Structure by Architecture: Disentangled Representations without Regularization

Felix Leeb ‡ Guilia Lanzillotta † Yashas Annadani § Michel Besserve †§ Stefan Bauer †§ Bernhard Schölkopf §

Abstract

We study the problem of self-supervised structured representation learning using autoencoders for generative modeling. Unlike most methods which rely on matching an arbitrary, relatively unstructured, prior distribution for sampling, we propose a sampling technique that relies solely on the independence of latent variables, thereby avoiding the trade-off between reconstruction quality and generative performance inherent to VAEs. We design a novel autoencoder architecture capable of learning a structured representation without the need for aggressive regularization. Our structural decoders learn a hierarchy of latent variables, akin to structural causal models, thereby ordering the information without any additional regularization. We demonstrate how these models learn a representation that improves results in a variety of downstream tasks including generation, disentanglement, and extrapolation using several challenging and natural image datasets.

1 Introduction

Deep learning has achieved strong results on a plethora of challenging tasks. However, performing well on a highly specific dataset is usually insufficient to satisfactorily solve real-world problems [1, 2]. This has lead to a particular interest in consistently learning more structured representations with useful properties to help with a variety of downstream tasks [3, 4, 6]. Here, deep learning provides a flexible paradigm to train complex architectures based on autoencoders [7, 8] which are typically latent variable models, thus allowing us to embed powerful inductive biases into the models to further structure the representations. However, it is still largely an open question as to what kinds of structure in a representation are the most effective for generative modeling and how to learn such structures without supervision [9–13]. One direction that may contribute to an answer is causal modeling, as it focuses on the underlying (causal) mechanisms that generate the observations, instead of relying on (possibly spurious) correlations [14–19].

With the versatility of deep learning on one hand, and the conceptual insights of causality on the other, our contributions herein include:

• We propose an architecture called the Structural Autoencoder (SAE), where the structural decoder emulates a general acyclic structural causal model to learn a hierarchical representation that can separate and order the underlying factors of variation in the data.
We provide a sampling method that can be used for any autoencoder-based generative models which does not use an explicit regularization objective and instead relies on independence in the latent space.

We investigate how well the encoder and decoder are able to extrapolate upon seeing novel samples.

We release our code at *anonymized*.

### 1.1 Related Work

The most popular autoencoder based method is the Variational Autoencoder (VAE) [20] and the closely related $\beta$VAE [21]. These methods focus on matching the distribution in the latent space to a known prior distribution by regularizing the reconstruction training objective [22, 23]. Although this structure is convenient for generative modeling and even tends to disentangle the latent space to some extent, it comes at the cost of somewhat blurry images due to posterior collapse and holes in the latent space [9, 21, 24–26].

To mitigate the double-edged nature of VAEs, less aggressive regularization techniques have been proposed such as the Wasserstein Autoencoder (WAE), which focuses on the aggregate posterior [27]. Unfortunately, WAEs generally fail to produce a particularly meaningful or disentangled latent space [28], unless weak supervision is available [29].

A more structured alternative is the Variational Ladder autoencoder (VLAE) [30] which separates the latent space into separate chunks each of which is processed at different levels of the encoder and decoder (called "rungs"). Our proposed architecture takes inspiration from the VLAEs but crucially we do not use the variational regularization, and instead use a sampling method based on hybridization [31]. We also infuse the information from the latent variables using Str-Tfm layers (see section 2.2) which are based on the Ada-IN layers from Style-GANs [32], which significantly improves on the ladder rungs.

### 2 Methods

#### 2.1 Causal Representation Learning

Graphical causal modeling builds on random variables

$$S_i := f_i(PA_i, U_i), \quad (i = 1, \ldots, n),$$

connected by a directed acyclic graph (DAG) whose edges represent direct causation. Each $S_i$ is computed using a function $f_i$ depending on its parents $PA_i$ in the graph and on an unexplained noise variable $U_i$, ensuring that the “mechanism” (1) can represent any conditional distribution $p(S_i|PA_i)$. The noises $U_1, \ldots, U_n$ are assumed to be jointly independent. The DAG along with the mechanisms (1) is referred to as a Structural Causal Model (SCM) [14]. Any joint distribution of the $S_i$ can be expressed as an SCM using suitable $f_i$ and $U_i$. However, the SCM also represents interventions (e.g., fixing values of $U_i$ or $S_i$), and it makes it explicit how statistical dependences between the $S_i$ are generated by mechanisms (1).

Real-world observations, however, are often not structured into meaningful causal variables and mechanisms to begin with. E.g., images are high-dimensional, and it is hard to learn objects and their causal relationships from data [33]. One may thus attempt to try and learn a representation consisting of causal variables or disentangled “factors” which are statistically independent [21]. However, in an SCM it is not the $S_i$ that should be statistically independent, but the $U_i$. For this reason, our representations will comprise the $U_i$ as latent variables, driving the causal mechanisms in a way that can structurally implement (1). This embeds an SCM into a larger model whose inputs and outputs may be high-dimensional and unstructured (e.g., images) [34,16].

Given (high-dimensional) $X = (X_1, \ldots, X_d)$, we first use an encoder $f_{enc}: \mathbb{R}^d \to \mathbb{R}^n$ taking $X$ to a latent representation $U = (U_1, \ldots, U_n)$. Next, we apply the SCM, with the $U_i$ feeding into subsequent computation layers according to a causal ordering (see Supplement), i.e., the root node(s) in the DAG only depend on “their” noise variables, while later ones depend on their noise and potentially also those of their parents, and so on. The depth in the network thus corresponds
Figure 1: The Structural Decoder reconstructs (or generates) an image from a latent vector $q$ by first splitting $q$ into $K$ segments each of which transforms the image features at a different layer of the model using a Str-Tfm layer (green box where all pixels $x_j$ are transformed with the same $z_s$ and $z_b$).

to a causal ordering (cf. Figure 1 below). Finally, we apply a decoder $f_{\text{dec}}$ taking us back to $\mathbb{R}^d$. If the causal graph were known, the topology of the network implementing the SCM could be fixed accordingly. Below, we assume that we do not know it, and thus our decoder will learn the composition of SCM and decoder, with the SCM effectively becoming an unidentified part thereof.

