Video Reenactment as Inductive Bias for Content-Motion Disentanglement

J. F. Hernández Albarracín, and A. 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 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

Juan F. Hernández Albarracín is with Institute of Computing, University of Campinas, SP, Brazil, e-mail juan.albarracin@ic.unicamp.br. Adín Ramírez Rivera is with Department of Informatics, University of Oslo, Norway, e-mail adinr@uio.no; and part of this work was done at the Institute of Computing, University of Campinas.

Juan F. Hernández Albarracín was funded by the São Paulo Research Foundation (FAPESP) under grant No. 2017/16144-2. A. Ramírez Rivera was funded by the Brazilian National Council for Scientific and Technological Development (CNPq) under grant No. 307425/2017-7; and in part by FAPESP under grant No. 2019/07257-3. Juan F. Hernández Albarracín and A. Ramírez Rivera were funded by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior—Brasil (CAPES)—Finance Code 001.

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

Pre-print to appear in IEEE Trans. on Image Processing.
Digital Object Identifier 10.1109/TIP.2022.3153140.

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 Locatello et al. [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-C 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.

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,

[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.

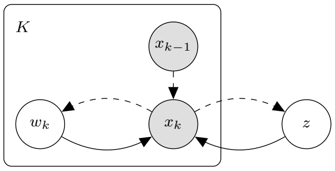


Fig. 1. In the generative model (solid arrows), K chunks $\{x_k\}$ (observed) share the same content z, while having their own motion w_k. During inference (dashed arrows), the latent variables z and w_k are inferred from each chunk, while each chunk x_k also depends on the previous one.

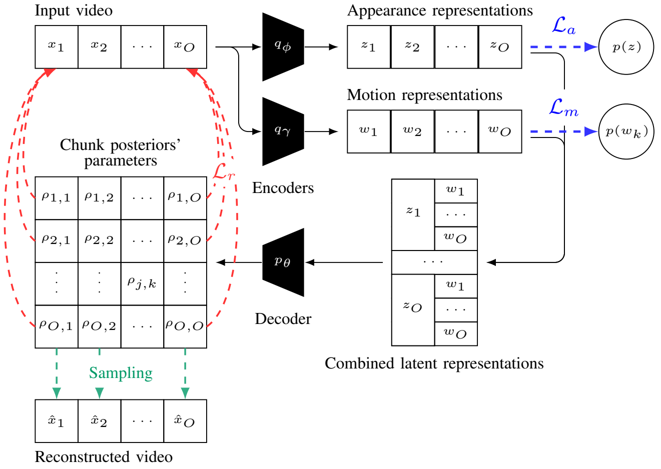


Fig. 2. We feed consecutive chunks $\{x_k\}_{k=1}^{O}$ to the encoders q_ϕ and q_γ, yielding their representations, $\{w_k\}_{k=1}^{O}$ and $\{z_j\}_{j=1}^{O}$. We concatenate all combinations of z_j's and w_k's, and decode them to obtain the p.d.f. parameters $\rho_{j,k}$ for the k-th chunk posteriors $p_\theta(x_k \mid w_k, z_j)$. 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.

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.,

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

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

$$\begin{aligned} \arg\max_{\phi,\gamma,\theta} \mathop{\mathbb{E}}_{\tilde{q}(x_{1:k})} \sum_k \Big\{ & \mathop{\mathbb{E}}_{q_\phi} \mathop{\mathbb{E}}_{q_\gamma} \left[\log p_\theta(x_k \mid w_k, z)\right] \\ & - \mathrm{KL}(q_\gamma(w_k \mid x_k) \parallel p(w_k)) \\ & - \mathrm{KL}(q_\phi(z \mid x_k) \parallel p(z)) \Big\}. \end{aligned} \tag{2}$$

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 $q_\phi(z \mid x_k)$ and $q_\gamma(w_k \mid x_k)$ (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 $\tilde{q}(x_{1:k}) = \prod_k q(x_k \mid x_{k-1})$ that models a Markovian temporal relation between them.[2] We approximate the chunk distribution through a sampling process on the videos, and 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 $q_\phi(z \mid x_k)$ and $q_\gamma(w_k \mid x_k)$, 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 $z \sim q_\phi(z \mid x_k)$—cf. Section III-A. To force our model to learn this constraint, we train it to maximize $\log p_\theta(x_k \mid w_k, z_j)$ 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 $\log p(x_k \mid w_k, z)$ term (2) to fulfill this constraint. So our final reconstruction loss is

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

where $z_j \sim q_\phi(z \mid x_j)$, $w_k \sim q_\gamma(w_k \mid x_k)$, 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 $p_\theta(x_k \mid w_k, z_j)$, 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

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

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

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

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

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 $p(x_k \mid w_k, z)$ to be equivalent, i.e., $p(x \mid w_l^d, z_j^s) \equiv p(x \mid w_l^d, z_i^s)$, where $z_i^s \sim q(z \mid s_i)$, $z_j^s \sim q(z \mid s_j)$, and $w_l^d \sim q(w_k \mid d_l)$, by minimizing the KL divergence between every two posteriors that fit the described case. Let

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

be our BRL, where $\mathrm{SKL}(P \parallel Q) = \frac{1}{2}(\mathrm{KL}(P \parallel Q) + \mathrm{KL}(Q \parallel P))$ 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

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

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.

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 10 000 videos, except for CK+ (320), LPC (200 000), 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.

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.

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 (cf. Section IV-C and Appendix F).

A. 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 Ridgeway and Mozer [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

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	**0.01±0.01**	0.11±0.08	0.53±0.10	140.25±51.13	0.81±0.03	0.02±0.02	0.00	0.64±0.01	80.27±3.17
dis-VAE	0.50	0.00	0.08±0.06	0.40±0.03	**71.24±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	**0.01**	0.03±0.02	0.54±0.05	136.00±81.12	0.63±0.02	0.00	0.00	**0.79±0.02**	62.75±9.87
MTC-VAE	0.50±0.02	**0.01**	**0.41±0.14**	0.67±0.06	119.47±51.00	**0.93±0.06**	**0.11±0.11**	**0.60±0.40**	0.67±0.01	**41.72±3.31**
MTC-VAE*	0.50±0.01	**0.01**	0.39±0.11	**0.73±0.02**	100.80±46.82	0.86±0.01	0.00	0.11±0.03	0.67±0.01	42.59±4.09
	CK+					MMNIST				
β-TCVAE	0.79±0.05	**0.03±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	**0.71±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	**38.79±17.63**	0.52±0.01	0.01	0.02±0.02	0.58±0.02	179.08±50.49
MTC-VAE	**0.86±0.04**	0.02±0.01	**0.13±0.05**	0.66±0.12	63.13±22.50	**0.95±0.04**	**0.11±0.07**	**0.10±0.05**	0.68±0.01	**102.11±0.99**
MTC-VAE*	0.85±0.02	**0.03±0.01**	0.05±0.02	**0.68±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	0.04±0.01	**0.79±0.03**	79.34±6.18	0.74±0.05	**0.05±0.04**	0.23±0.03	0.51±0.01	44.78±3.32
dis-VAE	0.70±0.02	0.01	0.01	**0.79**	97.07±1.73	**0.76±0.03**	0.01±0.01	0.11±0.03	**0.78±0.01**	62.84±3.45
SVG-LP	0.61±0.05	0.00	0.00	**0.79±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	**0.04±0.01**	**0.10±0.01**	0.78	**57.18±6.43**	0.72±0.04	0.01±0.01	0.73±0.05	0.63±0.02	**28.79±1.15**
MTC-VAE*	**0.92±0.01**	0.02±0.02	0.01	0.77±0.01	105.79±5.86	0.70±0.09	0.04±0.02	**0.76±0.10**	0.66±0.06	43.86±13.15

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.

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
	MTC-VAE	0.27±0.05	**0.19±0.07**	**0.08±0.03**
	MTC-VAE*	**0.31±0.02**	0.14±0.05	0.05±0.02
dSprites	β-TCVAE	0.28±0.01	0.02±0.01	0.01
	dis-VAE	0.28	0.02±0.01	0.01
	SVG-LP	0.29	0.00	0.00
	MTC-VAE	**0.33±0.02**	**0.11±0.01**	**0.02**
	MTC-VAE*	0.29±0.01	0.07±0.02	0.01
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	0.01	0.01±0.01
	MTC-VAE	0.41±0.07	**0.21±0.05**	**0.89±0.01**
	MTC-VAE*	**0.43±0.06**	0.18±0.03	**0.89**

B. Video Reenactment

We generated 10 000 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 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

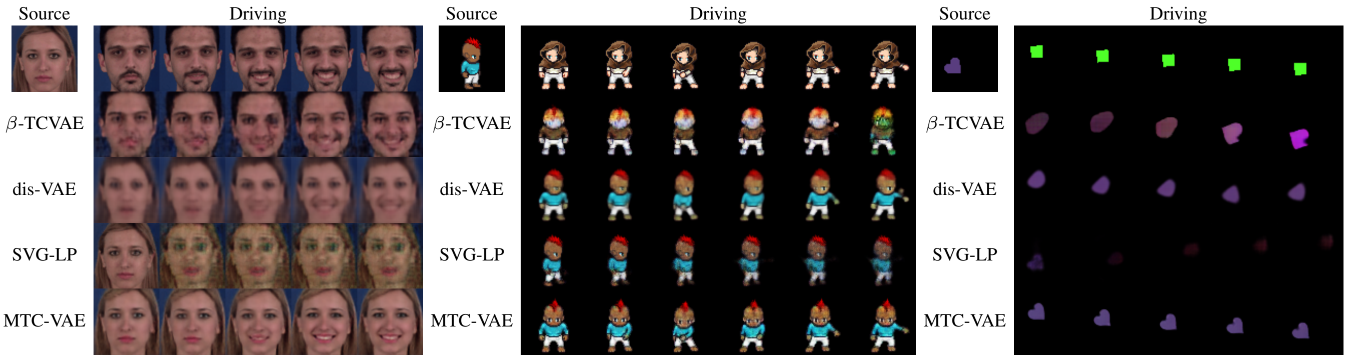


Fig. 3. Reenactment results. Each set shows the reenacted video of each method with the appearance of *source* and the motion of *driving*.

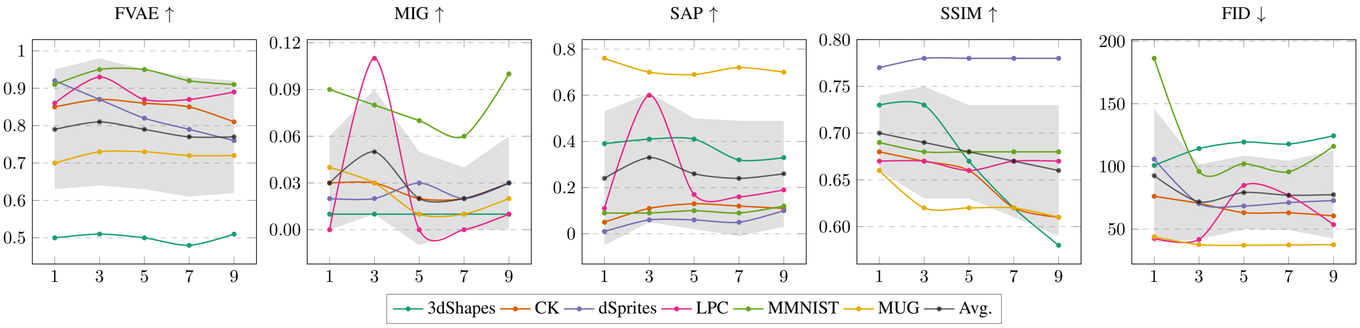


Fig. 4. Study of the chunk size vs. several metrics. The gray area shows one standard deviation away from the average plot.

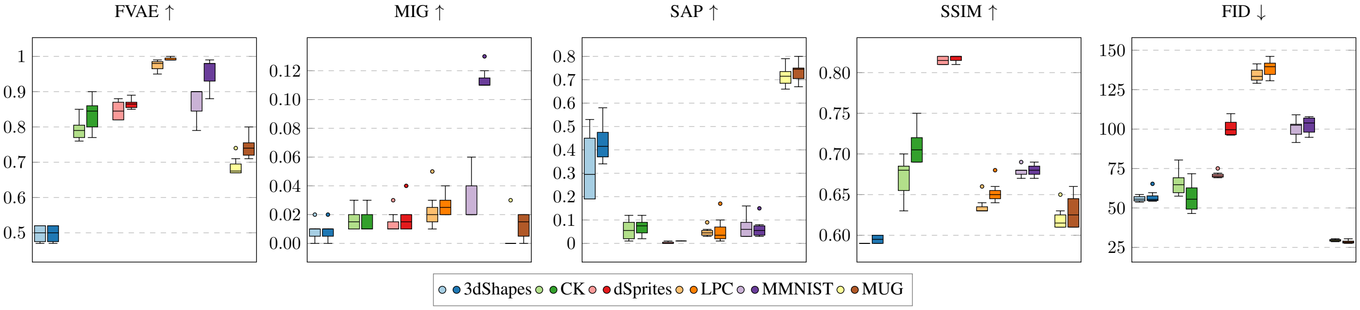


Fig. 5. Ablation on BRL. Light colors indicate the absence of $\mathcal{L}_b$ ($\lambda = 0$), while dark colors indicate its presence ($\lambda = 1$).

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.

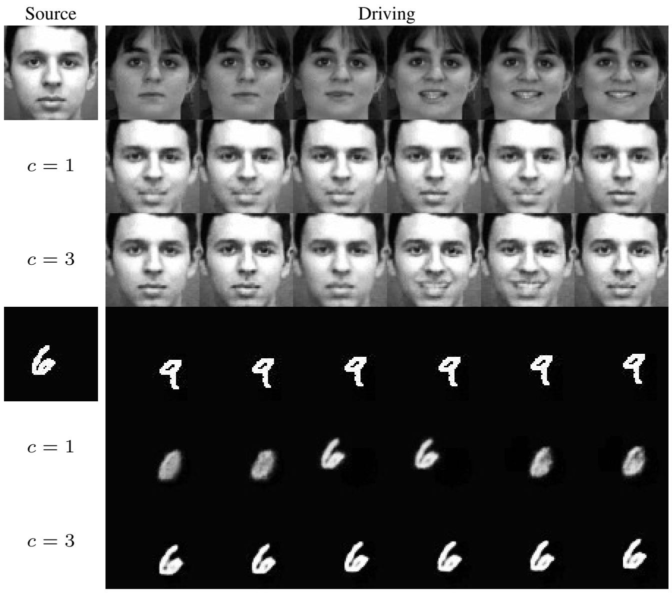


Fig. 6. Qualitative comparison of performances for the frame version of MTC-VAE ($c = 1$) and the chunk version with a temporal neighborhood of $c = 3$ in reenactment quality and inter-chunk consistency.

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	**1.00**	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	0.60±0.01
MTC-VAE	**0.95±0.01**	**0.59±0.01**	0.97±0.01	**0.68±0.07**	**0.61±0.01**	**0.63±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	**0.99±0.01**	0.32±0.03	**0.26±0.05**	**1.00**	0.54±0.06
dis-VAE	0.43±0.02	0.95±0.01	**0.54±0.08**	0.16±0.01	**1.00**	0.55±0.02
SVG-LP	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	**1.00**	**0.79±0.05**
MTC-VAE*	**0.68±0.03**	0.97±0.01	0.45±0.07	0.19±0.02	**1.00**	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	**1.00**	0.27±0.05	0.99±0.01	0.99±0.01
Wall hue	0.95±0.07	**1.00**	0.53±0.11	0.97±0.03	0.97±0.03
Obj. hue	0.82±0.10	**1.00**	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	**0.98±0.03**	0.97±0.04
Final size	0.30±0.09	0.25±0.02	0.46±0.06	**0.51±0.03**	0.50±0.03
Shape	0.25±0.06	0.26±0.02	0.25±0.02	**0.38±0.02**	0.37±0.03
Init. persp.	0.20±0.06	0.17	0.25±0.01	**0.28±0.03**	0.27±0.02
Final persp.	0.17±0.05	0.18±0.01	0.15±0.01	**0.25±0.05**	0.24±0.02
			dSprites		
R	0.02	0.03	0.01	**0.07**	0.04±0.01
G	0.02	0.03	0.01	**0.08±0.01**	0.04±0.01
B	0.03±0.01	0.03	0.01	**0.07±0.01**	0.03
Shape	0.46±0.02	0.45±0.01	0.34±0.01	**0.51±0.05**	0.50±0.04
Scale	0.44±0.03	0.50±0.01	0.18±0.01	**0.56±0.03**	0.47±0.03
Rot.	0.12±0.01	0.10±0.01	0.03	**0.49±0.04**	0.10±0.04
Traj.	0.62±0.02	0.59±0.01	0.60±0.01	**0.63±0.03**	0.60±0.03
			LPC		
Body	0.18±0.01	0.54±0.04	0.13	**0.97±0.01**	**0.97±0.01**
Gender	0.60±0.04	0.91±0.03	0.50	**1.00**	0.98±0.02
Shirt	0.72±0.03	0.96±0.03	0.66±0.04	**1.00**	0.85±0.08
Pants	0.68±0.04	0.89±0.02	0.18±0.01	**0.99**	0.87±0.07
Hair	**0.97±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	**1.00**	0.99±0.01
Action	0.62±0.04	0.64±0.03	0.61±0.04	0.64±0.03	**0.69±0.03**
Perspective	0.94±0.03	**1.00**	0.58±0.03	0.72±0.14	0.98±0.02

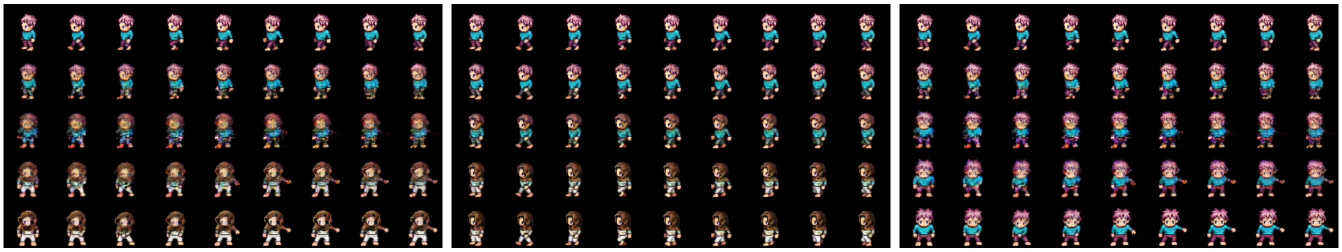
C. Ablation Studies

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 $c = 5$ (cf. 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.1 presents more detailed results.

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 (cf. Appendix F). Since the complexity of the model is quadratic w.r.t. to O, higher values are not worth considering.



(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.

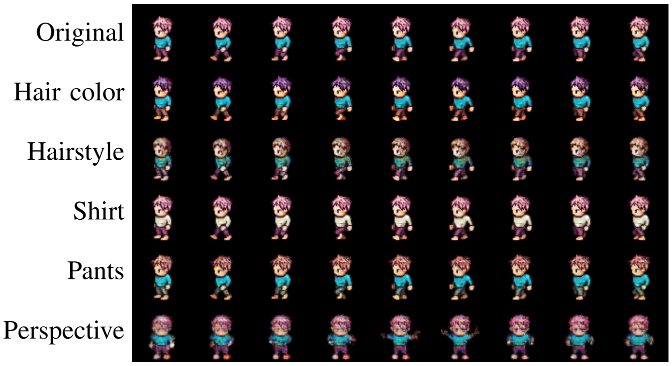


Fig. 8. Some controllable visual traits by traversing specific latent units.

D. 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.

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 (cf. 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 (cf. Table B.1). In both scenarios, the chunk-wise version of our model outperformed the frame-wise version (MTC-VAE*) most of the times.

E. 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 $q_\phi(z \mid x_k)$ or $q_\gamma(w_k \mid x_k)$. 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 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 $q_\gamma(w_k \mid x_k)$. 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 $q_\gamma(w_k \mid x_k)$. 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.

Appendix A
Derivation of the ELBO

In this section, we present the derivations of the statements introduced in Section IV to construct the loss functions of our model based on the Evidence Lower Bound (ELBO) of the expected log-likelihood of our model.

Let the video x be a sequence of K chunks, $x = (x_k)_{k=1}^{K}$. Similarly, let $w = (w_k)_{k=1}^{K}$ be the sequence of motion representations for all the chunks on the video x. For the k-th chunk, we model the content and motion as independent latent variables z and w_k.

