We are More than Our Joints: Predicting how 3D Bodies Move **Appendix**

MOJO is implemented using SMPL-X but any other parametric 3D body model could be used, e.g. [39, 54]. To do so, one only needs to implement the recursive fitting of 3D pose to observed markers. This is a straightforward optimization problem. In this paper, we use SMPL-X to demonstrate our MOJO idea, because we can exploit the large-scale AMASS dataset to train/test our networks. Moreover, SMPL-X is rigged with a skeleton like other body models in computer graphics, so it is completely compatible with standard skeletal techniques.

A. More Method Details

Marker placements in our work. In our experiments, we use two kinds of marker placements. The first (default) one is the CMU [1] setting with 41 markers. The second one is the SSM2 [34] setting with 67 markers. These marker settings are illustrated in Fig. S1.

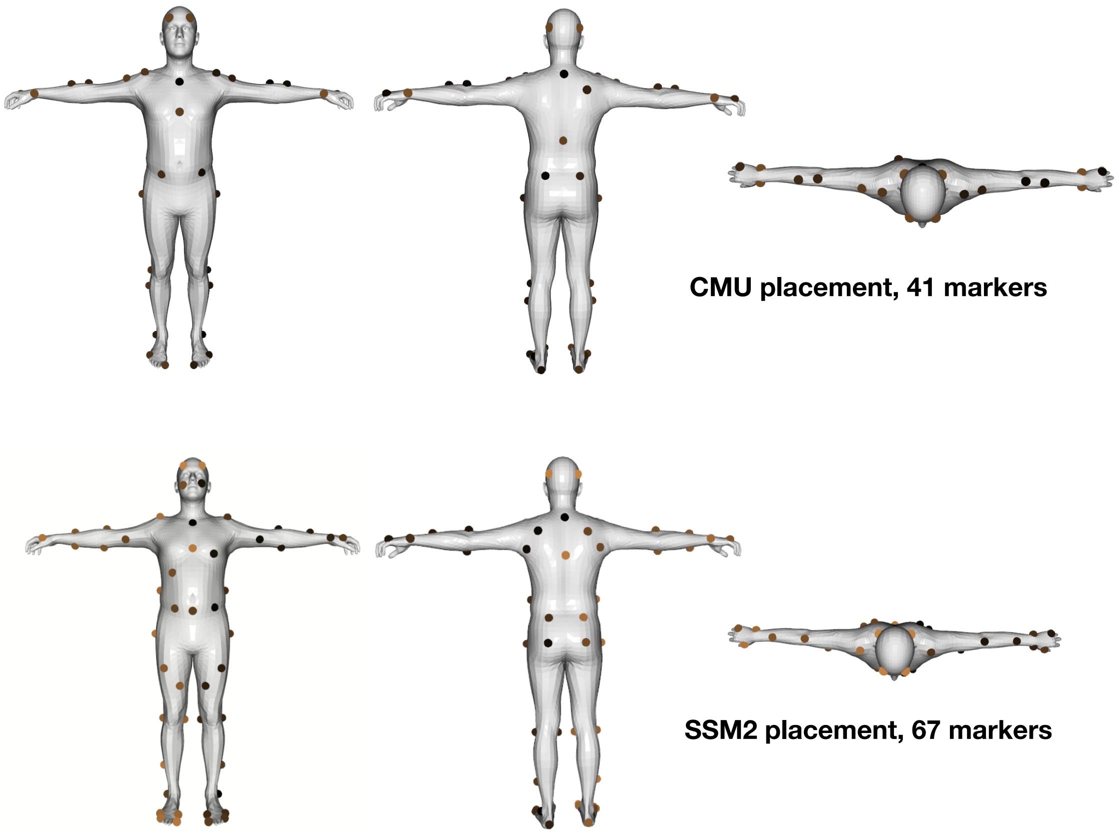


Figure S1: Illustration of our marker settings. The markers are denoted by 3D spheres attached to the SMPL-X body surface. From left to right: the front view, the back view and the top view.

Network architectures. We have demonstrated the CVAE architecture of MOJO in Sec. 3, and compare it with several baselines and variants in Sec. 4. The architectures of the used motion generators are illustrated in Fig. S2. Compared to the CVAE of MOJO, VAE+DCT has no residual connections at the output, and the velocity reconstruction loss is replaced by a loss to minimize $|\boldsymbol{x}_M - \boldsymbol{y}_0|^2$ [56]. MOJO-DCT-proj encodes the motion $\boldsymbol{Y}$ into a single feature vector, rather than a set of frequency components.