By choosing the decoder topology, we will ensure that different parts of the noise vector should feed into separate subsequent components/mechanisms (using connections skipping layers). In view of the above considerations, this means that in principle, any SCM can be learned and embedded in this architecture. In analogy to structural causal models, we will below refer to this as a structural decoder.

The above shows that an autoencoder with independent latent variables can be viewed as the composition of an (anticausal) encoder that recognizes or reconstructs causal drivers in the world, and a (causal) generator that maps the low dimensional latent representation (of the noises driving the causal model) back to the high dimensional world. In practice, without some form of supervision or side information, the learned SCM is not guaranteed to match the true SCM.

2.2 Structural Decoders

Our structural decoder splits the latent vector into $K D_k$-dimensional separate segments for a total latent space dimensionality of $D = K \times D_k$. Each segment is applied to its own Structural-Transform (Str-Tfm) layer (as seen in Figure 1) to produce a scale $z_s$ and bias $z_b$ which are then used to transform the features much like in Ada-IN [32] except without the normalization.

Each Str-Tfm layer thus acts like an $f_i$ in (1) by integrating the information from a latent variable $U_i$ to transform the features $PA_i$ from earlier layers producing a hierarchical structure in the latent space. Since the decoder uses significantly more layers for the first few latent dimensions than the last few, more nonlinear information is naturally encoded in the first few segments. Meanwhile, the low-level linear features are only processed by the relatively shallow last few layers of the decoder.

2.3 Hybrid Sampling

For generative modeling, it is necessary to sample novel latent vectors that are transformed into (synthetic) observations using the decoder. Usually, this is done by regularizing the training objective so the posterior matches some simple prior (e.g. the standard normal). However, in practice, regularization techniques can fail to match the prior perfectly and actually exacerbate the information bottleneck, leading to blurry samples from holes in the learned latent distribution and unused latent dimensions due to posterior collapse [35] [26] [36] [37]. Instead of trying to match some prior distribution in the latent space, we suggest an alternative sampling method that eliminates the need for any regularization of the loss. Inspired by [31], we refer to it as hybrid sampling: the model stores a
finite set of $N (= 128)$ latent vectors, selected uniformly at random from the training set. To generate a new latent vector, a value for each of the $D (= 12)$ dimensions is selected independently from the $N$ stored latent vectors uniformly at random (see figure 2). This allows the model to generate $N^D$ latent vectors.

Hybrid sampling implicitly relies on the latent dimensions being independent since resampling the marginal of the posterior independently breaks any existing correlations, however, this is consistent with the causal perspective of the latent variables as noises driving an SCM. Independence between latent variables also aligns with the objective to disentangle the latent factors of variation. Note that hybrid sampling is directly applicable to any learned representation as it does not affect training at all, however the fidelity of generated samples will diminish if there are strong correlations between latent dimensions, so maximizing the independence between latent variables becomes the implicit goal. Since the structural decoders already split the latent space into separate segments, each full segment is hybridized together, instead of independently resampling dimensions that contribute to the same segment, which would destroy any learned structure or dependence within a segment.

3 Experiments

We train the proposed architectures and baselines on four different datasets: 3D-Shapes [38], the three variants ("toy", "sim", and "real") of the MPI3D Disentanglement dataset [39], as well as Celeb-A [40] and the Robot Finger Dataset (RFD) [41].

After training our models on a standard 70-10-20 (train-val-test) split of the datasets we evaluate the quality of the reconstructions based on the reconstruction loss (using binary cross entropy loss) and the Fréchet Inception Distance (FID) [42] as in [43]. The FID is able to capture higher level visual features and can be used to directly compare the reconstructed and generated sample quality, while the binary cross entropy is a purely pixelwise comparison.

Next we compare the performance of the hybrid sampling method, which can be applied to all of the models, to the prior based sampling, which instead only makes sense for the models that use regularization (VLAE, VAE, $\beta$VAE, and WAE). Finally we take a closer look at the learned representations to understand how the model architecture affects the induces certain structure and possibly disentanglement.

3.1 Models

All models use the same CNN backbone for both the encoder and decoder with the same number of convolution layers and filters, each of which is followed by group normalization and a MISH nonlinearity [44], where the encoder uses max-pooling to down-sample the extracted features and the decoder uses bilinear up-sampling to up-sample features (see the appendix for details). For 3D-Shapes and the MPI3D datasets, the encoder and decoder have 12 convolution layers each with

\footnote{The CelebA dataset is available for non-commercial research purposes (no license provided). It does show images of individuals, and permission may not have been obtained; however, our understanding is that usage of these images is permitted since the individuals are celebrities. We will remove those experiments from the paper if this is considered inappropriate.}

Figure 2: In hybrid sampling, each component of latent vectors (colored boxes) are uniformly randomly resampled to produce novel latent samples. In general, each latent dimension is resampled independently, however, since the SAE models, each segment (which may include multiple latent dimensions) is hybridized.
64 filters, the latent space has 12 dimensions, and the models are trained for 100k iterations, while for CelebA each the encoder/decoder has 16 convolution layers each with twice as many filters, the latent space is 32 dimensional, and the models are trained for 200k iterations. For the RFD dataset, the settings are the same as for CelebA, except that the number of channels per convolution layer was doubled again to 256. All training and evaluation was done using V100 GPUs, where training on the smaller datasets took about 2 GPU hours per model, and around 16-20 GPU hours for CelebA and RFD.

