Supplemental Material for TEDi: Temporally-Entangled Diffusion for Long-Term Motion Synthesis

1 IMPLEMENTATION DETAILS

Our TEDi framework is implemented with PyTorch, with training and inference performed on an NVIDIA A40 GPU. The diffusion denoising model architecture is a 1D UNet, similar to a 2D UNet used in regular image diffusion models. We use 2 layers for downsampling and 2 layers for upsampling. We use Adam as our optimizer with lr= $1e-4$. For the training data, we use motions from the CMU motion dataset which are natively 120fps. We then sample windows of 500 frames with a stride of 100 frames. There are 2548 motions in total and we discard 721 which contain less than 500 frames. We obtain 30k 500-frame windows after processing. In particular, this means that we choose $K = T = 500$, where T is the total number of diffusion steps and K is the motion buffer length (Sec. 3.1). As a remark, a larger value of K requires the dataset to have sufficiently many longer sequence motions, which is more restrictive. With $K < T$, where the motion buffer is smaller than the number of diffusion steps, we will have to apply multiple denoising steps in each buffer iteration before we can obtain a clean frame. Thus, we will not obtain a clean frame in every diffusion step. This will also result in larger gaps in similarity between two consecutive frames due to the increased gap between their corresponding noises.

In addition to the two noise injection schemes introduced in Sec. 3.3, we incorporate curriculum training with the regular diffusion loss for 100k iterations, and a step scheduler with $\gamma = 0.1$ is used for the learning rate. In particular, the regular diffusion loss determines a fixed β_t per batch, and it exposes the model to an easier optimization objective that encourages smoothness between frames and helps the positional and foot contact losses (Sec. 3.3.1) to better regularize training. Training takes approximately three days for 500k iterations.