Supplementary Material
Behavior-Driven Synthesis of Human Dynamics

Andreas Blattmann* Timo Milbich* Michael Dorkenwald* Björn Ommer
Heidelberg Collaboratory for Image Processing, IWR, Heidelberg University, Germany

Contents

A Behavior Model
A.1. Training and Implementation Details . . . .
A.2. Protocols of Ablation Studies . . . . . .
A.3. Additional Results . . . . . .

B Posture and Appearance Model
B.1. Architecture and Losses . . . . . .
B.2. Training Details . . . . . .
B.3. Additional Results . . . . . .

A. Behavior Model

A.1. Training and Implementation Details

Behavior model Most of the implementation details of our behavior model are already described in the main paper in Sec. 4. We train our model on a single Titan Xp using ADAM [5] optimizer with learning rate 0.0001 which is decreased after $10, 25$ and 35 epochs. For data preprocessing, we normalize the posture keypoints to have zero mean and unit variance.

Invertible Transformation $\mathcal{T}_\xi$ To highlight the need for learning an explicit mapping between the prior $p(z_\beta)$ and the posterior $q(z_\beta|\boldsymbol{x}, x_t)$, we plot in Fig. 1 2D UMAP [11] visualizations of samples drawn from these distributions without and with using $\mathcal{T}_\xi$. Fig. 1 (a) shows a clear mismatch between both distributions. Fig. 1 (b) demonstrates that applying the transformation $\mathcal{T}_\xi$ helps to align prior and posterior, which is also reflected by the results discussed in the paragraph 'Behavior Sampling'.
Our normalizing flow model $\mathcal{T}_\xi$ is implemented as a stacked sequence of 15 invertible neural networks based on an input dimensionality of $D = 1024$. Each consists of 3 blocks of subsequently applied actnorm [6], affine coupling layers [2] and shuffling layers. The affine coupling layers consist of 2 fully connected layers with dimensionality $D = 1024$. We trained the normalizing flow model on a single Titan Xp for 5 epochs with batchsize 64 and ADAM [5] optimizer with learning rate 6.5×10^{-6}.

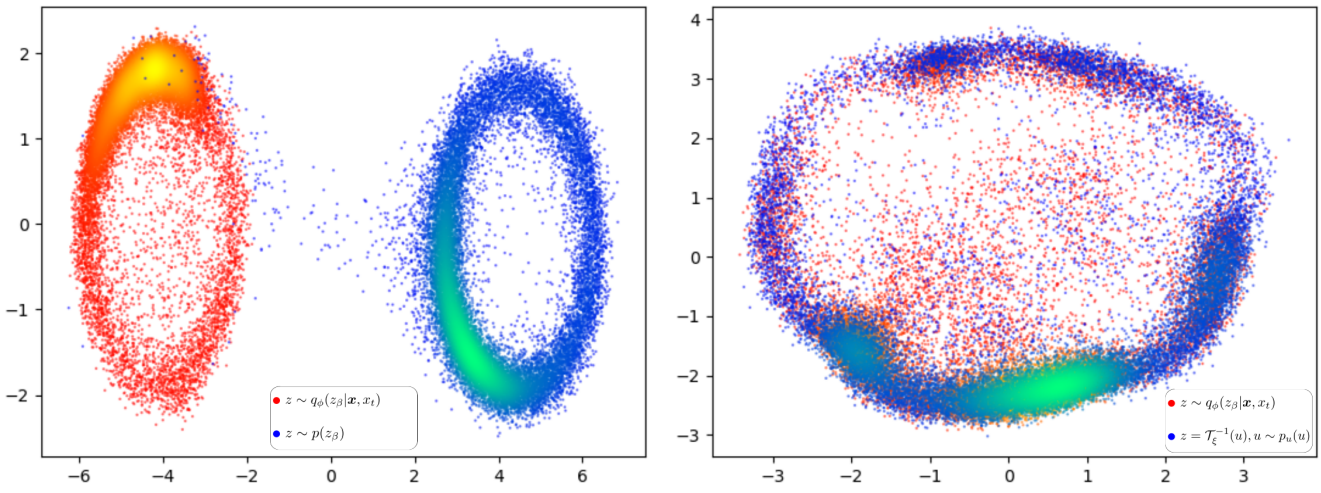
*Indicates equal contribution

A.2. Protocols of Ablation Studies

Sample-Reality Classifier In Fig. 4 (b) of our main paper, we evaluate the quality of our generations with a recurrent binary classifier similar to [1]. The task of the classifier is to distinguish between 25k samples ground-truth sequences and 25k synthesized generations based on samples from the prior distribution. The classifier consists of a single layer GRU network with 256 hidden dimension for feature extraction, followed by a fully connected layer before applying the sigmoid function for binary classification. We optimize the classifier via stochastic gradient descent for 2k iterations, with a batch size of 256, a learning rate of 0.001 and a momentum of 0.9.