We compare four kinds of autoencoder architectures. The first type is our Structural Autoencoders (SAE) which use a conventional encoder and a structural decoder with 2, 3, 4, 6, or 12 Str-Tfm layers placed evenly between convolution layers (and 16 for CelebA). Since the total number of dimensions in the latent space is constant, the number of dimensions per segment is 6, 4, 3, 2, 1 for the SAE-2, SAE-3, SAE-4, SAE-6, and SAE-12 respectively. The simplest "baseline" architecture uses the traditional "hourglass" architecture, in addition to a variety of different regularization methods: (1) unregularized autoencoders ("AE"), (2) Wasserstein-autoencoders ("WAE") [27], (3) VAEs [45], and a $\beta$VAE [21] (with the experimentally determined best setting of $\beta = 2$). The next more structured baseline architecture is called "AdaAE-12" (and referred to as "Adaptive" as it is identical to the SAE-12 model except that instead of splitting the latent vector into 12 segments with one dimension used per Str-Tfm layer, each Str-Tfm layer receives the entire latent vector, so the architecture is effectively the same as using Ada-IN layers [32].

The final type of architecture we investigate is the Variational Ladder Autoencoder [30] which also uses a structured architecture to learn a hierarchical representation, but unlike our Structural Autoencoders, VLAEs also use the variational regularization and use an encoder architecture that roughly mirrors the decoder. Another subtle, but significant difference between the SAE and VLAE architecture is that the ladder rungs in the VLAE decoder concatenate the features from the latent space segment of the corresponding rung in the encoder, instead of using the segment to scale and shift the convolution features directly. Just like for the SAE models, we include variants of the VLAEs with 2, 3, 4, 6, and 12 rungs.

3.2 Extrapolation

While the encoder and decoder are generally optimized jointly, the tight coupling combined with the inherent asymmetry in the respective learning tasks begs the question which of the two is the weaker link when it comes to generalization. To this end, we investigate to what extent the encoder vs the decoder can extrapolate to novel observations.

First, both the encoder and decoder are trained jointly on a subset of 3D-Shapes where only three distinct shapes exist (instead of four, as the ball is missing) for 80k iterations. Then either the encoder only, the decoder only, or both are trained for another 20k iterations on the full 3D-Shapes training dataset. The reconstruction error with for samples not seen during any part of training is compared for each of the variants and each of the architectures to identify whether updating the encoder or decoder has a more significant impact on performance and how well the encoder extrapolates compared to the decoder.

4 Results

In terms of reconstruction quality, the structural autoencoder architecture consistently outperforms the baselines (seen in figure 3). As expected, unregularized methods like the SAE, AdaAE, and AE tend to have significantly better reconstruction quality. However, the structured architectures like the SAE and VLAE also improve performance somewhat.

In comparing the SAE models to the AdaAE architecture, we see that there can be a slight penalty in reconstruction quality incurred from splitting the latent vector into segments. However this is more than made up for in the quality of the generated samples (shown in figure 4), where the SAE models perform significantly better than the baselines. Even the regularized models such as VLAE and VAEs usually generate higher quality samples using the hybrid sampling than when sampling from the prior they were trained to match. This can qualitatively be observed from figure 5b. Surprisingly, the AdaAE architecture actually outperforms all other models on CelebA using hybrid sampling. This may be explained by the severe information bottleneck experienced when embedding CelebA into
Figure 3: Results on the reconstruction quality for all models and datasets. The "Baseline" models correspond to traditional "hourglass" CNN architectures, while the "Structural" models use our novel architectures to further structure the learned representation.

Figure 4: Comparison of the generative models using different sampling methods. Note that our SAE models perform well without having to regularize the latent space towards a prior. In fact, even with the conventional "hourglass" architecture (in orange), the hybrid sampling method generates relatively high quality samples.
(a) Here we use hybridized chunks of latent vector to show how different aspects of the resulting generated image can be affected using our SAE-16 architecture and hybrid sampling technique. For each row the corresponding quarter of latent dimensions (8/32) are hybridized (see section 2.3) while the remaining 3/4 are fixed. This shows how the SAE architecture is able to order partially disentangled factors of variation from high-level (more nonlinear, like facial expressions and features) to low-level (such as color/lighting) without any additional regularization or supervision.

(b) Samples generated using hybrid and prior-based sampling using several models trained on CelebA. Note that the hybrid sampling tends to produce relatively high quality samples both for our proposed SAE and AdaAE architectures as well as baselines.

Figure 5: CelebA Controllable Generation and Sampling Comparison

only 32 dimensions. Since the AdaAE model produces the best reconstructions, the generated samples have a higher fidelity even though any learned structure between latent dimensions is disregarded by the hybrid sampling.

In general, if we consider the distribution of the latent variables (i.e., the push-forward of the data distribution into the latent space), then sampling from a simple factorized prior can introduce at least two types of errors: (1) errors due to not taking into account statistical dependences among latent variables, and (2) errors due to sampling from "holes" in the latent distribution if the prior does not match it everywhere. Whenever (2) is the dominating source of error, hybrid sampling is effective.

4.1 Hierarchical Structure

To get a rough idea of how the representations learned using the structural decoders differ from more conventional architectures, figure 6 shows the one dimensional latent traversals (i.e., each row corresponds to the decoder outputs when incrementally increasing the corresponding latent dimension at a time from the min to the max value observed). The traversals illustrate the hierarchical structure in the representation learned by the SAE models: the information encoded to the first few latent variables can be more nonlinear with respect to the output (pixel) space, as the decoder has more layers to process that information, while the more linear information can be embedded in the last few latent dimensions. This aligns nicely with the ordering of placing more "high-level" information (such as object shape or viewpoint) first, followed by the "low-level" information (such as color). This means the structural decoder architecture biases the representation to separate and order the information necessary for reconstruction (and generation) in a meaningful way, and thereby tends to disentangle the underlying factors of variation better.

