BAMM: Bidirectional Autoregressive Motion Model

Supplementary Material

A Overview

The supplementary material is organized into the following sections:

- Section B: Length prediction vs length restriction
- Section C: Length diversity with high-quality motion generation.
- Section D: Temporal Motion Editing
- Section E: Implementation Details
- Section F: Limitation

B Length prediction vs length restriction

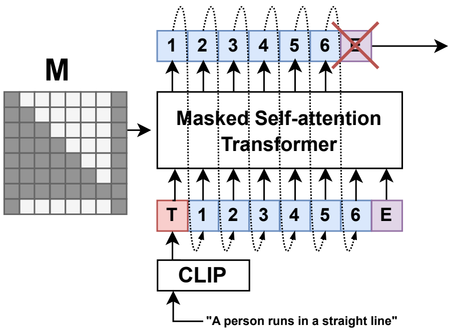

M
1
2
3
4
5
6
E
Masked Self-attention Transformer
T
1
2
3
4
5
6
E
CLIP
"A person runs in a straight line"

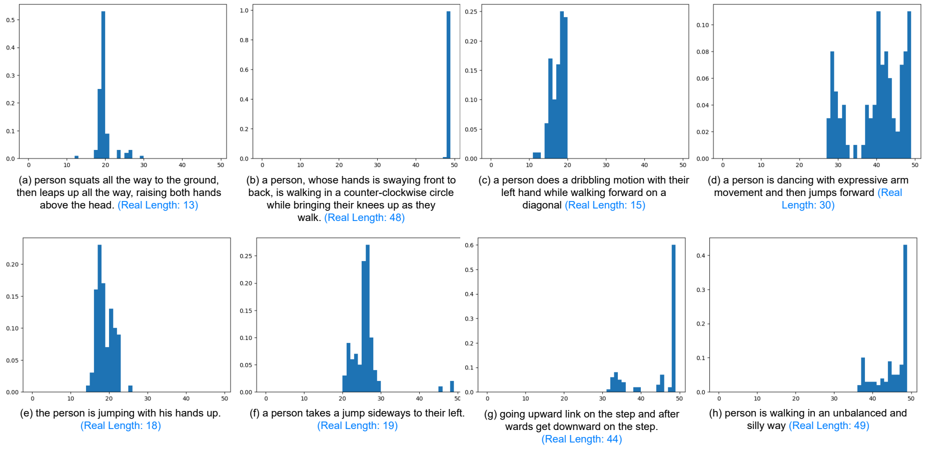
Fig. 8: Generate motion with length constrain by input [END] as a condition and remove [END] output prediction.

Length prediction. Naturally, BAMM has the ability to predict the [END] token to stop generating when it seems appropriate which automatically predicts the correlated motion length from previous token conditions without relying on an external length estimator as shown in the first iteration of Fig. 3.

Length restriction. In tasks such as temporal motion editing that require specific motion lengths, our model can generate motion constrained by input motion length. This is achieved by applying the [END] token as an input condition to constrain where generation should stop. During training, the [END] token is already randomly conditioned. However, in this scenario, [END] serves as an input condition rather than an output prediction. To ensure uninterrupted generation until reaching the desired length without prematurely stopping due to [END] predictions, we force the model to predict only the K indices of the codebook, explicitly excluding [END] predictions from the output logits. Therefore, the first iteration in Fig. 3 can be modified to Fig. 8.

C Length diversity with high-quality motion generation

The benefit of our BAMM model's integrated length predictor is that it enhances motion realism and quality, as the model can re-evaluate every iteration. Additionally, the generated length is of a broader range, reflecting the diversity of motion while being correlated with the currently generated motions. The histogram in Fig. 9 illustrates various motion token lengths generated from the same textual description. For each textual description example, we generate 1000 samples and calculate the probability density of the predicted number of token lengths. Given a prompt, the predicted length of BAMM is generally diverse. The motions involving detailed, lengthy and sequential actions tend to have the maximum motion length, aligning closely with the ground truth, as shown in Fig. 9 (b).



(a) person squats all the way to the ground, then leaps up all the way, raising both hands above the head. (Real Length: 13)

(b) a person, whose hands is swaying front to back, is walking in a counter-clockwise circle while bringing their knees up as they walk. (Real Length: 48)

(c) a person does a dribbling motion with their left hand while walking forward on a diagonal (Real Length: 15)

(d) a person is dancing with expressive arm movement and then jumps forward (Real Length: 30)

(e) the person is jumping with his hands up. (Real Length: 18)

(f) a person takes a jump sideways to their left. (Real Length: 19)

(g) going upward link on the step and after wards get downward on the step. (Real Length: 44)

(h) person is walking in an unbalanced and silly way (Real Length: 49)

Fig. 9: Histogram of motion token lengths. 1000 motions are generated for each textual description to calculate the estimated probability density of the token length. The corresponding lengths from the dataset HumanML3D [16] are called Real Length and highlighted in **blue** text. The length of motion is four times the token length.

