Contrastively Disentangled Sequential Variational Audoencoder

Anonymous Author(s)
Affiliation
Address
email

Abstract

Self-supervised disentangled representation learning is a critical task in sequence modeling. The learnt representations contribute to better model interpretability and data generation, and improves sample efficiency for downstream tasks. We propose a novel sequence representation learning method, named Contrastively Disentangled Sequential Variational Autoencoder (C-DSVAE), to extract and separate the static (time-invariant) and dynamic (time-variant) factors in the latent space. Different from previous sequential variational autoencoder (VAE) methods, we use a novel evidence lower bound (ELBO) which maximizes the mutual information (MI) between input and the latent factors, and penalizes mutual information between static and dynamic variables. We leverage contrastive estimations of the MI terms, together with simple yet effective augmentation techniques, to introduce additional inductive bias. Our experiments show that C-DSVAE significantly outperforms previous state-of-the-art methods on multiple metrics.

1 Introduction

The goal of self-supervised learning methods is to extract useful and general representations without any supervision, and to further facilitate downstream tasks such as generation and prediction [1]. Despite the difficulty of this task, many existing works have shed light on this field across different domains such as computer vision [2, 3, 4], natural language processing [5, 6, 7] and speech processing [8, 9, 10, 11, 12] (see also huge number of references in these papers). While the quality of the learnt representations improves gradually, recent research starts to put more emphasis on learning disentangled representations. This is because disentangled latent variables may capture separate variations of the data generating process which contain semantic meanings, provide the opportunity to remove unwanted variations for lower sample complexity of downstream learning [13, 14], and allow more controllable generations [15, 16, 17]. These advantages lead to a rapidly growing research area, studying various principles and algorithmic techniques for disentangled representation learning [18, 19, 20, 21, 22, 23, 24]. One concern raised in [22] is that without any inductive bias, it is extremely hard to learn meaningful disentangled representations. On the other hand, this concern could be much alleviated in the scenarios where the known structure of the data can be exploited.

In this work, we are concerned with representation learning for sequence data, which has a unique structure to utilize for disentanglement learning. More specifically, for many sequence data, the variations can be explained by a dichotomy of a static (time-invariant) factor and dynamic (time-variant) factors, each varies independently from the other. For example, representations of a video recording the movements of a cartoon character should be disentangled into the character identity (static) and the actions (dynamic). For audio data, the representations shall be able to separate the speaker information (static) from the linguistic information (dynamic).

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

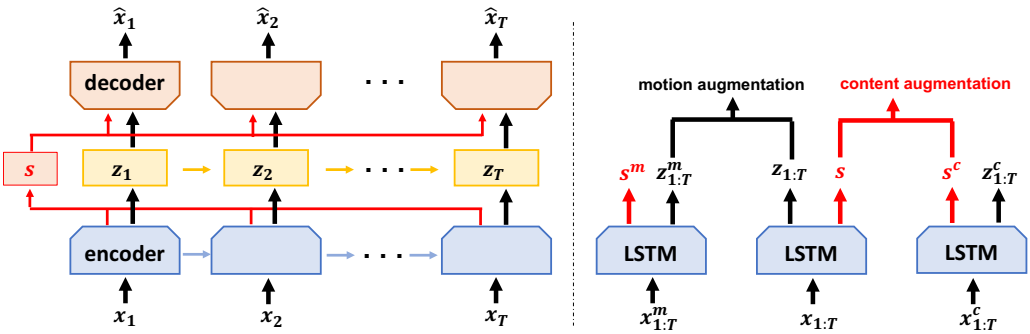


Figure 1: Illustration of the C-DSVAE model. **Left panel**. General structure of the sequence-to-sequence autoencoding process: each frame is passed to the LSTM cell; dynamic factors $z_{1:T}$ are extracted for each time step; the static factor s is extracted by summarizing the full sequence; generation/reconstruction frame i depends on s and z_i. **Right panel**. Contrastive learning module of C-DSVAE: $x^m_{1:T}$ is the motion augmentation of $x_{1:T}$ and $x^c_{1:T}$ is its content augmentation; dynamic factors $z^m_{1:T}$ of $x^m_{1:T}$ can be seen as a positive sample to the anchor $z_{1:T}$ in contrastive estimation, and similarly the static factor s^c of $x^c_{1:T}$ is a positive example for s.

We propose Contrastively Disentangled Sequential Variational Audoencoder (C-DSVAE), a method for achieving clean separation of the static and the dynamic factors for sequence data. Our method extends the previously proposed sequential VAE framework, and performs learning with a different evidence lower bound (ELBO) which naturally contains mutual information (MI) terms for encouraging disentanglement. Due to the difficulty in estimating high dimensional complex distributions (e.g., for the dynamic factors), we further incorporate contrastive estimation of the mutual information terms with systematic data augmentation strategies which affect either the static or the dynamic factors of input sequences. The new estimation method turns out to be more effective than the minibatch sampling-based estimate, and introduces additional inductive bias towards invariances. To our knowledge, we are the first to synergistically combine contrastive estimation and sequential generative models in a principled manner for learning disentangled representations. We demonstrate C-DSVAE on 4 datasets from the video and audio domains, and the experimental results show that our method consistently outperforms previous methods, both quantitatively and qualitatively.

