The Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI-23)

MultiAct: Long-Term 3D Human Motion Generation from Multiple Action Labels

Taeryung Lee[*1], **Gyeongsik Moon**[*3], **Kyoung Mu Lee** [1,2]

[1] IPAI, Seoul National University, Korea
[2] Dept. of ECE & ASRI, Seoul National University, Korea
[3] Meta Reality Labs Research
trlee94@snu.ac.kr, mks0601@meta.com, kyoungmu@snu.ac.kr

Abstract

We tackle the problem of generating long-term 3D human motion from multiple action labels. Two main previous approaches, such as action- and motion-conditioned methods, have limitations to solve this problem. The action-conditioned methods generate a sequence of motion from a single action. Hence, it cannot generate long-term motions composed of multiple actions and transitions between actions. Meanwhile, the motion-conditioned methods generate future motions from initial motion. The generated future motions only depend on the past, so they are not controllable by the user's desired actions. We present **MultiAct**, the first framework to generate long-term 3D human motion from multiple action labels. MultiAct takes account of both action and motion conditions with a unified recurrent generation system. It repetitively takes the previous motion and action label; then, it generates a smooth transition and the motion of the given action. As a result, MultiAct produces realistic long-term motion controlled by the given sequence of multiple action labels. Code is publicly available in https://github.com/TaeryungLee/MultiAct_RELEASE.

Introduction

Modeling and generation of realistic human motion play an essential role in computer vision and robotics, including automated avatars for AI assistant (Neuhaus et al. 2019), virtual reality (Ahuja et al. 2021) and human-robot interaction (Chan et al. 2021). However, despite decades of efforts to model human motions, generating controllable long-term 3D human motions remains a challenging problem.

Fig. 1 categorizes 3D human motion generation methods by conditions used in generation. The action-conditioned methods (Cai et al. 2018; Petrovich, Black, and Varol 2021; Guo et al. 2020) generate a short-term motion from an action label, and the motion-conditioned methods (Barsoum, Kender, and Liu 2018; Habibie et al. 2017; Yuan and Kitani 2020) generate future motion based on the previous motion.

However, both methods have limitations in solving our challenging target problem: *"How to generate realistic long-term motion controlled by multiple actions labels?"*. Action-conditioned methods can only produce the individual action motions, but not the realistic long-term motions composed of multiple actions and transitions between them. Simple linear interpolation between individually generated action motions can produce multiple-action motions. However, those interpolated transitions are unrealistic since they do not consider the adjoining motion context. On the other hand, most motion-conditioned methods cannot control the generated motions. Some works (Wang et al. 2021b,a; Cao et al. 2020) have tried to control the generated motion indirectly, but still, controlling the motion with a series of actions remains challenging. Simply combining above two methods together still does not handle the target problem: Producing an action motion with the action-conditioned method, and then generating the transition motion with the motion-conditioned method fails to generate realistic multiple-action motion since the generated transition motion is not guaranteed to be consistent with the following action motion. This limitation motivates us to handle both conditions in a unified model.

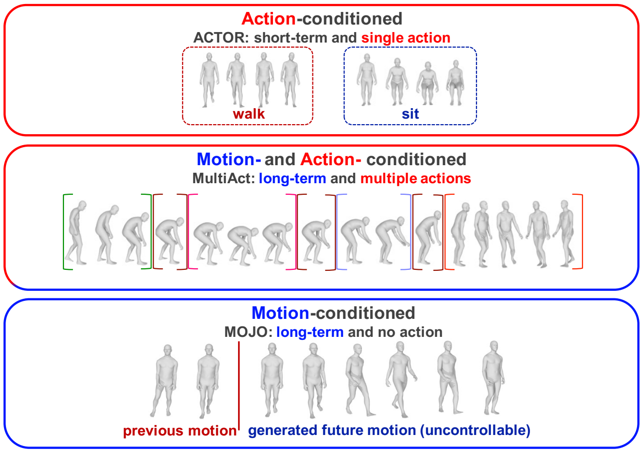

Action-conditioned
ACTOR: short-term and single action
walk
sit
Motion- and Action- conditioned
MultiAct: long-term and multiple actions
Motion-conditioned
MOJO: long-term and no action
previous motion
generated future motion (uncontrollable)

Figure 1: Categorization. We show long-term motions generated by our MultiAct (mid). Compared to ours, action-conditioned ACTOR (Petrovich, Black, and Varol 2021) (top) can only generate short-term single actions. Motion-conditioned MOJO (Zhang, Black, and Tang 2021) (bottom) cannot control the generated motion with desired actions. MultiAct handles both conditions in a single model to generate long-term motions of multiple actions.

We propose a novel recurrent framework, **MultiAct**, to

[*] These authors contributed equally.

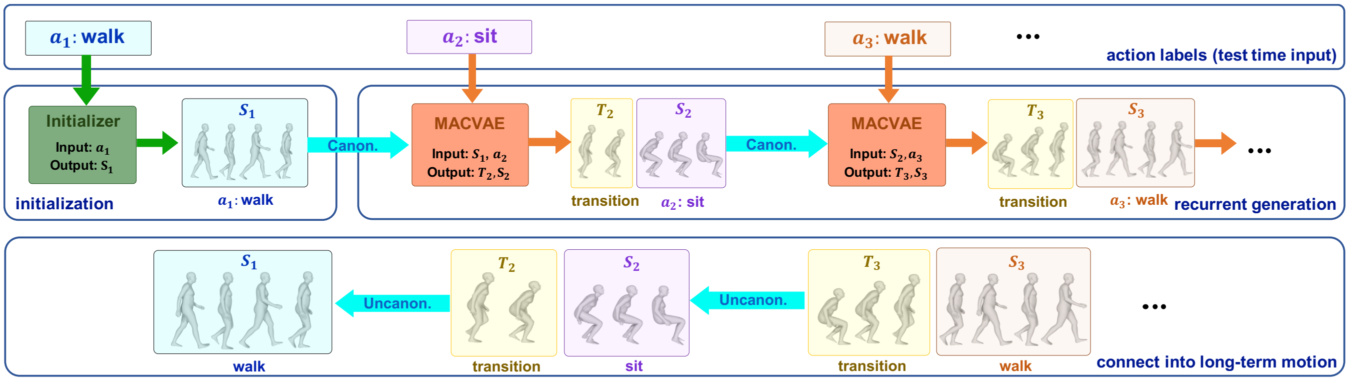


Figure 2: The overview. We introduce four main steps of MultiAct to generate long-term human motion from multiple action labels. MultiAct takes a series of action labels $(a_1, a_2, ...)$. We use action-conditioned model to generate the initial S_1 from a_1. We canonicalize the previous motion. Then, given the canonicalized previous motion S_{i-1} and next action label a_i of time step i, MACVAE generates the next motions (T_i, S_i). We bring each local motion pair to the global coordinate (*i.e.*, uncanonicalize) and connect it to the previous motion.

overcome the limitations of previous approaches by handling both conditions at once. Fig. 2 illustrates the overview of our framework. MultiAct recurrently passes the previous motion and a current action label to generate the transition and the current action motion. Fig. 1 shows an example of long-term motion from our model using the action label sequence *(bend down, grab and lift, put down, turn around)*.

