Appendix of "MotionStreamer: Streaming Motion Generation via Diffusion-based Autoregressive Model in Causal Latent Space"

A. Implementation Details

For the Causal TAE, both the encoder and decoder are based on the 1D causal ResNet blocks [6]. The temporal downsampling rate l is set to 4 and all motion sequences are cropped to $N = 64$ frames during training. We train the first 1900K iterations with a learning rate of 5e-5 and the remaining 100K iterations with a learning rate of 2.5e-6. We use the AdamW optimizer [9] with $[\beta_1, \beta_2] = [0.9, 0.99]$ and a batch size of 128. We provide an ablation study on the hyperparameter λ of root loss L_{root} in Tab. 1. The latent dimension d_c and hidden size are set to 16 and 1024, respectively. The latent dimension significantly impacts the compression rate, while the hidden size affects the model's capacity. Both factors influence reconstruction and subsequent generation quality, requiring a careful trade-off between compression efficiency and generative performance. Ablation studies on the latent dimension and hidden size are provided in Tab. 2. To further improve the quality of the reconstructed motion, we add a linear layer after the embedded Gaussian distribution parameters as a latent adapter to get a lower-dimensional and more compact latent space for subsequent sampling, as proposed in [2].

For the Transformer inside the AR model, we use the architecture akin to LLaMA [14] with 12 layers, 12 attention heads and 768 hidden dimension. The ablation for different scales of the Transformer is provided in Tab. 3. Block size is set to 78 and we choose RoPE [12] as the positional encoding. For the diffusion head after Transformer, we use MLPs with 1792 hidden dimension and 9 layers. The output vectors of the Transformer serve as the condition of denoising via AdaLN [11]. We adopt a cosine noise schedule with 50 steps for the DDPM [8] denoising process following [13]. During training, the minimum and maximum length of motion sequences are set to 40 and 300 for both datasets. We insert an additional reference end latent at the end of each motion sequence to indicate the stop of generation. For Two-Forward strategy, a cosine scheduler is employed to control the ratio of replaced motion tokens, which can be formulated as: $\gamma_t = \frac{1}{2}(1 - \cos(\frac{\pi t}{T}))$, where t is current iteration step and T is the total number of iterations. When $t = 0$, $\gamma_t = 0$, indicating that no generated motion tokens in the first forward pass are replaced, thus relying on the ground-truth motion tokens only. When $t = T$, $\gamma_t = 1$, indicating that all generated motion tokens in the first forward pass are replaced, thus relying on the generated motion tokens only. We use the same optimizer as the Causal TAE and a batch size of 256. The initial learning rate is 1e-4 after 10K warmup iterations and decay to 0 for another 90K iterations using cosine learning rate scheduler. Our experiments are conducted on A800 GPUs.

λ	FID ↓	MPJPE ↓
5.0	0.696	25.2
6.0	0.684	24.8
7.0	**0.661**	**22.9**
8.0	0.682	24.2
9.0	0.704	26.8

Table 1. **Analysis of** λ on the HumanML3D [4] test dataset.

B. Causal TAE Architecture

The detailed architecture of the Causal TAE is shown in Fig. 3 and Tab. 4. Input motion sequences are first encoded into a latent space with a 1D causal ResNet. The latent space is then projected to a sequence of Gaussian distribution parameters. Then a linear adapter is applied to the embedded Gaussian distribution parameters to lower the dimension of latent space. Sampling is performed in the lower-dimensional latent space. The decoder comprises a mirror process to progressively reconstruct the motion sequence.

C. AR Model Architecture

We provide an ablation study on the architecture of the AR model, including the number of Transformer layers, attention heads, hidden dimension, and the number of diffusion head layers, as shown in Tab. 3. We finally leverage the 12-layer, 12-head, 768-hidden dimension, and 9-layer diffusion head architecture.

