The Wanderings of Odysseus in 3D Scenes
Appendix

A. Additional Experiments

A.1. Analysis on GAMMA

Marker predictor analysis. The marker predictor performances are tested on motion primitives from **ACCAD** and **HumanEva**. Following the metrics in [76,80], the evaluation is w.r.t. prediction diversity and accuracy. In this case, the diversity is the higher the better, and other metrics measuring the prediction errors are the lower the better. The results are shown in Tab. S1. The 1-frame-based predictors have much larger diversity, but lower accuracies than the 2-frame-based predictors, because generating motion from a static pose is more uncertain. Moreover, the rollout training can slightly improve diversity. Note that our work focuses on motion generation, and hence the accuracy is less important.

	ACCAD [4], *20 sub., 6,298 MPs*					**HumanEva** [62], *3 sub., 2021 MPs*				
Method	*Diversity*↑	*ADE*↓	*FDE*↓	*MMADE*↓	*MMFDE*↓	*Diversity*↑	*ADE*↓	*FDE*↓	*MMADE*↓	*MMFDE*↓
predictor-1f	1.64	0.42	0.69	0.57	0.81	2.02	0.30	0.48	0.42	0.55
predictor-2f	0.50	0.19	0.34	0.44	0.60	0.53	0.16	0.28	0.39	0.48
predictor-ro-1f	1.72	0.44	0.72	0.58	0.83	2.08	0.33	0.51	0.44	0.57
predictor-ro-2f	0.88	0.21	0.36	0.44	0.59	0.95	0.18	0.31	0.37	0.46

Table S1. Analysis on marker predictors. '-xf' denotes the motion seed has x frames. '-ro' denotes the model is fine-tuned with rollout. 'sub' and 'MPs' denote number of subjects and motion primitives, respectively.

Fig. S1 illustrates the difference between the one frame-based and the two frame-based models, respectively. Provided a single frame, the generated motion primitive is more diverse than the generated motion based on two frames. Because of the motion dynamics and the body inertia, the motion becomes more deterministic. We also tried predicting 7 frames based on 3 frames, but did not observe obvious differences with the 2-frame-based generation.

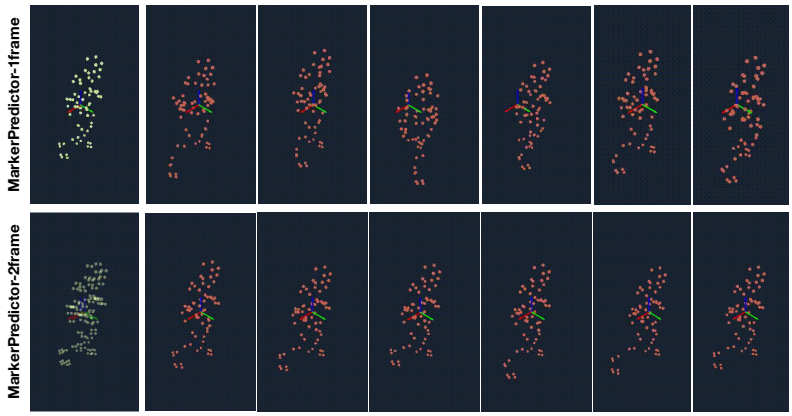

Figure S1. Visualization of generated motion primitives. In each row, the motion seed (green) is on the left, and the last frames of six generated primitives (red) are on the right.

Body regressor analysis. The performance is tested on motion primitives from **HumanEva**, **SSM**, **ACCAD** and **KIT**. We calculate the averaged marker distance (AMD) and the averaged mesh vertices distance (AVD) to the ground truth motion primitive for accuracy evaluation. The results are shown in Tab. S2. One can see that our proposed body regressors are accurate, and achieve comparable performance with Mosh and Mosh++, e.g. [44, Sec. 4.4].

A.2. Influence of Bodies on Motion Generation

First, we fix the initial pose and the motion target, and interpolate the first two components from -6 to 6 in the shape space to obtain 10 bodies. We find all bodies can reach the target, with the minimal/maximal steps being 12/53. The first row of

	male		female	
	AMD(mm)	AVD(mm)	AMD(mm)	AVD(mm)
HumanEva *3 sub., 2021 MPs*	3.26	4.80	3.52	5.52
SSM *3 sub., 285 MPs*	-	-	6.23	8.24
ACCAD *20 sub., 6,298 MPs*	7.99	9.61	11.39	12.04
KIT *55 sub., 130,331 MPs*	3.53	4.88	3.95	5.64

Table S2. Evaluation of body regressors w.r.t. reconstruction errors.

