PhysDiff: Physics-Guided Human Motion Diffusion Model Supplementary Material

Ye Yuan Jiaming Song Umar Iqbal Arash Vahdat Jan Kautz
NVIDIA
https://nvlabs.github.io/PhysDiff

1. Details of Evaluation Metrics

We use the open source code[1] of MDM [8] to compute the motion-based metrics: *FID*, *R-Precision*, and *Accuracy*. The physics-based metrics are implemented as follows. For ground penetration (*Penetrate*), we compute the distance between the ground and the lowest body mesh vertex below the ground. For floating (*Float*), we compute the distance between the ground and the lowest body mesh vertex above the ground. For both *Penetrate* and *Float*, we have a tolerance of 5 mm to account for geometry approximation. For foot sliding (*Skate*), we find foot joints that contact the ground in two adjacent frames and compute their average horizontal displacement within the frames. The overall physics error metric *Phys-Err* is the sum of *Penetrate*, *Float*, and *Skate*.

2. Details of Motion Diffusion

As mentioned in the main paper, we tested PhysDiff with two state-of-the-art denoiser networks, MDM [8] and MotionDiffuse [10] and showed that PhysDiff can improve both of them. We directly use the pretrained models in their codebase. Please refer to their paper and code for additional details.

For diffusion sampling, we use 50 timesteps with $\eta = 0$. We also use classifier-free guidance with the guidance coefficient set to 2.5. For text-to-motion generation on HumanML3D [1], the data is represented by a 263-dim vector that consists of 3D joint positions, rotations, and velocities, following Guo *et al*. [1]. To perform the physics-based motion projection, we first convert the 3D joint positions into joint angles of the SMPL model [3] using inverse kinematics and then apply physics-based motion imitation. For action-to-motion generation, the data is represented by joint rotations, so no inverse kinematics is required.

[1] https://github.com/GuyTevet/motion-diffusion-model

Parameter	Value
Num. of simulation environments	8192
Episode horizon	32
Num. of epochs	4000
Num. of mini-epochs	6
Learning rate	2×10^{-5}
PPO clip ϵ	0.2
Discount factor γ	0.99
GAE coefficient λ	0.95
Reward weights $(w_{\mathrm{p}}, w_{\mathrm{v}}, w_{\mathrm{j}}, w_{\mathrm{q}})$	(0.6, 0.1, 0.2, 0.1)
Reward parameters $(\alpha_{\mathrm{p}}, \alpha_{\mathrm{v}}, \alpha_{\mathrm{j}}, \alpha_{\mathrm{q}})$	(60, 0.2, 100, 40)
Elements of diagonal covariance $\mathbf{\Sigma}$	0.173

Table 1. Hyperparameters for physics-based motion imitation.

3. Details of Physics-Based Motion Imitation

Physics Simulation and Character. We use IsaacGym [5] as our physics simulator for its ability to perform massively parallel simulation on GPUs. The simulation runs at 60Hz while the policy controls the character at 30Hz. The character is automatically created from SMPL parameters following the approach in SimPoE [9].

Policy Training. The motion imitation policy uses a three-layer MLP with hidden dimensions (1024, 1024, 512) and ReLU activations. The elements of the policy's diagonal covariance matrix $\mathbf{\Sigma}$ are set to 0.173. We also normalize the policy's input state using a running estimate of the mean and variance of the state. We train the policy using the AMASS [4] human motion database. Since HumanML3D is a text-annotated version of AMASS, we use the same training split as HumanML3D and do not use additional data for fair comparison. We created 8192 parallel simulation environments in IsaacGym to collect training samples. Each RL episode has a horizon of 32 frames. We train the policy for 4000 epochs where each epoch collects 262,144 samples from running all environments for an episode. The reward weights $(w_{\mathrm{p}}, w_{\mathrm{v}}, w_{\mathrm{j}}, w_{\mathrm{q}})$ are set to (0.6, 0.1, 0.2, 0.1), and the reward parameters $(\alpha_{\mathrm{p}}, \alpha_{\mathrm{v}}, \alpha_{\mathrm{j}}, \alpha_{\mathrm{q}})$ are set to (60, 0.2, 100, 40). Proximal policy optimization (PPO [7])

is used to train the policy. The clipping coefficient ϵ in PPO is set to 0.2. The discount factor γ for the Markov decision process (MDP) is set to 0.99. We also use the generalized advantage estimator GAE(λ) [6] to estimate the advantage for policy gradient, and the GAE coefficient λ is 0.95. At the end of each epoch, we update the policy by iterating over the samples for 6 mini-epochs with a mini-batch size of 512. The update is performed via Adam [2] with a base learning rate of 2×10^{-5}. We clip the gradient if its norm is larger than 50.

References

[1] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5152–5161, 2022. 1

[2] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[3] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015. 1

[4] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5442–5451, 2019. 1

[5] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu based physics simulation for robot learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. 1

[6] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015. 2

[7] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 1

[8] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022. 1

[9] Ye Yuan, Shih-En Wei, Tomas Simon, Kris Kitani, and Jason Saragih. Simpoe: Simulated character control for 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1

[10] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *arXiv preprint arXiv:2208.15001*, 2022. 1