Figure 7 evaluates how disentangled the representations are for five different random seeds (used to initialize the network parameters) using common metrics. Most noteworthy is that the SAE-12 model consistently achieves very high disentanglement scores. This shows, empirically, that the SAE architecture promotes independence between latent variables (especially SAE-12). We may explain this as a consequence of splitting up the latent dimensions so that each variable has a unique parameterization in the decoder, making different latent variables less likely to be processed in the same way. Unsurprisingly, as you split the latent space into fewer pieces like in SAE-6 or SAE-4, each segment gets more entangled, decreasing the overall score.
Figure 6: Latent traversals of several models trained on 3D-Shapes, in the original order. Note the ordering of the information in the structural decoder models (SAE-12 and SAE-3) where higher level, nonlinear features (like shape and orientation) are encoded in the first few dimensions, which are located deeper in the network.

| Model       | DCI-D | IRS  | MIG  | ModExp |
|-------------|-------|------|------|--------|
| SAE-12      | 0.999 | 0.855| 0.586| 0.968  |
| SAE-6       | 0.815 | 0.712| 0.133| 0.969  |
| SAE-4       | 0.693 | 0.560| 0.218| 0.918  |
| VLAE-12     | 0.829 | 0.676| **0.662** | 0.929  |
| VLAE-6      | 0.785 | 0.689| 0.326| 0.929  |
| VLAE-4      | 0.690 | 0.544| 0.282| 0.900  |
| AdaAE-12    | 0.272 | 0.477| 0.046| 0.862  |
| AE          | 0.326 | 0.610| 0.093| 0.880  |
| VAE         | 0.441 | 0.712| 0.252| 0.904  |
| βVAE        | 0.196 | 0.640| 0.107| 0.834  |
| WAE         | 0.205 | 0.640| 0.057| 0.958  |

Figure 7: Disentanglement scores for 3D-Shapes. The DCI-D metric corresponds to the DCI-disentanglement score [46], IRS is the Interventional Robustness Score [47], MIG is the Mutual Information Gap [48], and ModExp refers to the Modularity Explicitness score [49] (for all these metrics higher is better). The figure on the right shows how the scores vary across five models with different random seeds as "x"s (and the resulting mean and standard deviation as lines). Note that not only does the SAE architecture consistently achieve high disentanglement scores, but its performance is also less sensitive to the random seed.

For a real world demonstration of how well SAE models are able to order information in the latent space, figure 5a shows generated CelebA samples when varying only a quarter of the latent vector at a time. The labels on the left are empirically chosen by the authors to describe roughly what kind of semantic information that quarter of the latent space appears to contain. Although the inductive biases are not strong enough to fully disentangle the factors of variation into individual latent dimensions. The hierarchical representation encourages a diffuse kind of disentanglement where information pertaining to higher-level features tend to be encoded in the first few dimensions while lower level factors of variation show up towards the last few dimensions.

SAE models achieve this structured disentanglement using the Str-Tfm layers as opposed to the standard Ada-In layers. Each Str-Tfm layer only has access to a small segment of the latent space, and that segment is not directly seen by any other part of the decoder. It thus has to learn to use the information in the corresponding segment to transform the pixels in the output image. In contrast, the Ada-In layers used by the AdaAE allows information from anywhere in the latent vector to leak into any part of the decoder. Consequently, the AdaAE does not disentangle the representation at all (although it achieves impressive results for reconstruction nonetheless).
4.2 Extrapolation

Perhaps unsurprisingly, none of the models were particularly adept at zero-shot extrapolation to observations that were not in the initial training data. As seen in figure 8 on the right comparing the first and second columns, without any update, the reconstructed images filter out the novel information in the sample (in this case, ball shape), and instead reconstruct a similar sample seen during training (a cylinder).

If only the encoder is updated on some observations with the additional shape while the decoder is frozen, then the reconstruction performance increases somewhat, but some deformations and artifacts become visible in the reconstruction. This suggests the frozen decoder struggles to adequately extrapolate, even when the encoder extends the representation to include the ball.

In contrast, when only the decoder is updated, the reconstructions qualitatively look much more similar to the original observations. This also suggests that although the encoder can be expected to generally lose out any information it has not been trained to encode into the latent space due to the bottleneck, as long as the changes to the observations are mild, the representation may still extrapolate provided the decoder can learn to reconstruct any novel features.

| Model   | Neither | Encoder | Decoder | Both  |
|---------|---------|---------|---------|-------|
| SAE-12  | 13.21   | 7.7     | 0.42    | 0.34  |
| LV-VAE-12 | 18.37  | 7.69    | 1.55    | 0.62  |
| VAE     | 12.97   | 8.78    | 0.44    | 0.46  |
| βVAE    | 15.49   | 8.31    | 1.35    | 0.52  |
| AE      | 11.81   | 7.31    | 0.38    | 0.35  |
| WAE     | 11.68   | 7.87    | 0.37    | 0.35  |

Figure 8: The table shows the average reconstruction error (MSE x1000) on novel observations (example shown on the right) after updating either the encoder, decoder, both, or neither in the extrapolation setting (see section 3.2). Note that all models perform significantly better when updating the decoder than the encoder, and reach a reconstruction quality that is almost indistinguishable from the model when updating both the encoder and decoder. Furthermore, note that the SAE-12 generally outperforms the all variational baselines, suggesting the aggressive regularization of VAEs makes updating the representation more difficult.

5 Conclusion

While VAEs provide a principled approach to generative modeling with autoencoders, in practice, the regularization sharpens the information bottleneck by penalizing any dependence of the posterior on the observation [37], resulting in a trade-off between reconstruction quality and matching the prior. While this can encourage a more compact and even disentangled representation, it also tends to result in blurry generated samples and relatively poor reconstructions.

This motivated us to look for an alternate sampling method that does not require aggressive regularization as VAEs. Our hybrid sampling technique relies on independence between latent variables instead of expecting the learned posterior to match a (relatively unstructured) prior. This effectively unifies the goals of achieving a disentangled and samplable representation.

To that end, we propose the structural autoencoder architecture inspired by structural causal models, which orders information in the latent space, while also, as shown by our experiments, encourages independence. Notably, it does so without additional loss terms or regularization. The SAE architecture produces high quality reconstructions and generated samples, improving extrapolation, as well as achieving a significant degree of disentanglement across a variety of datasets.