Average Regression Error (RE) In Tab. 1 of our main paper we provide an explicit quantitative evaluation of the disentanglement of posture and behavior. We adopt the experiments of [10] and train a Multi-Layer Perceptron (MLP) consisting of 3 linear layers with 512, 256 and 51 neurons to predict the keypoint locations of postures in sequence $\boldsymbol{x}_\beta$ at different time-steps T based on their corresponding extracted behavior representation z_β. Therefore, we train the MLP for 20 epochs with Adam [5] optimizer and a learning rate of 1×10^{-3} on the test set as described in the main paper. Intuitively, if z_β captures no information about posture, RE is high and converges to 0 is lots of posture information is captured.

Action Classifier In Sec. 4 of our main paper we evaluate the informativeness of the behavior representation z_β by means of their benefit as a feature representation for action classification on Human3.6M dataset [4] (*acc.* values of the evaluated models in Tab. 1). For this purpose, we directly train a linear classifier on top of the frozen behavior encoder. For training and evaluation we use the same train-test split as described in the main paper. For the validation classifier which results in an test accuracy of 45% ('gt:0.45', Tab. 1, main paper), we train a classifier with trainable feature



(a) (b)

Figure 1. 2D-UMAP of prior and posterior samples without (a) and with (b) learning the normalizing flow transformation $\mathcal{T}_\xi$ for distribution alignment.

representation which has the same architecture as our behavior encoder $q_\phi(z_\beta|\boldsymbol{x}, x_t)$ to predict the action labels from ground truth sequences of 50 frames.

A.3. Additional Results

Subsequently, we show additional visual results depicted as figures in this manuscript or as videos in the folder *'videos'*.

Behavior Transfer We show more examples of behavior transfer in Fig. 2-7, both as postures and RGB images, similar to Fig. 3 of our main paper to further demonstrate the effectiveness of our approach. Moreover, we also show videos based on both our model and the cAE/cVAE models which we quantitatively evaluated in Sec. 4 (Quantitative evaluation).

(i) cAE: The video *'behavior_transfer_CAE.mp4'* shows behavior re-enactments based on the cAE model. The topmost row depicts the source behavior sequence $\boldsymbol{x}_\beta$, while the leftmost column shows different target postures x_t. Based on these we show all pairwise combinations. We see that in general the cAE model quickly warps from x_t to some early posture of $\boldsymbol{x}_\beta$. Next, it almost exactly copies the remaining posture sequence $\boldsymbol{x}_\beta$. Thus, given a certain $\boldsymbol{x}_\beta$ each re-enacted sequence is identical and independent of the given target pose x_t, rather than transferring only the behavior dynamics to the observed target postures. This is explained by the missing disentanglement of posture and behavior, which allows the cAE model to fully capture the complete posture information of $\boldsymbol{x}_\beta$ in z_β.

(ii) cVAE: The video *'behavior_transfer_CVAE.mp4'* shows behavior re-enactments based on the cVAE model. The topmost row depicts the source behavior sequence $\boldsymbol{x}_\beta$, while the leftmost column shows different target postures x_t. Based on these we show all pairwise combinations. We observe that this model predicts a likely future continuation based on the target posture x_t, thus not using the behavior representation z_β for additional, dedicated information describing the source sequence $\boldsymbol{x}_\beta$. This is explained by posterior collapse, i.e. mean and variance of $q_\phi(z_\beta|\boldsymbol{x}, x_t)$ collapsing to almost constant values.

(iii) Ours: The videos *'behavior_transfer1.mp4'* and *'behavior_transfer2.mp4'* show behavior re-enactments based on our proposed behavior transfer model. In both videos, the topmost row depicts the source behavior sequence $\boldsymbol{x}_\beta$, while the leftmost column shows different target postures x_t. Based on these we show all pairwise combinations. We see that our model extracts the behavior dynamics from diverse source sequences $\boldsymbol{x}_\beta$ and successfully transfers them to arbitrary target postures x_t resulting in meaningful re-enactments of behavior β. Moreover, *'behavior_transfer1_RGB.mp4'* and *'behavior_transfer2_RGB.mp4'* show RGB video syntheses of our results using our model for posture-appearance transfer (see Appendix B and main paper).

Behavior Sampling We now compare syntheses of novel behavior based on samples z_β drawn from the prior distribution $p(z_\beta)$, with and without using the transformation $\mathcal{T}_\xi$ for correcting the mismatch with the posterior $q_\phi(z_\beta|\boldsymbol{x}, x_t)$. For this purpose, we recursively synthesize behavior using sampled behavior representations z_β and the last posture of the previously generated posture sequence. For detailed comparison, we show such a concatenated posture sequence without using $\mathcal{T}_\xi$ in video *'sample_loop_prior.mp4'* and with $\mathcal{T}_\xi$ in *'sample_loop_flow.mp4'*. We observe that the first suffers from synthesis artifacts due to out-of-distribution samples z_β, which in particularly become evident at the beginning of each behavior synthesis. In contrast, the recursively generated sequence using transformation $\mathcal{T}_\xi$ does not exhibit such artifacts and consequently results in a much smoother and more realistic sequence of diverse human behavior.
Moreover in video *'samples.mp4'* we show behavior synthesis based on random sampling z_β from the prior distribution which are then transformed using $\mathcal{T}_\xi$. The leftmost column depicts the target postures x_t with each performing 6 randomly sampled behaviors. Note, that for each target posture x_t we use different samples z_β.

