

## APPENDIX FOR MOTIONGPT3

This appendix provides qualitative comparison results (Appendix B), discussion of continuous/discrete motion representation (Appendix A), additional quantitative results and ablation (Appendix C) on motion branch size and connection type, motion supervision scheme, training stages. We also provide more implementation details in Appendix D. Please note our examination of metrics report on TMR evaluator (Appendix C.2), analysis on bimodal architecture (Appendix C.3), and ablation on our training scheme (Appendix C.6).

**Website & Video** A supplementary website provides visualizations of quantitative results, motion data, and demonstration videos. A standalone video is also available on the website and at `supp/website/static/videos/MotionGPT3/_Video.mp4`, showcasing (i) text-to-motion comparisons, (ii) motion-captioning comparisons, and (iii) additional results on motion generation and captioning.

**Code** Example code files are available in the supplementary materials, which cover the training and evaluation processes of our MotionGPT3 models, along with several example results.

### A DISCRETE TOKEN VS CONTINUOUS TOKEN

**Reconstruction Task** We compare continuous latents from MLD VAE (Xin et al., 2023) with discrete latents from VQ-VAE (Jiang et al., 2023) for motion reconstruction under identical settings. As shown in Appendix A, VQ-VAE yields higher errors on MPJPE, PAMPJPE, ACCL, and APE/AVE for root, trajectory, pose, joints, indicating reduced fidelity and temporal smoothness relative to the continuous VAE. This gap is expected: quantization maps a continuous motion manifold to a finite codebook and turns motion modeling into token classification, introducing unavoidable approximation error and information loss. In practice, many distinct frames collapse to the same code (one-to-many mapping), yielding ambiguous reconstructions and frame-wise noise that harms smoothness.

Table 4: Reconstruction performance of a continuous VAE (Xin et al., 2023) versus a discrete VQ-VAE (Jiang et al., 2023). The VQ-VAE shows consistently higher errors, consistent with information loss introduced by quantized encoding and decoding. Appendix D.3 presents the metric definitions.

Method	MPJPE	PAMPJPE	ACCL	APE				AVE			
				root	traj	pose	joints	root	traj	pose	joints
VAE	43.906	31.356	5.93	0.0581	0.0504	0.0277	0.0619	0.0179	0.0177	0.0012	0.0185
VQ	46.828	33.668	7.629	0.0829	0.0804	0.0316	0.0930	0.0240	0.0239	0.0015	0.0253

**On Discrete VQ Latents** Recent work Guo et al. (2024) addresses the limitations of codebook capacity using hierarchical Residual Vector Quantization (RVQ) with separate predictors for base and residual tokens. Wang et al. (2025) combines quantized and continuous latents via post-training quantization. Cho et al. (2025) introduces a diffusion-based decoder that progressively maps discrete tokens back to continuous raw motions, improving fidelity and smoothness, and uses the symmetric Jerk Percentage Error (sJPE) to detect under-reconstruction and frame noise. Despite these advances, discrete pipelines remain prone to expressiveness bottlenecks and token-induced jitter. We further evaluate VQ and VAE latents within dual-stream architecture under single-task training. The results (Tab. 5) consistently favor continuous representations for both generation and understanding. While discrete codes facilitate token-based modeling, continuous representations better capture fine-grained dynamics and motion continuity, achieving higher alignment and synthesis quality with fewer epochs.

Table 5: Discrete (VQ) vs. continuous (VAE) motion representations under single-task training. We report both T2M on R@1, FID, MMDist, DIV and M2T on R@3, BLEU@1/4, ROUGE. With fewer training epochs, VAE-variants achieve stronger alignment and better quality. VQ requires extended training of 399 epochs while still remains behind on most alignment and language scores.

Representation	Text-to-Motion					Motion-to-Text				
	epoch	R@1↑	FID ↓	MM Dist ↓	DIV→	epoch	R@3↑	Bleu@1↑	Bleu@4↑	Rouge↑
VQ	199	0.258	0.542	5.364	9.274	99	0.765	47.043	7.234	39.244
	399	0.300	0.454	4.937	9.626	199	0.752	41.579	6.304	35.746
VAE	199	0.525	0.191	2.667	10.095	99	0.859	50.707	8.383	38.225

054  
055  
056  
057  
058  
059  
060  
061  
062  
063  
064  
065  
066  
067  
068  
069  
070  
071  
072  
073  
074  
075  
076  
077  
078  
079  
080  
081  
082  
083  
084  
085  
086  
087  
088  
089  
090  
091  
092  
093  
094  
095  
096  
097  
098  
099  
100  
101  
102  
103  
104  
105  
106  
107

