Video Reenactment as Inductive Bias for Content-Motion Disentanglement

Juan F. Hernández Albarracín and Adín Ramírez Rivera, *Senior Member, IEEE*

***Abstract*— Independent components within low-dimensional representations are essential inputs in several downstream tasks, and provide explanations over the observed data. Video-based disentangled factors of variation provide low-dimensional representations that can be identified and used to feed task-specific models. We introduce MTC-VAE, a self-supervised motion-transfer VAE model to disentangle motion and content from videos. Unlike previous work on video content-motion disentanglement, we adopt a chunk-wise modeling approach and take advantage of the motion information contained in spatiotemporal neighborhoods. Our model yields independent per-chunk representations that preserve temporal consistency. Hence, we reconstruct whole videos in a single forward-pass. We extend the ELBO's log-likelihood term and include a Blind Reenactment Loss as an inductive bias to leverage motion disentanglement, under the assumption that swapping motion features yields reenactment between two videos. We evaluate our model with recently-proposed disentanglement metrics and show that it outperforms a variety of methods for video motion-content disentanglement. Experiments on video reenactment show the effectiveness of our disentanglement in the input space where our model outperforms the baselines in reconstruction quality and motion alignment.**

***Index Terms*— Disentangled representations, video reenactment, variational inference, generative models, self-supervised learning.**

I. Introduction

WHILE the goal of representation learning is to obtain low-dimensional vectors useful for a diverse set of tasks, Disentangled Representation Learning (DRL) captures independent factors of variation within the observed data. These disentangled representations are robust and interpretable, simplify several downstream tasks like classification and Visual Question Answering [1], and support diverse content generation tasks [2], [3]. DRL shifted from unsupervised to weakly- and self-supervised methods, as inductive biases have shown to be fundamental in Deep Generative Models (DGM) [4], [5]. DRL methods from video separate *time independent* (a.k.a. content) from *dependent* (a.k.a. motion) factors of variation. While content features must be forced to have a low variance throughout the sequence, motion ones are expected to change.

Disentangling information from videos is of major importance since it can ease tasks that depend on the spatiotemporal data. For instance, prediction tasks could rely on the independent representations of the objects or only on their temporal information. These independence could not only ease the load on the downstream tasks but also enforce fairness and privacy over the data. DRL from videos has been approached as a sequential learning process forcing temporal consistency among frames. This problem is commonly addressed with Recurrent Neural Networks (RNN), due to their capacity of modeling temporal data of variable length. Although architectures based exclusively on 3D Convolutional Neural Networks (3D-CNN) have been used in general representation learning from videos for downstream tasks [6], [7], few works rely only on convolutional architectures for DRL and posterior video generation [8], [9], despite their capacity of modeling whole videos, as they are constrained to fixed-length sequences.

Taking into account the great suitability of Variational Autoencoders (VAE) for unsupervised tasks [10], [11], we propose a self-supervised DRL model that takes advantage of local spatio-temporal regularity to reconstruct videos by disentangling their content and motion while learning a robust representation space. Motion-Transfer Chunk Variational Autoencoder (MTC-VAE) is a Variational Autoencoder that models temporal segments (a.k.a. chunks) as independent random variables, maps them into a disentangled latent distribution, and maps them back consistently. When modeling chunks as independent, the reconstructed videos may not be temporally consistent. Hence, we preserve the temporal dependency that naturally exists among the chunks by assuming a Markovian relation between consecutive chunks at inference time. To enforce it, we incorporate two inductive biases in our model: (i) We assume content features as stationary and motion ones as non-stationary in our model's log-likelihood. (ii) Video Reenactment (VR) is equivalent to swapping the motion representation of two videos and mapping them to the

Manuscript received May 7, 2021; revised September 16, 2021, November 29, 2021, and January 6, 2022; accepted January 22, 2022. Date of publication March 2, 2022; date of current version March 14, 2022. The work of Juan F. Hernández Albarracín was supported in part by the São Paulo Research Foundation (FAPESP) under Grant 2017/16144-2; and in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior—Brazil (CAPES)—Finance Code 001. The work of Adín Ramírez Rivera was supported in part by the Brazilian National Council for Scientific and Technological Development (CNPq) under Grant 307425/2017-7; in part by FAPESP under Grant 2019/07257-3; and in part by CAPES—Finance Code 001. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Dong Xu. *(Corresponding author: Adín Ramírez Rivera.)*

Juan F. Hernández Albarracín is with the Institute of Computing, University of Campinas, Campinas 13083-970, Brazil (e-mail: juan.albarracin@ic.unicamp.br).

Adín Ramírez Rivera is with the Department of Informatics, University of Oslo, 0316 Oslo, Norway, and also with the Institute of Computing, University of Campinas, Campinas 13083-970, Brazil (e-mail: adinr@uio.no).

The source code is available at https://gitlab.com/mipl/mtc-vae.

This article has supplementary downloadable material available at https://doi.org/10.1109/TIP.2022.3153140, provided by the authors.

Digital Object Identifier 10.1109/TIP.2022.3153140

input space. We show that this duality (independence at generation time, and dependence at inference time) is successful at representing video sequences for both disentanglement and reconstruction.

Our contributions are: (i) A self-supervised DGM for VR and content-motion disentanglement from arbitrary-length videos through a simple 3D-CNN architecture in a single forward pass, improving over existing methods. (ii) Even assuming chunk independence, we significantly ease the disentangled motion-content feature inference and consistent video reconstruction, due to our inductive biases, and the self-supervised representation learning scheme. (iii) We show, that chunk-wise is better suited for DRL and video synthesis than frame-wise modeling for long videos. Moreover, we highlight that, unlike SotA VR models, MTC-VAE is suited to learn disentangled low-dimensional representations. VR models rely on entangled high-dimensional features and bypass information through the architecture to achieve better reconstruction at the cost of bloated features. In contrast, our objective is to obtain independent factors of variation that are expressive enough for simple generators to create natural videos.

II. Related Work

A. General Disentangled Representation Learning

Seminal works on DRL are mostly unsupervised, and the majority rely on VAEs. InfoGAN [12], however, is the most relevant exception. It uses control variables (categorical, discrete, or continuous) in the latent representation as inductive biases while penalizing mutual information among the latent units in an adversarial framework. β-VAE [13] includes the β hyper-parameter into the VAE's ELBO to leverage independence among the latent scalars, leading to a higher-quality disentanglement. Later approaches (e.g., β-TCVAE [14] and FactorVAE [15]) penalize Total Correlation among the latent scalars, yielding a better trade-off between disentanglement and reconstruction quality. The ground-breaking work by [4] showed that unsupervised methods for DRL are extremely weak. Posterior works have shifted to weakly- and self-supervised approaches. Hence, our proposed MTC-VAE introduces inductive biases in the latent space, such as explicit latent factors to represent content and motion features, with sufficient encoded information to guarantee VR from them.

B. Disentangled Representations From Video

These works focus on disentangling time-dependent from time-independent features for each frame of the video and then enforcing inter-frame consistency. Common setups of these approaches perform pose-content disentanglement while achieving consistency using RNNs and GANs [16]–[19]. Instead of pose-content disentanglement, some works separate deterministic from stochastic features [20], [21]. Most of the works in this area are applied to video prediction, but recent ones have started to be tested on VR tasks [8], [9], [22], [23]. Few of them [8], [9] rely on 3D-convolutional generators, but are constrained to fixed-length videos. The rest use RNNs to capture the temporal relation between frames or segments at generation time, to perform either video reconstruction, prediction, or sequence-to-sequence translation. Although MTC-VAE models dependent chunks at inference time, it assumes independence at generation time. These assumptions simplify the tasks of reconstruction and VR since, to reconstruct a chunk of a video, it does not need to reconstruct the previous ones. Therefore, the chunkwise approach takes the best of both worlds at not being constrained either to fixed-length-sequences or sequential generation.