Two critical challenges exist in recurrently generating long-term motions from a sequence of action labels. The first is, in each recurrent step, to generate motion that smoothly continues from the given previous motion while following the desired action. We resolve the first challenge with a novel recurrent module MACVAE (**M**otion- and **A**ction-**C**onditioned **VAE**). The core idea of MACVAE is to concurrently generate action motions and transitions from the joint condition of the action label and previous motion. As a result, the generated transition is aware of the context in both adjoining motions, which is not considered in simple interpolation techniques.

The second challenge is the ground geometry losing problem during the canonicalization (*i.e.*, normalization, *Abbr.* canon.). The canon. brings the previous motion into normalized form during training and testing, potentially ending in any location and facing any direction. Such a process relieves the burden of motion-conditioned models to learn highly varying input motion space.

However, previous zero-canon. (Zhang, Black, and Tang 2021) wipes out the global rotation that holds the geometry between the body and the ground. Losing the information about the ground geometry leads to physical implausibility during the recurrent generation. We adjust this problem with the face-front canon. to disentangle and retain only the relevant information from the input motion. The experiment supports that our face-front canon. is not an ad-hoc visualization technique but an irreplaceable input normalization method used in training, single-step, and long-term testing.

To the best of our knowledge, our work is the first approach to synthesizing unseen long-term 3D human motion from multiple action labels. We show that our MultiAct outperforms the best combination of previous SOTA methods to generate long-term motion from multiple action labels, besides handling such problem within a single model. The experimental comparison is conducted on the quality of action motion and transition in single-step and long-term generations. Our contributions can be summarized as follows.

- We propose MultiAct, a novel recurrent framework to generate long-term 3D human motion controlled by a sequence of action labels.
- Our MACVAE concurrently generates action motions and realistic transitions aware of adjoining motion context. Generated action motions and transitions are more realistic than previous SOTA methods.
- Our face-front canon. assures the local coordinate system of each recurrent step shares the ground geometry. We empirically validate the irreplaceability of the face-front canon. by qualitative and quantitative results.

Related Works

Motion-conditioned human motion generation. Early works regressed deterministic future motions (Aksan, Kaufmann, and Hilliges 2019; Fragkiadaki et al. 2015). Recently, stochastic approaches (Zhang, Black, and Tang 2020; Chen et al. 2020) show promising results with progress in conditional generative models (Sohn, Lee, and Yan 2015). HP-GAN (Barsoum, Kender, and Liu 2018), DLow (Yuan and Kitani 2020) and recurrent VAE (Habibie et al. 2017) predicted future motions by employing stochastic generative model. HuMoR (Rempe et al. 2021) and HM-VAE (Li et al. 2021) trained VAE to optimize the mesh estimation on sparse observations. (Mao, Liu, and Salzmann 2021) proposed a model fixing the motion of the partial body and generating the motion for the remaining part. In contrast to the previous methods that cannot control the motion (Yuan and Kitani 2020; Li et al. 2021) or indirectly controls (Mao, Liu,

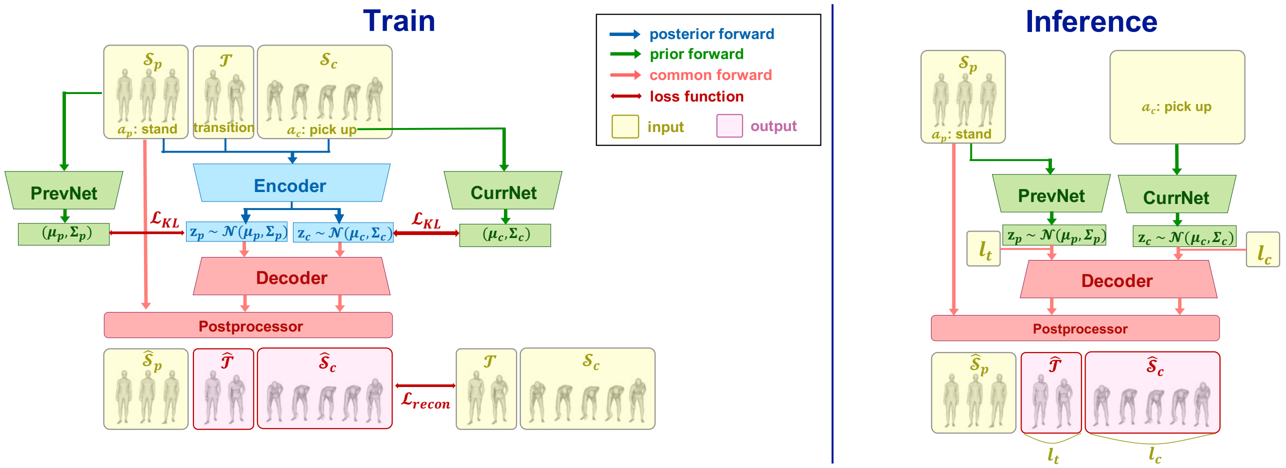


Figure 3: Model. Our model MACVAE aims to generate continuous transition and action motion $[\hat{\mathcal{T}},\ \hat{\mathcal{S}_c}]$ from previous action motion $\mathcal{S}_p$ and action label $\mathfrak{a}_c$ at inference time. For training, MACVAE encodes the training input $[\mathcal{S}_p,\ \mathcal{T},\ \mathcal{S}_c]$, $\mathfrak{a}_p$ and $\mathfrak{a}_c$ into posterior distribution of latent vectors z_p and z_c (blue). Through common forward of decoder and postprocessor, we reconstruct desired output $[\hat{\mathcal{T}},\ \hat{\mathcal{S}_c}]$ (pink). Since we do not have input motion $[\mathcal{S}_p,\ \mathcal{T},\ \mathcal{S}_c]$ at the inference time, we sample z_p and z_c using prior networks of PrevNet and CurrNet, respectively (green). We minimize divergence and reconstruction errors in training.

and Salzmann 2021), our method directly controls the generated motions by action labels while inheriting the spirit to generate the future from the past.

