Topographic VAEs learn Equivariant Capsules

Anonymous Author(s)
Affiliation
Address
email

Abstract

In this work we seek to bridge the concepts of topographic organization and equivariance in neural networks. To accomplish this, we introduce the Topographic VAE: a novel method for efficiently training deep generative models with topographically organized latent variables. We show that such a model indeed learns to organize its activations according to salient characteristics such as digit class, width, and style on MNIST. Furthermore, through topographic organization over time (i.e. temporal coherence), we demonstrate how predefined latent space transformation operators can be encouraged for observed transformed input sequences – a primitive form of unsupervised learned equivariance. We demonstrate that this model successfully learns sets of equivariant features (i.e. "capsules") directly from sequences and achieves higher likelihood on correspondingly transforming test sequences. Equivariance is verified quantitatively by measuring the commutativity of the inference network and the sequence transformations. Finally, we show the model is capable of learning to be approximately equivariant to complex transformations, expanding upon the capabilities of existing group equivariant neural networks.

1 Introduction

Many parts of the brain are organized topographically. Famous examples are the ocular dominance maps and the orientation maps in V1. What is the advantage of such organization and what can we learn from it to develop better inductive biases for deep neural network architectures?

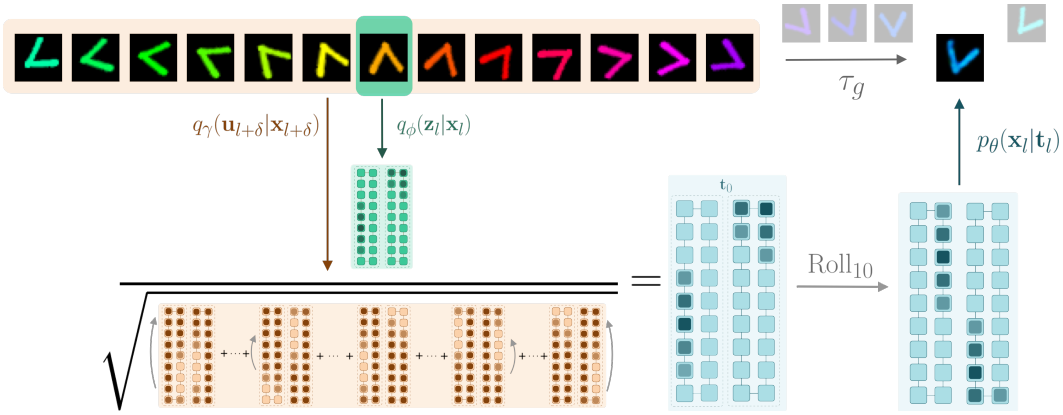

Figure 1: Overview of the Topographic VAE with shifting temporal coherence. The combined color/rotation transformation in input space τ_g becomes encoded as a Roll within the equivariant capsule dimension. The model is thus able decode unseen sequence elements by encoding a partial sequence and rolling activations within the capsules. We see this completes a commutative diagram.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

One potential explanation for the emergence of topographic organization is provided by the principle of redundancy reduction [1]. In the language of Information Theory, redundancy wastes channel capacity, and thus to represent information as efficiently as possible, the brain may strive to transform the input to a neural code where the activations are statistically maximally independent. In the machine learning literature, this idea resulted in Independent Component Analysis (ICA) which linearly transforms the input to a new basis where the activities are independent and sparse [2, 12, 28, 41]. It was soon realized that there are remaining higher order dependencies (such as correlation between absolute values) that can not be transformed away by a linear transformation. For example, along edges of an image, linear-ICA components (e.g. gabor filters) still activate in clusters even though the sign of their activity is unpredictable. This led to new algorithms that explicitly model these remaining dependencies through a topographic organization of feature activations [26, 42, 43, 51]. Such topographic models were shown to better match the generative structure of the data, yielding more efficient representations. Furthermore, the topographically organized features were reminiscent of pinwheel structures observed in V1, encouraging multiple comparisons with topographic organization in the biological visual system [27, 29, 39].

A second, almost independent body of literature developed the idea of "equivariance" of neural network feature maps under symmetry transformations. The idea of equivariance is that symmetry transformations define equivalence classes as the orbits of their transformations, and we wish to maintain this structure in the deeper layers of a neural network. For instance, for images, asserting a rotated image contains the same object for all rotations, the transformation of rotation then defines an orbit where the elements of that orbit can be interpreted as pose or angular orientation. When an image is processed by a neural network, we want features at different orientations to be able to be combined to form new features, but we want to ensure the relative pose information between the features is preserved for all orientations. This has the advantage that the equivalence class of rotations for the complex composite features is guaranteed to be maintained, allowing for the extraction of invariant features, a unified pose, and increased data efficiency. Such ideas are reminiscent of the capsule networks of [20, 21, 46], and indeed formal connections to equivariance have been made [38]. Interestingly, by explicitly building neural networks to be equivariant, we additionally see geometric organization of activations into these equivalence classes. The insight of this connection between topographic organization and equivariance hints at a possibility to encourage approximate equivariance from an induced topology in feature space.

To build a model, we need to ask what mechanisms could induce topographic organization of transformations in biological and artificial neural networks? We have argued that removing dependencies between latent representations of input data is a possible mechanism, and indeed, as discussed, there is a rich literature on how to achieve that unsupervised. However, for the more structured organisation of equivariant capsule representations, the usual approach is to hard-code this structure into the network, or to encourage it through regularization terms [4, 14]. To achieve this through *unsupervised learning*, we propose to incorporate another key inductive bias: "temporal coherence" [17, 23, 48, 52]. The principle of temporal coherence, or "slowness", asserts than when processing correlated sequences, we wish for our representations to change smoothly and slowly over space and time. Thinking of time sequences as symmetry transformations on the input, we desire features undergoing such transformations to be grouped into equivariant capsules. We therefore suggest that encouraging slow feature transformations to take place *within a capsule* could induce such grouping from sequences alone.

In the following sections we will explain the details of our Topographic VAE model which lies at the intersection of topographic organization, equivariance, and temporal coherence, thereby learning approximate equivariant capsules from sequence data completely unsupervised.

2 Related Work

The history of statistical models upon which this work builds is vast, including sparse coding [41], Independant Component Analysis (ICA) [2, 12, 28], Slow Feature Analysis (SFA) [49, 52], and many variants [25, 26]. Most related to this work are topographic generative models including Generative Topographic Maps [5], Bubbles [24], Topographic ICA [26], and the Topographic Product of Student's-t [43, 51]. Prior work on learning equivariant and invariant representations is similarly vast and also has a deep relationship with these generative models. Specifically, Independant Subspace Analysis [25], models involving temporal coherence [17, 23, 48, 52], and Adaptive Subspace Self

Organizing Maps [33] have all demonstrated the ability to learn invariant feature subspaces and even 'disentangle' space and time [18]. Our work assumes a similar generative model to these works while additionally allowing for efficient estimation of the model through variational inference [31, 45]. Although our work is not the first to combine Student's-t distributions and variational inference [6], it is the first to provide an efficient method to do so for Topographic Student's-t distributions.

Another line of work has focused on constructing neural networks with equivariant representations separate from the framework of generative modeling. Analytically equivariant networks such as Group Equivariant Neural Networks [11], and other extensions [54, 9, 53, 15, 44, 50, 16] propose to explicitly enforce symmetry to group transformations in neural networks through structured weight sharing. Other methods propose supervised and self-supervised methods to learn the equivariant or invariant structure from pairs of examples or transformations [4, 14, 13]. One related example in this category uses a group sparsity regularization term to similarly learn topographic features for the purpose of modeling invariance [30]. We believe the unsupervised approach presented in this paper provides a valuable alternative, principled view of how such structure could be learned in biological neural networks.

Furthermore, the idea of disentangled representations [3] has also been been connected to equivariance and representation theory in multiple recent papers [10, 8, 19, 7]. Our work shares a fundamental connection to this distributed operator definition of disentanglement, where the slow roll of capsule activations can be seen as the latent operator. Recently, the authors of [32] demonstrated that incorporating the principle of 'slowness' in a variational autoencoders (VAEs) yields the ability to learn disentangled representations from natural sequences. While similar in motivation, the generative model proposed in [32] is unrelated to topographic organization and equivariance, and is more aligned with traditional notions of disentanglement.

Finally, and importantly, in the neuroscience literature, another popular explanation for topographic organization arises as the solution to the 'wiring length' minimization problem [34]. Recently, models which attempt to incorporate wiring length constraints have been shown to yield topographic organization of higher level features, ultimately resembling the 'face patches' found in primates [37]. Interestingly, the model presented in this paper organizes activity based on the same statistical property (local correlation) as the wiring length proxies developed in [37], but from a generative modeling perspective, demonstrating a potential generative modeling solution to the same problem.

3 Background

The model in this paper is a first attempt at bridging two yet disjoint classes of models: Topographic Generative Models, and Equivariant Neural Networks. In this section, we will provide a brief background on these two frameworks.

3.1 Topographic Generative models

Inspired by Topographic ICA, the class of Topographic Generative models can be understood as generative models where the joint distribution over latent variables does not factorize into entirely independent factors, as is commonly done in ICA or VAEs, but instead has a more complex 'local' correlation structure. The locality is defined by arranging the latent variables into an n-dimensional lattice or grid, and organizing variables such that those which are closer together on this grid have greater correlation of activities than those which are further apart. In the related literature, activations which are nearby in this grid are defined to have higher-order correlation, e.g. correlations of squared activations (aka 'energy'), asserting that all first order correlations are removed by the initial ICA de-mixing matrix.

Such generative models can be seen as hierarchical generative models where there exist higher level independent 'variance generating' variables $\mathbf{V}$ which are combined locally to generate the variances $\boldsymbol{\sigma} = \phi(\mathbf{WV})$ of the lower level topographic variables $\mathbf{T} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}^2\mathbf{I})$, for an appropriate non-linearity ϕ. The variables $\mathbf{T}$ are thus independent conditioned on $\boldsymbol{\sigma}$. As we will show in the next section, the Topographic Product of Student's-t model can be seen to be a member of this class when $\mathbf{V} = \mathbf{U}^2$, $\mathbf{U} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\phi(\cdot) = \frac{\sqrt{\nu}}{\sqrt{\cdot}}$, where ν is the number of degrees of freedom.

Other related models which can be described under this umbrella include *Independent Subspace Analysis* (ISA) [25], and certain models which leverage the principle of *temporal coherence* [23]. ISA can be seen as a special case of a topographic generative model where all variables within a

predefined subspace (or 'capsule') share a common variance. Temporal coherence can be defined as the correlation of energy between time steps for a given variable and has been achieved in prior literature [24] by extending the topographic neighborhoods of the TICA model over the time dimensions. We will expand upon this idea Section 4.5 to induce latent space operators for observed transformations, thereby learning equivariant capsules.

3.2 Group Equivariant Neural Networks

Equivariance is the mathematical notion of symmetry for functions. A function is said to be an equivariant map if the the result of transforming the input and then computing the function is the same as first computing the function and then transforming the output. In other words, the function and the transformation commute. Formally, $f(\tau_\rho[\mathbf{x}]) = \Gamma_\rho[f(\mathbf{x})]$, where τ and Γ denote the (potentially different) operators on the domain and co-domain respectively, but are indexed by the same element ρ.

It is well known that convolutional maps in neural networks are translation equivariant, i.e., given a translation Γ_ρ (applied to each feature map separately) and a convolutional map $f(\cdot)$, we have $f(\Gamma_\rho[\mathbf{x}]) = \Gamma_\rho[f(\mathbf{x})]$. This can be extended to other transformations (e.g. rotation or mirroring) using Group convolutions (G-convolutions) [11]. As a result of the design of G-convolutions, feature maps that are related to each other by a rotation of the filter/input are grouped together. Moreover, a rotation of the input results in a transformation (i.e., a permutation and rotation) on the activations of each of these groups in the output. Hence, we can think of these equivalence class groups as capsules where transformations of the input only cause structured transformations *within* a capsule.

4 The Generative Model

The generative model proposed in this paper is based on the Topographic Product of Student's-t (TPoT) model as developed in [51, 43]. In the following, we will show how a TPoT random variable can be constructed from a set of independent univariate standard normal random variables, enabling efficient training through variational inference. Subsequently, we will construct a new model where topographic neighborhoods are extended over time, introducing temporal coherence and encouraging the unsupervised learning of equivariant subspaces we call 'capsules'.

4.1 The Product of Student's-t Model

We assume that that our observed data is generated by a latent variable model where the joint distribution over observed and latent variables $\mathbf{x}$ and $\mathbf{t}$ factorizes into the product of the conditional and the prior. The prior distribution $p_\mathbf{T}(\mathbf{t})$ is assumed to be a Topographic Product of Student's-t (TPoT) distribution, and we parameterize the conditional distribution with a flexible function approximator:

$$p_{\mathbf{X},\mathbf{T}}(\mathbf{x},\mathbf{t}) = p_{\mathbf{X}|\mathbf{T}}(\mathbf{x}|\mathbf{t})p_\mathbf{T}(\mathbf{t}) \qquad p_{\mathbf{X}|\mathbf{T}}(\mathbf{x}|\mathbf{t}) = p_\theta(\mathbf{x}|g_\theta(\mathbf{t})) \qquad p_\mathbf{T}(\mathbf{t}) = \mathrm{TPoT}(\mathbf{t};\nu) \tag{1}$$

The goal of training is thus to learn the parameters θ such that the marginal distribution of the model $p_\theta(\mathbf{x})$ matches that of the observed data. Unfortunately, the marginal likelihood is generally intractable except for all but the simplest choices of g_θ and $p_\mathbf{T}$ [42]. Prior work has therefore resorted to techniques such as contrastive divergence with Gibbs sampling [51] to train TPoT models as energy based models. In the following section, we instead demonstrate how TPoT variables can be constructed as a deterministic function of Gaussian random variables, enabling the use of variational inference and efficient maximization of the likelihood through the evidence lower bound (ELBO).