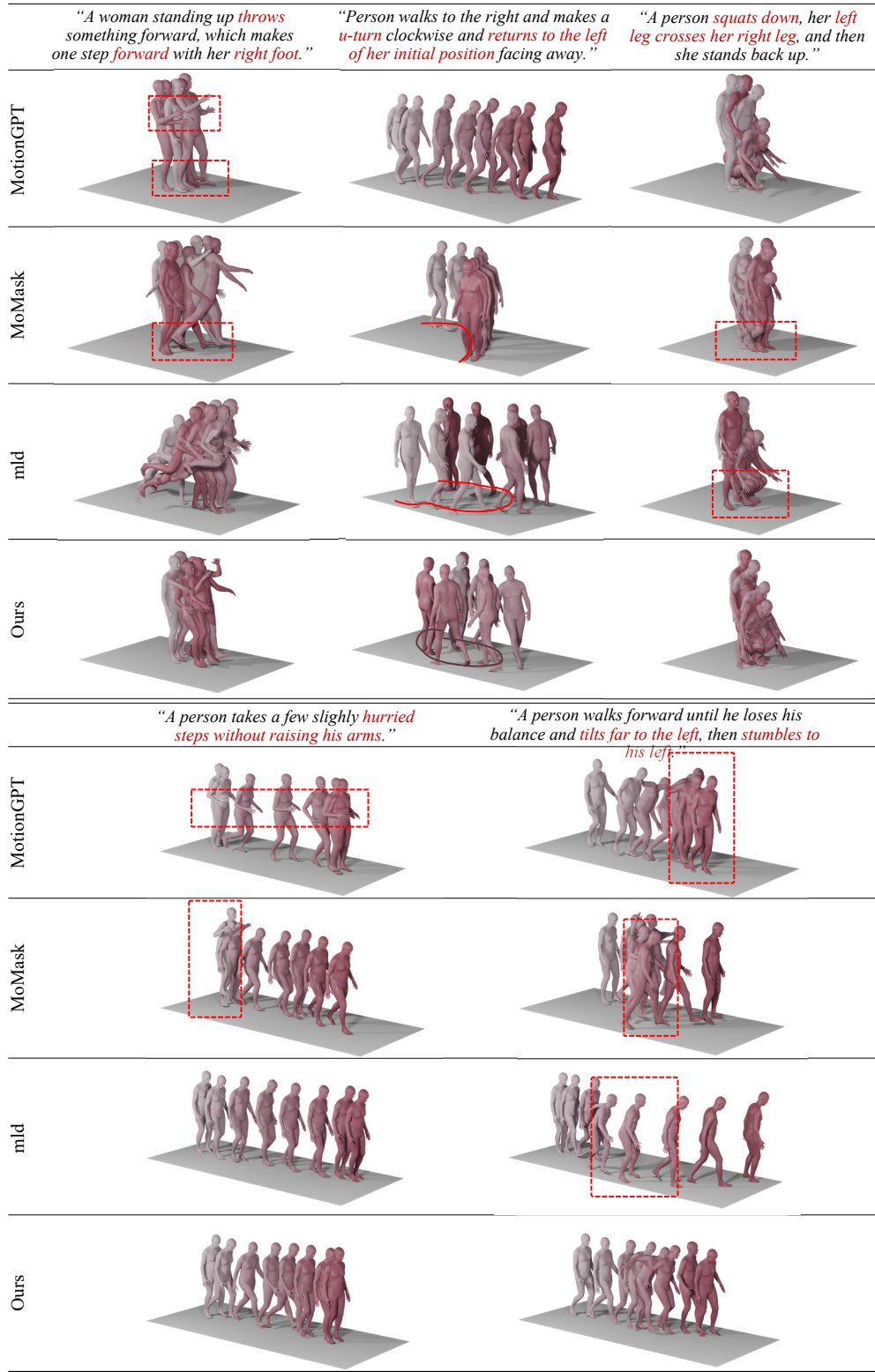


Figure 5: Qualitative comparison of text-driven motion generation on HumanML3D (Guo et al., 2022a). Baselines ( (Jiang et al., 2023; Guo et al., 2024; Xin et al., 2023)) are run with their official released checkpoints. **Red annotations** (text, boxes, curves) highlight prompt–motion mismatches. Our bimodal motion-language framework yields motions that with closer correspondence to the textual prompt and smoother temporal coherence.



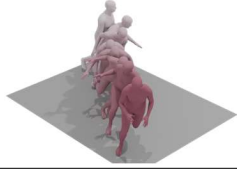
Input Motions			
Real	"a person with his hands on his hips does two brief squats."	"figure walks forward 3 steps with arms straight in front of him."	"a person who seems to evade something from their left side and run at a insane pace."
MotionGPT	"a person <i>sits down in a chair</i> and stands up."	"person is walking <i>down</i> slowly."	"a person runs forward, then turns and runs to the right."
TM2T	"a person is stand with their leg apart and arm bent at the elbow."	"a person walk forward with both arm raise <i>above their head</i> ."	"a person run forward and <i>jump over</i> something."
Ours	"a standing to step slightly blegged with right leg in a arched bend."	"a man walks slowly in a zombie walk with his arms stretched out in front of him."	"running fast against a small obstacle."

Figure 6: Example results on motion caption. The misalignment is highlighted with red.

**Generation Task (T2M)** With the same 200 training epochs, the VAE representation delivers substantially better quality than VQ, with 0.525 vs. 0.258 on R@1 and 2.667 vs. 5.364 on MMDist, while maintaining competitive diversity(DIV). Extending VQ to 399 epochs reduces FID to 0.454, however, it still lags in alignment, with R@1 0.300 and MMDist 4.937, indicating a lower quality ceiling for discrete codes.