C. Video Reenactment

Recent methods on VR work in the domain of human faces [24]–[27], human poses [28]–[31], or objects in general [32]–[36]. Their main objective is to generate realistic videos, while the representation is either irrelevant or a secondary objective. Instead, DRL models hold this objective as primary. Most of these methods rely on warping techniques assisted by spatial transformer networks [37] for frame-wise conditional video generation. To apply such transformations, the generator requires high-dimensional spatial information that would normally be lost in a low-dimensional latent representation. Hence, they either map to latent spaces that are larger than the original input space, to preserve spatial information, or bypass this information through skip connections from the encoder to the decoder. Thus, a low-dimensional latent representation is not enough to represent the whole video. In contrast, our proposal reconstructs videos while learning low-dimensional and factorized representations. We highlight that our method reconstructs videos exclusively from low-dimensional representations. Due to this restriction, we expect the perceptual quality and motion complexity of rendered videos to be higher in VR methods in comparison to DRL ones. Despite this limitation, we consider our work as a step towards bridging these two areas.

III. Proposed Approach: MTC-VAE

Given that content changes at a much slower rate than motion in a video, we propose to extract disentangled representations from local spatiotemporal neighborhoods (a.k.a. chunks). Content information of neighboring chunks changes so slowly that we may assume that it remains constant throughout a scene, while motion presents rapid changes. Unlike existing frame-wise approaches, we use chunks to better capture the temporal characteristics of the video (cf. Section IV-F for the impact of the temporal windows), and their relations to obtain a self-supervised learning signal.

MTC-VAE contains only 3D-convolutional streams and, unlike recurrent approaches, models chunks as independent random variables for the generative pass, yet Markovian-dependent for the inference one. Our formulation starts diverging from a standard two-latent-priors VAE when we extend our $\log p(x)$ to leverage inter-chunk consistency, which helps to reconstruct realistic videos, even though chunks are independently generated. We go further and introduce the self-supervised *blind reenactment loss* (BRL): another inductive bias that blindly simulates VR between two videos.

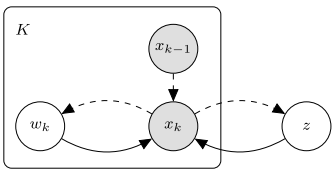

Fig. 1. In the generative model (solid arrows), K chunks $\{x_k\}$ (observed) share the same content z, while having their own motion w_k. During inference (dashed arrows), the latent variables z and w_k are inferred from each chunk, while each chunk x_k also depends on the previous one.

A. Chunk-Wise Video Modeling

We represent the video $x = (x_k)_{k=1}^K$ as a sequence of K non-overlapping and equally-sized chunks x_k of length c.[1] Similarly, we define $w = (w_k)_{k=1}^K$ as the sequence of motion representations of each x_k. For the k-th chunk, we model the content and motion as independent latent variables z and w_k, respectively. We assume z to be unique and shared across the chunks, as content remains constant through time. Fig. 1 depicts the graphical model for a video x.

Different from common frame-wise approaches, where w normally depend on previous frames, in the generative phase, we model all the motion representations $\{w_k\}$ as independent random variables. This assumption simplifies the generation process since it lets us generate a particular chunk without having to consider the previous ones in the video. A unique z for all the chunks sets an implicit dependence of each chunk to the whole video in the inference phase of the model.

Being the chunks independent, the joint probability of the model is the product of the conditionals of each chunk and their latent variables, i.e.,

$$p(x, w, z) = p(z) \prod_k p(x_k|w_k, z) p(w_k). \tag{1}$$

We model the generative process of a single chunk through a VAE [38], with content encoder $q_\phi(z|x_k)$, motion encoder $q_\gamma(w_k|x_k)$, and decoder $p_\theta(x_k|w_k, z)$ with parameters (ϕ, γ, θ), updated to maximize of the evidence lower bound (ELBO) of the expected log-likelihood

$$\arg\max_{\phi,\gamma,\theta} \mathop{\mathbb{E}}_{\tilde{q}(x_{1:k})} \sum_k \Big\{ \mathop{\mathbb{E}}_{q_\phi} \mathop{\mathbb{E}}_{q_\gamma} \big[\log p_\theta(x_k|w_k, z)\big] - \mathrm{KL}\; q_\gamma(w_k|x_k) p(w_k) - \mathrm{KL}\; q_\phi(z|x_k) p(z) \Big\}. \tag{2}$$

Fig. 2 shows the pipeline to calculate the ELBO (2). We maximize the expected reconstruction loss over the two latent variables w.r.t. their distributions $q_\phi(z|x_k)$ and $q_\gamma(w_k|x_k)$ (first term), and minimize the Kullback-Leibler divergence between these distributions w.r.t. their priors. We compute their expected value w.r.t. the empirical distribution of the chunks $\tilde{q}(x_{1:k}) = \prod_k q(x_k|x_{k-1})$ that models a Markovian temporal relation between them.[2] We approximate the chunk distribution through a sampling process on the videos, and

[1]For brevity, we assume that c divides the length of the video. However, we can model arbitrary-length videos by padding incomplete chunks to match c.

[2]We assume the first chunk to be distributed through $q(x_1|x_0) \equiv q(x_1)$ to simplify the notation.

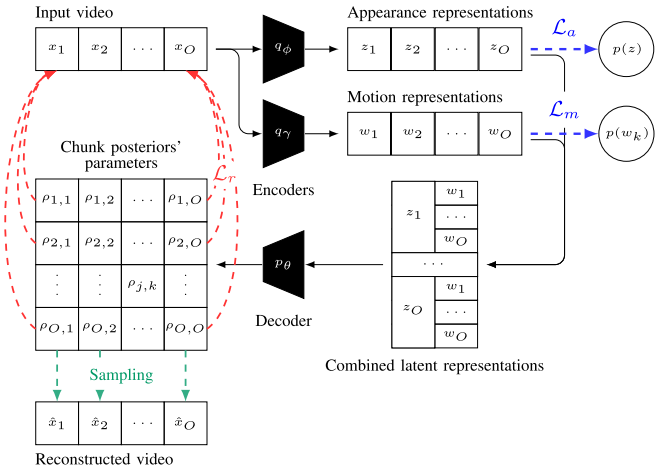

Fig. 2. We feed consecutive chunks $\{x_k\}_{k=1}^O$ to the encoders q_ϕ and q_γ, yielding their representations, $\{w_k\}_{k=1}^O$ and $\{z_j\}_{j=1}^O$. We concatenate all combinations of z_j's and w_k's, and decode them to obtain the p.d.f. parameters $\rho_{j,k}$ for the k-th chunk posteriors $p_\theta(x_k|w_k, z_j)$. Every posterior from w_k must generate x_k. We maximize the log-likelihood of each chunk under the corresponding set of posteriors. Chunk posteriors relate with the original chunks through $\mathcal{L}_r$. The latent prior distributions relate through $\mathcal{L}_a$ and $\mathcal{L}_m$. We sample from the chunk posterior by applying the Sigmoid function to the output of the decoder.

model all prior distributions as standard Gaussians. To generate a new video from the chunk posterior, we concatenate the expected values of the chunk posteriors, directly provided by the decoder. See Appendix A for further detail and proof of our formulation.

Our architecture consists of two encoders $q_\phi(z|x_k)$ and $q_\gamma(w_k|x_k)$, and one decoder $p_\theta(x_k)$. All of them have five 3D-convolutional layers, with Batchnorm and ReLU activations. The number of filters in the hidden layers of the decoder is double the number of filters in the encoders.

B. Inter-Chunk Consistency

As shown in Equation 2, we can train a VAE to independently reconstruct chunks. However, the independence assumption at generation time may cause the videos to not be smoothly rendered between chunks. To solve this issue, we force our model to yield a unique content representation z, regardless of the chunk from which it is inferred.