4.2 Constructing the Product of Student's-t Distribution

First, note that a univariate Student's-t random variable T with ν degrees of freedom can be defined as:

$$T = \frac{Z}{\sqrt{\frac{1}{\nu}\sum_i^\nu U_i^2}} \quad \text{with} \quad Z, U_i \sim \mathcal{N}(0,1) \;\; \forall i \tag{2}$$

Where Z and $\{U_i\}_{i=1}^\nu$ are independent standard normal random variables. If $\mathbf{T}$ is a multidimensional Student's-t random variable, composed of independent Z_i and U_i, then $\mathbf{T} \sim \mathrm{PoT}(\nu)$, i.e.:

$$\mathbf{T} = \left[\frac{Z_1}{\sqrt{\frac{1}{\nu}\sum_{i=1}^{\nu} U_i^2}}, \;\; \frac{Z_2}{\sqrt{\frac{1}{\nu}\sum_{i=\nu+1}^{2\cdot\nu} U_i^2}}, \;\; \cdots \;\; \frac{Z_n}{\sqrt{\frac{1}{\nu}\sum_{i=(n-1)\cdot\nu+1}^{n\cdot\nu} U_i^2}} \right] \sim \mathrm{PoT}(\nu) \tag{3}$$

Note that the Student's-t variable T is large when most of the $\{U_i\}_i$ in its set are small. We can therefore think of the $\{U_i\}_i$ as constraint violations rather then pattern matches: if the input matches all constraints $U_i \approx 0$, the corresponding T variables will activate [22].

4.3 Introducing Topography

To make the PoT distribution topographic, we strive to correlate the scales of T_j which are 'nearby' in our topographic layout. One way to accomplish this is by *sharing* some U_i-variables between neighboring T_j's. Formally, we define overlapping neighborhoods $\mathsf{N}(j)$ for each variable T_j and write:

$$\mathbf{T} = \left[\frac{Z_1}{\sqrt{\frac{1}{\nu}\sum_{i \in \mathsf{N}(1)} U_i^2}}, \quad \frac{Z_2}{\sqrt{\frac{1}{\nu}\sum_{i \in \mathsf{N}(2)} U_i^2}}, \quad \ldots \quad \frac{Z_n}{\sqrt{\frac{1}{\nu}\sum_{i \in \mathsf{N}(n)} U_i^2}} \right] \sim \mathrm{TPoT}(\nu) \tag{4}$$

With some abuse of notation, if we define $\mathbf{W}$ to be the adjacency matrix which defines our neighborhood structure, $\mathbf{U}$ and $\mathbf{Z}$ to be the vectors of random variables U_i and Z_j, we can write the above succinctly as:

$$\mathbf{T} = \left[\frac{Z_1}{\sqrt{\frac{1}{\nu} W_1 \mathbf{U}^2}}, \quad \frac{Z_2}{\sqrt{\frac{1}{\nu} W_2 \mathbf{U}^2}}, \quad \ldots \quad \frac{Z_n}{\sqrt{\frac{1}{\nu} W_n \mathbf{U}^2}} \right] = \frac{\mathbf{Z}}{\sqrt{\frac{1}{\nu}\mathbf{W}\mathbf{U}^2}} \sim \mathrm{TPoT}(\nu) \tag{5}$$

Due to non-linearities such as ReLUs, it is beneficial to allow the Z variables to model the mean and scale. We found this can be achieved with the following parameterization: $\mathbf{T} = \frac{\mathbf{Z}-\mu}{\sigma\sqrt{\frac{1}{\nu}\mathbf{W}\mathbf{U}^2}}$. In practice, we found that $\sigma = \sqrt{\nu}$ often works well, leading to the final simpler form:

$$\mathbf{T} = \frac{\mathbf{Z} - \mu}{\sqrt{\mathbf{W}\mathbf{U}^2}} \tag{6}$$

Given this construction, we observe that the TPoT generative model can instead be viewed as a latent variable model where all random variables are Gaussian and the construction of $\mathbf{T}$ in Equation 6 is the first layer of the generative 'decoder': $g_\theta(\mathbf{t}) = g_\theta(\mathbf{u}, \mathbf{z})$. In Section 5 we then leverage this interpretation to show how an approximate posterior for the latent variables $\mathbf{Z}$ and $\mathbf{U}$ can be trained through variational inference.

4.4 Capsules as Disjoint Topologies

One setting of neighborhood structure $\mathbf{W}$ which is of particular interest is when there exist multiple sets of disjoint neighborhoods. Statistically, the variables of two disjoint topologies are completely independent. An example of a capsule neighborhood structure is shown in Figure 2. The idea of independant subspaces has previously been shown to learn invariant feature subspaces in the linear setting and is present in early work on Independent Subspace Analysis [25] and Adaptive Subspace Self Organizing Maps (ASSOM) [33]. It is also very reminiscent of the transformed sets of features present in a group equivariant convolutional neural network. In the next section, we will show how temporal coherence can be leveraged to induce the encoding of observed transformations into the internal dimensions of such capsules thereby yielding unsupervised equivariant capsules.

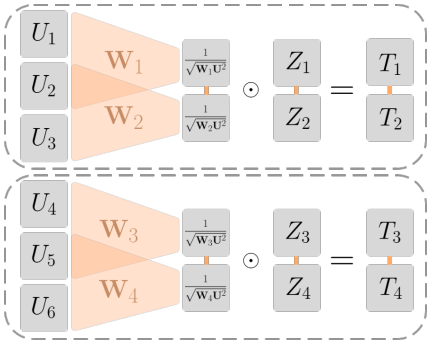

Figure 2: An example of a neighborhood structure which induces disjoint topologies (aka capsules). Lines between variables T_i indicate that sharing of U_i, and thus correlation.

4.5 Temporal Coherence and Equivariance

We now describe how the induced topographic organization can be leveraged to learn a basis of equivariant capsules for observed transformation sequences. The resulting representation is composed of a large set of 'capsules' where the dimensions inside the capsule are topographically structured, but between the capsules there is independence. To benefit from sequences of input, we encourage topographic structure over time between sequentially permuted activations within a capsule, a property we refer to as *shifting temporal coherence*. In following we describe how this is formally achieved.

4.5.1 Temporal Coherence

Temporal Coherence can be measured as the correlation of squared activation between time steps. One way we can achieve this in our model is by having T_j share U_i between time steps. Formally, the generative model is identical to Equation 1, factorizing over timesteps denoted by subscript l, i.e. $p_{\mathbf{X}_l,\mathbf{T}_l}(\mathbf{x}_l, \mathbf{t}_l) = p_{\mathbf{X}_l|\mathbf{T}_l}(\mathbf{x}_l|\mathbf{t}_l)p_{\mathbf{T}_l}(\mathbf{t}_l)$. However, $\mathbf{T}_l$ is now a function of a sequence $\{\mathbf{U}_{l+\delta}\}_{\delta=-L}^{L}$:

$$\mathbf{T}_l = \frac{\mathbf{Z}_l - \mu}{\sqrt{\mathbf{W}\left[\mathbf{U}_{l+L}^2; \cdots; \mathbf{U}_{l-L}^2\right]}} \tag{7}$$

Where $\left[\mathbf{U}_{l+L}^2; \cdots; \mathbf{U}_{l-L}^2\right]$ denotes vertical concatenation of the column vectors $\mathbf{U}_l$, and L can be seen as the window size. We see that the choice of $\mathbf{W}$ now defines correlation structure over time. In prior work on temporal coherence [24**?**], the grouping over time is such that a given variable $T_{l,i}$ has correlated energy with *the same spatial location* (i) at a previous time step $(l-1)$ $\big($i.e. $\mathrm{cov}(T_{l,i}^2, T_{l-1,i}^2) > 0\big)$. This can be implemented as:

$$\mathbf{W}\left[\mathbf{U}_{l+L}^2; \cdots; \mathbf{U}_{l-L}^2\right] = \sum_{\delta=-L}^{L} \mathbf{W}_\delta \mathbf{U}_{l+\delta}^2 \tag{8}$$

Where $\mathbf{W}_\delta$ defines the topography for a single timestep, and is typically the same for all timesteps.

4.5.2 Equivariance as Shifting Temporal Coherence

In our model, instead of requiring a single location to have correlated energies over a sequence (which has been shown to learn invariant features [24]), we would like variables at sequentially permuted locations *within a capsule* to have correlated energy between timesteps $\big(\mathrm{cov}(T_{l,i}^2, T_{l-1,i-1}^2) > 0\big)$. Similarly, this can be implemented as:

$$\mathbf{W}\left[\mathbf{U}_{l+L}^2; \cdots; \mathbf{U}_{l-L}^2\right] = \sum_{\delta=-L}^{L} \mathbf{W}_\delta \mathrm{Roll}_\delta(\mathbf{U}_{l+\delta}^2) \tag{9}$$

Where $\mathrm{Roll}_\delta(\mathbf{U}_{l+\delta}^2)$ denotes a cyclic permutation of δ steps along the capsule dimension. The exact implementation, continuous and non-cyclic extensions of Roll can be found in the appendix.

5 Topographic VAE

To train the parameters of the generative model θ, we use the above formulation to parameterize an approximate posterior for $\mathbf{t}$ in terms of a deterministic transformation of approximate posteriors over simpler Gaussian latent variables $\mathbf{u}$ and $\mathbf{z}$. Explicitly:

$$q_\phi(\mathbf{z}_l|\mathbf{x}_l) = \mathcal{N}\big(\mathbf{z}_l; \mu_\phi(\mathbf{x}_l), \sigma_\phi(\mathbf{x}_l)\mathbf{I}\big) \qquad p_\theta(\mathbf{x}|g_\theta(\mathbf{t})) = p_\theta(\mathbf{x}|g_\theta(\mathbf{z}, \mathbf{u})) \tag{10}$$

$$q_\gamma(\mathbf{u}_l|\mathbf{x}_l) = \mathcal{N}\big(\mathbf{u}_l; \mu_\gamma(\mathbf{x}_l), \sigma_\gamma(\mathbf{x}_l)\mathbf{I}\big) \qquad \mathbf{t}_l = \frac{\mathbf{z}_l - \mu}{\sqrt{\mathbf{W}\left[\mathbf{u}_{l+L}^2; \cdots; \mathbf{u}_{l-L}^2\right]}} \tag{11}$$

We denote this model the Topographic VAE (TVAE) and optimize the parameters θ, ϕ, γ (and μ) through the ELBO, summed over the sequence length S:

$$\sum_{l=1}^{S} \mathbb{E}_{Q_{\phi,\gamma}(\mathbf{z}_l,\mathbf{u}_l|\mathbf{x}_l)}\left([\log p_\theta(\mathbf{x}_l|g_\theta(\mathbf{t}_l))] - D_{KL}[q_\phi(\mathbf{z}_l|\mathbf{x}_l)||p_{\mathbf{Z}}(\mathbf{z}_l)] - D_{KL}[q_\gamma(\mathbf{u}_l|\mathbf{x}_l)||p_{\mathbf{U}}(\mathbf{u}_l)]\right) \tag{12}$$

where $Q_{\phi,\gamma}(\mathbf{z}_l, \mathbf{u}_l|\mathbf{x}_l) = q_\phi(\mathbf{z}_l|\mathbf{x}_l)\prod_{\delta=-L}^{L} q_\gamma(\mathbf{u}_{l+\delta}|\mathbf{x}_{l+\delta})$. Practically, we observe layer-wise *weight normalization* [47] is required to achieve topographic organization, this is detailed in the appendix.

6 Experiments

In the following experiments, we demonstrate the viability of the Topographic VAE as a novel method for training deep topographic generative models. Additionally, we quantitatively verify that shifting

temporal coherence yields approximately equivariant capsules by computing an 'equivariance loss' and a correlation metric inspired by the disentanglement literature. We show that equivariant capsule models yield higher likelihood than baselines on test sequences, and qualitatively support these results with visualizations of sequences reconstructed purely from rolled capsule activations.

6.1 Evaluation Methods

As depicted in Figure 1, we make use of *capsule traversals* to qualitatively visualize the transformations learned by our network. Simply, these are constructed by encoding a partial sequence into a $\mathbf{t}_0$ variable, and decoding sequentially Roll'd copies of this variable. Explicitly, in the top row we show the data sequence $\{\mathbf{x}_l\}_l$, and in the bottom row we show the decoded sequence: $\{g_\theta(\mathrm{Roll}_l(\mathbf{t_0}))\}_l$.