**Understanding Task (M2T)** At matched training over 99 epochs, VAE attains stronger retrieval and language scores than VQ, with 0.859 vs. 0.765 on R@3 and 50.707 vs. 47.043 on BLEU@1, while ROUGE is comparable. Prolonged VQ training to 199 epochs unexpectedly reduces performance across all language scores with R@3 dropping by 0.13 and BLEU@1 by 5.464, suggesting optimization instability and poorer generalization under the discrete setting.

## B QUALITATIVE RESULTS

We provide qualitative comparisons for text-to-motion (T2M) Fig. 5, motion-to-text (M2T) Fig. 6, and a multi-task gallery Fig. 7. To ensure fair visualization, we use a fixed list of prompts from the HumanML3D test split; all clips are rendered at 20 FPS with identical camera and skeleton settings. Baselines are run with the their official checkpoints. Overall, MotionGPT3 exhibits stronger smoother transitions and text-motion alignment. Discrete-code variants tend to show token-induced frame noise and temporal drift, whereas single-stream models can produce semantically inconsistent motions on complex prompts. Additional examples and animations are provided in the supplementary video.

Table 6: Comprehensive comparison of text-to-motion generation on HumanML3D (Guo et al., 2022a). We report generation-only models (Gen. only) here, and visualize unified dual-task models (Gen. & Und.) in Fig. 8. Real denotes ground-truth statistics; arrows ( $\rightarrow$ ) indicate that values closer to Real are desirable.  $\dagger$  marks our single-task model trained for 200 epochs, and MotionGPT3 is the unified three-stage model. Best and second-best results are **bold** and underlined.

Methods	R@1	R@2	R@3	FID↓	MMDist↓	DIV→	MModality↑
Real	0.511 $\pm$ 0.003	0.703 $\pm$ 0.003	0.797 $\pm$ 0.002	0.002 $\pm$ 0	2.974 $\pm$ 0.008	9.503 $\pm$ 0.065	
T2M (Guo et al., 2022b)	0.457 $\pm$ 0.002	0.639 $\pm$ 0.003	0.74 $\pm$ 0.003	1.067 $\pm$ 0.002	3.34 $\pm$ 0.008	9.188 $\pm$ 0.002	2.09 $\pm$ 0.083
MLD (Xin et al., 2023)	0.481 $\pm$ 0.003	0.673 $\pm$ 0.003	0.772 $\pm$ 0.002	0.473 $\pm$ 0.013	3.169 $\pm$ 0.01	9.724 $\pm$ 0.082	<u>2.413</u> $\pm$ 0.079
MotionDiffuse (Zhang et al., 2024)	0.491 $\pm$ 0.001	0.681 $\pm$ 0.001	0.782 $\pm$ 0.001	0.63 $\pm$ 0.001	3.113 $\pm$ 0.001	9.410 $\pm$ 0.049	1.553 $\pm$ 0.042
T2M-GPT (Zhang et al., 2023a)	0.491 $\pm$ 0.003	0.68 $\pm$ 0.003	0.775 $\pm$ 0.002	0.116 $\pm$ 0.004	3.118 $\pm$ 0.011	9.761 $\pm$ 0.081	1.856 $\pm$ 0.011
ReMoDiffuse (Zhang et al., 2023b)	0.51 $\pm$ 0.005	0.698 $\pm$ 0.006	0.795 $\pm$ 0.004	0.103 $\pm$ 0.004	2.974 $\pm$ 0.016	9.018 $\pm$ 0.075	1.795 $\pm$ 0.043
DiverseMotion (Lou et al., 2023)	0.515 $\pm$ 0.003	0.706 $\pm$ 0.002	0.809 $\pm$ 0.002	0.072 $\pm$ 0.004	2.941 $\pm$ 0.007	9.683 $\pm$ 0.102	1.869 $\pm$ 0.089
MoMask (Guo et al., 2024)	0.521 $\pm$ 0.002	0.713 $\pm$ 0.002	0.807 $\pm$ 0.002	<u>0.045</u> $\pm$ 0.002	2.958 $\pm$ 0.008	9.620 $\pm$ 0.064	1.241 $\pm$ 0.04
MotionAnything (Zhang et al., 2025)	0.546 $\pm$ 0.003	0.735 $\pm$ 0.002	0.829 $\pm$ 0.002	<b>0.028</b> $\pm$ 0.005	2.859 $\pm$ 0.01	<b>9.521</b> $\pm$ 0.083	<b>2.705</b> $\pm$ 0.06
<b>MotionGPT3<math>\dagger</math></b>	0.533 $\pm$ 0.002	0.731 $\pm$ 0.002	0.826 $\pm$ 0.003	0.239 $\pm$ 0.008	<u>2.797</u> $\pm$ 0.007	9.688 $\pm$ 0.107	1.560 $\pm$ 0.052
<b>MotionGPT3</b>	<b>0.553</b> $\pm$ 0.003	<b>0.747</b> $\pm$ 0.002	<b>0.837</b> $\pm$ 0.003	0.208 $\pm$ 0.006	<b>2.725</b> $\pm$ 0.008	9.700 $\pm$ 0.096	1.018 $\pm$ 0.038

$\dagger$  We train our model on single T2M task for 200 epochs.

