Structural Autoencoders Improve Representations for Generation and Transfer

Anonymous Author(s)
Affiliation
Address
email

Abstract

We study the problem of structuring a learned representation to significantly improve performance without supervision. Unlike most methods which focus on using side information like weak supervision or defining new regularization objectives, we focus on improving the learned representation by structuring the architecture of the model. We propose a self-attention based architecture to make the encoder explicitly associate parts of the representation with parts of the input observation. Meanwhile, our structural decoder architecture encourages a hierarchical structure in the latent space, akin to structural causal models, and learns a natural ordering of the latent mechanisms. We demonstrate how these models learn a representation which improves results in a variety of downstream tasks including generation, disentanglement, and transfer using several challenging and natural image datasets.

1 Introduction

Deep learning has facilitated strong results on a plethora of interesting 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 learning more general representations which can help transfer to a variety of downstream tasks [3, 4, 5]. Here deep learning provides a flexible paradigm to train complex architectures based on autoencoders [6, 7] allowing us to embed powerful inductive biases into our models to further structure the representations making them more useful. However, it is still largely open what kinds of structure in a representation are most effective for transfer learning and how to learn such structures without supervision [8]. One direction that may contribute to an answer is causal inference, as it focuses on identifying the underlying (causal) mechanisms that generated the observations, instead of relying on (possibly spurious) correlations [9, 10, 11].

With the versatility of deep learning in one hand, and the conceptual insights of causality in 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 separates and orders the underlying factors of variation in the data.
- We extend our SAEs by structuring the architecture of the encoder to include a self-attention mechanism to produce the *Attention-Structural autoencoder* (ASAE - pronounced "assay"), which makes it possible to understand where information in the latent space comes from in the observation.
- We provide a sampling method that can be used for autoencoder based generative models which does not use an explicit regularization objective and instead relies on independence in the latent space.
- We investigate the structure of the representation learned by our models to understand how an attention mechanism in the encoder can improve the interpretability of the representation,

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

and how the hierarchy of mechanisms learned by the structural decoder can produce a both disentangled and continuous representation.

- We evaluate our architectures against several common baselines using challenging datasets including natural images on reconstruction, generation, disentanglement, fairness, abstract visual reasoning and transfer.

Related Work The most popular autoencoder based method is the Variational Autoencoder (VAE) [12] and the closely related βVAE [13]. These methods focus on matching the distribution in the latent space to a known prior distribution by regularizing the reconstruction training objective. Although this structure is convenient for generative modelling and even tends to disentangle the latent space to some extent, it comes at the cost of somewhat blurry images due to mode collapse and holes in the latent space [8, 13, 14, 15]. Our more structured architectures are related to the Variational Ladder autoencoder (VLAE) [16] which separates the latent space into separate chunks each of which is processed at different levels of the encoder and decoder. However, unlike VLAEs, the attention mechanism in our ASAEs explicitly produces an attention map to make the model more interpretable and our Str-Tfm layers in the structural decoder integrate information from the latent space in a more structured way, similarly to the Ada-in layers from Style-GANs [17].

2 Method

Causal Representation Learning Graphical causal modeling builds on random variables

$$S_i := f_i(\mathbf{PA}_i, U_i), \quad (i = 1, \dots, n), \tag{1}$$

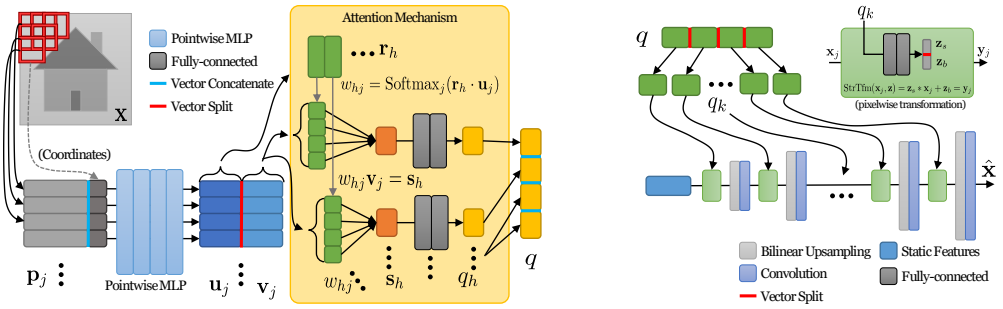
connected by a directed acyclic graph (DAG) whose edges represent direct causation. Each S_i is computed using a function f_i depending on its parents $\mathbf{PA}_i$ in the graph and on an *unexplained* noise variable U_i, ensuring that the "mechanism" (1) can represent any conditional distribution $p(S_i|\mathbf{PA}_i)$. The noises $U_1, \dots, U_n$ are assumed to be jointly independent. The DAG along with the mechanisms (1) is referred to as a Structural Causal Model (SCM) [9]. 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).