D. Classifier-free guidance

We adopt the classifier-free guidance (CFG) [7] technique to improve the generation quality of the autoregressive motion generator. Specifically, during training, we replace

Methods	Reconstruction		Generation					
	FID ↓	MPJPE ↓	FID ↓	R@1 ↑	R@2 ↑	R@3 ↑	MM-D ↓	Div →
Real motion	-	-	0.002	0.702	0.864	0.914	15.151	27.492
(12,512)	8.862	38.5	21.078	0.600	0.759	0.827	17.143	27.456
(12,1024)	1.710	31.2	12.778	0.628	0.779	0.845	16.756	27.408
(12,1280)	2.035	32.9	12.872	0.624	0.780	0.849	16.587	27.455
(12,1792)	1.563	28.3	11.916	0.628	0.782	0.850	16.468	27.461
(12,2048)	1.732	28.9	13.394	0.611	0.770	0.831	16.852	27.417
(14,512)	2.902	33.6	16.612	0.607	0.772	0.836	16.947	27.328
(14,1024)	0.838	27.5	11.933	0.627	0.778	0.840	16.593	27.443
(14,1280)	0.919	26.4	12.603	0.603	0.772	0.841	16.863	27.414
(14,1792)	0.732	24.8	11.828	0.628	0.776	0.848	16.652	27.122
(14,2048)	1.370	26.5	12.261	0.621	0.768	0.841	16.734	27.417
(16,512)	1.300	30.3	14.096	0.605	0.770	0.839	16.882	27.306
(16,1024)	0.661	22.9	**11.790**	**0.631**	**0.802**	**0.859**	**16.081**	27.284
(16,1280)	1.087	25.0	12.975	0.598	0.761	0.831	17.002	27.403
(16,1792)	0.540	22.0	11.992	0.630	0.767	0.846	16.644	27.419
(16,2048)	1.547	26.2	12.778	0.604	0.755	0.824	16.897	27.306
(18,512)	2.043	27.7	19.150	0.553	0.701	0.775	17.776	27.345
(18,1024)	0.656	23.4	11.838	0.619	0.775	0.840	16.816	27.356
(18,1280)	0.820	23.1	11.815	0.629	0.801	0.847	16.816	27.461
(18,1792)	1.045	22.1	12.514	0.612	0.774	0.840	16.915	27.412
(18,2048)	0.595	21.5	11.803	0.613	0.801	0.832	17.004	27.451
(20,512)	0.531	24.5	12.247	0.613	0.765	0.832	16.920	27.277
(20,1024)	**0.379**	**19.9**	11.814	0.630	0.765	0.847	16.802	27.485
(20,1280)	0.429	20.1	16.465	0.557	0.705	0.774	17.680	**27.490**
(20,1792)	0.548	20.1	11.845	0.616	0.776	0.842	16.919	27.392
(20,2048)	0.690	20.7	11.910	0.625	0.782	0.844	16.785	27.346

Table 2. **Ablation Study of different Causal TAE architecture designs** on HumanML3D [4] test set. Each generation model remains the same. MPJPE is measured in millimeters. (16, 1024) indicates the latent dimension and hidden size of the Causal TAE.

10% of the text within a batch with a blank text as unconditioned samples, while during inference, CFG is applied to the denoising process of the diffusion head, which can be formulated as:

$$\epsilon_g = \epsilon_u + s(\epsilon_c - \epsilon_u). \tag{1}$$

where ϵ_g is the guided noise, ϵ_u is the unconditioned noise, ϵ_c is the conditioned noise, s is the CFG scale. We provide an ablation study on the CFG scale s in Fig. 1. Finally, we choose $s = 4.0$ for all experiments.

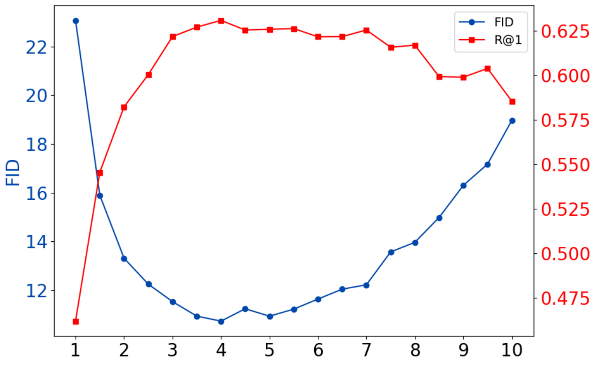

Figure 1. **Ablation of CFG scale** on HumanML3D [4] test set. $scale = 1$ means do not use CFG.

E. Failure of Inverse Kinematics

Post-processing for 263-dimensional motion representation. Most previous works [5, 13, 15] uses 263-dimensional motion representation [4].