Action-conditioned human motion generation. (Cai et al. 2018) have presented GAN-based action-conditioned 2D human motion generation. More recently, MUGL (Maheshwari, Gupta, and Sarvadevabhatla 2021), Action2Motion (Guo et al. 2020) and ACTOR (Petrovich, Black, and Varol 2021) proposed VAE-based action-conditioned 3D human motion generation models. While such methods are limited to generating individual short-term motion of single action, our MultiAct can generate the long-term motion of multiple actions from joint conditions of motion and action. PSGAN (Yang et al. 2018) is the 2D skeleton motion generating model conditioned on both the initial pose and an action label. In contrast to PSGAN, which generates single-action 2D motion in pixel space, MultiAct generates motion of multiple actions in 3D space.

Motion in-betweening. In-betweening (Harvey et al. 2020; Zhou et al. 2020; Duan et al. 2021) models generate the transition connecting two given motions. Especially, the SSMCT (Duan et al. 2021) is known to be the SOTA model for in-betweening. Since our work is the first proposed method to generate long-term motion from multiple actions, we combine previous SOTA methods ACTOR (Petrovich, Black, and Varol 2021) and SSMCT (Duan et al. 2021) into a unified framework as a baseline to compare the quality of generated long-term multiple action motion.

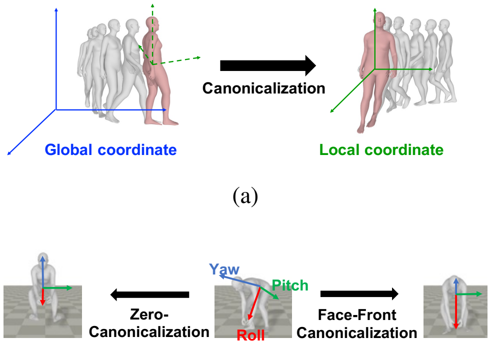
Text-conditioned human motion generation. A series of methods (Ahn et al. 2018; Stoll et al. 2020; Lin et al. 2018), including two concurrent works (Guo et al. 2022; Petrovich, Black, and Varol 2022) generate the human motion from given text. Text-conditioned methods map the text-described continuous semantic space into the motion space using the language models. On the other hand, the motivation of our model is to produce the smoothly connected long-term motion precisely from the sequence of discrete action labels.

MACVAE

Fig. 3 shows the overview of MACVAE. MACVAE is the recurrent unit of the overall framework MultiAct, used for generating the motion pair of (transition, action motion). For the training, MACVAE takes the continuous motion $[\mathcal{S}_p,\ \mathcal{T},\ \mathcal{S}_c]$ in length of L frames, and action labels $\mathfrak{a}_p$ and $\mathfrak{a}_c$. We note the previous and current action motions to $\mathcal{S}_p$ and $\mathcal{S}_c$, respectively, with a transition $\mathcal{T}$ between them. The action labels of $\mathcal{S}_p$ and $\mathcal{S}_c$ are denoted to $\mathfrak{a}_p, \mathfrak{a}_c \in A \subset \mathbb{Z}^+$, respectively. From inputs, MACVAE is trained to reconstruct continuous transition and action motion $[\hat{\mathcal{T}},\ \hat{\mathcal{S}_c}]$, where the hat notation denotes generated output, targeting to be close to GT $[\mathcal{T},\ \mathcal{S}_c]$. The inference stage of MACVAE takes previous action motion $\mathcal{S}_p$, an action label $\mathfrak{a}_c$, desired motion lengths l_t and l_c to generate future motion $[\hat{\mathcal{T}},\ \hat{\mathcal{S}_c}]$. The generated $\hat{\mathcal{S}_c}$ is a motion that belongs to the given action label $\mathfrak{a}_c$, and generated $\hat{\mathcal{T}}$ smoothly connects in between $\mathcal{S}_p$ and $\hat{\mathcal{S}_c}$.

Inputs and Outputs

3D human motion representation. We represent 3D human motion of length l as a sequence of 3D human pose representations $(\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_l) \in \mathbb{R}^{315 \times l}$. We note the pose representation of ith frame to $\mathbf{p}_i = (\mathbf{r}_i, \mathrm{vec}(\theta_i), \mathbf{x}_i) \in \mathbb{R}^{315}$, a concatenation of global rotation $\mathbf{r}_i \in \mathbb{R}^6$, 3D joint rotations $\theta_i \in \mathbb{R}^{51 \times 6}$ and 3D translation $\mathbf{x}_i \in \mathbb{R}^3$. Rotations $\mathbf{r}_i$ and θ_i respectively represent one global rotation of the human body and the other 51 rotations of human joints, defined in the SMPL-H body model (Romero, Tzionas, and Black 2017),



(a)

(b)

Figure 4: Canonicalization. Fig. 4a illustrates the face-front canon. of the previous motion. The last frame (red) is facing front after canon. Fig. 4b visually compares the face-front and zero-canon.

in 6D rotation (Zhou et al. 2019). The 3D translation $\mathbf{x}$ provides the displacement of the root joint. Furthermore, 3D mesh vertices V_i and joint coordinates J_i can be obtained by forwarding the pose $\mathbf{p}_i$ to the differentiable SMPL-H layer.

Face-front canonicalization of motions. Applying the face-front canon. to pre-canonicalized motion sequence $\{\mathbf{p}'_i = (\mathbf{r}'_i, \mathrm{vec}(\theta'_i), \mathbf{x}'_i)\}_{i=1}^{l}$ with respect to anchor frame index i_* formulates the canonicalized motion $\{\mathbf{p}_i\}_{i=1}^{l}$ in local coordinate system as following, where the local coordinate system is defined as 3D space of each recurrent step:

1. Apply rotation function $f(\mathbf{r}'_i; \mathbf{r}'_{i_*}) = \mathbf{r}_i$ on global rotation $\mathbf{r}'_i$ of each frame i in $\mathbf{p}'_i$. The function $f(*; \mathbf{r}'_{i_*}) : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is a fixed rotation function uniquely determined by the global rotation $\mathbf{r}'_{i_*} = E(\alpha, \beta, \gamma)$ of anchor frame i_*, where $E(*, *, *)$ means the ZYX-Euler angle representation of rotation. The formulation of the function is $f(\mathbf{r}_{i_*}; \mathbf{r}_{i_*}) = E(\alpha, \beta, 0)$.
2. Convert the global translation into relative translation: $\mathbf{x}'_{\mathrm{rel,i}} = \mathbf{x}'_{\mathbf{i}} - \mathbf{x}'_{\mathbf{i}_*}$. Then, we transform the relative translation $\mathbf{x}'_{\mathrm{rel,i}}$ into translation as the local coordinate $\mathbf{x}_{\mathbf{i}}$ applying the rotation $f(\mathbf{0}; \mathbf{r}'_{i_*})$. (*i.e.*, align the trajectory with the viewing direction).
3. Formulate the canonicalized motion $\{\mathbf{p}_i\}_{i=1}^{l}$ with $\{\mathbf{r}_i\}_{i=1}^{l}$ and $\{\mathbf{x}_{\mathbf{i}}\}_{i=1}^{l}$, while keeping the other 51 local joint rotations $\{\theta_i\}_{i=1}^{l}$ the same with $\{\theta'_i\}_{i=1}^{l}$.