We part from the assumption that content is constant throughout the video, and so does its latent representation $z \sim q_\phi(z|x_k)$—cf. Section III-A. To force our model to learn this constraint, we train it to maximize $\log p_\theta(x_k|w_k, z_j)$ for every j, i.e., maximize the log-likelihood of a chunk x_k given its own motion w_k and any z_j content representation—cf. Fig. 2. We extend the $\log p(x_k|w_k, z)$ term (2) to fulfill this constraint. So our final reconstruction loss is

$$\mathcal{L}_r(\theta, \phi, \gamma) = \sum_{k=1}^O \sum_{j=1}^O \big[\log p_\theta(x_k|w_k, z_j)\big], \tag{3}$$

where $z_j \sim q_\phi(z|x_j)$, $w_k \sim q_\gamma(w_k|x_k)$, and O is defined as the *order of the model* that restricts the number of chunks used to calculate the loss. As Fig. 2 shows, the decoder outputs the distribution parameters $\rho_{j,k}$ of each chunk likelihood

$p_\theta(x_k|w_k, z_j)$, used in $\mathcal{L}_r$. Due to its combinatory nature, it is impractical to apply $\mathcal{L}_r$ to all the chunks. Hence, for each forward pass, we consider only a sequence of $O \leq K$ consecutive chunks of x, starting at a random frame.

The second and third terms of the expected log-likelihood (2) correspond to the regularization terms of the motion and content distributions, respectively. That is, we compute

$$\mathcal{L}_m(\gamma) = -\sum_{k=1}^{O} \mathrm{KL}\ q_\gamma(w_k|x_k)p(w_k), \text{ and} \tag{4}$$

$$\mathcal{L}_a(\phi) = -\sum_{k=1}^{O} \mathrm{KL}\ q_\phi(z|x_k)p(z), \tag{5}$$

on O consecutive chunks instead of the whole video—c.f. Fig. 2.

Unlike other variational inference methods of grouped observations [39]–[42], we opted for the extended log-probability term (3), considering different combinations of appearance features, to yield stronger gradients for chunk-consistency, instead of averaging the shared representations in the group.

C. Blind Reenactment Loss

Our proposed Blind Reenactment Loss (BRL) loss increases the likelihood $\log p(x_k|w_k, z)$ of our ELBO given any encoded chunks. It aims at leveraging content-motion disentanglement by doing VR between a source video S and a driving video D. The motion representation of S is replaced by the one of D, to reconstruct a reenacted video with the object of interest from S moving like the one in D. This translation can be achieved uniquely if the content and motion representations of both videos are disentangled. The main difficulty is that, in principle, we would need to train our model with ground-truth reenacted videos. However, we opt for self-supervised training and take advantage of our chunk-based approach.

Consider two chunks s_i and s_j from S, and one chunk d_l from D. Assuming constant content throughout the video, if we independently reenact s_i and s_j w.r.t. d_l, the two reconstructed chunks must be the same since s_i and s_j have the same content. To achieve this objective, we force the corresponding chunk posteriors $p(x_k|w_k, z)$ to be equivalent, i.e., $p(x|w_l^d, z_j^s) \equiv p(x|w_l^d, z_i^s)$, where $z_i^s \sim q(z|s_i)$, $z_j^s \sim q(z|s_j)$, and $w_l^d \sim q(w_k|d_l)$, by minimizing the KL divergence between every two posteriors that fit the described case. Let

$$\mathcal{L}_b(\theta, \phi, \gamma) = -\sum_{l=1}^{O}\sum_{j=1}^{O}\sum_{i=1}^{O} \mathrm{SKL}\left(p_\theta(x|w_l^d, z_j^s) \,\|\, p_\theta(x|w_l^d, z_i^s)\right). \tag{6}$$

be our BRL, where $\mathrm{SkL}\ PQ = \frac{1}{2}(\mathrm{KL}PQ + \mathrm{KL}QP)$ is a symmetrical operator. This loss involves two empirical distributions of unobservable samples, so we are not aware, at training time, of whether the sampled videos are correctly reenacted. If there is disentanglement, posteriors sharing the same motion of D and *any* content of S must be equivalent, regardless of their samples.

The BRL must be optimized along with $\mathcal{L}_r$ (3) to prevent posterior collapse. Notice that, if $O = 1$, then $j = i = 1$ and $\mathcal{L}_b = 0$, so this objective can only be optimized for $O \geq 2$.

D. General Loss Function

We define the general objective to be maximized as

$$\mathcal{L} = \mathcal{L}_r + \lambda\mathcal{L}_b + \beta(\mathcal{L}_a + \mathcal{L}_m), \tag{7}$$

where β comes from β-VAE by [13], and λ weights $\mathcal{L}_b$. Each element in the batch is conformed by a sequence of O chunks, so $\mathcal{L}$ can be calculated independently for every element.

IV. Experiments

We evaluated MTC-VAE in DRL, VR, and downstream tasks. Although MTC-VAE does not require labels in training time, we used labels to asses disentanglement, and to split the training and testing datasets. We detail the implementation of the model and the experimental setup in Appendix B.

A. Datasets

(i) Cohn-Kanade (CK+) facial dataset [43], [44], (ii) Liberated Pixel Cup (LPC) sprites, (iii) Moving MNIST (MMNIST) [45], (iv) Deepmind's dSprites, (v) Deepmind's 3dShapes, and (vi) Multimedia Understanding Group (MUG) facial dataset [46]. We generated videos from the images of dSprites and 3dShapes, forming sequences of objects moving in linear and curved trajectories, or changing their perspective. Each dataset contains 10000 videos, except for CK+ (320), LPC (200000), and MUG (700). We report the average model performance in a 5-fold cross-validation setup (80% for training and 20% for testing). Appendix B-C provides further detail about the datasets, as well as the factors of variation.

B. Baselines

We compared our method against the Disentangled Sequential Autoencoder (dis-VAE) [22], SVG-LP [20], and β-TCVAE [14]. The first two are frame-wise approaches that disentangle time-dependent from time-independent factors. Although SVG-LP namely disentangles deterministic from stochastic features, they force the deterministic features to remain constant, while the stochastic ones change from frame to frame, like a content-motion modeling. β-TCVAE is an unsupervised disentanglement model, tested so far on images, so we extended it to 3D-CNNs to support chunks.

C. Hyper-Parameters

After a hyper-parameter search in the models (see details in Appendix B), we tuned the β parameter and the latent space size. For dSprites, LPC and MMNIST, $\beta = 1$, and $\beta = 5$ for the other datasets. Regarding the latent space dimensionality (where each dimension is a *latent unit*), $\dim(z) = 14$, $\dim(w_k) = 7$ for CK+, LPC, and MUG, $\dim(z) = 12$, $\dim(w_k) = 6$ for 3dShapes, $\dim(z) = 12$, $\dim(w_k) = 4$ for dSprites, and $\dim(z) = 8$, $\dim(w_k) = 4$ for MMNIST. We performed ablation studies on λ, c, O, and β (c.f. Section IV-C and Appendix F).