The results obtained thus suggest that in order to enforce a desired structure on the representations, implicit architectural bias might be a valid alternative to explicit assumptions on the latent space distribution, whose accuracy is often questionable outside of the synthetic data setting.
While it is encouraging how far one can get with a suitable architectural bias, future work should assay whether the learned models can be structured further by more explicit forms of causal training. For instance, we could explicitly encourage, during training, that hybridization produce realistic samples, or that across domain shifts or actions/interventions external to the network, only a sparse set of the latent factors change (cf. the Sparse Mechanism Shift Hypothesis, [50]).

Acknowledgements

This work was supported by the German Federal Ministry of Education and Research (BMBF): Tübingen AI Center, FKZ: 01IS18039B, and by the Machine Learning Cluster of Excellence, EXC number 2064/1 – Project number 390727645. The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Felix Leeb.

References

[1] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In International conference on artificial neural networks, pages 270–279. Springer, 2018.

[2] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. arXiv preprint arXiv:1911.02685, 2019.

[3] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence, 35(8): 1798–1828, 2013.

[4] Michael Tschannen, Olivier Bachem, and Mario Lucic. Recent advances in autoencoder-based representation learning. arXiv preprint arXiv:1812.05069, 2018.

[5] Emmanuel Bengio, Valentin Thomas, Joelle Pineau, Doina Precup, and Yoshua Bengio. Independently controllable features, 2017. URL arXiv:1703.07718.

[6] Adityanarayanan Radhakrishnan, Karren Yang, Mikhail Belkin, and Caroline Uhler. Memorization in overparameterized autoencoders. arXiv preprint arXiv:1810.10333, 2018.

[7] Dana H Ballard. Modular learning in neural networks. In AAAI, pages 279–284, 1987.

[8] Sascha Lange and Martin Riedmiller. Deep auto-encoder neural networks in reinforcement learning. In The 2010 International Joint Conference on Neural Networks (IJCNN), pages 1–8. IEEE, 2010.

[9] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. arXiv preprint arXiv:1811.12359, 2018.

[10] Valentin Khrulkov, Leyla Mirvakhabova, Ivan Oseledets, and Artem Babenko. Disentangled representations from non-disentangled models. arXiv preprint arXiv:2102.06204, 2021.

[11] Rui Shu, Yining Chen, Abhishek Kumar, Stefano Ermon, and Ben Poole. Weakly supervised disentanglement with guarantees. arXiv preprint arXiv:1910.09772, 2019.

[12] Shang-Fu Chen, Jia-Wei Yan, Ya-Fan Su, and Yu-Chiang Frank Wang. Deep representation decomposition for feature disentanglement. arXiv preprint arXiv:2011.00788, 2020.

[13] Weili Nie, Tero Karras, Animesh Garg, Shoubhik Debnath, Anjul Patney, Ankit Patel, and Animeshree Anandkumar. Semi-supervised stylegan for disentanglement learning. In International Conference on Machine Learning, pages 7360–7369. PMLR, 2020.

[14] J. Pearl. Causality: Models, Reasoning, and Inference. Cambridge University Press, New York, NY, 2nd edition, 2009.
[15] J. Peters, D. Janzing, and B. Schölkopf. *Elements of Causal Inference - Foundations and Learning Algorithms*. MIT Press, Cambridge, MA, USA, 2017.

[16] B. Schölkopf. Causality for machine learning, 2019. arXiv:1911.10500.

[17] Christos Louizos, Uri Shalit, Joris Mooij, David Sontag, Richard Zemel, and Max Welling. Causal effect inference with deep latent-variable models. *arXiv preprint arXiv:1705.08821*, 2017.

[18] Jovana Mitrovic, Brian McWilliams, Jacob Walker, Lars Buesing, and Charles Blundell. Representation learning via invariant causal mechanisms. *arXiv preprint arXiv:2010.07922*, 2020.

[19] Xinwei Shen, Furui Liu, Hanze Dong, Qing Lian, Zhitang Chen, and Tong Zhang. Disentangled generative causal representation learning. *arXiv preprint arXiv:2010.02637*, 2020.

[20] Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes, 2013. URL arXiv:1312.6114.

[21] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2(5):6, 2017. URL https://pdfs.semanticscholar.org/a902/26c41b7f9f8b06007609f39f82757073641e2.pdf.

[22] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. A sober look at the unsupervised learning of disentangled representations and their evaluation. *Journal of Machine Learning Research (JMLR)*, 2020.

[23] Sharon Zhou, Eric Zelikman, Fred Lu, Andrew Y Ng, and Stefano Ermon. Evaluating the disentanglement of deep generative models through manifold topology. *arXiv preprint arXiv:2006.03680*, 2020.

[24] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in $\beta$-VAE. *arXiv preprint arXiv:1804.03599*, 2018.

[25] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*, 2018. URL https://arxiv.org/pdf/1802.05983.pdf.

[26] Jan Stühmer, Richard Turner, and Sebastian Nowozin. Independent subspace analysis for unsupervised learning of disentangled representations. In *International Conference on Artificial Intelligence and Statistics*, pages 1200–1210. PMLR, 2020.

[27] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schölkopf. Wasserstein auto-encoders. In 6th *International Conference on Learning Representations (ICLR)*, May 2018. URL https://openreview.net/forum?id=HkL7n1-0b.

[28] Paul K Rubenstein, Bernhard Schoelkopf, and Ilya Tolstikhin. On the latent space of Wasserstein auto-encoders. *arXiv preprint arXiv:1802.03761*, 2018.

[29] Jun Han, Martin Renqiang Min, Ligong Han, Li Erran Li, and Xuan Zhang. Disentangled recurrent wasserstein autoencoder. *arXiv preprint arXiv:2101.07496*, 2021.

[30] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Learning hierarchical features from deep generative models. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 4091–4099. JMLR.org, 2017.

[31] Michel Besserve, Arash Mehrjou, Rémy Sun, and Bernhard Schölkopf. Counterfactuals uncover the modular structure of deep generative models. *arXiv preprint arXiv:1812.03253*, 2018.