During the training phase, we canonicalize $[\mathcal{S}_p,\ \mathcal{T},\ \mathcal{S}_c]$, so we put $l = l_p + l_t + l_c$ and anchor frame $i_* = l_p$, where l_p, l_t, l_c is the respective length of $\mathcal{S}_p$, $\mathcal{T}$ and $\mathcal{S}_c$. For the generation phase, we only canonicalize the previous motion $\mathcal{S}_p$, thus we use $l = l_p$ and anchor frame $i_* = l_p$. In summary, face-front canonicalized motion faces front at frame l_p by making the *yaw* rotation in ZYX-Euler angle to zero. Fig. 4a shows the visible result of canon. on previous motion.

For comparison, we use the zero-canon. (Zhang, Black, and Tang 2021) that shares all but rotation function $f_{\mathrm{zero}}(\mathbf{r}'_{i_*}; \mathbf{r}'_{i_*}) = E(0, 0, 0)$ instead of $f(\mathbf{r}'_{i_*}; \mathbf{r}'_{i_*}) = E(\alpha, \beta, 0)$. Fig. 4b shows the visual comparison of face-front and zero-canon. Our method normalizes the input motion, preserving the information about how the human body is bent toward the floor. To that end, our face-front canon. guides MACVAE to generate motions with consistent ground geometry. Effects of the canon. methods are tested and visualized in the Ablation section.

Architecture

The key idea of MACVAE is to explicitly model the previous motion $\mathcal{S}_p$ and current motion $\mathcal{S}_c$ by embedding them into separate latent vectors z_p and z_c, respectively. To this end, we employ a CVAE architecture (Sohn, Lee, and Yan 2015), which consists of an encoder, decoder, and additionally with two simple prior networks (Wang and Wan 2019): CurrNet and PrevNet. Encoder encodes $[\mathcal{S}_p,\ \mathcal{T},\ \mathcal{S}_c]$, $\mathfrak{a}_p$ and $\mathfrak{a}_c$ into the posterior distribution parameters of latent vectors $z_p, z_c \in \mathbb{R}^d$, where $d = 512$ is an inner dimension of Transformer. PrevNet and CurrNet estimate the prior distribution of latent vectors from only the test time inputs. Decoder and postprocessor reconstructs the generation target $\hat{\mathcal{T}}$ and $\hat{\mathcal{S}_c}$ from the latent vectors z_p and z_c.

Encoder. Encoder encodes all the training input $[\mathcal{S}_p,\ \mathcal{T},\ \mathcal{S}_c]$, $\mathfrak{a}_p$ and $\mathfrak{a}_c$ into parameters of Gaussian posterior distributions: μ_p^{post}, μ_c^{post}, Σ_p^{post} and Σ_c^{post}. Estimated parameters are used to sample latent vectors $z_p \sim \mathcal{N}(\mu_p^{\mathrm{post}},\ \Sigma_p^{\mathrm{post}\,2})$ and $z_c \sim \mathcal{N}(\mu_c^{\mathrm{post}},\ \Sigma_c^{\mathrm{post}\,2})$ by reparameterization trick (Kingma and Welling 2014). Transformer encoder primarily embeds the training inputs $[\mathcal{S}_p,\ \mathcal{T},\ \mathcal{S}_c] \in \mathbb{R}^{L \times 315}$, $\mathfrak{a}_p$, and $\mathfrak{a}_c$ into 2D tensor of shape $\mathbb{R}^{L \times d}$. For each frame $i \in \{1, ..., L\}$, we linearly embed the pose $\mathbf{p}_i$ into vectors of dimension $\mathbb{R}^{d/2}$. At the same time, we learn the embedding of dimension $\mathbb{R}^{d/2}$ for each action $\mathfrak{a} \in A$ and assign the action embedding to each frame i, using the corresponding action label as an index. Those vectors are concatenated into a d-dimensional vector and stacked into a 2D tensor of shape $\mathbb{R}^{L \times d}$. Transformer layers encode the embedded inputs into a 2D tensor of shape $\mathbb{R}^{L \times d}$. We pass the encoded tensor through a temporal convolution layer, take a mean along time dimension into the vector of shape $\mathbb{R}^d$ and pass it into four separate output FC layers. Each output layers linearly estimate the parameters of Gaussian posterior μ_p^{post}, Σ_p^{post}, μ_c^{post} and Σ_c^{post}, respectively. We sample latent vectors z_p and z_c from estimated parameters, then pass them into the decoder.

Decoder. The decoder takes two latent vectors z_p, z_c, that hold the context of previous and following motions, respectively. It also takes desired length of reconstructed motion l_t, l_c. Then reconstructs $[\tilde{\mathcal{T}},\ \tilde{\mathcal{S}_c}]$ using the transformer decoder, where $\tilde{\mathcal{T}}$ and $\tilde{\mathcal{S}_c}$ denotes reconstructed transition and current motion, respectively. While training, we use the length of GT motions $\mathcal{T}, \mathcal{S}_c$ for l_t, l_c, respectively. We provide l_t and l_c as input for test time generation. The Transformer decoder takes three inputs: key, value, and query. We build both key and value by "expanding" the given latent vectors z_p and z_c into 2D tensor of shape $\mathbb{R}^{(l_t + l_c) \times d}$. To this end, we repetitively stack z_p for l_t times, then z_c for l_c times into 2D tensor for key and value. The sinusoidal positional encoding is used as a query. From inputs above, Transformer decoder outputs a sequence of $(l_t + l_c)$ vectors of dimension $\mathbb{R}^d$. We pass the decoded vectors through a temporal convolution layer, then linearly project each $\mathbb{R}^d$ dimension vector into poses of dimension $\mathbb{R}^{315}$. We group the first l_t 3D human poses into $\tilde{\mathcal{T}} \in \mathbb{R}^{l_t \times 315}$, and following l_c poses into $\tilde{\mathcal{S}_c} \in \mathbb{R}^{l_c \times 315}$. Putting $\tilde{\mathcal{T}}$ and $\tilde{\mathcal{S}_c}$ together, decoder outputs the reconstructed motion $[\tilde{\mathcal{T}},\ \tilde{\mathcal{S}_c}]$.

Postprocessor. Postprocessor is a single 2D convolution layer of kernel size (315×5) to smooth the gap between previous and generated motion. Postprocessor takes previous motion $\mathcal{S}_p$ and generated motion $[\tilde{\mathcal{T}},\ \tilde{\mathcal{S}_c}]$. We first concatenate them into 2D tensor $[\mathcal{S}_p,\ \tilde{\mathcal{T}},\ \tilde{\mathcal{S}_c}] \in \mathbb{R}^{L \times 315}$, and pass through temporal convolution layer. We discard the previous motion $\mathcal{S}_p$ part, and output the remaining $[\hat{\mathcal{T}},\ \hat{\mathcal{S}_c}]$ as a final generation.