We are interested in maximizing the expected log-likelihood of the videos w.r.t. the data empirical distribution $q(x)$. First, let's consider it based on the sets x and w such that

$$\begin{aligned}
\mathop{\mathbb{E}}_{q(x)} \log p(x) &= \mathop{\mathbb{E}}_{q(x)} \log \iint p(x, w, z) \, \mathrm{d}w \, \mathrm{d}z, && (8)\\
&= \mathop{\mathbb{E}}_{q(x)} \log \iint \frac{q(w, z \mid x)}{q(w, z \mid x)} p(x, w, z) \, \mathrm{d}w \, \mathrm{d}z, && (9)\\
&\geq \mathop{\mathbb{E}}_{q(x)} \mathop{\mathbb{E}}_{q(w,z \mid x)} \left[\log \frac{p(x, w, z)}{q(w, z \mid x)} \right], && (10)\\
&= \mathop{\mathbb{E}}_{q(x)} \mathop{\mathbb{E}}_{q(w,z \mid x)} \left[\log \frac{p(x \mid w, z) p(w) p(z)}{q(w \mid x) q(z \mid x)} \right], && (11)\\
&= \mathop{\mathbb{E}}_{q(x)} \mathop{\mathbb{E}}_{q(w,z \mid x)} \left[\log p(x \mid w, z) + \log \frac{p(w)}{q(w \mid x)} \right. \\
&\qquad \left. + \log \frac{p(z)}{q(z \mid x)} \right]. && (12)
\end{aligned}$$

However, we are interested in modeling the chunks and their respective latent variables. Hence, we need a change in the variable. First, let's consider the video distribution based on its chunks as $q(x) = \prod_{k=1}^{K} q(x_k \mid x_{k-1})$, such that $q(x_1 \mid x_0) \equiv q(x_1)$, i.e., we consider the video as a Markov chain of chunks. Then, by plugging into the sequence representations of the video x and the motion latent variable w, we get

$$\begin{aligned}
\mathop{\mathbb{E}}_{q(x)} \log p(x) \geq \mathop{\mathbb{E}}_{\prod_k q(x_k \mid x_{k-1})} \mathop{\mathbb{E}}_{\prod_k q(w_k, z \mid x_k)} \Bigg[& \\
& \log \prod_k p(x_k \mid w_k, z) + \log \frac{\prod_k p(w_k)}{\prod_k q(w_k \mid x_k)} \\
& + \log \frac{\prod_k p_k(z)}{\prod_k q(z \mid x_k)} \Bigg]. && (13)
\end{aligned}$$

We denote the content prior distribution over the sequence of chunks as $p(z) = \prod_k p_k(z)$. (Abusing notation, we will refer to these priors as $p_k(z) = p(z)$ since they are all the same over the sequence.) Then, we can simplify the expected log-likelihood as

$$\begin{aligned}
\mathop{\mathbb{E}}_{q(x)} \log p(x) &\geq \mathop{\mathbb{E}}_{\prod_k q(x_k \mid x_{k-1})} \sum_k \mathop{\mathbb{E}}_{q(w_k, z \mid x_k)} \Bigg[\log p(x_k \mid w_k, z) \\
&\quad + \log \frac{p(w_k)}{q(w_k \mid x_k)} + \log \frac{p(z)}{q(z \mid x_k)} \Bigg], && (14)\\
&= \mathop{\mathbb{E}}_{\prod_k q(x_k \mid x_{k-1})} \sum_k \Bigg\{ \mathop{\mathbb{E}}_{q(w_k, z \mid x_k)} \left[\log p(x_k \mid w_k, z) \right] \\
&\quad - \mathrm{KL}(q(w_k \mid x_k) \parallel p(w_k)) \\
&\quad - \mathrm{KL}(q(z \mid x_k) \parallel p(z)) \Bigg\}. && (15)
\end{aligned}$$

Notice that the final function corresponds to the expectation over the empirical chain of chunks. In our experiments, we simulate this process by sampling throughout the video to obtain the chunks and then compute the summation over the losses.

Appendix B
Implementation Details

A. Architecture

Our model consists of two encoding streams, corresponding to $q_\phi(z \mid x_k)$ and $q_\gamma(w_k \mid x_k)$, and one decoding stream, corresponding to $p_\theta(x_k)$, defined in Section III). All the streams 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.

As previous works on DRL from video and VR [33, 56, 57], we used an appearance-suppressed input to the motion encoding stream. In our case, we added a layer that calculates the optical flow of the chunk with the Lucas-Kanade method [58].

We use a *Bernoulli observation VAE* where the observed samples of the decoder are used as logits of a Bernoulli distribution, in contrast with the traditional Gaussian observations. We observed a remarkable superiority at reconstruction time of the Bernoulli observations, particularly for videos where the proportion of the object of interest w.r.t. the background is reasonably low. We consider standard Normal priors for both the content and motion latent representations, i.e., $p(z) = \mathcal{N}(0, 1)$, and $p(w_k) = \mathcal{N}(0, 1)$ for all k.

Algorithm 1 Chunk Sequence Learning training procedure algorithm.

```
Input: batch $\mathcal{X}$, video length $T_x$, chunk sequence length $O$, chunk size $c$
for each $x \in \mathcal{X}$ do
  $h \sim U(1, T_x - cO)$    ▷ Random starting position
  Sample $\{x_k\}_{k=1}^{O}$, $O$ consecutive size $c$ chunks starting at $h$
      ▷ Approx. of $x_k \sim q(x_k \mid x_{k-1}), \forall\, k$
  for $k = 1$ to $O$ do
    $w_k \sim q(w_k \mid x_k)$
    for $j = 1$ to $O$ do
      $z_j \sim q(z \mid x_j)$
      $\mathcal{L}_r$ += $\log p_\theta(x_k \mid w_k, z_j)$
    end for
    $\mathcal{L}_a$ −= $\mathrm{KL}(q(z \mid x_k) \parallel p(z))$
    $\mathcal{L}_m$ −= $\mathrm{KL}(q(w_k \mid x_k) \parallel p(w_k))$
  end for
end for
for each $s, d \in \mathcal{X} \times \mathcal{X}, s \neq d$ do
  $h_s \sim U(1, T_s - cO)$
  $h_d \sim U(1, T_d - cO)$
  Sample $\{s_k\}_{k=1}^{O}$, $O$ consecutive size $c$ chunks starting at $h_s$
  Sample $\{d_k\}_{k=1}^{O}$, $O$ consecutive size $c$ chunks starting at $h_d$
  for $l = 1$ to $O$ do
    $w_l^d \sim q(w_k \mid d_l)$
    for $j = 1$ to $O$ do
      $z_j \sim q(z \mid s_j)$
      for $i = 1$ to $O$ do
        $z_i \sim q(z \mid s_i)$
        $\mathcal{L}_b$ −= $\mathrm{SKL}(p_\theta(x_l \mid w_l^d, z_j^s) \parallel p_\theta(x_l \mid w_l^d, z_i^s))$
            ▷ SKL as defined in Equation 6
      end for
    end for
  end for
end for
$\mathcal{L} = \mathcal{L}_r + \lambda \mathcal{L}_b + \beta(\mathcal{L}_a + \mathcal{L}_m)$
$(\phi, \gamma, \theta)$ −= $\nabla_{(\phi,\gamma,\theta)} \mathcal{L}$   ▷ Backprop the loss over the parameters
```

The loss functions $\mathcal{L}_r$, $\mathcal{L}_a$, and $\mathcal{L}_m$ require consecutive chunks of a unique video, while $\mathcal{L}_b$ requires chunks of the source and the driving video. In order to train all the losses in the same forward pass, we feed the model with batches of O-tuples of consecutive chunks, as shown in the Algorithm 1, which describes in detail the training procedure of our model. For $\mathcal{L}_b$, we create a reversed copy of the batch to be used as the batch of driving videos, while the original batch corresponds to the source videos. This gives a sense of completeness for training because it ensures that source videos will also act as driving videos, and vice-versa, in the same forward pass. Although Algorithm 1 is expressed so, for each batch, all the possible pairs of videos are used as source and driving, in practice, it is unfeasible because the calculation of $\mathcal{L}_b$ takes cubic time w.r.t. O. We can argue that, by means of the stochastic batched training, most of the possible pairs of videos can be covered for our model, if trained for enough time.

B. Model Training

Making use of labels describing the factors of variation in a video, such as the identity of the object of interest or its motion, we split the datasets in training-test and tested our model in two generalization scenarios. We will refer to this as a *soft generalization* scenario, in which the model is requested to reconstruct novel videos from contents and motions seen in training time. We included two *hard generalization* scenarios: the *appearance holdout* scenario, in which the model is requested to reconstruct novel videos with appearances that were not seen in training time, and the *motion holdout* scenario, in which the model is requested to reconstruct novel videos with motions that were not seen in training time. The quantitative results presented in the main text of this paper (Table I, and Figs. 3 and 4) correspond to the soft generalization scenario. We show in Appendix F the quantitative performance of MTC-VAE and the baselines in the three generalization scenarios, as well as detailed results on the ablation studies, corresponding to the soft generalization scenario.

To run the complete set of experiments, including the baselines, the hyper-parameter search, and the ablation study of our model, we used a total of 12 GPUs Titan X, Titan Xp, RTX 2080 Ti, RTX 5000, GTX 1080 Ti, and Tesla P100. However, our model can be executed in a single GPU of 12 GB memory, and the training time varies from 20 minutes to 12 hours, depending on the length of the videos, the chunk size, and more importantly, the order of the model. Given two videos, the chunk-wise reenactment process takes no more than 2 seconds.

C. Data

Cohn-Kanade (CK+) facial expressions dataset. [43, 44] 326 gray-scale videos of 64×64 pixels of 118 characters performing six actions: anger, disgust, fear, happy, sad, and surprise. This dataset only provides two-factor labels: identity and expression.

Multimedia Understanding Group (MUG) facial expressions dataset. [46] 931 RGB videos of 64×64 pixels of 52 characters performing six actions: anger, disgust, fear, happy, sad, and surprise. This dataset only provides two-factor labels: identity and expression.

Liberated Pixel Cup (LPC).[3] We generated 10 000 RGB videos of 64×64 pixels creating 24 motions classes performed by the characters, which correspond to six actions (walk, spellcast, thrust, shoot, hurt, and slash) times four perspectives (front, back, left, and right). For content, we combined different genders, body types, hairstyles, and clothes, creating a large number of different identities. In total, we generated 10 000 videos for training. Table B.1 shows the factors of variation used to evaluate multi-factor disentanglement (Table II). For content-motion disentanglement (Table I), we joined these factors in two supersets, as pointed in the **S** column in Table B.1.

Moving MNIST (MMNIST). [45] We generated 10 000 binary videos of 64×64 pixels with ten identities, corresponding to the digits from 0 to 9. All the videos have 32 frames. The digits follow linear trajectories from random starting points. We created 14 motion classes that distinguish the direction of the trajectory (e.g., down, diagonal up, right-left, left-right). This dataset only provides two-factor labels: identity and motion.

[3] https://github.com/jrconway3/Universal-LPC-spritesheet as as September, 2019.
http://lpc.opengameart.org/ as September, 2019.

TABLE B.1. Factors of variation for each multi-factor dataset.

Factor (size)	Labels	S
3dShapes		
floor_hue (10)	[0-9]	C
wall_hue (10)	[0-9]	
object_hue (10)	[0-9]	
shape (4)	[0-3]	
init_size (8)	[0-7]	M
final_size (8)	[0-7]	
init_angle (15)	[0-14]	
final_angle (15)	[0-14]	
speed (2)	[1-2]	
dSprites		
R (256)	[0-255]	C
G (256)	[0-255]	
B (256)	[0-255]	
orientation (40)	[0-39]	
shape (3)	[0-2]	
scale (6)	[0-5]	
x_init (32)	[0-31]	M
y_init (32)	[0-31]	
x_final (32)	[0-31]	
y_final (32)	[0-31]	
speed (3)	[1-3]	
trajectory (2)	linear arc	
LPC		
Sex (2)	male female	C
Body(10)	light dark dark2 darkelf orc redorc darkelf2 tanned tanned2 skeleton	
Shirt (8)	longsleeve_maroon sleeveless_teal sleeveless_maroon longsleeve_teal longsleeve_brown sleeveless_white longsleeve_white sleeveless_brown	
Pants (4)	magenta red teal white	
Hat (8)	none bandana_red cap_leather chain helmet_golden helmet_metal hood_chain hood_cloth	
Hair (17)	none bangslong2_brunette bedhead_brunette bangslong2_green swoop_red mohawk_red shoulderl_raven plain_blue loose_blue shoulderl_pink messy2_raven bedhead_green messy2_pink swoop_white mohawk_white loose_blonde plain_blonde	
Action (6)	spellcast thrust walk slash shoot hurt	M
Perspective (4)	back front left right	

dSprites. We took the data provided in Deepmind's project[4] and generated 10 000 videos of 64 × 64 pixels from the images provided. The moving sprites have all possible sizes and shape types, yielding a large number of different identities. We can tweak the starting position, the final position, the velocity, and the type of trajectory (either linear or curved) of the sprite. This yields an explosive number of motion classes so, when taking the disentanglement metrics, we decided to label the videos with either linear or curved trajectory. Table B.1 shows the factors of variation used to evaluate multi-factor disentanglement (Table II). For content-motion disentanglement (Table I), we joined these factors in two supersets, as pointed in the **S** column in Table B.1.

3dShapes. We took the data provided in the Deepmind's project[5] and generated 10 000 videos of 64 × 64 pixels from the images provided. We can take the hue of the floor, the shape, and the walls, as well as the type of shape, yielding different identities. Regarding motion, we teak the size of the shape (yielding a heart-beat-like motion) and the perspective (yielding a camera-motion effect), attaining a large (but not explosive) number of motion classes. Table B.1 shows the factors of variation used to evaluate multi-factor disentanglement (Table II). For content-motion disentanglement (Table I), we joined these factors in two supersets, as pointed in the **S** column in Table B.1.

D. Baselines

As said in Section IV, we compared our method against dis-VAE by Li and Mandt [22], SVG-LP by Denton and Fergus [20], and β-TCVAE by Chen et al. [14]. We executed code already available for the three models. In the case of β-TCVAE, we extended the code made available by its authors,[6] so their convolutional streams become 3D ones, in order to support chunks of videos. For SVG-LP, we used the official code provided by the authors[7] For dis-VAE, we used a public reproduction of the method[8] whose results on the LPC dataset seem to match with the ones presented in the paper. In particular, we used the encoder referred as "full q" by the authors.

We tuned the hyper-parameters of the three models, by testing a small set of variations, as described below, on all the datasets, in the soft generalization scenario, and extracted the five evaluation metrics (MIG, FVAE, SAP, SSIM, and FID). For dis-VAE, we contrasted the "factored q" against the "full q" in order to determine which model had the best disentanglement and reconstruction performance. We determined that the latter had the best performance. For β-TCVAE we tunned β and λ, and the effect of annealing each one of them while training. For SVG-LP we tested between the VGG and the DC-GAN architectures, concluding that the latter attained the best results, so we used it for comparison. We determined that annealing λ while keeping β fixed (1.0 for MMNIST and dSprites and 5.0 for the rest of datasets) obtained the best results. The best baseline configurations for each dataset were compared against our method, as shown in Table F.6.

E. Metrics Calculation

In order to calculate the disentanglement metrics, we took all the videos of the test set (20%, according to the 5-fold cross-validation setup mentioned in Section IV), divided them into chunks, and calculated the latent representations of each one of the chunks. In the case of dis-VAE, the representations were per frame. In total, for CK+, approximately 64 videos were used to calculate MIG, FVAE, and SAP while, for the rest of datasets, approximately 2000 videos were used.

We evaluated content-motion disentanglement for the five datasets (cf. Table I), by considering only two factors of variation. The 3dShapes, dSprites, and LPC datasets contain more than two factors, so we composed them, as noted in the **S** column in Table B.1 to attain only the content-motion factors. As stated in Section IV-A, when the number of factors is not equal to the number of units (in our case, the number of units

[4] https://github.com/deepmind/dsprites-dataset as September, 2019.

[5] https://github.com/deepmind/3d-shapes as September, 2019.

[6] https://github.com/rtqichen/beta-tcvae as December, 2019.

[7] https://github.com/edenton/svg as May, 2020.

[8] https://github.com/mazzzystar/Disentangled-Sequential-Autoencoder as December, 2019.

is significantly higher than 2), the MIG and SAP metrics are expected to be low.

Although MIG is a relatively popular metric, it penalizes dispersed representations, by considering the information gap between the first and second units that best represent a factor. Thus, when one factor of variation is equally represented by more than one unit, that gap is expected to be low, and so does the metric.

SAP is also thought to be low when there is a mismatch between the number of units and the number of factors since this metric is based on the classification accuracy estimation (using a Linear SVM classifier) when each 1d unit is used to classify examples under each factor.

We consider the FVAE metric to be the most suitable for the objective of motion disentanglement since it only penalizes the undesirable case in which one latent unit represents more than one factor of variation, and we are only considering two factors that we expect to be fully disentangled.

For the reconstructions metrics, in theory, we can generate $n(n-1)$ reenacted videos, where n is the number of videos in the test set. It was straightforward to generate 10 000 reenacted videos for all datasets, except for CK+, which had approximately 4000 videos. We used all the generated videos to calculate SSIM and FID.

Appendix C
Performance of Training with Partial Representations

Aiming at reducing the computational cost of training MYC-VAE when O is high, without reducing its performance, we conducted an experiment to assess the effect of subsampling the number of combinations to calculate the extended log-likelihood (3) and the Blind Reenactment Loss (6). We set $O = 4$, but instead of reconstructing O times the input sequence, we reconstruct only two, by randomly sampling two of the O appearance representations. Notice that, if we sampled only one appearance representation, the BRL calculation would not be possible (see Eq. 6). Due to time and computer restrictions, we performed those experiments only in the MUG dataset.

Table C.1 shows the results of our experiments. These results suggest that a full representation may increase modularity (higher FVAE), while a partial representation seems to deal better with explicitness (higher SAP) and reconstruction quality (SSIM and FID). However, the difference between those methods is not big enough to say one is better than the other, reinforcing our hypothesis that higher orders may just add too much redundancy to the training, without improving performance. This in part may explain why our experiments showed that optimal values of O are 2 or 3 in terms of cost/benefit, even for long sequences like the ones in MUG. We also noticed that using the partial representations yielded smaller architectures (about 25% less trainable parameters), less GPU memory (about 50%) and a lower execution time (about 50%).

Appendix D
Experiments on High-Resolution MUG

We tested the effectiveness of MTC-VAE on high-resolution inputs by training it on a 256×256 version of the MUG dataset, which we will call it as MUG-HQ.

The architecture to process the 256×256 input contains two more convolutional layers in the encoders and the decoder than the 64×64 version. Also, we doubled the size of the content and motion latent representations, and trained our model for 60 hours, while the 64×64 model took 24 hours to converge.

In table D.1 we compare the performance of MTC-VAE between MUG and MUG-HQ, in order to better analyze how the model was affected with a high-resolution input. The SSIM and FID metrics behaved as expected: the larger the input, the harder to reconstruct it, and the harder to yielding samples that belong to the data distribution. The performance on the downstream classification task was practically unaffected in content classification, while it had a slight drop in Motion classification.

In general, we expected all the metrics to worsen for high-resolution videos. For that reason, the increase on the FVAE and SAP metrics is somehow surprising for us. Our conclusion is that the increase in spatial resolution enhanced the quality of the representations, in particular, the content one, by providing mode discriminant information. On the other side, the motion representation presented a lower action classification performance, suggesting that it contains less discriminating information.

Figures D.1 to D.8 show some examples of how successful was the reenactment task in yielding realistic videos with accurate poses.

Appendix E
Experiments on the Tai Chi Dataset

The Tai Chi dataset consists of 1191 sequences downloaded from YouTube of several Tai Chi movements in diverse scenarios. The videos are cropped and aligned, in such a way that the character occupies the most of the frame an remains in the center. We performed a set of experiments on this dataset, in order to test the limitations of MTC-VAE in reconstructing high-complexity real-world scenes.

That being said, we expect the performance of MTC-VAE to fall behind VR-SotA models [28–36, 56, 57, 59, 60], given that such models rely on high-dimensional structured representations that preserve spatial information, while our model, aiming at providing a meaningful and disentangled low-dimensional representation, has an important disadvantage, as it cannot preserve spatial information so accurately.

Table E.1 shows the comparison of the performance of MTC-VAE w.r.t. the baselines. Given that the ground truth of the dataset only provide identity (i.e., content) labels, it is not possible to calculate the disentanglement metrics (FVAE, MIG and SAP). Hence, we only report the SSIM and FID metrics, besides the accuracy on content classification. Our model outperforms the others in realism (FID) and loses to β-TCVAE on structural similarity (SSIM). Finally, the features

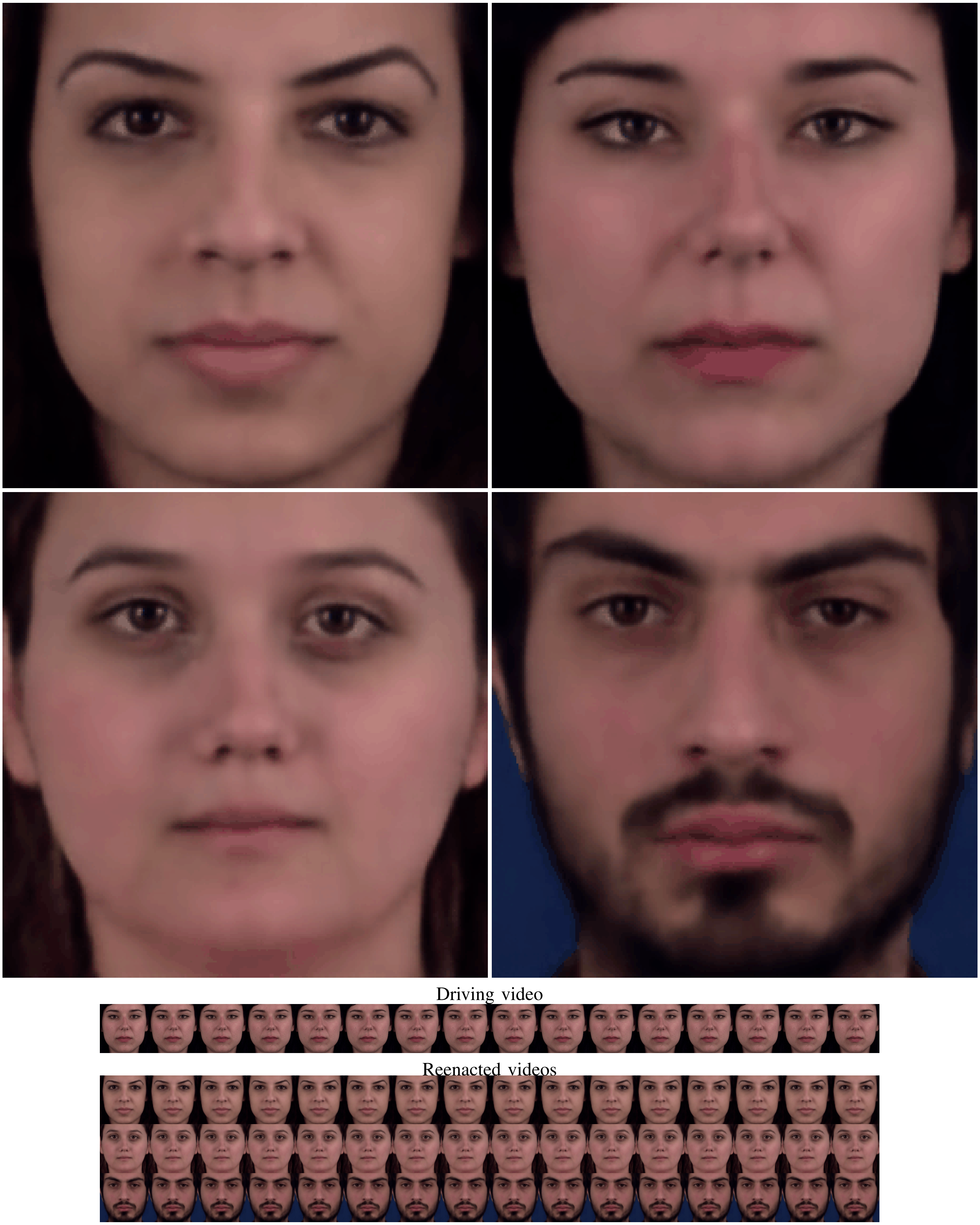


Fig. D.1. MUG-HQ: Reenactment examples. Above: selected frames at full resolution. Below: complete sequences.