2 Method

We denote the input sequence as $x_{1:T} = \{x_1, x_2, ..., x_T\}$ where x_i represents the input features at time step i (e.g., x_i could be a video frame or spectrogram features of a short audio segment), and T is the sequence length. The latent representations are divided into the static factor s, and the dynamic factors $z_{1:T}$ where z_i is the learnt dynamic representation at time step i.

2.1 Probabilistic Model

We assume that in the ground-truth generation process, z_i depends on $z_{<i} = \{z_0, z_1, ..., z_{i-1}\}$ where $z_0 = \mathbf{0}$, and the observation x_i is independent of other frames conditioned on z_i and s. Furthermore, we assume the static variable s and the dynamic variables $z_{1:T}$ are independent from each other, i.e., $p(s, z_{1:T}) = p(s)p(z_{1:T})$. Formally, let $z = (s, z_{1:T})$ and we have

$$p(x_{1:T}, z) = p(z)p(x_{1:T}|z) = [p(s)\prod_{i=1}^{T} p(z_i|z_{<i})]\prod_{i=1}^{T} p(x_i|z_i, s) \tag{1}$$

$p(z)$ is the prior for sampling z. Our formulation captures the general intuition that we can separate the variations of time sequences into the time-dependent component (described by z_i's), and a static component (described by s) which remains the same for frames in the same sequence but differ between sequences. For example, in the cartoon character video (see Fig. 3), the hair, shirt and pants should be kept the same when the character walks around, but different videos can have different characters. Similarly, for speech utterances, the phonetic transcription controls the vocal tract motion

and the sound being produced over time, but the speaker identity remains the same for each utterance. In this work, we refer to the dynamic component as "motion" and the static component as "content".

For the prior distributions, we choose $p(s)$ to be the standard Gaussian $\mathcal{N}(0, I)$, and $p(z_i|z_{<i})$ to be $\mathcal{N}(\mu(z_{<i}), \sigma^2(z_{<i}))$ where $\mu(\cdot)$ and $\sigma(\cdot)$ are modeled by LSTMs [25]. In operations, the drawn sample s is shared throughout the sequence. To draw a sample of z_t, we first take a sample of z_{t-1} as the input of LSTM. Forwarding the LSTM for one step gives us the distribution of z_t, from which we draw a sample with the reparameterization trick.

To extract the latent representations given only observed data $x_{1:T}$ where motion and content are mixed together, we learn a posterior distribution $q(z|x_{1:T})$ where motion and content are disentangled. That is, similar to the prior, our posterior has a factorized form:

$$q(z|x_{1:T}) = q(z_{1:T}, s|x_{1:T}) = q(z_{1:T}|x_{1:T})q(s|x_{1:T}) = q(s|x_{1:T}) \prod_{i=1}^{T} q(z_i|z_{<i}, x_{\leq i}) \tag{2}$$

The posterior distributions are also modeled by LSTMs. A nested sampling procedure, resembling that of the prior, applies to the posterior of dynamic variables. Similar parameterization of dynamic variables by recurrent networks have been proposed in prior works [26, 27, 28]. The standard loss function for learning the latent representations is an evidence lower bound (ELBO, [29, 30], also see a derivation in the Appendix):

$$\max_{p,q} \ \mathbb{E}_{x_{1:T} \sim p_D} \mathbb{E}_{q(z|x_{1:T})} \left[\log p(x_{1:T}|z) - KL[q(z|x_{1:T})||p(z)]\right] \tag{3}$$

where p_D is the empirical data distribution. Due to the parameterization of the posterior where s and $z_{1:T}$ are mutually independent, the KL-divergence term reduces to

$$KL[q(z|x_{1:T})||p(z)] = KL[q(s|x_{1:T})||p(s)] + KL[q(z_{1:T}|x_{1:T})||p(z_{1:T})], \tag{4}$$

where the second term is approximated with sample trajectories of the dynamic variables.

2.2 Our approach: Mutual information-based disentanglement

Several existing sequence representation learning methods [29, 30, 17] are built based on the loss function (3). However, this formulation also brings several issues. The KL-divergence regularizes the posterior of static or dynamic to be close to the corresponding prior. When modeling them with powerful neural architectures like LSTMs, it is possible for the KL-divergence to be close to zero yet at the same time, the posteriors become non-informative of the inputs; this is a potential issue for deep generative models [31]. Techniques have been proposed to alleviate this issue, including adjusting the relative weights of the loss terms to regularize the capacity of posteriors [19], replacing the individual posteriors in the KL terms with aggregated posteriors [32, 33, 34] and enforcing structures (such as disentangled representations) in the posteriors [21, 20, 35].

Our principled approach for learning useful representations for sequences is inspired by these prior work, and at the same time incorporating the unique sequential structure of data. Without inductive biases, the goal of disentanglement can hardly be achieved since it is possible to find entangled s and $z_{1:T}$ that explains the data equally well (in fact, by Theorem 1 of [22], there could exist infinitely many such entangled factors). The problem may seem less severe in our formulation, since s is shared across time and it is hard for s to capture the dynamics. Nevertheless, since both s and z_i are used in generating x_i, it is still possible for z_i to carry some static information. In the extreme case where each z_i encompasses s, s is no longer indispensable in generation. This issue motivated prior works to optimize (estimate of) mutual information among latent variables and inputs [30, 17, 36].