TABLE I
PERFORMANCE FOR CONTENT-MOTION DISENTANGLEMENT AND DATA REALISM. (* $c = 1$)

	FVAE ↑	MIG ↑	SAP ↑	SSIM ↑	FID ↓	FVAE ↑	MIG ↑	SAP ↑	SSIM ↑	FID ↓
	3dShapes					LPC				
β-TCVAE	0.50 ± 0.02	$\mathbf{0.01 \pm 0.01}$	0.11 ± 0.08	0.53 ± 0.10	140.25 ± 51.13	0.81 ± 0.03	0.02 ± 0.02	0.00 ± 0.00	0.64 ± 0.01	80.27 ± 3.17
dis-VAE	0.50 ± 0.00	0.00 ± 0.00	0.08 ± 0.06	0.40 ± 0.03	$\mathbf{71.24 \pm 12.35}$	0.92 ± 0.02	0.02 ± 0.02	0.01 ± 0.01	0.78 ± 0.01	71.70 ± 1.76
SVG-LP	0.50 ± 0.00	$\mathbf{0.01 \pm 0.00}$	0.03 ± 0.02	0.54 ± 0.05	136.00 ± 81.12	0.63 ± 0.02	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.79 \pm 0.02}$	62.75 ± 9.87
MTC-VAE	0.50 ± 0.02	$\mathbf{0.01 \pm 0.00}$	$\mathbf{0.41 \pm 0.14}$	0.67 ± 0.06	119.47 ± 51.00	$\mathbf{0.93 \pm 0.06}$	$\mathbf{0.11 \pm 0.11}$	$\mathbf{0.60 \pm 0.40}$	0.67 ± 0.01	$\mathbf{41.72 \pm 3.31}$
MTC-VAE*	0.50 ± 0.01	$\mathbf{0.01 \pm 0.00}$	0.39 ± 0.11	$\mathbf{0.73 \pm 0.02}$	100.80 ± 46.82	0.86 ± 0.01	0.00 ± 0.00	0.11 ± 0.03	0.67 ± 0.01	42.59 ± 4.09
	CK+					MMNIST				
β-TCVAE	0.79 ± 0.05	$\mathbf{0.03 \pm 0.02}$	0.06 ± 0.04	0.50 ± 0.07	116.74 ± 24.70	0.66 ± 0.07	0.04 ± 0.04	0.04 ± 0.04	$\mathbf{0.71 \pm 0.03}$	152.56 ± 16.93
dis-VAE	0.71 ± 0.02	0.01 ± 0.01	0.04 ± 0.02	0.61 ± 0.05	71.48 ± 3.09	0.64 ± 0.04	0.02 ± 0.02	0.03 ± 0.02	0.70 ± 0.02	149.43 ± 9.73
SVG-LP	0.70 ± 0.06	0.02 ± 0.01	0.04 ± 0.02	0.02 ± 0.00	$\mathbf{38.79 \pm 17.63}$	0.52 ± 0.01	0.01 ± 0.00	0.02 ± 0.02	0.58 ± 0.02	179.08 ± 50.49
MTC-VAE	$\mathbf{0.86 \pm 0.04}$	0.02 ± 0.01	$\mathbf{0.13 \pm 0.05}$	0.66 ± 0.12	63.13 ± 22.50	$\mathbf{0.95 \pm 0.04}$	$\mathbf{0.11 \pm 0.07}$	$\mathbf{0.10 \pm 0.05}$	0.68 ± 0.01	$\mathbf{102.11 \pm 0.99}$
MTC-VAE*	0.85 ± 0.02	$\mathbf{0.03 \pm 0.01}$	0.05 ± 0.02	$\mathbf{0.68 \pm 0.13}$	76.16 ± 19.37	0.91 ± 0.04	0.09 ± 0.05	0.09 ± 0.04	0.69 ± 0.01	186.25 ± 23.55
	dSprites					MUG				
β-TCVAE	0.57 ± 0.03	0.00 ± 0.00	0.04 ± 0.01	$\mathbf{0.79 \pm 0.03}$	79.34 ± 6.18	0.74 ± 0.05	$\mathbf{0.05 \pm 0.04}$	0.23 ± 0.03	0.51 ± 0.01	44.78 ± 3.32
dis-VAE	0.70 ± 0.02	0.01 ± 0.00	0.01 ± 0.00	$\mathbf{0.79 \pm 0.00}$	97.07 ± 1.73	$\mathbf{0.76 \pm 0.03}$	0.01 ± 0.01	0.11 ± 0.03	$\mathbf{0.78 \pm 0.01}$	62.84 ± 3.45
SVG-LP	0.61 ± 0.05	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.79 \pm 0.02}$	98.14 ± 9.20	0.64 ± 0.04	0.02 ± 0.01	0.38 ± 0.04	0.50 ± 0.15	101.59 ± 37.00
MTC-VAE	0.91 ± 0.02	$\mathbf{0.04 \pm 0.01}$	$\mathbf{0.10 \pm 0.01}$	0.78 ± 0.00	$\mathbf{57.18 \pm 6.43}$	0.72 ± 0.04	0.01 ± 0.01	0.73 ± 0.05	0.63 ± 0.02	$\mathbf{28.79 \pm 1.15}$
MTC-VAE*	$\mathbf{0.92 \pm 0.01}$	0.02 ± 0.02	0.01 ± 0.00	0.77 ± 0.01	105.79 ± 5.86	0.70 ± 0.09	0.04 ± 0.02	$\mathbf{0.76 \pm 0.10}$	0.66 ± 0.06	43.86 ± 13.15

D. Content-Motion Disentanglement

We obtained the latent representations from the trained models for the test set and, using ground-truth labels, we calculated the Mutual Information Gap (MIG) [14], the Factor-VAE (FVAE) disentanglement metric [15], and the Separated Attribute Predictability Score (SAP) [47].

Assessing disentanglement quality is narrowly application-related [48], [49]. We adhere to the criteria defined by [49], by which we may evaluate disentanglement based on either *modularity* (i.e., each unit contains information of at most one factor), *compactness* (i.e., each factor is ideally encoded by at most one unit) or *explicitness* (i.e., each factor is easily recovered from its code).

Since our objective is to encode two factors of variation (content and motion) in various latent units, our main interest is modularity. Compactness, although desirable, is expected to not be fulfilled, as content and motion are complex factors that can barely be represented in few latent units. Explicitness is important to estimate the effectiveness of disentangled representations for downstream tasks, like classification.

MIG and SAP heavily penalize representations that are not compact, by depending on the mean difference between the first and second most predictive/informative units. Hence, FVAE is the metric that interests us the most, as it measures both modularity and explicitness. We report results on MIG and SAP for completeness since, besides assessing compactness, to some extent, MIG also assesses modularity, and SAP, explicitness.

For β-TCVAE and MTC-VAE, we split every test video into chunks and calculated one latent vector per chunk. For dis-VAE and SVG-LP, we obtained one vector per frame. We aggregated the multiple factors, provided in 3dShapes, dSprites, and LPC, into two categories: time-dependent and time-independent, yielding two factors, to reduce the risk of over-estimation of disentanglement performance, due to pairs of disentangled factors while the rest are entangled.

Table I shows the performance of the models on the content-motion disentanglement. We included the results obtained for the frame-wise version of MTC-VAE (i.e., $c = 1$) to compare against dis-VAE and SVG-LP. Both the chunk and frame versions of MTC-VAE are the ones with the best disentanglement performance, followed by β-TCVAE and dis-VAE. It is remarkable that SVG-LP uses skip connections from the encoder to the decoder, so most of the appearance information is not in the latent representation. This is reflected in the fact that it attained the poorest performance. In general, the chunk version of MTC-VAE outperforms the frame version.

Although MTC-VAE is trained for motion-content disentanglement, we can argue that this task can be used as a step towards multi-factor disentanglement. To show our point, we calculated MIG, FVAE, and SAP considering all the factors of variation provided in the datasets' metadata. Table II shows the results for 3dShapes, dSprites, and LPC since the others only provide motion-content labels. In all cases, MTC-VAE (both frame and chunk versions) significantly outperforms the baselines. The second best method was β-TCVAE, which is expected since it has been already tested on multi-factor disentanglement for images. Table II demonstrates that multi-factor disentanglement is a significantly harder task, but it is remarkable that MTC-VAE features are more disentangled than the others, even when the model was not trained for this specific task. We provide a list and a description of the factors of variation considered for each dataset in Appendix B.