Behavior Nearest Neighbors To also demonstrate visually that our learned representation z_β actually captures behavior dynamics while discarding posture information, we find nearest neighbours to the ground-truth training sequences. Therefore, we re-enact a source behavior $\boldsymbol{x}_\beta$ using a random target posture x_t. Next, we find its nearest neighbour in the training sequences based on *(i)* distance between behavior representations z_β and *(ii)* average distances between postures sequences (based on alignment w.r.t. the pelvis keypoints). The video *'nearest_neighbors.mp4'* shows our results: Each column depicts a separate example showing the 'Source Behavior', the 'Nearest Neighbor based on Behavior representation', the 'Behavior Re-enactment of Source Behavior' and the 'Nearest Neighbor based on Posture', i.e. average posture distance. We observe that while there exist close training sequences in terms of posture, the nearest neighbors based on z_β show *similar* behavior dynamics while being *dissimilar* in posture.

Behavior Interpolation To further analyze the regularity of our behavior representation z_β, we interpolate between the behavior observed in two sequences $\boldsymbol{x}_\beta^1$ and $\boldsymbol{x}_\beta^2$. To this end, we first extract their corresponding behavior representations z_β^1, z_β^2 and interpolate between them at equidistant steps, i.e. $(1-\lambda)\cdot z_\beta^1 + \lambda \cdot z_\beta^2;\ \ \lambda \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. Next, we generate a sequence of interpolated behavior using our decoder $p_\theta(\boldsymbol{x}|z_\beta, x_t)$ with x_t being the first frame of $\boldsymbol{x}_\beta^1$, respectively $\boldsymbol{x}_\beta^2$. Note, that for $\lambda \in \{0, 1.0\}$ we basically reconstruct the source sequences $\boldsymbol{x}_\beta^1$, $\boldsymbol{x}_\beta^2$. We show the resulting posture sequences in *'interpolations_01.mp4'-'interpolations_03.mp4'* and with additional RGB image overlay in *'interpolations_rgb_01.mp4'-'interpolations_rgb_03.mp4'*.

Behavior Generalization We now demonstrate the robustness of our proposed model to unseen behavior dynamics by leaving out sets of entire classes during training[1] and, subsequently, performing behavior transfers based on source sequences $\boldsymbol{x}_\beta$ sampled from these classes.
We show results for both excluding walking actions ('walking', 'walking dog', 'walking together') in video *'behavior_transfer_generalization_walking.mp4'* and sitting actions ('sitting', 'sitting down', 'purchases') in video *'behavior_transfer_generalization_sitting.mp4'*. The top rows depict the source behaviors $\boldsymbol{x}_\beta$ and the leftmost columns show the target postures x_t. In both cases our model is able to correctly infer the body dynamics characterizing these actions.

B. Posture and Appearance Model

Our proposed conditional framework for disentanglement can also be applied for the task of appearance transfer. Instead of disentangling posture from behavior, we disentangle posture from appearance of persons depicted on static images and use the resulting model to generate RGB video sequences based on the re-enacted posture sequences as reported in the main paper. Note that the posture and appearance model operates on 2D keypoints. Therefore, we project the 3D keypoints locations of the re-enacted sequences onto the image plane. Subsequently, we provide implementation details and additional experiments on DeepFashion [7] and Market1501 [15] datasets.

B.1. Architecture and Losses

Our model for appearance transfer is based on a UNet architecture similar to VUnet [3]. The UNet maps from posture x_t, i.e. keypoint skeletons, to RGB images with appearance information added at the bottleneck which is extracted from some image I_α by an appearance encoder. Now, we provide implementation details for the posture- and appearance encoder, as well as the decoder.

Appearance encoder: The appearance encoder, which is the equivalent of the behavior encoder for the task of posture-appearance disentanglement, is implemented as a fully convolutional network. We gradually downsample the input image I_α up to a spatial size of 4×4. Each downsampling stage consists of 2 ResNet blocks and downsampling is performed using a convolutional layer with stride 2. We double the number of feature channels at every stage up to

[1] Note, that we only use labels for excluding training sequences in this experiment, but not for the training procedure itself.