Our method tries to achieve clean disentanglement of s and $z_{1:T}$, by approximately optimizing the following objective function, which introduces additional mutual information terms to the naive evidence lower bound in (3):

$$\begin{aligned}
&\max_{p,q} \mathbb{E}_{x_{1:T} \sim p_D} \mathbb{E}_{q(z|x_{1:T})}[\log p(x_{1:T}|z)] - KL[q(z)||p(z)] \\
&\quad = \mathbb{E}_{x_{1:T} \sim p_D} \mathbb{E}_{q(z|x_{1:T})}[\log p(x_{1:T}|z)] - (KL[q(s|x_{1:T})||p(s)] + KL[q(z_{1:T}|x_{1:T})||p(z_{1:T})]) \\
&\qquad + I_q(s; x_{1:T}) + I_q(z_{1:T}; x_{1:T}) - I_q(z_{1:T}; s)
\end{aligned} \tag{5}$$

where aggregated posteriors are defined as $q(z) = q(s, z_{1:T}) = \mathbb{E}_{p_D}[q(s|x_{1:T})q(z_{1:T}|x_{1:T})]$, $q(s) = \mathbb{E}_{p_D}[q(s|x_{1:T})]$, $q(z_{1:T}) = \mathbb{E}_{p_D}[q(z_{1:T}|x_{1:T})]$, and mutual information terms are defined as

Figure 2: Data augmentations on SM-MNIST and Sprites. **Left panel**. First row is raw input sequence of moving digits. Second row reverses sequence order or raw sequence, serving as content augmentation. Third row stretches frames and enhances the color, which changes the digit styles but does not change the digit movement and thus forms a motion augmentation. **Right panel**. First row is raw input character video. Second row is content augmentation with a random order. Third row is motion augmentation produced by distorting colors and adding Gaussian noises.

$I_q(s; x_{1:T}) = \mathbb{E}_{q(s,x_{1:T})}[\log \frac{q(s|x_{1:T})}{q(s)}]$, $I_q(z_{1:T}; x_{1:T}) = \mathbb{E}_{q(z_{1:T},x_{1:T})}[\log \frac{q(z_{1:T}|x_{1:T})}{q(z_{1:T})}]$ ($I_q(s; x_{1:T})$ is defined similarly). The intuition behind (5) is simple: besides explaining the data and matching the prior, $z_{1:T}$ and s should contain useful information from $x_{1:T}$ while excluding redundant information from each other. We further justify (5) by showing that it still forms a valid ELBO.

Theorem 1. *With our parameterization of* $(s, z_{1:T})$*,* (5) *is a valid lower bound of the data log-likelihood* $\mathbb{E}_{x_{1:T}\sim p_D} \log(x_{1:T})$.

The full proof can be found in Appendix. With this guarantee, we can follow the motivation of [19, 21] to adjust additional weight coefficient α to the KL terms, and β to the MI terms $I(s; x_{1:T})$ and $I(z_{1:T}; x_{1:T})$. It remains to estimate the loss (5) for optimization. The per-sample KL term for $z_{1:T}$ is estimated with usual Monte-Carlo sampling with trajectories of $z_{1:T}$. For the mutual information terms, we tried two estimates. The first estimate uses a standard mini-batch weighted sampling (MWS) following [21, 30]. Due to the high dimensionality of $z_{1:T}$ and the complex dependency between time steps, it may be hard to estimate the distributions accurately, so we also explore a non-parametric, contrastive estimate of $I_q(s; x_{1:T})$ and $I_q(z_{1:T}; x_{1:T})$ with additional data augmentation, which we detail below.

2.3 C-DSVAE: Contrastive estimation with augmentation

A contrastive estimate of $I(z_{1:T}; x_{1:T})$ can be defined as follows

$$\mathcal{C}(z_{1:T}) = \mathbb{E}_{p_D} \log \frac{\phi(z_{1:T}, x^+_{1:T})}{\phi(z_{1:T}, x^+_{1:T}) + \sum_{j=1}^{n} \phi(z_{1:T}, x^j_{1:T})} + \log(n+1). \tag{6}$$

where x^+ is a "positive" sequence, while x^j, $j = 1, \ldots, n$ is a collection of n negative sequences. When $\phi(z_{1:T}, x_{1:T}) = \frac{q(x_{1:T}|z_{1:T})}{q(x_{1:T})}$, one can show that (6) approximates $I_q(z_{1:T}, x_{1:T})$; see Appendix A for a proof. To implement (6), $z_{1:T}$ is the trajectory obtained by using the mean at each time step from $q(z_{1:T}|x_{1:T})$ for a input sequence $x_{1:T}$, while $x^j_{1:T}$ is a randomly sampled sequence from the minibatch. Similar contrastive estimate is defined for s. To provide meaningful positive samples that encourage invariance of learnt representations, we obtain $x^+_{1:T}$ by systematically perturbing $x_{1:T}$. In Sec A.4, we discuss how augmented data gives good estimate of (6) under additional assumptions.

Content augmentation The static factor (e.g., character identity in videos or the speaker in audios) is shared across all the time steps, and should not be affected by the exact order of the frames. We therefore randomly shuffle or simply reverse the order of time steps to generate the content augmentation of $x_{1:T}$ and denote it by $x^c_{1:T}$. The static and dynamic latent factors of $x^c_{1:T}$, estimated by q, are denoted by s^c and $z^c_{1:T}$. This is an inexpensive yet useful strategy, applicable to both audios and videos. Similar ideas were introduced in [30] albeit not used for contrastive estimation.