162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215

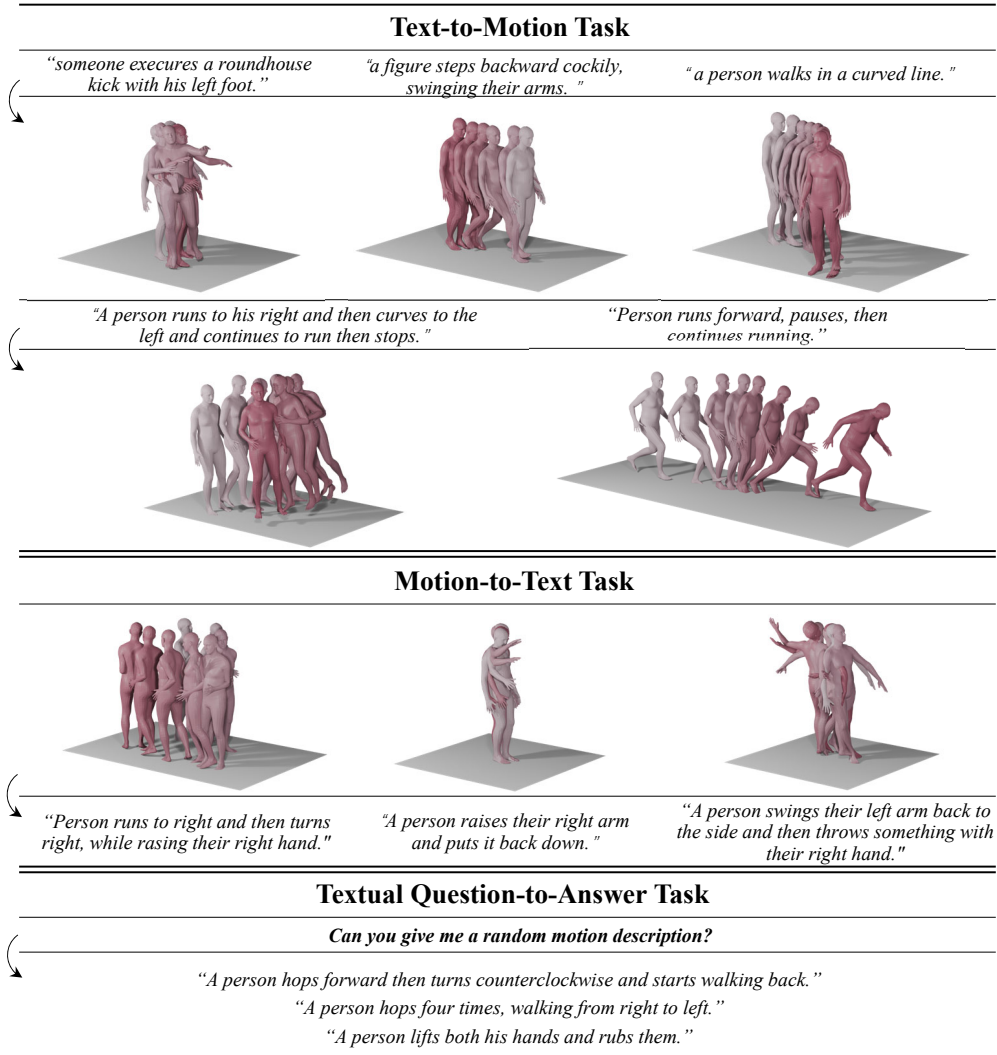


Figure 7: Gallery for the results of MotionGPT3. Top: text-to-motion generation. Middle: motion-to-text captioning. Bottom: textual question answering about motion. Examples are produced by our unified model trained with instruction-based objectives (three-stage scheme). Animated visualizations are provided in the supplementary video.

## C ADDITIONAL EXPERIMENTS

This section provides supplementary evaluations that complement the main results. First, we report a comprehensive comparison including additional generation-only (Gen. only) and unified dual-task baselines (Gen. & Und.) (Appendix C.1) and further assess text-to-motion with the TMR retrieval evaluator (Appendix C.2). We then analyze design choices of the Cross-Modal Attention (CMA) (Appendix C.3), examine scaling effects of both language and motion branches (Appendix C.4), and analyze the diffusion-based supervision used for motion generation (Appendix C.5). Finally, we evaluate the effectiveness of the three-stage training strategy (Appendix C.6).

### C.1 QUANTITATIVE RESULTS

Tab. 6 reports the full text-to-motion results on HumanML3D, grouped by training regime (generation-only vs. unified dual-task). Notably, evaluated on the HumanML3D dataset by T2M evaluator (Guo et al., 2022b), recent models consistently achieve very high scores, and several recent approaches (Guo et al., 2024; Zhang et al., 2025; Li et al., 2024; Wu et al., 2024; 2025), including MotionGPT3, achieve scores even above those of the ground-truth data (*Real*).

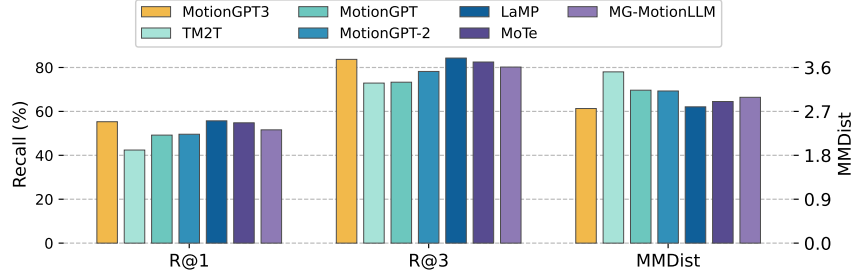


Figure 8: Comparison on text-to-motion, with recent approaches trained with unified tasks (Gen. & Und.). Our model performs better than recent unified models: MotionGPT (Jiang et al., 2023), TM2T (Guo et al., 2022c), MotionGPT-2 (Wang et al., 2024), MG-MotionLLM (Wu et al., 2025), MoTe (Wu et al., 2024), and comparable with LaMP (Li et al., 2024).

