

Supplementary Materials: StableMoFusion: Towards Robust and Efficient Diffusion-based Motion Generation Framework

Anonymous Authors

This supplementary material provides detailed definitions of footskate cleanup loss (Appendix A), more qualitative results (Appendix B), and additional experiments on the diffusion samplers (Appendix C).

A FOOTSKATE CLEANUP LOSS

Here we provide detailed definitions of footskate cleanup loss. The complete loss function is as Equation 1. Pose loss L_{pose} minimize the mean squared error(MSE) of motion pose to keep semantic invariant. We use Euler angles to represent motion and convert keypoints to Euler angles by HybrIK algorithm [4]. Trajectory loss $L_{trajectory}$ has the same function as L_{pose} by constraining velocity of root bone. Foot contact loss L_{foot} use MSE to fix foot. \hat{P} is keypoints of footskate cleaned motion. In the early stage of the algorithm, we use V_{23} to calculate target anchored points, denoted as p . VGRF loss L_{vGRFs} keeps valid foot pose by minimizing the mean squared logarithmic error.

$$L = \omega_q L_{pose} + \omega_f L_{foot} + \omega_t L_{trajectory} + \omega_v L_{vGRFs} \quad (1)$$

$$L_{pose}(P, \hat{P}) = \|HybrIK(P) - HybrIK(\hat{P})\|_2^2 \quad (2)$$

$$L_{foot}(P, \hat{P}, V_{23}, P_{S_{23}}) = \sum_{j_{23}}^{J_{skating}} \sum_{f_{23}}^{F_{skating}} (\hat{P}_j - p) \quad (3)$$

$$L_{trajectory}(P, \hat{P}) = \|(P_0^{1:H} - P_0^{0:H-1}) - (\hat{P}_0^{1:H} - \hat{P}_0^{0:H-1})\|_2^2 \quad (4)$$

$$L_{vGRFs}(P, \hat{P}, V_{22}^\theta) = \|\log(1 + V_{22}^\theta(P)) - \log(1 + V_{22}^\theta(\hat{P}))\|_2^2 \quad (5)$$

B QUALITATIVE RESULTS

B.1 Videos

To more visually demonstrate the effect of our method in generating motions, We have provided supplemental videos in the **Videos.zip**, which contains folders: *T2M_Comparison_Demos* and *Footskate_Cleanup_Demos*. We recommend the supplemental video to see these motion results.

Folder **T2M_Comparison_Demos** contains six comparison videos showcasing our method alongside other text-to-motion approaches. Each video presents the motions generated by our StableMoFusion, MDM [11], and MotionDiffuse [12] models, conditioned on the same text prompts. These generated motions are visualized with mesh and compared against each other, with the ground truth (GT) motions serving as a reference point, as depicted in Figure 1.

Folder **Footskate_Cleanup_Demos** shows four visual comparisons of the motions generated by our StableMoFusion without footskate cleanup (labeled *footskate*), after footskate cleanup using Underpressure [8] (labeled *underpressure*), and after footskate cleanup using our method (labeled *ours*), as shown in Figure 2. The videos visualize how well our method removes foot skating resulting from diffusion-generated motions. However, applying Underpressure directly to our framework and SMPL skeleton leads to

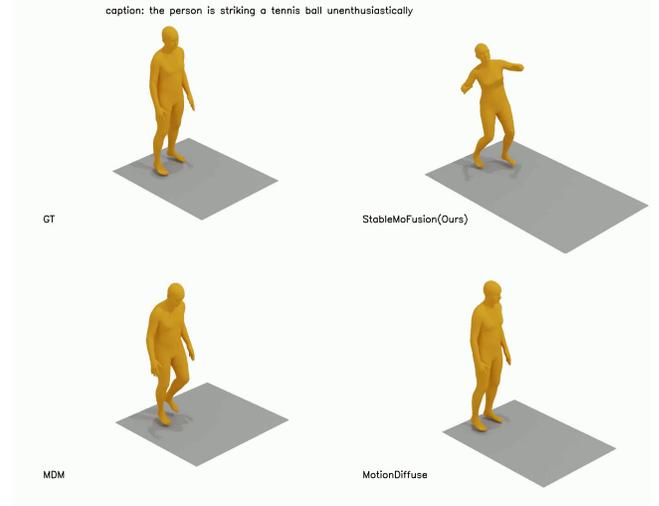


Figure 1: Video frame display of folder *T2M_Comparison_Demos*.