To measure equivariance quantitatively, we measure an *equivariance error* similar to [14]. The equivariance error can be seen as the difference between traversing the two distinct paths of the commutative diagram, and provides some measure of how precisely the function and the transform commute. Formally, for a sequence of length S, and $\hat{\mathbf{t}} = \mathbf{t}/||\mathbf{t}||_2$, the error is defined as:

$$\mathcal{E}_{eq}(\{\mathbf{t}_l\}_{l=1}^{S}) = \sum_{l=1}^{S-1}\sum_{\delta=1}^{S-l} \left|\left|\mathrm{Roll}_\delta(\hat{\mathbf{t}}_l) - \hat{\mathbf{t}}_{l+\delta}\right|\right|_1 \tag{13}$$

Additionally, inspired by existing disentanglement metrics, we measure the degree to which observed transformations in capsule space are correlated with input transformations by introducing a new metric we call $\mathrm{CapCorr}_y$. Simply, this metric computes correlation between the amount of observed roll of a capsule's activation at two timesteps l and $l+\delta$, and the shift of the ground truth generative factors y_l in that same time. Formally, for correlation coefficient Corr:

$$\mathrm{CapCorr}(\mathbf{t}_l, \mathbf{t}_{l+\delta}, y_l, y_{l+\delta}) = \mathrm{Corr}\left(\mathrm{argmax}\left[\mathbf{t}_l \star \mathbf{t}_{l+\delta}\right], |y_l - y_{l+\delta}|\right) \tag{14}$$

Where $\star$ is discrete cross-correlation across the capsule dimension, and the correlation coefficient is computed across the entire dataset. We see the argmax of the cross-correlation is an estimate of the degree to which a capsule activation has shifted from time l to $l+\delta$. To extend this to multiple capsules, we can replace the argmax function with the mode of the argmax computed for all capsules. We provide additional details and extensions of this metric in the appendix. For measuring capsule-metrics on baseline model which do not naturally have capsules, we simply divide the latent space into a fixed set of corresponding capsules and capsule dimensions.

6.2 Topographic VAE without Temporal Coherence

To validate the TVAE is capable of learning topographically organized representations with deep neural networks, we first perform experiments on a Topographic VAE without Temporal Coherence. The model is constructed as in Equations 10 and 11 with $L = 0$, and is trained to maximize Equation 12. We fix $\mathbf{W}$ such that globally the latent variables are arranged in a grid on a 2-dimensional torus (a single capsule), and locally $\mathbf{W}$ sums over 5x5 2D groups of variables. In this setting, $\mathbf{W}$ can be easily implemented as 2D convolution with a 5x5 kernel of 1's, stride 1, and cyclic padding. We see that training the model with 3-layer MLP's for the encoders and decoder indeed yields a 2D topographic organization of higher level features. In Figure 3, we show the maximum activating image for each final layer neuron of the capsule, plotted as a flattened torus. We see that the neurons become arranged according to class, orientation, width, and other learned features.

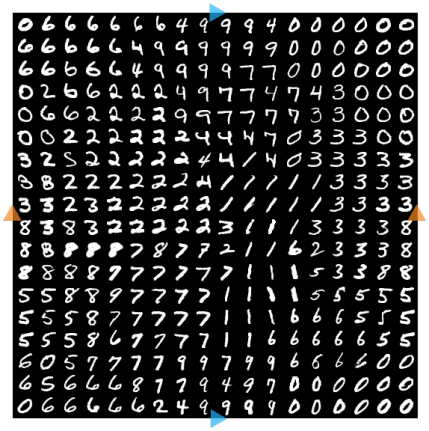

Figure 3: Maximum activating images for a Topographic VAE trained with a 2D torus topography on MNIST.

6.3 Learning Equivariant Capsules

In the remaining experiments, we provide evidence that the Topographic VAE can be leveraged to learn equivariant capsules by incorporating shifting temporal coherence into a 1D baseline topographic

model. We compare against two baselines: standard normal VAEs (without weight normalization), and models that have non-shifting 'stationary' temporal coherence (denoted 'BubbleVAE' [24]).

In all experiments we use a 3-layer MLP with ReLU activations for both encoders and the decoder. We arrange the latent space into 15 circular capsules each of 15-dimensions for dSprites [40], and 18 circular capsules each of 18-dimensions for MNIST [36]. Example sequences $\{\mathbf{x}_l\}_{l=1}^{S}$ are formed by taking a random initial example, and sequentially transforming it according to one of the available transformations: (X-Pos, Y-Pos, Orientation, Scale) for dSprites, and (Color, Scale, Orientation) for MNIST. All transformation sequences are cyclic such that when the maximum transformation parameter is reached, the subsequent value returns to the minimum. We denote the length of a full transformation sequence by S, and the time-extent of the induced temporal coherence (i.e. the length of the input sequence) by $2L$. For simplicity, both datasets are constructed such that the sequence length S equals the capsule dimension (for dSprites this involves taking a subset of the full dataset and looping the scale 3-times for a scale-sequence). Exact details and subsets are in the appendix.

In Figure 4, we show the capsule traversals for TVAE models with $L \approx \frac{1}{3}S$. We see that despite the $\mathbf{t}_0$ variable encoding only $\frac{2}{3}$ of the sequence, the remainder of the transformation sequence can be decoded nearly perfectly by permuting the activation through the full capsule – implying the model has learned to be equivariant to full sequences while only observing partial sequences per training point. Furthermore, we see that the model is able to successfully learn all transformations simultaneously for the respective datasets. Capsule traversals for the non-equivariant baselines, as well as TVAEs with smaller values of L (which learn only equivariance to partial sequences) are shown in the appendix. We note that the capsule traversal plotted in Figure 1 uses a separate dataset where color and rotation transformations are combined simultaneously. This is intended solely to demonstrate the ability of the TVAE to learn complex transformations, but such combined transformations are not present in the remainder of the experiments presented in this section. Additional results involving complex learned transformations (such as perspective transforms) are shown in the appendix.

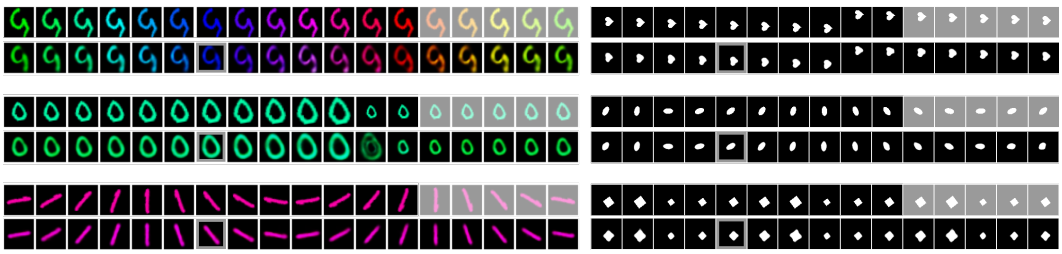

Figure 4: Capsule Traversals for TVAE models on dSprites and MNIST. The top rows show the encoded sequences (with greyed-out images held-out), and the bottom rows show the images generated from by decoding sequentially rolled copies of the initial activation $\mathbf{t}_0$ (indicated by a grey border).

In Table 1 we measure the equivariance error and log-likelihood (reported in nats) of the test data under our trained MNIST models as estimated by importance sampling with 10 samples. We observe that models which incorporate temporal coherence (BubbleVAE and TVAE with $L > 0$) achieve higher log-likelihood than the baseline VAE, while the TVAE models with shifting temporal coherence achieve the highest likelihood and the lowest equivariance error . Surprisingly, we also see that the TVAE with no temporal coherence ($L = 0$) still achieves significantly lower equivariance error than the VAE baseline. By examining the capsule traversals of this model (see appendix), we posit that this is due to the induced smoothness of the representation within capsules.

Table 1: Log Likelihood and Equivariance Error on MNIST for different settings of temporal coherence length L relative to sequence length S. Mean $\pm$ std. over 3 random initalizations.

Model (L, S)	TVAE $L = \frac{1}{2}S$	TVAE $L = \frac{5}{36}S$	TVAE $L = 0$	BubbleVAE $L = \frac{5}{36}S$	VAE $L = 0$
$\log p(\mathbf{x}) \uparrow$	-188.5 $\pm$ 1.0	$\mathbf{-184.7} \pm 0.7$	-217.3 $\pm$ 1.3	190.0 $\pm$ 0.3	-189.0 $\pm$ 0.8
$\mathcal{E}_{eq} \downarrow$	$\mathbf{386.7} \pm 1.0$	3149.6 $\pm$ 20.7	3047.8 $\pm$ 59.2	2522.5 $\pm$ 21.3	13273.9 $\pm$ 0.5

To further understand how capsules transform for observed input transformations, in Table 2 we measure $\mathcal{E}_{eq}$ and the CapCorr metric on the dSprites dataset for the four proposed transformations.

We see that the TVAE with $L \geq \frac{1}{3}S$ achieves perfect correlation – implying the learned representation indeed permutes cyclically within capsules for observed transformation sequences. Further, we see that this correlation gradually decreases as L decreases, eventually reaching the same level as the baselines. We see that contrary to the equivariance loss, the correlation of transformations for $L = 0$ is not higher than the standard VAE. We believe this to be due to the fundamental difference between the metrics. $\mathcal{E}_{eq}$ measures continuous L1 similarity which is still low when a representation is locally smooth (even if the change of the representation does not exactly follow the observed transformation), whereas $\mathrm{CapCorr}$ more strictly measures the correspondence between the transformation of the input and the transformation of the representation. Additionally, $\mathcal{E}_{eq}$ is misleadingly low for invariant representations, whereas $\mathrm{CapCorr}$ strictly measures equivariance. Interestingly, we also see the BubbleVAE achieves even lower correlation than the VAE baseline, agreeing with the assumption that such a model should learn invariant features.

Table 2: Equivariance error ($\mathcal{E}_{eq} \downarrow$) and correlation of observed capsule roll with ground truth factor shift ($\mathrm{CapCorr} \uparrow$) for the dSprites dataset. Mean $\pm$ standard deviation over 3 random initalizations.

Model (L, S)	TVAE $L = \frac{1}{2}S$	TVAE $L = \frac{1}{3}S$	TVAE $L = \frac{1}{6}S$	TVAE $L = 0$	BubbleVAE $L = \frac{1}{6}S$	VAE $L = 0$
$\mathrm{CapCorr}_X \uparrow$	$\mathbf{1.0} \pm 0$	$\mathbf{1.0} \pm 0$	0.68 ± 0.02	0.17 ± 0.03	0.07 ± 0.02	0.24 ± 0.01
$\mathrm{CapCorr}_Y \uparrow$	$\mathbf{1.0} \pm 0$	$\mathbf{1.0} \pm 0$	0.69 ± 0.02	0.17 ± 0.04	0.08 ± 0.01	0.25 ± 0.01
$\mathrm{CapCorr}_O \uparrow$	$\mathbf{1.0} \pm 0$	$\mathbf{1.0} \pm 0$	0.57 ± 0.01	0.11 ± 0.02	0.05 ± 0.01	0.09 ± 0.01
$\mathrm{CapCorr}_S \uparrow$	$\mathbf{1.0} \pm 0$	$\mathbf{1.0} \pm 0$	0.42 ± 0.01	0.52 ± 0.01	0.26 ± 0.00	0.47 ± 0.00
$\mathcal{E}_{eq} \downarrow$	$\mathbf{131} \pm 0$	964 ± 9	2313 ± 33	1075 ± 20	1397 ± 29	6929 ± 1

7 Future Work & Limitations

The model presented in this work has a number of limitations in its existing form which we believe to be interesting directions for future research. Foremost, the model is challenging to compare directly with existing disentanglement and equivariance literature since it requires an input sequence which determines the transformations reachable through the capsule roll. We believe this limitation could be overcome in multiple manners, either by increasing the capsule dimension beyond 1D circles, or by increasing the sparsity between capsules such that individual capsules begin to specialize on transformations. Related to this, we note the temporal coherence proposed in our model is not 'causal' ($\mathbf{t}_0$ depends on future $\mathbf{x}_l$). However this may be easily remedied when desired, and future work may only be a function of past inputs. We provide more details on these proposals in the supplementary material.

We additionally note that a priori definition of topographic structure may be a burden to model developers. While true, we know that the construction of appropriate priors is always a challenging task in latent variable models, and we observe that our proposed TVAE achieves strong performance even with improper specification. Furthermore, in future work, we believe adding learned flexibility to the parameters $\mathbf{W}$ may alleviate some of this burden.

Finally, we note that while this work does demonstrate improved log-likelihood and equivariance error, the study is inherently preliminary and does not examine all important benefits of equivariant representations. Specifically, further study of this model in terms of the sample complexity, semi-supervised classification accuracy, and invariance through capsule pooling would be enlightening.

8 Conclusion

In the above work we introduce the Topographic Variational Autoencoder as a method to train deep topographic generative models, and show how topography can be leveraged to learn equivariant sets of features, a.k.a. capsules, directly from sequences of data with no other supervision. Ultimately, we believe these results may shine some light on how biological brains could hard-wire themselves to more effectively learn representations with equivariant capsule structure. In terms of broader impact, the learned transformations of our model could be used to generate more realistic transformations of 'deepfakes', enhancing disinformation. Given that the model learns *approximate* equivariance, we caution against the over-reliance on equivariant properties as these have no formal guarantees yet.

References

[1] Horace B Barlow et al. Possible principles underlying the transformation of sensory messages. *Sensory communication*, 1(01), 1961.

[2] Anthony J. Bell and Terrence J. Sejnowski. An Information-Maximization Approach to Blind Separation and Blind Deconvolution. *Neural Computation*, 7(6):1129–1159, 11 1995.

[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] Gregory W. Benton, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. Learning invariances in neural networks. *CoRR*, abs/2010.11882, 2020.

[5] Christopher Bishop, Markus Svensen, and Christopher Williams. Gtm: The generative topographic mapping. *Neural Computation*, 10:215–234, 05 1997.

[6] Benedikt Boenninghoff, Steffen Zeiler, Robert M. Nickel, and Dorothea Kolossa. Variational autoencoder with embedded student-t mixture model for authorship attribution. *ArXiv*, abs/2005.13930, 2020.

