Supplementary Material: Action-Conditioned 3D Human Motion Synthesis with Transformer VAE

Mathis Petrovich[1] Michael J. Black[2] Gül Varol[1]
[1] LIGM, École des Ponts, Univ Gustave Eiffel, CNRS, France
[2] Max Planck Institute for Intelligent Systems, Tübingen, Germany
{mathis.petrovich,gul.varol}@enpc.fr, black@tue.mpg.de
https://imagine.enpc.fr/~petrovim/actor

This document provides additional experiments (Section A), additional qualitative results (Section B), and implementation details (Section C).

A. Additional experiments

We ablate the model and vary key parameters to evaluate the influence of design choices on the quality of the results. In particular, we present results on the effect of λ_{KL} (Section A.1), batch size (Section A.2), number of layers (Section A.3), and the rotation representation for SMPL pose parameters (Section A.4).

A.1. Weight of the KL loss

As explained in Section 3.2 of the main paper, we empirically show the importance of the weighting between the reconstruction loss and the KL loss, controlled by λ_{KL}. Table A.1 presents results for several values of λ_{KL} and we find that there is a trade-off between diversity and realism that is best balanced at $\lambda_{KL} = 1\text{e}{-5}$. We use this value in all our experiments.

A.2. Influence of the batch size

As pointed out in Section 3.2 of the main paper, we find that the batch size significantly influences the performance. In Table A.2, we report results with batch sizes of 10, 20, 30, 40 for a fixed learning rate. The best performance is obtained at 20, which is used in all our experiments.

A.3. Number of layers

We experiment with the number of Transformer layers in both of our encoder and decoder architectures. Table A.3 summarizes the results. While 2 and 4 layers are sub-optimal, the performance difference between 6 and 8 layers is minimal. We use 8 layers in all our experiments.

A.4. SMPL pose parameter representation

In Table A.4, we explore different rotation representations for SMPL pose parameters. Note that we also preserve the loss on the vertices $\mathcal{L}_V$ in all rows. We find that an axis-angle representation is difficult to train due to discontinuities, while others, such as quaternions, rotation matrices and 6D continuous representations [9] are similar in performance on NTU-13. On UESTC, we obtain the best performance with the 6D representation and use this in all our experiments.

B. Additional qualitative results

Figure A.1 demonstrates the diversity of our generated motions for additional actions on NTU-13 and UESTC.

Video. We provide a supplemental video at [8] to illustrate qualitatively the diversity in our generations and compare with Action2Motion [3]. Moreover, we visualize the effect of using a combined reconstruction loss defined both on rotations and vertex coordinates, as opposed to a single loss. We further present results of changing the duration of the generations. We also inspect the latent space by interpolating the noise vector. Finally, we present the denoising capability of our model by encoding-decoding through our latent space. This takes jerky motions and produces smooth but natural looking motion.

Jitter removal for Action2Motion [3]. Besides the quantitative improvement of ACTOR over Action2Motion, we observe qualitatively that Action2Motion generations have significant temporal jitter. To investigate whether our improvement stems from this difference, we removed jitter (using 1€ filter) from Action2Motion generations (that we obtained with their code). The result becomes worse (FID: 0.41 $\rightarrow$ 0.63, Acc: 94.3% $\rightarrow$ 93.0%)[1], perhaps because the real data also has considerable jitter. This suggests that our significant quantitative improvement can be attributed to other factors such as more distinguishable actions.

C. Implementation details

Architectural details. For all our experiments, we set the embedding dimensionality to 256. In the Transformer, we set the number of layers to 8, the number of heads in multi-head attention to 4, the dropout rate to 0.1 and the dimension of the intermediate feedforward network to 1024. As

[1]These two values for Action2Motion does not match Table 3 of the paper because we use our own evaluation script and normalized the ground truth shape by taking the average shape of SMPL.

in GPT-3 [1] and BERT [2], we use Gaussian Linear Error Units (GELU) [4] in our Transformer architecture.