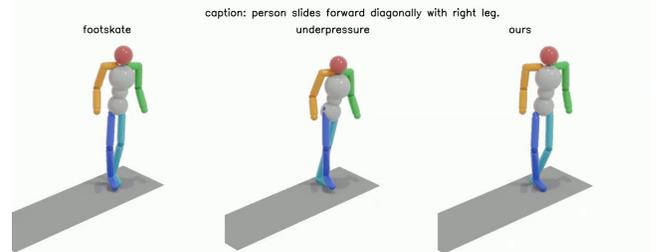


Figure 2: Video frame display of folder *Footskate_Cleanup_Demos*.

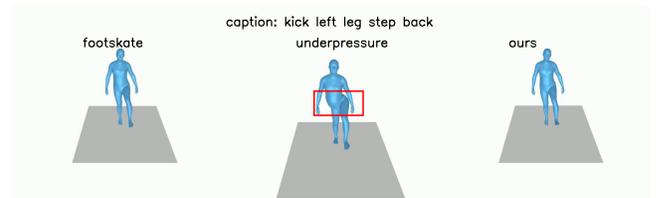
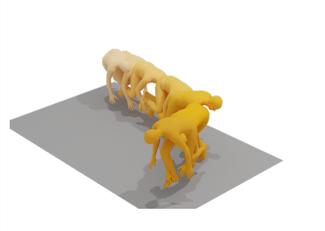


Figure 3: Video frame display of *kick_mesh_demo.mp4*

noticeable jitter and even motion distortion, as depicted in Figure 3. We must do some complex post-processing for underpressure results to retarget its motion to SMPL skeleton, while our method processes the original motion directly instead of retargeting.

Table 1: Comparisons with the state-of-the-art methods on text-conditional motion synthesis task. All provided methods are trained on the HumanML3D [1] dataset and all samples are generated with the same text prompts and motion length.

Text Prompts	MDM [11]	MotionDiffuse [12]	StabelMoFusion (ours)
A person crawls on the ground from east to west then goes back			
A person runs back and forth			
A person stands up from laying, walks in a circle, and lays down again			

B.2 Sequence Figures

To facilitate visualization and explication within the text, we adopt a method akin to the previous approach, wherein the entire motion sequence is rendered into a composite image by stacking all frames, as depicted in Table 1 and Table 2.

Table 1 shows comparisons between our method, MDM [11], and MotionDiffuse [12]. We highlight that StabelMoFusion achieves a balance between text-motion consistency and motion quality. For example, when prompted with *A person stands up from laying, walks in a clockwise circle, and lays down again*, our resultant motion encapsulates a full circular movement and concludes with the reclining action. For the prompt *A person runs back and forth*, our generated motion portrays a complete back-and-forth journey.

Figure 4 provides more samplers generated from various text prompts by our StabelMoFusion framework. Our framework is able to generate high-quality motions that reflect the detailed description.

Table 2 provides more comparisons of the motions generated by our StabelMoFusion before and after footskate cleanup. The foot-slip is evident from the red box in the sequence figures of the multi-frame stacks, and is appreciably removed.

Table 2: More examples to illustrate the effect of footskate cleanup in our diffusion framework.

Motion Type	StabelMoFusion (w/o footskate cleanup)	StabelMoFusion (w/ footskate cleanup)
wave		
kick		
slide		
stand		

233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290

291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348



(a) he walks forward and then turns around fast and walks back



(b) the person is striking a tennis ball unenthusiastically



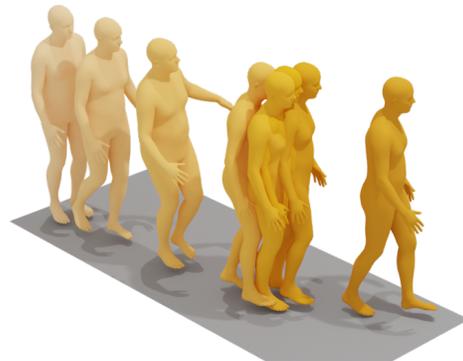
(c) the person is dancing the waltz.