[7] Diane Bouchacourt, Mark Ibrahim, and Stéphane Deny. Addressing the topological defects of disentanglement via distributed operators. *ArXiv*, abs/2102.05623, 2021.

[8] Taco Cohen and M. Welling. Transformation properties of learned visual representations. *CoRR*, abs/1412.7659, 2015.

[9] Taco Cohen and M. Welling. Steerable cnns. *ArXiv*, abs/1612.08498, 2017.

[10] Taco Cohen and Max Welling. Learning the irreducible representations of commutative lie groups. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1755–1763, Bejing, China, 22–24 Jun 2014. PMLR.

[11] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999, 2016.

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

[13] Marissa Connor, Gregory Canal, and Christopher Rozell. Variational autoencoder with learned latent structure. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 2359–2367. PMLR, 13–15 Apr 2021.

[14] Nichita Diaconu and Daniel E. Worrall. Learning to convolve: A generalized weight-tying approach. *CoRR*, abs/1905.04663, 2019.

[15] Marc Finzi, Samuel Stanton, Pavel Izmailov, and Andrew Gordon Wilson. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3165–3176. PMLR, 13–18 Jul 2020.

[16] Marc Finzi, M. Welling, and Andrew Gordon Wilson. A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups. *ArXiv*, abs/2104.09459, 2021.

[17] Peter Földiák. Learning invariance from transformation sequences. *Neural Computation*, 3:194–200, 06 1991.

[18] Will Grathwohl and Aaron Wilson. Disentangling space and time in video with hierarchical variational auto-encoders. *CoRR*, abs/1612.04440, 2016.

[19] Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a definition of disentangled representations. *ArXiv*, abs/1812.02230, 2018.

[20] Geoffrey E. Hinton, Alex Krizhevsky, and Sida D. Wang. Transforming auto-encoders. In Timo Honkela, Włodzisław Duch, Mark Girolami, and Samuel Kaski, editors, *Artificial Neural Networks and Machine Learning – ICANN 2011*, pages 44–51, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[21] Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with EM routing. In *International Conference on Learning Representations*, 2018.

[22] Geoffrey E. Hinton and Yee-Whye Teh. Discovering multiple constraints that are frequently approximately satisfied. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, UAI'01, page 227–234, 2001.

[23] Jarmo Hurri and Aapo Hyvärinen. Simple-Cell-Like Receptive Fields Maximize Temporal Coherence in Natural Video. *Neural Computation*, 15(3):663–691, 03 2003.

[24] A. Hyvärinen, J. Hurri, and Jaakko J. Väyrynen. A unifying framework for natural image statistics: spatiotemporal activity bubbles. *Neurocomputing*, 58-60:801–806, 2004.

[25] Aapo Hyvärinen and Patrik Hoyer. Emergence of phase-and shift-invariant features by decomposition of natural images into independent feature subspaces. *Neural computation*, 12(7):1705–1720, 2000.

[26] Aapo Hyvärinen, Patrik O Hoyer, and Mika Inki. Topographic independent component analysis. *Neural computation*, 13(7):1527–1558, 2001.

[27] Aapo Hyvärinen, Jarmo Hurri, and Patrick O Hoyer. *Natural image statistics: A probabilistic approach to early computational vision.*, volume 39. Springer Science & Business Media, 2009.

[28] Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000.

[29] Aapo Hyvärinen and Patrik O. Hoyer. A two-layer sparse coding model learns simple and complex cell receptive fields and topography from natural images. *Vision Research*, 41(18):2413–2423, 2001.

[30] Koray Kavukcuoglu, Marc'Aurelio Ranzato, Rob Fergus, and Yann LeCun. Learning invariant features through topographic filter maps. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1605–1612. IEEE, 2009.

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

[32] David Klindt, Lukas Schott, Yash Sharma, Ivan Ustyuzhaninov, Wieland Brendel, Matthias Bethge, and Dylan Paiton. Towards nonlinear disentanglement in natural data with temporal sparse coding. *ArXiv*, abs/2007.10930, 2021.

[33] Teuvo Kohonen. Emergence of invariant-feature detectors in the adaptive-subspace self-organizing map. *Biological cybernetics*, 75(4):281–291, 1996.

[34] Alexei A Koulakov and Dmitri B Chklovskii. Orientation preference patterns in mammalian visual cortex: a wire length minimization approach. *Neuron*, 29(2):519–527, 2001.

[35] Quoc Le, Alexandre Karpenko, Jiquan Ngiam, and Andrew Ng. Ica with reconstruction cost for efficient overcomplete feature learning. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

[36] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[37] Hyodong Lee, Eshed Margalit, Kamila M. Jozwik, Michael A. Cohen, Nancy Kanwisher, Daniel L. K. Yamins, and James J. DiCarlo. Topographic deep artificial neural networks reproduce the hallmarks of the primate inferior temporal cortex face processing network. *bioRxiv*, 07/2020 2020.

[38] Jan Eric Lenssen, Matthias Fey, and Pascal Libuschewski. Group equivariant capsule networks. *arXiv preprint arXiv:1806.05086*, 2018.

[39] Libo Ma and Liqing Zhang. Overcomplete topographic independent component analysis. *Neurocomputing*, 71(10-12):2217–2223, 2008.

[40] Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

[41] Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision research*, 37(23):3311–3325, 1997.

[42] Simon Osindero, Max Welling, and Geoffrey E. Hinton. Topographic Product Models Applied to Natural Scene Statistics. *Neural Computation*, 18(2):381–414, 02 2006.

[43] Simon Kayode Osindero. *Contrastive Topographic Models*. PhD thesis, University of London, 2004.

[44] Siamak Ravanbakhsh, Jeff Schneider, and Barnabas Poczos. Equivariance through parameter-sharing. *ArXiv*, abs/1702.08389, 2017.

[45] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *ArXiv*, abs/1401.4082, 2014.

[46] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. *arXiv preprint arXiv:1710.09829*, 2017.

[47] Tim Salimans and Diederik P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *ArXiv*, abs/1602.07868, 2016.

[48] James V. Stone. Learning Perceptually Salient Visual Parameters Using Spatiotemporal Smoothness Constraints. *Neural Computation*, 8(7):1463–1492, 10 1996.

[49] Richard Turner and Maneesh Sahani. A maximum-likelihood interpretation for slow feature analysis. *Neural computation*, 19:1022–38, 05 2007.

[50] Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco Cohen. 3d steerable cnns: Learning rotationally equivariant features in volumetric data. *ArXiv*, abs/1807.02547, 2018.

[51] Max Welling, Simon Osindero, and Geoffrey E Hinton. Learning sparse topographic representations with products of student-t distributions. In *Advances in neural information processing systems*, pages 1383–1390, 2003.

[52] Laurenz Wiskott and Terrence J Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural computation*, 14(4):715–770, 2002.

[53] Daniel Worrall and Max Welling. Deep scale-spaces: Equivariance over scale. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[54] Daniel E. Worrall, Stephan J. Garbin, Daniyar Turmukhambetov, and Gabriel J. Brostow. Harmonic networks: Deep translation and rotation equivariance. *ArXiv*, abs/1612.04642, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? Yes, see Figure 1 for an example of a learned highly complex transformation, and Sections 5 and 6 for the proposed model and empirical demonstration of claimed contributions.
   (b) Did you describe the limitations of your work? Yes, see Section 7
   (c) Did you discuss any potential negative societal impacts of your work? Yes, see Section **??**
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? Yes.
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? N/A
   (b) Did you include complete proofs of all theoretical results? N/A
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? Yes, see supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? Yes, see supplementary material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? Yes, see Tables 1 and 2.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? Yes, see supplementary material.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? Yes, see Section 6
   (b) Did you mention the license of the assets? Yes, see supplementary meterial.
   (c) Did you include any new assets either in the supplemental material or as a URL? Yes, see supplementary material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? N/A
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? N/A
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? N/A
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? N/A
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? N/A

A Experiment Details

The code for reproducing all experiments in this paper can be found in the following GitHub repository: `https://github.com/ReallyAnonNeurips2021/TopographicVAE`

A.1 Optimizer Parameters

Given the differences between the training procedures of the model presented in Section 6.2, and those in Section 6.3, the optimizer parameters for the two settings differed slightly. The 2D Topographic VAE without Temporal Coherence presented in Figure 3 was trained with stochastic gradient descent on batches of size 128, using a learning rate of 1×10^{-3}, and standard momentum of 0.9. All models in Section 6.3 were trained with stochastic gradient descent on batches of size 8 (due to each batch-example being a length 15 or 18 sequence), using a learning rate of 1×10^{-4}, and standard momentum of 0.9. All models were trained for 100 epochs. As mentioned in Section 5, we applied layer-wise weight normalization [47] to all topographic models (TVAE & BubbleVAE), but not to the baseline VAE as we found its likelihood was significantly higher without weight normalization. The details of the weight normalization method are in Section A.8 below.

A.2 Initalization

All weights of the models were initialized with uniformly random samples from $U(-\frac{1}{\sqrt{m}}, \frac{1}{\sqrt{m}})$, where m is the number of input units. For all topographic models including BubbleVAE, μ was initialized to a large value (30.0) as this was observed to increase the speed of convergence and was sometimes necessary for observed topographic organization in deeper models. For the 2D topographic model in Figure 2, μ was initialized to 5.

A.3 Model Architectures

All models presented in this paper make use of the same 3-Layer MLP for parameterizing the encoders and decoders. Specifically, the model is constructed as 3 fully connected layers with ReLU activations in-between the layers. For MNIST, the layers of both the $\mathbf{u}$ and $\mathbf{z}$ encoders have (972, 648, 648) output units each for the first, second, and third layers respectively. The 648 units in the third layer are divided into two sets to compute the mean and log standard deviation of the respective u's and z's, yielding 324 t variables. This is then divided into 18 capsules, each of 18 dimensions. The layers of the decoder have (648, 972, 2352) output units respectively. For dSprites, both encoder layers have output sizes (674, 450, 450), where the resulting 225 t variables are divided into 15 capsules, each of 15 dimensions. The decoder layers then have output sizes (450, 675, 4096). We note the non-topographic VAE baselines make use of only a single encoder for the Gaussian variable $\mathbf{z}$ (as $\mathbf{u}$ is not needed), and do not incorporate a μ parameter.

A.4 Choice of $\mathbf{W}$

For all topographic models (TVAE and BubbleVAE) in Section 6.3, the global topographic organization afforded by $\mathbf{W}$ was fixed to a set of 1-D tori ('circular capsules') as depicted in Figure 1. The model presented in Section 6.2 organizes its variables as a single 2-D torus. Practically, multiplication by $\mathbf{W}$ was performed by convolution over the appropriate dimensions (time & capsule dimension) with a kernel of all 1's, taking advantage of circular padding to achieve toroidal structure.

A.5 Choice of $\mathbf{W}_\delta$

The choice of $\mathbf{W}_\delta$ determines the local topographic structure within a single timestep. For all TVAE models with $L > 0$ we experimented with local neighborhood sizes (denoted K) of 3 units (effective kernel size 3 in the capsule dimension), and 1 unit (no neighborhood). For MNIST it was observed that $K = 3$ performed best, while $K = 1$ worked best for dSprites. This is likely due to the slower, smoother, and more overlapping transformations constructed on MNIST, whereas our subset of dSprites contained non-smooth transformations where the overlap between successive images was smaller (e.g. due to sub-setting, see Section A.10), which made larger neighborhood sizes $K > 1$ less fitting. For TVAE models with $L = 0$, $\mathbf{W}_\delta = \mathbf{W}$ was fixed to sum over neighborhoods of size $K = 9$ for MNIST and $K = 10$ for dSprites. These values were chosen arbitrarily to be close to half

of the capsule size. For BubbleVAE models, the extent of topographic organization in the capsule dimension was set to be equal to the organization in time dimension $K = 2L$. We additionally experimented with $K = 3$ to match the TVAE, but observed this performed worse in on all metrics.

A.6 Choice of L

The choice of L determines the extent of temporal coherence where $2L$ equals the input sequence length. For Table 1, we experimented with values of L in the set $\{0, \frac{5}{36}S, \frac{1}{4}S, \frac{1}{2}S\}$ for both the TVAE and BubbleVAE. Both the BubbleVAE and TVAE achieved highest likelihoods at $L = \frac{5}{36}S$, and TVAE achieved lowest equivariance error at $L = \frac{1}{2}S$. We additionally included TVAE experiments with $L = \frac{13}{36}S$ for purposes of visualization in Figures 1 and 4. For Table 2, we experimented with values of L in the set $\{0, \frac{1}{6}S, \frac{4}{15}S, \frac{1}{3}S, \frac{2}{5}S, \frac{1}{2}S\}$ for both TVAE and BubbleVAE, and presented a broad selection in the table. The results of all models are shown in Section B below.

A.7 Hyperparameter Selection

Hyperparameters such as learning rate, batch size, number of capsules, capsule size, and ultimately model architecture were chosen to allow for quick training on limited resources and were not tuned significantly. Since it was conceptually simpler to have an equal number of capsule dimensions and sequence elements, this limited the number of capsules we could then train efficiently. In Section C.1 we explain how a model with fewer capsule dimensions than sequence elements could be constructed.