E. Video Reenactment

We generated 10000 videos, each one from a source video S and driving video D. For β-TCVAE and MTC-VAE, we fixed the content representation of the first chunk of S, replicated it, and concatenated each replica to the motion representation of each chunk in D. Due to the assumption of appearance preservation throughout the video, our model must be able to reconstruct the video from the appearance representation of any of their chunks. We decided to use the first chunk of each video for easinesses in the implementation. The reenacted video was obtained by decoding the resulting vectors. For dis-VAE, we obtained the content representation from the

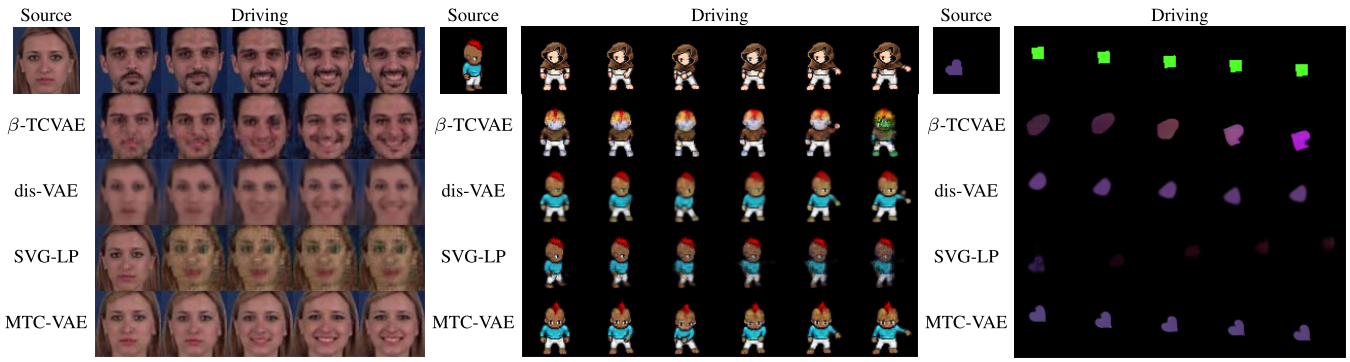

Fig. 3. Reenactment results. Each set shows the reenacted video of each method with the appearance of *source* and the motion of *driving*.

TABLE II
MULTI-FACTOR DISENTANGLEMENT (* $c = 1$)

		FVAE ↑	MIG ↑	SAP ↑
3dShapes	β-TCVAE	0.21 ± 0.03	0.07 ± 0.04	0.03 ± 0.02
	dis-VAE	0.19 ± 0.01	0.03 ± 0.01	0.01 ± 0.01
	SVG-LP	0.18 ± 0.01	0.02 ± 0.01	0.01 ± 0.00
	MTC-VAE	0.27 ± 0.05	$\mathbf{0.19 \pm 0.07}$	$\mathbf{0.08 \pm 0.03}$
	MTC-VAE*	$\mathbf{0.31 \pm 0.02}$	0.14 ± 0.05	0.05 ± 0.02
dSprites	β-TCVAE	0.28 ± 0.01	0.02 ± 0.01	0.01 ± 0.00
	dis-VAE	0.28 ± 0.00	0.02 ± 0.01	0.01 ± 0.00
	SVG-LP	0.29 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
	MTC-VAE	$\mathbf{0.33 \pm 0.02}$	$\mathbf{0.11 \pm 0.01}$	$\mathbf{0.02 \pm 0.00}$
	MTC-VAE*	0.29 ± 0.01	0.07 ± 0.02	0.01 ± 0.00
LPC	β-TCVAE	0.32 ± 0.07	0.16 ± 0.09	0.03 ± 0.01
	dis-VAE	0.22 ± 0.01	0.04 ± 0.01	0.06 ± 0.05
	SVG-LP	0.17 ± 0.00	0.01 ± 0.00	0.01 ± 0.01
	MTC-VAE	0.41 ± 0.07	$\mathbf{0.21 \pm 0.05}$	$\mathbf{0.89 \pm 0.01}$
	MTC-VAE*	$\mathbf{0.43 \pm 0.06}$	0.18 ± 0.03	$\mathbf{0.89 \pm 0.00}$

mean of the frames' appearances and sequentially calculated the motion representations. For SVG-LP, we obtained the representation from the inference model of the first frame of S and concatenated it with each representation yielded by the learned prior on each frame of D. For β-TCVAE, since we do not know which units correspond to content and which ones to motion, we considered the classification scheme used to calculate the FVAE metric, which returns an estimate of the units that are more likely to represent either content and motion. Based on these criteria, we swapped the units that are more likely to represent motion from D to S.

Our metrics are frame-wise Structural Similarity (SSIM) [50] to quantify identity preservation after reenactment (i.e., whether the reenacted video contains the content of S and no leaked content of D), and frame-wise Fréchet Inception Distance (FID) [51] to assess the realism of the reenacted videos. Table I shows the performance of the models for SSIM and FID. In half of the cases, MTC-VAE outperforms the baselines, but its superiority is not as significant as it is in disentanglement.

Due to the lack of metrics to assess that the reenacted video mimics D, we provide a qualitative assessment between videos reenacted by the models and their corresponding source videos. Fig. 3 shows some examples. It can be seen that MTC-VAE yields reenacted videos that are better synchronized w.r.t. D than the baselines. Also, in terms of sharpness, identity preservation, and inter-chunk consistency, MTC-VAE shows a clear advantage. In general, dis-VAE was more successful in representing time-dependent features than β-TCVAE. Qualitatively, SVG-LP yielded the poorest reenactment.

Additional results are in Appendix G. We explored the limits of our model on high-resolution videos (Appendix D) and on a real-world human-action dataset (Appendix E). Although it has shown to be robust in high-resolution videos, our experiments on human-action datasets make evident the fact that exclusively-CNN-based architectures fall short in reconstructing large motions [52], [53], like the ones done by the human body. We show that the yielded representations are successful in capturing the semantics of the content and motion of the videos, which suggests that our model obtains meaningful representations of any kind of data. However, its effectiveness for reconstruction and reenactment is restricted to motions with fewer degrees of freedom (like simple trajectories, facial expressions, and a reduced set of human actions). These experiments reveal that the bottleneck of the model is the decoder.

F. Ablation Studies

We conducted ablation studies to determine the impact of the chunk size (c), the order of the model (O), the hyperparameter β, and the presence/absence of the Blind Reenactment Loss (λ). Figs. 4 and 5 show, respectively, charts on the ablative study on $c \in \{1, 3, 5, 7, 9\}$ and $\lambda \in \{0, 1\}$. In Appendix F, we present complete examples with all the cases on the ablation study, tables with the detailed scores, and the ablation on O.

In Fig. 4, we plotted the curves of the metrics as a function of c. Most of them peaked in 3 or 5 for FVAE and SAP, meaning that middle-sized chunks are preferable. For SSIM, when $c > 5$, there is a slight decrease on performance and, although for $c \leq 5$ performance is similar, it reaches is lowest variability at $c = 5$ (c.f. gray curve). FID shows a heterogeneous behavior among the datasets. For CK+ and LPC, the greater the chunk size, the better the performance while the opposite stands for 3dShapes. For MMNIST, middle values attain the best performance, while LPC shows its worst performance at the same values. Table F.I presents more detailed results.

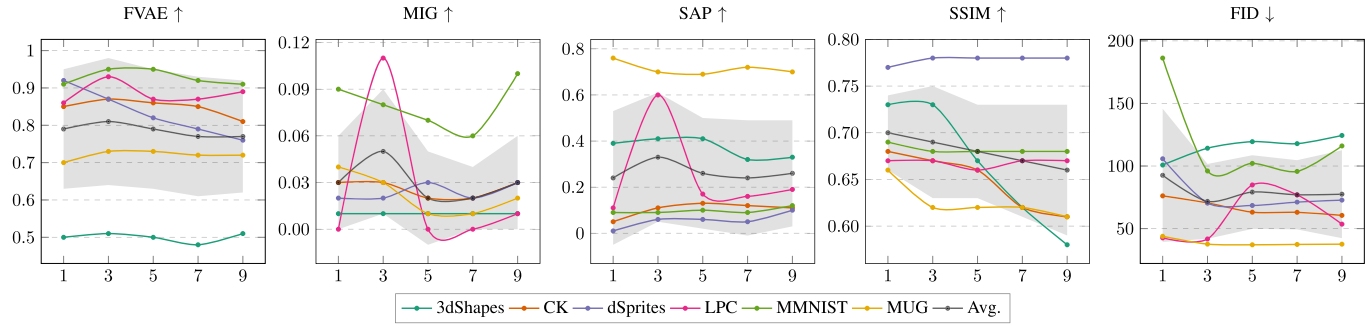

Fig. 4. Study of the chunk size vs. several metrics. The gray area shows one standard deviation away from the average plot.