Generally speaking, high evaluation scores indicate that the generated motions can correspond well with the text, and approximate high-fidelity motions. **However**, considering that the GT embeddings effectively represent an upper bound for matching scores, the practical significance of differences among methods achieving near or above GT performance might be limited. This reflects a limitation of the T2M evaluator, where the metrics are computed in a learned embedding space which relies on contrastive learning on HumanML3D. Methods that overfit to that space can saturate the proxy and even surpass the ground-truth reference, without a proportionate improvement in motion fidelity. Hence, differences among methods near or above the Real line are hard to interpret.

## C.2 EVALUATION WITH TMR

For a more nuanced assessment, we further evaluate with the TMR retrieval framework (Petrovich et al., 2023), under four gallery protocols: **All**, entire test set, **All with threshold**, gallery items whose textual similarity to the query exceeds a fixed threshold, **Dissimilar**, a 100-pair subset with mutually distant texts, and **Small batches**, mini-batch of size 32 to mimic Guo et al. (2022b) setting. We report text-motion retrieval with R@1/2/3/5/10 and MedR, results are summarized in Tab. 7.

Not all methods in Tab. 6 release TMR results or checkpoints, so the comparison is limited to publicly available models. Within this scope, although inferior to Li et al. (2024) on T2M metrics (Fig. 8), our method achieves significantly stronger retrieval under the TMR evaluator and improves substantially over the T2M baseline Guo et al. (2022b) across protocols. These findings suggest that our model achieves more robust cross-modal alignment rather than overfitting to a specific evaluator.

Table 7: Retrieval on HumanML3D with the TMR evaluator (Petrovich et al., 2023). We report R@1/2/3/5/10 and MedR for text-motion retrieval under the four official protocols: (a) All, (b) All with threshold, (c) Dissimilar subset, and (d) Small batches (see Appendix C.2 for definitions). Results for TEMOS (Petrovich et al., 2022), T2M (Guo et al., 2022b), and TMR (Petrovich et al., 2023) are taken from the TMR paper. LaMP (Li et al., 2024) is reported only for (d). MotionGPT and MotionGPT3 are evaluated with the released checkpoints using the official TMR code. MotionGPT3 attains strong performance across protocols.

Protocol	Methods	Text-motion retrieval						Motion-text retrieval					
		R@1↑	R@2↑	R@3↑	R@5↑	R@10↑	MedR↓	R@1↑	R@2↑	R@3↑	R@5↑	R@10↑	MedR↓
All	TEMOS	2.12	4.09	5.87	8.26	13.52	173.00	3.86	4.54	6.94	9.38	14.00	183.25
	T2M	1.80	3.42	4.79	7.12	12.47	81.00	2.92	3.74	6.00	8.36	12.95	81.50
	TMR	5.68	10.59	14.04	20.34	30.94	28.00	9.95	12.44	17.95	23.56	32.69	28.50
	MotionGPT	7.16	12.50	15.85	21.53	30.20	38.00	11.31	13.91	19.39	24.13	31.80	36.25
	MotionGPT3	<b>9.60</b>	<b>17.36</b>	<b>22.45</b>	<b>30.43</b>	<b>41.06</b>	<b>17.00</b>	<b>14.90</b>	<b>18.20</b>	<b>24.43</b>	<b>31.32</b>	<b>40.72</b>	<b>17.50</b>
All with threshold	TEMOS	5.21	8.22	11.14	15.09	22.12	79.00	5.48	6.19	9.00	12.01	17.10	129.00
	T2M	5.30	7.83	10.75	14.59	22.51	54.00	4.95	5.68	8.93	11.64	16.94	69.50
	TMR	11.60	15.39	20.50	27.72	38.52	19.00	13.20	15.73	22.03	27.65	37.63	21.50
	MotionGPT	14.32	21.01	25.94	33.39	43.84	15.00	14.42	16.83	22.70	27.69	35.06	30.50
	MotionGPT3	<b>20.73</b>	<b>27.03</b>	<b>34.03</b>	<b>42.66</b>	<b>52.97</b>	<b>9.00</b>	<b>19.34</b>	<b>22.40</b>	<b>29.40</b>	<b>36.91</b>	<b>46.30</b>	<b>13.00</b>
Dissimilar subset	TEMOS	33.00	42.00	49.00	57.00	66.00	4.00	35.00	44.00	50.00	56.00	70.00	3.50
	T2M	34.00	48.00	57.00	72.00	84.00	3.00	34.00	47.00	59.00	72.00	83.00	3.00
	TMR	47.00	61.00	71.00	80.00	86.00	2.00	48.00	63.00	69.00	80.00	84.00	2.00
	MotionGPT	51.00	64.00	71.00	74.00	80.00	1.00	53.00	62.00	68.00	76.00	81.00	1.00
	MotionGPT3	<b>68.00</b>	<b>77.00</b>	<b>85.00</b>	<b>92.00</b>	<b>95.00</b>	<b>1.00</b>	<b>63.00</b>	<b>73.00</b>	<b>83.00</b>	<b>89.00</b>	<b>93.00</b>	<b>1.00</b>
Small batches	TEMOS	40.49	53.52	61.14	70.96	84.15	2.33	39.96	53.49	61.79	72.40	85.89	2.33
	T2M	52.48	71.05	80.65	89.66	96.58	1.39	52.00	71.21	81.11	89.87	96.78	1.38
	TMR	67.16	81.32	86.81	91.43	95.36	1.04	67.97	81.20	86.35	91.70	95.27	1.03
	LaMP	67.18	81.90	87.04	92.00	95.73	-	68.02	82.10	87.50	92.20	96.90	-
	MotionGPT	58.07	69.91	74.34	79.17	86.36	1.18	58.71	69.64	74.36	79.45	86.02	1.16
	MotionGPT3	<b>74.25</b>	<b>86.70</b>	<b>91.29</b>	<b>94.82</b>	<b>97.35</b>	<b>1.00</b>	<b>74.00</b>	<b>86.86</b>	<b>91.04</b>	<b>94.62</b>	<b>97.35</b>	<b>1.00</b>