[32] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. URL http://openaccess.thecvf.com/content_CVPR_2019/papers/Karras_A_Style-Based_Generator_Architecture_for_Generative_Adversarial_Networks_CVPR_2019_paper.pdf.
[33] D. Lopez-Paz, R. Nishihara, S. Chintala, B. Schölkopf, and L. Bottou. Discovering causal signals in images. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 58–66, 2017.

[34] Raphael Suter, Djorde Miladinovic, Bernhard Schölkopf, and Stefan Bauer. Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness. arXiv preprint arXiv:1811.00007, 2018.

[35] Bin Dai and David Wipf. Diagnosing and enhancing VAE models. arXiv preprint arXiv:1903.05789, 2019.

[36] James Lucas, George Tucker, Roger Grosse, and Mohammad Norouzi. Understanding posterior collapse in generative latent variable models. 2019.

[37] Matthew D Hoffman and Matthew J Johnson. Elbo surgery: yet another way to carve up the variational evidence lower bound. In Workshop in Advances in Approximate Bayesian Inference, NIPS, volume 1, page 2, 2016.

[38] Chris Burgess and Hyunjik Kim. 3d shapes dataset, 2018.

[39] Muhammad Waleed Gondal, Manuel Wüthrich, Djorde Miladinovic, Francesco Locatello, Martin Breidt, Valentin Volchkov, Joel Akpo, Olivier Bachem, Bernhard Schölkopf, and Stefan Bauer. On the transfer of inductive bias from simulation to the real world: a new disentanglement dataset. arXiv preprint arXiv:1906.03292, 2019.

[40] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In Proceedings of International Conference on Computer Vision (ICCV), December 2015.

[41] Andrea Dittadi, Frederik Träuble, Francesco Locatello, Manuel Wüthrich, Vaibhav Agrawal, Ole Winther, Stefan Bauer, and Bernhard Schölkopf. On the transfer of disentangled representations in realistic settings. arXiv preprint arXiv:2010.14407, 2020.

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

[43] Will Williams, Sam Ringer, Tom Ash, John Hughes, David MacLeod, and Jamie Dougherty. Hierarchical quantized autoencoders. arXiv preprint arXiv:2002.08111, 2020.

[44] Diganta Misra. Mish: A self regularized non-monotonic neural activation function. arXiv preprint arXiv:1908.08681, 4, 2019.

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

[46] Cian Eastwood and Christopher K Williams. A framework for the quantitative evaluation of disentangled representations. 2018.

[47] Raphael Suter, Dorde Miladinovic, Stefan Bauer, and Bernhard Schölkopf. Interventional robustness of deep latent variable models. stat, 1050:31, 2018.

[48] Ricky TQ Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders. arXiv preprint arXiv:1802.04942, 2018.

[49] Karl Ridgeway and Michael C Mozer. Learning deep disentangled embeddings with the t-statistic loss. arXiv preprint arXiv:1802.05312, 2018.

[50] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. Proceedings of the IEEE, 109(5):612–634, 2021.

[51] Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. arXiv preprint arXiv:1711.00848, 2017.
A Appendix

A.1 Causal ordering

Suppose we are given (high-dimensional) \( X = (X_1, \ldots, X_d) \) (think of \( X \) as an image with pixels \( X_1, \ldots, X_d \)), from which we should construct \( S_1, \ldots, S_n \) \((n \ll d)\) as well as causal mechanisms

\[
S_i := f_i(\mathbf{PA}_i, U_i), \quad (i = 1, \ldots, n).
\]  

(2)

To this end, we first use an encoder \( f_{\text{enc}} : \mathbb{R}^d \rightarrow \mathbb{R}^n \) taking \( X \) to a latent bottleneck representation comprising \( U = (U_1, \ldots, U_n) \). The next step is a map \( f(U) \) implementing the structural assignments \( f_1, \ldots, f_n \) as a function of \( U \). We construct it as follows: we evaluate the \( f_i \) of a root node \( i \), i.e., \( f_i \) depends only on \( U_i \). In the step, we evaluate any node \( j \) which depends only on its \( U_j \) and possibly other variables that have already been computed. We iterate until there are no nodes left. This terminates (since the graph is acyclic) and yields a unique \( f(U) \), but the order \( \pi(i) \) in which the \( f_i \) get evaluated need not be unique. It is referred to as a causal or topological ordering \([13]\), satisfying \( \pi(i) < \pi(j) \) whenever \( j \) is a descendant of \( i \). This embeds the SCM into the network starting from the bottleneck \( U = (U_1, \ldots, U_n) \) with the \( U_i \) feeding into subsequent computation layers according to a causal ordering. This structure reflects the fact that the root node(s) in the DAG only depend on "their" noise variables, while later ones depend on their noise and those of their parents, and so on. Finally, we apply a decoder \( f_{\text{dec}} : \mathbb{R}^n \rightarrow \mathbb{R}^d \). The system can be trained using reconstruction error to satisfy \( f_{\text{dec}} \circ f \circ f_{\text{enc}} \approx \text{id} \) on the observed images.

Recall that for a causally sufficient system, the set of noises \( U_1, \ldots, U_n \) are assumed to be jointly independent. If, in contrast, only a subset of the causal variables are modelled, then the noises will in the generic case be dependent. We would expect that the architectural bias implemented by the structural decoder, however, may still be a sensible one.