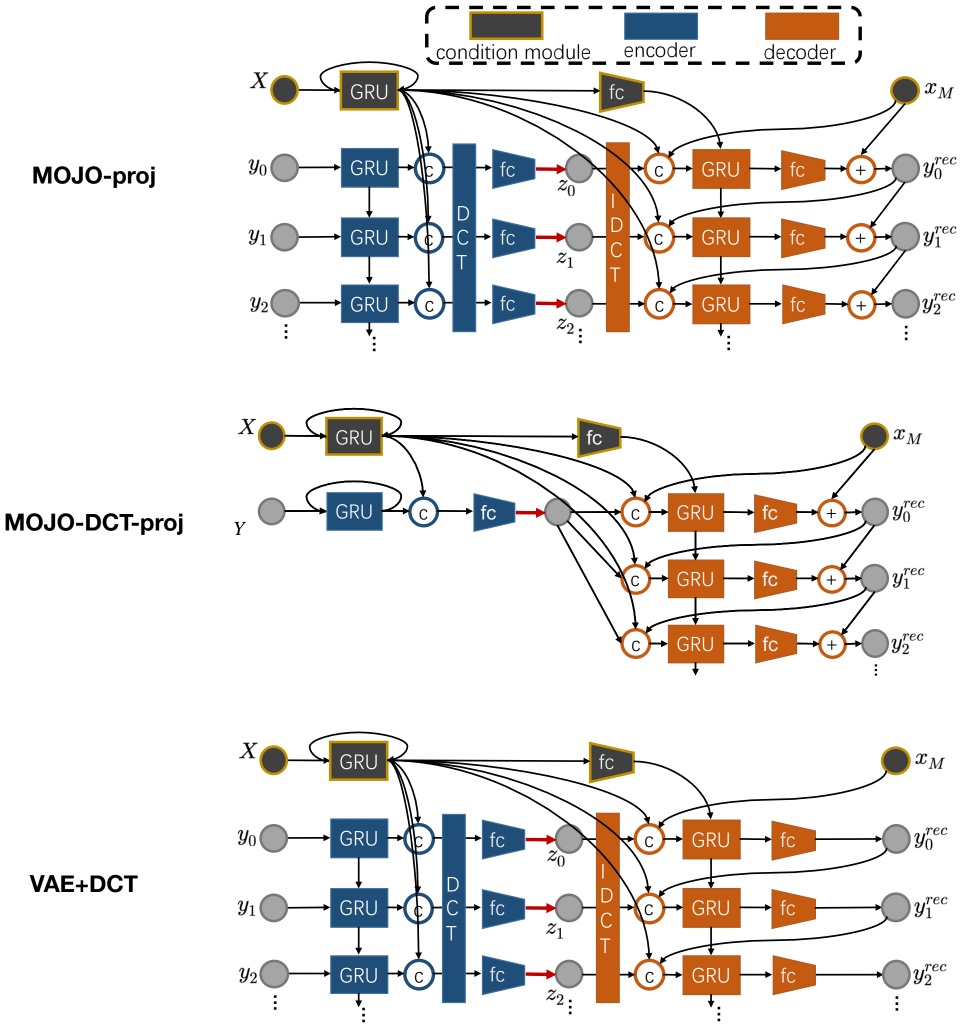


Figure S2: From top to bottom: (1) The CVAE architecture of MOJO. (2) The CVAE architecture of MOJO-DCT, which is used in Tables 1 3 4. (3) The architecture of VAE+DCT, which is evaluated in Tab. 2. Note that we only illustrate the motion generators here. The recursive projection scheme can be added during the testing time.

B. More Experimental Details

Implementation. We use PyTorch v1.6.0[Paszke et al. 2019] in our experiments. In the training loss Eq. (1), we empirically set $\alpha = 3$ in all cases. We find a larger value causes over-smooth and loses motion details, whereas a smaller value can cause jitters. For the models with the latent DCT space, the robust KLD term with the loss weight 1 is employed. For models without the latent DCT space, we use a standard KLD term, and set its weight to 0.1. The weights of the KLD terms are not annealed during training. In the fitting loss Eq. (2), we empirically set $\{\lambda_1, \lambda_2\} = \{0.0005, 0.01\}$. Smaller values can degrade the pose realism with e.g. twisted torsos, and larger values reduce the pose variations in motion. Our code is publicly released, which delivers more details.