Fig. S2 shows 4 corner cases. These similar actions indicate the limited influence of the body shape. Second, we fix the body shape and randomly sample 10 poses from VPoser [52]. We find 3 bodies cannot reach the target within 60 steps, indicating the initial pose can largely influence motion generation. If the initial pose is not about walking, the body first adjusts itself to a 'ready-to-walk' pose (via stepping back, jumping, etc.), and then walks out. The second row of Fig. S2 shows 3 examples.

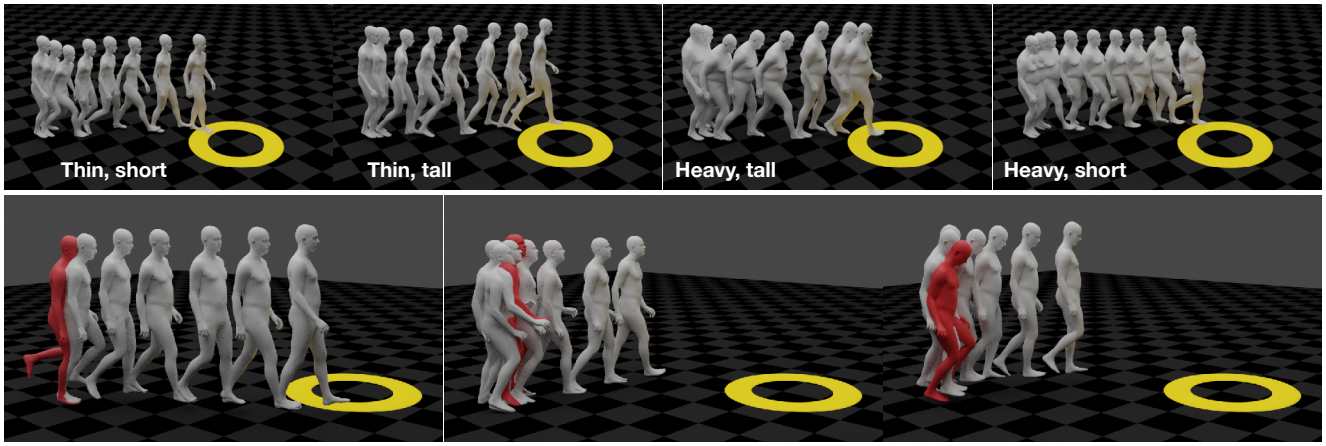

Figure S2. Influence of the body shape and the initial body pose on motion generation. See videos at https://yz-cnsdqz.github.io/eigenmotion/GAMMA/.

A.3. Comparison with MOJO [80]

Both our method and MOJO use the marker-based body representation and RNN-based neural networks for motion modelling. In contrast to long-term motion synthesis, MOJO [80] is proposed for stochastic motion prediction and uses a different setting than our method GAMMA. Specifically, MOJO is exploited to predict 3-second motions with a 1-second motion seed. Therefore, we only perform comparison w.r.t. the motion realism, e.g. [80, Table 4].

Here, we randomly select 900 static bodies from **ACCAD**, and generate 20 primitives ($\sim$ 5 sec) for each body. Using the same foot skating metric with MOJO, we obtain 0.064, which is significantly better than MOJO (0.341 and 0.278) and is comparable to the ground truth (0.067). In addition, Fig. 3 in our paper shows that our results have similar perceptual scores with the ground truth, whereas [80, Table 4] shows a much larger gap between the generated motion and the ground truth. These results can indicate the advantage of our motion primitive representation.

A.4. Comparison with Other Motion Representations

Our proposed method GAMMA comprises a marker predictor to generate future markers and a body regressor to recover 3D bodies from predicted markers. Therefore, the motion generation is performed in an 'indirect' manner. In contrast, a direct approach is to generate SMPL-X body parameters without any intermediate body representations. In this case, we can only use a single neural network to model their temporal relations.