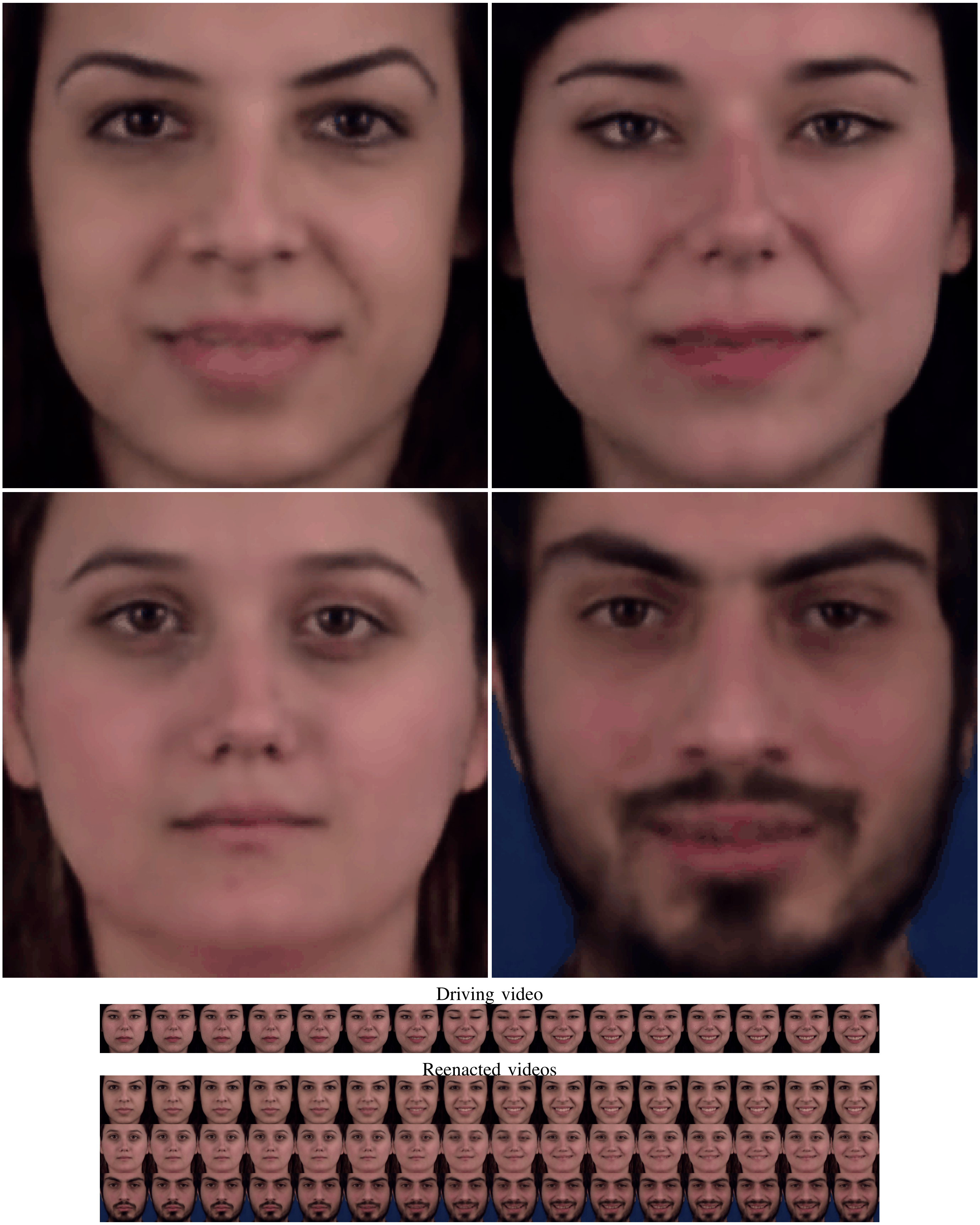


Fig. D.2. MUG-HQ: Reenactment examples. Above: selected frames at full resolution. Below: complete sequences.

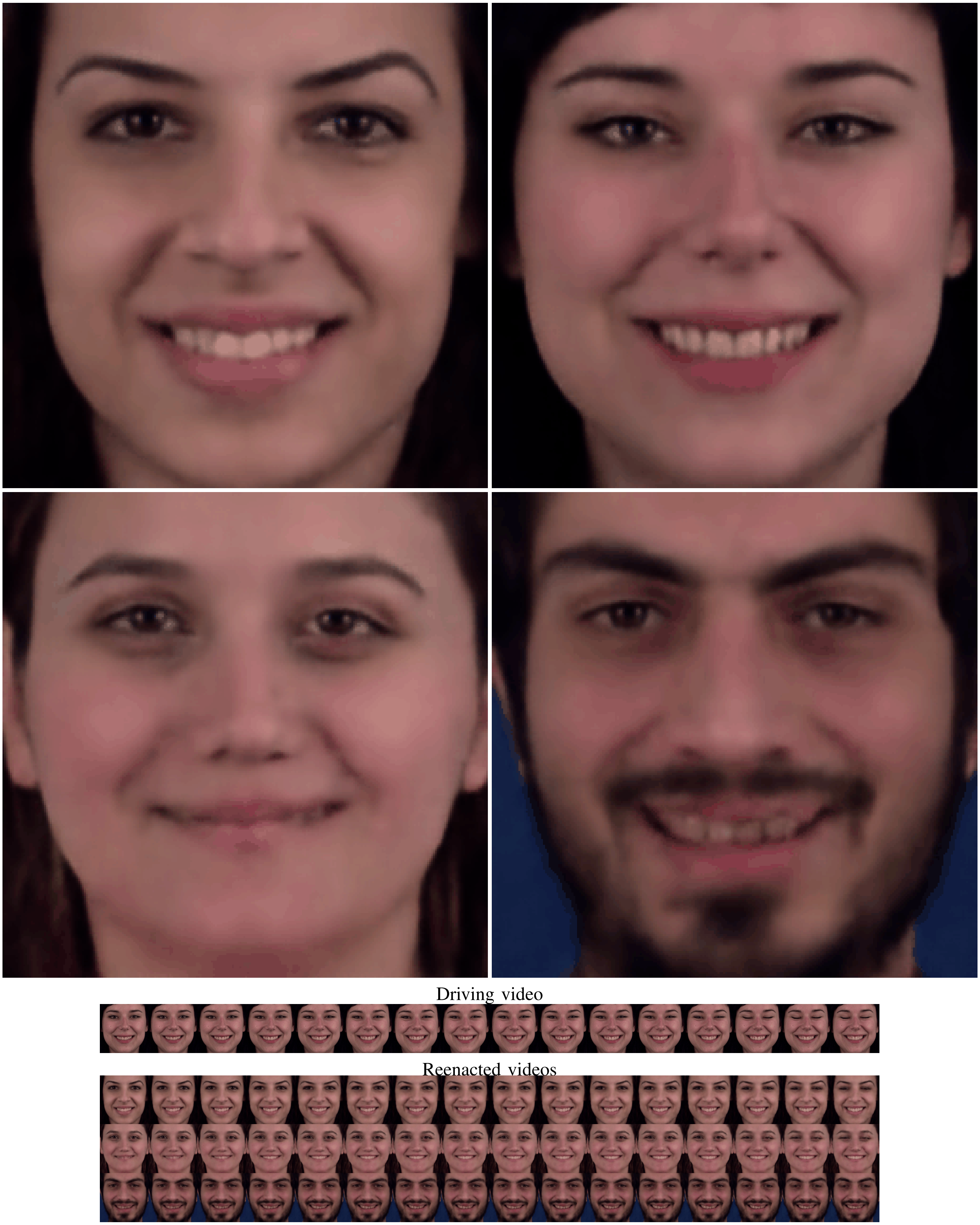


Fig. D.3. MUG-HQ: Reenactment examples. Above: selected frames at full resolution. Below: complete sequences.

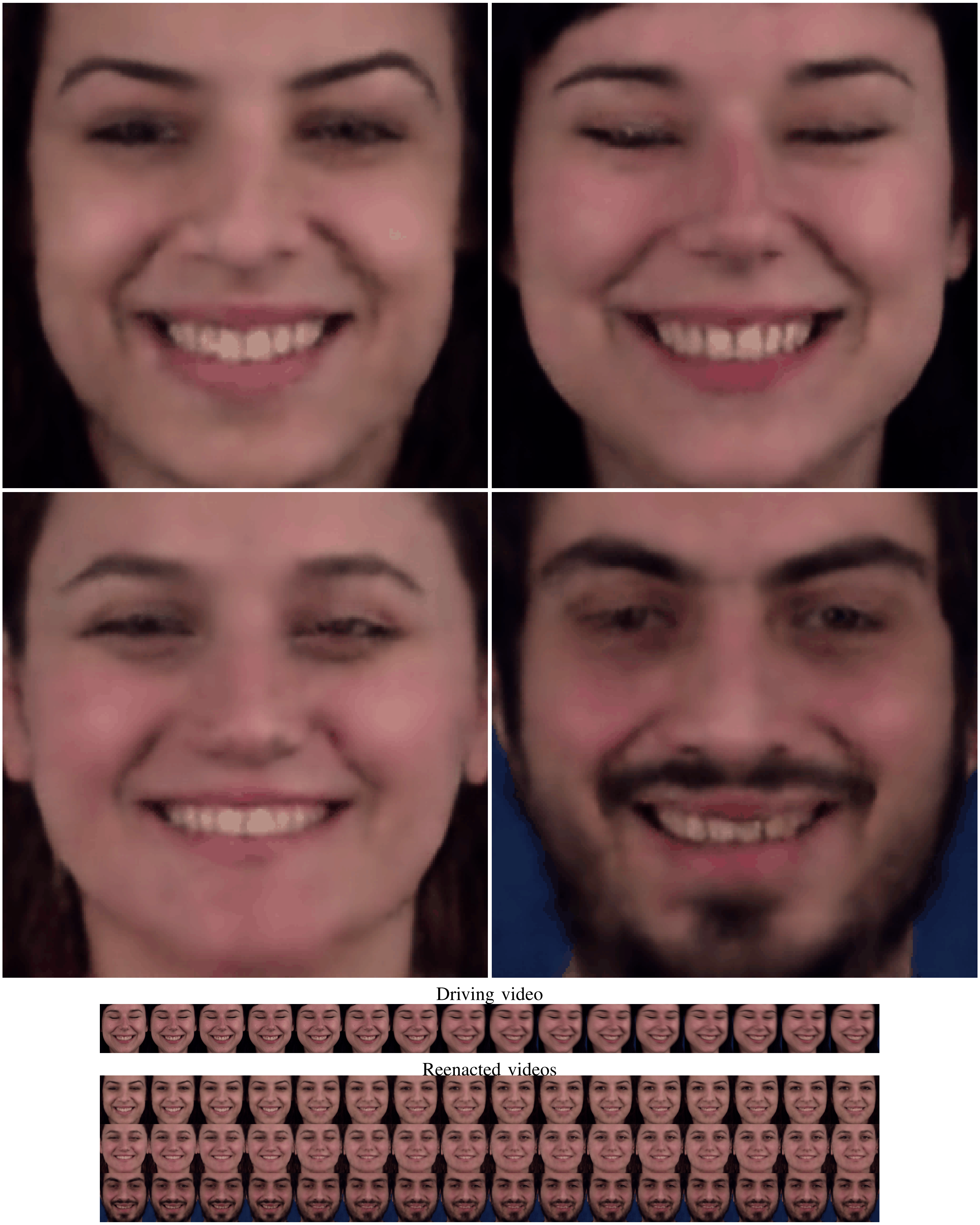


Fig. D.4. MUG-HQ: Reenactment examples. Above: selected frames at full resolution. Below: complete sequences.

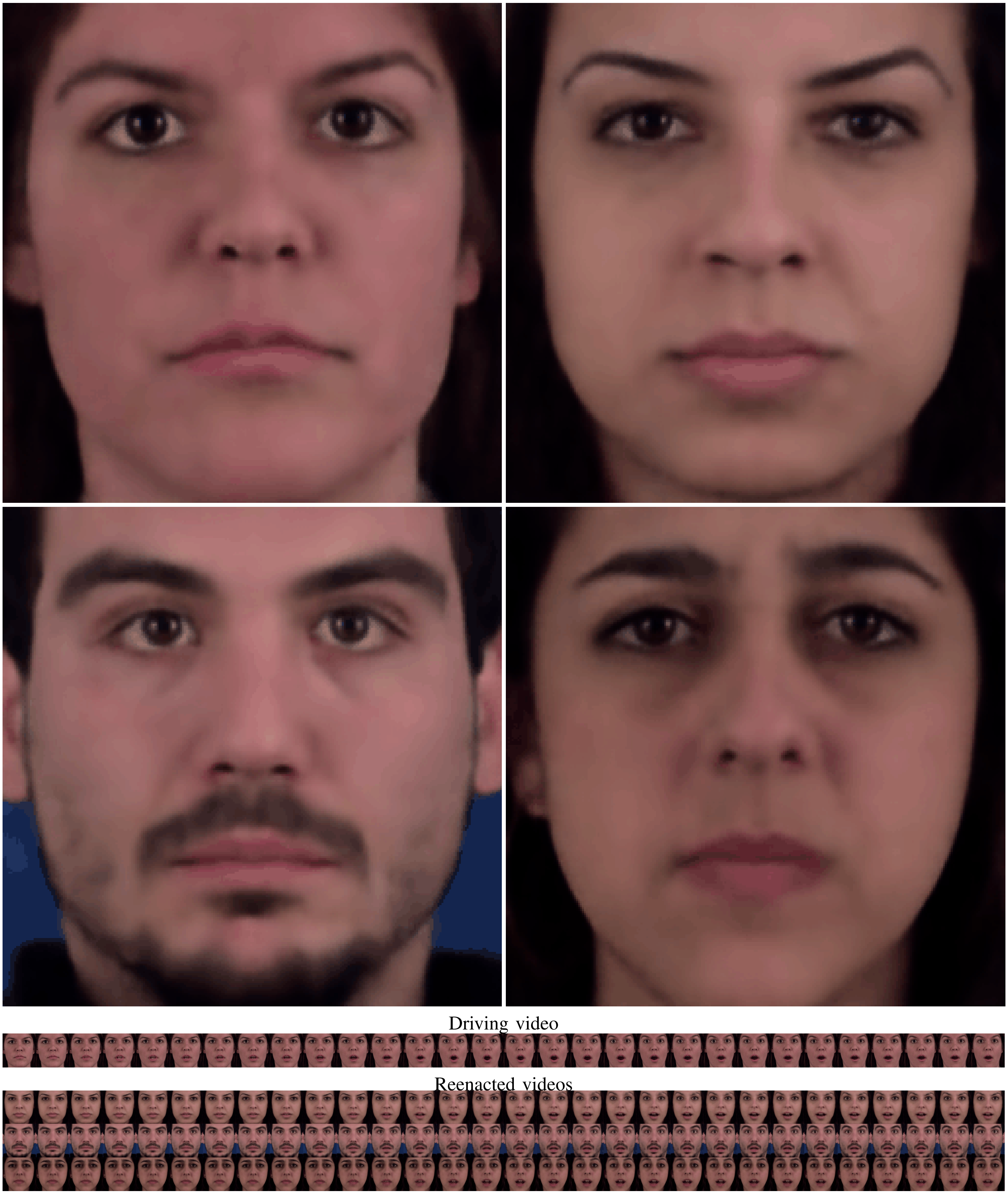


Fig. D.5. MUG-HQ: Reenactment examples. Above: selected frames at full resolution. Below: complete sequences.

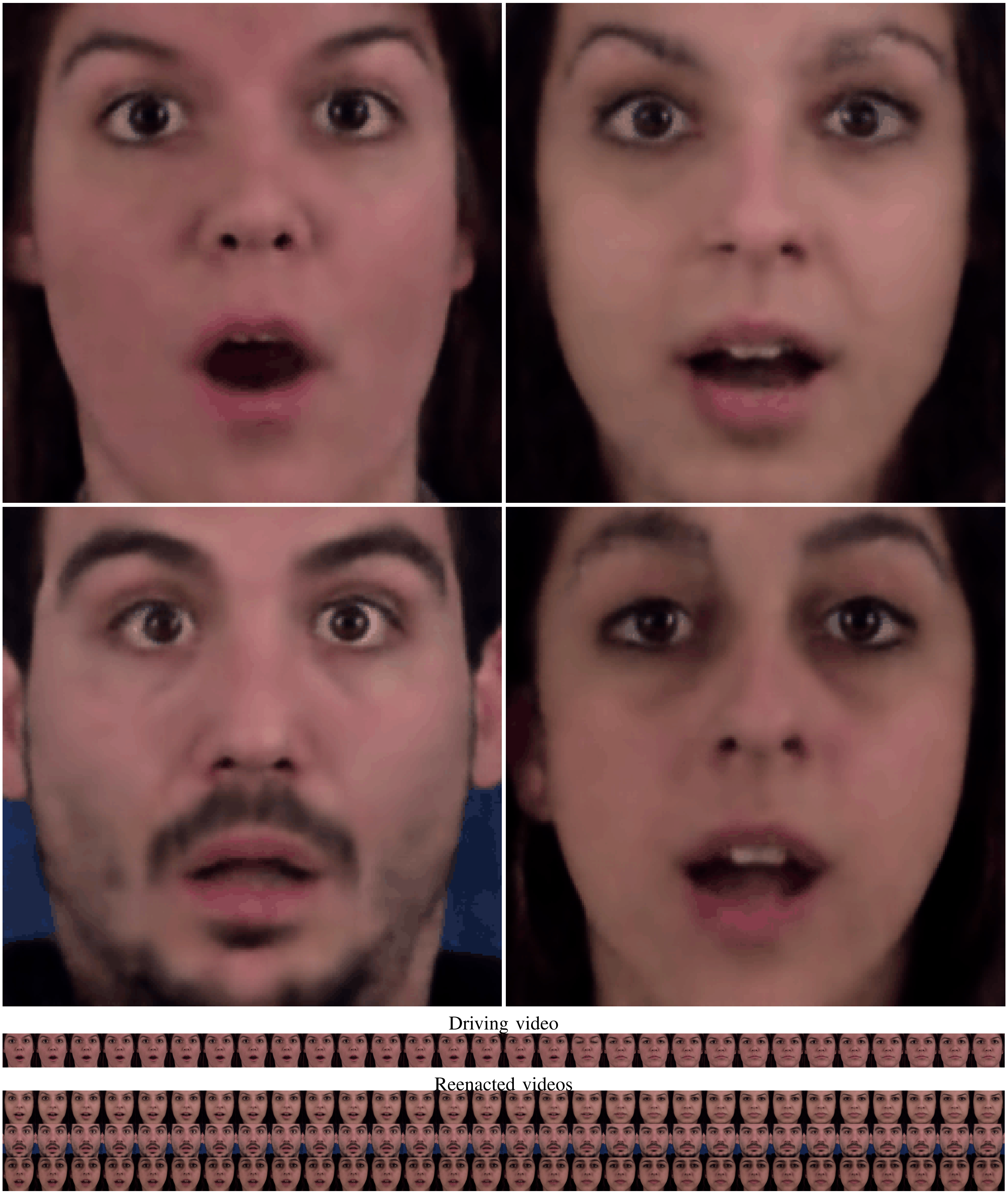


Fig. D.6. MUG-HQ: Reenactment examples. Above: selected frames at full resolution. Below: complete sequences.