Library credits. Our models are implemented with PyTorch [6], and we use PyTorch3D [5] to perform differentiable conversion between rotation representations. We integrate the differentiable SMPL layer using the PyTorch implementation of SMPL-X [7].

Metrics. For the evaluation metrics, we use the implementations provided by Action2Motion [3].

Runtime. Training takes 24 hours for 2K epochs on NTU, 19h hours for 5K epochs on HumanAct12, and 33 hours for 1K epochs on UESTC on a single Tesla V100 GPU, using 4GB GPU memory with batch size 20.

Training with sequences of variable durations. As explained in Section 4.2 of the main paper, we finetune our model with variable-durations after pretraining on fixed-durations. For this, we restore the model weights from the fixed-duration pretraining and finetune for 100 additional epochs, with the same training hyperparameters.

	UESTC					NTU-13			
	FID$_{tr}\downarrow$	FID$_{test}\downarrow$	Acc.$\uparrow$	Div.$\rightarrow$	Multimod.$\rightarrow$	FID$_{tr}\downarrow$	Acc.$\uparrow$	Div.$\rightarrow$	Multimod.$\rightarrow$
Real	$2.93^{\pm0.26}$	$2.79^{\pm0.29}$	$98.8^{\pm0.1}$	$33.34^{\pm0.32}$	$14.16^{\pm0.06}$	$0.02^{\pm0.00}$	$99.8^{\pm0.0}$	$7.07^{\pm0.02}$	$2.27^{\pm0.01}$
$\lambda_{KL} = 1e{-}3$	$460.72^{\pm90.36}$	$490.12^{\pm36.10}$	$34.4^{\pm1.4}$	$20.69^{\pm0.60}$	$1.25^{\pm0.00}$	$13.79^{\pm0.03}$	$46.6^{\pm0.7}$	$5.79^{\pm0.04}$	$1.53^{\pm0.01}$
$\lambda_{KL} = 1e{-}4$	$367.95^{\pm94.07}$	$390.68^{\pm41.02}$	$38.1^{\pm0.9}$	$20.91^{\pm0.38}$	$9.19^{\pm0.08}$	$9.90^{\pm0.02}$	$50.3^{\pm1.0}$	$6.15^{\pm0.04}$	$2.86^{\pm0.02}$
$\lambda_{KL} = 1e{-}5$	$20.02^{\pm1.79}$	$23.64^{\pm3.59}$	$90.5^{\pm0.4}$	$32.77^{\pm0.48}$	$14.64^{\pm0.07}$	$0.17^{\pm0.00}$	$96.4^{\pm0.2}$	$7.08^{\pm0.03}$	$2.12^{\pm0.01}$
$\lambda_{KL} = 1e{-}6$	$34.13^{\pm5.52}$	$39.74^{\pm3.57}$	$77.4^{\pm0.8}$	$29.60^{\pm0.35}$	$18.08^{\pm0.08}$	$13.83^{\pm0.03}$	$46.6^{\pm0.7}$	$5.78^{\pm0.04}$	$1.54^{\pm0.01}$
$\lambda_{KL} = 1e{-}7$	$80.05^{\pm7.71}$	$83.68^{\pm12.55}$	$47.1^{\pm2.1}$	$25.06^{\pm0.15}$	$19.96^{\pm0.08}$	$7.04^{\pm0.03}$	$43.0^{\pm2.1}$	$6.17^{\pm0.03}$	$4.18^{\pm0.01}$

Table A.1: **Weighting the KL loss term:** To obtain a good trade-off between diversity and realism, it is important to find the balance between the reconstruction loss term and the KL loss term in training. We set the weight λ_{KL} to $1e{-}5$ in our training.