Real-world observations, however, are often not structured into meaningful causal variables and mechanisms to begin with. One may attempt to try and learn a *representation* consisting of causal variables or disentangled "factors" which are statistically independent [13]. However, in an SCM it is not the S_i that should be independent, but the U_i. For this reason we believe that the representation should 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) [18, 11].

Given (high-dimensional) $X = (X_1, \dots, X_d)$, we first use an **encoder** $f_{enc} : \mathbb{R}^d \to \mathbb{R}^n$ taking X to a latent representation $U = (U_1, \dots, U_n)$. 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 those of their parents, and so on. The depth in the network thus corresponds to a causal ordering (cf. Figure 1b below). Finally, we apply a **decoder** f_{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 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.



(a) Attention Encoder (b) Structural Decoder

Figure (1) An overview of the proposed architectures and how they are used to extract the latent vector q from input observation $\mathbf{x}$ in 1a, and then recover the reconstruction $\hat{\mathbf{x}}$ from the latent vector q in 1b. Note that j corresponds to the patch/pixel index.

Structural Decoders Our *structural decoder* splits the latent vector into $K \in \{4, 6, 12\}$ segments of $\{3, 2, 1\}$ dimensions, respectively, and each segment is applied to its own *Structural-Transform* (Str-Tfm) layer (as seen in Figure 1b).

One way to interpret how this architecture structures the latent space is that each Str-Tfm layer acts just like an f_i in equation 1 by integrating the information from a latent variable U_i to transform the features $\mathbf{PA}_i$ from earlier layers. Consequently, the more high-level, abstract factors of variation may be processed deeper in the decoder in the first few layers, while the more low-level, linear features are treated towards the end of the model, thereby resulting in a hierarchical structure in the latent space.

Attention Encoders We propose including a self-attention mechanism in the encoder to make the latent space more interpretable. The encoder employs $H \in \{4, 6, 12\}$ attention heads to select patches from the input image to process based on an activation mask from self-attention [19]. The input image is split into overlapping 4x4 patches (represented with red boundaries in Figure 1a) each of which are processed independently, with the corresponding position coordinates of the patch concatenated to break the translation invariance as in [20]. The attention heads non-competitively extract features using a soft self-attention mechanism across all patches. Finally, each head encodes the information it extracted into $\{3, 2, 1\}$ dimensions (corresponding to q_h), depending on the number of heads such that when concatenated the latent vector has 12 dimensions (see Figure 1a).

This architecture links up parts of the latent vector with parts of the observations and helps learning the encoder. This is also why we propose only using a single self-attention layer, unlike other transformer-style architectures [19]. Since each attention head specifies an activation map over the whole input image to encode information in its part of the latent space, we can explicitly see where in the image the information contained in each segment of the latent space originated. Additionally, if a part of the image is particularly informative, multiple heads can focus on it because the attention mechanism is non-competitive, allowing specialization without competition. The ASAE architecture is inspired by recurrent independent mechanisms [21], except that instead of the attention being over different events in a dynamical system the model attends to different factors of variation in a distribution over images.

Generation For generative modelling, 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 by estimating the difference between the latent distribution and some desired prior distribution [12, 22]. However, in practice, regularization techniques can fail to match the prior perfectly, leading to model collapse and blurry samples [23].