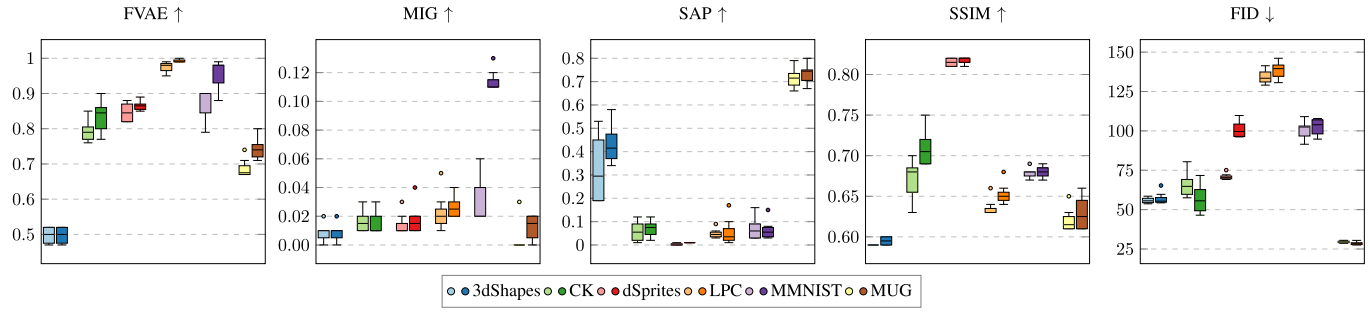

Fig. 5. Ablation on BRL. Light colors indicate the absence of $\mathcal{L}_b$ ($\lambda = 0$), while dark colors indicate its presence ($\lambda = 1$).

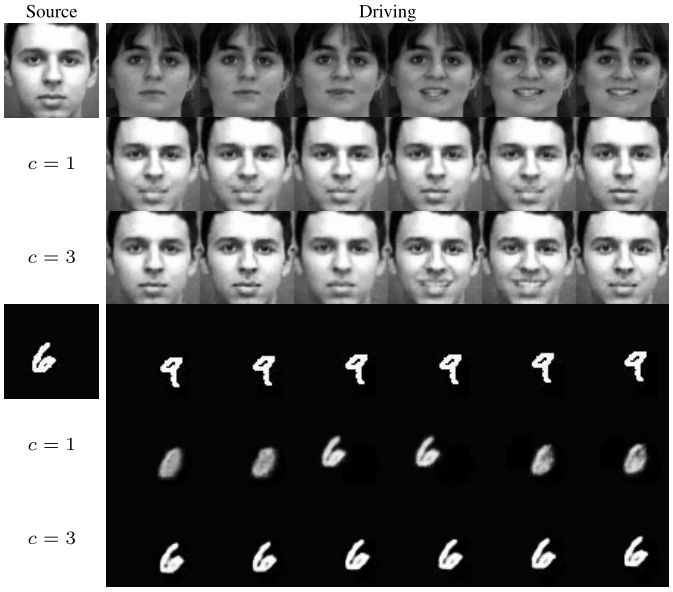

Fig. 6. Qualitative comparison of performances for the frame version of MTC-VAE ($c = 1$) and the chunk version with a temporal neighborhood of $c = 3$ in reenactment quality and inter-chunk consistency.

Although there is a pattern in most of the metrics pointing to a better performance with middle-sized chunks, numerically, the impact on the chunk size may be little significant for the metrics considered. A more explicit impact on the performance of using chunks ($c > 1$) instead of frames ($c = 1$) is qualitatively evidenced in both reenactment quality and inter-chunk consistency. As we do not count on metrics to quantify such properties, we depict in Fig. 6 the perceptual difference of performance between the frame and the chunk version of MTC-VAE. Both CK+ and MMNIST show poor reenactment performance for $c = 1$. This suggests that wider temporal neighborhoods eases motion encoding, to be transferred between videos more accurately, as well as it also eases smoothness. We show a thorough comparison in Appendix G.

Fig. 5 shows the impact of BRL on the performance metrics. The boxes correspond to the distribution of the five experiments associated with each configuration, due to the 5-fold cross-validation scheme. Boxes with light colors indicate the performance when $\lambda = 0$, and the ones with dark colors when $\lambda = 1$. Regarding disentanglement, it can be seen that the positive impact of the BRL is significant in general for FVAE, except for the 3dShapes datasets. For MIG and SAP, the impact is not that significant, however, this is expected, since both metrics measure compactness, and the BRL loss is not designed for this objective. Regarding reconstruction metrics (SSIM and FID), its impact was not significant and, in the case of FID, it showed to decrease the performance in dSprites, LPC and MMNIST. Regarding the order of the model, we concluded that optimal values of O are 2 or 3, depending on the length of the videos in the dataset (c.f. Appendix F). Since the complexity of the model is quadratic w.r.t. to O, higher values are not worth considering.

G. Performance on Downstream Tasks

To evaluate the robustness of the learned disentangled representations, we extracted them from the datasets, and trained a Linear Support Vector Machine to assess whether they are linearly separable. We chose a simple classifier, as more sophisticated ones are prone to work around weaker representations, hindering the comparison between our model and the baselines. We tested the models in (i) content-motion and (ii) multi-factor classification.

(a) Full latent space **(b)** Content subspace **(c)** Motion subspace

Fig. 7. Latent-space traversals on LPC. The upper and lower sequences are, respectively, the start and endpoints of the traversals.

TABLE III
CONTENT (C)/MOTION (M) CLASSIFICATION ACCURACY ($*\,c = 1$)

	3dShapes		CK+		dSprites	
	C	M	C	M	C	M
β-TCVAE	0.53 ± 0.05	0.44 ± 0.05	0.90 ± 0.02	0.52 ± 0.05	0.22 ± 0.01	0.62 ± 0.02
dis-VAE	0.48 ± 0.01	0.42 ± 0.01	$\mathbf{1.00 \pm 0.00}$	0.62 ± 0.04	0.54 ± 0.06	0.59 ± 0.01
SVG-LP	0.11 ± 0.02	0.11 ± 0.01	0.87 ± 0.04	0.60 ± 0.02	0.00 ± 0.00	0.60 ± 0.01
MTC-VAE	$\mathbf{0.95 \pm 0.01}$	$\mathbf{0.59 \pm 0.01}$	0.97 ± 0.01	$\mathbf{0.68 \pm 0.07}$	$\mathbf{0.61 \pm 0.01}$	$\mathbf{0.63 \pm 0.03}$
MTC-VAE*	0.46 ± 0.01	0.41 ± 0.02	0.94 ± 0.01	0.63 ± 0.08	0.20 ± 0.08	0.60 ± 0.03
	LPC		MMNIST		MUG	
β-TCVAE	0.11 ± 0.01	$\mathbf{0.99 \pm 0.01}$	0.32 ± 0.03	$\mathbf{0.26 \pm 0.05}$	$\mathbf{1.00 \pm 0.00}$	0.54 ± 0.06
dis-VAE	0.43 ± 0.02	0.95 ± 0.01	$\mathbf{0.54 \pm 0.08}$	0.16 ± 0.01	$\mathbf{1.00 \pm 0.00}$	0.55 ± 0.02
SVG-LP	0.00 ± 0.00	0.54 ± 0.03	0.14 ± 0.01	0.14 ± 0.01	0.48 ± 0.04	0.34 ± 0.01
MTC-VAE	0.65 ± 0.01	0.93 ± 0.04	0.48 ± 0.10	0.20 ± 0.06	$\mathbf{1.00 \pm 0.00}$	$\mathbf{0.79 \pm 0.05}$
MTC-VAE*	$\mathbf{0.68 \pm 0.03}$	0.97 ± 0.01	0.45 ± 0.07	0.19 ± 0.02	$\mathbf{1.00 \pm 0.00}$	0.70 ± 0.08