	UESTC					NTU-13			
	FID$_{tr}\downarrow$	FID$_{test}\downarrow$	Acc.$\uparrow$	Div.$\rightarrow$	Multimod.$\rightarrow$	FID$_{tr}\downarrow$	Acc.$\uparrow$	Div.$\rightarrow$	Multimod.$\rightarrow$
Real	$2.93^{\pm0.26}$	$2.79^{\pm0.29}$	$98.8^{\pm0.1}$	$33.34^{\pm0.32}$	$14.16^{\pm0.06}$	$0.02^{\pm0.00}$	$99.8^{\pm0.0}$	$7.07^{\pm0.02}$	$2.27^{\pm0.01}$
Batch size = 10	$283.28^{\pm94.40}$	$309.15^{\pm33.90}$	$39.7^{\pm1.5}$	$23.24^{\pm0.43}$	$15.73^{\pm0.11}$	$13.95^{\pm0.03}$	$46.2^{\pm0.6}$	$5.77^{\pm0.05}$	$1.56^{\pm0.01}$
Batch size = 20	$20.02^{\pm1.79}$	$23.64^{\pm3.59}$	$90.5^{\pm0.4}$	$32.77^{\pm0.48}$	$14.64^{\pm0.07}$	$0.17^{\pm0.00}$	$96.4^{\pm0.2}$	$7.08^{\pm0.03}$	$2.12^{\pm0.01}$
Batch size = 30	$23.37^{\pm2.95}$	$26.06^{\pm1.28}$	$89.7^{\pm0.5}$	$32.07^{\pm0.58}$	$14.59^{\pm0.05}$	$0.18^{\pm0.00}$	$96.2^{\pm0.2}$	$7.07^{\pm0.04}$	$2.13^{\pm0.01}$
Batch size = 40	$25.36^{\pm1.82}$	$28.22^{\pm2.16}$	$89.2^{\pm0.7}$	$32.22^{\pm0.44}$	$14.52^{\pm0.10}$	$0.26^{\pm0.00}$	$95.4^{\pm0.1}$	$7.06^{\pm0.05}$	$2.10^{\pm0.01}$

Table A.2: **Batch size:** We observe sensitivity of the Transformer VAE training to different batch sizes and report performances at several batch size values. We set this hyperparameter to 20 in our training.

	UESTC					NTU-13			
	FID$_{tr}\downarrow$	FID$_{test}\downarrow$	Acc.$\uparrow$	Div.$\rightarrow$	Multimod.$\rightarrow$	FID$_{tr}\downarrow$	Acc.$\uparrow$	Div.$\rightarrow$	Multimod.$\rightarrow$
Real	$2.93^{\pm0.26}$	$2.79^{\pm0.29}$	$98.8^{\pm0.1}$	$33.34^{\pm0.32}$	$14.16^{\pm0.06}$	$0.02^{\pm0.00}$	$99.8^{\pm0.0}$	$7.07^{\pm0.02}$	$2.27^{\pm0.01}$
2-layers	$34.66^{\pm2.58}$	$37.17^{\pm3.53}$	$84.9^{\pm0.6}$	$30.87^{\pm0.36}$	$15.83^{\pm0.08}$	$0.24^{\pm0.00}$	$94.6^{\pm0.2}$	$7.07^{\pm0.03}$	$2.22^{\pm0.01}$
4-layers	$23.93^{\pm1.50}$	$26.75^{\pm1.99}$	$88.9^{\pm0.5}$	$32.24^{\pm0.76}$	$15.06^{\pm0.06}$	$0.19^{\pm0.00}$	$96.1^{\pm0.2}$	$7.09^{\pm0.04}$	$2.10^{\pm0.01}$
6-layers	$21.68^{\pm1.78}$	$24.92^{\pm2.09}$	$89.0^{\pm0.6}$	$32.61^{\pm0.41}$	$15.31^{\pm0.05}$	$0.16^{\pm0.00}$	$96.6^{\pm0.1}$	$7.09^{\pm0.04}$	$2.11^{\pm0.01}$
8-layers	$20.02^{\pm1.79}$	$23.64^{\pm3.59}$	$90.5^{\pm0.4}$	$32.77^{\pm0.48}$	$14.64^{\pm0.07}$	$0.17^{\pm0.00}$	$96.4^{\pm0.2}$	$7.08^{\pm0.03}$	$2.12^{\pm0.01}$

Table A.3: **Number of layers:** We use 8 layers in both the encoder and the decoder of the Transformer VAE. While the performance degrades at 2 or 4 layers, we see marginal gains after 6 layers.