We suggest an alternative sampling method. Inspired by [24], we refer to it as *hybrid* sampling: The model keeps a small set of N ($= 128$) latent vectors stored (much smaller than the training dataset). 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. This, in principle, allows the model to generate N^D unique latent vectors. It works best if the latent dimensions are independent from another (consistent with the causal view of the latent variables as noises driving an SCM), which is much milder than assuming the latent distribution matches an isotropic normal, and holds especially well for disentangled representations. Note that hybrid sampling is directly applicable to

any learned representation even if it does not have a prior to sample from, however the generated sample quality can be poor if the representation is not sufficiently disentangled. Alternatively, one could always perform a density estimation step on the latent vectors after training, and then sample from that density.

3 Experiments

We train our proposed architectures as well as several baselines on three different datasets: 3D-Shapes [25], and two variants ("toy" and "real") of the MPI3D Disentanglement dataset [26], as well as Celeb-A [27]. 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) [28]. 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 each of the methods using a variety of different metrics to elucidate the quality of generated samples (using our "hybrid" sampling method as well as the standard "prior" sampling): disentanglement [29, 30], fairness [31], and abstract visual reasoning capabilities [32].

Finally, we investigate how well our proposed architectures enable the learned representations to adapt to new observations in transfer experiments, described below.

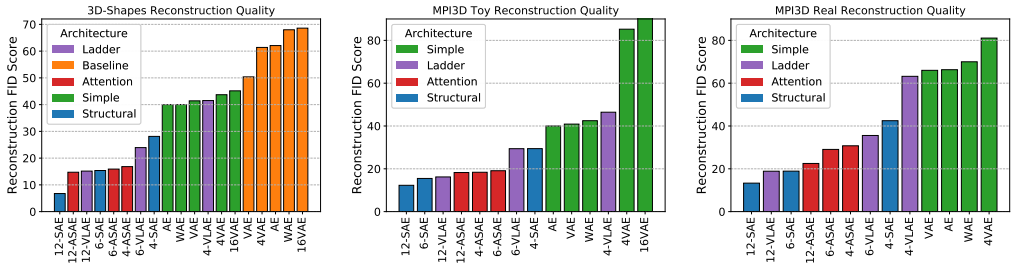
Models All models have similar sizes (about 1M trainable parameters) with a 12 dimensional latent space (32 for CelebA), and are trained with the same hyperparameters for 100k iterations, so that the only significant difference is the architecture and, depending on the model, the regularization (for details, see Supplement). All the models were trained on the same training set, hyperparameters were tuned for all methods using the validation set, and all results presented here are on the test set (using samples not seen during training or validation).

We use three kinds of baseline architectures. The first is called "baseline" and features the default convolutional architecture used in [8]. The next kind of baseline is called "simple" and uses the same basic "hourglass" architecture but with several minor improvements. For the "baseline" and "simple" architectures, we train unregularized autoencoders ("AE"), Wasserstein-autoencoders ("WAE") [22], VAEs, and several βVAEs with varying β parameters (denoted βVAE or, for instance, 4VAE if $\beta = 4$). The final baseline is the VLAE which also uses a structured architecture to learn a hierarchical representation. We evaluate our experiments with an $L \in \{4, 6, 12\}$ layer VLAE model (denoted "L-VLAE" or "LL"), which uses very similar hyperparameters as our ASAE models. For our novel architectures, the models using a "simple" encoder + a structural decoder are denoted "K-SAE" (or "KK" for short), while the models using an attention encoder + a $K = 12$ structural decoder are called "H-ASAE" (or "HH" for short).

Transfer For the 3D-Shapes and both MPI3D variants, we select one degree of freedom, in this case the shape of the object in the image, and split the training and test set along that factor of variation into a larger one called "stage 1" and two smaller datasets (denoted "stage 2" and "stage 3"). In our case, stage 1 for 3D-Shapes contains all boxes and cylinders, while stage 2 only contains balls, and stage 3 only capsules. For both MPI3D datasets, we use all shapes except for the boxes and cones in stage 1, followed by boxes only in stage 2, and only cones in stage 3. Lastly, we call the "overall" dataset the union of all three stages. The models are trained on each stage in series with 80k iterations for stage 1, and 20k for stages 2 and 3. The samples, Using this setup, we can evaluate how much a model can adjust the learned representation to include new objects.