The model 'MOJO-DCT-proj' has a latent dimension of 128. However, 'MOJO-proj' suffers from overfitting with the same latent dimension, and hence we set its latent dimension to 16. We use these dimensions in all experiments.

AMASS sequence canonicalization. To unify the sequence length and world coordinates, we canonicalize **AMASS** as follows: *First*, we trim the original sequences into 480-frame (4-second) subsequences, and downsample them from 120fps to 15fps. The condition sequence $\boldsymbol{X}$ contains 15 frames (1s) and the future sequence $\boldsymbol{Y}$ contains 45 frames (3s). *Second*, we unify the world coordinates as in [60]. For each subsequence, we reset the world coordinate to the SMPL-X [40] body mesh in the first frame: The horizontal X-axis points in the direction from the left hip to the right hip, the Z-axis is the negative direction of gravity, and the Y-axis is pointing forward. The origin is set to to the body's global translation.

More discussions on *MMADE* and *MMFDE*. As in [56], we use *MMADE* and *MMFED* to evaluate prediction accuracy when the input sequence slightly changes. They are regarded as multi-modal version of *ADE* and *FDE*, respectively. Let's only demonstrate *MMADE* with more details here, since the same principle applies to *MMFDE*.

The *ADE* can be calculated by

$$e_{ADE}(\mathcal{Y}) = \frac{1}{T} \min_{\boldsymbol{Y} \in \mathcal{Y}} |\boldsymbol{Y} - \boldsymbol{Y}_{gt}|^2, \tag{3}$$

in which $\boldsymbol{Y}$ is a predicted motion, $\mathcal{Y}$ is the set of all predicted motions, and $\boldsymbol{Y}_{gt}$ is the ground truth future motion. In this case, the *MMADE* can be calculate as

$$e_{MMADE}(\mathcal{Y}) = \mathbb{E}_{\boldsymbol{Y}^* \in \mathcal{Y}_S} \left[\frac{1}{T} \min_{\boldsymbol{Y} \in \mathcal{Y}} |\boldsymbol{Y} - \boldsymbol{Y}^*|^2 \right] \tag{4}$$

with

$$\mathcal{Y}_s = \{\boldsymbol{Y}^* \in \mathcal{Y}_{gt} \,|\, d(\boldsymbol{X}^*, \boldsymbol{X}_{gt}) < \eta\}, \tag{5}$$

with $\mathcal{Y}_{gt}$ is the set of all ground truth future motion, $\boldsymbol{X}$ denotes the corresponding motion in the past, $d(\cdot)$ is a difference measure, and η is a pre-defined threshold. In the work of DLow [56], the difference measure $d(\cdot)$ is based on the L2 difference between the *last frames* of the two motion sequences from the past.

Body deformation metric. We measure the markers on the head, the upper torso and the lower torso, respectively. Specifically, according to **CMU** [1], we measure ('LFHD', 'RFHD', 'RBHD', 'LBHD') for the head, ('RSHO', 'LSHO', 'CLAV', 'C7') for the upper torso, and ('RFWT', 'LFWT', 'LBWT', 'RBWT') for the lower torso. For each rigid body part P, the deformation score is the variations of marker pair-wise distances, and is calculated by

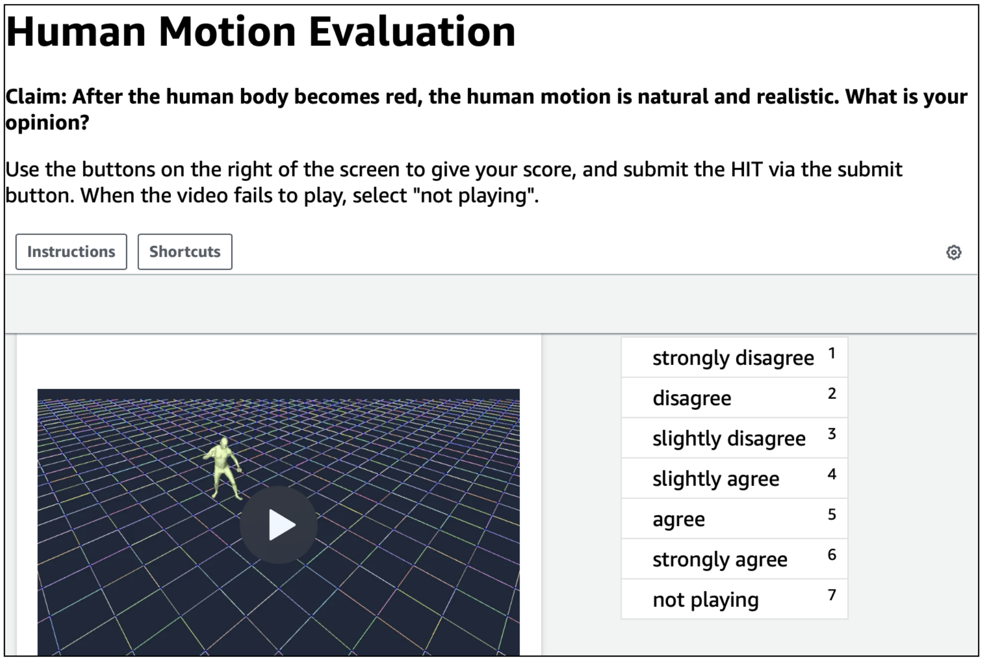
$$s_d(P) = \mathbb{E}_{\boldsymbol{Y}} \left[\sum_{(i,j) \in P} \sigma_t(|\boldsymbol{v}_i^t - \boldsymbol{v}_j^t|^2) \right], \tag{6}$$