A.8 Weight Normalization

As noted in Section 5, it was found that some form of weight normalization at each layer was necessary in order to achieve topographic organization. This is consistent with prior work on overcomplete Topographic Product of Student's-t models [51] and other overcomplete ICA methods [28] where the same technique has been used to prevent degenerate solutions. In the ICA literature, a *Projected Gradient Descent* method was used to enforce this constraint [35]. In our work, we observe *Weight Normalization* [47], achieved by reparameterization, works equally well while frequently obtaining higher log-likelihoods. Simply, *Weight Normalization* is achieved by reparamterizing the weight vectors (rows $\mathbf{w}_i$ of $\mathbf{W}$) as the product of a normalized vector $\frac{\mathbf{v}_i}{||\mathbf{v}_i||_2}$ and a 'rescaling' scalar g_i, i.e:

$$\mathbf{w}_i = g_i \cdot \frac{\mathbf{v}_i}{||\mathbf{v}_i||_2} \tag{15}$$

In the table below we show a comparison of Projected Gradient Descent (PGD) and weight normalization (WN) for the TVAE ($L = \frac{5}{36}S$) and VAE ($L = 0$) on the MNIST Dataset.

Table 3: Comparison of weight normalization methods for TVAE ($L = \frac{5}{36}S$) and baseline VAE. PGD stands for Projected Gradient Descent [35], while WN stands for Weight Normalization [47]. Mean $\pm$ std. over 3 random initalizations.

Model Norm. Method	TVAE PGD	TVAE WN	VAE PGD	VAE WN	VAE None
$\log p(\mathbf{x}) \uparrow$	-189.0 $\pm$ 0.1	-184.9 $\pm$ 0.8	-209.1 $\pm$ 1.1	-192.5 $\pm$ 0.5	-189.0 $\pm$ 0.8
$\mathcal{E}_{eq} \downarrow$	3196.7 $\pm$ 5.0	3137.9 $\pm$ 6.4	13262.0 $\pm$ 0.5	13273.2 $\pm$ 1.4	13273.9 $\pm$ 0.5

A.9 MNIST Transformations

The first set of experiments presented in this paper are based on the MNIST dataset [36] (MIT Licence). For Section 6.2 (Figure 3) an MNIST training set of 48,000 images was used, while the standard test set of 10,000 images was used to compute the maximum activating image. For Section 6.3 (Figure 4 and Table 1), sequences of MNIST images were created by picking a random training image (with a random transformation 'pose') and successively transforming it according to one of the 3 available transformations (e.g. only one attribute is changed per sequence). The available transformations consisted of rotation, color (hue rotation), and scale with increments of 20-degrees for

rotation and color, and 3.66% increments for scale. Since scale is inherently non-cyclic, the bounds of the transformation were set at 60% and 126%, and the transformations were constructed to be periodic such then once scale reached 126%, the next element was at 60% scale. The final sequences were thus constructed to be 18 images long, where each element in the batch had an independently randomly chosen transformation. Again, the likelihood $\log p(\mathbf{x})$ and equivariance error $\mathcal{E}_{eq}$ were computed on the held-out 10,000 example test set, where the same random transformation sequences were applied.

A.10 dSprites Transformations

The second set of experiments presented in this paper are based on the dSprites dataset [40] (Apache-2.0 License). To reduce computational complexity of this dataset, we took a subset of the dataset which consisted of all 3 shapes, the largest 5 scales, and every other example from the first 30 orientations, x-positions, and y-positions. The resulting dataset thus had 50,625 total images (3 shapes, 5 scales, 15 orientations, 15 x-positions, 15 y-positions), compared to the original 737,280 images. To construct sequences, we followed the same procedure as for MNIST, whereby first a random example and transformation were chosen, and a sequence of 15 images was constructed where only the chosen transformation was applied successively. We define the transformations available for sequences as scale, orientation, x-position, and y-position, omitting shape since smooth shape transforms are not present in the dSprites dataset. Again, we define all transformations to be cyclic such that once the 15th element is reached, the 1st element follows. For scale transformations, we simply loop over all 5 scales 3 times per sequence. We observe that although these sequences do not match the latent priors exactly, the models still train relatively well, implying some degree of robustness.

A.11 Capsule Correlation Metric ($\mathrm{CapCorr}$)

Here we define $\mathrm{CapCorr}$ more precisely as it is implemented in our work. First, we denote the ground truth transformation parameter of the sequence at timestep l as y_l (e.g. the rotation angle at timestep l for a rotation sequence), and the corresponding activation at time l as $\mathbf{t}_l$. Next, to get an arbitrary starting point, we let $l = \Omega$ denote the timestep when y_l is at its canonical position (e.g. rotation angle 0, x-position 0, or scale 1). We see Ω is not necessarily 0 since the first timestep of each sequence ($l = 0$) is a randomly transformed example. Then, we observe that we can measure the approximate observed roll in the capsule dimension between time 0 and Ω as a 'phase shift' by computing the index of the maximum value of a discrete (periodic) cross-correlation of $\mathbf{t}_\Omega$ and $\mathbf{t}_0$:

$$\mathrm{ObservedRoll}(\mathbf{t}_\Omega, \mathbf{t}_0) = \mathrm{argmax}\left[\mathbf{t}_\Omega \star \mathbf{t}_0\right] \tag{16}$$

Where $\star$ is discrete (periodic) cross-correlation across the (cyclic) capsule dimension and argmax is also subsequently performed over the capsule dimension. Then, the $\mathrm{CapCorr}$ metric for a single capsule is given as:

$$\mathrm{CapCorr}(\mathbf{t}_\Omega, \mathbf{t}_0, y_\Omega, y_0) = \mathrm{Corr}\left(\mathrm{ObservedRoll}(\mathbf{t}_\Omega, \mathbf{t}_0), |y_\Omega - y_0|\right) \tag{17}$$

Where the correlation coefficient Corr is then computed across all examples for the entire dataset. In our experiments we use the Pearson correlation coefficient for Corr. We thus see this metric is the correlation of the estimated observed capsule roll with the shift in ground truth generative factors, which is equal to 1 when the model is perfectly equivariant. To extend this definition to multiple capsules, we estimate $\mathrm{ObservedRoll}$ for each capsule separately, and then correlate the mode of all $\mathrm{ObservedRoll}$ values with the true shift in ground truth generative factors. We see empirically that the $\mathrm{ObservedRolls}$ for all capsules are almost always identical (i.e. all capsules roll simultaneously for each transformation), therefore computing the mode does not destroy significant information. Finally, for transformation sequences which have multiple timesteps where y_l is at the canonical position (e.g. scale transformations on dSprites where scale is looped 3 times), we select $l = \Omega$ to be the one from this possible set which yields the minimal absolute distance between $|y_\Omega - y_0|$ and $\mathrm{ObservedRoll}(\mathbf{t}_\Omega, \mathbf{t}_0)$.

A.12 Definition of Roll for Capsules

As stated in Section 4.5.2, $\mathrm{Roll}_\delta(\mathbf{u})$, is defined as a cyclic permutation of δ steps along the capsule dimension of $\mathbf{u}$. Explicitly, if $\mathbf{u}$ is divided into C capsules each with D dimensions, the Roll_δ operation can be written as:

$$\begin{aligned}\mathrm{Roll}_\delta(\mathbf{u}) &= \mathrm{Roll}_\delta\left([u_1, u_2, \ldots, u_{C\cdot D}]\right) \\ &= [u_D, u_1, \ldots, u_{D-1}, u_{2\cdot D}, u_{D+1}, \ldots, u_{2\cdot D-1}, u_{3\cdot D}, \ \ldots \ , \ \ldots \ u_{C\cdot D-1}]\end{aligned} \tag{18}$$

B Extended Results

In this section we include an extended version of Tables 1 & 2, showing all tested settings of the TVAE & BubbleVAE. Additionally, in Figure 5 below, we plot the average (over all transformations) $\mathrm{CapCorr}$ metric for all settings of L as shown in Table 5. We observe the TVAE achieves perfect correlation ($\mathrm{CapCorr} = 1$) for $L \geq \frac{1}{3}$, and steadily decreasing correlation for lower values of L.

Table 4: Log Likelihood and Equivariance Error on MNIST for all models tested. Mean $\pm$ std. over 3 random initalizations.

Model	TVAE	TVAE	TVAE	TVAE	TVAE
(L, S)	$L = \frac{1}{2}S$	$L = \frac{13}{36}S$	$L = \frac{1}{4}S$	$L = \frac{5}{36}S$	$L = 0$
K	$K = 3$	$K = 3$	$K = 3$	$K = 3$	$K = 9$
$\log p(\mathbf{x}) \uparrow$	-188.5 $\pm$ 1.1	-187.1 $\pm$ 1.2	-186.9 $\pm$ 0.6	$\mathbf{-184.7} \pm 0.7$	-217.3 $\pm$ 1.3
$\mathcal{E}_{eq} \downarrow$	$\mathbf{386.7} \pm 1.1$	990.6 $\pm$ 10.7	2045.7 $\pm$ 3.3	3149.6 $\pm$ 20.8	3047.8 $\pm$ 59.2

Model	BubbleVAE	BubbleVAE	BubbleVAE	BubbleVAE	VAE
(L, S)	$L = \frac{1}{2}S$	$L = \frac{1}{4}S$	$L = \frac{5}{36}S$	$L = \frac{5}{36}S$	$L = 0$
K	$K = 2L$	$K = 2L$	$K = 2L$	$K = 3$	$K = 1$
$\log p(\mathbf{x}) \uparrow$	-196.3 $\pm$ 1.1	-203.2 $\pm$ 2.1	-190.0 $\pm$ 0.3	-190.4 $\pm$ 0.9	-189.0 $\pm$ 0.8
$\mathcal{E}_{eq} \downarrow$	2066.5 $\pm$ 596.7	1005.3 $\pm$ 12.7	2522.5 $\pm$ 21.3	3277.0 $\pm$ 15.6	13273.9 $\pm$ 0.5

Table 5: Equivariance error and $\mathrm{CapCorr}$ for all models tested on the dSprites dataset. Mean $\pm$ standard deviation over 3 random initalizations.

Model	TVAE	TVAE	TVAE	TVAE	TVAE	TVAE
(L, S)	$L = \frac{1}{2}S$	$L = \frac{2}{5}S$	$L = \frac{1}{3}S$	$L = \frac{4}{15}S$	$L = \frac{1}{6}S$	$L = 0$
$\mathrm{CapCorr}_X \uparrow$	$\mathbf{1.0} \pm 0$	$\mathbf{1.0} \pm 0$	$\mathbf{1.0} \pm 0$	0.96 $\pm$ 0.01	0.68 $\pm$ 0.02	0.17 $\pm$ 0.03
$\mathrm{CapCorr}_Y \uparrow$	$\mathbf{1.0} \pm 0$	$\mathbf{1.0} \pm 0$	$\mathbf{1.0} \pm 0$	0.97 $\pm$ 0.01	0.69 $\pm$ 0.02	0.17 $\pm$ 0.04
$\mathrm{CapCorr}_O \uparrow$	$\mathbf{1.0} \pm 0$	$\mathbf{1.0} \pm 0$	$\mathbf{1.0} \pm 0$	0.90 $\pm$ 0.01	0.57 $\pm$ 0.01	0.11 $\pm$ 0.02
$\mathrm{CapCorr}_S \uparrow$	$\mathbf{1.0} \pm 0$	$\mathbf{1.0} \pm 0$	$\mathbf{1.0} \pm 0$	0.97 $\pm$ 0.00	0.42 $\pm$ 0.01	0.52 $\pm$ 0.01
$\mathcal{E}_{eq} \downarrow$	$\mathbf{131} \pm 0$	679 $\pm$ 4	964 $\pm$ 9	1298 $\pm$ 5	2313 $\pm$ 33	1075 $\pm$ 20

Model	BubbleVAE	BubbleVAE	BubbleVAE	BubbleVAE	BubbleVAE	VAE
(L, S)	$L = \frac{1}{2}S$	$L = \frac{2}{5}S$	$L = \frac{1}{3}S$	$L = \frac{4}{15}S$	$L = \frac{1}{6}S$	$L = 0$
$\mathrm{CapCorr}_X \uparrow$	0.17 $\pm$ 0.01	0.14 $\pm$ 0.01	0.13 $\pm$ 0.02	0.12 $\pm$ 0.03	0.07 $\pm$ 0.02	0.24 $\pm$ 0.01
$\mathrm{CapCorr}_Y \uparrow$	0.16 $\pm$ 0.01	0.14 $\pm$ 0.00	0.12 $\pm$ 0.01	0.12 $\pm$ 0.02	0.08 $\pm$ 0.01	0.25 $\pm$ 0.00
$\mathrm{CapCorr}_O \uparrow$	0.12 $\pm$ 0.03	0.13 $\pm$ 0.01	0.10 $\pm$ 0.01	0.09 $\pm$ 0.01	0.05 $\pm$ 0.01	0.09 $\pm$ 0.01
$\mathrm{CapCorr}_S \uparrow$	0.52 $\pm$ 0.00	0.53 $\pm$ 0.00	0.52 $\pm$ 0.02	0.49 $\pm$ 0.03	0.26 $\pm$ 0.00	0.47 $\pm$ 0.00
$\mathcal{E}_{eq} \downarrow$	3543 $\pm$ 441	6922 $\pm$ 6	2037 $\pm$ 924	2420 $\pm$ 1192	1397 $\pm$ 29	6929 $\pm$ 1

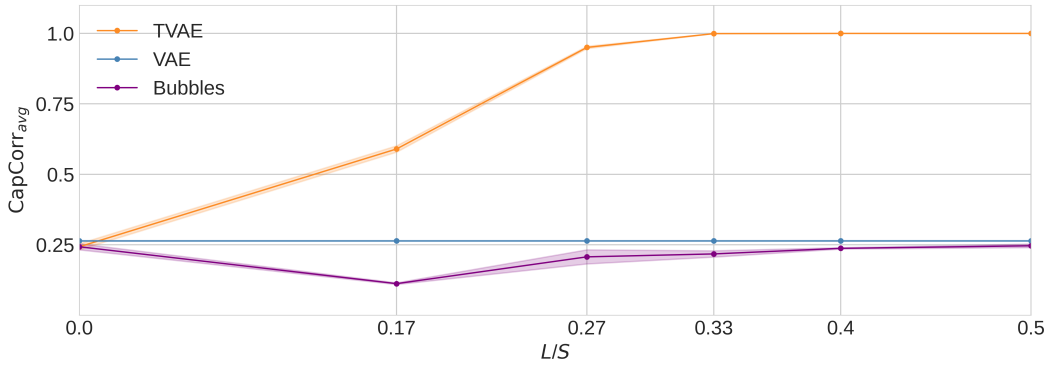

Figure 5: Average $\mathrm{CapCorr}$ over all transformations vs. Temporal Coherence Length L.