Method	DeepFashion		Market1501	
	IS	SSIM	IS	SSIM
VUnet (Esser et al. 2018)	3.09	0.79	3.21	**0.35**
DIG [9]	**3.23**	0.61	3.44	0.10
PG^2 [8]	3.09	0.76	**3.46**	0.25
Ours	3.08	**0.80**	3.16	**0.35**

Table 1. Evaluation of our shape-appearance transfer model based on image quality metrics on DeepFashion [7] and Market1501 [15] (Reconstruction Setting).

a maximum number of 128 which is then kept fixed. At the bottleneck we compute mean and variance both based on the layer outputs of spatial size 8 and 4 [3].

UNet encoder and decoder : Both the encoder and decoder branch of the UNet are similarly designed as the appearance encoder with skip connections connecting them at each stage. For upsampling in the decoder we use bilinear interpolation. At the bottleneck, we concatenate the feature maps of the posture stream with the encodings of the appearance encoder.

Auxilliary decoder: The auxilliary decoder consists of two convolutional layers with kernel size 8 and 4. It takes the appearance encodings as input (both at spatial sizes 8 and 4) and outputs one vector for each with dimensionality 256. Following that, we add 6 linear layers (with dimenionalities 512,512,256,128,64,34) to predict the posture keypoints.

Optimization: For optimizing the likelihood $\mathbb{E}_{q_\phi}(\boldsymbol{x}|z_\alpha, x_t)$ similar to Eq. (5), with z_α denoting the appearance encoding, we employ both standard pixel-wise mean squared error and a perceptual loss [3]. The latter is a feature matching loss and often used to emphasize on structural information such as contours and texture. It is formulated as

$$\mathcal{L}_{\alpha,feat} = \sum_k \lambda_k \cdot \|F_k(I_\alpha) - F_k(\tilde{I}_\alpha)\|_1 \qquad (1)$$

where F_k denote feature layers of a pretrained VGG19 network [13], the weights λ_k control the amount contribution of each layer k, I_α is the target image to be reconstructed and $\tilde{I}_\alpha$ its reconstruction, i.e. output of the decoder. Note that the model does not require image pairs of persons with the same appearance label and can hence be trained solely by reconstructing static image frames.

B.2. Training Details

Human3.6m On Human3.6M, we train our appearance model for 150k iterations using ADAM optimizer [5] with learning rate 0.0005. During the alternating optimization, we perform 5 update steps of the auxiliary decoder for each update step of the appearance model. Further, we set $I_{\text{KL}} = 1000$, $\gamma_{\text{C}} = 1, \lambda_k = 1 \,\forall\, k$ and use no inplane normalization [3].

DeepFashion On DeepFashion [7], we train our appearance model for 200k iterations using ADAM optimizer [5] with learning rate 0.0005. During the alternating optimization, we perform 10 update steps of the auxiliary decoder for each update step of the appearance model. Further, we set $I_{\text{KL}} = 1000$, $\gamma_{\text{C}} = 5, \lambda_k = 1 \,\forall\, k$ and use inplane normalization [3].

Market On Market [15], we train our appearance model for 150k iterations using ADAM optimizer [5] with learning rate 0.0005. During the alternating optimization, we perform 5 update steps of the auxiliary decoder for each update step of the appearance model. Further, we set $I_{\text{KL}} = 1000$, $\gamma_{\text{C}} = 1, \lambda_k = 1 \,\forall\, k$ and use inplane normalization [3].

B.3. Additional Results

To evaluate our model for appearance transfer also on established datasets dedicated to this task, we report in Fig. 1 Inception Score (IS) [12] and Structured Similarity (SSIM) [14] on DeepFashion [7] and Market1501 [15] dataset. We observe that our model performs competitively with the state-of-the-art on human shape-appearance transfer, thus indicating the general applicability of our disentanglement framework. Moreover, in Fig. 8 we provide example appearance transfers. Top rows depict the target posture and leftmost columns depict the source appearance. Similarly, in Fig. 9 we show transfers between posture and appearance for the Market1501 [15] dataset.