TABLE IV
CLASSIFICATION ACCURACY IN MULTIPLE FACTORS ($*\,c = 1$)

Factor	β-TCVAE	dis-VAE	SVG-LP	MTC-VAE	MTC-VAE*
	3dShapes				
Floor hue	0.94 ± 0.04	$\mathbf{1.00 \pm 0.00}$	0.27 ± 0.05	0.99 ± 0.01	0.99 ± 0.01
Wall hue	0.95 ± 0.07	$\mathbf{1.00 \pm 0.00}$	0.53 ± 0.11	0.97 ± 0.03	0.97 ± 0.03
Obj. hue	0.82 ± 0.10	$\mathbf{1.00 \pm 0.00}$	0.17 ± 0.02	0.95 ± 0.04	0.95 ± 0.04
Init. size	0.66 ± 0.25	0.97 ± 0.04	0.66 ± 0.18	$\mathbf{0.98 \pm 0.03}$	0.97 ± 0.04
Final size	0.30 ± 0.09	0.25 ± 0.02	0.46 ± 0.06	$\mathbf{0.51 \pm 0.03}$	0.50 ± 0.03
Shape	0.25 ± 0.06	0.26 ± 0.02	0.25 ± 0.02	$\mathbf{0.38 \pm 0.02}$	0.37 ± 0.03
Init. persp.	0.20 ± 0.06	0.17 ± 0.00	0.25 ± 0.01	$\mathbf{0.28 \pm 0.03}$	0.27 ± 0.02
Final persp.	0.17 ± 0.05	0.18 ± 0.01	0.15 ± 0.01	$\mathbf{0.25 \pm 0.05}$	0.24 ± 0.02
	dSprites				
R	0.02 ± 0.00	0.03 ± 0.00	0.01 ± 0.00	$\mathbf{0.07 \pm 0.00}$	0.04 ± 0.01
G	0.02 ± 0.00	0.03 ± 0.00	0.01 ± 0.00	$\mathbf{0.08 \pm 0.01}$	0.04 ± 0.01
B	0.03 ± 0.01	0.03 ± 0.00	0.01 ± 0.00	$\mathbf{0.07 \pm 0.01}$	0.03 ± 0.00
Shape	0.46 ± 0.02	0.45 ± 0.01	0.34 ± 0.01	$\mathbf{0.51 \pm 0.05}$	0.50 ± 0.04
Scale	0.44 ± 0.03	0.50 ± 0.01	0.18 ± 0.01	$\mathbf{0.56 \pm 0.03}$	0.47 ± 0.03
Rot.	0.12 ± 0.01	0.10 ± 0.01	0.03 ± 0.00	$\mathbf{0.49 \pm 0.04}$	0.10 ± 0.04
Traj.	0.62 ± 0.02	0.59 ± 0.01	0.60 ± 0.01	$\mathbf{0.63 \pm 0.03}$	0.60 ± 0.03
	LPC				
Body	0.18 ± 0.01	0.54 ± 0.04	0.13 ± 0.00	$\mathbf{0.97 \pm 0.01}$	$\mathbf{0.97 \pm 0.01}$
Gender	0.60 ± 0.04	0.91 ± 0.03	0.50 ± 0.00	$\mathbf{1.00 \pm 0.00}$	0.98 ± 0.02
Shirt	0.72 ± 0.03	0.96 ± 0.03	0.66 ± 0.04	$\mathbf{1.00 \pm 0.00}$	0.85 ± 0.08
Pants	0.68 ± 0.04	0.89 ± 0.02	0.18 ± 0.01	$\mathbf{0.99 \pm 0.00}$	0.87 ± 0.07
Hair	$\mathbf{0.97 \pm 0.02}$	0.89 ± 0.04	0.28 ± 0.01	0.91 ± 0.04	0.92 ± 0.01
Hat	0.69 ± 0.01	0.88 ± 0.02	0.56 ± 0.00	$\mathbf{1.00 \pm 0.00}$	0.99 ± 0.01
Action	0.62 ± 0.04	0.64 ± 0.03	0.61 ± 0.04	0.64 ± 0.03	$\mathbf{0.69 \pm 0.03}$
Perspective	0.94 ± 0.03	$\mathbf{1.00 \pm 0.00}$	0.58 ± 0.03	0.72 ± 0.14	0.98 ± 0.02

For the first scenario, we used the same ground-truth labels to calculate appearance/motion disentanglement, and report the obtained accuracies in Table III, showing that recognizing content is easier than actions. In most of the datasets, our model outperforms the baselines in both content and motion.

For the second scenario, we used the same ground-truth labels to calculate multi-factor disentanglement. This scenario was harder for all the models (c.f. Table IV). However, ours outperformed the rest in most of the cases. This is expected since none of them was trained for multi-factor disentanglement. Notice that each row in Table IV is a classification scheme on different sets of classes. E.g., for dSprites, factor *R* represents the red RGB contribution of the shape, so it is a 256-class problem, while factor *Shape* is a 4-class problem, as there are only four different shapes in the dataset (c.f. Table B.1). In both scenarios, the chunk-wise version of our model outperformed the frame-wise version (MTC-VAE*) most of the times.

H. Latent-Space Traversals

We include some examples of latent-space traversals on the LPC dataset, to show how MTC-VAE could be used for conditional video generation. Fig. 7 shows three trajectories, between two videos x_0 and x_1, separated by 5 steps. The leftmost trajectory traverses the whole latent space, so it is possible to see the complete transformation from x_0 to x_1. The central trajectory is done in the content subspace while remaining stationary in the motion space, so it can be seen how the endpoint is a video with the appearance of x_1 and the motion of x_0. The opposite can be observed in the rightmost trajectory, which only traverses the motion subspace.

All the trajectories are linear, so it is expected that examples in the middle do not look plausible, due to a high probability of sampling outside either $q_\phi(z|x_k)$ or $q_\gamma(w_k|x_k)$. To correctly traverse the latent space requires awareness of its topology. We leave as future work to explore more sophisticated methods to traverse the space of our model [54], [55].

Fig. 8 shows some examples of controllable video generation. We highlight that we do not expect to perform this task perfectly, as we focus exclusively on content-motion disentanglement, so it is normal that visual traits that should be independent (e.g. hair color and skin color) happen to be entangled in the representation. However, it is possible to independently traverse each latent unit of the space and manually check which visual traits were affected. The sequences of Fig. 8 are the endpoints of the trajectories (Appendix G shows the complete trajectories), and each one shows a visual trait that was affected by traversing latent units. Most of them were affected by only one unit: hair color ($z[5]$), hairstyle ($z[7]$), shirt color ($z[11]$), and pants color ($z[13]$). Motion-related units were more difficult to traverse, since independent

Fig. 8. Some controllable visual traits by traversing specific latent units.

motion traits of the video remain more entangled than the appearance ones, as shown by our results on multi-factor classification (Table IV). This means that traversals have a high risk of sampling outside the support of $q_\gamma(w_k|x_k)$. The last example in Fig. 8 was constructed by traversing $w[0]$, $w[4]$, and $w[6]$, and it is clear that we sampled outside $q_\gamma(w_k|x_k)$. This set of experiments show that it is possible to interpret, to some extent, the meaning of the components of the latent representations.