C Proposed Model Extensions

C.1 Extensions to Roll & $\mathrm{CapCorr}$

The Roll operation can be seen as defining the speed at which $\mathbf{t}$ transforms corresponding to an observed transformation. For example, with Roll defined as in Section A.12 above, we implicitly assume that for each observed timestep, we would like the representation $\mathbf{t}$ to cyclically permute 1-unit within the capsule. For this to match the observed data, it requires the model to have an equal number of capsule dimensions and sequence elements. If we wish to reduce the size of our representation, we need a 'partial permutation' for each observed transformation. For a single capsule with D elements, an example of a simple linear version of such a partial permutation (for $0 < \alpha \leq 1$) can be implemented as:

$$\mathrm{Roll}_\alpha(\mathbf{u}) = \left[\alpha u_D + (1-\alpha)u_1,\ \ \alpha u_1 + (1-\alpha)u_2,\ \ \ldots,\ \ \alpha u_{D-1} + (1-\alpha)u_D\right] \tag{19}$$

A slightly more principled partial roll for periodic signals could also be achieved by performing a phase shift of the signal in Fourier space, and performing the inverse Fourier transform to obtain the resulting rolled signal. To extend the $\mathrm{CapCorr}$ metric to similarly allow for partial Rolls, we see that we can simply redefine the $\mathrm{ObservedRoll}$ (originally given by discrete cross-correlation) to be given by the argmax of the inner product of a sequentially partially rolled activation with the initial activation $\mathbf{t}_\Omega$. Formally:

$$\mathrm{ObservedRoll}(\mathbf{t}_\Omega, \mathbf{t}_0) = \mathrm{argmax}\left[\mathbf{t}_\Omega \cdot \mathrm{Roll}_0(\mathbf{t}_0),\ \mathbf{t}_\Omega \cdot \mathrm{Roll}_\alpha(\mathbf{t}_0),\ \ldots\ , \mathbf{t}_\Omega \cdot \mathrm{Roll}_{D-\alpha}(\mathbf{t}_0)\right] \tag{20}$$

C.2 Non-Cyclic Capsules

We can also see that there is nothing beyond convenience which inherently requires the capsules to be circular (i.e. have periodic boundary conditions). To implement linear capsules, we propose one solution is to add L additional U_i variables to both the left and right boundaries of each capsule. In this way, the vector $\mathbf{U}$ is larger than the vector $\mathbf{Z}$ and can be seen as a 'padded' version, where the padding is composed of independant random variables. Additionally, the transformation sequences can then be padded on both sides by replicating the first and final elements L times. The construction of $\mathbf{T}$ variables is then performed identically as in Equations 7 and 9. The choice of a cyclic or non-cyclic Roll operation which can be defined as filling the boundaries with 0 since these values will not be used as part of the computation.

C.3 Multi-dimensional Temporally Coherent Capsules

In consideration of transformations which may naturally live in multiple dimensions, we wish to extend the original model to support multi-dimensional capsules. Such multi-dimensional capsules could additionally support more well-defined 'disentanglement' of transformations by encouraging each transformation to be axis-aligned with one dimension of each capsule. We see that in the non-temporally coherent case ($L = 0$), the model can easily be extended to capsules of multiple dimensions through multi-dimensional neighborhoods. An example of a model with 2-dimensional neighborhoods is presented in Figure 3. However, when considering shifting temporal coherence as we defined in Section 6.3, it is not clear how the shift operator or the neighborhoods should be defined for higher dimensional capsules. In this section we propose to modify the definitions of $\mathbf{T}$ in Equations 7 and 9 with an extension resembling 'group sparsity' in the denominator.

First, we again assume that each input sequence is an observation of a single transformation at a time. Formally, the multi-dimensional capsules are then constructed by arranging $\mathbf{U}$ into a D dimensional lattice. In such a model, we desire to roll and sum only along a single axis of the lattice for a given sequence. Incorporating this into the construction of $\mathbf{T}$ yields the following:

$$\mathbf{T}_l = \frac{\mathbf{Z}_l - \mu}{\sum_{d=1}^{D} \sqrt{\mathbf{W}^d \left[\mathbf{U}^2_{l+L}; \cdots; \mathbf{U}^2_{l-L}\right]}} = \frac{\mathbf{Z}_l - \mu}{\sum_{d=1}^{D} \sqrt{\sum_{\delta=-L}^{L} \mathbf{W}^d_\delta \mathrm{Roll}^d_\delta(\mathbf{U}^2_{l+\delta})}} \tag{21}$$

Where $\mathbf{W}^d_\delta$ refers to a matrix which sums locally along the d^{th} dimension of each capsule, and not at all along the others, and similarly Roll^d_δ rolls only along the d^{th} dimension. In practice we observe such models can indeed disentangle up to 2 distinct transformations, but become more challenging to optimize for higher dimensions. We believe this is potentially due to the exponential growth in capsule size with increasing dimension, but leave further exploration to future work.

C.4 Causal Temporal Coherence

Finally, as noted in the Conclusion, the sequence model in this paper is not 'causal', meaning that each variable $\mathbf{T}_l$ requires variables from future timesteps in the sequence ($\mathbf{U}_{l+\delta}$ for $\delta > 0$). Although for the purpose of learning equivariance in practice this may not be an issue, it may be relevant for some online learning applications. We can modify Equations 7 and 9 by changing the matrix $\mathbf{W}$ (implemented as convolution) to a causal convolution (i.e. masking out $\mathbf{W}_\delta$ for $\delta > 0$). Formally:

$$\mathbf{T}_l = \frac{\mathbf{Z}_l - \mu}{\sqrt{\mathbf{W}\left[\mathbf{U}_l^2; \cdots; \mathbf{U}_{l-L}^2\right]}} = \frac{\mathbf{Z}_l - \mu}{\sqrt{\sum_{\delta=-L}^{0} \mathbf{W}_\delta \mathrm{Roll}_\delta(\mathbf{U}_{l+\delta}^2)}} \tag{22}$$

In a causal setting, it is also likely the transformations are no longer assumed to be circular. We thus refer the reader to Section C.2 above on non-circular capsules, which can be combined with Equation 22, to achieve such a model.

D Capsule Traversals

In this section we provide a set of 12 capsule traversals for each of the models presented in main text. The traversals are randomly selected such that all transformations (and dSprites shapes) are shown evenly. Unlike the main section, we additionally include a middle row which shows the direct reconstruction of the input without any rolling (i.e. $\{g_\theta(\mathbf{t}_l)\}_l$). We find the direct reconstructions valuable to determine if poor traversals are due to bad reconstructions (low $\log p_\theta(\mathbf{x}|\mathbf{t})$) or a lack of equivariance (high $\mathcal{E}_{eq}$). For example, with the baseline VAE models, we see that the reconstructions in the middle row are accurate for the full sequence, while the capsule traversals obtained by sequentially rolling the initial activation (shown in the bottom row) are nothing like the input transformation (top row). In all traversals, the left-most image corresponds to $\mathbf{t}_0$, and thus input sequences of length $2L$ cover both the left and right edges when $L > 0$. Additionally in the figure captions we include the value of K, the topographic neighborhood size within a single timestep, to make the difference between the TVAE and VAE at $L = 0$ clear.

Finally, in Figures 18 & 19 at the end of the section, we include capsule traversals for models trained on MNIST with more complex transformations such as combined color & rotation (as shown in Figure 1), and combined color & perspective transforms. These models were trained in an identical manner to the other MNIST models, with the same architecture, only changing the transformation sequences of the dataset.