A.2 Training Procedure

A.2.1 Architecture Details

As described in the main paper, the basic convolutional backbone of all models is the same. For the smaller datasets, 3D-Shapes and the MPI3D datasets (where observations are 64x64 pixels), the encoder and decoder each have 12 convolutional blocks. Each block has a convolutional layer with 64 channels and a kernel size of 3x3 and stride of 1 (unless otherwise specified), followed by a group normalization layer and then a MISH nonlinearity \([44]\). In the encoder, the features are downsampled using a 2x2 Max Pooling layer right after the convolution every third layer starting with the first one and the first convolution layer uses a kernel size of 5x5. In the decoder, every third convolution layer is immediately preceded by a 2x2 bilinear upsampling. For our structured modules (SAE and AdaAE), the specified number of Str-Tfm layers are placed evenly in between the convolution blocks. For SAE models, the latent space is always split evenly between Str-Tfm layers, and each layer uses a three hidden layer network to process the latent space segment into the scale and bias vectors which are then applied to all pixels individually of the features. For the VLAE models, the inference and generative ladder rungs each also have a three hidden layers to process the features into and out of the separate latent space segments respectively.

While the latent space was always 12 dimensional for 3D-Shapes and MPI3D, for Celeb-A we use a 32 dimensional latent space. For Celeb-A, we also expand the 12 block backbone to 16 blocks and double the filters per convolution layer to 128. The exact sizes and connectivity of the models can be seen in the configuration files of the attached code, but overall, each of the 3D-Shapes and MPI3D models have approximately 1-1.2M trainable parameters, while for CelebA the models have 6-7M parameters.

A.2.2 Training Details

All models used the same training hyperparameters, which included using an Adam optimizer with a learning rate of 0.0005 and momentum parameters of \( \beta_1 = 0.9 \) and \( \beta_2 = 0.999 \). For the smaller datasets (3D-Shapes, MPI3D) the models were trained for 100k iterations and a batch size of 128, while for Celeb-A the models were trained for 200k iterations and a batch size of 32. The hyperparameters for the RFD dataset the same as for Celeb-A, except that the number of channels per convolution layer was doubled and the learning rate was decreased by a factor of 10.
The models are implemented using Pytorch and were trained on the in-house computing cluster using Nvidia V100 32GB GPUs, so that training a single model takes about 3-4 hours on the smaller datasets and 7-10 hours for CelebA.

A.3 Additional Results

A.3.1 3D-Shapes

Figure 9: 3D-Shapes reconstruction quality comparison between models using the reconstruction loss (binary cross entropy) and the Fréchet Inception Distance (FID) between the original and reconstructed observations (lower is better for both). Each "x" is a model trained with a unique random seed using the architecture/regularization corresponding to the color. The performance of all the seeds are averaged and plotted as circles "o". Firstly, this plot shows nicely how the reconstruction FID (y-axis) can help quantify the quality of the reconstructed samples when a pixelwise comparison (x-axis) is saturated. Next, the multiple seeds help identify two different regimes of performance, one including all the models between the AE (orange) and the SAE-12 (dark blue), while the other regime starts at the \( \beta \)-VAE (pink) and reaches to the VLAE-12 (red). These regimes separate the models that use the variational regularization loss from the models that only use a reconstruction loss (or a regularization on the aggregated posterior like the WAE). Lastly, the SAE-12 consistently performs best both in terms of reconstruction FID and the pixelwise loss.
Figure 10: Comparison of the hybrid and prior-based sampling method for all different models. Each bar corresponds to a unique random seed. (lower is best)

Table 1: Disentanglement and Completeness scores for 3D-Shapes. The DCI-d metric corresponds to the DCI-disentanglement score and DCI-c to the completeness score [46]. IRS is a similar disentanglement metric [47], the MIG is the Mutual Information Gap [48], the SAP score is the Separated Attribute Predictability score [51], and ModExp refers to the Modularity Explicitness score [49]. (for all these metrics higher is better)
Figure 11: Several disentanglement metrics for all the models and each of the seeds. (for all these metrics higher is better)
Figure 12: Latent Traversals of several models for 3D-Shapes. Each row shows the generated image when varying the corresponding latent dimension while fixing the rest of the latent vector. For the SAE and VLAE models, the groups of dimensions that are fed into the same Str-Tfm layer (or ladder rung) are grouped together. Note the disentangled segments achieved by the SAE models and the consistent ordering of factors of variation.

A.3.2 Extrapolation

We present results on a variant of the extrapolation experiment discussed in section 3.2. Instead of modifying the shape in the initial training dataset, we remove the two most extreme camera angles in either direction (removing 4/15 of the full dataset).

As seen from figure 13 in this setting the warping and generally lower fidelity experienced by only updating the encoder compared to updating the decoder is very apparent.
Table 2: Disentanglement and Completeness scores for MPI3D-Toy. (for all these metrics higher is better)

| Model  | DCI-d | IRS | MIG  | SAP  | ModExp | DCI-c |
|--------|-------|-----|------|------|--------|-------|
| SAE-12 | 0.642 | 0.370 | 0.487 | 0.287 | 0.938 | 0.566 |
| SAE-6  | 0.454 | 0.553 | 0.094 | 0.060 | 0.918 | 0.404 |
| SAE-4  | 0.316 | 0.535 | 0.119 | 0.078 | 0.919 | 0.283 |
| SAE-3  | 0.380 | 0.461 | 0.113 | 0.068 | 0.933 | 0.359 |
| SAE-2  | 0.252 | 0.557 | 0.030 | 0.011 | 0.908 | 0.252 |
| VLAE-12| 0.414 | 0.667 | 0.323 | 0.201 | 0.909 | 0.518 |
| VLAE-6 | 0.436 | 0.623 | 0.277 | 0.102 | 0.927 | 0.541 |
| VLAE-4 | 0.415 | 0.591 | 0.182 | 0.120 | 0.936 | 0.546 |
| VLAE-3 | 0.301 | 0.557 | 0.151 | 0.084 | 0.877 | 0.380 |
| VLAE-2 | 0.262 | 0.634 | 0.130 | 0.058 | 0.936 | 0.311 |
| AdaAE-12 | 0.208 | 0.546 | 0.080 | 0.061 | 0.919 | 0.191 |
| AE     | 0.186 | 0.632 | 0.043 | 0.020 | 0.911 | 0.170 |
| VAE    | 0.093 | 0.621 | 0.078 | 0.044 | 0.861 | 0.108 |
| βVAE   | 0.046 | 0.987 | 0.004 | 0.051 | 0.998 | 0.051 |
| WAE    | 0.203 | 0.633 | 0.028 | 0.023 | 0.904 | 0.171 |

Figure 13: Same as figure 8 except for the camera angle setting. Although the results are generally very consistent, note how much better the SAE-12 model performs than the VLAE-12 in this setting.