V. Conclusion

Our proposed MTC-VAE for content-motion disentanglement learns to represent videos as a consistent sequence of chunks that are independent at generation time, but dependent at inference time. It considers two extensions to the VAE formulation: (i) training the model such that each chunk implicitly contains information about the whole video under the assumption of content invariability, while separating motion per chunk, and (ii) using the task of video reenactment as an inductive bias to leverage the learning of independent content and motion representations. MTC-VAE yields less latent vectors to represent a video (one per chunk, instead of per frame). To reconstruct one video, it is trained with chunks modeled as independent random variables at generation time. Given that a chunk does not depend on the reconstruction of the previous one, all chunks in a video can be reconstructed in a single forward-pass. The experiments show the capacity of our chunk-wise approach in learning time-dependent and -independent representations from videos as well as the positive impact of video reenactment as an inductive bias to improve such representations. Our ablative study on the size of the chunks shows a better disentanglement and VR performance of middle-sized chunks, over the frame-wise approach. We also showed the superiority of MTC-VAE for multiple-factor disentanglement, even though it was not explicitly trained for more than two factors. We explored the limits of our model in additional experiments on high-resolution videos (Appendix D) and on a real-world human-action dataset (Appendix E). These experiments reveal that the bottleneck of the model is the decoder, whose enhancement we leave for future work as well as exploring different latent and data priors, and devising fusion strategies for the chunks to yield more informative gradients and a better reconstruction, as well as disentanglement quality.

References

[1] F. Locatello *et al.*, "A commentary on the unsupervised learning of disentangled representations," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2020.
[2] P. Chen *et al.*, "Generating visually aligned sound from videos," *IEEE Trans. Image Process.*, vol. 29, pp. 8292–8302, 2020.
[3] A. Ramesh *et al.*, "Zero-shot text-to-image generation," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2021.
[4] F. Locatello *et al.*, "Challenging common assumptions in the unsupervised learning of disentangled representations," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019.
[5] R. Shu *et al.*, "Weakly supervised disentanglement with guarantees," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020.
[6] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017.
[7] C. Feichtenhofer *et al.*, "SlowFast networks for video recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2019.
[8] Y. Wang *et al.*, "G^3AN: Disentangling appearance and motion for video generation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020.
[9] A. Aich *et al.*, "Non-adversarial video synthesis with learned priors," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020.
[10] K. Su, X. Liu, and E. Shlizerman, "Predict & cluster: Unsupervised skeleton based action recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 9631–9640.
[11] H. Shi *et al.*, "Bidirectional long short-term memory variational autoencoder," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2018, p. 165.
[12] X. Chen *et al.*, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2016.
[13] I. Higgins *et al.*, "β-VAE: Learning basic visual concepts with a constrained variational framework," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.
[14] T. Q. Chen *et al.*, "Isolating sources of disentanglement in variational autoencoders," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2018.
[15] H. Kim and A. Mnih, "Disentangling by factorising," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2018.
[16] E. Denton and V. Birodkar, "Unsupervised learning of disentangled representations from video," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017.
[17] R. Villegas *et al.*, "Decomposing motion and content for natural video sequence prediction," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.
[18] J.-T. Hsieh *et al.*, "Learning to decompose and disentangle representations for video prediction," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2018.
[19] Y. Ge *et al.*, "FD-GAN: Pose-guided feature distilling GAN for robust person re-identification," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2018.
[20] E. Denton and R. Fergus, "Stochastic video generation with a learned prior," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2018.
[21] A. X. Lee *et al.*, "Stochastic adversarial video prediction," 2019, *arXiv:1804.01523*.
[22] Y. Li and S. Mandt, "Disentangled sequential autoencoder," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2018.
[23] Y. Zhu *et al.*, "S3VAE: Self-supervised sequential VAE for representation disentanglement and data generation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020.
[24] E. Zakharov *et al.*, "Few-shot adversarial learning of realistic neural talking head models," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2019.
[25] Y. Nirkin, Y. Keller, and T. Hassner, "FSGAN: Subject agnostic face swapping and reenactment," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2019.
[26] L. Chen *et al.*, "Lip movements generation at a glance," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018.
[27] H. Zhou *et al.*, "Talking face generation by adversarially disentangled audio-visual representation," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2019.
[28] C. Chan *et al.*, "Everybody dance now," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2019.
[29] Y. Zhou *et al.*, "Dance dance generation: Motion transfer for internet videos," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, 2019.
[30] L. Liu *et al.*, "Neural rendering and reenactment of human actor videos," *ACM Trans. Graph.*, 2019.
[31] Z. Yang *et al.*, "TransMoMo: Invariance-driven unsupervised video motion retargeting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020.

[32] A. Bansal *et al.*, "Recycle-GAN: Unsupervised video retargeting," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018.
[33] A. Siarohin *et al.*, "Animating arbitrary objects via deep motion transfer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019.
[34] A. Siarohin *et al.*, "Motion representations for articulated animation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 13653–13662.
[35] L. Zhao *et al.*, "Learning to forecast and refine residual motion for image-to-video generation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018.
[36] J. Xie *et al.*, "Motion-based generator model: Unsupervised disentanglement of appearance, trackable and intrackable motions in dynamic patterns," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2020.
[37] M. Jaderberg *et al.*, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2015.
[38] P. D. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2013.
[39] F. Locatello *et al.*, "Weakly-supervised disentanglement without compromises," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2020.
[40] M. F. Mathieu *et al.*, "Disentangling factors of variation in deep representation using adversarial training," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2016.
[41] D. Bouchacourt, R. Tomioka, and S. Nowozin, "Multi-level variational autoencoder: Learning disentangled representations from grouped observations," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2018.
[42] H. Hosoya, "Group-based learning of disentangled representations with generalizability for novel contents," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019.
[43] T. Kanade, J. F. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recognit.*, 2000.
[44] P. Lucey *et al.*, "The extended Cohn-Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression," in *Proc. IEEE Int. Conf. Comput. Vis., Pattern Recognit. Workshops (CVPRW)*, 2010.
[45] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using LSTMs," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2015.
[46] N. Aifanti, C. Papachristou, and A. Delopoulos, "The MUG facial expression database," in *Proc. Int. Workshop Image Audio Anal. Multimedia Interact. Services (WIAMIS)*, 2010.
[47] A. Kumar, P. Sattigeri, and A. Balakrishnan, "Variational inference of disentangled latent concepts from unlabeled observations," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.
[48] C. Eastwood and C. K. I. Williams, "A framework for the quantitative evaluation of disentangled representations," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.
[49] K. Ridgeway and M. C. Mozer, "Learning deep disentangled embeddings with the F-statistic loss," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2018.
[50] Z. Wang *et al.*, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, 2004.
[51] M. Heusel *et al.*, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017.
[52] A. Siarohin *et al.*, "Deformable GANs for pose-based human image generation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018.
[53] G. Balakrishnan *et al.*, "Synthesizing images of humans in unseen poses," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018.
[54] J. C. Ye and W. K. Sung, "Understanding geometry of encoder-decoder CNNs," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019.
[55] Z. Song, O. Koyejo, and J. Zhang, "Toward a controllable disentanglement network," *IEEE Trans. Cybern.*, 2020.

Juan F. Hernández Albarracín received the bachelor's degree in computer engineering from the National University of Colombia, in 2014, and the M.Sc. degree in computer science from the University of Campinas, Brazil, in 2017, where he is currently pursuing the Ph.D. degree. He has experience in machine learning and computer vision, focusing on evolutionary computing, deep learning, and generative models applied for image/video classification and synthesis.

Adín Ramírez Rivera (Senior Member, IEEE) received the B.Eng. degree in computer engineering from the Universidad de San Carlos de Guatemala (USAC), Guatemala, in 2009, and the M.Sc. and Ph.D. degrees in computer engineering from Kyung Hee University, South Korea, in 2013. He is currently an Associate Professor with the Department of Informatics, University of Oslo, Norway. His research interests are video understanding (including video classification, semantic segmentation, spatiotemporal feature modeling, and generation) and understanding and creating complex feature spaces.