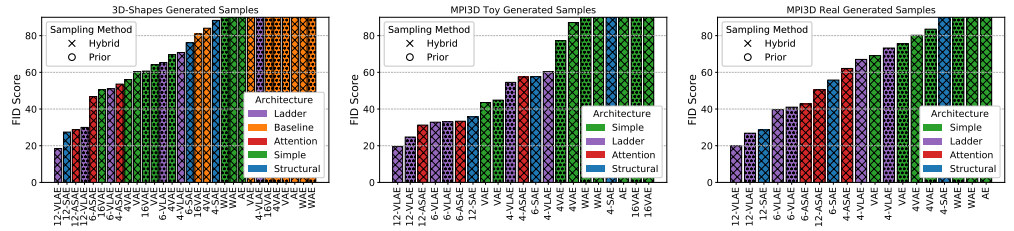
4 Results

In terms of reconstruction quality, the 12-SAE model consistently outperforms all baselines and other variants (seen in Figure 2). This suggests that although the attention encoders outperform most baselines, it appears the SAE architecture is sufficient for learning high quality representations, so the attention mechanism may somewhat interfere with the structural decoder. Additionally, from comparing the "simple" and "baseline" results, the FID is a useful metric for differentiating the quality



(a) 3D-Shapes (b) MPI3D Toy (c) MPI3D Real

Figure (2) Results on the reconstruction quality for all models and datasets (using the test set). The "Baseline" and "Simple" models correspond to traditional "hourglass" CNN architectures, while the "Structural" and "Attention" models use our novel architectures to further structure the learned representation. For both reconstruction loss and FID, lower is better. Our proposed architectures consistently perform best in both of the reconstruction metrics.



(a) 3D-Shapes (b) MPI3D Toy (c) MPI3D Real

Figure (3) Comparison of the generative models using different sampling methods. Note that our structured models (Structural and Attention) perform well without having to regularize the latent space towards a prior. Note, moreover, that our hybrid sampling consistently improves VAE samples, compared to sampling from the prior.

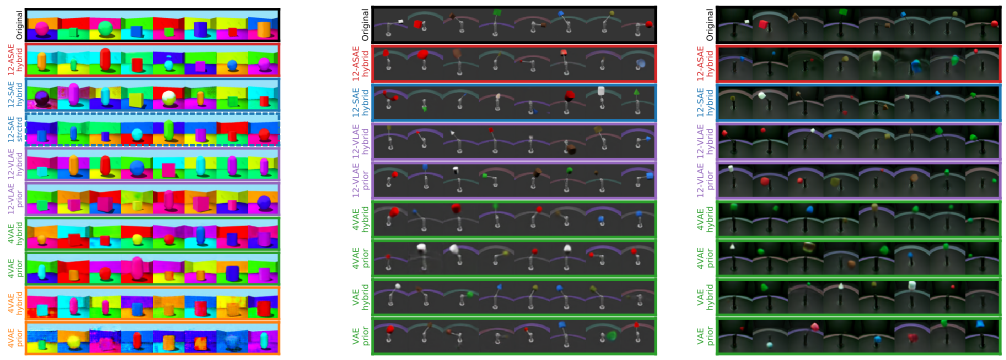
of the models beyond a simple pixelwise loss function (for further discussion, see the supplementary material).

Figure 4 shows some examples of generated samples for several models and different sampling methods. Interestingly, both the sampling from the prior, denoted "prior", and our "hybrid" sampling strategy appears to generate quality samples. At first glance, the 12-SAE model appears to be mixing colors when using the "hybrid" sampling, however, as we discuss in section 4, our proposed naive "hybrid" sampling method actually neglects some of the useful structure learned by the structural decoder. When taking the additional structure into account (as seen by the 12-SAE "strctrd" row in Figure 4a), the generated samples become virtually indistinguishable from real samples from the dataset (in the top row).

Quantitatively, we see exactly how much of a difference the architecture makes, and perhaps more surprisingly, the sampling technique (seen in Figure 3). For almost every model, our hybrid sampling actually improves sampling results. This opens the door to deep generative models that do not require some aggressive regularization, which often requires carefully tuned curricula, to enable 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.