Motion augmentation In motion augmentation, we would like to maintain the actions and movements while replacing the content with a meaningful alternative. Thanks to recent efforts in contrastive learning, multiple effective strategies have been proposed. For video datasets, we adopt the combination of cropping, color distortion, Gaussian blur and reshaping [4, 37]. For audio datasets, we use

classical unsupervised voice conversion algorithms [38, 39]. The motion augmentation of $x_{1:T}$ is denoted by $x^m_{1:T}$ with static factor s^m and dynamic factors $z^m_{1:T}$ estimated by q.

Note that $x^m_{1:T}$ is the motion augmentation of $x_{1:T}$, and $x_{1:T}$ in turn is also the motion augmentation of $x^m_{1:T}$. Similarly, s and s^c are mutually "positive" samples w.r.t. the static factor. During training, it is efficient to generate the augmentations for each sequence in the minibatch, and apply contrastive estimation on both original data and augmented data. This leads to the following final estimates:

$$I_q(z_{1:T}; x_{1:T}) \approx \frac{1}{2}(\mathcal{C}(z_{1:T}) + \mathcal{C}(z^m_{1:T})), \qquad I_q(s; x_{1:T}) \approx \frac{1}{2}(\mathcal{C}(s) + \mathcal{C}(s^c)) \tag{7}$$

where we used $\phi(z_{1:T}(x_{1:T}), x^*_{1:T}) = \exp(cossim(z_{1:T}, z^*_{1:T})/\tau)$ with * indicating the latent variables for augmented or negative sequences (extracted by sampling from q), and $cossim(\cdot,\cdot)$ is the cosine similarity function and $\tau = 0.5$ is a temperature parameter. Plugging (7) into (5) gives our final learning objective with contrastive estimation. We name our method contrastively disentangled sequential variational encoder (C-DSVAE), and a full model illustration is shown in Fig. 1.

In practice, we found contrastive estimation to be more effective than MWS based on Gaussian probabilities, see an ablation study in Appendix F.2. Interestingly, we observe with contrastive estimation and augmentation, $I_q(z_{1:T}; s)$ (last term in (5)) decreases even if we do not include its estimation in optimization (see Appendix F), indicating good disentanglement between $z_{1:T}$ and s.

As it will become evident in the experiments, our method achieves cleaner separation of the dynamic and static variables than previous methods. We think this is partly due to the inductive bias introduced by our approach: in order to consistently map two sequences with the same $z_{1:T}$ but different s (which varies independently from $z_{1:T}$ according to the generative model) to the same latent representation, we must discard information of input regarding s; similar argument holds for the static variables. But since at the same time, we enforce that extracted $z_{1:T}$ and s together can reconstruct $x_{1:T}$ so that no information is lost in the auto-encoding process, we must have the learnt $z_{1:T}$ and s to recover the data generating factors. To our knowledge, we are the first to use contrastive estimation with augmentation in sequential VAE, for achieving disentanglement of variations.

3 Related Work

In terms of general frameworks, disentangled representation learning methods can be categorized into GAN-based [40] and VAE-based [41] approaches. GAN-based methods like MoCoGAN [42] aim at generating videos from content noise and motion noise. These model typically do not explicitly learn the encoder from inputs to latent variables, rendering them less applicable for representation learning.

The VAE framework parameterizes both the encoder (variational posterior) and the decoder (reconstruction mapping), and optimize them jointly with a well-defined ELBO objective. The encoder allows us to directly reason about and control properties of the extracted features, which can be directly used for downstream tasks. Many variants of VAE were proposed to encourage disentanglement/interpretability of *individual latent dimensions*. β-VAE [19] adds a coefficient to the per-sample KL-divergence term to better constrain the information bottleneck. FactorVAE [20] and β-TCVAE [21] explicitly separate out the total correlation term, and adjust its coefficient in the objective to encourage per-dimension disentanglement of the *aggregated* posterior. From a theoretical stand point, [22] points out that without further inductive biases, it is impossible to learn disentangled representations with standard VAE.

Vanilla VAE is extended to the recurrent version to handle sequence data, and prior works have explored different approaches for separating content and motion. FHVAE [43] first attempted a hierarchical VAE model which uses "sequence-dependent variables" (corresponding to the speaker factor) and "sequence-independent variables" (corresponding to the transcription factors) for modeling speech sequences, and performs learning with the per-sequence ELBO while respecting the hierarchy in designing the posterior. FHVAE does not use additional loss terms for encouraging disentanglement. Different from FHVAE, DSVAE [29] explicitly models the static and dynamic factors in its graphical model and the posterior has a factorized form. A few recent works (including ours) inherit the cleaner formulation of DSVAE in terms of graphical model and prior/posterior parameterization. S3VAE [30] introduced additional loss terms to the objective of DSVAE in a more ad-hoc fashion, such as a triplet loss that encourages invariance of learnt static factors to permuted frame order (spiritually similar to our contrastive estimation for $I(s, x_{1:T})$), an additional prediction loss on $z_{1:T}$ leveraging

Table 1: Disentanglement metrics on Sprites.