AR. layers	AR. heads	AR. dim	Diff. layers	FID ↓	R@1 ↑	R@2 ↑	R@3 ↑	MM-D ↓	Div →
8	8	512	2	14.336	0.598	0.747	0.802	16.983	27.287
8	8	512	3	13.764	0.602	0.758	0.819	16.972	27.242
8	8	512	4	12.893	0.608	0.764	0.828	16.661	27.351
8	8	512	9	11.823	0.623	0.772	0.835	16.655	27.385
8	8	512	16	12.460	0.621	0.778	0.849	16.784	27.410
12	12	768	2	11.899	0.601	0.763	0.828	16.952	27.406
12	12	768	3	11.798	0.630	0.779	0.844	16.761	**27.482**
12	12	768	4	12.051	0.604	0.762	0.829	16.940	27.401
12	12	768	9	**11.790**	**0.631**	**0.802**	**0.859**	**16.081**	27.284
12	12	768	16	11.825	0.624	0.773	0.844	16.757	27.341
16	16	1024	2	12.836	0.606	0.765	0.832	16.901	27.319
16	16	1024	3	12.436	0.601	0.761	0.830	16.919	27.302
16	16	1024	4	13.005	0.614	0.763	0.830	16.967	27.196
16	16	1024	9	12.093	0.614	0.778	0.843	16.850	27.308
16	16	1024	16	11.812	0.630	0.780	0.846	16.598	27.286

Table 3. **Ablation study of AR Model architecture** on HumanML3D [4] test set. For each architecture, we use the same Causal TAE.

The representation can be written as follows:

$$x = \{\dot{r}^a, \dot{r}^x, \dot{r}^z, r^y, j^p, j^v, j^r, c\}, \tag{2}$$

where the root is projected on the XZ-plane (ground plane), $\dot{r}^a \in \mathbb{R}^1$ denotes root angular velocity along the Y-axis, $(\dot{r}^x, \dot{r}^z \in \mathbb{R})$ are root linear velocities on the XZ-plane, $r^y \in \mathbb{R}$ is the root height, $j^p \in \mathbb{R}^{3(K-1)}$, $j^v \in \mathbb{R}^{3K}$, and $j^r \in \mathbb{R}^{6(K-1)}$ are local joint positions, local velocities, and local rotations relative to the root, K is the number of joints (including the root), and $c \in \mathbb{R}^4$ is the contact label. For SMPL characters, we have $K = 22$ and we get $2+1+1+3\times 21+3\times 22+6\times 21+4 = 263$ dimensions. In the original implementation [4], the joint rotation is directly solved using Inverse Kinematics (IK) with relative joint positions. In such way, the joint loses twist rotation and directly applying the joint rotation to the character faces a lot of rotation error [3], as shown in Fig. 2. To overcome this issue, previous works [5, 13, 15] only uses the positions and employs SMPLify [1] to solve the real SMPL joint rotation. This process is time-consuming (around 60 seconds for a 10 seconds motion clip) and also introduces unnatural results like jittering head [13]. Most data in the HumanML3D [4] dataset comes from the AMASS [10] dataset. As the AMASS dataset provides the SMPL joint rotation, we slightly modify the motion representation by directly using the SMPL joint rotation and make it a 6D rotation for better learning. Consequently, we remove the slow post-processing step and easily drive the SMPL character with the generated rotations. The processing scripts to obtain our 272-dim motion representation are available at `https://github.com/Li-xingXiao/272-dim-Motion-Representation`.

F. Limitations and Future work

Limitations. Despite its effectiveness, the streaming generation paradigm limits the applications of motion inbetweening and localized editing of intermediate tokens, as it inherently relies on unidirectional modeling. This limitation restricts flexibility in scenarios requiring fine-grained adjustments, such as seamlessly inserting new motions between existing frames or interactively refining motion details while preserving global coherence.

Future work. Future work could explore hybrid strategies that allow bidirectional refinement without compromising streaming generation. One potential way is to predict a set of future latents at each step, which could enable motion inbetween and localized editing while preserving streaming manner.