Table 1 summarizes several additional metrics of our models: disentanglement, fairness (measured as in [31]) and abstract visual reasoning [32]. We see that the more structured models, especially our SAEs and ASAEs, consistently produce more disentangled, and more fair representations. For more results, including an analysis of the results on Celeb-A, please see Supplement.

Hierarchical Structure To get a rough idea of how the representations learned using the structural decoders are different from more conventional architectures, Figure 5 shows the one dimensional traversals (i.e., each row corresponds to the decoder outputs when varying a single latent dimension



(a) 3D-Shapes (b) MPI3D Toy (c) MPI3D Real

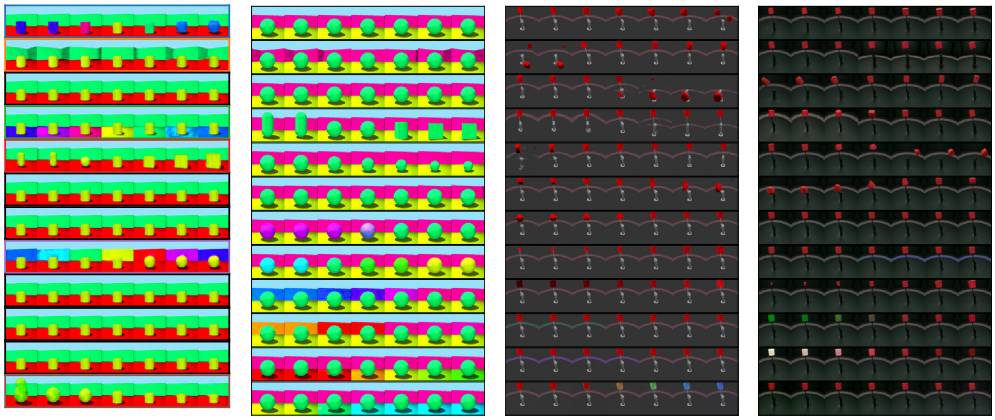
Figure (4) Qualitative comparison of samples generated by different model architectures (border color) and sampling techniques (labeled on the left side). Our 12-SAE model produces high quality samples using hybrid sampling, even though it was not regularized towards a prior. For 3D-Shapes (left), we can also use the structure of the representation in our hybrid sampling technique to further improve the samples as seen in "strctrd" sampling and the dashed border (see section 4 for details).

	3D-SHAPES				MPI3D TOY			MPI3D REAL		
	DCI	IRS	FAIR.	AVR (%)	DCI	IRS	FAIR.	DCI	IRS	FAIR.
AE	0.19	0.51	2.62	99.90 $\pm$ 0.04	0.07	0.63	2.35	0.07	0.61	2.19
VAE	0.89	0.74	5.19	91.32 $\pm$ 6.14	0.34	0.62	2.50	0.22	0.64	2.33
βVAE	0.54	0.64	4.11	95.42 $\pm$ 5.98	0.25	0.66	2.35	0.27	0.73	2.29
12-VLAE	0.85	0.83	**7.90**	90.86 $\pm$ 6.46	0.44	0.60	2.62	0.20	0.58	2.23
6-VLAE	0.70	0.64	7.10	86.44 $\pm$ 0.26	0.36	0.61	2.52	0.34	0.60	2.32
4-VLAE	0.53	0.73	3.58	86.13 $\pm$ 0.46	0.40	0.69	2.53	**0.40**	0.66	2.33
12-SAE	**0.97**	**0.89**	7.72	99.71 $\pm$ 0.35	**0.57**	0.63	**2.81**	0.30	0.59	2.32
6-SAE	0.80	0.81	6.62	95.52 $\pm$ 6.27	0.50	0.63	2.80	0.33	0.63	2.37
4-SAE	0.70	0.68	5.38	**99.95** $\pm$ **0.04**	0.39	0.63	2.59	0.16	0.60	2.28
12-ASAE	0.91	0.82	6.57	98.16 $\pm$ 2.52	0.46	0.70	2.59	0.38	**0.79**	**2.48**
6-ASAE	0.59	0.72	4.34	96.46 $\pm$ 2.16	0.50	0.71	2.58	0.27	0.71	2.35
4-ASAE	0.66	0.76	4.40	94.81 $\pm$ 7.09	0.46	**0.72**	2.42	0.29	0.75	2.45

Table (1) The DCI metric corresponds to the DCI-disentanglement score [29], IRS is a similar disentanglement metric [30], fairness is the $-\log$ of the unfairness score [31], and the abstract visual reasoning (AVR) score is defined in [32]. For all of these metrics larger is better. Note that the more structured models (in particular our SAE, but also the VLAE) always outperform simpler baselines, and for the more challenging MPI3D-Real dataset, the attention mechanism in the ASAE becomes more beneficial.

