Generating Detailed Character Motion from Blocking Poses, Supplemental

1 Hyperparameters

1.1 Diffusion Model Hyperparameters for Pretrained *R* and *U*

- Number of Transformer Layers: 8
- Latent Dimension Size: 512
- Number of Attention Heads: 4
- Feed-forward Dimension Size: 1024
- Dropout: 0.1
- Activation: gelu
- Noise Schedule: cosine
- Training Batch Size: 64
- Learning rate: 10^{-4}

We train both models for about 24 hours on a NVIDIA Tesla V100 GPU. We run inference with T=1000 diffusion steps, which takes about 30 seconds for a batch size of 64 on a single GPU.

1.2 Constraint Refinement Module

We run constraint refinement every N diffusion steps. In our implementation, we have found that $N = 100$ works well. Empirically, we find that if N is too small, the diffusion model does not have sufficient time to produce a detailed, realistic prediction before the constraints are re-applied. We also have found it useful to skip constraint refinement at very high diffusion steps, e.g., $t > 700$, because the prediction of the denoised motion is not as coherent at this point.

2 Long Motions

The underlying models are trained to generate motions of a fixed length (60 frames). However, following observations in [Tseng et al.(2022), Goel et al.(2025)], the technique can be extended to handle arbitrary-length input $\mathbf{X}$. Tseng et al. [Tseng et al.(2022)] introduce an inference-time stitching procedure, where each intermediate subsequence is constrained such that its first half matches the last half of the preceding subsequence; the final predictions are then concatenated into a longer motion $\mathbf{Y}$. Goel et al. [Goel et al.(2025)] extend this idea by similarly preprocessing the input condition $\mathbf{X}$ into a batch of overlapping subsequences of length F, each sharing half of its frames with the previous one.

We adopt the same stitching procedure, but integrate it with our constraint refinement technique. At each refinement step, we first expand both the spliced condition $\mathbf{X}$ and intermediate subsequences $\mathbf{Y}^U$ to the full target motion length. We then apply the blending step described in the main paper to update $\mathbf{X}$. Finally, we re-splice $\mathbf{X}$ into its batched subsequence representation to serve as the condition for the next denoising steps of R.

3 Metrics

The goal of our quantitative evaluation is to study how well our techniques can generate *high quality* motion when given loose keyframe constraints. We evaluate quality using three standard metrics.

Jitter We calculate jitter as the average of the third derivative of joint positions, in world space. Jitter is a common quantitative metric for evaluating motion quality; higher jitter is an indicator for poor motion quality.

FootSkate We adopt the foot skating ratio metric from [Karunratanakul et al.(2023)], which measures the proportion of frames where a foot joint moves more than a predefined threshold while it is in ground contact. A higher *FootSkate* indicates worse physical realism due to excessive sliding of the feet during supposed contact.

FID Fréchet Inception Distance (FID) is a commonly used metric to evaluate the quality and realism of generated motion. It compares the distribution of features extracted from real and generated motions using a pretrained feature encoder (we convert our motions to the HumanML3D representation using code provided by [Petrovich et al.(2024)] and use an encoder provided by [Guo et al.(2020)]). Then, we fit a multivariate Gaussian to each set of features, and FID is calculated as the Frechet distance between the two distributions. We calculate FID over the entire test set, using all test motions $\mathbf{Y}$ as the ground truth and comparing them to corresponding generated motions. Lower FID values indicate that the generated motions are statistically more similar to real motions, reflecting higher realism.

4 Motion Representation

We use the same global motion representation as Goel et al [Goel et al.(2025)]. Each motion $\mathbf{X}$ and $\mathbf{Y}$ is represented as a sequence of poses in the SMPL format [Loper et al.(2015)]. For pose state at frame f, we represent each of the 24 joint angles using a 6D continuous representation [Zhou et al.(2020)]. Each pose also contains a single 3-dim global translation. We use a binary label for the heel and toe of both feet to represent contact with the ground, $\mathbf{b} \in \{0, 1\}$. We also include the global joint positions as a redundant representation.

5 Reproducibility

We will release all code and metadata for others to build on.

References

[Goel et al.(2025)] Purvi Goel, Haotian Zhang, C Karen Liu, and Kayvon Fatahalian. 2025. Generative Motion Infilling from Imprecisely Timed Keyframes. In *Computer Graphics Forum*. Wiley Online Library, e70060.

[Guo et al.(2020)] Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. 2020. Action2motion: Conditioned generation of 3d human motions. In *Proceedings of the 28th ACM International Conference on Multimedia*. 2021–2029.

[Karunratanakul et al.(2023)] Korrawe Karunratanakul, Konpat Preechakul, Supasorn Suwajanakorn, and Siyu Tang. 2023. Guided Motion Diffusion for Controllable Human Motion Synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2151–2162.

[Loper et al.(2015)] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-Person Linear Model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 34, 6 (Oct. 2015), 248:1–248:16.

[Petrovich et al.(2024)] Mathis Petrovich, Or Litany, Umar Iqbal, Michael J. Black, Gül Varol, Xue Bin Peng, and Davis Rempe. 2024. Multi-Track Timeline Control for Text-Driven 3D Human Motion Generation. arXiv:2401.08559 [cs.CV] https://arxiv.org/abs/2401.08559

[Tseng et al.(2022)] Jonathan Tseng, Rodrigo Castellon, and C Karen Liu. 2022. EDGE: Editable Dance Generation From Music. *arXiv preprint arXiv:2211.10658* (2022).

[Zhou et al.(2020)] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. 2020. On the Continuity of Rotation Representations in Neural Networks. arXiv:1812.07035 [cs.LG]