In contrast, the models that rely on separated length estimators not only suffer from inaccurate motion length, but also lack diversity in the generated motions. We demonstrate this effect on the experiment with a pre-trained length estimator from [16] for MMM [29] and MoMask [14], both of which require input length methods. To investigate the impact of length diversity on the quality of generated motion, we compare the motion generation performance under two motion length sampling strategies: top-1 sampling and multinomial sampling In Table 4. The top-1 sampling always chooses the motion length predicted with the highest probability or confidence by the length estimator. Multinomial sampling generates random motion length drawn from the prediction probability or confidence distribution. As shown in Fig. 9, both MMM and MoMask experience degraded performance in terms of R-precision and FID when multinomial sampling is adopted. This is because multinomial sampling can generate diverse motion lengths that the models cannot adapt to.

Table 7: Comparison of text-conditional motion synthesis using different length samping stategies on HumanML3D [16] dataset

Methods	R-Precision ↑			FID ↓	MM-Dist ↓	Diversity ↑	MModality ↑
	Top-1 ↑	Top-2 ↑	Top-3 ↑				
MMM Top-1	0.504	0.696	0.794	0.080	2.998	9.411	1.164
MMM Multinomial	0.492	0.685	0.782	0.099	3.063	9.319	1.18
MoMask Top-1	0.522	0.715	0.811	0.090	2.945	9.647	1.239
MoMask Multinomial	0.520	0.713	0.809	0.120	2.957	**9.731**	1.235
BAMM (Ours)	**0.525**	**0.720**	**0.814**	**0.055**	**2.919**	9.717	**1.687**

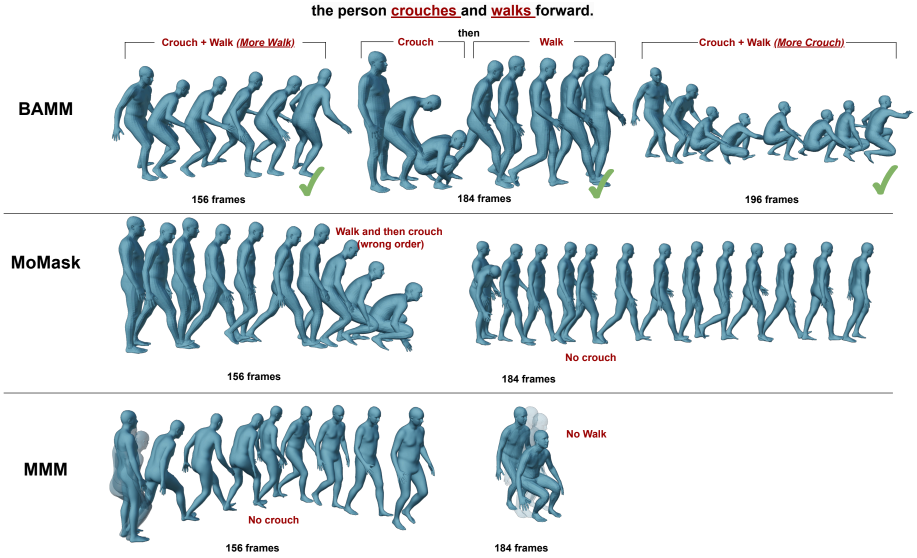

the person crouches and walks forward.
Crouch + Walk (More Walk)
Crouch
then
Walk
Crouch + Walk (More Crouch)
BAMM
156 frames
184 frames
196 frames
MoMask
Walk and then crouch (wrong order)
156 frames
No crouch
184 frames
MMM
No crouch
156 frames
No Walk
184 frames

Fig. 10: Visualization comparing different input lengths to state-of-the-art methods with the prompt "the person crouches and walks forward." with ground truth length of 196 frames. BAMM generates diverse motions correlated with various lengths while MMM and MoMask are sensitive to the different length inputs.

In Fig. 10, we demonstrate how a single prompt can lead to variations in motion, showcasing the diversity of motion correlated with different lengths. Using the textual description "the person crouches and walks forward.", we observe different interpretations of the motion generated by BAMM. For instance, the first and last samples show variations such as ’crouching while walking forward,’ with the last sample exhibiting a deeper crouch. In contrast, the middle sample depicts separate actions of ’crouching’ and ’walking forward.’ Each sample has a unique length corresponding to its motion. However, MoMask and MMM are sensitive to varying lengths, resulting in inaccuracies in their generated motions when the lengths are not precise.

