges support from the Miami Foundation. FDV acknowledges support from NSF-CAREER 1750358.

References

Ahlqvist, E., Storm, P., Käräjämäki, A., Martinell, M., Dorkhan, M., Carlsson, A., Vikman, P., Prasad, R. B., Aly, D. M., Almgren, P., et al. Novel subgroups of adult-onset diabetes and their association with outcomes: a data-driven cluster analysis of six variables. *The lancet Diabetes & endocrinology*, 6(5):361–369, 2018.

Alvarez-Melis, D. and Jaakkola, T. S. Towards robust interpretability with self-explaining neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 7786–7795, 2018.

American Diabetes Association. Diagnosis and classification of diabetes mellitus. *Diabetes Care*, 28(1):S37, 2005.

Ankerst, M., Breunig, M. M., Kriegel, H.-P., and Sander, J. Optics: Ordering points to identify the clustering structure. *ACM Sigmod record*, 28(2):49–60, 1999.

Arcones, M. A. and Gine, E. On the bootstrap of u and v statistics. *The Annals of Statistics*, pp. 655–674, 1992.

Barlow, H. B. Possible principles underlying the transformation of sensory messages. *Sensory communication*, 1 (01), 1961.

Bengio, Y. Deep learning of representations: Looking forward. In *International Conference on Statistical Language and Speech Processing*, pp. 1–37. Springer, 2013.

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

Chen, T. Q., Li, X., Grosse, R. B., and Duvenaud, D. K. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, pp. 2610–2620, 2018.

Choi, J., Hwang, G., and Kang, M. Discond-vae: disentangling continuous factors from the discrete. *arXiv preprint arXiv:2009.08039*, 2020.

Comon, P. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.

Creager, E., Madras, D., Jacobsen, J.-H., Weis, M., Swersky, K., Pitassi, T., and Zemel, R. Flexibly fair representation learning by disentanglement. In *International Conference on Machine Learning*, pp. 1436–1445. PMLR, 2019.

Defferrard, M., Bresson, X., and Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29:3844–3852, 2016.

Dupont, E. Learning disentangled joint continuous and discrete representations. In *Advances in Neural Information Processing Systems*, pp. 710–720, 2018.

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

Esmaeili, B., Wu, H., Jain, S., Bozkurt, A., Siddharth, N., Paige, B., Brooks, D. H., Dy, J., and van de Meent, J.-W. Structured disentangled representations. *stat*, 1050:29, 2018.

Fischler, M. A. and Bolles, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, volume 3, 2017.

Jeong, Y. and Song, H. O. Learning discrete and continuous factors of data via alternating disentanglement. *arXiv preprint arXiv:1905.09432*, 2019.

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

Kim, M., Wang, Y., Sahu, P., and Pavlovic, V. Relevance factor vae: Learning and identifying disentangled factors. *arXiv preprint arXiv:1902.01568*, 2019.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Klindt, D. A., Schott, L., Sharma, Y., Ustyuzhaninov, I., Brendel, W., Bethge, M., and Paiton, D. Towards nonlinear disentanglement in natural data with temporal sparse coding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=EbIDjBynYJ8.

Kumar, A., Sattigeri, P., and Balakrishnan, A. Variational inference of disentangled latent concepts from unlabeled observations. *arXiv preprint arXiv:1711.00848*, 2017.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *nature*, 521(7553):436–444, 2015.

Locatello, F., Bauer, S., Lucic, M., Rätsch, G., Gelly, S., Schölkopf, B., and Bachem, O. Challenging common assumptions in the unsupervised learning of disentangled representations. *arXiv preprint arXiv:1811.12359*, 2018.

Locatello, F., Poole, B., Rätsch, G., Schölkopf, B., Bachem, O., and Tschannen, M. Weakly-supervised disentanglement without compromises. *arXiv preprint arXiv:2002.02886*, 2020a.

Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., and Kipf, T. Object-centric learning with slot attention. *Advances in Neural Information Processing Systems*, 33, 2020b.

Mahapatra, S. and Chandola, V. S-isomap++: multi manifold learning from streaming data. In *2017 IEEE International Conference on Big Data (Big Data)*, pp. 716–725. IEEE, 2017.

Mahler, B. I. Contagion dynamics for manifold learning. *arXiv preprint arXiv:2012.00091*, 2020.

Marx, C., Phillips, R., Friedler, S., Scheidegger, C., and Venkatasubramanian, S. Disentangling influence: Using disentangled representations to audit model predictions. *Advances in Neural Information Processing Systems*, 32: 4496–4506, 2019.

Matthey, L., Higgins, I., Hassabis, D., and Lerchner, A. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

Mescheder, L., Geiger, A., and Nowozin, S. Which training methods for gans do actually converge? In *International conference on machine learning*, pp. 3481–3490. PMLR, 2018.

Omohundro, S. M. *Five balltree construction algorithms*. International Computer Science Institute Berkeley, 1989.

Parascandolo, G., Kilbertus, N., Rojas-Carulla, M., and Schölkopf, B. Learning independent causal mechanisms. In *International Conference on Machine Learning*, pp. 4036–4044. PMLR, 2018.

Ridgeway, K. A survey of inductive biases for factorial representation-learning. *arXiv preprint arXiv:1612.05299*, 2016.

Ross, A., Chen, N., Hang, E. Z., Glassman, E. L., and Doshi-Velez, F. Evaluating the interpretability of generative models by interactive reconstruction. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021.

Schmidhuber, J. Learning factorial codes by predictability minimization. *Neural computation*, 4(6):863–879, 1992.

Siddharth, N., Paige, B., Van de Meent, J.-W., Desmaison, A., Goodman, N., Kohli, P., Wood, F., and Torr, P. Learning disentangled representations with semi-supervised deep generative models. In *Advances in neural information processing systems*, pp. 5925–5935, 2017.

Singh, K. K., Ojha, U., and Lee, Y. J. Finegan: Unsupervised hierarchical disentanglement for fine-grained object generation and discovery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6490–6499, 2019.

Sorrenson, P., Rother, C., and Köthe, U. Disentanglement by nonlinear ica with general incompressible-flow networks (gin). *arXiv preprint arXiv:2001.04872*, 2020.

Sugiyama, M., Suzuki, T., and Kanamori, T. Density-ratio matching under the bregman divergence: a unified framework of density-ratio estimation. *Annals of the Institute of Statistical Mathematics*, 64(5):1009–1044, 2012.

Sønderby, C. K., Raiko, T., Maalø e, L., Sønderby, S. K., and Winther, O. Ladder Variational Autoencoders. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29*, pp. 3738–3746. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6275-ladder-variational-autoencoders.pdf.

Träuble, F., Creager, E., Kilbertus, N., Goyal, A., Locatello, F., Schölkopf, B., and Bauer, S. Is independence all you need? on the generalization of representations learned from correlated data. *arXiv preprint arXiv:2006.07886*, 2020.

Wang, W., Huang, Y., Wang, Y., and Wang, L. Generalized autoencoder: A neural network framework for dimensionality reduction. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 490–497, 2014.

Yang, M., Liu, F., Chen, Z., Shen, X., Hao, J., and Wang, J. Causalvae: Structured causal disentanglement in variational autoencoder. *arXiv preprint arXiv:2004.08697*, 2020.

Zemel, R. S. *A minimum description length framework for unsupervised learning*. University of Toronto, 1994.

Zhang, K. and Shasha, D. Simple fast algorithms for the editing distance between trees and related problems. *SIAM journal on computing*, 1989.