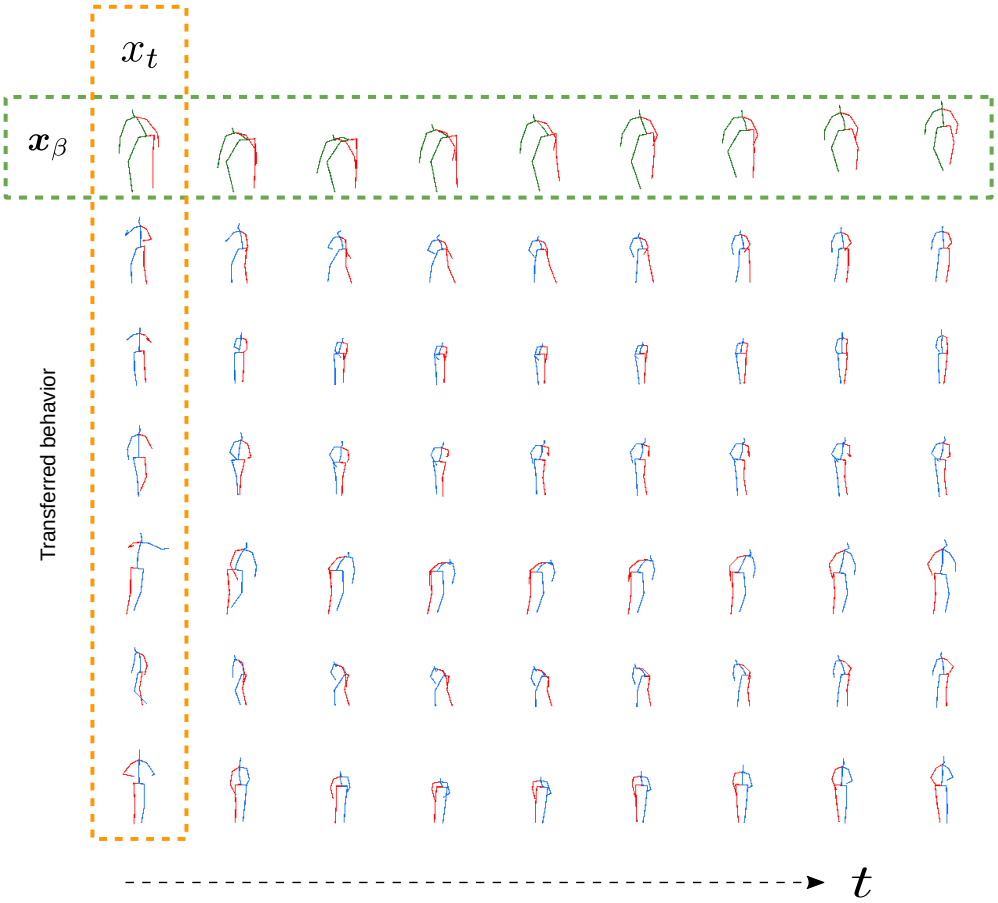


Figure 2. *Behavior Transfer on Human3.6m [4].* We transfer fine-grained, characteristic body dynamics of an observed behavior x_β to unrelated, significantly different target postures x_t. Best viewed in PDF when zoomed in.

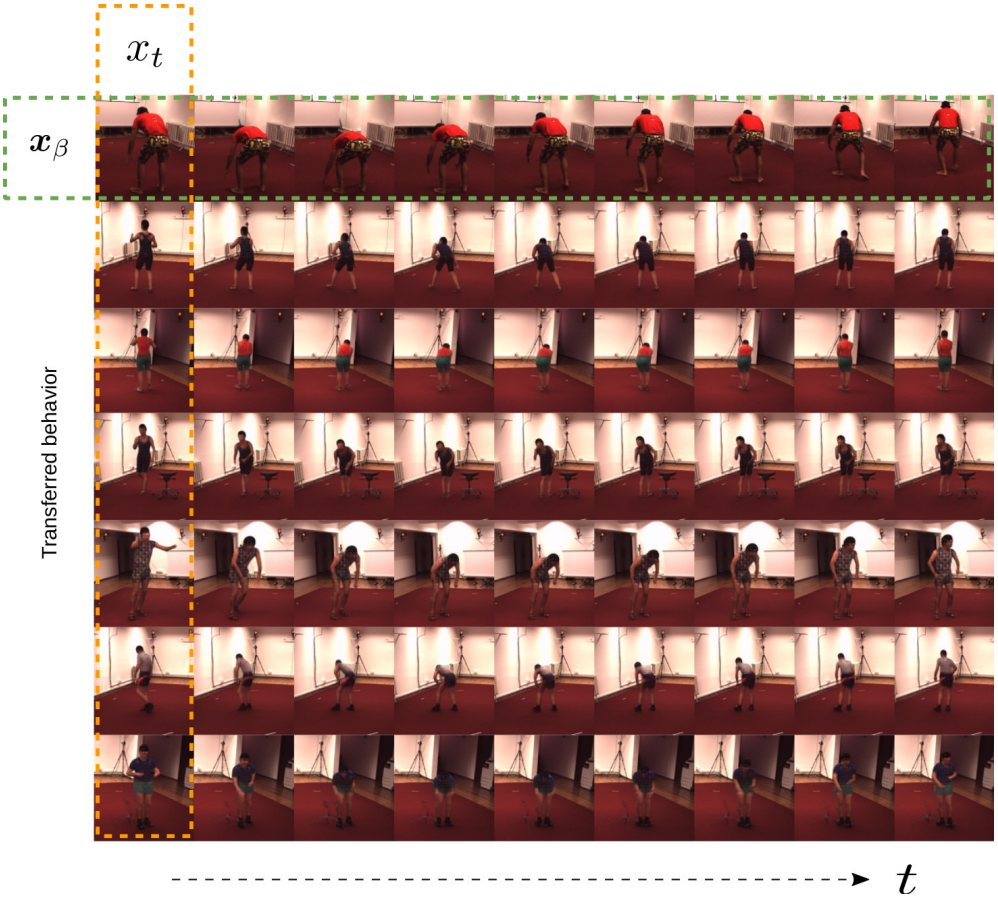


Figure 3. *Translation of Fig. 2 to RGB images*. We transfer fine-grained, characteristic body dynamics of an observed behavior $\boldsymbol{x}_\beta$ to unrelated, significantly different target postures x_t. Best viewed in PDF when zoomed in.