(d) a person walks up stairs



(e) a person is on his knees and then gets up by pushing himself up with his right hand



(f) a person walks in a left diagonal then stops with hands slightly raised.



(g) a person is doing jumping jacks



(h) a person jump ropes

Figure 4: More samples of our StableMoFusion for text-to-motion synthesis.

C ADDITIONAL EXPERIMENTS ON SAMPLERS

In this section, we will show the effect of incorporating five experienced discrete-time samplers into the motion diffusion framework. To select the most suitable sampler for efficient inference, we initially used the pre-trained models in MotionDiffuse [12] to evaluate and analyze these samplers, namely DDPM [2], DDIM [9], DPMSolver [6, 7], PNDM [5], and DEIS [13], which preceded our development of the model architecture.

These samplers can be categorized into two groups based on whether adding additional noise in each reverse step: Ordinary Differential Equations (ODE) [3] and Stochastic Differential Equation (SDE) [10] samplers.

C.1 Experimental setup.

We use the trained models of MotionDiffuse [12] to evaluate the inference effects of the five samplers on both For a fair comparison, All inference experiments use the same $batchsize = 1024$ and set $seed = 0$. Use DDPM [2] with $T = 1,000$ as control group (Ctrl).

C.2 ODE Samplers

ODE samplers accelerate DDPM by solving ODEs on manifold without additional noise. These approaches construct a deterministic sampling trajectory that traverses from noise space to the target data distribution. ODE samplers have been shown to produce less discretization error than the SDE samplers, however, they will eventually reach the upper limit of their performance due to their deterministic sampling trajectories from noise to the data distribution, which leads to a certain cumulative error as shown in Table 3 and Table 4.

Although ODE samplers can significantly decrease the number of sampling steps from 1000 by a certain amount, they are only capable of generating motions with FID around 2.0 on KIT-ML dataset and FID around 1.25 on HumanML3D dataset.

Table 3: Comparison of samplers on MotionDiffuse using the HumanML3D test set. The minimum sampling step is selected if its FID and R Precision (top3) are within 5% of the optimal result.

	Sampler	Minimum Sampling Steps	FID ↓	R Precision (top3) ↑
ODE	DDIM	500	1.253	0.764
	PNDM	200	1.297	0.763
	DEIS	20	1.281	0.761
	DPMSolver++	20	1.235	0.764
	SDE	DDPM (Ctrl)	1000	0.709
	DDPM	500	0.731	0.787
	SDE DPMSolver++	20	0.680	0.774
	SDE DPMSolver++ Karras	10	0.521	0.781

C.3 SDE Samplers

Since SDE samplers introduce additional noise during the iterative inference process, the stochasticity of their sampling trajectories helps to reduce the cumulative error, which is crucial for diversity and realism in diffusion-based motion generation, as shown in Table 3 and Table 4.

Table 4: Comparison of samplers on MotionDiffuse using the KIT-ML test set. The minimum sampling step is selected if its FID and R Precision (top3) are within 5% of the optimal result.

	Sampler	Minimum Sampling Steps	FID ↓	R Precision (top3) ↑
ODE	DDIM	200	2.012	0.711
	PNDM	50	2.069	0.736
	DEIS	2	2.006	0.720
	DPMSolver++	4	1.962	0.734
SDE	DDPM (Ctrl)	1000	1.673	0.740
	DDPM	500	1.712	0.743
	SDE DPMSolver++	5	1.590	0.743
	SDE DPMSolver++ Karras	5	0.886	0.727

SDE samplers are capable of generating higher quality motions than the ODE sampler.