### C.3 MOTION BRANCH WITH CROSS-MODAL CONNECTION

Our hybrid model allows asymmetric capacities for the text and motion branches and supports different patterns of cross-modal information exchange. In this section we focus on *where* to place cross-modal attention (CMA) in the backbone for the text-to-motion task, keeping all other factors fixed (Tab. 9). Ablation on branch capacity is deferred to Appendix C.4.

We explore several CMA schedules that differ only in the layers where cross-modal connections are enabled (Tab. 9). Across paired settings with the same spacing pattern, shifting the CMA blocks to later layers typically improves generation quality and distribution similarity to the ground-truth (i.e., lower FID and MMDist, higher R-Precision). This is most evident in  $B_1$  v.s.  $B_2$ : both use uniformly spaced CMA with identical count,  $B_1$  enable CMA from the first to the second-last layer, while  $B_2$  shifts them by one layer to span the second through the last layer. Despite the minor offset,  $B_2$  achieves noticeably better scores. The same tendency appears in  $B_3$ – $B_4$ ,  $C_1$ – $C_3$ , and  $D_1$ – $D_2$ , where the distribution pattern is matched but the CMA positions differ.

We further enable CMA in the last  $L$  layers and sweep  $L$  (Fig. 9). Increasing  $L$  from 2 to 5 generally improves quality, reflected by lower FID scores and MMDist and higher R-Precision. However, the trend is **non-monotonic**: we observe a slight drop at  $L = 6$  in R-Precision, relative to  $L = 5$ , may suggesting that late but not ubiquitous CMA is preferable.

Table 8: Cross-modal attention (CMA) configurations used in ablation. (a) Layer-wise CMA schedules for configurations A–D across the 12-layer backbones. Within each branch, the symbol  $\Leftrightarrow$  marks a cross-modal attention (CMA) operation at that layer, blanks empty indicate intra-modal attention only. (b) Schematic diagrams for configurations A,  $B_1$ ,  $C_1$ .

Model	0	1	2	3	4	5	6	7	8	9	10	11
A	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$
$B_1$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$
$B_2$		$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$		$\Leftrightarrow$		$\Leftrightarrow$		$\Leftrightarrow$
$B_3$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$							
$B_4$							$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$
$C_1$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$								
$C_2$					$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$				
$C_3$									$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$
$D_1$	$\Leftrightarrow$											
$D_2$												$\Leftrightarrow$

Table 9: Quantitative results for several CMA settings on T2M with 200K training iterations, settings visualized in Tab. 8. The text branch is pretrained GPT-2 (124M) and the motion branch has 114M parameters. Increasing the number of CMA layers and placing them later in the network generally improves performance, and A is the best among tested settings. See Appendix C.3 for further analysis.

	R@1 $\uparrow$	R@2 $\uparrow$	R@3 $\uparrow$	FID $\downarrow$	MMDist $\downarrow$	DIV $\rightarrow$	MModality $\uparrow$
Real	0.518	0.713	0.813	-	2.811	9.976	-
A	<b>0.536</b>	<b>0.728</b>	<b>0.819</b>	0.241	<b>2.767</b>	10.379	2.454
$B_1$	0.502	0.707	0.807	0.311	2.895	10.318	2.489
$B_2$	0.508	0.714	0.812	0.288	2.8637	10.261	2.315
$B_3$	0.508	0.712	0.811	0.22	2.879	10.405	2.664
$B_4$	<u>0.514</u>	<u>0.716</u>	<u>0.814</u>	0.243	<u>2.839</u>	10.347	2.534
$C_1$	0.506	0.701	0.801	0.236	2.894	10.386	2.684
$C_2$	0.502	0.702	0.795	0.285	2.948	10.333	2.631
$C_3$	0.503	0.705	0.803	<u>0.171</u>	2.886	10.221	2.819
$D_1$	0.473	0.663	0.767	0.283	3.105	<b>10.176</b>	<b>3.770</b>
$D_2$	0.477	0.672	0.777	<b>0.164</b>	3.092	<u>10.189</u>	<u>3.197</u>