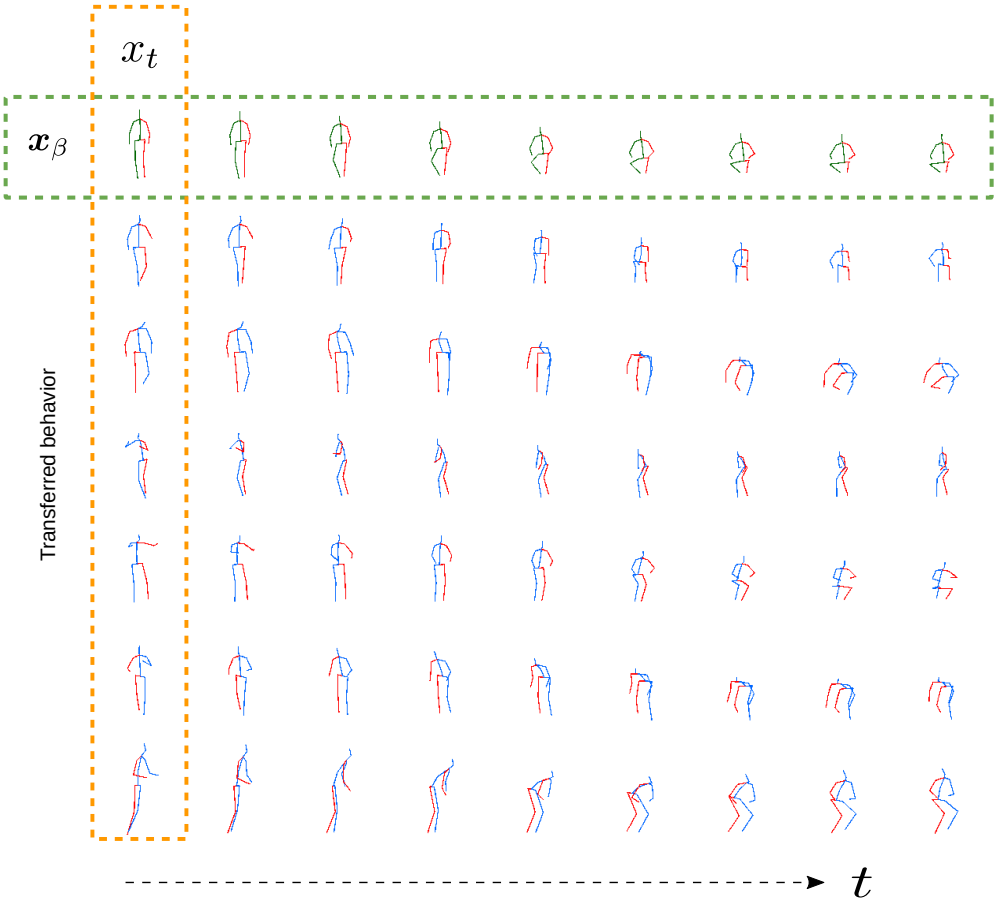


Figure 4. *Behavior Transfer on Human3.6m [4].* We transfer fine-grained, characteristic body dynamics of an observed behavior x_β to unrelated, significantly different target postures x_t. Best viewed in PDF when zoomed in.

