
Supplementary Material for “Smooth and Flexible Camera Movement Synthesis via Temporal Masked Generative Modeling”

Anonymous Author(s)

Affiliation

Address

email

1 A Differences Between Real-time and Offline Paradigms

2 We illustrate the diagrams of the real-time and offline inference process in Figure 1. For real-time
3 inference, we append a [MASK] token following the corresponding camera tokens (Some tokens next
4 to the token we need to predict) and utilize both the long- and short-term memories as conditional
5 inputs to predict the masked token. For offline inference, we removed the input of long-term memories
6 and only used $\tilde{\mathbf{A}}_t^s$ and $\tilde{\mathbf{M}}_t^s$ as conditions to generate the corresponding camera movement. We input
7 the [MASK] token of equal length into the Temporal Conditional Masked Transformer (CMT) and
8 take the corresponding output as the generated result.

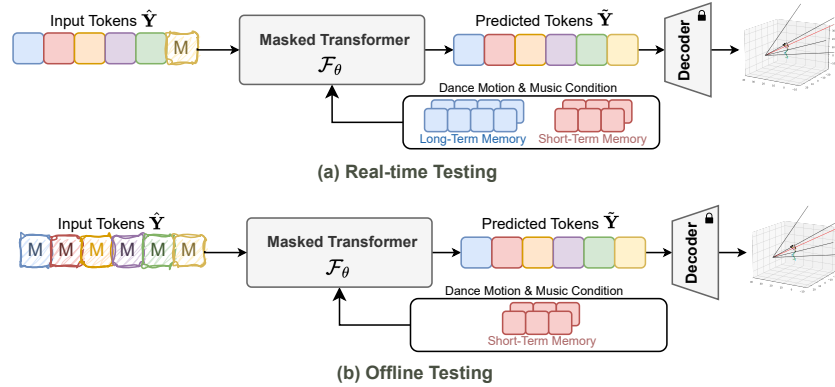


Figure 1: Diagram of the real-time and offline inference process.

9 B Additional Experiments

10 B.1 Test Computational Consumption Analysis

11 DanceCamera3D [1] employs a diffusion model to generate camera movements, requiring multiple
12 denoising steps during the testing phase to produce the final outputs. Under equivalent testing condi-
13 tions, our method achieves a generation speed approximately 40 times faster than DanceCamera3D,
14 which utilizes DDIM as its sampler. DanceCamAnimator [2], on the other hand, generates complete
15 camera movements progressively through three stages, beginning with keyframes. Unlike an end-
16 to-end approach, this multi-stage process introduces cumulative errors and requires greater human
17 intervention. In comparison, TemMEGA reduces these errors and dependencies while achieving a
18 nearly three times faster generation speed than DanceCamAnimator.

19 B.2 Generative Continuity Analysis

20 We analyzed the continuity of long sequence generation by DanceCamera3D, DanceCamAnimator,
 21 and TemMEGA. During DanceCamera3D inference, they generate 5-second subsequences with
 22 a stride of 2.5 seconds, then interpolate the overlapping slices to enforce consistency with linear
 23 decaying weight. The intersection of the two clips is not generated naturally, so the continuity
 24 is not good enough. DanceCamAnimator uses a three-stage strategy to improve the continuity
 25 problem. However, when the test sequence is short and the keyframe cannot be found, the first stage
 26 of DanceCamAnimator cannot work properly, which makes the entire pipeline unusable. This is
 27 why DanceCamAnimator cannot complete the generation under real-time settings. TemMEGA can
 28 achieve generation under different paradigms by changing the number and position of [MASK].
 29 TemMEGA can achieve generation under different paradigms by changing the number and position
 30 of [MASK]. For long sequence generation, we can treat the generated part as the first half of the input
 31 token and fill the second half with [MASK] to achieve prediction, so our method performs better in
 32 terms of continuity.

33 B.3 More Visualization Results

34 **Camera Movement Visualization Results.** We selected two segments of generated results and
 35 visualized them every 40 frames in Figure 2. We also provide video files, please check out other files
 36 (Camera1.mp4, Camera2.mp4, Camera3.mp4).

37 **Rendering Visualization Results.** We provide video files, please check out other files (Ren-
 38 dering1.mp4, Rendering2.mp4, Rendering3.mp4). Accompanying video files (Online_video.mp4,
 39 Offline_video.mp4) are provided to visually compare our method with state-of-the-art approaches
 40 under both online and offline settings. The results on the right correspond to our method, whereas
 41 those on the left are generated by DanceCamera3D.

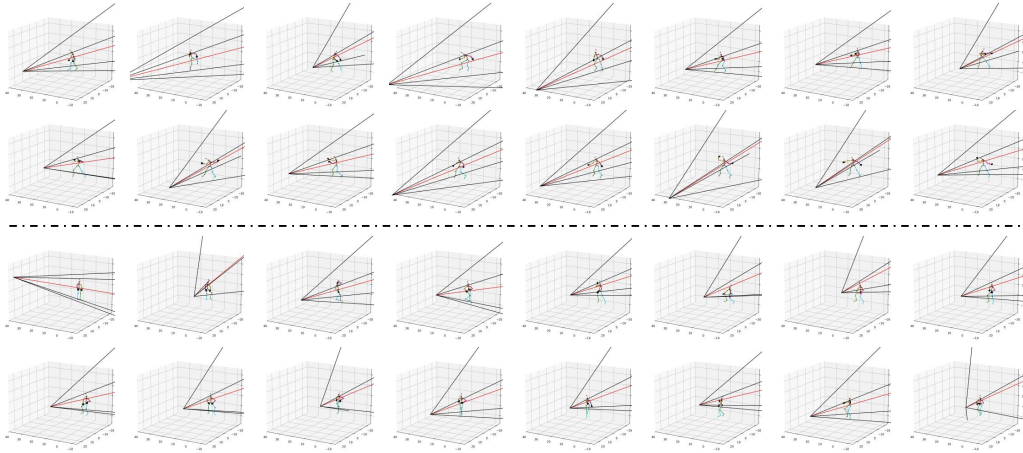


Figure 2: Visualization results for camera movement. Best view in color and zooming in.

42 References

- 43 [1] Zixuan Wang, Jia Jia, Shikun Sun, Haozhe Wu, Rong Han, Zhenyu Li, Di Tang, Jiaqing Zhou, and Jiebo Luo.
 44 Dancecamera3d: 3d camera movement synthesis with music and dance. In *Proceedings of the IEEE/CVF*
 45 *Conference on Computer Vision and Pattern Recognition*, pages 7892–7901, 2024.
- 46 [2] Zixuan Wang, Jiayi Li, Xiaoyu Qin, Shikun Sun, Songtao Zhou, Jia Jia, and Jiebo Luo. Dancecamanimator:
 47 Keyframe-based controllable 3d dance camera synthesis. In *ACM Multimedia 2024*.