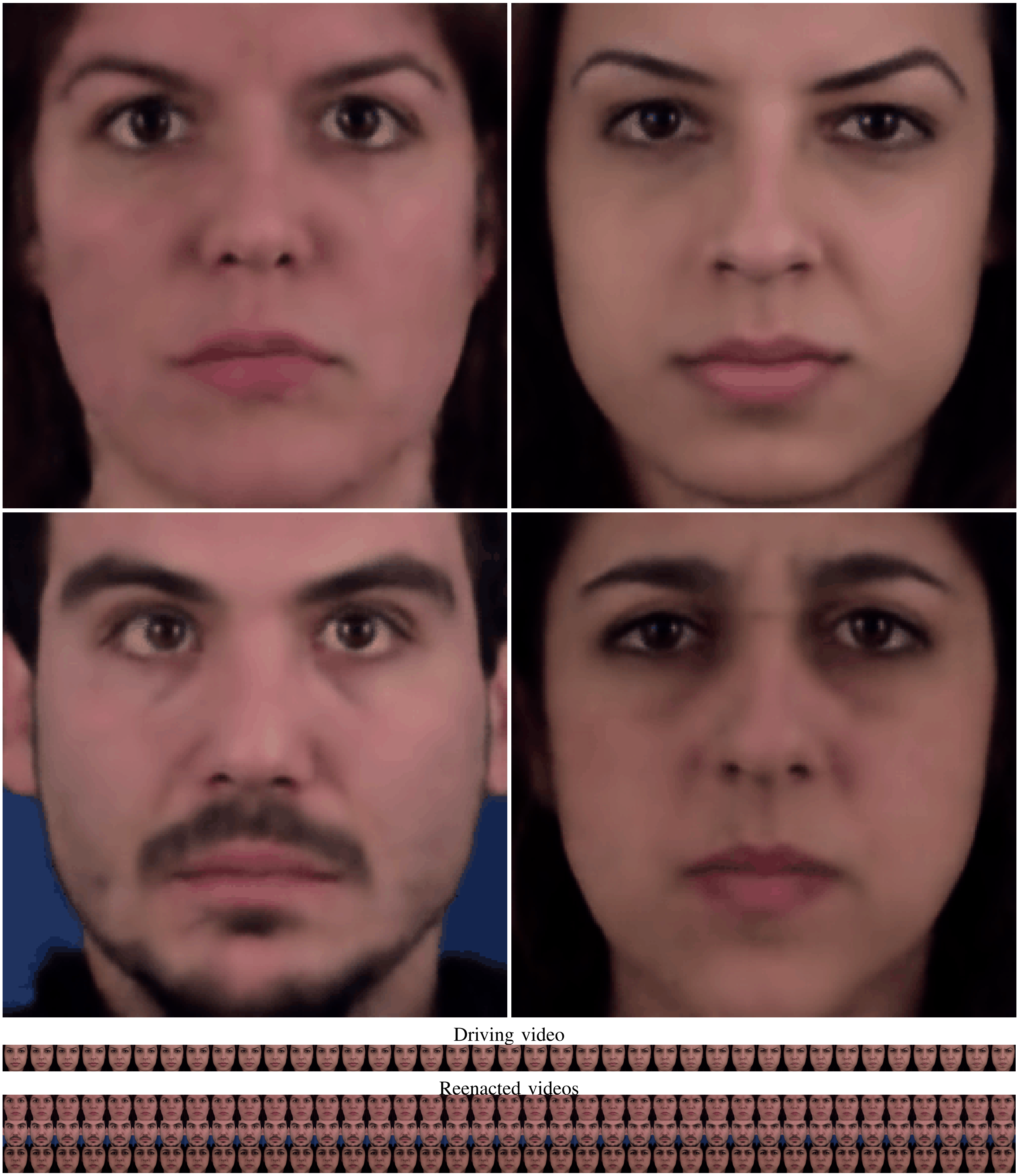


Fig. D.7. MUG-HQ: Reenactment examples. Above: selected frames at full resolution. Below: complete sequences.

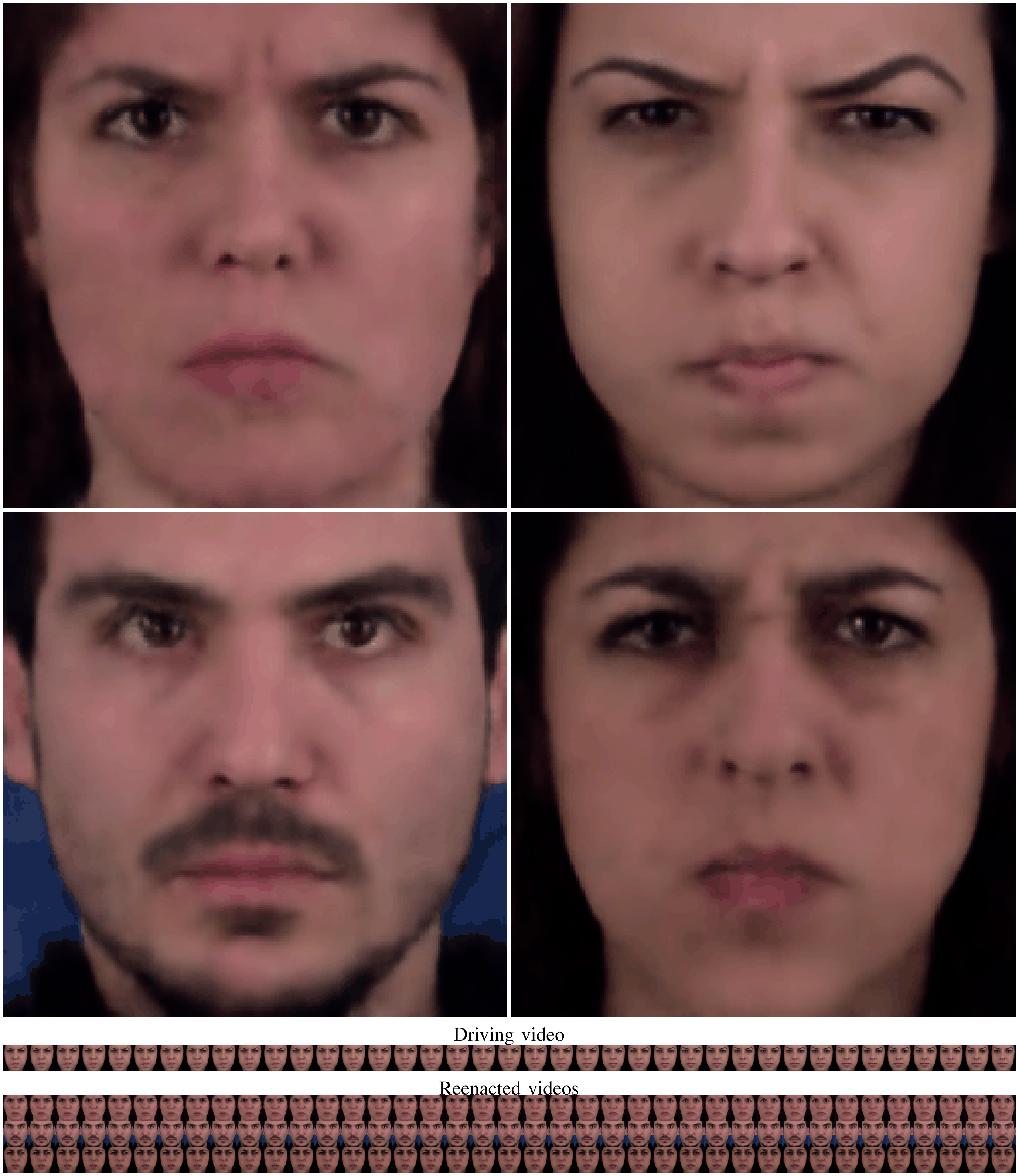


Fig. D.8. MUG-HQ: Reenactment examples. Above: selected frames at full resolution. Below: complete sequences.

TABLE C.1. Comparison of the model performances with and without subsampling when $O = 4$ in the MUG dataset.

	Disentanglement			Reconstruction		Accuracy	
	FVAE ↑	MIG ↑	SAP ↑	SSIM ↑	FID ↓	C ↑	M ↑
Full	**0.78±0.04**	**0.01±0.01**	0.82±0.04	0.45±0.01	39.44±3.55	1.0	**0.40±0.09**
Partial	0.77±0.06	0.00	**0.87±0.02**	**0.47±0.01**	**40.40±2.05**	1.0	0.39±0.04

TABLE D.1. Results on MUG-HQ compared with its low-quality version. The lower part indicates the performance on downstream tasks.

Metrics	Values HQ	Values LQ
FVAE ↑	**0.77±0.04**	0.72±0.04
MIG ↑	**0.02±0.01**	0.01±0.01
SAP ↑	**0.83±0.03**	0.73±0.05
SSIM ↑	0.61±0.01	**0.63±0.02**
FID ↓	41.12±1.07	**28.79±1.15**
Content ↑	0.99±0.01	**1.00**
Motion ↑	0.63±0.04	**0.79±0.05**

TABLE E.1. Performance for content-motion disentanglement and data realism in the Tai Chai dataset

	SSIM ↑	FID ↓	Content Class. Acc. ↑
β-TCVAE	**0.81±0.04**	244.77±3.85	0.93±0.02
dis-VAE	0.73±0.05	215.51±2.41	0.88±0.04
SVG-LP	0.69±0.07	201.82±0.91	0.48±0.04
MTC-VAE	0.78±0.02	**183.24±1.17**	**0.98±0.02**

yielded by MTC-VAE significantly outperforms the baselines' when used to classify the identity of the character.

Figures E.1 to E.12 show examples of VR by MTC-VAE, and confirm our expectation of our model not being competitive when compared to SotA methods, due to the reasons presented above. It is important to mention that, besides the complexity of the motions in the video, the highly heterogeneous backgrounds significantly hinders the reconstruction task.

However, Figures E.1 to E.12 allow us to qualitatively assess the disentanglement performance of our model. Notice how the appearance is preserved in each row of the matrices of images, while the only trait that changes is the instantaneous pose (i.e., motion). Although blurry, it is possible to see that the overall deformation of the body to yield a pose is, at some extent, correctly transferred w.r.t. the driving video, and that the identity of the character as well as the background (i.e., content) is preserved, meaning that both the content and motion representations have the correct meaningful information to reconstruct the video, and the bottleneck in the reconstruction process is in the decoder.

Solutions to handle this problem include explicitly modeling the background (i.e., having identity, motion, and background representations), and using deformations modules based on Spatial Transformer Networks [37]. Such solutions are considered as promising future work, but outside of the scope of our proposal in this manuscript.

Appendix F
Detailed Quantitative Results

We present the performance of MTC-VAE and the baselines for each the soft generalization and the two hard generalization scenarios, w.r.t. the three disentanglement metrics introduced

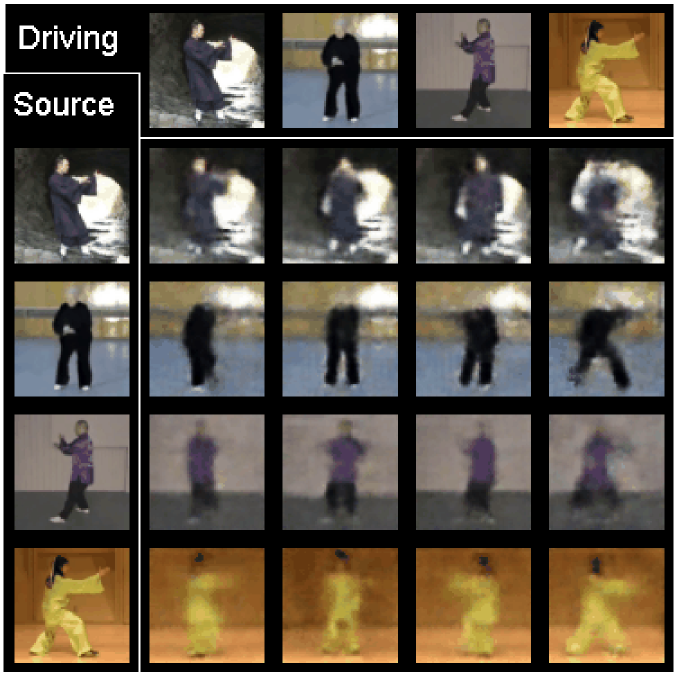


Fig. E.1. Tai Chi: Reenactment Examples.

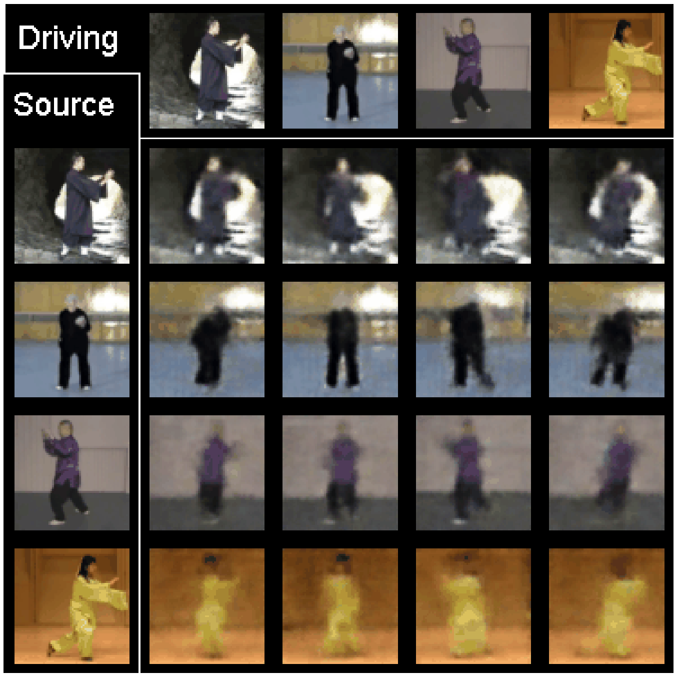


Fig. E.2. Tai Chi: Reenactment Examples.

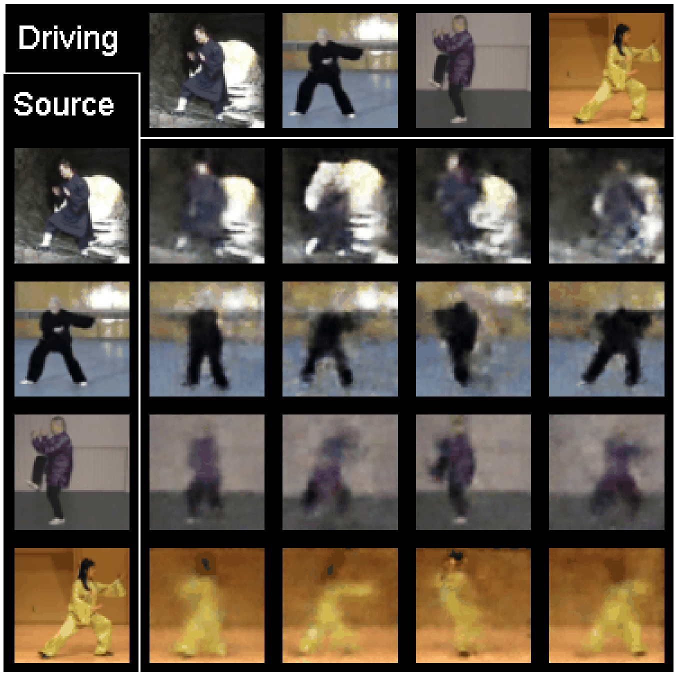


Fig. E.3. Tai Chi: Reenactment Examples.

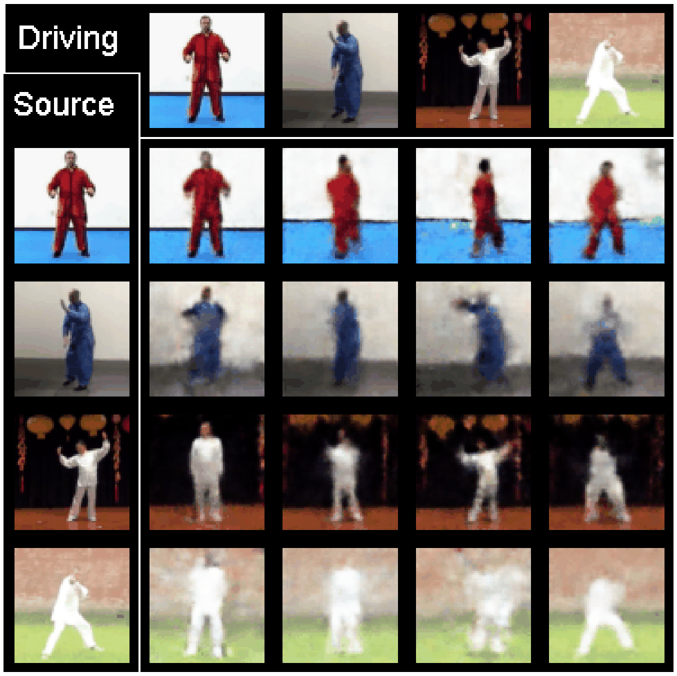


Fig. E.5. Tai Chi: Reenactment Examples.

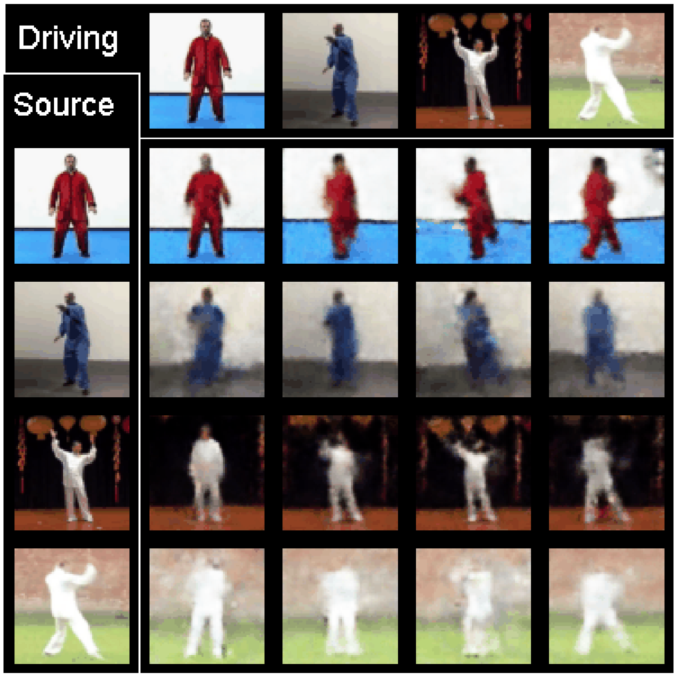


Fig. E.4. Tai Chi: Reenactment Examples.

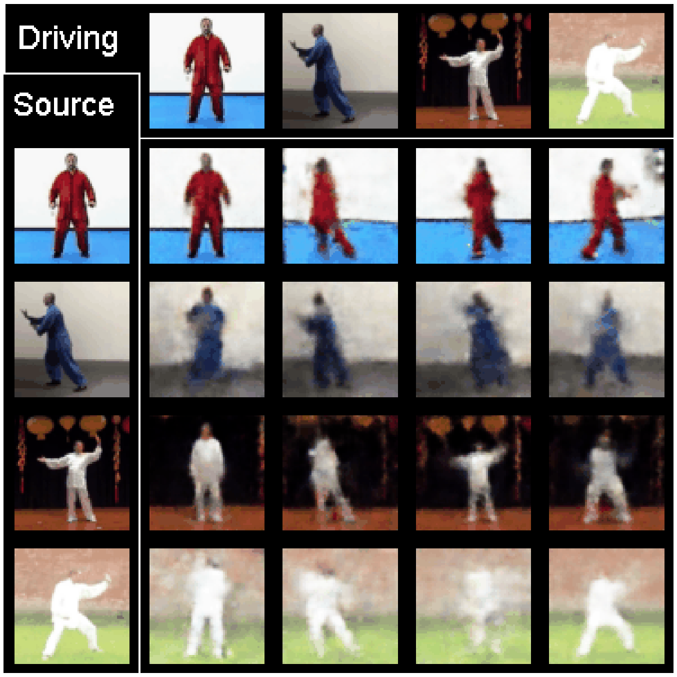


Fig. E.6. Tai Chi: Reenactment Examples.

in Section IV-A and the two reconstruction metrics introduced in Section IV-B.

Tables F.1, F.2, and F.3 shows the performance in the soft generalization scenario for our ablation studies presented in the main text. In particular, Tables F.1 and F.2 have the same data as, respectively, in Figs. 4 and 5. The discussion on these results is provided in Section IV-C.

Table F.3 shows that performance on disentanglement depends on the dataset, and it can be related to the length of the videos. E.g., MMNIST, the dataset with the longest videos, presented better disentanglement performance at the higher orders ($O = 4$), while the rest showed better performance in middle-sized orders ($O = 2, 3$). On the other hand, for reconstruction, it seems that the best performance was obtained, in general, for $O = 1$. It is important to point that the memory and time required to train the model significantly increase as O grows. We consider that having high-order models is not optimal in terms of cost-benefit. Also, order 1 may achieve better reconstruction results, but present poorer disentanglement results. Optimal values of O can be 2 or 3.

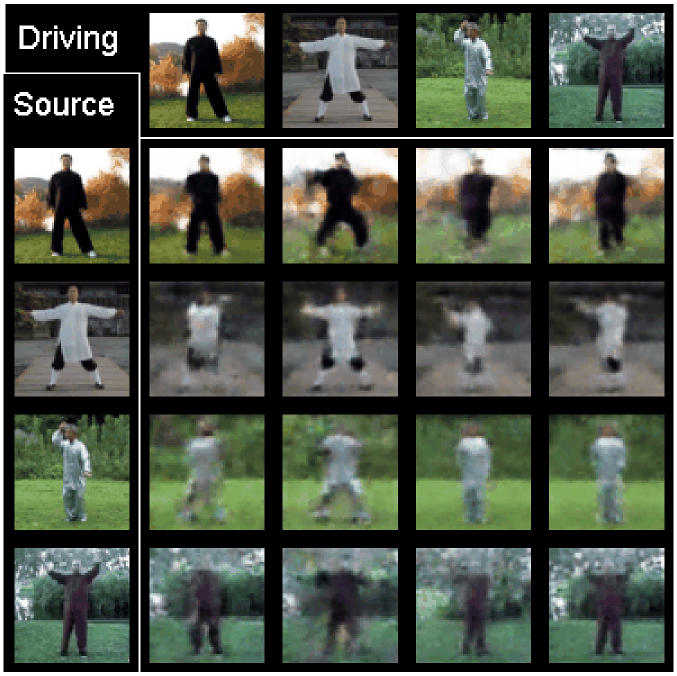


Fig. E.7. Tai Chi: Reenactment Examples.

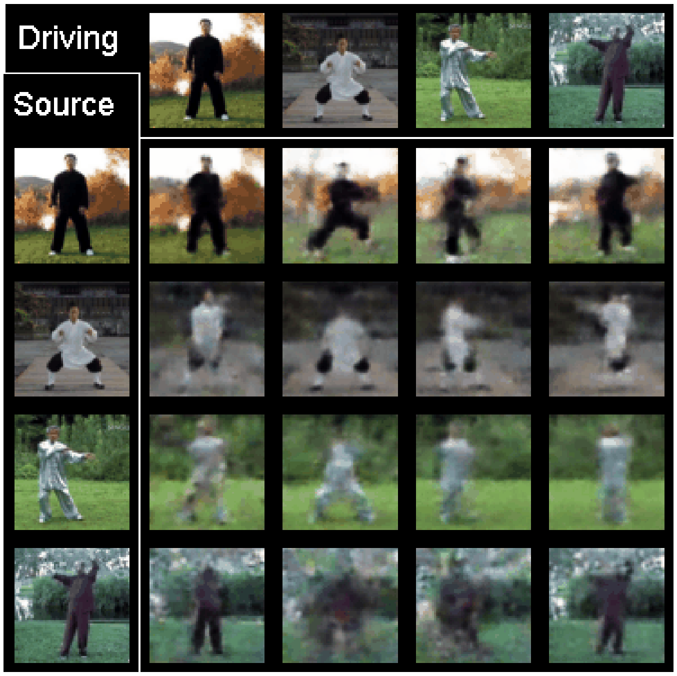


Fig. E.8. Tai Chi: Reenactment Examples.