in which $\boldsymbol{v}_i^t$ denotes the location of the marker i at time t, σ_t denotes the standard deviation along the time dimension, and $\mathbb{E}_{\boldsymbol{Y}}[\cdot]$ denotes averaging the scores of different predicted sequences.

User study interface. Our user study is performed via AMT. The interface is illustrated in Fig. S3. We set a six-point Likert scale for evaluation. Each video is evaluated by three subjects.

Performance of the original VAE setting in DLow. Noticeably, the DLow CVAE with the original setting cannot directly work on body markers, although it works well with joint locations. Following the evaluation in Tab. 1, the original VAE setting in DLow gives (diversity=81.10, ADE=2.79, FDE=4.71, MMADE=2.81, MMFDE=4.71, FSE=0.0031). The diversity is much higher, but the accuracy is considerably worse. Fig. S4 shows that its predicted markers are not valid.

Influence of the given frames. Based on the CMU markers, we train another two MOJO versions with different input/output sequence lengths. The results are in Tab. S1. As the predicted sequence becomes shorter and the input sequence becomes longer, we can see that the accuracy increases but the diversity decreases consistently, indicating that MOJO becomes more confident and deterministic. Such behavior is similar to other state-of-the-art methods like DLow. Furthermore, to predict even longer motion sequences, one could use a sliding window, and recursively input the predicted sequence to the pre-trained MOJO model to generate new sequences.



Human Motion Evaluation

Claim: After the human body becomes red, the human motion is natural and realistic. What is your opinion?

Use the buttons on the right of the screen to give your score, and submit the HIT via the submit button. When the video fails to play, select "not playing".

Figure S3: The user interface of our perceptual study on AMT.

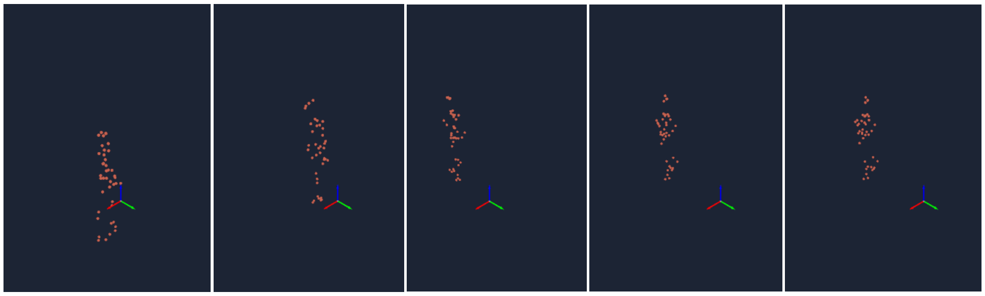


Figure S4: Illustrations of the invalid body markers predicted by the original DLow VAE setting. From left to right are five predicted frames over time.