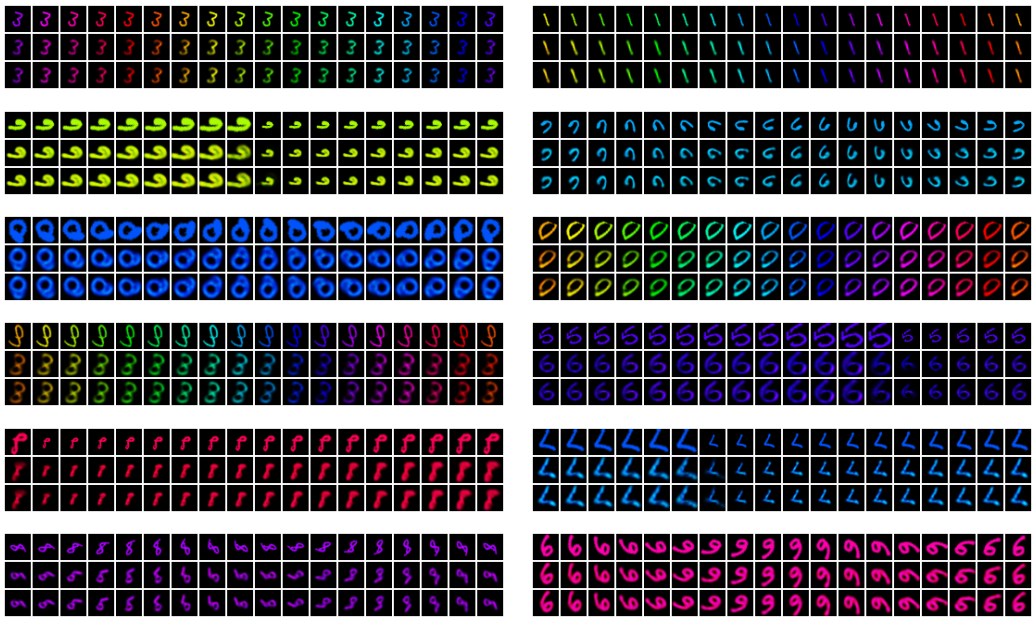

Figure 6: MNIST TVAE $L = \frac{1}{2}S$, $K = 3$

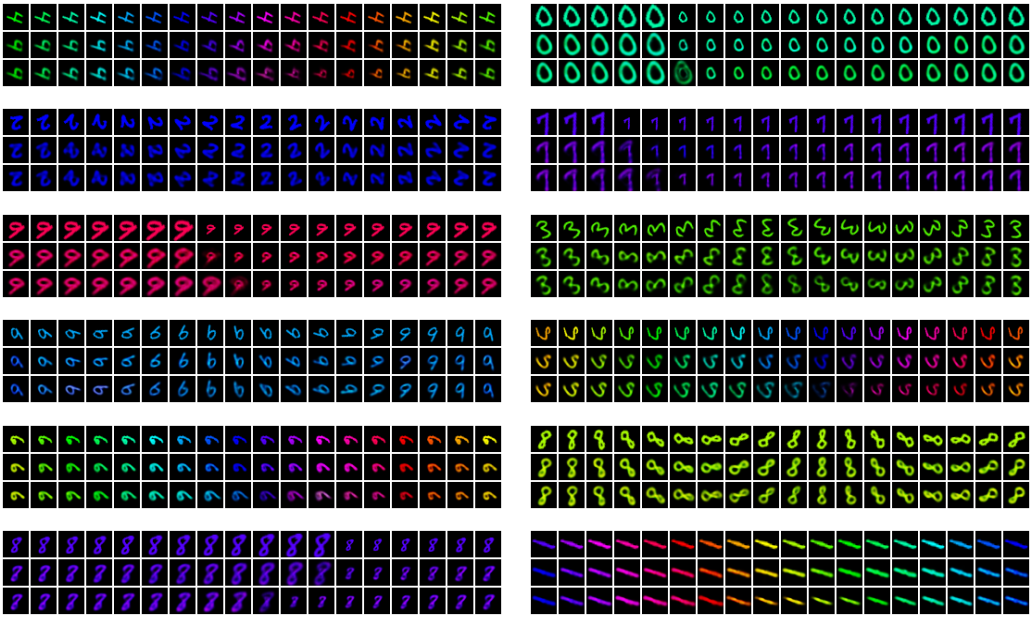

Figure 7: MNIST TVAE $L = \frac{13}{36}S$, $K = 3$

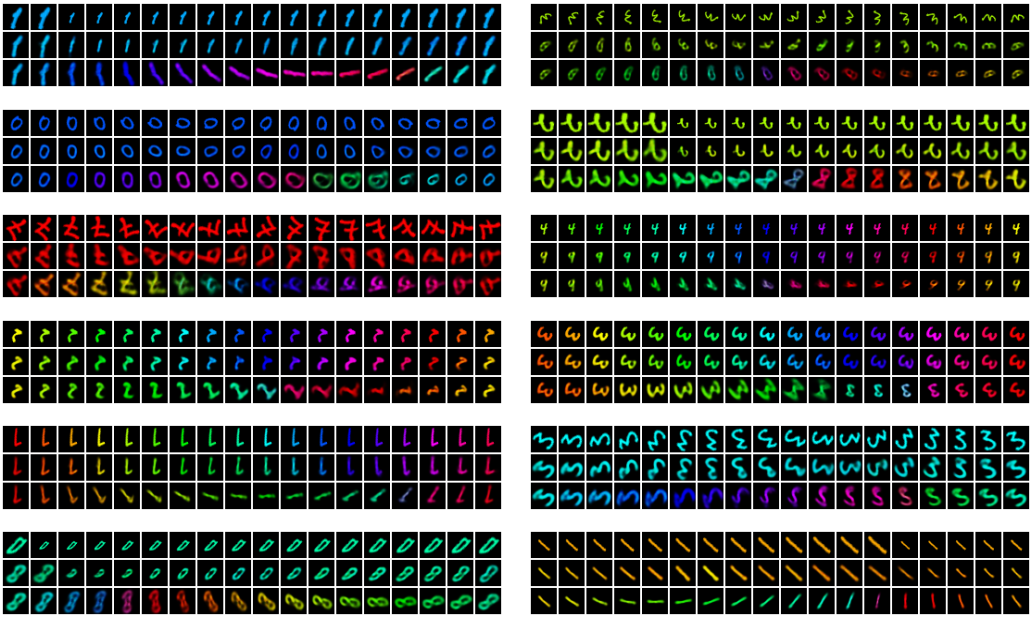

Figure 8: MNIST TVAE $L = \frac{5}{36}S$, $K = 3$

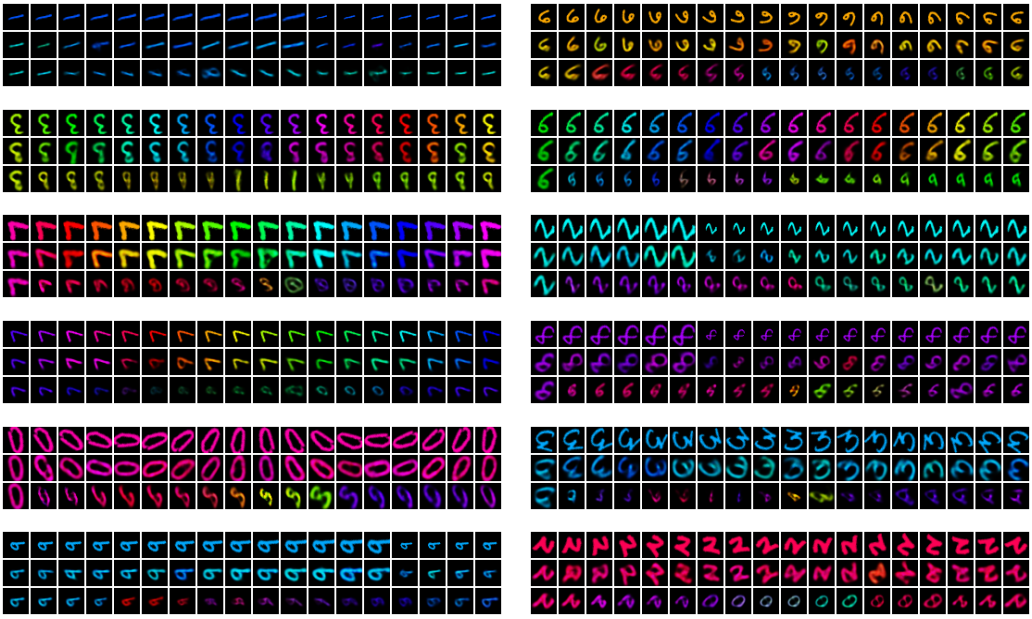

Figure 9: MNIST TVAE $L = 0$, $K = 9$

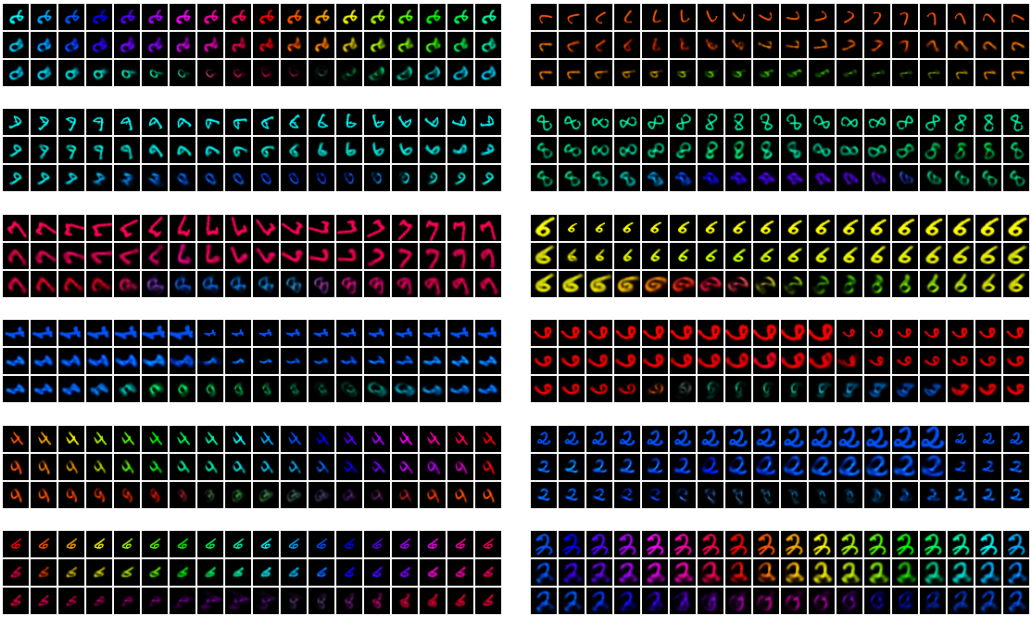

Figure 10: MNIST BubbleVAE $L = \frac{5}{36}S$, $K = 2L$

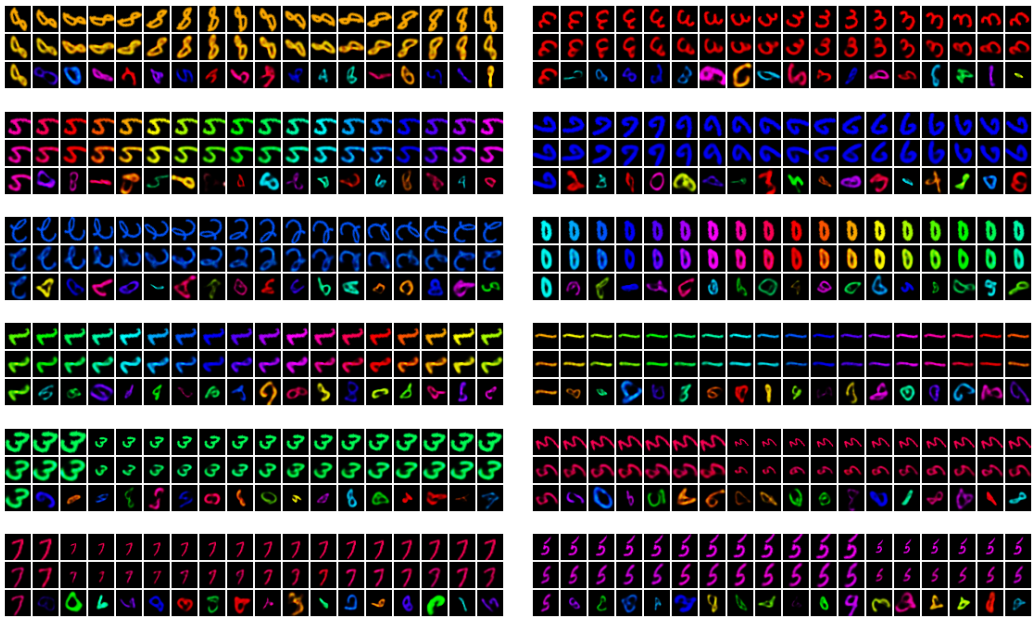

Figure 11: MNIST VAE $L = 0$, $K = 1$

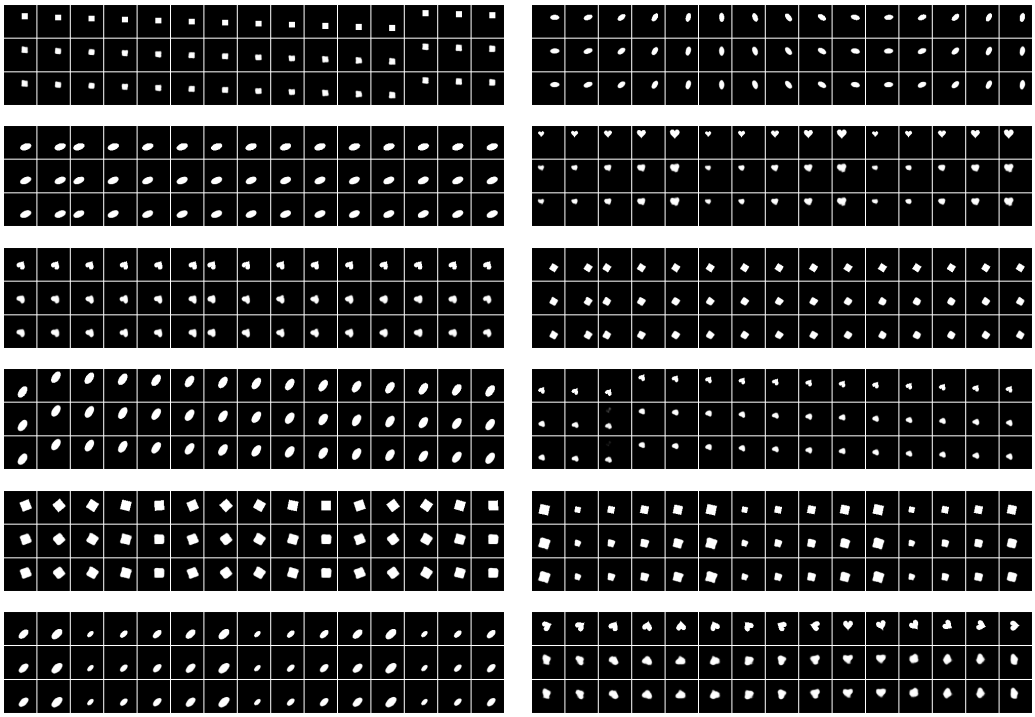

Figure 12: dSprites TVAE $L = \frac{1}{2}S$, $K = 1$

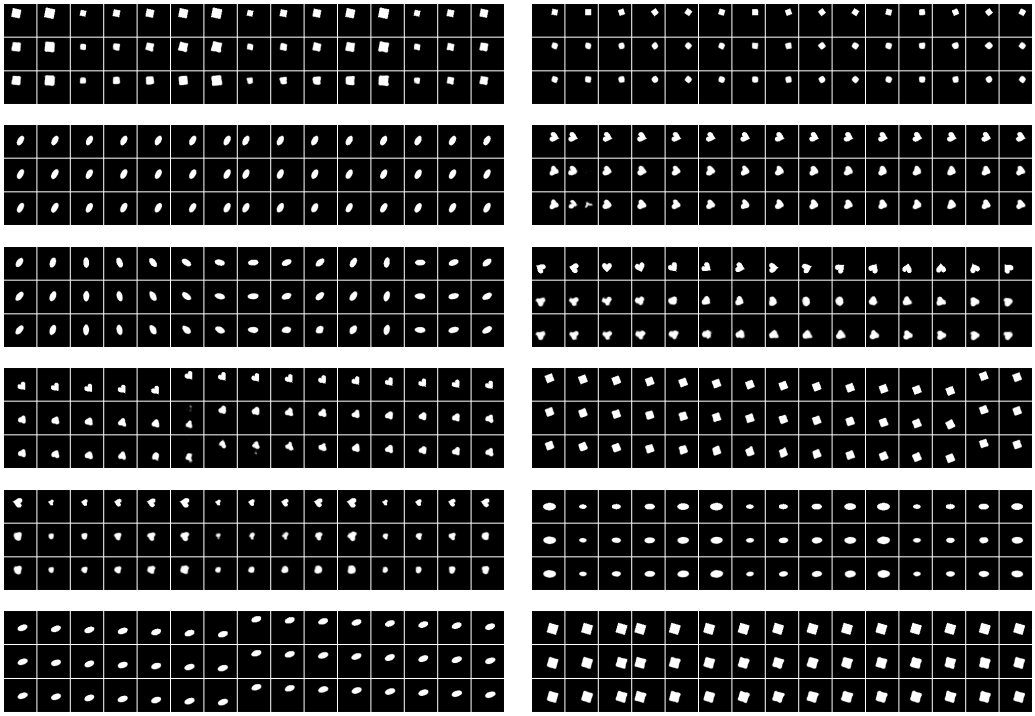