Table F.6 presents the comparison of MTC-VAE and the baselines on the two hard generalization scenarios. In general, the dominance of MTC-VAE over the baselines persists in both scenarios.

Finally, Table F.5 complements Table II, by showing the multi-factor disentanglement performance for the hard generalization scenarios. The superiority of MTC-VAE, either in its frame and chunk version, is easily spotted.

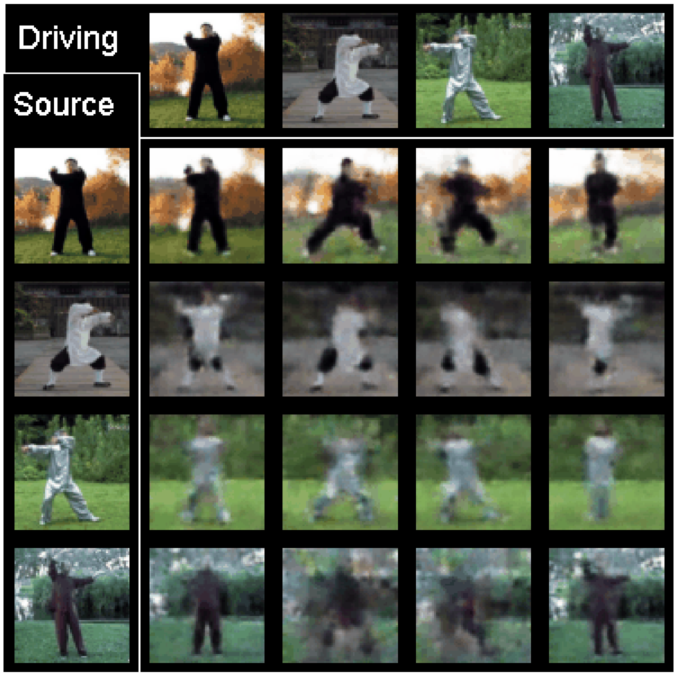


Fig. E.9. Tai Chi: Reenactment Examples.

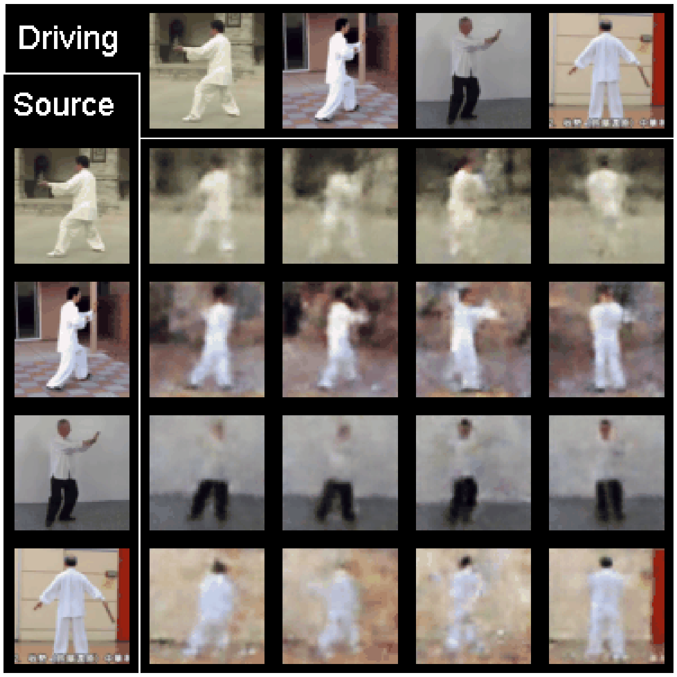


Fig. E.10. Tai Chi: Reenactment Examples.

Appendix G
Detailed Qualitative Results

In this section, we present traversal examples for LPC (Figs. G.1, G.2, G.3, G.4, G.5, G.6, G.7 and G.8), reenactment examples for LPC (Figs. G.9), 3dShapes (Figs. G.10, G.11, and G.12), dSprites (Figs. G.13), CK+ (Figs. G.14, G.15, and G.16), and MMNIST (Figs. G.18, G.19, and G.20). Besides the comparison with the baselines, we included examples for the ablation studies on the chunk size (c), impact of the Blind Reenactment Loss (λ), and the order of the model (O). Recall

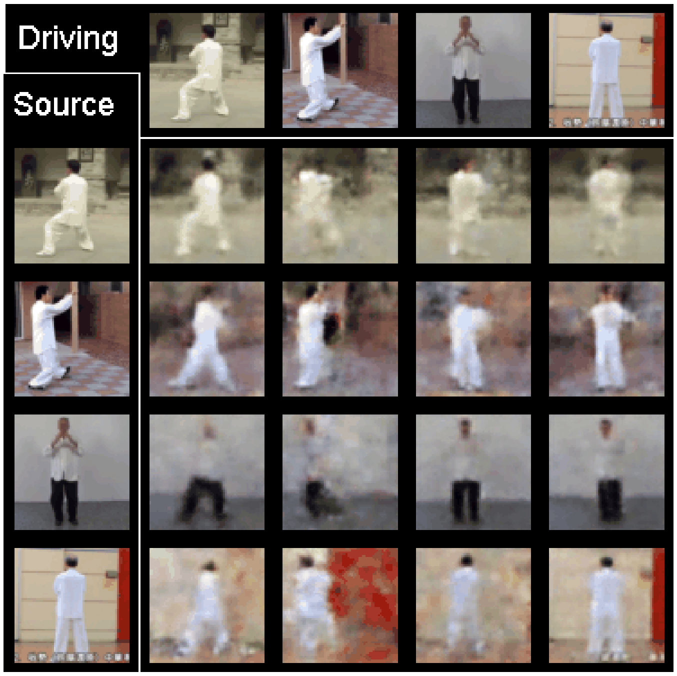


Fig. E.11. Tai Chi: Reenactment Examples.

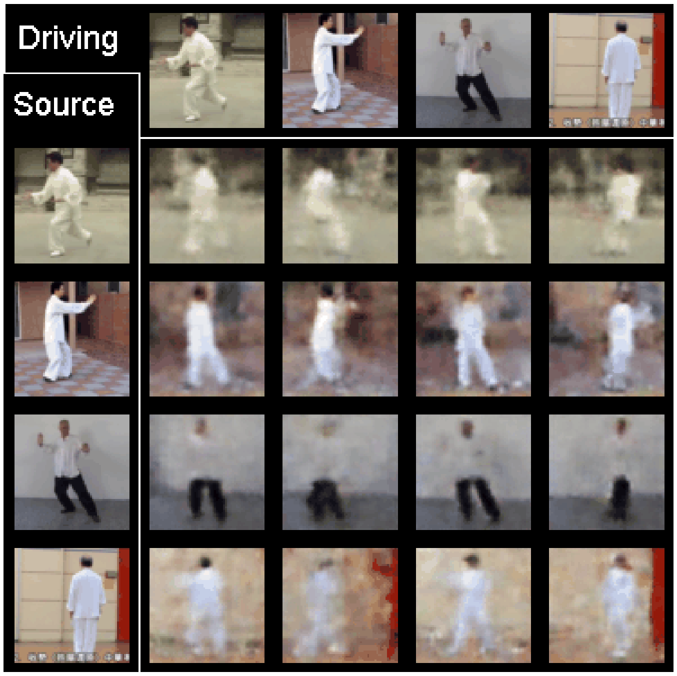


Fig. E.12. Tai Chi: Reenactment Examples.

TABLE F.1. Ablation on the chunk size.

	c	FVAE ↑	MIG ↑	SAP ↑	SSIM ↑	FID ↓
3dShapes	1	0.50±0.01	0.01	0.39±0.11	**0.73±0.02**	**100.80±46.82**
	3	**0.51±0.01**	0.01	**0.41±0.10**	**0.73±0.03**	114.23±47.63
	5	0.50±0.02	0.01	**0.41±0.10**	0.67±0.06	119.47±51.00
	7	0.48	0.01	0.32±0.09	0.62±0.07	117.88±56.90
	9	**0.51**	0.01	0.33±0.07	0.58±0.08	124.43±57.12
CK+	1	0.85±0.02	**0.03±0.01**	0.05±0.02	**0.68±0.13**	76.16±19.37
	3	**0.87±0.03**	**0.03±0.01**	0.11±0.03	0.67±0.12	70.56±19.92
	5	0.86±0.04	0.02±0.01	**0.13±0.04**	0.66±0.12	63.13±22.50
	7	0.85±0.04	0.02±0.01	0.12±0.05	0.62±0.10	**63.03±18.15**
	9	0.81±0.05	**0.03±0.01**	0.11±0.03	0.61±0.10	60.56±17.75
dSprites	1	**0.92±0.01**	0.02±0.02	0.01	0.77±0.01	105.79±5.86
	3	0.89±0.03	0.02±0.02	**0.06±0.05**	**0.80±0.04**	78.40±23.19
	5	0.82±0.01	**0.03±0.01**	**0.06±0.01**	0.78	**68.48±3.78**
	7	0.79±0.03	0.02±0.01	0.05±0.01	0.78	71.17±3.45
	9	0.76±0.01	**0.03±0.01**	0.10±0.01	0.78	72.77±2.82
LPC	1	0.86±0.01	0.00	0.11±0.03	**0.67±0.01**	**42.59±4.09**
	3	0.88±0.04	**0.02±0.06**	**0.24±0.23**	**0.67±0.01**	47.40±8.02
	5	0.87±0.01	0.00	0.17±0.01	0.66±0.01	84.87±7.81
	7	0.87±0.01	0.00	0.16±0.02	**0.67±0.01**	76.92±8.12
	9	**0.89±0.01**	0.01±0.01	0.19±0.03	**0.67±0.01**	53.57±5.01
MMNIST	1	0.91±0.04	0.09±0.05	0.09±0.04	**0.69±0.01**	186.25±23.55
	3	**0.95±0.02**	0.08±0.04	0.09±0.05	0.68±0.01	96.04±11.28
	5	**0.95±0.04**	0.07±0.02	0.10±0.04	0.68±0.01	102.11±9.89
	7	0.92±0.06	0.06±0.02	0.09±0.03	0.68±0.01	**95.69±13.54**
	9	0.91±0.05	**0.10±0.06**	**0.12±0.06**	0.68±0.01	116.05±13.38
MUG	1	0.70±0.09	**0.04±0.02**	0.75±0.08	**0.66±0.06**	43.86±13.15
	3	**0.73±0.02**	0.03±0.02	0.70±0.04	0.62±0.02	37.62±1.31
	5	0.72±0.03	0.02±0.01	**0.76±0.10**	0.63±0.02	**31.18±3.07**
	7	0.72±0.05	0.01±0.01	0.72±0.03	0.62±0.02	37.36±0.74
	9	0.72±0.05	0.02±0.01	0.70±0.04	0.61±0.02	37.56±1.06

TABLE F.2. Ablation on Blind Reenactment Loss.

	λ	FVAE ↑	MIG ↑	SAP ↑	SSIM ↑	FID ↓
3dShapes	0	0.50±0.02	0.01±0.01	0.35±0.15	0.67±0.06	120.08±53.40
	0.25	0.50±0.02	0.01±0.01	0.36±0.09	0.67±0.06	120.32±52.34
	0.5	0.50±0.02	0.01±0.01	0.39±0.18	0.67±0.06	**119.84±55.63**
	1	0.50±0.02	0.01	**0.41±0.14**	0.67±0.06	**119.47±51.00**
	2	0.50±0.02	0.01±0.01	**0.41±0.02**	0.67±0.06	**119.34±54.23**
	5	0.50±0.02	0.01±0.01	0.40±0.11	0.67±0.06	120.15±51.94
CK+	0	0.80±0.03	0.02±0.01	0.11±0.06	0.63±0.11	71.33±22.35
	0.25	0.81±0.03	0.02±0.01	0.11±0.06	0.64±0.14	67.73±22.37
	0.5	0.78±0.09	0.02±0.01	0.12±0.08	0.64±0.03	64.49±22.76
	1	**0.86±0.04**	0.02±0.01	0.13±0.05	**0.66±0.12**	63.13±22.50
	2	0.84±0.07	0.02±0.01	0.15±0.04	**0.66±0.10**	**60.26±22.42**
	5	0.80±0.04	0.02±0.01	**0.17±0.02**	0.64±0.37	65.57±22.84
dSprites	0	0.82±0.09	0.02±0.02	0.01	0.81	**65.83±7.82**
	0.25	0.82±0.06	0.02±0.02	0.01	0.81	69.85±6.32
	0.5	0.84±0.02	0.02±0.02	0.01	0.81	82.23±6.84
	1	**0.85±0.04**	**0.04±0.02**	0.01	**0.82**	85.43±9.26
	2	**0.85±0.05**	0.04±0.02	0.01	**0.82**	87.85±8.41
	5	0.83±0.02	0.01±0.01	0.01	**0.82**	90.15±7.56
LPC	0	0.88±0.03	0.01±0.01	**0.22±0.04**	0.67±0.01	55.81±15.59
	0.25	0.86±0.03	0.01±0.01	0.22±0.02	0.67±0.01	50.56±3.73
	0.5	0.85±0.08	**0.02±0.05**	0.22±0.01	0.67±0.01	45.27±9.54
	1	0.89±0.01	**0.02±0.05**	0.21±0.19	0.67±0.01	**40.05±4.26**
	2	0.87±0.01	0.01±0.01	0.22±0.04	0.67±0.01	39.51±5.42
	5	**0.90±0.02**	0.01±0.01	0.22±0.04	0.67±0.01	38.29±7.25
MMNIST	0	0.89±0.04	0.05±0.03	0.08±0.04	0.68±0.01	**100.61±13.03**
	0.25	0.92±0.07	0.05±0.05	0.09±0.07	0.68±0.01	101.11±9.63
	0.5	0.93±0.05	**0.07±0.02**	0.09±0.02	0.68±0.01	100.97±9.92
	1	**0.95±0.04**	**0.07±0.02**	**0.10±0.05**	0.68±0.01	102.11±9.89
	2	0.94±0.02	**0.07±0.01**	0.09±0.01	0.68±0.01	102.34±9.75
	5	0.94±0.01	**0.07±0.02**	0.09±0.01	0.68±0.01	102.82±9.51
MUG	0	0.72±0.05	0.01±0.01	0.74±0.04	0.62±0.02	34.86±6.32
	0.25	0.72±0.05	0.01±0.01	**0.76±0.02**	0.62±0.02	27.03±0.55
	0.5	0.72±0.04	0.01±0.01	**0.76±0.05**	0.62±0.01	**24.14±0.72**
	1	0.72±0.04	**0.02±0.02**	0.74±0.07	**0.63±0.03**	29.68±0.76
	2	**0.75±0.07**	**0.02±0.01**	0.73±0.05	0.62±0.02	30.46±1.13
	5	0.74±0.04	**0.02±0.01**	0.72±0.02	0.62±0.01	32.21±0.98

that the default configuration for MTC-VAE (fifth line in each figure) is $c = 5$, $\lambda = 1$, and $O = 2$.

References

[1] Francesco Locatello et al. "A Commentary on the Unsupervised Learning of Disentangled Representations". In: *AAAI Conf. Artif. Intell. (AAAI)*. 2020.

[2] Peihao Chen et al. "Generating Visually Aligned Sound From Videos". In: *IEEE Trans. Image Process.* 29 (2020), pp. 8292–8302.

[3] Aditya Ramesh et al. "Zero-Shot Text-to-Image Generation". In: *Inter. Conf. Mach. Learn. (ICML)*. 2021.

[4] Francesco Locatello et al. "Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations". In: *Inter. Conf. Mach. Learn. (ICML)*. 2019.

[5] Rui Shu et al. "Weakly Supervised Disentanglement with Guarantees". In: *Inter. Conf. Learn. Represent. (ICLR)*. 2020.

[6] Joao Carreira and Andrew Zisserman. "Quo Vadis, Action Recognition? A New Model and the Kinetics

TABLE F.3. Ablation on the order of the model.

	O	FVAE ↑	MIG ↑	SAP ↑	SSIM ↑	FID ↓
3dShapes	1	0.50±0.02	0.01	0.30±0.14	0.64±0.05	**104.14±38.20**
	2	0.50±0.02	0.01	0.41±0.14	**0.67±0.06**	119.47±51.00
	3	0.50±0.02	0.01	**0.46±0.12**	**0.67±0.06**	123.91±52.84
	4	0.50±0.02	0.01	0.37±0.12	**0.67±0.06**	127.32±51.50
CK+	1	0.80±0.04	0.02±0.01	0.05±0.03	**0.69±0.12**	65.79±24.73
	2	0.86±0.04	0.02±0.01	0.13±0.05	0.66±0.12	**63.13±22.50**
	3	**0.87±0.03**	0.02±0.01	0.13±0.04	0.64±0.12	67.57±19.31
	4	0.86±0.04	0.02±0.01	0.06±0.03	0.68±0.13	65.02±24.44
dSprites	1	0.86±0.02	0.00	**0.12±0.02**	**0.87±0.01**	**66.29±3.00**
	2	0.85±0.06	**0.03±0.01**	0.06±0.03	0.78±0.01	74.30±15.84
	3	0.89±0.01	0.02±0.01	0.02±0.01	0.78	81.29±8.84
	4	**0.91±0.04**	**0.03±0.03**	0.00±0.01	0.78±0.01	116.81±19.72
LPC	1	0.87±0.01	0.00	0.19±0.03	0.67±0.01	**44.70±4.00**
	2	**0.88±0.03**	**0.02±0.05**	**0.21±0.22**	0.67±0.01	54.85±18.06
	3	0.86±0.01	0.00	0.20±0.01	0.67±0.01	56.50±5.10
	4	0.86±0.02	0.01	0.19±0.01	**0.68±0.01**	57.14±4.70
MMNIST	1	0.80±0.02	0.02±0.02	0.03±0.03	**0.72±0.03**	**96.10±23.98**
	2	0.95±0.04	0.07±0.02	0.10±0.05	0.68±0.01	102.11±9.89
	3	**0.98±0.02**	0.07±0.03	0.10±0.06	0.68±0.01	100.41±11.68
	4	0.96±0.02	**0.10±0.05**	**0.11±0.07**	0.68±0.01	103.31±9.65
MUG	1	0.71±0.03	0.02±0.02	0.73±0.08	0.62±0.02	**29.30±0.67**
	2	0.72±0.05	0.02±0.02	0.72±0.06	0.62±0.03	38.69±6.05
	3	**0.73±0.04**	0.02±0.02	0.77±0.08	**0.64±0.03**	30.34±1.86
	4	0.70±0.02	0.02±0.01	**0.75±0.07**	0.63±0.02	**29.67±0.65**

TABLE F.4. Ablation on β.

	λ	FVAE ↑	MIG ↑	SAP ↑	SSIM ↑	FID ↓
3dShapes	1	0.50±0.03	0.03±0.02	0.20±0.02	0.59	68.57±10.95
	5	0.50±0.03	0.03±0.01	**0.25±0.02**	**0.60**	**58.01±4.58**
CK+	1	0.76±0.03	0.17±0.03	0.74±0.04	0.64±0.02	85.62±20.89
	5	**0.84±0.05**	**0.23±0.03**	**0.84±0.09**	**0.71±0.02**	**59.13±10.25**
dSprites	1	**0.91±0.02**	**0.04±0.01**	**0.10±0.01**	0.78	**57.18±6.43**
	5	0.87±0.02	0.02±0.01	0.06±0.01	0.78	70.41±5.21
LPC	1	**0.93±0.06**	**0.11±0.11**	**0.60±0.40**	0.67±0.01	**41.72±3.31**
	5	0.87±0.01	0.01±0.01	0.03±0.01	0.67±0.01	44.27±4.61
MMNIST	1	**0.96±0.05**	**0.28±0.05**	**0.86±0.01**	**0.68±0.01**	**103.59±5.57**
	5	0.56±0.03	0.04±0.03	0.17±0.11	0.67±0.01	150.93±4.36
MUG	1	0.72±0.04	0.01±0.01	0.73±0.05	0.63±0.02	**28.79±1.15**
	5	**0.74±0.03**	**0.03±0.02**	**0.85±0.05**	**0.66±0.02**	32.55±0.71

Dataset”. In: *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2017.

[7] Christoph Feichtenhofer et al. “SlowFast Networks for Video Recognition”. In: *IEEE Inter. Conf. Comput. Vis. (ICCV)*. 2019.

[8] Yaohui Wang et al. “G^{3}AN: Disentangling Appearance and Motion for Video Generation”. In: *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2020.

[9] Abhishek Aich et al. “Non-Adversarial Video Synthesis with Learned Priors”. In: *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2020.