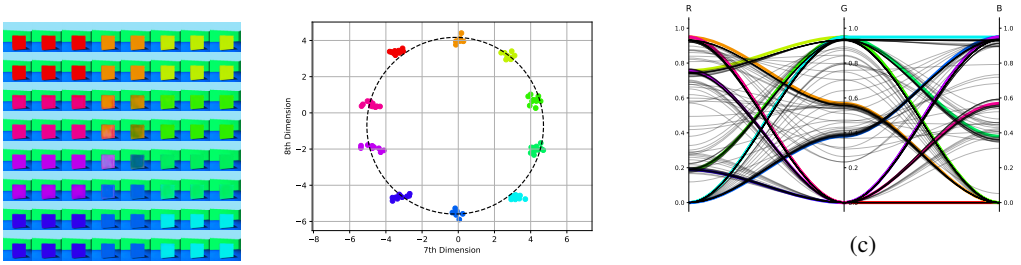
at a time). Note that for all of the models the traversals are presented in their original order. The traversals illustrate the hierarchical structure in the representation learned by the SAE models as the information for what one might consider higher-level factors of variation (such as object shape or viewpoint) are in the first few dimensions, all the degrees of freedom relating to color (low-level) appear towards the bottom.

The βVAE effectively disentangles the observations in the conventional sense, so that each latent dimension is independent and corresponds to at most one factor of variation. Meanwhile, the 12-SAE clearly uses *two* latent dimensions for each of the three color degrees of freedom (corresponding to object, wall, and floor color) in 3D-Shapes. Upon closer inspection, the 12-SAE model achieves a more desirable, and subtle structured representation. In fact, the 3D-Shapes dataset was generated by independently sampling one of ten hues for the object, floor, and wall. These hues have periodic structure, which may not be obvious to the casual observer. The 12-SAE model recognized the periodicity in the hues and consequently embedded each of the three periodic degrees of freedom into a 2D space, thereby using two latent dimension, in order to keep the representation continuous. The additional periodic structure and its benefits can be seen more clearly in Figure 6. Similarly the 12-SAE model separates two different components of the color in the MPI3D datasets: the hue and saturation, without any supervision.



(a) βVAE 3D-Shapes (b) 12-SAE 3D-Shapes (c) 12-SAE MPI3D-Toy (d) 12-SAE MPI3D-Real

Figure (5) Traversals of a baseline βVAE model and the structural decoder in their original order. Note the ordering of the information in the structural decoder models where higher level, nonlinear features are encoded in the first few dimensions, which are located deeper in the network, and the later dimensions contain more low-level information such as foreground or background color (for more traversals, see the supplement).

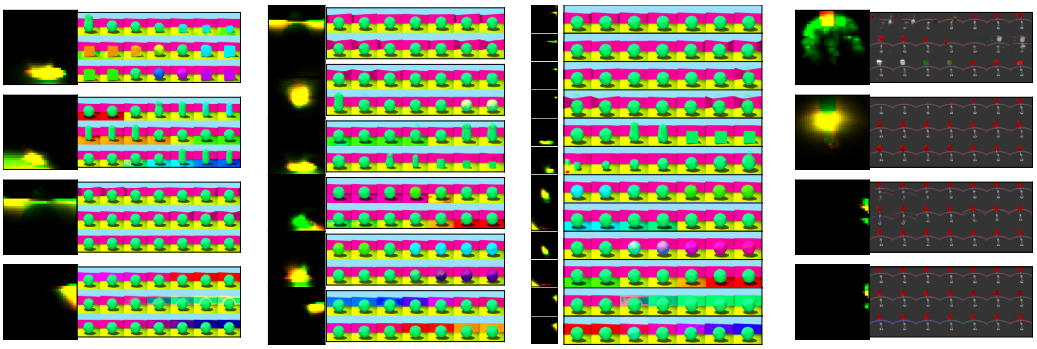


(a) (b) (c)