C.4 Karras Sigma

The Karras Sigma [3] $\sigma(t) = \sqrt{t}$ corresponds to constant-velocity thermal diffusion, which enables fast and good sampling in image synthesis. Karras sigma also improves the quality of motion diffusion generation, as shown in Table 3 and Table 4. By using Karras Sigma, the quality of the DPM-Solver++ sampler for motion generation is improved from 1.6 FID to 0.886 on KIT-ML dataset and .

Using Karras Sigma in sampler can improve the quality of motion diffusion generation.

C.5 Ablation on StableMoFusion

We also re-validated the efficiency of the selected one, SDE variant of second-order DPM-Solver++ with Karras Sigmas (SDE DPM-Solver++ Karras), compared to other samplers in our final framework, the results of which are shown in Table 5.

Table 5: Comparison of samplers on StableMoFusion using the KIT test set. The minimum sampling step is selected if its FID and R Precision (top3) are within 5% of the optimal result.

Sampler	Minimum Sampling Steps	FID ↓	R Precision (top3) ↑
DDIM	200	0.243	0.794
DDPM	500	0.253	0.793
SDE DPMSolver++	5	0.246	0.796
SDE DPMSolver++ Karras	10	0.209	0.780

Note that, the minimum sampling steps vary when utilizing the same sampler across different datasets or different frameworks.

REFERENCES

- [1] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. 2022. Generating Diverse and Natural 3D Human Motions From Text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5152–5161.
- [2] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising Diffusion Probabilistic Models. *Advances in neural information processing systems* 33 (2020), 6840–6851.
- [3] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. 2022. Elucidating the Design Space of Diffusion-Based Generative Models. *Advances in Neural Information Processing Systems* 35 (2022), 26565–26577.

465	[4] Jiefeng Li, Siyuan Bian, Chao Xu, Zhicun Chen, Lixin Yang, and Cewu Lu. 2023. HybriK-X: Hybrid Analytical-Neural Inverse Kinematics for Whole-body Mesh Recovery. <i>arXiv preprint arXiv:2304.05690</i> (2023).	523
466		524
467	[5] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. 2021. Pseudo Numerical Methods for Diffusion Models on Manifolds. In <i>International Conference on Learning Representations</i> .	525
468		526
469	[6] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. 2022. DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps. <i>Advances in Neural Information Processing Systems</i> 35 (2022), 5775–5787.	527
470		528
471	[7] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. 2022. DPM-Solver++: Fast Solver for Guided Sampling of Diffusion Probabilistic Models. <i>arXiv preprint arXiv:2211.01095</i> (2022).	529
472		530
473	[8] Lucas Mourot, Ludovic Hoyet, François Le Clerc, and Pierre Hellier. 2022. UnderPressure: Deep Learning for Foot Contact Detection, Ground Reaction Force Estimation and Footskate Cleanup. <i>Computer Graphics Forum</i> 41, 8 (Dec. 2022), 195–206. https://doi.org/10.1111/cgf.14635	531
474		532
475	[9] Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020. Denoising Diffusion Implicit Models. In <i>International Conference on Learning Representations</i> .	533
476		534
477	[10] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2020. Score-Based Generative Modeling through Stochastic Differential Equations. <i>arXiv preprint arXiv:2011.13456</i> (2020).	535
478		536
479	[11] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. 2022. Human Motion Diffusion Model. <i>arXiv preprint arXiv:2209.14916</i> (2022).	537
480		538
481	[12] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. 2022. MotionDiffuse: Text-Driven Human Motion Generation with Diffusion Model. <i>arXiv preprint arXiv:2208.15001</i> (2022).	539
482		540
483	[13] Qinsheng Zhang and Yongxin Chen. 2022. Fast Sampling of Diffusion Models with Exponential Integrator. In <i>NeurIPS 2022 Workshop on Score-Based Methods</i> .	541
484		542
485		543
486		544
487		545
488		546
489		547
490		548
491		549
492		550
493		551
494		552
495		553
496		554
497		555
498		556
499		557
500		558
501		559
502		560
503		561
504		562
505		563
506		564
507		565
508		566
509		567
510		568
511		569
512		570
513		571
514		572
515		573
516		574
517		575
518		576
519		577
520		578
521		579
522		580