Prior networks: CurrNet and PrevNet. For inference, prior networks are used instead of an encoder: to generate the Gaussian distribution of latent vectors z_p and z_c from the test time inputs, $\mathcal{S}_p$ and $\mathfrak{a}_c$, respectively. CurrNet assigns the learnable tokens $\mu_c^{\text{prior}}(\mathfrak{a}), \Sigma_c^{\text{prior}}(\mathfrak{a}) \in \mathbb{R}^d$ for each action $\mathfrak{a} \in A$, then outputs the embedded tokens $\mu_c^{\text{prior}}(\mathfrak{a}_c)$ and $\Sigma_c^{\text{prior}}(\mathfrak{a}_c)$ for given action label $\mathfrak{a}_c$. PrevNet takes the previous action motion $\mathcal{S}_p$ as input and estimates the parameters μ_p^{prior} and Σ_p^{prior}. Instead of using the entire $\mathcal{S}_p$, we only embed the last few frames of $\mathcal{S}_p$ into μ_p^{prior} and Σ_p^{prior} using a single linear layer, and add the learnable token of dimension $\mathbb{R}^d$ corresponding to *transition*. This is to prevent our model from being overfitted to seen previous motions $\mathcal{S}_p$, which could result in poor generalization to unseen previous motions.

CurrNet and PrevNet are designed to estimate the latent distribution from the test time input. However, they are divided into separate modules to impose the different roles to z_p and z_c: z_p to deliver accurate embedding of previous motions to the decoder, and z_c to provide a detailed description of the action label $\mathfrak{a}_c$. As we do not have $[\mathcal{S}_p,\ \mathcal{T},\ \mathcal{S}_c]$ at test phase, we need to sample z_p and z_c with only test time input $\mathcal{S}_p$ and $\mathfrak{a}_c$. Thus, we rely on prior networks to sample z_p and z_c at the testing phase, while the divergence between prior and posterior distributions is minimized in training.

Training Objectives

Reconstruction. Our first objective is to minimize L1 reconstruction losses $\mathcal{L}_V = \sum_{i=1}^{l_t+l_c} ||V_i - \hat{V}_i||_1$ and $\mathcal{L}_P = \sum_{i=1}^{l_t+l_c} ||\mathbf{p}_i - \hat{\mathbf{p}}_i||_1$. We denote the GT 3D mesh vertices and pose representation of i'th frame to V_i and $\mathbf{p}_i$, respectively. Similarly, the predicted mesh vertices and pose of i'th frame are denoted to $\hat{V}_i$ and $\hat{\mathbf{p}}_i$, respectively. In addition, we use mesh vertex acceleration loss $\mathcal{L}_{\text{acc}}$ to enhance the quality of the generated motion. Each terms are combined into unified reconstruction loss $\mathcal{L}_{\text{recon}} = \mathcal{L}_V + \mathcal{L}_P + \lambda_{\text{acc}}\mathcal{L}_{\text{acc}}$.

Minimizing divergence of distributions. The second objective is to match the prior and posterior distribution, measured by Kullback-Leibler divergence $\mathcal{L}_{\text{KL}}$. Minimizing $\mathcal{L}_{\text{KL}}$ leads our model to reconstruct $[\hat{\mathcal{T}},\ \hat{\mathcal{S}_c}]$ with mostly from the information in $\mathcal{S}_p$ and $\mathfrak{a}_c$. The total loss is the sum of reconstruction and divergence losses: $\mathcal{L} = \mathcal{L}_{\text{recon}} + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}}$.

MultiAct

Fig. 2 shows overview of MultiAct. MultiAct recurrently generates a long-term human motion $[S_1,\ T_2,\ S_2,\ ....,\ S_N]$ from series of action labels $(a_1, ..., a_N) \in A^n$, for the variable sequence length N. We use the recurrent cell MACVAE, which takes previous motion S_{t-1} and action a_t to generate $[T_t;\ S_t]$. Note that we denote motions in global coordinate system (*i.e.*, 3D space of long-term motion) of MultiAct to S_t, T_t for each timestep t. The motions canonicalized into the local coordinate system that is provided into MACVAE for each step are denoted to $\mathcal{S}_p, \mathcal{T}$ and $\mathcal{S}_c$. In each recurrent step, we give S_{t-1} and a_t of MultiAct into $\mathcal{S}_p$ and $\mathfrak{a}_c$ of MACVAE. In return, generated motion $[\hat{\mathcal{T}},\ \hat{\mathcal{S}_c}]$ of MACVAE is saved as $[T_t;\ S_t]$ in MultiAct. We show the 3-step pipeline of MultiAct below.

1. **Initialize.** We first generate the initial action motion S_1 with separately trained ACTOR (Petrovich, Black, and Varol 2021) model using the first action label a_1.
2. **Recurrent generation.** We have S_{t-1} from initialization or the previous recurrent step. First, we canon. S_{t-1} onto the local coordinate system and denote to S'_{t-1} as described in the MACVAE section. Second, canonicalized S'_{t-1} and a_t are passed into MACVAE, and generates $[T'_t;\ S'_t]$. Generated S'_t follows the action a_t, and T'_t connects between S'_{t-1} and S'_t. Finally, S'_t is passed to the next step as the previous uncanonicalized motion S_t. We repeat this recurrence for the given sequence of actions.
3. **Connect into long-term motion.** From the recurrent generation, we have a sequence of local motions $S_1, [T'_2;\ S'_2], ..., [T'_N;\ S'_N]$. We connect them inductively into a global coordinate, assuming that we have connected motions $[S_1;\ T_2;\ ...;\ S_t]$ up to time step t. We uncanon. $[T'_{t+1};\ S'_{t+1}]$ onto global coordinate (*i.e.*, connect to the last frame of S_t) then concatenate them. At last, we have connected long-term motion $[S_1;\ ...;\ T_N;\ S_N]$, which is controlled by action labels $a_1, ..., a_N$.

Experiments

Datasets

BABEL (Punnakkal et al. 2021) is the only dataset that consists of a long-term human motion with sequential action labels. The set of action labels in the BABEL contains *transition* as a sole action label, like other labels, such as walk and sit. Furthermore, the action label *transition* comes in between other action labels, forming an alternating sequence of action labels. This precisely fits our goal of modeling long-term motion with alternating action motions and transitions. We use training and validation split for training and testing of our model, respectively. Especially for the testing, we use the *unseen previous motion inputs from the test set*, which allows our experiment to demonstrate that MultiAct successfully generalizes to unseen motions. The detail of data sampling is illustrated in supplementary materials.