Figure (6) (a) 2D traversal of the latent space of a 12-SAE model. This shows the periodicity of the object hues that is learned by the model, and embedded in 2D without supervision. (b) Each point corresponds to the 2D projection of a latent vector from validation samples, where the point's color corresponds to the hue of the object. The dashed line is a best fit of a circle through all the points demonstrating the periodic structure. (c) A parallel coordinate plot showing the RGB values of the hues observed during training in their respective color, and the colors of the objects in the 2D traversal in black. Note that with our SAE model, the object hue changes gradually and the decoder can produce novel object colors never seen during training

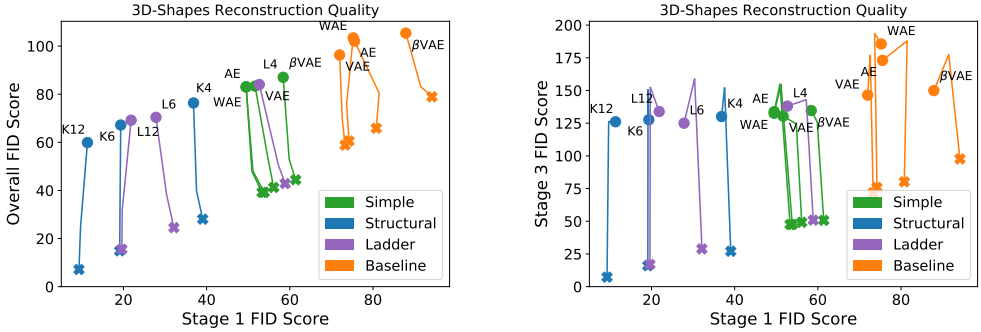
Specialization The attention heads in our ASAE architecture produce explicit attention maps over the input image, which can be used to identify where the information in each part of the latent vector comes from, as seen in Figure 7. For instance, for the MPI3D toy dataset seen in Figure 7d, the first head specializes on the object, the second on the object outline and surroundings, and the remaining two heads attend to the background. As the number of heads increases, as in Figures 7a to 7c, they learn to focus not only on a specific part of the image, but also on a specific factor of variation. This leads to some degree of redundancy where several heads may attend to the same part of the sample, but extract slightly different information, which would not be possible if all heads were competing as in some segmentation models.

Transfer We evaluate the transfer capabilities of our models in two ways: first, how the overall performance changes relative to the performance on the initial training set (from stage 1) as the model is updated in stage 2 and 3 (Figure 8a), and second, how the performance on only the final training set (stage 3) is affected (Figure 8b). Aside from the SAE models consistently outperforming the baselines, for the 12-SAE model, the performance on the stage 1 set is is not negatively affected by the stage 2 or 3 sets, suggesting that, unlike the other models, the 12-SAE does not overfit in the later stages, forgetting stage 1. Furthermore, the performance on the stage 3 set is not impeded by stage 2 for the 12-SAE model, suggesting that the representation can readily be adapted to a novel object as in stage 3.



(a) 4-ASAE (b) 6-ASAE (c) 12-ASAE (d) 4-ASAE

Figure (7) For each of the models above (a-d), the left column shows the activations of each of the attention heads where the activation for a single sample is in the red channel and the average activation over a batch of samples is in the green channel (so appearing yellow when they overlap). This shows that the heads are specializing on specific parts of the image and that as the number of heads increases, some heads focus on similar regions, but different information.



(a) Overall Performance (b) Performance on final test set

Figure (8) Performance in terms of FID of reconstructed samples of each of the models after the initial training on stage 1 only (denoted by •) vs. after updating the model with stage 2, and finally stage 3 (denoted by ×). For FID, lower is better. Note that in Figure 8a, for the K12 (12-SAE), K6 (6-SAE), and L12 (12-VLAE) models, the stage 1 set error does not increase after stage 2 and 3, each of which contain a novel object. Meanwhile, Figure 8b shows how performance on the samples in stage 3 is related to the performance on stage 1 samples over the course of training all three stages. Here, the 12-SAE model does not overfit to the intermediate stage 2 before the final stage 3.