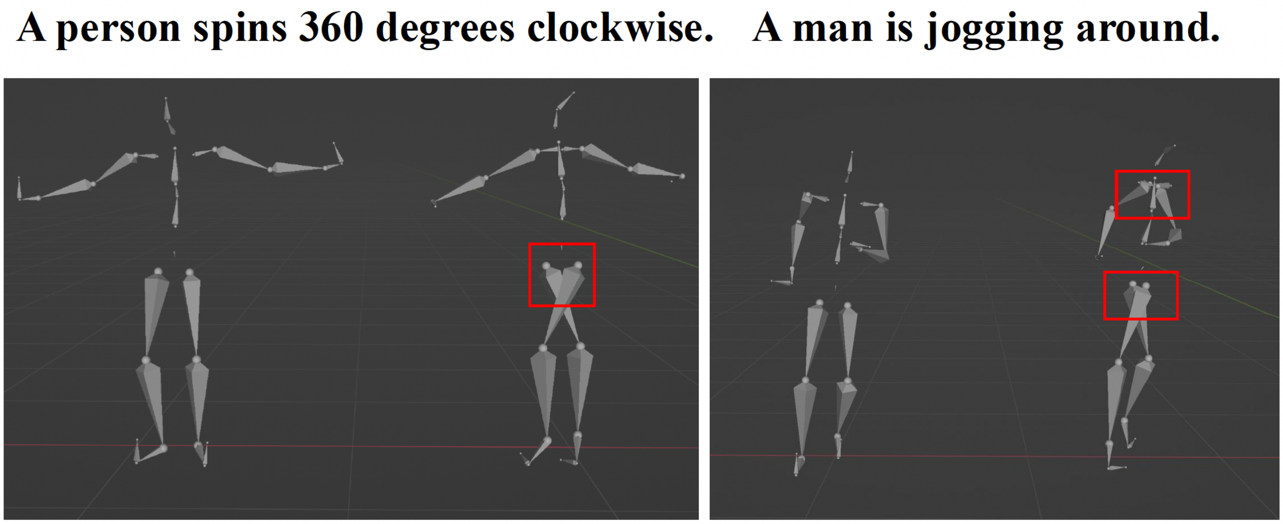

Figure 2. **Failure of Inverse Kinematics.** The joint rotation is directly solved using IK with relative joint positions, which leads to unnatural results like jittering body parts.

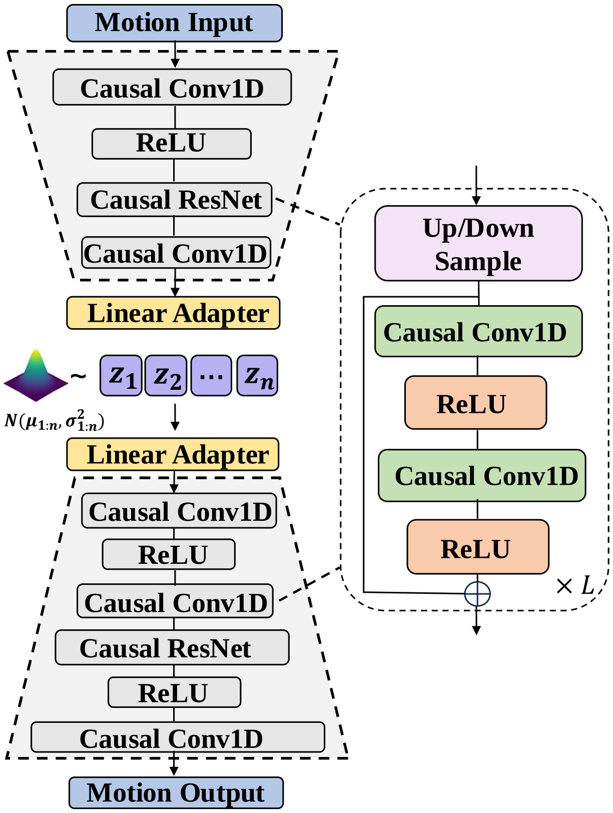

Figure 3. **Architecture of Causal TAE.** Motion latents are sampled in a continuous causal latent space.