To investigate their performance, here we use two body representations in the motion primitive setting, i.e. 1) the SMPL parameters $\boldsymbol{x} = (r, \Phi, \theta)$, and 2) the HuMoR kinematic feature [59] $\boldsymbol{x} = (r, \dot{r}, \Phi, \dot{\Phi}, \theta, J, \dot{J})$, in which r, Φ, θ, J denotes the root translation, orientation, joint rotations, and joint locations, respectively. All rotations are in axis-angle. We denote them as 'SMPL params' and 'HuMoR primitive', respectively.

Since we can directly obtain 3D bodies by inputting these features into SMPL-X [52], we only use a modified version of the marker predictor (see Fig. 2) for motion modeling. Except for the input and output dimensions, other network hyper-parameters are the same for a fair comparison. The training procedures follow Sec. 3.2.2, i.e., first training with individual motion primitives and then training with the rollout. Moreover, we replace the feature reconstruction loss with a forward kinematic loss as in [59], since directly reconstructing these features can hardly produce valid results.

We measure contact scores as in Fig. 3, and the body deformation w.r.t. the temporal variation (i.e. std) of piece-wise distances of rigid body markers (see [Tab.3, MOJO]). In this case, the contact score is the higher the better, and the body deformation is the lower the better. The results are in Fig. S3. Compared to 'HuMoR', other models on motion primitives have better contact scores. In particular, 'HuMoR primitive' obviously outperforms 'HuMoR'. This can indicate the advantage of the motion primitive setting for improving motion realism. In addition, compared to 'HuMoR primitive' and 'SMPL params', our methods perform better w.r.t. the body deformation. Invalid results, e.g., twisted bodies, are observed in their visualizations, but not observed in our results. This can indicate the advantage of the marker-based representation for constraining body DoFs.

In the literature, a number of various motion representations have been proposed. A thorough comparison between motion representations is interesting to explore, but is out of the scope of this paper.

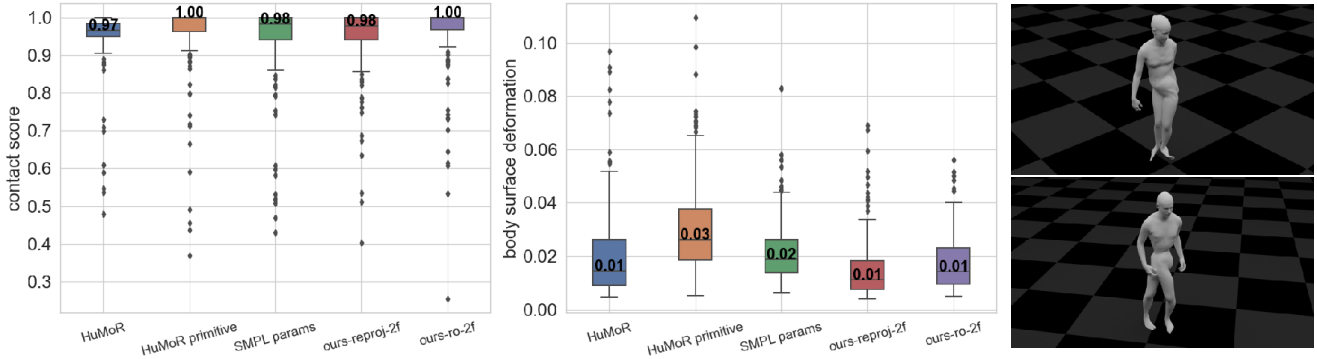

Figure S3. From left to right: contact scores, rigid body deformation measures, and typical failure cases of SMPL parameters (top) and HuMoR feature (bottom). Except 'HuMoR primitive' and 'SMPL params', other results are from Fig. 3.

A.5. Runtime Test

We perform a runtime test with a single Nvidia Quadro 6000 GPU. We generate motions in parallel for a batch of human bodies, and the results are shown in Tab. S3. This shows that our algorithm can produce motion efficiently. Note that each motion primitive spans 0.25 seconds, and therefore our method can simultaneously drive 256 SMPL-X bodies to move in real-time.

num. of bodies in a batch	1024	512	256	128	64
runtime (sec. per primitive)	0.85	0.5	0.25	0.15	0.1

Table S3. Runtime test on a single GPU.