5 Conclusion

We proposed autoencoder architectures motivated by structural causal models which learned hierarchical latent spaces that focus on identifying the underlying (causal) factors of variation, rather than merely focusing on the independence of latent dimensions. In addition, the "hybrid" sampling method opens the door to structuring the latent space well beyond matching some simple prior distribution without sacrificing the ability to generate high quality samples. The attention based encoder showed promise in specializing to meaningful parts of the image, suggesting that such an architecture could in principle be used to semantically segment images without supervision. We observed that the structural decoder consistently learned to order factors of variation into a hierarchical latent space that can include useful relationships between latent dimensions by focusing on the underlying mechanisms. Perhaps most significantly, for our SAE and ASAE models, impressive results in terms of generation, disentanglement, and transfer were achieved without explicitly regularizing the latent space, not even towards independence. While it is conceivable that this is largely achieved by biasing the decoder towards a causal generator inspired by SCMs, much remains to be understood as to just how a structure can be learned by architecture.

6 Broader impact

Statistical machine learning has in recent years focused heavily on optimizing the performance of systems on supervised i.i.d. problems. While this leads to impressive results on a number of problems, we believe that it has several shortcomings that apart from hampering scientific progress may also have broader impact:

1. the best performing supervised i.i.d. systems usually require large hand-labelled datasets, as well as significant compute resources. This creates an imbalance which makes it hard for researchers or engineers without these resources to compete. Note that it also creates a market for human labelling labor.[1]
2. major resources are invested into building supervised i.i.d. systems that are highly specialized and only applicable to constrained scenarios. This creates narrow task experts with non-transferable skills. Note that these systems are also prone to the *value alignment problem*: they optimize on a given task, without regard for unintended side consequences.[2]

The field of causal learning tries to mitigate the heavy reliance on single i.i.d. datasets and instead attempts to build systems that transfer knowledge and solve many tasks. This often comes in combination with an element of stronger interpretability of models (our paper above is an examples thereof). In the long term, we hope that causal learning will help build machines that approach "common sense" and enable rational and fair decision making in a variety of disciplines.

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.

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

[7] 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.

[8] 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.

[9] J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, New York, NY, 2nd edition, 2009.

[10] J. Peters, D. Janzing, and B. Schölkopf. *Elements of Causal Inference - Foundations and Learning Algorithms*. MIT Press, Cambridge, MA, USA, 2017.

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

[1] `https://www.kdnuggets.com/2017/06/acquiring-quality-labeled-training-data.html`

[2] Stuart Russell, *Human compatible: Artificial intelligence and the problem of control*, Penguin, 2019

[12] Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes, 2013.

[13] 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.

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

[15] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*, 2018.

[16] 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.

[17] 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.

[18] Raphael Suter, Djorđe Miladinović, Bernhard Schölkopf, and Stefan Bauer. Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness. *arXiv preprint arXiv:1811.00007*, 2018.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[20] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. In *Advances in Neural Information Processing Systems*, 2018.

[21] Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf. Recurrent independent mechanisms. *arXiv preprint arXiv:1909.10893*, 2019.

[22] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schölkopf. Wasserstein auto-encoders. In *6th International Conference on Learning Representations (ICLR)*, May 2018.

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

[24] Michel Besserve, Arash Mehrjou, Rémy Sun, and Bernhard Schölkopf. Counterfactuals uncover the modular structure of deep generative models, 2018. ICLR 2020.

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

[26] Muhammad Waleed Gondal, Manuel Wüthrich, Djorđe Miladinović, 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.

[27] 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.

[28] 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.

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

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

[31] Francesco Locatello, Gabriele Abbati, Thomas Rainforth, Stefan Bauer, Bernhard Schölkopf, and Olivier Bachem. On the fairness of disentangled representations. In *Advances in Neural Information Processing Systems*, pages 14584–14597, 2019.

[32] Sjoerd van Steenkiste, Francesco Locatello, Jürgen Schmidhuber, and Olivier Bachem. Are disentangled representations helpful for abstract visual reasoning? In *Advances in Neural Information Processing Systems*, 2019.