Methods	Acc↑	IS↑	H(y|x)↓	H(y)↑
MoCoGAN	92.89%	8.461	0.090	2.192
DSVAE	90.73%	8.384	0.072	2.192
R-WAE	98.98%	8.516	0.055	2.197
S3VAE	99.49%	8.637	0.041	2.197
C-DSVAE	**99.99%**	**8.871**	**0.014**	**2.197**

Table 2: Disentanglement metrics on MUG.

Methods	Acc↑	IS↑	H(y|x)↓	H(y)↑
MoCoGAN	63.12%	4.332	0.183	1.721
DSVAE	54.29%	3.608	0.374	1.657
R-WAE	71.25%	5.149	0.131	1.771
S3VAE	70.51%	5.136	0.135	1.760
C-DSVAE	**81.16%**	**5.341**	**0.092**	**1.775**

external supervision for the motion labels, and MI minimization for $I(s, z_{1:T})$. We differ from S3VAE in that we naturally introduce the MI terms from the fundamental principle of VAE, and use augmentation instead of external supervision for learning the dynamic factors. Also based on the DSVAE formulation, R-WAE [17] proposes to replace the distance measure between the aggregated posterior and the prior with the Wasserstein distance, with the belief that the KL divergence is too restrictive. In the end, Maximum Mean Discrepancy (MMD) based estimate or GAN-based estimate of Wasserstein distance is used, both of which require heavier tuning of hyperparameters.

Another relevant research area is that of nonlinear ICA, which tries to understand the conditions under which disentanglement can be achieved, possibly with contrastive learning or VAE [44, 45, 46, 23, 24]. Our general approach is well-aligned with the current understanding in this direction. That is, while general disentanglement (per-dimension disentanglement) is hard to achieve without stringent assumptions on the data generation process, we can hope to achieve certain level of disentanglement (in our case, group-wise disentanglement between static and dynamic factors) given auxiliary information (which we provide through augmentation). Our intuitive arguments for the reason of separation (see last paragraph of Sec 2.3) shows that VAE and contrastive learning may be complementary to each other, and combining them could potentially achieve better disentanglement.

4 Experiments

We compare C-DSVAE with state-of-the-art sequence disentanglement learning methods: FH-VAE [43], MoCoGAN [42], DSVAE [29], S3VAE [30] and R-WAE [17], with the same experimental setups used by them.

4.1 Datasets

Sprites [47] is a cartoon character video dataset. Each character's (dynamic) motion is captured by three actions (walking, spellcasting, slashing) and three directions (left, front, right). The (static) content comprises of each character's skin color, tops color, pants color and hair color. Each color has 6 variants. Each sequence is composed of 8 frames of RGB images with size 64×64.

MUG [48] is a facial expression video dataset. The static factor corresponds to the person's identity. Each individual performs 6 expressions (motion): anger, fear, disgust, happiness, sadness and surprise. Following [42], each sequence contains 15 frames of RGB images with size 64×64 (after resizing).

SM-MNIST ([49], Stochastic Moving MNIST) records the random movements of two digits. Each sequence contains 15 grayscale images of size 64×64. Individual digits are collected from MNIST.

TIMIT [50] is a corpus of read speech for acoustic-phonetic studies and speech recognition. The utterances are produced by American speakers of eight major dialects reading phonetically rich sentences. Following [43, 17], we extract per-frame spectrogram features (with a shift size of 10ms) from audio, and segments of 200ms duration (20 frames) are chunked from original utterances and are then treated as independent sequences for learning.

All datasets are separated into training, validation and testing sets following [43, 30, 17].

4.2 Disentanglement Metrics

The quantitative performance measures the model from two aspects: first, by fixing either the static or dynamic factors and randomly sample the other, how well the fixed factor can be recognized; and

Table 3: Disentanglement metrics on SM-MNIST.

Methods	Acc↑	IS↑	H(y|x)↓	H(y)↑
MoCoGAN	74.55%	4.078	0.194	0.191
DSVAE	88.19%	6.210	0.185	2.011
R-WAE	94.65%	6.940	0.163	2.147
S3VAE	95.09%	7.072	0.150	2.106
C-DSVAE	**97.84%**	**7.163**	**0.145**	**2.176**

Table 4: Disentanglement metrics on TIMIT.

Methods	content EER↓	motion EER↑
FHVAE	5.06%	22.77%
DSVAE	5.64%	19.20%
R-WAE	4.73%	23.41%
S3VAE	5.02%	25.51%
C-DSVAE	**4.03%**	**31.81%**

second, for the randomly sampled factor, how different the generated sequence is from the original one. To this end, we pretrain a separate classifier C to identify the static or dynamic factors (if available). The classifier is carefully trained with full supervision and thus qualifies as a judge.

We use 5 metrics to evaluate the disentanglement performance: accuracy, inception score, inter-entropy, intra-entropy and equal error rate. Accuracy (Acc) measures how well the fixed factor can be identified by the classifier C. Inception Score (IS) [51] computes the KL-divergence between the conditional predicted label distribution $p(y|x_{1:T})$ and marginal predicted label distribution $p(y)$ from C. Inter-Entropy is similar to IS but measures the diversity solely on the marginal $p(y)$; higher $H(y)$ indicates that the generated sequences are more diverse. Intra-Entropy measures the entropy over $p(y|x)$; lower intra-entropy means more confident prediction. Equal error rate (EER) [52] is used in the TIMIT experiment to evaluate speaker identification accuracy.