A.3.3 MPI3D-Toy
Figure 14: Latent Traversals of several models for MPI3D-Toy. Each row shows the generated image when varying the corresponding latent dimension while fixing the rest of the latent vector. For the SAE and VLAE models, the groups of dimensions that are fed into the same Str-Tfm layer (or ladder rung) are grouped together. Note the disentangled segments achieved by the SAE models and the consistent ordering of factors of variation.
### A.3.4 MPI3D-Sim

Table 3: Disentanglement and Completeness scores for MPI3D-Sim. (for all these metrics higher is better)

| Model       | DCI-d | IRS | MIG | SAP | ModExp | DCI-c |
|-------------|-------|-----|-----|-----|--------|-------|
| SAE-12      | 0.411 | 0.508 | 0.238 | 0.153 | 0.930  | 0.389 |
| SAE-6       | 0.294 | 0.479 | 0.052 | 0.037 | 0.928  | 0.300 |
| SAE-4       | 0.380 | 0.519 | 0.139 | 0.097 | 0.927  | 0.356 |
| SAE-3       | 0.269 | 0.482 | 0.046 | 0.027 | 0.902  | 0.256 |
| SAE-2       | 0.119 | 0.536 | 0.019 | 0.017 | 0.930  | 0.121 |
| VLAE-12     | 0.220 | 0.634 | 0.093 | 0.064 | 0.863  | 0.282 |
| VLAE-6      | 0.290 | 0.575 | 0.051 | 0.041 | 0.797  | 0.279 |
| VLAE-4      | 0.372 | 0.669 | 0.174 | 0.094 | 0.945  | 0.394 |
| VLAE-3      | 0.242 | 0.554 | 0.135 | 0.091 | 0.892  | 0.302 |
| VLAE-2      | 0.127 | 0.662 | 0.060 | 0.061 | 0.868  | 0.154 |
| AdaAE-12    | 0.159 | 0.481 | 0.022 | 0.013 | 0.893  | 0.129 |
| AE          | 0.157 | 0.526 | 0.033 | 0.026 | 0.855  | 0.143 |
| VAE         | 0.070 | 0.850 | 0.056 | 0.032 | 0.828  | 0.079 |
| βVAE        | 0.060 | **0.850** | 0.054 | 0.014 | 0.926  | 0.065 |
| WAE         | 0.129 | 0.548 | 0.033 | 0.018 | 0.881  | 0.120 |
Figure 15: Latent Traversals of several models for MPI3D-Sim. Each row shows the generated image when varying the corresponding latent dimension while fixing the rest of the latent vector. For the SAE and VLAE models, the groups of dimensions that are fed into the same Str-Tfm layer (or ladder rung) are grouped together. Note the disentangled segments achieved by the SAE models and the consistent ordering of factors of variation.
### A.3.5 MPI3D-Real

| Model   | DCI-d | IRS  | MIG  | SAP  | ModExp | DCI-c |
|---------|-------|------|------|------|--------|-------|
| SAE-12  | 0.314 | 0.543| 0.118| 0.079| 0.908  | 0.272 |
| SAE-6   | 0.295 | 0.535| 0.074| 0.053| 0.879  | 0.277 |
| SAE-4   | 0.270 | 0.523| 0.085| 0.034| 0.876  | 0.264 |
| SAE-3   | 0.306 | 0.495| 0.119| 0.072| 0.884  | 0.300 |
| SAE-2   | 0.130 | 0.527| 0.025| 0.020| 0.908  | 0.130 |
| VLAE-12 | 0.291 | 0.579| 0.217| 0.129| 0.914  | 0.332 |
| VLAE-6  | 0.410 | 0.668| 0.279| 0.183| 0.863  | 0.414 |
| VLAE-4  | 0.235 | 0.513| 0.113| 0.046| 0.899  | 0.223 |
| VLAE-3  | 0.249 | 0.611| 0.147| 0.074| 0.897  | 0.255 |
| VLAE-2  | 0.126 | 0.644| 0.074| 0.037| 0.914  | 0.170 |
| AdaAE-12| 0.164 | 0.530| 0.034| 0.017| 0.952  | 0.147 |
| AE      | 0.143 | 0.563| 0.048| 0.030| 0.858  | 0.132 |
| VAE     | 0.080 | 0.602| 0.020| 0.004| 0.875  | 0.085 |
| $\beta$VAE | 0.090 | 0.659| 0.021| 0.014| 0.869  | 0.108 |
| WAE     | 0.159 | 0.587| 0.042| 0.038| 0.837  | 0.150 |

Table 4: Disentanglement and Completeness scores for MPI3D-Real. (for all these metrics higher is better)
Figure 16: Latent Traversals of several models for MPI3D-Real. Each row shows the generated image when varying the corresponding latent dimension while fixing the rest of the latent vector. For the SAE and VLAE models, the groups of dimensions that are fed into the same Str-Tfm layer (or ladder rung) are grouped together. Note the disentangled segments achieved by the SAE models and the consistent ordering of factors of variation.
A.3.6 Celeb-A

(a) Original samples (from test set)

(b) SAE-16 Reconstructions

(c) AdaAE-16 Reconstructions
(d) VLAE-16 Reconstructions

(e) VAE Reconstructions

(f) AE Reconstructions
(d) VLAE-16 Prior Sampling

(e) VAE Hybrid Sampling

(f) VAE Prior Sampling
A.3.7 RFD

(a) SAE-16

(b) VLAE-16
Figure 19: Latent traversals for the RFD dataset. Each row corresponds to a 1D traversal of the corresponding latent dimension while the other latent dimensions are fixed. Note the ordering of information in the more structured models like the SAE-16 and VLAE-16.