	UESTC					NTU-13			
	FID$_{tr}\downarrow$	FID$_{test}\downarrow$	Acc.$\uparrow$	Div.$\rightarrow$	Multimod.$\rightarrow$	FID$_{tr}\downarrow$	Acc.$\uparrow$	Div.$\rightarrow$	Multimod.$\rightarrow$
Real	$2.93^{\pm0.26}$	$2.79^{\pm0.29}$	$98.8^{\pm0.1}$	$33.34^{\pm0.32}$	$14.16^{\pm0.06}$	$0.02^{\pm0.00}$	$99.8^{\pm0.0}$	$7.07^{\pm0.02}$	$2.27^{\pm0.01}$
Axis-angle	$513.39^{\pm98.35}$	$531.88^{\pm43.41}$	$16.4^{\pm0.4}$	$19.75^{\pm0.44}$	$1.81^{\pm0.00}$	$14.98^{\pm0.03}$	$41.7^{\pm0.7}$	$5.29^{\pm0.02}$	$1.96^{\pm0.01}$
Quaternion	$281.9^{\pm87.5}$	$305.02^{\pm21.97}$	$41.2^{\pm1.0}$	$23.48^{\pm0.39}$	$14.57^{\pm0.06}$	$0.20^{\pm0.00}$	$95.6^{\pm0.3}$	$7.08^{\pm0.04}$	$2.23^{\pm0.01}$
Rotation matrix	$277.14^{\pm76.59}$	$300.29^{\pm29.53}$	$41.6^{\pm1.9}$	$22.25^{\pm0.30}$	$14.56^{\pm0.10}$	$0.17^{\pm0.00}$	$95.9^{\pm0.2}$	$7.08^{\pm0.04}$	$2.19^{\pm0.01}$
6D continuous	$20.02^{\pm1.79}$	$23.64^{\pm3.59}$	$90.5^{\pm0.4}$	$32.77^{\pm0.48}$	$14.64^{\pm0.07}$	$0.17^{\pm0.00}$	$96.4^{\pm0.2}$	$7.08^{\pm0.03}$	$2.12^{\pm0.01}$

Table A.4: **SMPL pose parameter representation:** We investigate different rotation representations for the SMPL pose parameters. While on NTU-13, all except axis-angle representations perform similarly, the best performing representation on UESTC is the 6D continuous representation [9]. Note that the action recognition model which is used for evaluation is based on 6D rotations on UESTC and joint coordinates on NTU-13. Therefore, we convert each generation to these representations before evaluation.