Method	FID$_{\text{train}}$ ↓	FID$_{\text{test}}$ ↓	Acc. top1 ↑	Acc. top5 ↑	Div. →	Multimod. →
Real-train	$0.019^{\pm 0.004}$	$0.90^{\pm 0.014}$	$0.94^{\pm 0.0014}$	$1.00^{\pm 0.001}$	$6.87^{\pm 0.124}$	$3.29^{\pm 0.058}$
Real-test	$0.90^{\pm 0.018}$	$0.095^{\pm 0.0034}$	$0.68^{\pm 0.0039}$	$0.89^{\pm 0.0035}$	$\underline{6.75^{\pm 0.045}}$	$\underline{3.73^{\pm 0.031}}$
MACVAE (Ours)	$0.74^{\pm 0.019}$	$0.97^{\pm 0.015}$	$\mathbf{0.64^{\pm 0.009}}$	$\mathbf{0.86^{\pm 0.003}}$	$\mathbf{6.74^{\pm 0.020}}$	$\mathbf{3.72^{\pm 0.019}}$
w/o separate latent	$1.33^{\pm 0.087}$	$1.43^{\pm 0.073}$	$0.51^{\pm 0.096}$	$0.72^{\pm 0.012}$	$6.55^{\pm 0.12}$	$4.45^{\pm 0.036}$
$\mathfrak{a}_c$ to PrevNet	$1.28^{\pm 0.035}$	$1.45^{\pm 0.092}$	$0.49^{\pm 0.018}$	$0.72^{\pm 0.011}$	$6.55^{\pm 0.061}$	$4.69^{\pm 0.056}$
whole $\mathcal{S}_p$ to PrevNet	$0.91^{\pm 0.022}$	$1.10^{\pm 0.092}$	$0.56^{\pm 0.004}$	$0.78^{\pm 0.003}$	$6.61^{\pm 0.043}$	$4.29^{\pm 0.038}$
w/o canon.	$1.21^{\pm 0.036}$	$1.12^{\pm 0.026}$	$0.52^{\pm 0.0035}$	$0.77^{\pm 0.002}$	$6.60^{\pm 0.046}$	$4.25^{\pm 0.037}$
with zero-canon.	$\mathbf{0.72^{\pm 0.025}}$	$\mathbf{0.89^{\pm 0.017}}$	$\mathbf{0.64^{\pm 0.018}}$	$0.85^{\pm 0.0064}$	$6.71^{\pm 0.076}$	$4.35^{\pm 0.13}$

Table 1: Ablation. We ablate the performance of single-step MACVAE to generate action motions against alternative designs. The second block ablates the high-level design idea. Effects of canon. methods are presented in the third block. Symbol → means closer to the real (underlined) is better.

Evaluation Metrics

We use frechet inception distance **(FID)**, action recognition accuracy **(Acc.)**, diversity **(Div.)**, and multimodality **(Multimod.)** as the measurement of the quality of the generated motions, following previous works (Petrovich, Black, and Varol 2021; Guo et al. 2020). **FID**$_{\text{train}}$ and **FID**$_{\text{test}}$ represents distribution divergence from generated samples to training and test set, respectively. **Acc.** measures how likely generated motions are classified to their action label by the pre-trained action recognition model. Lower **FID** and higher **Acc.** implies the better quality. Meanwhile, **Div.** and **Multimod.** show the variance of the generated motion across all actions and within each action, respectively. The value closer to the real data (underlined in Tab. 1) is better.

Ablation Study

For all ablation studies, we report the performance when the previous motions are from unseen test sets.

Separate latent embedding. *'w/o separate latent'* in Tab. 1 shows that our separate latent embedding of z_c and z_p is highly beneficial in every evaluation metric. Removal of separate latent embedding changes our model to unify the prior networks (*i.e.*, generate z from $\mathcal{S}_p$ and $\mathfrak{a}_c$), and decoder to use only z to reconstruct the output motion $[\hat{\mathcal{T}}, \hat{\mathcal{S}_c}]$.

Inputs of PrevNet. *'$\mathfrak{a}_c$ to PrevNet'* in Tab. 1 shows that the score drops when action label $\mathfrak{a}_c$ is additionally provided to PrevNet, while our original input is only the previous motion $\mathcal{S}_p$. The PrevNet is expected to deliver the context of the previous motion to the decoder so that the decoder can smoothly connect previous and generated motions. In this regard, the context delivered by the PrevNet should contain information about how the previous motion ends, not an action label of the previous motion, as multiple motions can correspond to the action label. As a result, providing action label $\mathfrak{a}_c$ to PrevNet leads to inferior results.

'whole $\mathcal{S}_p$ to PrevNet' in Tab. 1 shows that using whole previous motion $\mathcal{S}_p$ degrades the performance compared to ours (using last 4 frames). Passing too many frames to PrevNet can memorize unnecessary frames during the training stage. Memorization of such unnecessary frames of the training set results in overfitting to the training set. Our decision to use only the last four frames of the previous motion is to deliver minimal and only necessary information and to prevent the system from simply memorizing the training set.

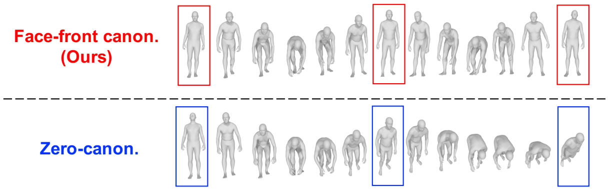


Figure 5: Qualitative comparison of the canonicalization. We show the long-term motion generated from *("stand", "bend", "stand", "bend", "stand")* with face-front (red) and zero-canonicalization (blue).

Canonicalization. The two settings in the last block of Tab. 1 and Fig. 5 demonstrate necessity of our face-front canon. We show that our face-front canon. plays an irreplaceable role in generating a long-term motion of multiple actions, which is more than an ad-hoc visualization method but a normalization process that has a decisive effect on both quantitative and qualitative results. The details of canon. methods are introduced in MACVAE section and Fig. 4a. Both zero- (Zhang, Black, and Tang 2021) and face-front canon. simplify the highly varying motion space, which leads to the performance gain.