MOJO frame setting	*Div.*↑	*ADE*↓	*FDE*↓	*MMADE*↓	*MMFDE*↓	*BDF*↓
			ACCAD			
pred. 55 with 5	**32.666**	1.303	1.793	1.409	2.210	0
pred. 45 with 15	20.676	1.214	1.919	1.306	2.080	0
pred. 15 with 45	3.728	**0.642**	**1.048**	**0.701**	**1.093**	0
			BMLhandball			
pred. 55 with 5	**27.045**	1.032	1.170	1.054	1.192	0
pred. 45 with 15	16.806	0.949	1.139	1.001	1.172	0
pred. 15 with 45	3.574	**0.666**	**0.993**	**0.759**	**1.051**	0

Table S1: Varying the length of the input and output sequence, denoted as 'predicting Y frames with X frames'. The setting '*pred. 45 with 15*' (predicting 3 sec with 1 sec) was reported in Tab. 6. Best results are in boldface.

C. More Analysis on the Latent DCT Space

Analysis on the latent space dimension. First, we use **Human3.6M** to analyze the latent DCT space, with joint locations to represent the human body in motion. Tab. S2 shows the performance under various settings. Similar to Tab. 2, as

Method	*Diversity*	*ADE*	*FDE*	*MMADE*	*MMFDE*
(32d)VAE+DCT	3.405	0.432	0.544	0.533	0.589
(32d)VAE+DCT+1	7.085	0.419	0.514	0.515	0.547
(32d)VAE+DCT+5	12.007	0.415	**0.510**	0.505	0.542
(32d)VAE+DCT+10	13.103	0.417	0.513	0.507	0.544
(32d)VAE+DCT+20	14.642	0.418	0.516	0.510	0.548
(64d)VAE+DCT	3.463	0.429	0.544	0.532	0.587
(64d)VAE+DCT+1	7.254	0.417	0.514	0.513	0.547
(64d)VAE+DCT+5	12.554	**0.413**	**0.510**	**0.504**	**0.540**
(64d)VAE+DCT+10	14.233	0.414	0.514	0.506	0.546
(64d)VAE+DCT+20	**15.462**	0.416	0.517	0.508	0.548

Table S2: Model performances with different latent dimensions and number of frequency bands with DLow on the **Human3.6M** dataset. Best results are in boldface. This table is directly comparable with Tab. 2, which shows the results with the 128d latent space (same with [56]).

Method	*Diversity*	*ADE*	*FDE*	*MMADE*	*MMFDE*	*FSE*(10^{-3})
(8d)MOJO-DCT-proj	0.027	2.119	3.145	2.143	3.153	-2.6
(16d)MOJO-DCT-proj	0.060	2.105	3.134	2.125	3.133	-3.5
(32d)MOJO-DCT-proj	0.152	2.045	3.071	2.065	3.068	-3.9
(64d)MOJO-DCT-proj	17.405	1.767	2.213	1.790	2.219	0.2
(128d)MOJO-DCT-proj	**21.504**	**1.608**	**1.914**	**1.628**	**1.919**	**0.0**
(8d)MOJO-proj	20.236	**1.525**	1.893	1.552	1.893	-0.8
(16d)MOJO-proj	23.660	1.528	1.848	**1.550**	1.847	0.4
(32d)MOJO-proj	24.448	1.554	1.850	1.573	1.846	1.0
(64d)MOJO-proj	24.129	1.557	**1.820**	1.576	**1.819**	1.0
(128d)MOJO-proj	**25.265**	1.620	1.852	1.636	1.851	**2.5**

Table S3: Performances with various latent space dimensions on **BMLhandball**. Best results of each model are in boldface.

DLow is applied on more frequency bands, the diversity consistently grows, and the motion prediction accuracies are stable. Noticeably, VAE+DCT with a 32d latent space outperforms the baseline [56] (see Tab. 2), indicating that our latent DCT space has better representation power.

Additionally, for the marker-based representation, we evaluate the influence of the latent feature dimension using the **BMLhandball** dataset. The results are presented in Tab. S3, in which DLow is applied for MOJO-DCT-proj. According to the investigations on the **Human3.6M** dataset, we apply DLow on the lowest 20% (the lowest 9) frequency bands in MOJO-proj, corresponding to VAE+DCT+20 in Tab. S2. We can see that the MOJO-DCT-proj performs best with a 128d latent space, yet is worse than most cases of MOJO-proj, which indicates the representation power of the latent DCT space. In the meanwhile, different versions of MOJO-proj perform comparably with different latent dimensions. As the feature dimension increases, the diversity consistently increases, whereas the prediction accuracies decrease in most cases. Therefore, in our experiments, we set the latent dimensions of MOJO-DCT-proj and MOJO-proj to 128 and 16, respectively.