4.3 Hyperparameters and Architectures

As is commonly done in VAE learning [19], in (5) we add coefficients α to the KL terms and β to the contrastive terms, while the coefficient of $I(s; z_{1:T})$ is fixed to be 1. In our experiments, α is tuned over $\{0.6, 0.9, 1.0, 2.0\}$ and β is tuned over $\{0.1, 0.2, 0.5, 0.7, 1.0, 2.0, 5.0\}$. We use the Adam optimizer with learning rate chosen from $\{0.0005, 0.001, 0.0015, 0.002\}$ through grid search. Models are trained until convergence and the one with best validation set accuracy is chosen for testing. Our network architectures follow those of [30]: motion priors and posteriors are both parameterized by uni-directional LSTMs, which output μ and σ at each time step, while the content posterior produces μ and σ for the whole sequence with a bi-directional LSTM. See Appendix C.2 for further details.

4.4 Quantitative Results

Tables 1, 2, 3, 4 demonstrate the disentanglement evaluations on the datasets; ↑ means the higher the value the better the performance and ↓ means the reverse. In Table 1, we fix $z_{1:T}$ and randomly sample s from $p(s)$ for the evaluation. We observe that R-WAE, S3VAE, C-DSVAE can all achieve high accuracies in predicting a total of 9 classes (3 action $\times$ 3 directions), and C-DSVAE still outperforms the others. On the other hand, when we fix s and randomly sample $z_{1:T}$, all three methods and DSVAE achieve near perfect accuracies for predicting characters identities (and thus we do not show this results here). Similarly, in Table 2, we evaluate the ability to maintain the facial expression by sampling s and fixing $z_{1:T}$. C-DSVAE outperforms R-WAE and S3VAE by over 10% relatively in the prediction accuracy, and its intra-entropy is lower by 30% relatively. Table 3 measures how well the static factor s, which corresponds to a total of 100 digit combinations, is recognized when the dynamic factors are randomly sampled (the dynamic factors are almost continuous and hard to be categorized). While R-WAE, S3VAE, and C-DSVAE can all generate high-quality moving digits (see Sec 4.5), C-DSVAE non-trivially outperforms the others on all metrics.

For TIMIT, we extract both the content factor (s, corresponds to the speaker identity) and the dynamic factors ($z_{1:T}$, corresponds to transcription) for the sequences. And for each of them, we compute the cosine similarity of representations between pairs of sequences, based on which we perform thresholding to classify if the two sequences are produced by the same speaker. Through varying the threshold, we compute the EER for the speaker verification task [43, 29]. For a well disentangled latent space, we expect the EER on s (content EER) to be low, while the EER on $z_{1:T}$ (motion EER) to be high. The EER results are given in Table 4. Compared with the previous SOTA by R-WAE, C-DSVAE reduces the content EER by over 10% relatively.

Figure 3: Manipulating the static and dynamic factors for Sprites. Row 1, 2, 4 are raw test input sequences, while row 3, 5 are manipulated generations. **Left panel**. Row 3 uses s from row 1 and $z_{1:T}$ from row 2. Row 5 uses s from row 1 and $z_{1:T}$ from row 4. **Right panel**. Row 3 uses $z_{1:T}$ from row 1 and s from row 2. Row 5 uses $z_{1:T}$ from row 1 and s from row 4.

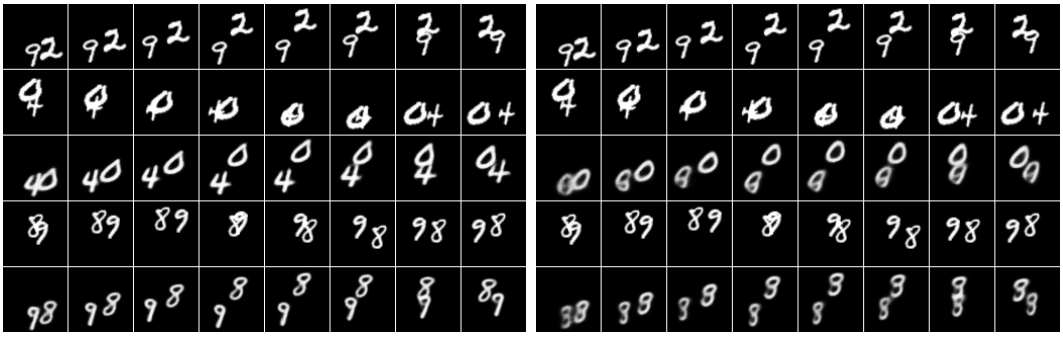
(a) C-DSVAE (b) S3VAE

Figure 4: Fix the facial expression on MUG. Row 1, 2, 4 are raw test sequences with different facial expressions. We replace $z_{1:T}$ of row 2, 4 with $z_{1:T}$ of row 1 and compare the performance of C-DSVAE and S3VAE and hence row 3, 5 should display happiness. S3VAE produces blurred sequences (zoom in to see the details) while C-DSVAE generations are clean and sharp.