We observed that zero-canon. (Zhang, Black, and Tang 2021) suffers from loss of the floor geometry during recurrent generation (Fig. 5) since it wipes out the global roll and pitch rotation which determines the ground. One example is the motion that ends in a bent-down position, as in Fig. 4b. The zero-canon. maps the motion "feet in the air" in the local coordinate, as illustrated in Fig. 4b. As a result, generated future motion places feet back on the ground in the local coordinate. However, in a global coordinate (*i.e.*, real-world), such motion is equivalent to leaning towards the ground. (Fig. 5)

As illustrated in the lower part of Fig. 5 and the previous paragraph, such a problem can not be handled by post-generation alignment during the visualization as the generated output is physically implausible. Our face-front canon. is the proper normalization method that leads to the physi-

Method	FID$_{\text{train}}$ ↓	FID$_{\text{test}}$ ↓	Acc. top1 ↑	Acc. top5 ↑	Div. →	Multimod. →
Real-train	$0.019^{\pm0.004}$	$0.90^{\pm0.014}$	$0.94^{\pm0.0014}$	$1.00^{\pm0.001}$	$6.87^{\pm0.124}$	$3.29^{\pm0.058}$
Real-test	$0.90^{\pm0.018}$	$0.095^{\pm0.0034}$	$0.68^{\pm0.0039}$	$0.89^{\pm0.0035}$	$6.75^{\pm0.045}$	$3.73^{\pm0.031}$
MACVAE (Ours)	$\mathbf{0.74^{\pm0.019}}$	$\mathbf{0.97^{\pm0.015}}$	$\mathbf{0.64^{\pm0.009}}$	$\mathbf{0.86^{\pm0.003}}$	$\mathbf{6.74^{\pm0.020}}$	$\mathbf{3.72^{\pm0.019}}$
ACTOR	$1.48^{\pm0.043}$	$1.53^{\pm0.032}$	$0.50^{\pm0.0078}$	$0.75^{\pm0.0041}$	$6.52^{\pm0.64}$	$4.42^{\pm0.043}$

Table 2: Comparison with SOTA: ACTOR. We present the performance of single-step MACVAE to generate action motions from unseen previous motion and an action label. Previous action-conditioned SOTA ACTOR (Petrovich, Black, and Varol 2021) is given as a baseline for comparison. Symbol → means closer to the real (underlined) is better.

Method	FID$_{\text{train}}$ ↓	FID$_{\text{test}}$ ↓
Ours (prev. motion from testset)	$\mathbf{0.87^{\pm0.052}}$	$\mathbf{0.67^{\pm0.027}}$
Ours (prev. motion from ACTOR)	$\mathbf{2.20^{\pm0.67}}$	$\mathbf{2.03^{\pm0.73}}$
ACTOR + SSMCT (w. align)	$6.34^{\pm0.10}$	$5.85^{\pm0.059}$
ACTOR + SSMCT (w. o. align)	$7.26^{\pm0.046}$	$7.20^{\pm0.068}$
ACTOR + Interpolation	$13.25^{\pm0.35}$	$13.36^{\pm0.29}$

Table 3: Comparison with SOTA: ACTOR+SSMCT. We report FID score of generated transition from Ours, compared to combination of SOTA methods (ACTOR + SSMCT).

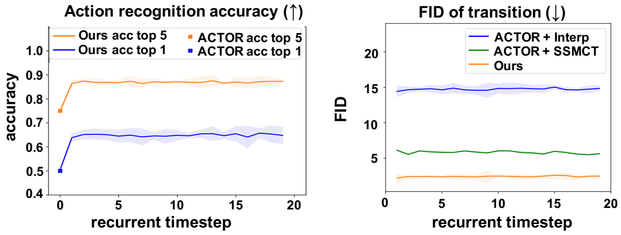


Figure 6: Long-term generation. We report the action recognition accuracy of generated long-term motion by MultiAct (left). FID$_{\text{test}}$ of transition (right) is also reported.

cally plausible generation result in each step (that shares the global ground geometry). The qualitative (Fig. 5) and quantitative result (last block of Tab. 1) shows that our face-front canon successfully overcomes such a problem.

Comparison with State-Of-The-Art Methods

Since our MultiAct is the earliest attempt to generate long-term motion from multiple actions, we do not have directly comparable prior works. As an alternative, we combine two SOTA models, action-conditioned ACTOR (Petrovich, Black, and Varol 2021) and in-betweening SSMCT (Duan et al. 2021) trained on BABEL into a unified pipeline: SSMCT connects the individually generated action motion from ACTOR. Such SOTA-combined method is the best-known method before our work to generate multiple-action motions. We evaluate the quality of transition and action motion in both single-step and long-term generation.

Action motion. In the SOTA-combined pipeline, the generation of action motion is solely dependent on the ACTOR; thus, we compare ours to ACTOR. In Tab. 2, we have observed that single-action motions from MACVAE using the test set, *which consists of unseen previous motions*, outperform the SOTA method ACTOR in all evaluation metrics.

The left plot in Fig. 6 shows that our generated long-term motion shows much higher recognition accuracy than ACTOR and maintains the quality after twenty steps of repetitive recurrence. Note that we have sampled the input action sequence from the test set so that such a result shows that MultiAct generalizes well to the unseen permutation of action labels that are not provided during the training. Note that MultiAct uses ACTOR to initialize S_1; thus, the metric score of the first step is identical to ACTOR.

Transition. Tab. 3 compares the quality of our single-step transition against SOTA-combined pipeline: SSMCT-generated transition in between two ACTOR-generated action motions. The quality of our transition in both previous motion conditions largely outperforms both the SOTA-combined method and linear interpolation baseline. As ACTOR generates motions independent of previous motions, the initial status of generated motions can be very different from the last one of previous motions (*e.g.*, seeing the opposite direction), which can result in unnatural in-betweening of SSMCT. The table shows that although we fix this issue by manually aligning ACTOR's action motions before performing in-betweening with SSMCT, ours still largely outperforms it. We also report the long-term transition quality of MultiAct on the right of Fig. 6. Our method outperforms the other baselines throughout twenty steps of recurrence.

The performance of the SOTA-combined pipeline heavily depends on the first step, ACTOR, since the second step, SSMCT, takes the ACTOR-generated motions as an input. The lower quality of ACTOR-generated motions (as shown in Tab. 2) leads to the inferior transition quality in the second step. The result supports our idea of simultaneously generating transition and action motion from the joint condition of motion and action. To summarize, our MultiAct outperforms the previous SOTA on the proposed task: to generate long-term motion from multiple actions.

Conclusion

We present MultiAct, the first framework to generate long-term 3D human motion of multiple actions recurrently. For the recurrent generation of long-term motion composed of transitions and actions, our model concurrently generates transition and action motion from the joint condition of action and motion. As a result, our MultiAct has outperformed the previous SOTA-combined method in generating long-term motion of multiple actions.

Acknowledgements

This work was supported in part by the IITP grants [No.2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University), No.2022-0-00156, No. 2021-0-02068, and No.2022-0-00156], and the NRF grant [No. 2021M3A9E4080782] funded by the Korea government (MSIT), and AIRS Company in Hyundai Motor Company & Kia Corporation through HMC/KIA-SNU AI Consortium Fund.

References

Ahn, H.; Ha, T.; Choi, Y.; Yoo, H.; and Oh, S. 2018. Text2action: Generative adversarial synthesis from language to action. In *ICRA*.