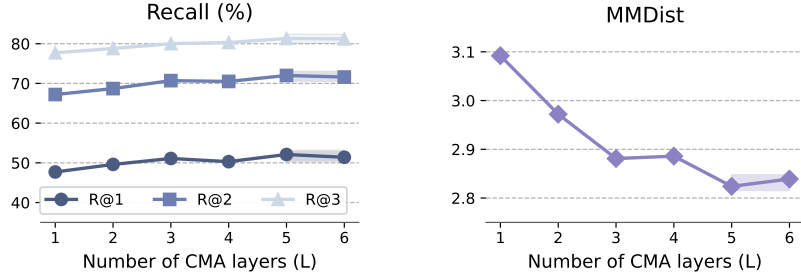


Figure 9: Ablation on the number of cross-modal attention (CMA) layers for T2M on HumanML3D. CMA is enabled in the last  $L$  layers ( $L \in 1, \dots, 6$ ). Performance improves as  $L$  increases up to 5 layers, then shows slight degradation at 6, indicating a **non-monotonically pattern**.

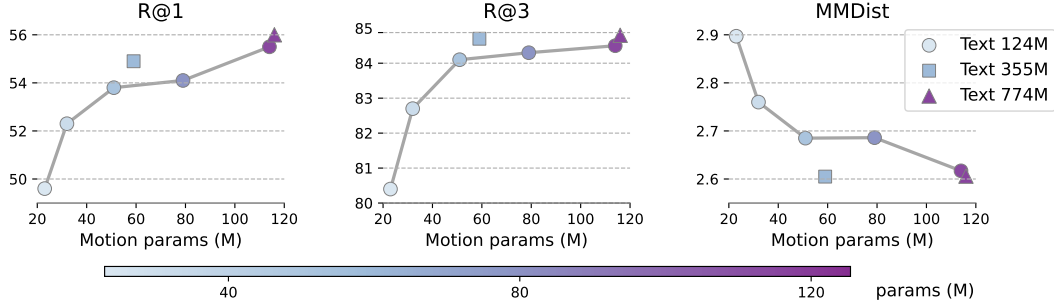


Figure 10: Ablation on branch capacity on motion generation. All models are trained for 200K iterations. A 124M text branch already performs competitively with much larger backbones (355M/774M). Our model can achieve competitive performance with only halved motion parameters ( $\sim 51$ M).

#### C.4 ABLATION ON MODEL SIZE

We examine model size along three axes while keeping all other settings fixed: (i) the overall capacity, achieved by scaling the text and motion branches proportionally, (ii) motion-branch capacity with a fixed text branch, and (iii) language-backbone size with a comparable motion branch.

Tab. 10 compares overall **backbone sizes**. With roughly  $3\times$  parameters, the medium model yields modest gains on R@k and MMDist despite slightly higher FID. This suggests greater capacity helps capture high-level semantics, though realizing its full benefit may require careful optimization.

With the text branch fixed to GPT-2 small (124M), we **scale the motion branch** from 23M to 114M parameters by setting hidden size to 76, 192, 384, 576, 768 (Fig. 10). Increasing motion capacity generally improves text–motion alignment (higher R@k) and reduces MMDist, while diversity remaining stable. In a high level A half-sized motion branch ( $\sim 51$ M) already offers a strong trade-off, delivering competitive overall performance, and, the best FID among the 124M-text configurations.

**Text-branch Size.** To isolate the effect of the language backbone, we replace GPT-2 small (124M) with GPT-2 medium (355M) and GPT-2 large (774M), keeping the motion branch of comparable size (pairs: 355M/59M vs. 124M/51M, and 774M/116M vs. 124M/114M). Larger text branches further improve alignment and lower MMDist, and tend to increase diversity/MModality.

Table 10: Effect of GPT-2 backbone size and CFG guidance scale  $\omega$  on text-to-motion task. All models are trained for 200K iterations. The medium backbone, with more parameters (692M v.s. 238M) consistently outperforms the small model.

Size	Text Params	Total Params	$\omega$	R@1 $\uparrow$	R@2 $\uparrow$	R@3 $\uparrow$	FID $\downarrow$	MMDist $\downarrow$	DIV $\rightarrow$
small	124M	238M	1.0	0.534	0.739	0.842	0.222	2.61	10.256
medium	355M	692M	1.0	0.558	0.756	0.852	0.235	2.553	10.238
small	124M	238M	3.0	0.552	0.759	0.852	0.173	2.554	10.239
medium	355M	692M	3.0	0.568	0.766	0.860	0.192	2.489	10.084

## C.5 VAE AND DIFFUSION HEAD

**Ablation on Diffusion Head.** We ablate the diffusion head  $\mathcal{H}$  in our motion branch to study how conditioning design affects generation. We vary (i) supervision with a diffusion head  $\mathcal{H}$  v.s. direct MSE regression, (ii) the mapping from backbone hidden states to the diffusion condition (multi-head attention, MHA vs. linear layer), (iii) the number of motion holders  $h\_num \in \{1, 4, 8\}$  used to query hidden states from the autoregressive backbone, and (iv) classifier-free guidance (CFG) at sampling. All variants are trained for 200K iterations with results in Tab. 11.