4.5 Qualitative Results

We now compare different methods qualitatively with the following tasks: given the extracted static factor s and dynamic factors $z_{1:T}$, we fix one of them and manipulate the other to see if we observe the desired variations in generated samples. Fig. 3 gives example generations for Sprites. We observe that the characters are well maintained when we fix s (left) and motion is well maintained when we fix $z_{1:T}$ (right), demonstrating clean disentanglement is achieved by C-DSVAE. Fig. 4 compares the generation when fixing the motion (facial expression) for C-DSVAE and S3VAE on MUG. We replace $z_{1:T}$ for each input sequence with the one from the first row, and therefore all images are expected to have the smiling expression. C-DSVAE and S3VAE both generate recognizable smiling faces and preserve the individual's identity, but the sequences generated by S3VAE have blur and are less sharp. In Fig. 5, we fix the digit movements and takes the contents (digit identities) from various inputs. Entangled factors might cause extracted $z_{1:T}$ to carry information of s. Row 3 of Fig. 5b shows such an example produced by R-WAE where one digit's motion factor carries the content information of "9" from row 1.

Note that S3VAE and R-WAE are often capable of generating recognizable videos, which is validated by our classifier $\mathcal{C}$ with good accuracies. Nevertheless, the results here show that C-DSVAE can learn better disentangled factors for high-quality generation. Results of additional generation tasks can be found in Appendix D, where we swap latent factors of different sequences, and show the learnt latent space can be smoothly interpolated.



(a) C-DSVAE (b) R-WAE

Figure 5: Fix the movement and replace the digits. Row 1, 2, 4 are input test sequences with different digits and movements. Row 3, 5 use $z_{1:T}$ from row 1 while retaining own s's. The sequences generated by C-DSVAE are clean and consistent, while R-WAE sometimes produces blurred or incorrect digits.

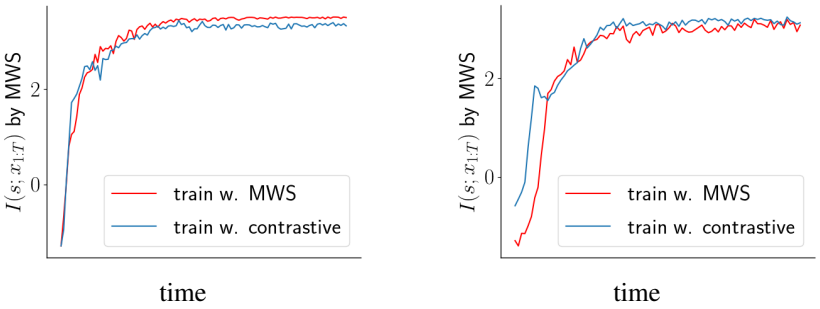


Figure 6: Dev set learning curves of MI between latent factors and inputs estimated by MWS, when training uses either MWS or contrastive estimation for the MI terms. Though learning with different estimates, the dev set MI curves closely follow each other. **Left**: Sprites. **Right**: TIMIT.

4.6 Other analysis

As mentioned in Sec 2, we have estimated the MI terms using both MWS and contrastive estimation with augmentation. In Fig. 6, we show the MWS estimation of $I(s; x_{1:T})$ on the dev set improves even when the C-DSVAE training objective uses the contrastive estimation. This shows that contrastive estimation could be a surrogate for MI in generative models, and its additional inductive bias can be a main contributor to superior performance (see Appendix F.2 for quantitative comparisons).

In Appendix E.1, we show that optimizing contrastive estimation with augmentation already leads to decreasing $I_q(s, z_{1:T})$. Further in Appendix F.1, our ablation study shows that motion augmentation and content augmentation are complementary to each other, and combining them gives the best performance.

5 Conclusion

We have proposed Contrastively Disentangled Sequential Variational Autoencoder (C-DSVAE) to learn disentangled static and dynamic latent factors for sequence data without external supervision. Our learning objective is a valid ELBO derived differently from prior works, and naturally encourages disentanglement. C-DSVAE uses contrastive estimation of mutual information terms to further injects inductive biases. Our method achieves the state-of-the-art results on multiple datasets, in terms of disentanglment metrics and generation quality. In the future, we may extend C-DSVAE to other domains such as text, biology and weather prediction, and to capture longer dependencies.

References

[1] 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.

[2] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[3] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *ICLR*, 2018.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[5] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, 2018.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[7] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[8] Alexei Baevski, Michael Auli, and Abdelrahman Mohamed. Effectiveness of self-supervised pre-training for speech recognition. *arXiv preprint arXiv:1911.03912*, 2019.

[9] Yu-An Chung and James Glass. Generative pre-training for speech with autoregressive predictive coding. In *ICASSP*, 2020.

[10] Weiran Wang, Qingming Tang, and Karen Livescu. Unsupervised pre-training of bidirectional speech encoders via masked reconstruction. In *ICASSP*, 2020.

[11] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *arXiv preprint arXiv:2006.11477*, 2020.

[12] Junwen Bai, Weiran Wang, Yingbo Zhou, and Caiming Xiong. Representation learning for sequence data with deep autoencoding predictive components. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=Naqw7EHIfrv`.

[13] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. *arXiv preprint arXiv:1706.08033*, 2017.

[14] Emily Denton and Vighnesh Birodkar. Unsupervised learning of disentangled representations from video. *arXiv preprint arXiv:1705.10915*, 2017.

[15] Yizhe Zhu, Mohamed Elhoseiny, Bingchen Liu, Xi Peng, and Ahmed Elgammal. A generative adversarial approach for zero-shot learning from noisy texts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1004–1013, 2018.