Ahuja, K.; Ofek, E.; Gonzalez-Franco, M.; Holz, C.; and Wilson, A. D. 2021. CoolMoves: User Motion Accentuation in Virtual Reality. *ACM IMWUT*, 5(2): 1–23.

Aksan, E.; Kaufmann, M.; and Hilliges, O. 2019. Structured prediction helps 3D human motion modelling. In *ICCV*.

Barsoum, E.; Kender, J.; and Liu, Z. 2018. Hp-gan: Probabilistic 3D human motion prediction via gan. In *CVPRW*.

Cai, H.; Bai, C.; Tai, Y.-W.; and Tang, C.-K. 2018. Deep Video Generation, Prediction and Completion of Human Action Sequences. In *ECCV*.

Cao, Z.; Gao, H.; Mangalam, K.; Cai, Q.-Z.; Vo, M.; and Malik, J. 2020. Long-Term Human Motion Prediction with Scene Context. In *ECCV*.

Chan, W. P.; Tran, T.; Sheikholeslami, S.; and Croft, E. 2021. An Experimental Validation and Comparison of Reaching Motion Models for Unconstrained Handovers: Towards Generating Humanlike Motions for Human-Robot Handovers. *arXiv preprint arXiv:2108.12780*.

Chen, W.; Wang, H.; Yuan, Y.; Shao, T.; and Zhou, K. 2020. Dynamic Future Net: Diversified Human Motion Generation. *arXiv preprint arXiv:2009.05109*.

Duan, Y.; Shi, T.; Zou, Z.; Lin, Y.; Qian, Z.; Zhang, B.; and Yuan, Y. 2021. Single-Shot Motion Completion with Transformer. *arXiv preprint arXiv:2103.00776*.

Fragkiadaki, K.; Levine, S.; Felsen, P.; and Malik, J. 2015. Recurrent network models for human dynamics. In *ICCV*.

Guo, C.; Zuo, X.; Wang, S.; and Cheng, L. 2022. TM2T: Stochastic and Tokenized Modeling for the Reciprocal Generation of 3D Human Motions and Texts. In *ECCV*.

Guo, C.; Zuo, X.; Wang, S.; Zou, S.; Sun, Q.; Deng, A.; Gong, M.; and Cheng, L. 2020. Action2Motion: Conditioned Generation of 3D Human Motions. In *ACM MM*.

Habibie, I.; Holden, D.; Schwarz, J.; Yearsley, J.; and Komura, T. 2017. A recurrent variational autoencoder for human motion synthesis. In *BMVC*.

Harvey, F. G.; Yurick, M.; Nowrouzezahrai, D.; and Pal, C. 2020. Robust Motion In-Betweening. In *SIGGRAPH*.

Kingma, D. P.; and Welling, M. 2014. Auto-Encoding Variational Bayes. In *ICLR*.

Li, J.; Villegas, R.; Ceylan, D.; Yang, J.; Kuang, Z.; Li, H.; and Zhao, Y. 2021. Task-Generic Hierarchical Human Motion Prior using VAEs.

Lin, A. S.; Wu, L.; Corona, R.; Tai, K.; Huang, Q.; and Mooney, R. J. 2018. Generating animated videos of human activities from natural language descriptions. *Learning*, 2018: 1.

Maheshwari, S.; Gupta, D.; and Sarvadevabhatla, R. K. 2021. MUGL: Large Scale Multi Person Conditional Action Generation with Locomotion.

Mao, W.; Liu, M.; and Salzmann, M. 2021. Generating Smooth Pose Sequences for Diverse Human Motion Prediction. In *ICCV*.

Neuhaus, R.; Laschke, M.; Theofanou-Fülbier, D.; Hassenzahl, M.; and Sadeghian, S. 2019. Exploring the impact of transparency on the interaction with an in-car digital AI assistant. In *ACM AutomotiveUI*.

Petrovich, M.; Black, M. J.; and Varol, G. 2021. Action-Conditioned 3D Human Motion Synthesis with Transformer VAE. In *ICCV*.

Petrovich, M.; Black, M. J.; and Varol, G. 2022. TEMOS: Generating diverse human motions from textual descriptions. In *ECCV*.

Punnakkal, A. R.; Chandrasekaran, A.; Athanasiou, N.; Quiros-Ramirez, A.; and Black, M. J. 2021. BABEL: Bodies, Action and Behavior with English Labels. In *CVPR*.

Rempe, D.; Birdal, T.; Hertzmann, A.; Yang, J.; Sridhar, S.; and Guibas, L. J. 2021. HuMoR: 3D Human Motion Model for Robust Pose Estimation. In *ICCV*.

Romero, J.; Tzionas, D.; and Black, M. J. 2017. Embodied Hands: Modeling and Capturing Hands and Bodies Together. *ACM TOG*, 36(6).

Sohn, K.; Lee, H.; and Yan, X. 2015. Learning Structured Output Representation using Deep Conditional Generative Models. In *NeurIPS*.

Stoll, S.; Camgoz, N. C.; Hadfield, S.; and Bowden, R. 2020. Text2Sign: Towards sign language production using neural machine translation and generative adversarial networks. *IJCV*, 128(4): 891–908.

Wang, J.; Xu, H.; Xu, J.; Liu, S.; and Wang, X. 2021a. Synthesizing Long-Term 3D Human Motion and Interaction in 3D Scenes. In *CVPR*.

Wang, J.; Yan, S.; Dai, B.; and Lin, D. 2021b. Scene-aware Generative Network for Human Motion Synthesis. In *CVPR*.

Wang, T.; and Wan, X. 2019. T-CVAE: Transformer-Based Conditioned Variational Autoencoder for Story Completion. In *IJCAI*.

Yang, C.; Wang, Z.; Zhu, X.; Huang, C.; Shi, J.; and Lin, D. 2018. Pose Guided Human Video Generation. In *ECCV*.

Yuan, Y.; and Kitani, K. 2020. Dlow: Diversifying latent flows for diverse human motion prediction. In *ECCV*.

Zhang, Y.; Black, M. J.; and Tang, S. 2020. Perpetual Motion: Generating Unbounded Human Motion. *arXiv preprint arXiv:2007.13886*.

Zhang, Y.; Black, M. J.; and Tang, S. 2021. We are More than Our Joints: Predicting how 3D Bodies Move. In *CVPR*.

Zhou, Y.; Barnes, C.; Jingwan, L.; Jimei, Y.; and Hao, L. 2019. On the Continuity of Rotation Representations in Neural Networks. In *CVPR*.

Zhou, Y.; Lu, J.; Barnes, C.; Yang, J.; Xiang, S.; and li, H. 2020. Generative Tweening: Long-term Inbetweening of 3D Human Motions. *arXiv preprint arXiv:2005.08891*.