B. More demonstrations on Methods

B.1. End-to-end Training of Generative Motion Primitives

Since all modules are differentiable, end-to-end training of GAMMA can be performed by minimizing

$$\mathcal{L} = \mathbb{E}_{\boldsymbol{\Theta}}[|M \circ \mathcal{M}(\boldsymbol{\Theta}, \boldsymbol{\beta}) - \boldsymbol{Y}|] + \mathbb{E}_{\boldsymbol{\Theta}}[|M \circ \Delta\mathcal{M}(\boldsymbol{\Theta}, \boldsymbol{\beta}) - \Delta\boldsymbol{Y}|] + \Psi\left(\text{KL-div}(q(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{Y})||\mathcal{N}(0, \boldsymbol{I}))\right) + \alpha|\boldsymbol{\theta}^h|^2, \quad (12)$$

in which the reconstruction terms are directly applied to the regressed body meshes. Such end-to-end training is much slower than training the marker predictor and the body regressor separately, particularly with the rollout.

In principle, this end-to-end training can make the motion generation more stable than training modules separately. For example, the body regressor can handle predicted markers as input, considering the prediction error accumulation. However, we don't observe benefits of such end-to-end training in our experiments (see Tab. 3 in Sec. 4). The error accumulation problem cannot be solved thoroughly. Out-of-distribution cases can happen, making the regressor fails and the predictor crashes. We think MOJO's reprojection approach is more robust, due to the pose regularizer in the optimization loss. However, it runs at 2sec/frame, which is too slow for long-term motion generation. We consider to propose a real-time and robust body optimizer in the future.

B.2. More Demonstrations about Policy Training

Besides the state, the action, and the reward, which are demonstrated in Sec. 3.3, we need another two variables for the PPO method.

Expected return. In our study, the expected return at time t, i.e., R_t, is based on the reward and a discount factor $0 < \gamma < 1$, which is set to 0.99 in our trials. The expected return is calculated as

$$R_t = \sum_{\tau=t}^{\infty} \gamma^{\tau-t} r_\tau, \tag{13}$$

in which the reward is defined in Eq. (4).

Advantage function. The advantage function is defined as the difference between the expected return and the value function produced by the critic network. It is given by

$$A(\boldsymbol{s}_t, \boldsymbol{z}_t) = A_t = R_t - V(\boldsymbol{s}_t). \tag{14}$$

In our trials, the advantage function A_t is normalized overall body clones in one simulation, which can empirically stabilize the training process.

The actor-critic network Compared to the actor (see Fig. 2), the critic network is only used during training for calculating the advantage function. It has a similar architecture to the actor and shares the GRU with it. The full actor-critic network is shown in Fig. S4.

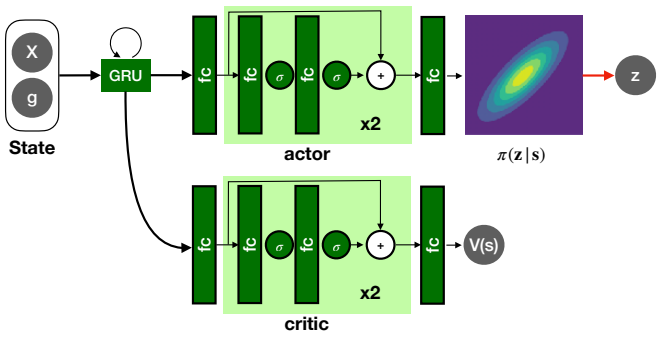

Figure S4. Architecture of the actor-critic network. Compared to the actor, the critic shares the GRU with it, and produces a scalar as the state value function.

PPO training. With the advantage function, the PPO term $\mathcal{L}_{PPO}$ in Eq.(10) is given by

$$\mathcal{L}_{PPO} = -\min\left(\frac{\pi(\boldsymbol{z}_t|\boldsymbol{s}_t)}{\pi^k(\boldsymbol{z}_t|\boldsymbol{s}_t)} A(\boldsymbol{z}_t, \boldsymbol{s}_t), \text{clip}\left(\frac{\pi(\boldsymbol{z}_t|\boldsymbol{s}_t)}{\pi^k(\boldsymbol{z}_t|\boldsymbol{s}_t)}, 1-\epsilon, 1+\epsilon\right) A(\boldsymbol{z}_t, \boldsymbol{s}_t)\right), \tag{15}$$

in which π^k is a pre-calculated value in the k-th simulation, ϵ is set to 0.2 in our trials. In addition, we stop updating the policy term when the KL-divergence between π and π^k, i.e. KLD$(\pi^k||\pi)$ is larger than 0.02. This PPO term is trained jointly with other terms.