[16] Yu Tian, Jian Ren, Menglei Chai, Kyle Olszewski, Xi Peng, Dimitris N Metaxas, and Sergey Tulyakov. A good image generator is what you need for high-resolution video synthesis. *arXiv preprint arXiv:2104.15069*, 2021.

[17] Jun Han, Martin Renqiang Min, Ligong Han, Li Erran Li, and Xuan Zhang. Disentangled recurrent wasserstein autoencoder. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=O7ms4LFdsX`.

[18] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *arXiv preprint arXiv:1606.03657*, 2016.

[19] 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*, 2016.

[20] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR, 2018.

[21] Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *NIPS*, 2018.

[22] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pages 4114–4124. PMLR, 2019.

[23] Ilyes Khemakhem, Diederik P. Kingma, Ricardo Pio Monti, and Aapo Hyvärinen. Variational autoencoders and nonlinear ica: A unifying framework, 2020.

[24] Francesco Locatello, Ben Poole, Gunnar Raetsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. Weakly-supervised disentanglement without compromises. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6348–6359. PMLR, 13–18 Jul 2020. URL `http://proceedings.mlr.press/v119/locatello20a.html`.

[25] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

[26] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. *arXiv preprint arXiv:1506.02216*, 2015.

[27] Anirudh Goyal, Alessandro Sordoni, Marc-Alexandre Côté, Nan Rosemary Ke, and Yoshua Bengio. Z-forcing: Training stochastic recurrent networks. *arXiv preprint arXiv:1711.05411*, 2017.

[28] Rahul G Krishnan, Uri Shalit, and David Sontag. Deep kalman filters. *arXiv preprint arXiv:1511.05121*, 2015.

[29] Yingzhen Li and Stephan Mandt. Disentangled sequential autoencoder. In *International Conference on Machine Learning*, pages 5670–5679. PMLR, 2018.

[30] Yizhe Zhu, Martin Renqiang Min, Asim Kadav, and Hans Peter Graf. S3vae: Self-supervised sequential vae for representation disentanglement and data generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6538–6547, 2020.

[31] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

[32] Jakub Tomczak and Max Welling. Vae with a vampprior. In *International Conference on Artificial Intelligence and Statistics*, 2018.

[33] Alexander Alemi, Ben Poole, Ian Fischer, Joshua Dillon, Rif A Saurous, and Kevin Murphy. Fixing a broken elbo. In *ICLR*, 2018.

[34] Hiroshi Takahashi, Tomoharu Iwata, Yuki Yamanaka, Masanori Yamada, and Satoshi Yagi. Variational autoencoder with implicit optimal priors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5066–5073, 2019.

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

[36] Kei Akuzawa, Yusuke Iwasawa, and Yutaka Matsuo. Information theoretic regularization for learning global features by sequential {vae}, 2021. URL https://openreview.net/forum?id=zfO1MwBFu-.

[37] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *NIPS*, 2019.

[38] Werner Verhelst and Marc Roelands. An overlap-add technique based on waveform similarity (wsola) for high quality time-scale modification of speech. In *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 554–557. IEEE, 1993.

[39] Jonathan Driedger and Meinard Müller. A review of time-scale modification of music signals. *Applied Sciences*, 6(2):57, 2016.

[40] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

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

[42] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1526–1535, 2018.

[43] Wei-Ning Hsu, Yu Zhang, and James R Glass. Unsupervised learning of disentangled and interpretable representations from sequential data. In *NIPS*, 2017.

[44] Aapo Hyvarinen and Hiroshi Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ica. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/file/d305281faf947ca7acade9ad5c8c818c-Paper.pdf.

[45] Aapo Hyvarinen and Hiroshi Morioka. Nonlinear ICA of Temporally Dependent Stationary Sources. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 460–469, Fort Lauderdale, FL, USA, 20–22 Apr 2017. PMLR. URL http://proceedings.mlr.press/v54/hyvarinen17a.html.

[46] Luigi Gresele, Paul K. Rubenstein, Arash Mehrjou, Francesco Locatello, and Bernhard Schölkopf. The incomplete rosetta stone problem: Identifiability results for multi-view nonlinear ica, 2019.

[47] Scott E Reed, Yi Zhang, Yuting Zhang, and Honglak Lee. Deep visual analogy-making. *Advances in neural information processing systems*, 28:1252–1260, 2015.

[48] Niki Aifanti, Christos Papachristou, and Anastasios Delopoulos. The mug facial expression database. In *11th International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS 10*, pages 1–4. IEEE, 2010.

[49] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In *International Conference on Machine Learning*, pages 1174–1183. PMLR, 2018.

[50] J. Garofolo, Lori Lamel, W. Fisher, Jonathan Fiscus, D. Pallett, N. Dahlgren, and V. Zue. Timit acoustic-phonetic continuous speech corpus. *Linguistic Data Consortium*, 1992.

[51] Zhiming Zhou, Weinan Zhang, and Jun Wang. Inception score, label smoothing, gradient vanishing and-log (d (x)) alternative. *arXiv e-prints*, pages arXiv–1708, 2017.

[52] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798, 2010.

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]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (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? [Yes] See Appendix A and theorem 1.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix 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] We included the code. The dataset is larger than 100MB, but can be found on a third-party GitHub[1].
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Sec 4 and Appendix C.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (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]
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]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (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]

[1] https://github.com/YingzhenLi/Sprites