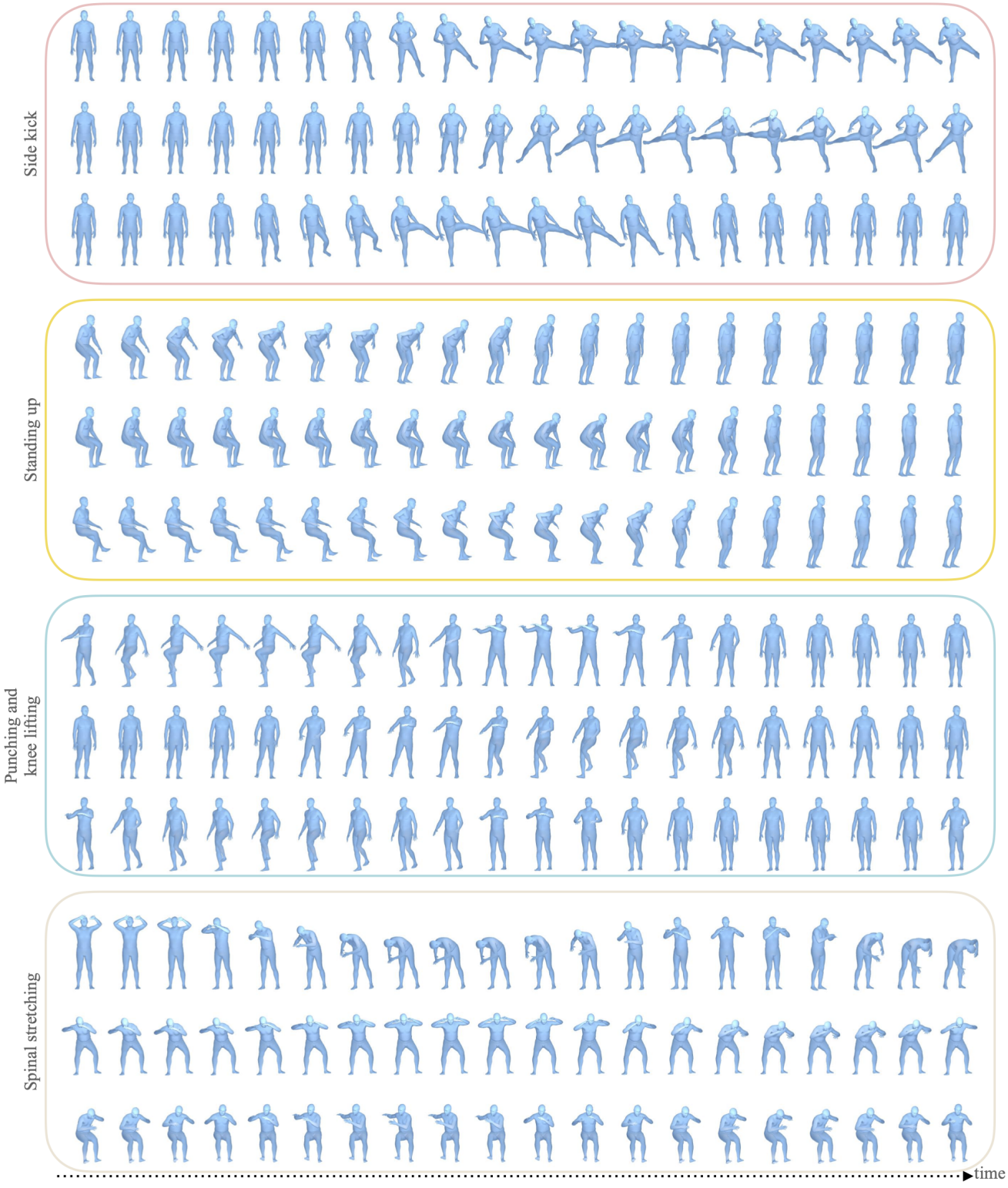

Figure A.1: **Additional qualitative results:** We provide more action categories from NTU-13 (top two actions: 'Side kick' and 'Standing up') and UESTC (bottom two actions: 'Punching and knee lifting' and 'Spinal stretching'). As in Figure 6 of the main paper, we show 3 generations per action. Our model generates different ways to perform the same action.

References

[1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 4171–4186, 2019. 2

[3] Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2motion: Conditioned generation of 3d human motions. In *ACM International Conference on Multimedia (ACMMM)*, pages 2021–2029, 2020. 1, 2

[4] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv:1606.08415*, 2016. 2

[5] Justin Johnson, Nikhila Ravi, Jeremy Reizenstein, David Novotny, Shubham Tulsiani, Christoph Lassner, and Steve Branson. Accelerating 3D deep learning with PyTorch3D. In *SIGGRAPH Asia 2020 Courses*, 2020. 2

[6] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 2

[7] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10967–10977, 2019. 2

[8] Mathis Petrovich, Michael J. Black, and Gül Varol. ACTOR project page: Action-conditioned 3D human motion synthesis with Transformer VAE. https://imagine.enpc.fr/~petrovim/actor. 1

[9] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 3