Figure 13: dSprites TVAE $L = \frac{1}{3}S$, $K = 1$

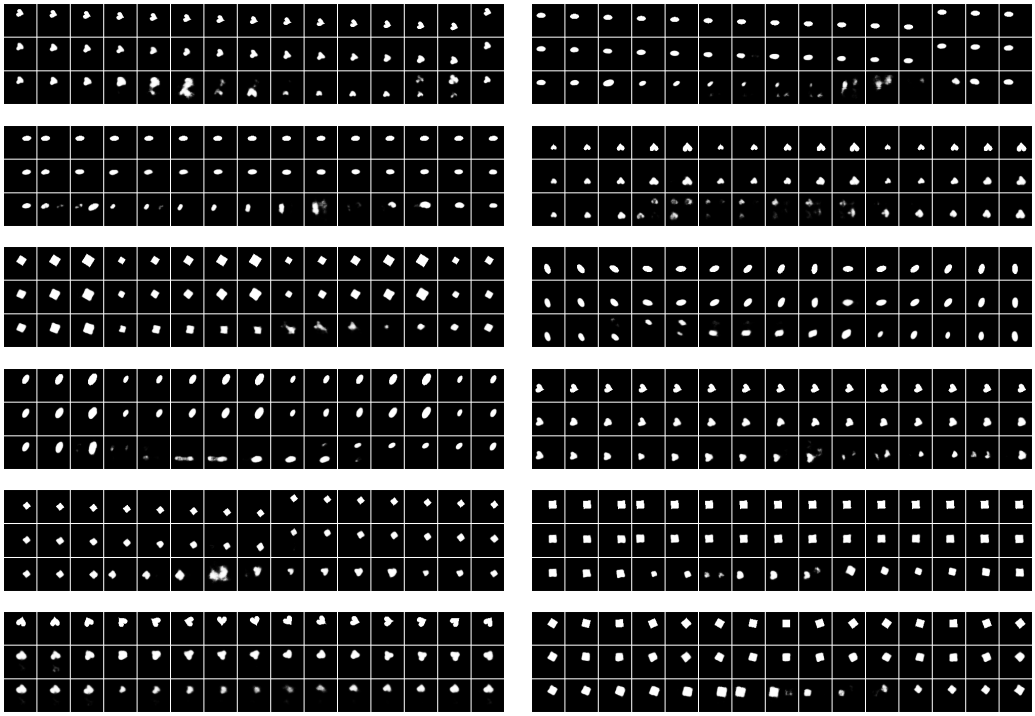

Figure 14: dSprites TVAE $L = \frac{1}{6}S$, $K = 1$

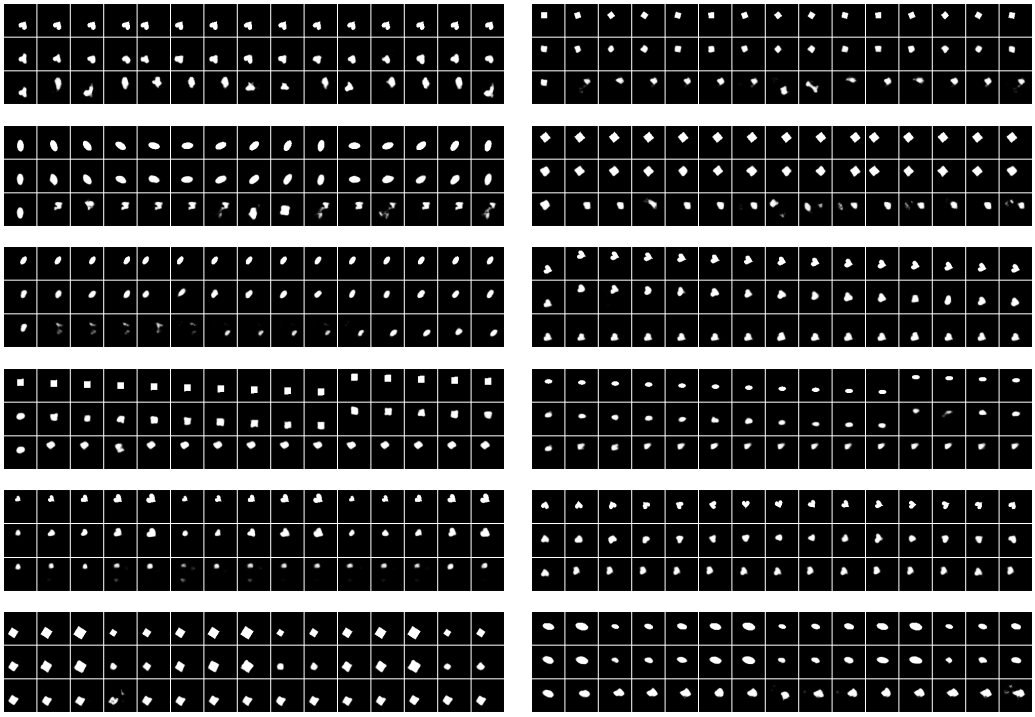

Figure 15: dSprites TVAE $L = 0$, $K = 10$

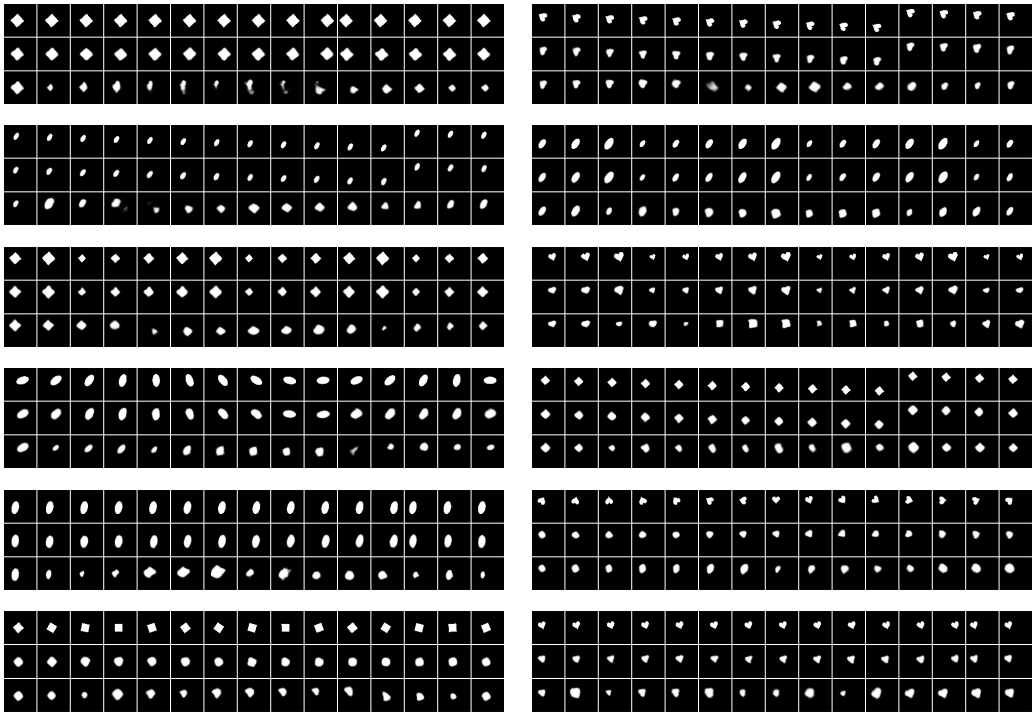

Figure 16: dSprites BubbleVAE $L = \frac{1}{6}S$, $K = 2L$

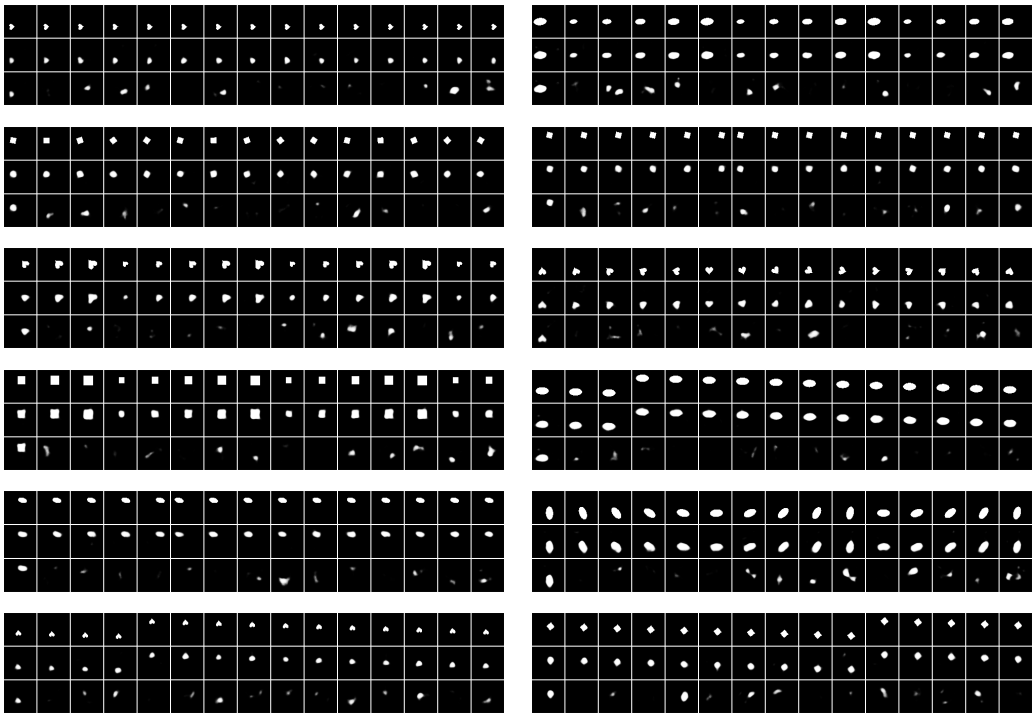

Figure 17: dSprites VAE $L = 0$, $K = 1$

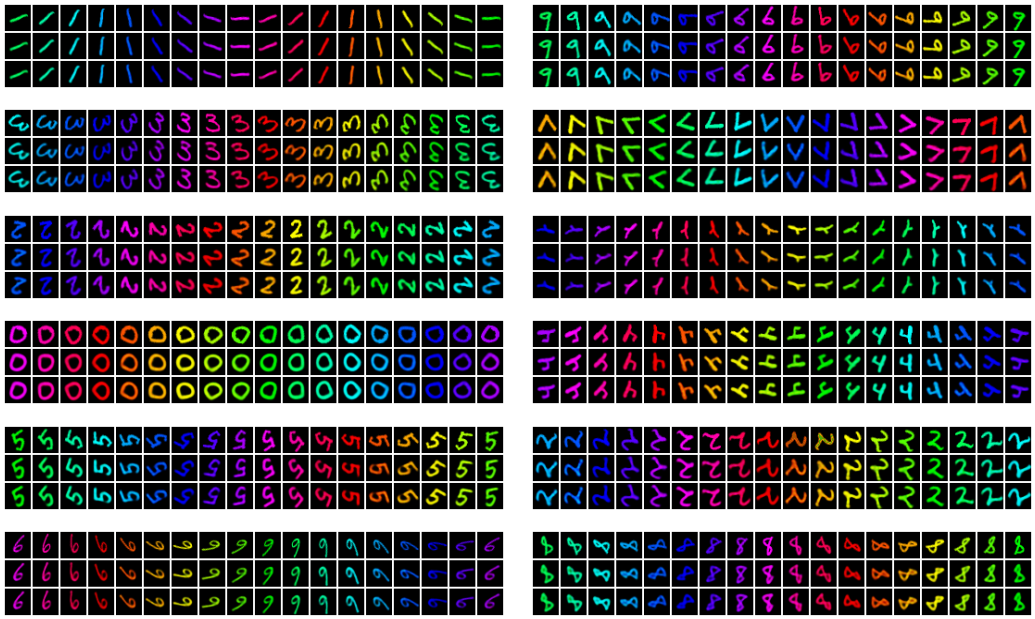

Figure 18: Combined Color & Rotation MNIST TVAE $L = \frac{13}{36}S$, $K = 3$

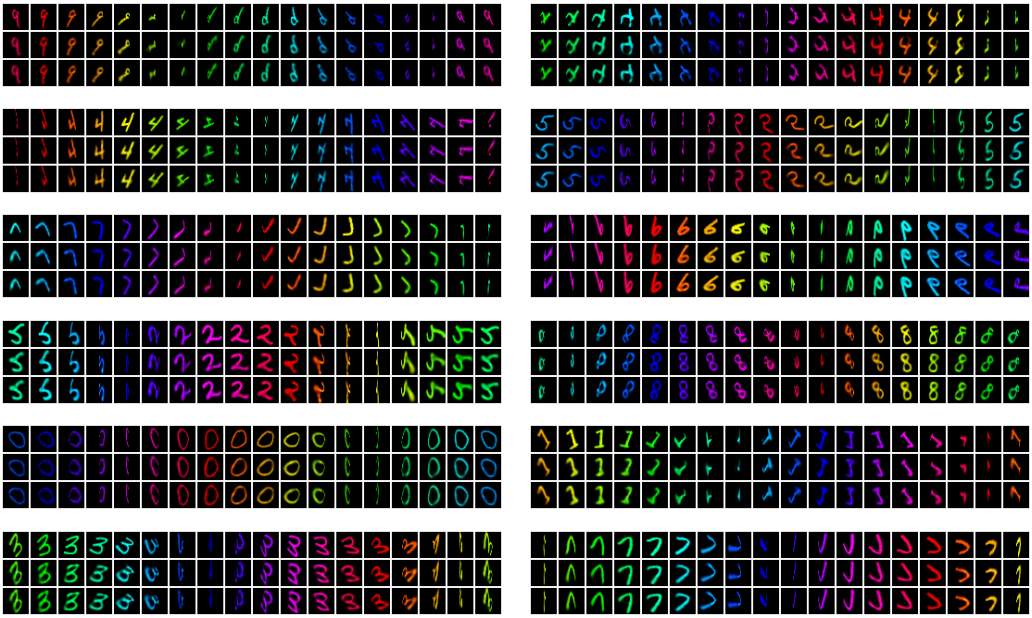

Figure 19: Combined Color & Perspective MNIST TVAE $L = \frac{13}{36}S$, $K = 3$