TABLE F.5. Detailed results for the hard generalization scenarios in multiple factor disentanglement. Comparison between MTC-VAE (ours) and the baselines. (* $c = 1$)

		FVAE ↑	MIG ↑	SAP ↑	FVAE ↑	MIG ↑	SAP ↑
		Appearance Holdout			Motion Holdout		
3dShapes	β-TCVAE	0.23±0.04	0.08±0.05	0.03±0.02	0.20±0.02	0.07±0.05	0.03±0.02
	dis-VAE	0.19±0.01	0.03±0.02	0.01±0.01	0.19	0.03±0.01	0.01
	SVG	0.18	0.01	0.01	0.19±0.01	0.01	0.01
	MTC-VAE	0.26±0.02	**0.22±0.07**	**0.09±0.02**	0.27±0.06	**0.16±0.04**	**0.07±0.01**
	MTC-VAE*	**0.30±0.02**	0.14±0.06	0.05±0.03	**0.30±0.03**	0.14±0.04	0.06±0.02
dSprites	β-TCVAE	0.33±0.04	0.03±0.02	0.02±0.02	0.29±0.02	0.02±0.02	0.01
	dis-VAE	0.36±0.03	0.02±0.01	0.01	0.39±0.02	0.03	0.02
	SVG	0.25±0.01	0.00	0.01±0.01	0.25	0.00	0.00
	MTC-VAE	**0.52±0.01**	**0.10±0.01**	**0.08±0.01**	0.38±0.01	0.04±0.03	0.02±0.01
	MTC-VAE*	0.50±0.05	0.07±0.03	0.05±0.02	**0.40±0.02**	**0.06±0.01**	**0.05±0.02**
LPC	β-TCVAE	0.49±0.05	0.11±0.03	0.06±0.03	0.46±0.08	0.10±0.04	0.05±0.02
	dis-VAE	0.31±0.01	0.05±0.01	0.03±0.01	0.33±0.06	0.05±0.03	0.03±0.02
	SVG	0.24±0.01	0.01±0.01	0.01±0.01	0.23±0.01	0.01±0.01	0.01±0.01
	MTC-VAE	0.54±0.19	0.15±0.02	0.08±0.03	0.53±0.09	**0.20±0.04**	0.10±0.02
	MTC-VAE*	**0.63±0.04**	**0.19±0.02**	**0.10±0.02**	**0.65±0.01**	**0.20±0.05**	**0.12±0.04**

TABLE F.6. Detailed results for the hard generalization scenarios. Comparison between MTC-VAE (ours) and the baselines, evaluating disentanglement and reconstruction. (* $c = 1$)

		FVAE ↑	MIG ↑	SAP ↑	SSIM ↑	FID ↓
		Appearance Holdout				
3dShapes	β-TCVAE	**0.51±0.02**	**0.01±0.01**	0.11±0.12	0.54±0.14	183.40±35.13
	dis-VAE	0.50	0.00	0.08±0.07	0.38±0.04	**83.38±12.11**
	SVG	0.50±0.01	**0.01**	0.03±0.04	0.57±0.04	219.82±22.97
	MTC-VAE	**0.51±0.02**	**0.01±0.01**	**0.42±0.15**	0.70±0.04	174.30±23.13
	MTC-VAE*	0.49±0.01	**0.01**	0.32±0.10	**0.75±0.01**	141.65±18.33
CK	β-TCVAE	0.83±0.05	**0.03±0.02**	0.09±0.04	0.44±0.01	122.20±22.80
	dis-VAE	0.72±0.01	0.01	0.05±0.01	**0.54±0.02**	74.08±4.05
	SVG	0.69±0.06	0.02±0.01	0.04±0.03	0.02	**61.26±3.58**
	MTC-VAE	**0.86±0.03**	0.02±0.01	**0.16±0.06**	0.50±0.01	89.22±12.63
	MTC-VAE*	0.85±0.03	**0.03±0.01**	0.07±0.03	0.50±0.01	100.44±5.98
dSprites	β-TCVAE	0.63±0.05	0.04±0.04	0.00	**0.84±0.02**	96.93±9.74
	dis-VAE	0.72±0.04	0.01	0.00	0.80±0.01	123.95±0.82
	SVG	0.52±0.01	0.00	0.00	0.78±0.01	96.97±5.41
	MTC-VAE	**0.87±0.01**	**0.06±0.01**	**0.01±0.01**	0.81	**93.47±3.66**
	MTC-VAE*	0.78±0.02	0.03±0.01	0.01	0.80±0.01	117.57±5.05
LPC	β-TCVAE	0.96±0.02	**0.04±0.03**	**0.05±0.07**	0.59±0.03	106.71±10.06
	dis-VAE	0.89±0.04	0.02±0.01	0.01±0.01	0.47	109.21±5.34
	SVG	0.69±0.04	0.01±0.01	0.02±0.02	0.21	118.80±11.47
	MTC-VAE	**0.99**	0.02±0.01	0.04±0.02	**0.70±0.01**	151.00±9.25
	MTC-VAE*	0.98±0.02	0.02	0.04±0.03	0.55±0.01	**54.66±3.59**
MMNIST	β-TCVAE	0.63±0.05	0.03±0.03	0.04±0.04	**0.69±0.02**	163.29±11.78
	dis-VAE	0.63±0.03	0.03±0.01	0.04±0.01	0.68±0.01	143.25±13.62
	SVG	0.52±0.01	0.01	0.03±0.01	0.57±0.01	195.32±53.31
	MTC-VAE	**0.94±0.05**	0.08±0.03	**0.11±0.07**	0.68±0.01	**99.18±14.73**
	MTC-VAE*	0.85±0.09	**0.12±0.05**	0.11±0.06	0.65±0.05	212.73±22.81
		Motion Holdout				
3dShapes	β-TCVAE	0.50±0.02	**0.01±0.01**	0.11±0.08	0.52±0.12	148.64±43.91
	dis-VAE	0.50	0.00	0.06±0.01	0.40±0.01	**71.36±3.63**
	SVG	0.50	**0.01±0.01**	0.03±0.02	0.58±0.01	155.14±11.55
	MTC-VAE	0.50±0.02	**0.01±0.01**	0.36±0.18	0.72	126.11±3.50
	MTC-VAE*	0.50±0.01	**0.01±0.01**	**0.44±0.15**	**0.75±0.01**	120.44±13.19
CK	β-TCVAE	0.78±0.06	0.03±0.01	0.04±0.02	0.57±0.06	100.63±17.65
	dis-VAE	0.71±0.03	0.01±0.01	0.02±0.01	0.65±0.01	69.43±1.19
	SVG	0.71±0.08	0.02±0.01	0.04±0.03	0.02±0.01	**22.08±6.69**
	MTC-VAE	**0.89±0.02**	0.02±0.02	**0.08±0.01**	0.77±0.01	41.03±5.07
	MTC-VAE*	0.87±0.03	**0.04±0.01**	0.05±0.02	**0.78±0.02**	57.02±5.48
dSprites	β-TCVAE	0.57±0.07	0.00	0.00	0.80±0.03	111.35±2.13
	dis-VAE	**0.80±0.06**	0.02	0.00	0.79±0.04	129.26±18.56
	SVG	0.52±0.01	0.00	0.00	0.78	**83.04±17.78**
	MTC-VAE	0.77±0.01	**0.06±0.02**	0.00	**0.82**	107.17±5.09
	MTC-VAE*	0.75±0.01	0.03	0.00	0.81±0.01	161.05±34.61
LPC	β-TCVAE	0.94±0.06	0.04±0.01	0.07±0.03	0.65±0.04	115.16±7.06
	dis-VAE	0.90±0.03	0.04±0.02	0.02±0.01	0.55±0.16	116.17±5.30
	SVG	0.72±0.04	0.02±0.01	0.03±0.01	0.23±0.03	**51.82±14.17**
	MTC-VAE	**0.99**	**0.05±0.02**	**0.09±0.04**	**0.67±0.04**	150.49±13.11
	MTC-VAE*	0.98±0.01	0.03±0.02	0.08±0.05	0.59±0.03	137.85±10.14
MMNIST	β-TCVAE	0.68±0.07	0.08±0.06	0.06±0.06	**0.73±0.03**	134.94±15.45
	dis-VAE	0.66±0.06	0.03±0.02	0.02±0.02	0.71±0.01	151.44±7.24
	SVG	0.52±0.01	0.01	0.01±0.01	0.59±0.02	164.17±63.63
	MTC-VAE	**0.96±0.02**	**0.16±0.09**	**0.11±0.04**	0.68±0.01	**103.55±8.84**
	MTC-VAE*	0.92±0.06	0.10±0.09	0.07±0.04	0.68±0.02	225.99±12.80

[10] Kun Su, Xiulong Liu, and Eli Shlizerman. “Predict & Cluster: Unsupervised Skeleton Based Action Recognition”. In: *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2020, pp. 9631–9640.

[11] H. Shi et al. “Bidirectional Long Short-Term Memory Variational Autoencoder”. In: *British Mach. Vis. Conf. (BMVC)*. 2018, p. 165.

[12] Xi Chen et al. “InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets”. In: *Adv. Neural Inf. Process. Sys. (NeurIPS)*. 2016.

[13] Irina Higgins et al. “β-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework.” In: *Inter. Conf. Learn. Represent. (ICLR)*. 2017.

[14] Tian Qi Chen et al. “Isolating Sources of Disentanglement in Variational Autoencoders”. In: *Adv. Neural Inf. Process. Sys. (NeurIPS)*. 2018.

[15] Hyunjik Kim and Andriy Mnih. “Disentangling by Factorising”. In: *Inter. Conf. Mach. Learn. (ICML)*. 2018.

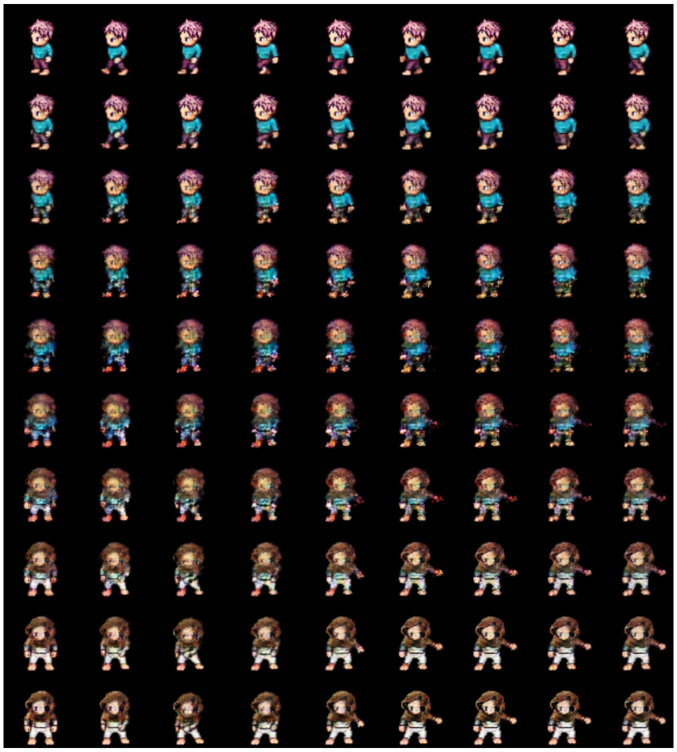


Fig. G.1. LPC: Latent-space traversal. Whole latent space.

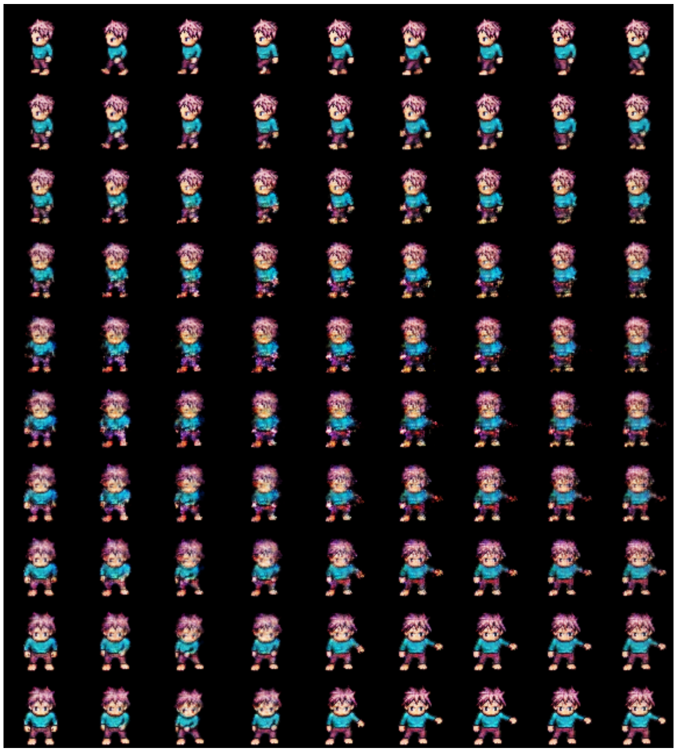


Fig. G.3. LPC: Latent-space traversal. Motion latent space.

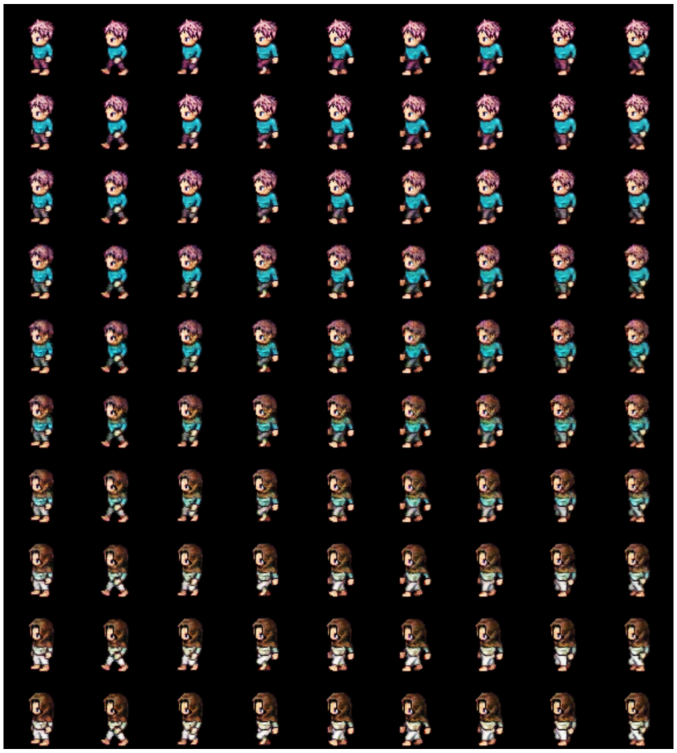


Fig. G.2. LPC: Latent-space traversal. Appearance latent space.

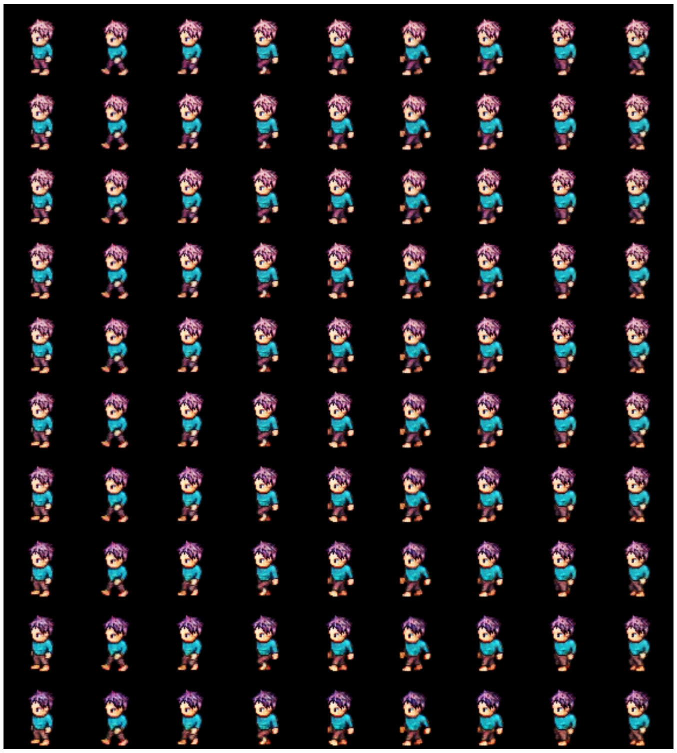


Fig. G.4. LPC: Latent-space traversal. Hair color controlled by unit z_5.

[16] Emily Denton and Vighnesh Birodkar. “Unsupervised Learning of Disentangled Representations from Video”. In: *Adv. Neural Inf. Process. Sys. (NeurIPS)*. 2017.

[17] Ruben Villegas et al. “Decomposing Motion and Content for Natural Video Sequence Prediction”. In: *Inter. Conf. Learn. Represent. (ICLR)*. 2017.

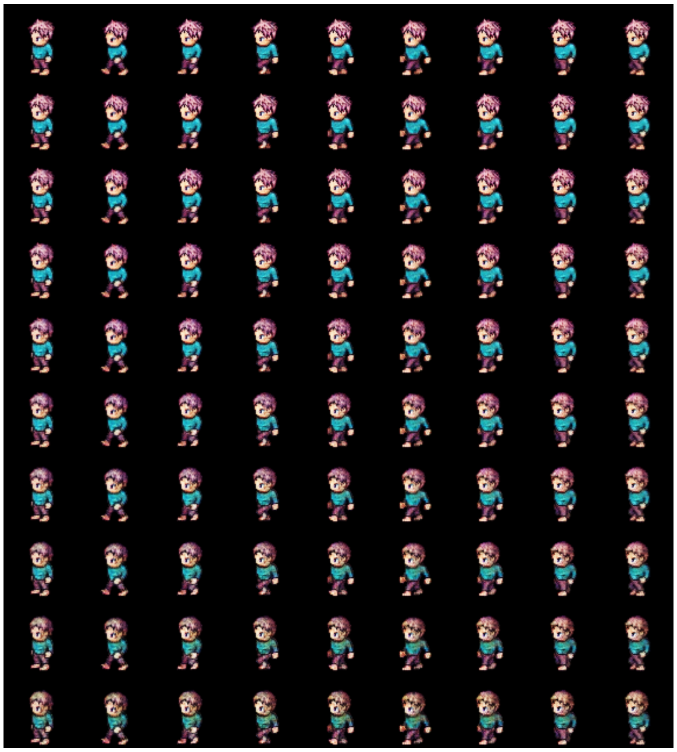


Fig. G.5. LPC: Latent-space traversal. Hairstyle controlled by unit z_7.

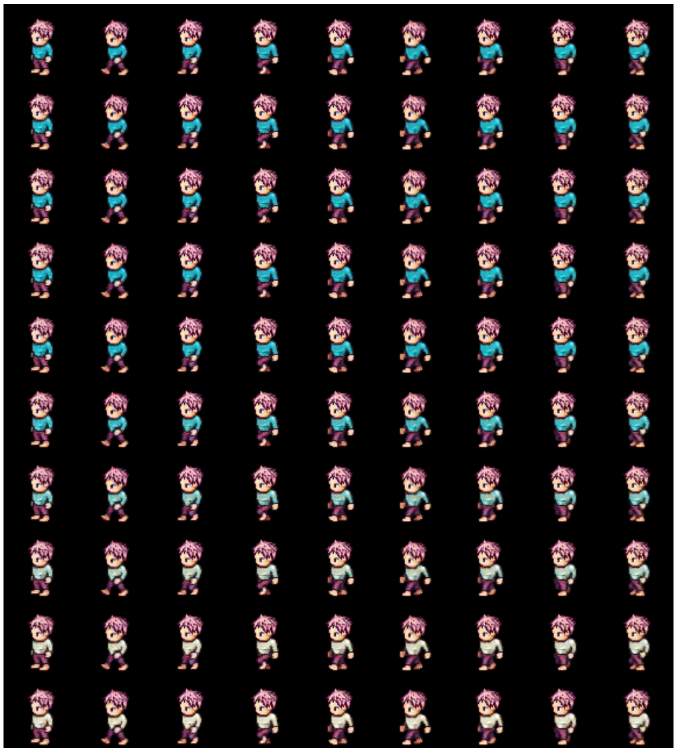


Fig. G.7. LPC: Latent-space traversal. Shirt color controlled by unit z_{11}.

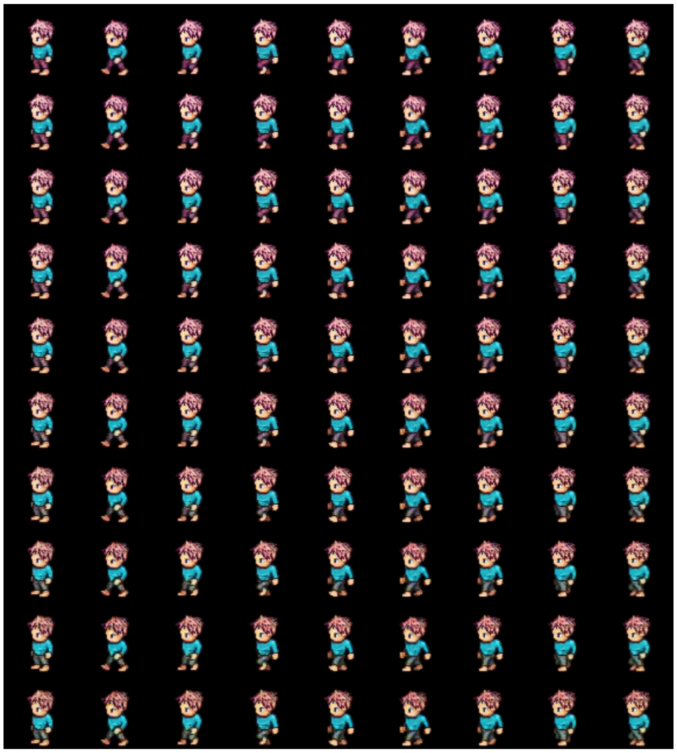


Fig. G.6. LPC: Latent-space traversal. Pants colors controlled by unit z_{13}.

[18] Jun-Ting Hsieh et al. “Learning to Decompose and Disentangle Representations for Video Prediction”. In: *Adv. Neural Inf. Process. Sys. (NeurIPS)*. 2018.

[19] Yixiao Ge et al. “FD-GAN: Pose-guided Feature Distilling GAN for Robust Person Re-identification”. In: *Adv. Neural Inf. Process. Sys. (NeurIPS)*. 2018.

[20] Emily Denton and Rob Fergus. “Stochastic Video Generation with a Learned Prior”. In: *Inter. Conf. Mach. Learn. (ICML)*. 2018.

[21] Alex X. Lee et al. “Stochastic Adversarial Video Prediction”. In: *ArXiv e-prints* arXiv:1804.01523 (2019).

[22] Yingzhen Li and Stephan Mandt. “Disentangled Sequential Autoencoder”. In: *Inter. Conf. Mach. Learn. (ICML)*. 2018.

[23] Yizhe Zhu et al. “S3VAE: Self-Supervised Sequential VAE for Representation Disentanglement and Data Generation”. In: *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2020.