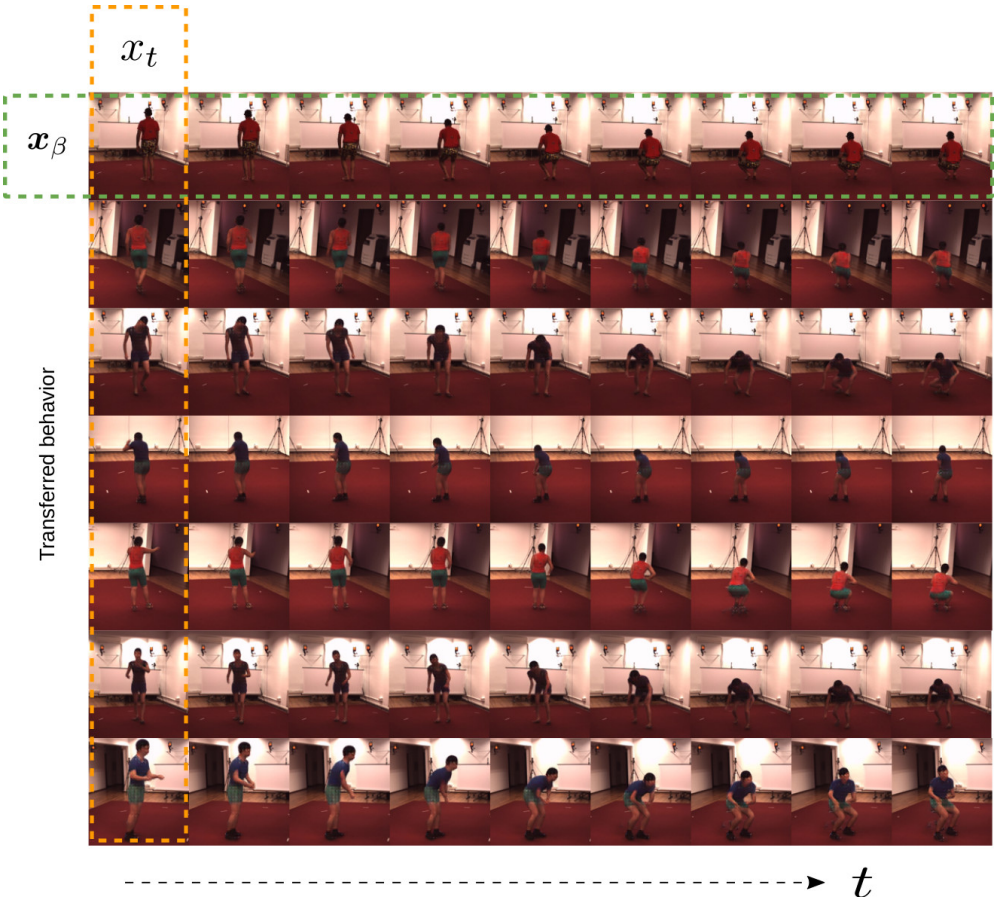


Figure 5. *Translation of Fig. 4 to RGB images*. We transfer fine-grained, characteristic body dynamics of an observed behavior $\boldsymbol{x}_\beta$ to unrelated, significantly different target postures x_t. Best viewed in PDF when zoomed in.

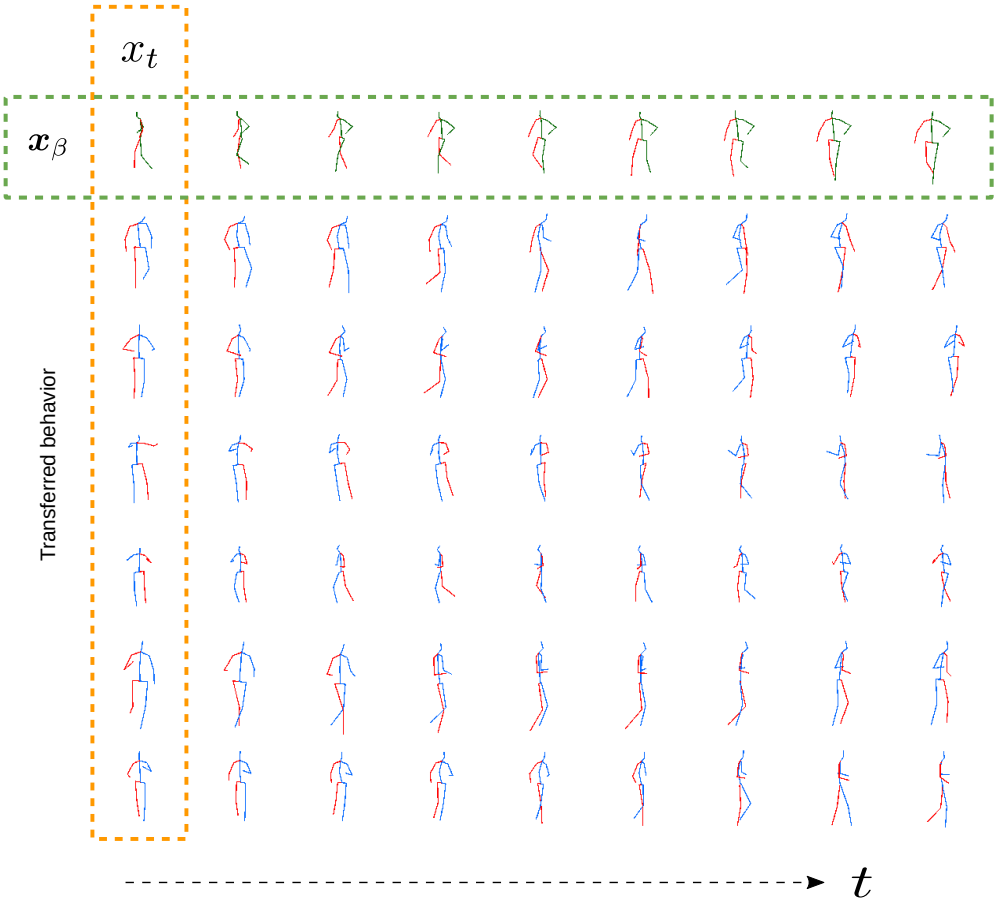


Figure 6. *Behavior Transfer on Human3.6m [4].* We transfer fine-grained, characteristic body dynamics of an observed behavior x_β to unrelated, significantly different target postures x_t. Best viewed in PDF when zoomed in.