Visualization of the latent DCT space. Since the latent space is in the frequency domain, we visualize the average frequency spectra of the inference posterior in Fig. S5, based on the VAE+DCT(128d) model and the **Human3.6M** dataset. We find that the bias of fc layers between the DCT and the inverse DCT can lead to stair-like structures. For both cases with and without the fc layer bias, we can observe that most information is concentrated at low-frequency bands. This fact can explain the performance saturation when employing DLow on more frequency bands, and also fits the energy compaction property of DCT. Moreover, we show the performance without the fc bias in Tab. S4. Compared to the results in Tab. 2, we find that the influence of these bias values is trivial. Therefore, in our experiments we preserve the bias values in these fc layers trainable.

Method	*Diversity*	*ADE*	*FDE*	*MMADE*	*MMFDE*
VAE+DCT-fcbias	3.442	0.431	0.547	0.525	0.584
VAE+DCT+1-fcbias	7.072	0.417	0.514	0.506	0.541
VAE+DCT+5-fcbias	13.051	**0.413**	**0.512**	**0.498**	**0.537**
VAE+DCT+10-fcbias	14.723	0.415	0.515	0.500	0.540
VAE+DCT+20-fcbias	**16.008**	0.415	0.517	0.501	0.542

Table S4: The model performances with zero values of the fc layer bias. '-fcbias' denotes no bias. The best results are in boldface, and can be directly compared with the results in Tab. 2.

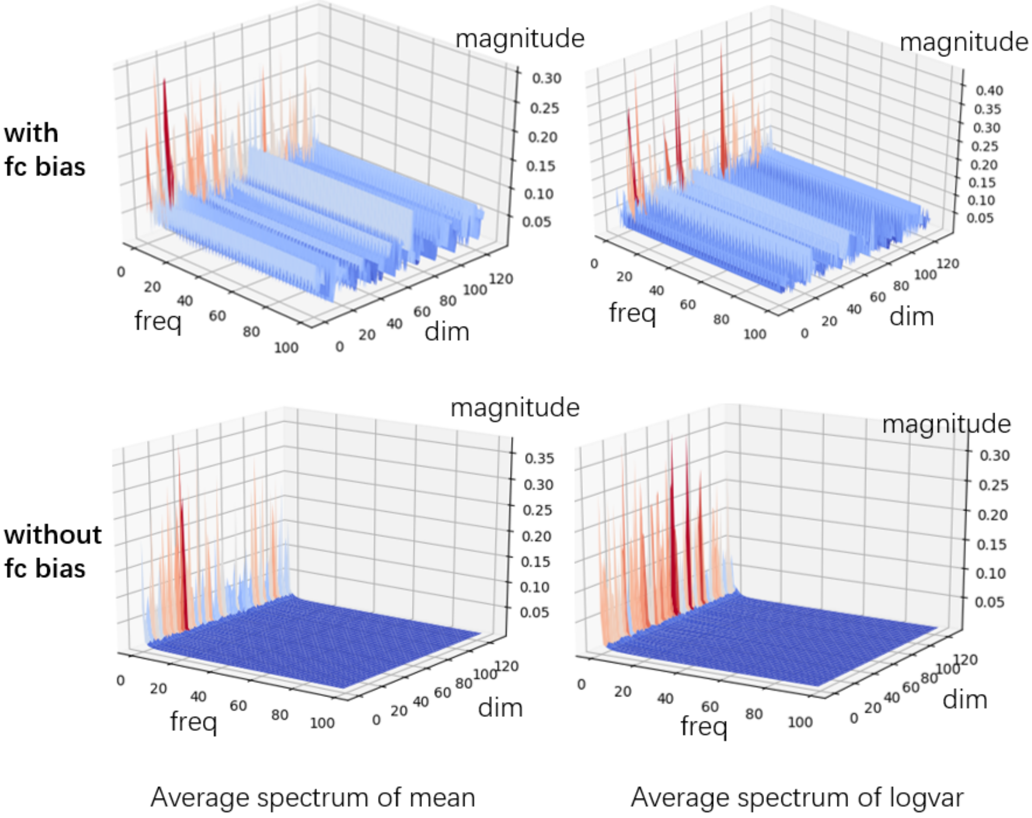


Figure S5: With the VAE+DCT(128d) model, we randomly select 5000 samples from the **Human3.6M** training set, and obtain their mean and the logarithmic variance values from the VAE encoder. To show the frequency spectra, we average all absolute values of mean or logarithmic variance. Note that when both mean and logarithmic variance are zero, the posterior is equal to the standard normal distribution, which only produces white noise without information.