[24] Egor Zakharov et al. “Few-Shot Adversarial Learning of Realistic Neural Talking Head Models”. In: *IEEE Inter. Conf. Comput. Vis. (ICCV)*. 2019.

[25] Yuval Nirkin, Yosi Keller, and Tal Hassner. “FSGAN: Subject Agnostic Face Swapping and Reenactment”. In: *IEEE Inter. Conf. Comput. Vis. (ICCV)*. 2019.

[26] Lele Chen et al. “Lip movements generation at a glance”. In: *European Conf. Comput. Vis. (ECCV)*. 2018.

[27] Hang Zhou et al. “Talking face generation by adversarially disentangled audio-visual representation”. In: *AAAI Conf. Artif. Intell. (AAAI)*. 2019.

[28] Caroline Chan et al. “Everybody Dance Now”. In: *IEEE Inter. Conf. Comput. Vis. (ICCV)*. 2019.

[29] Yipin Zhou et al. “Dance Dance Generation: Motion Transfer for Internet Videos”. In: *IEEE Inter. Conf. Comput. Vis. Wksps. (ICCVW)*. 2019.

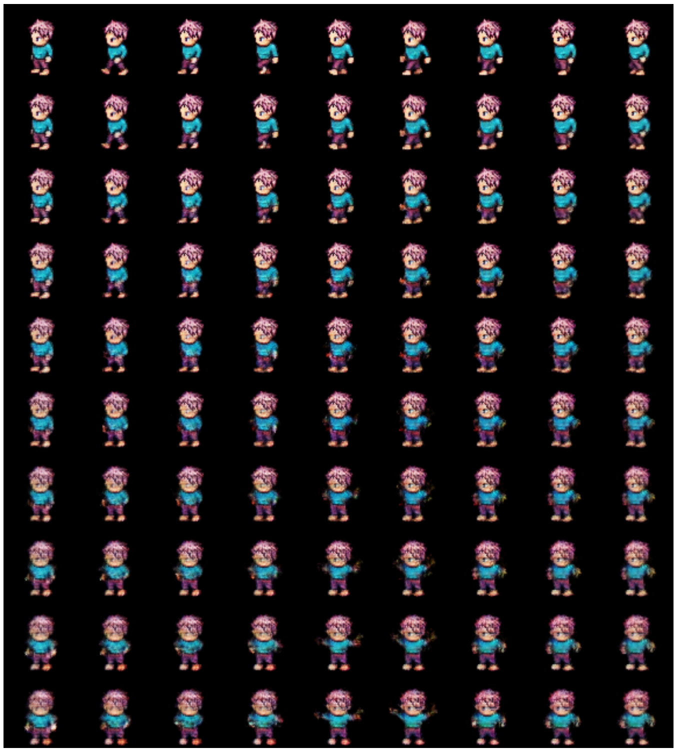


Fig. G.8. LPC: Latent-space traversal. Perspective controlled by units w_0, w_4, w_6.

[30] Lingjie Liu et al. "Neural Rendering and Reenactment of Human Actor Videos". In: *ACM Trans. Graphics*. 2019.

[31] Zhuoqian Yang et al. "TransMoMo: Invariance-Driven Unsupervised Video Motion Retargeting". In: *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2020.

[32] Aayush Bansal et al. "Recycle-GAN: Unsupervised Video Retargeting". In: *European Conf. Comput. Vis. (ECCV)*. 2018.

[33] Aliaksandr Siarohin et al. "Animating Arbitrary Objects via Deep Motion Transfer". In: *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2019.

[34] Aliaksandr Siarohin et al. "Motion Representations for Articulated Animation". In: *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2021, pp. 13653–13662.

[35] Long Zhao et al. "Learning to Forecast and Refine Residual Motion for Image-to-Video Generation". In: *European Conf. Comput. Vis. (ECCV)*. 2018.

[36] Jianwen Xie et al. "Motion-Based Generator Model: Unsupervised Disentanglement of Appearance, Trackable and Intrackable Motions in Dynamic Patterns". In: *AAAI Conf. Artif. Intell. (AAAI)*. 2020.

[37] Max Jaderberg et al. "Spatial Transformer Networks". In: *Adv. Neural Inf. Process. Sys. (NeurIPS)*. 2015.

[38] Diederik P. Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: *Inter. Conf. Learn. Represent. (ICLR)*. 2013.

[39] Francesco Locatello et al. "Weakly-Supervised Disentanglement Without Compromises". In: *Inter. Conf. Mach. Learn. (ICML)*. 2020.

[40] Michael F. Mathieu et al. "Disentangling factors of variation in deep representation using adversarial training". In: *Adv. Neural Inf. Process. Sys. (NeurIPS)*. 2016.

[41] Diane Bouchacourt, Ryota Tomioka, and Sebastian Nowozin. "Multi-Level Variational Autoencoder: Learning Disentangled Representations From Grouped Observations". In: *AAAI Conf. Artif. Intell. (AAAI)*. 2018.

[42] Haruo Hosoya. "Group-based Learning of Disentangled Representations with Generalizability for Novel Contents". In: *Inter. Joint Conf. Artif. Intell.* 2019.

[43] T. Kanade, J. F. Cohn, and Yingli Tian. "Comprehensive Database for Facial Expression Analysis". In: *IEEE Inter. Conf. Automat. Face Gesture Recog.* 2000.

[44] P. Lucey et al. "The Extended Cohn-Kanade Dataset (CK+): A Complete Dataset for Action Unit and Emotion-specified Expression". In: *IEEE Inter. Conf. Comput. Vis., Pattern Recog. Wksps. (CVPRW)*. 2010.

[45] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. "Unsupervised Learning of Video Representations Using LSTMs". In: *Inter. Conf. Mach. Learn. (ICML)*. 2015.

[46] N. Aifanti, C. Papachristou, and A. Delopoulos. "The MUG Facial Expression Database". In: *Inter. Wksp. Image Anal. Multimedia Interact. Serv. (WIAMIS)*. 2010.

[47] Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. "Variational Inference of Disentangled Latent Concepts from Unlabeled Observations". In: *Inter. Conf. Learn. Represent. (ICLR)*. 2018.

[48] Cian Eastwood and Christopher K. I. Williams. "A Framework for the Quantitative Evaluation of Disentangled Representations". In: *Inter. Conf. Learn. Represent. (ICLR)*. 2018.

[49] Karl Ridgeway and Michael C. Mozer. "Learning Deep Disentangled Embeddings With the F-Statistic Loss". In: *Adv. Neural Inf. Process. Sys. (NeurIPS)* (2018).

[50] Zhou Wang et al. "Image Quality Assessment: From Error Visibility to Structural Similarity". In: *IEEE Trans. Image Process.* 13.4 (2004).

[51] Martin Heusel et al. "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium". In: *Adv. Neural Inf. Process. Sys. (NeurIPS)*. 2017.

[52] Aliaksandr Siarohin et al. "Deformable GANs for Pose-based Human Image Generation". In: *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2018.

[53] Guha Balakrishnan et al. "Synthesizing Images of Humans in Unseen Poses". In: *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2018.

[54] Jong Chul Ye and Woon Kyoung Sung. "Understanding Geometry of Encoder-Decoder CNNs". In: *Inter. Conf. Mach. Learn. (ICML)* (2019).

[55] Zengjie Song, Oluwasanmi Koyejo, and Jiangshe Zhang. "Toward a Controllable Disentanglement Network". In: *IEEE Trans. Cybern.* (2020).

[56] Aliaksandr Siarohin et al. "First Order Motion Model for Image Animation". In: *Adv. Neural Inf. Process. Sys. (NeurIPS)*. 2019.

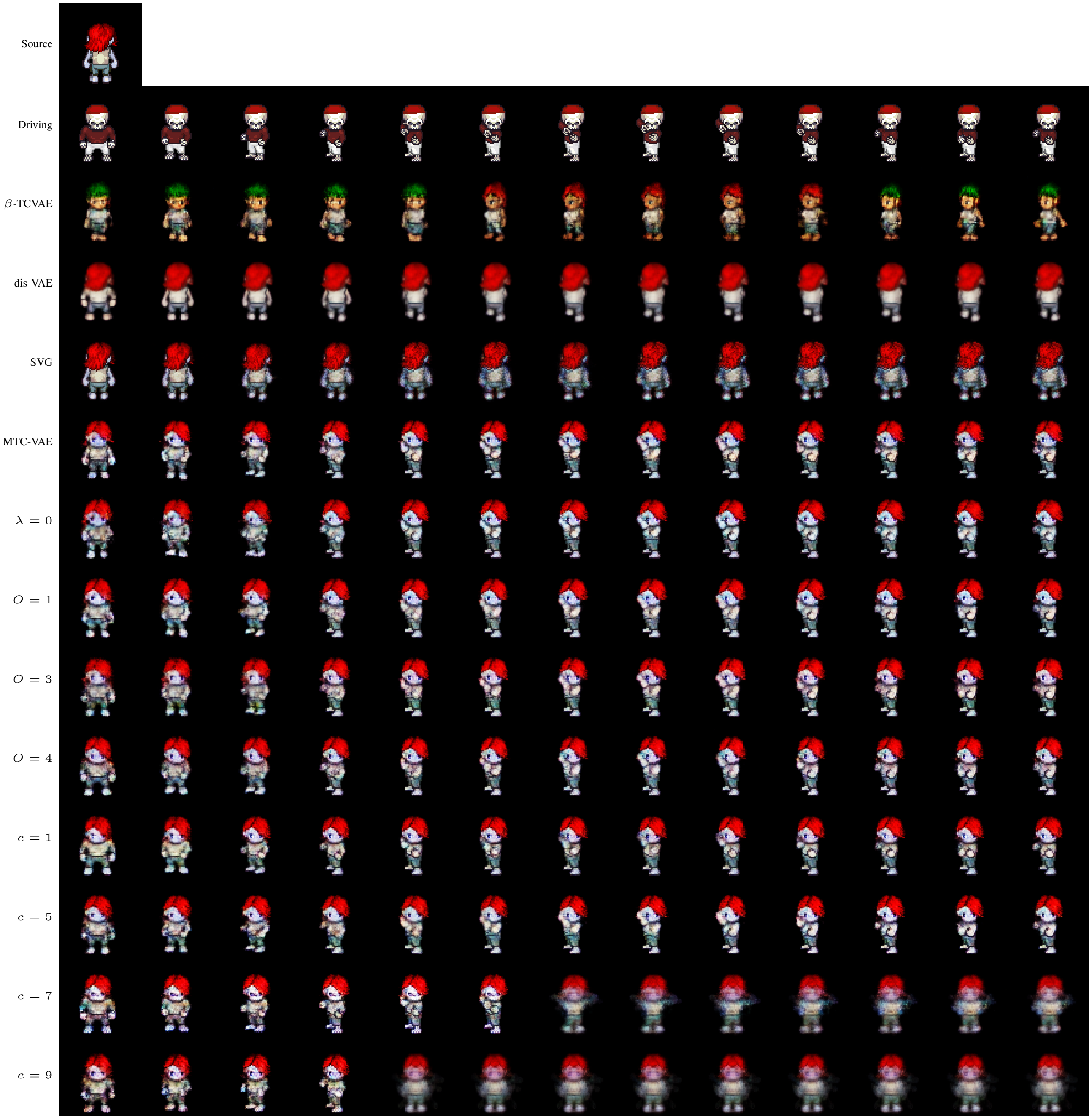


Fig. G.9. LPC: examples of reenactment for the soft generalization scenario. Comparison with the baselines (β-TCVAE and dis-VAE), and ablation study on the chunk size (c), Blind Reenactment Loss (λ), and order of the model (O).

[57] Kfir Aberman et al. “Learning character-agnostic motion for motion retargeting in 2D”. In: *ACM Trans. Graphics* 38.4 (2019).

[58] Bruce D. Lucas and Takeo Kanade. “An Iterative Image Registration Technique with an Application to Stereo Vision”. In: *Inter. Joint Conf. Artif. Intell.* 1981.

[59] Wen Liu et al. “Liquid Warping GAN: A Unified Framework for Human Motion Imitation, Appearance Transfer and Novel View Synthesis”. In: *IEEE Inter. Conf. Comput. Vis. (ICCV)*. 2019.

[60] Chung-Yi Weng, Brian Curless, and Ira Kemelmacher-Shlizerman. “Photo Wake-Up: 3D Character Animation from a Single Photo”. In: *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2019.

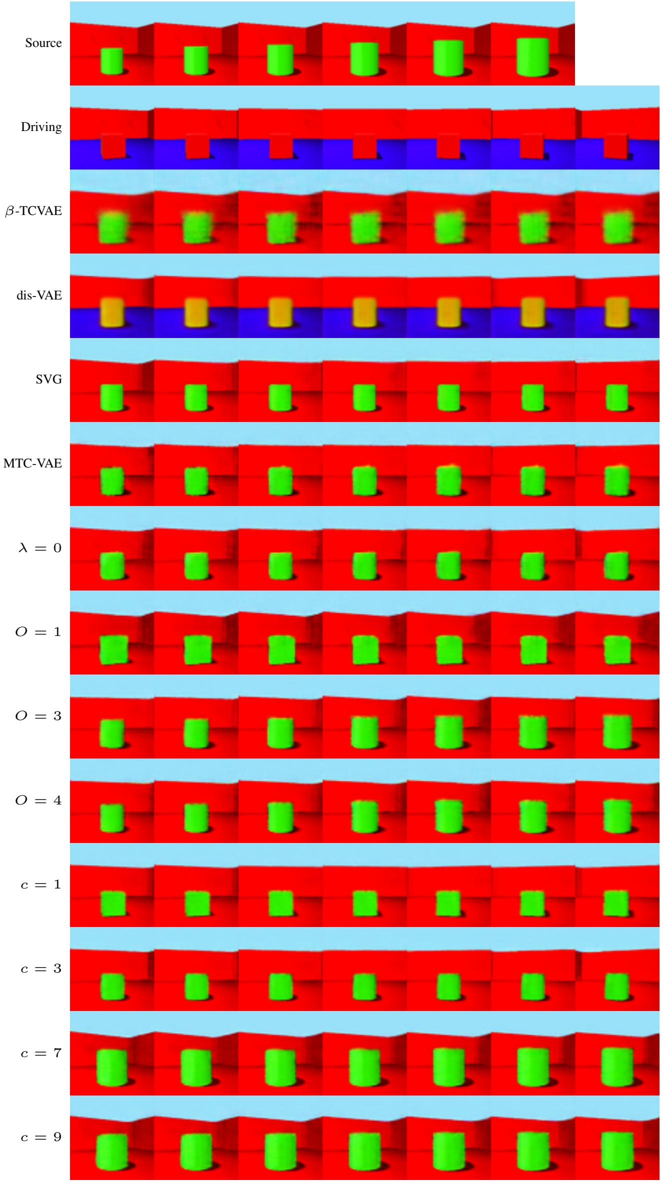


Fig. G.10. 3dShapes: examples of reenactment for appearance holdout. Comparison with the baselines (β-TCVAE and dis-VAE), and ablation study on the chunk size (c), Blind Reenactment Loss (λ), and order of the model (O).

Juan F. Hernández Albarracín is a Ph.D. candidate at University of Campinas, Brazil. He has a bachelor degree in Computer Engineering from National University of Colombia (2014), and a M.Sc. degree in Computer Science from University of Campinas (2017). 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.

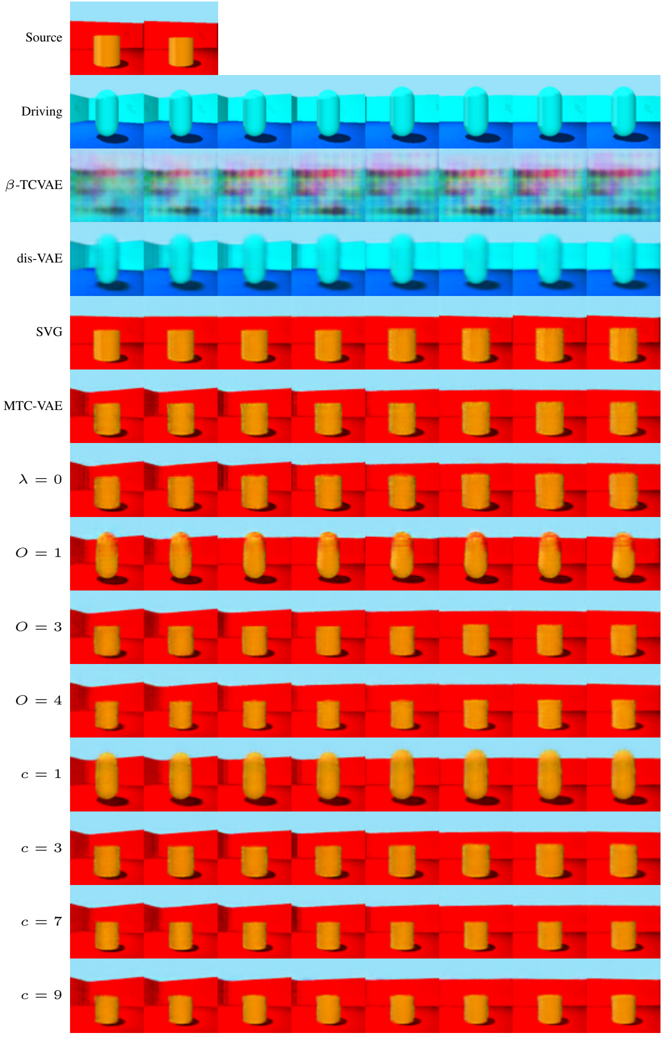


Fig. G.11. 3dShapes: examples of reenactment for motion holdout. Comparison with the baselines (β-TCVAE and dis-VAE), and ablation study on the chunk size (c), Blind Reenactment Loss (λ), and order of the model (O).

Adín Ramírez Rivera (S'12, M'14, SM'21) received his B.Eng. degree in Computer Engineering from Universidad de San Carlos de Guatemala (USAC), Guatemala in 2009. He completed his M.Sc. and Ph.D. degrees in Computer Engineering from Kyung Hee University, South Korea in 2013. He is currently an Associate Professor at 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.

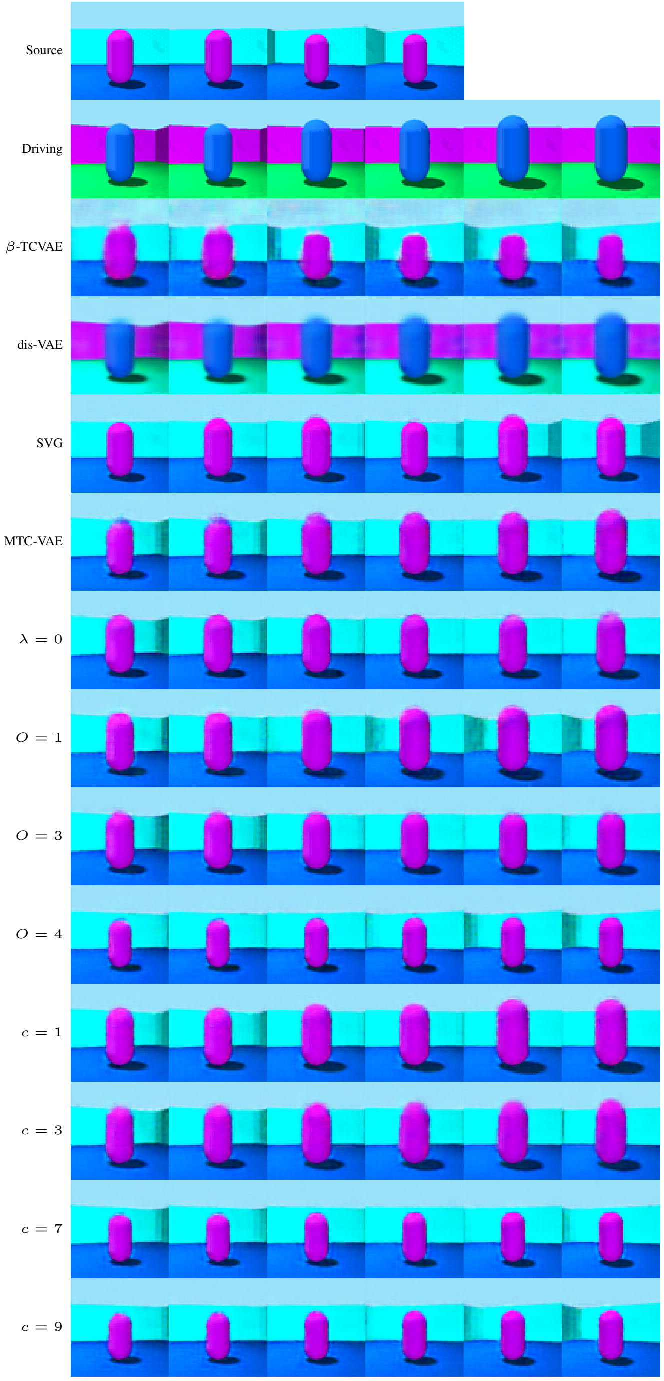


Fig. G.12. 3dShapes: examples of reenactment for the soft generalization scenario. Comparison with the baselines (β-TCVAE and dis-VAE), and ablation study on the chunk size (c), Blind Reenactment Loss (λ), and order of the model (O).