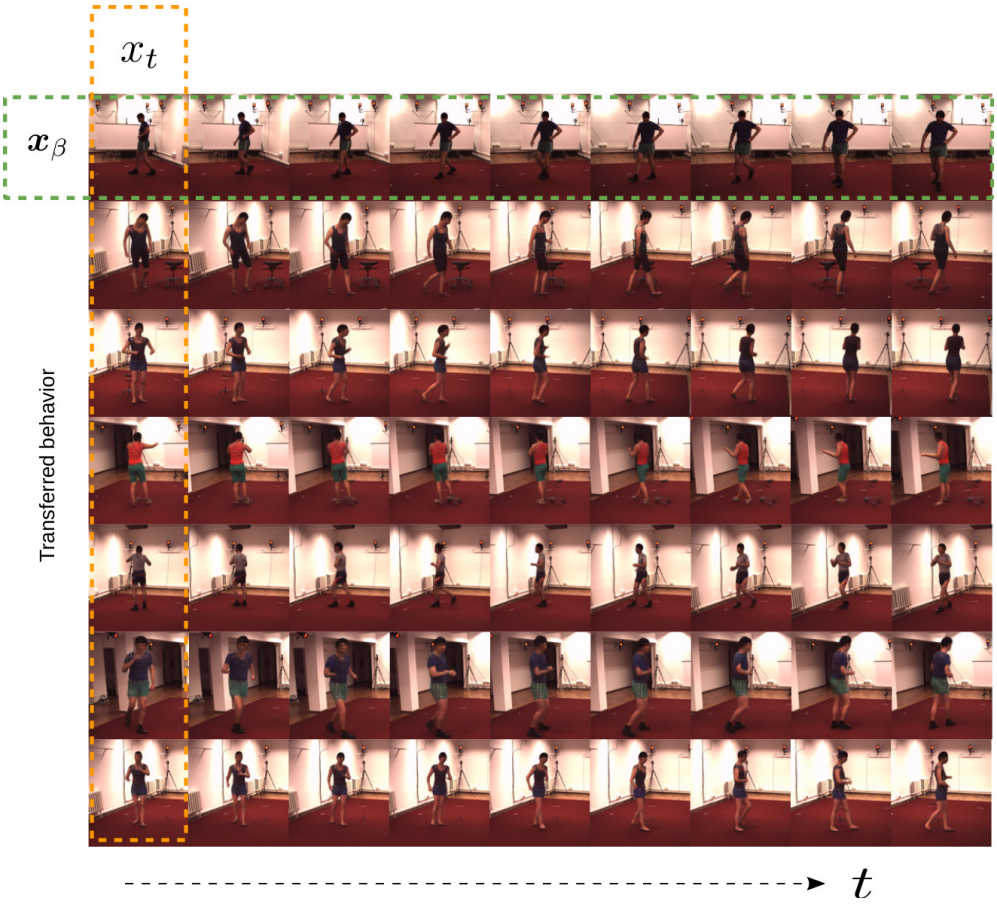


Figure 7. *Translation of Fig. 6 to RGB images*. We transfer fine-grained, characteristic body dynamics of an observed behavior $\boldsymbol{x}_\beta$ to unrelated, significantly different target postures x_t. Best viewed in PDF when zoomed in.

Figure 8. Posture-Appearance transfer on *DeepFashion* [7]. Top row depicts target posture and leftmost row depicts source appearance.

Figure 9. Posture-Appearance transfer on *Market1501* [15]. Top row depicts target posture and leftmost row depicts source appearance.

References

[1] Mohammad Sadegh Aliakbarian, Fatemeh Sadat Saleh, Mathieu Salzmann, Lars Petersson, Stephen Gould, and Amirhossein Habibian. Learning variations in human motion via mix-and-match perturbation, 2019.

[2] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. 2017.

[3] Patrick Esser, Ekaterina Sutter, and Björn Ommer. A variational u-net for conditional appearance and shape generation. 2018.

[4] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.

[5] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[6] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems 31*, 2018.

[7] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[8] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *Advances in Neural Information Processing Systems*. 2017.

[9] Liqian Ma, Qianru Sun, Stamatios Georgoulis, Luc Van Gool, Bernt Schiele, and Mario Fritz. Disentangled person image generation. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[10] Michael F Mathieu, Junbo Jake Zhao, Junbo Zhao, Aditya Ramesh, Pablo Sprechmann, and Yann LeCun. Disentangling factors of variation in deep representation using adversarial training. In *Advances in Neural Information Processing Systems*. 2016.

[11] Leland McInnes and John Healy. Umap: Uniform manifold approximation and projection for dimension reduction. 2018.

[12] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems 29*. 2016.

[13] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[14] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 2004.

[15] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, page 1116–1124, USA, 2015. IEEE Computer Society.