D Temporal Motion Editing

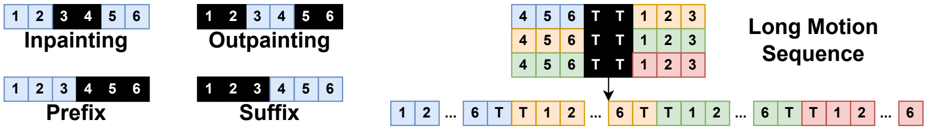


Fig. 11: Visualization of masking and conditional tokens for five temporal motion editing tasks: inpainting (in-betweening), outpainting, prefix, suffix, and long motion sequence. ■ indicates masked positions/areas

Since our BAMM model can utilize conditional tokens as inputs for generation, temporal motion editing can be accomplished by predicting the tokens in the masked positions that need modifications, conditioned on the unmasked tokens and text prompt, as illustrated in Fig. 11. The visualization results are in Fig. 7. In addition, the editing tasks are performed in the zero-shot manner. This means that during the model training, we do not apply any specific masks that correspond to editing tasks as shown in Fig 11 (left). Instead, we just randomly put 50% – 100% motion tokens in the masked areas.

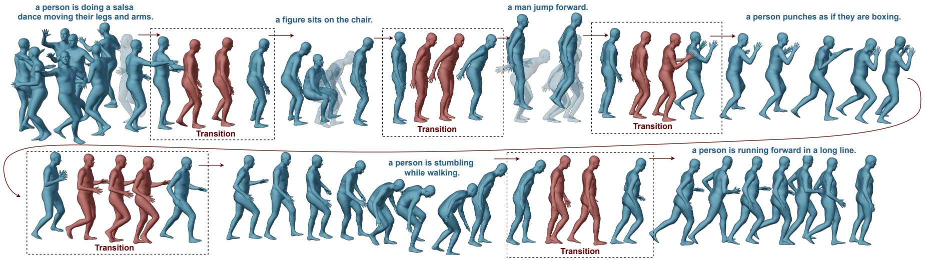

a person is doing a salsa dance moving their legs and arms.
a figure sits on the chair.
a man jump forward.
a person punches as if they are boxing.
Transition
Transition
Transition
Transition
a person is stumbling while walking.
a person is running forward in a long line.
Transition

Fig. 12: Visualization of Long Motion Sequence where **blue** frames represent individual motion segments prompted by textual descriptions. **Red** frames depict the intermediate transitions between these prompted segments, ensuring temporal coherence across the entire sequence.

Long Motion Sequence. Generating arbitrarily long motions presents a challenge due to the limited length of motion data in available datasets such as HumanML3D [16] and KIT [30], where no sample exceeds a duration of 10 seconds. To tackle this issue, we utilize the trained masked motion model as a prior for synthesizing long motion sequences without requiring additional training. Specifically, given a story consisting of multiple text prompts, our model first generates the motion token sequence for each prompt. Then, it generates transition motion tokens conditioned on the end of the previous motion sequence and the start of the next motion sequence.

E Implementation Details

The Motion tokenizer comprises six quantization layers, each with 512 codes and 512 embedding dimensions, along with skip connection and a dropout ratio of 0.2. Both the Masked Self-attention Transformer and Refinement Transformer consist of a six-layer encoder-only transformer architecture with six heads and an embedding size of 384. The batch size is set to 512 for both Motion Tokenizer and Masked Self-attention, while it is 64 for the Refinement Transformer. We use AdamW for optimization with a learning rate of 2e-4 which decreases by a factor of ten at 50,000 and 80,000 iterations. A masking ratio of 0.5 is applied for λ. During training, ground truth input is randomly replaced with random tokens with a probability of $\tau = 0.5$. For HumanML3D, the CFG scales are set to 4, 3, and 6 for the first, the second stages, and Residual Motion Refinement, respectively. For KIT, the corresponding scales are 2, 2, and 6.

F Limitation

While BAMM offers high-quality motion generation, it is important to note that its processing speed is slower in comparison to parallel decoding methods like MMM or MoMask. This delay stems from BAMM's cascaded generation process, which includes a unidirectional autoregressive decoding process followed by a bidirectional autoregressive decoding procedure and a residual motion refinement step. Despite this, it is worth mentioning that BAMM still outperforms motion space diffusion techniques such as MDM and MotionDiffuse in terms of speed by a large margin. Additionally, with an average generation time of 0.411 seconds per sample on an NVIDIA RTX A5000, BAMM remains sufficiently fast for practical use.