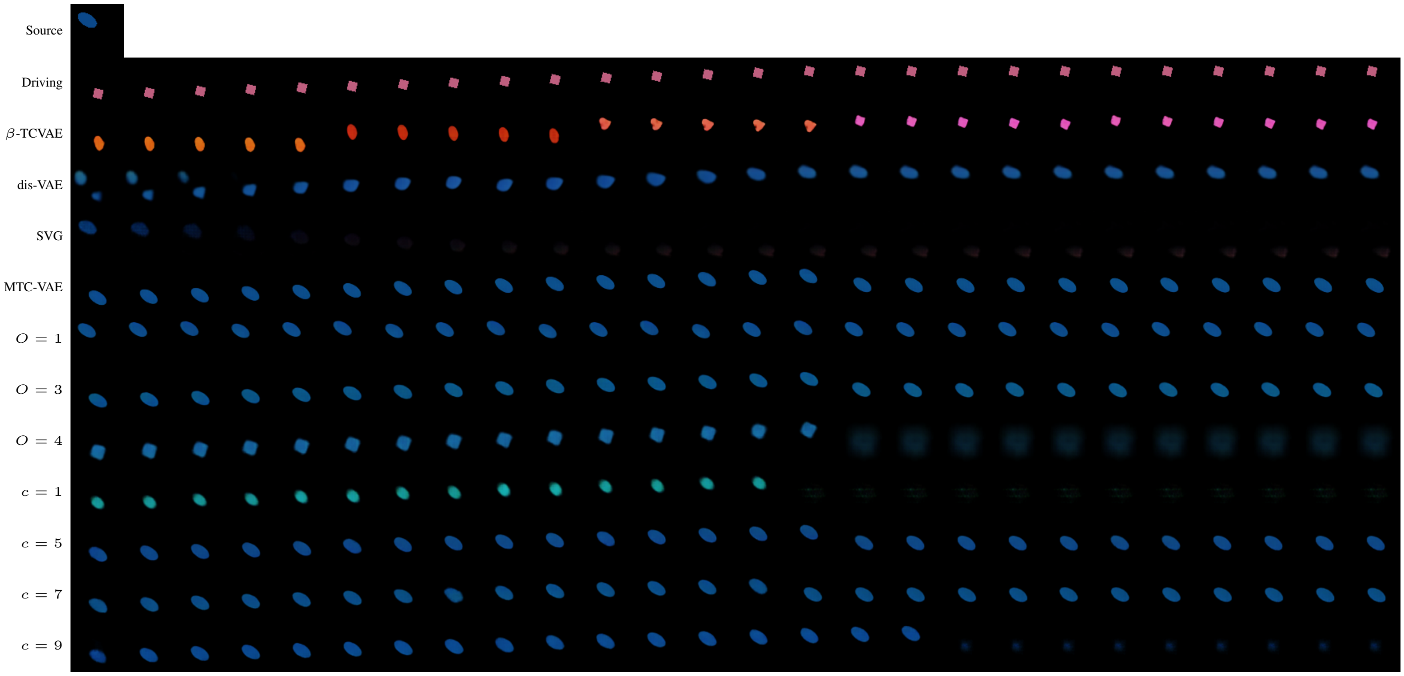


Fig. G.13. dSprites: examples of reenactment for the soft generalization scenario. Comparison with the baselines (β-TCVAE and dis-VAE), and ablation study on the chunk size (c), Blind Reenactment Loss (λ), and order of the model (O).

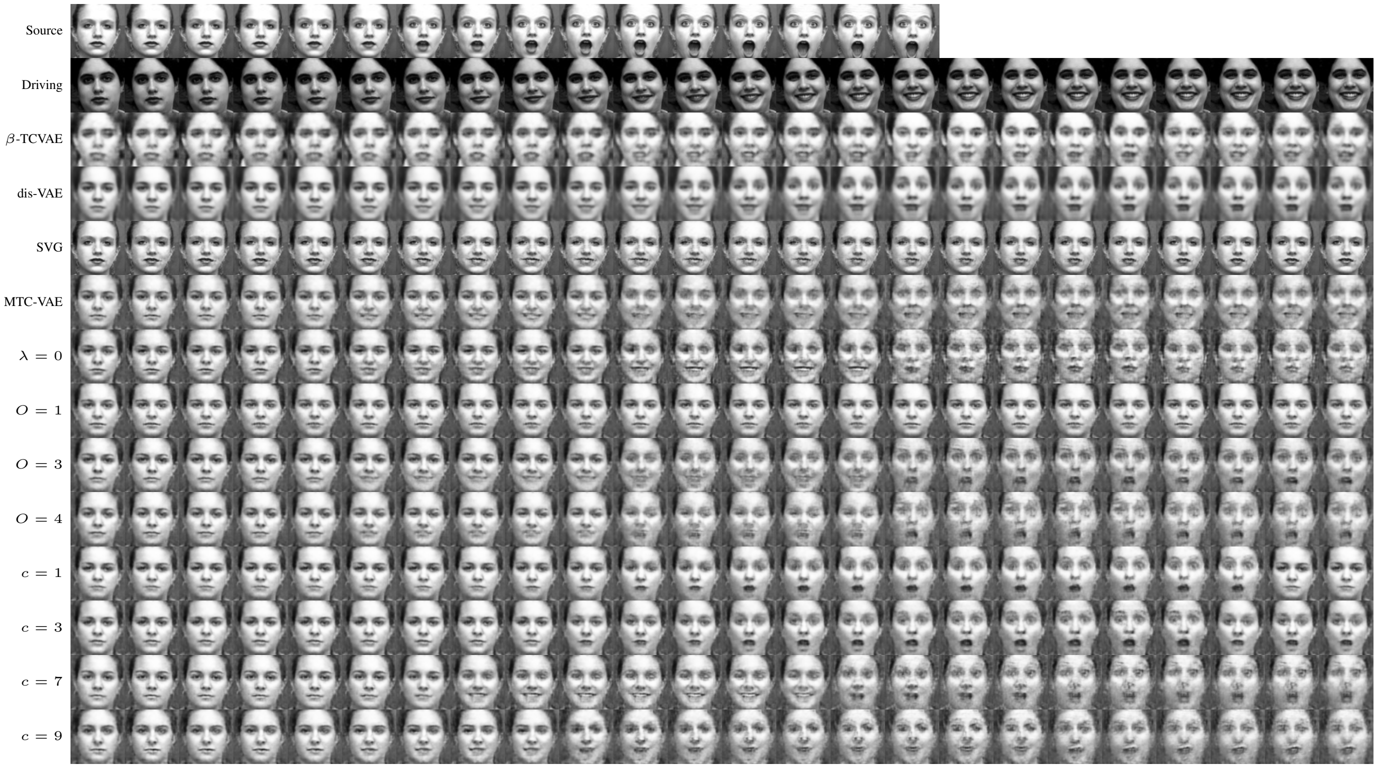


Fig. G.14. CK+: examples of reenactment for appearance holdout. Comparison with the baselines (β-TCVAE and dis-VAE), and ablation study on the chunk size (c), Blind Reenactment Loss (λ), and order of the model (O).

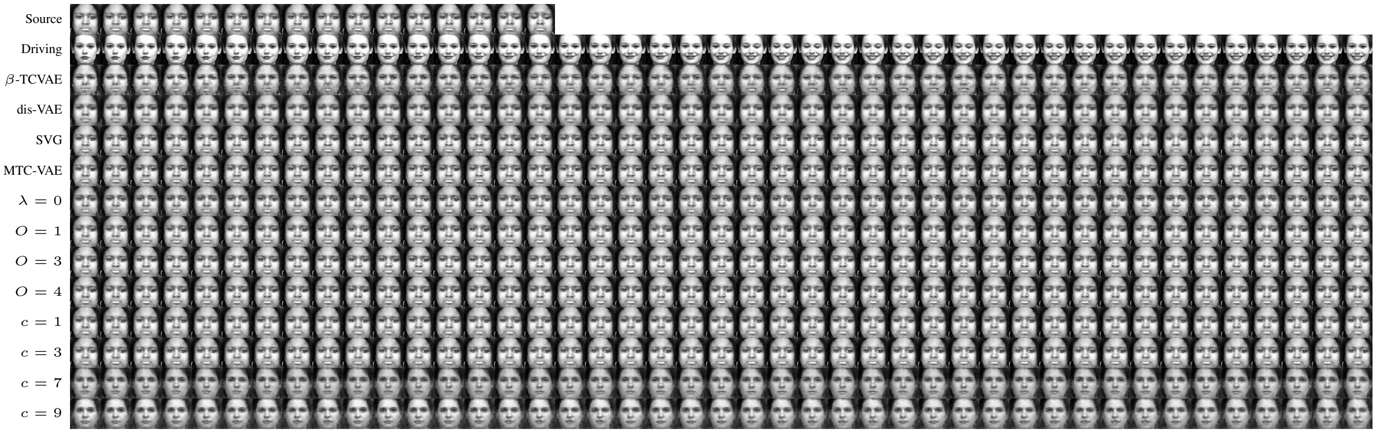


Fig. G.15. CK+: examples of reenactment for motion holdout. Comparison with the baselines (β-TCVAE and dis-VAE), and ablation study on the chunk size (c), Blind Reenactment Loss (λ), and order of the model (O).

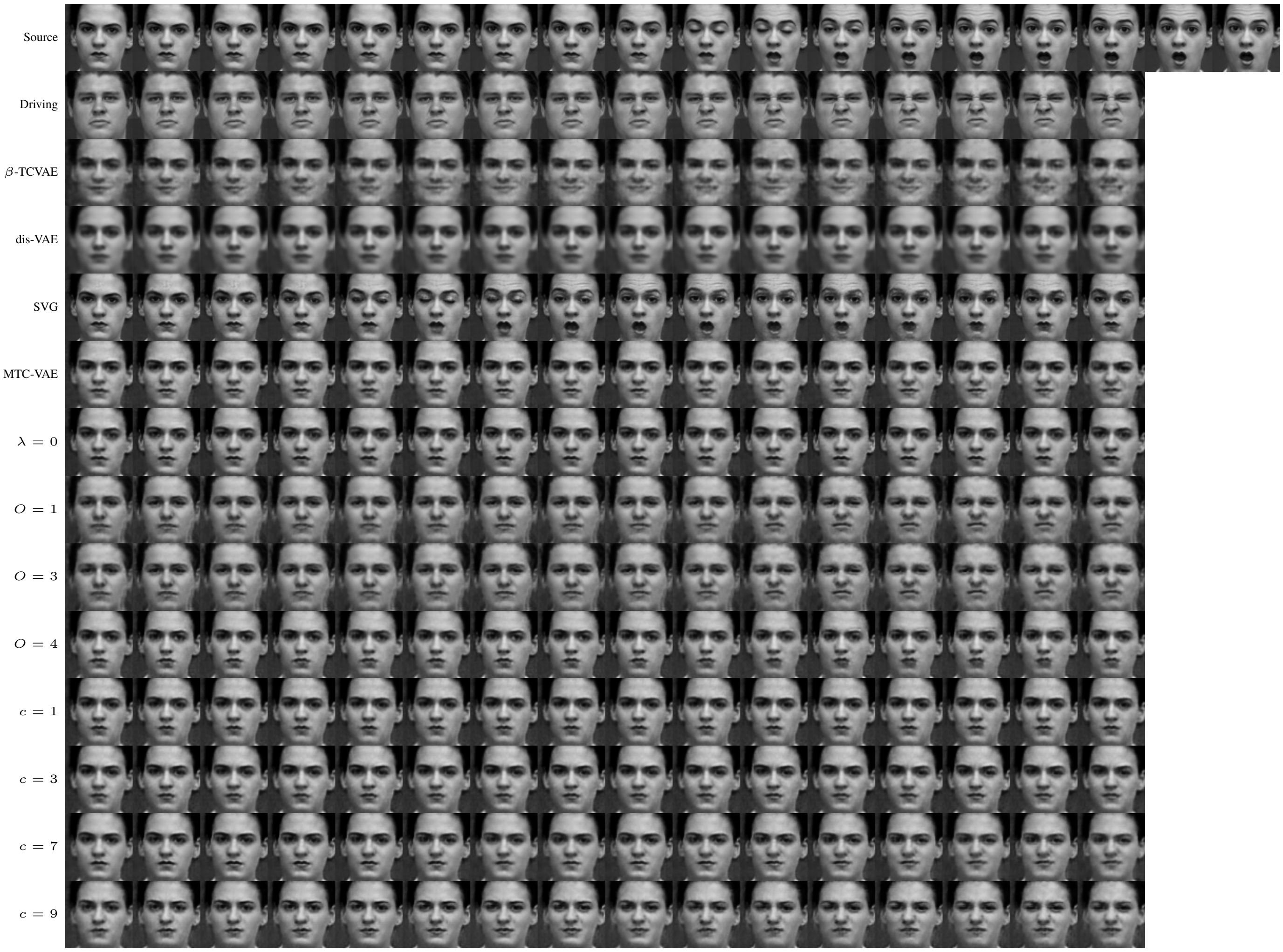


Fig. G.16. CK+: examples of reenactment for the soft generalization scenario. Comparison with the baselines (β-TCVAE and dis-VAE), and ablation study on the chunk size (c), Blind Reenactment Loss (λ), and order of the model (O).

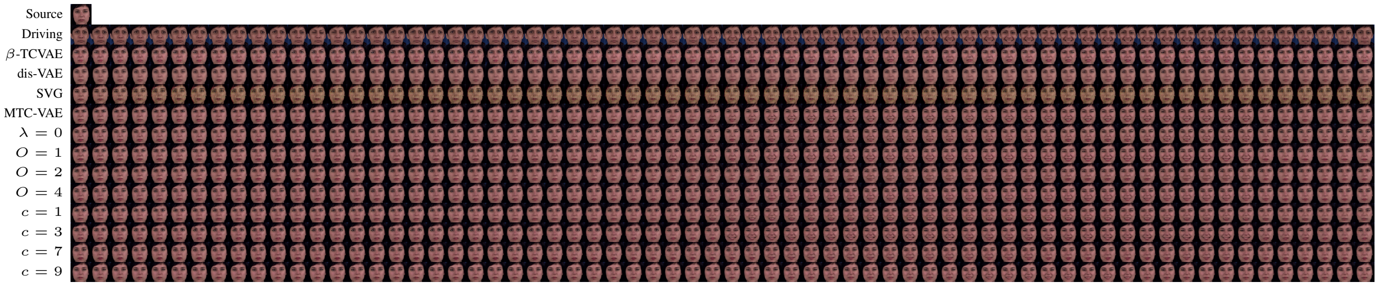


Fig. G.17. MUG: examples of reenactment for the soft generalization scenario. Comparison with the baselines (β-TCVAE and dis-VAE), and ablation study on the chunk size (c), Blind Reenactment Loss (λ), and order of the model (O).

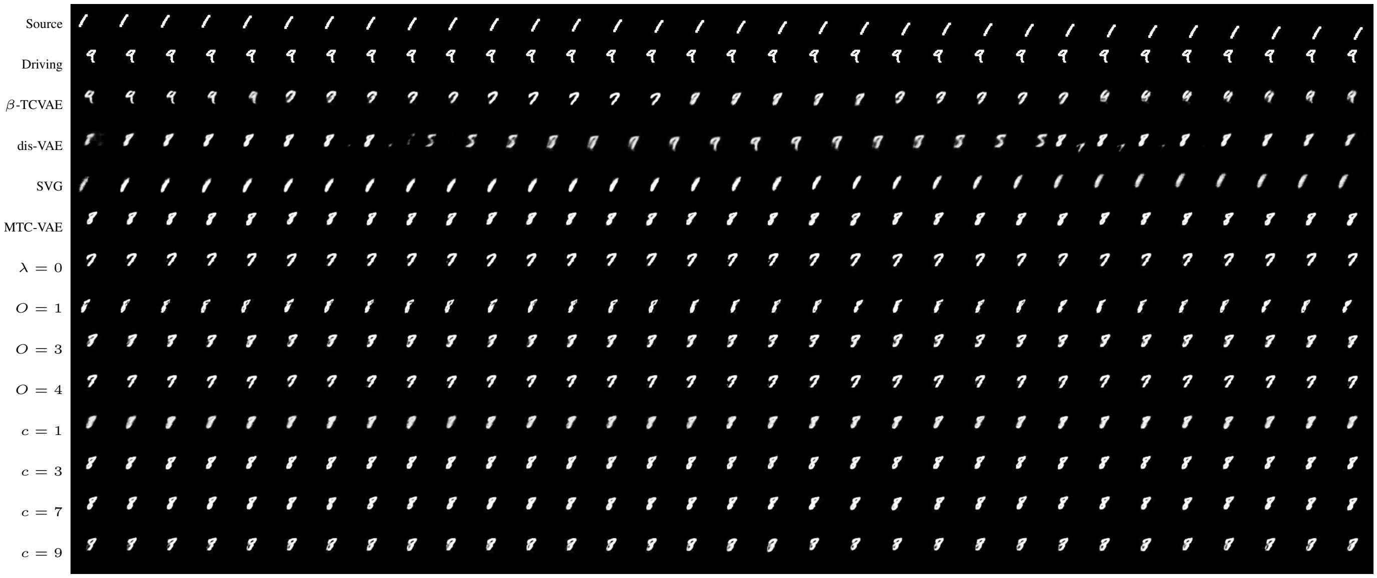


Fig. G.18. MMNIST: examples of reenactment for appearance holdout. Comparison with the baselines (β-TCVAE and dis-VAE), and ablation study on the chunk size (c), Blind Reenactment Loss (λ), and order of the model (O).

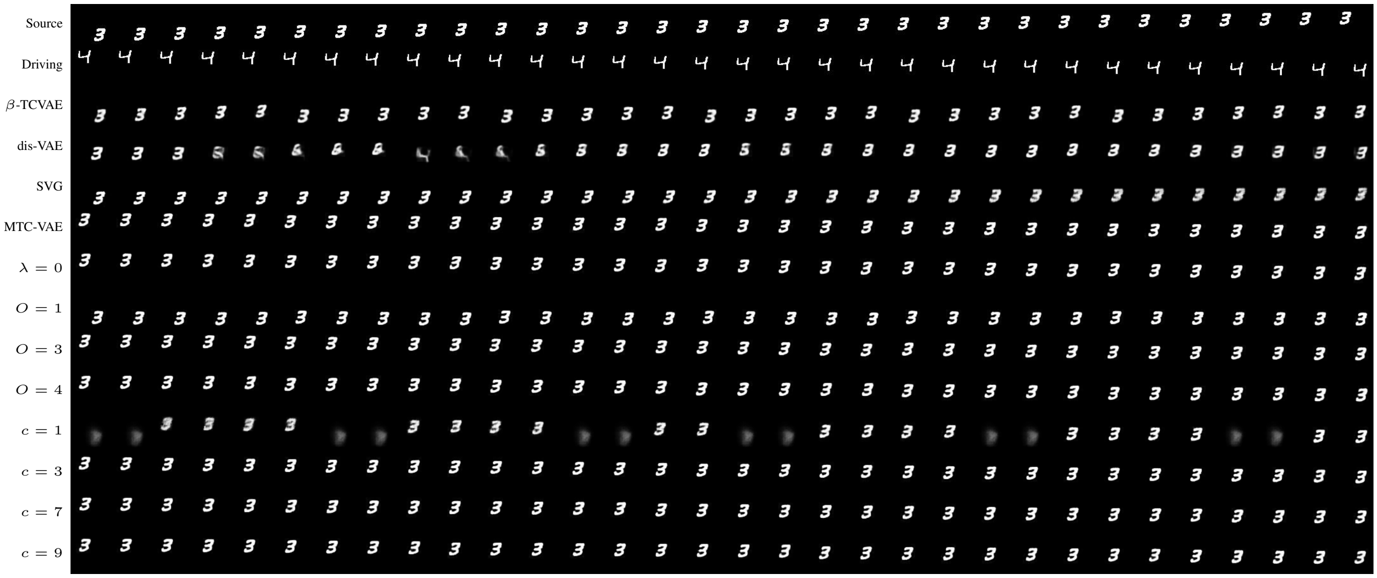


Fig. G.19. MMNIST: examples of reenactment for motion holdout. Comparison with the baselines (β-TCVAE and dis-VAE), and ablation study on the chunk size (c), Blind Reenactment Loss (λ), and order of the model (O).

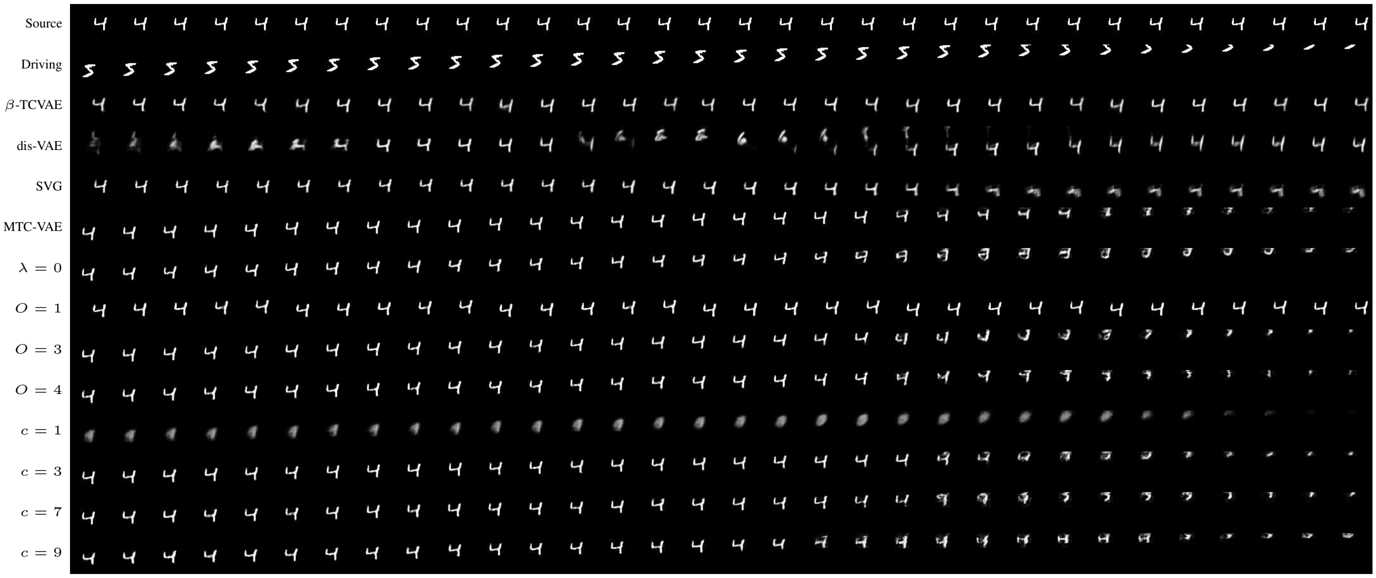


Fig. G.20. MMNIST: examples of reenactment for the soft generalization scenario. Comparison with the baselines (β-TCVAE and dis-VAE), and ablation study on the chunk size (c), Blind Reenactment Loss (λ), and order of the model (O).