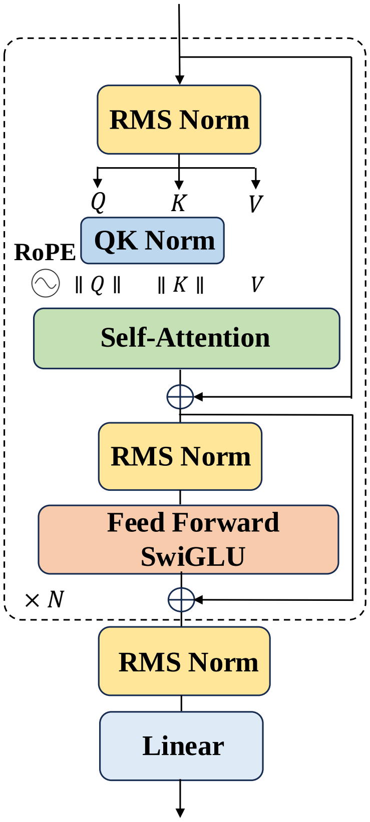

Figure 4. **Architecture of Transformer blocks in AR model.** QK Norm is applied to enhance training stability.

Components	Architecture
Causal TAE Encoder	(0): CausalConv1D(D_{in}, 1024, kernel_size=(3,), stride=(1,), dilation=(1,), padding=(2,))
	(1): ReLU()
	(2): 2 × Sequential(
	(0): CausalConv1D(1024, 1024, kernel_size=(4,), stride=(2,), dilation=(1,), padding=(2,))
	(1): CausalResnet1D(
	(0): CausalResConv1DBlock(
	(activation1): ReLU()
	(conv1): CausalConv1D(1024, 1024, kernel_size=(3,), stride=(1,), dilation=(9,), padding=(18,))
	(activation2): ReLU()
	(conv2): CausalConv1D(1024, 1024, kernel_size=(1,), stride=(1,), dilation=(1,), padding=(0,)))
	(1): CausalResConv1DBlock(
	(activation1): ReLU()
	(conv1): CausalConv1D(1024, 1024, kernel_size=(3,), stride=(1,), dilation=(3,), padding=(6,))
	(activation2): ReLU()
	(conv2): CausalConv1D(1024, 1024, kernel_size=(1,), stride=(1,), dilation=(1,), padding=(0,)))
	(2): CausalResConv1DBlock(
	(activation1): ReLU()
	(conv1): CausalConv1D(1024, 1024, kernel_size=(3,), stride=(1,), dilation=(1,), padding=(2,))
	(activation2): ReLU()
	(conv2): CausalConv1D(1024, 1024, kernel_size=(1,), stride=(1,), dilation=(1,), padding=(0,)))))
	(3): CausalConv1D(1024, 1024, kernel_size=(3,), stride=(1,), dilation=(1,), padding=(2,))
Causal TAE Decoder	(0): CausalConv1D(1024, 1024, kernel_size=(3,), stride=(1,), dilation=(1,), padding=(2,))
	(1): ReLU()
	(2): 2 × Sequential(
	(0): CausalResnet1D(
	(0): CausalResConv1DBlock(
	(activation1): ReLU()
	(conv1): CausalConv1D(1024, 1024, kernel_size=(3,), stride=(1,), dilation=(9,), padding=(18,))
	(activation2): ReLU()
	(conv2): CausalConv1D(1024, 1024, kernel_size=(1,), stride=(1,), dilation=(1,), padding=(0,)))
	(1): CausalResConv1DBlock(
	(activation1): ReLU()
	(conv1): CausalConv1D(1024, 1024, kernel_size=(3,), stride=(1,), dilation=(3,), padding=(6,))
	(activation2): ReLU()
	(conv2): CausalConv1D(1024, 1024, kernel_size=(1,), stride=(1,), dilation=(1,), padding=(0,)))
	(2): CausalResConv1DBlock(
	(activation1): ReLU()
	(conv1): CausalConv1D(1024, 1024, kernel_size=(3,), stride=(1,), dilation=(1,), padding=(2,))
	(activation2): ReLU()
	(conv2): CausalConv1D(1024, 1024, kernel_size=(1,), stride=(1,), dilation=(1,), padding=(0,)))))
	(1): Upsample(scale_factor=2.0, mode=nearest)
	(2): CausalConv1D(1024, 1024, kernel_size=(3,), stride=(1,), dilation=(1,), padding=(2,))
	(3) CausalConv1D(1024, 1024, kernel_size=(3,), stride=(1,), dilation=(1,), padding=(2,))
	(4): ReLU()
	(5): CausalConv1D(1024, D_{in}, kernel_size=(3,), stride=(1,), dilation=(1,), padding=(2,))

Table 4. **Detail architecture** of the proposed Causal TAE.