We observe that (i) Direct MSE supervision (the last row) yields the weakest performance, confirming the benefit of diffusion-based training. (ii) Increasing  $h\_num$  (b-d) enriches the conditioning signal and improves retrieval accuracy R@k, but also raises FID, suggesting a harder denoising problem. A moderate setting of  $h\_num = 4$  offers the best trade-off. (iii) With  $h\_num = 4$  and no CFG (c)(e), an MHA head outperforms a linear mapper, achieving higher R@3 (0.529 v.s. 0.521), lower MMDist (2.645 v.s. 2.713), and lower FID (0.166 v.s. 0.178). The advantage is even larger under sparse conditioning ( $h\_num = 1$  in (d)(e), FID: 0.164 vs. 0.283); (iv) Enabling CFG on the same configuration further improves both alignment and fidelity. Accordingly, we adopt diffusion supervision with an MHA head and  $h\_num = 4$  as the default.

**Guidance scale.** We sweep the CFG guidance scale  $\omega$  on the unified model (results in Tab. 7). Note that guidance is applied only to *motion generation*. Moderate guidance performs best:  $\omega = 5.0$  minimizes FID on the T2M, while  $\omega = 3.0$  yields the best R-Precision and MultiModal Distance. Very weak ( $\omega = 1$ ) or overly strong ( $\omega \geq 10$ ) guidance degrades alignment and diversity. We thus use  $\omega = 4$  in all main results.

Table 11: Ablation of motion generation head and loss on HumanML3D. All variants are trained on T2M task for 200k iterations, with same backbone and data. We vary: (i) supervision with diffusion head  $\mathcal{H}$  (Diff.) or direct MSE regression (MSE), (ii) mapping of multi-head attention (MHA) or Linear layer in  $\mathcal{H}$ , (iii) number of motion holder  $\langle \text{motion\_out} \rangle$  ( $h\_num$ ) (i.e., the count of hidden states passed from the backbone to  $\mathcal{H}$ ), and (iv) classifier-free guidance (CFG) at sampling.

ID	Loss	$h\_num$	Head	CFG	R@1 $\uparrow$	R@2 $\uparrow$	R@3 $\uparrow$	FID $\downarrow$	MMDist $\downarrow$	DIV $\rightarrow$	MModality $\uparrow$
(a)	Diff.	MHA	4	✓	<b>0.547</b>	<b>0.751</b>	<b>0.850</b>	<b>0.149</b>	<b>2.578</b>	10.041	2.265
(b)	Diff.	MHA	8	✗	0.531	0.733	0.836	0.185	2.655	10.154	2.198
(c)	Diff.	MHA	4	✗	0.529	0.730	<u>0.839</u>	0.166	<u>2.645</u>	10.012	2.350
(d)	Diff.	MHA	1	✗	0.525	0.729	0.831	<u>0.164</u>	2.678	10.090	2.514
(e)	Diff.	Linear	4	✗	0.521	0.731	0.829	0.178	2.713	9.985	<u>2.603</u>
(f)	Diff.	Linear	1	✗	0.525	0.729	0.829	0.283	2.689	10.069	<b>2.719</b>
(g)	MSE	Linear	4	-	0.518	0.725	0.823	0.276	2.705	9.758	2.175

Following MotionGPT’s evaluation protocol (Jiang et al., 2023), results are averaged over two runs. Best results are **bold**, second-best are underlined. The default configuration is gray-shaded.

Table 12: Ablation on guidance scale  $\omega$  in CFG, for motion latent diffusion on HumanML3D, with model trained on unified tasks. Best and second-best results are **bold** and underlined.

$\omega$	R@1	R@2	R@3	FID $\downarrow$	MMDist $\downarrow$	DIV $\rightarrow$	MModality $\uparrow$
Real	0.519	0.724	0.820	0.002	2.753	9.941	-
1	0.534	0.727	0.828	0.143	2.714	10.086	<b>1.717</b>
2	<b>0.555</b>	<u>0.754</u>	0.843	0.123	2.601	10.006	1.321
3	<u>0.554</u>	<b>0.756</b>	<b>0.850</b>	0.103	<b>2.585</b>	<b>9.926</b>	1.272
4	0.552	0.753	<u>0.848</u>	<u>0.098</u>	<u>2.589</u>	<u>9.911</u>	1.258
5	0.552	0.752	0.848	<b>0.094</b>	2.593	9.906	1.248
6	0.546	0.751	0.849	<b>0.094</b>	2.598	9.900	1.243
10	0.546	0.748	0.844	0.109	2.620	9.870	1.281
15	0.541	0.739	0.839	0.12	2.653	9.873	1.312
20	0.533	0.728	0.826	0.134	2.739	9.827	<u>1.385</u>

Following MotionGPT’s protocol, results are averaged over two runs. The default configuration is gray-shaded.



Table 13: Training-scheme evaluation on HumanML3D (Guo et al., 2022a), with protocol in Jiang et al. (2023). Stage1: T2M Pre-training, Stage2: Cross-Modal Alignment, Stage3: Joint Fintuning. ✓ marks enabled stages, while colors encode the state of text branch, updated or frozen. Jointly updating the text branch from the start improves early T2M in SI but degrades final T2M after SII and markedly lowers M2T scores (rows “Trained Text Branch”).

Type	Stage1	Stage2	Stage3	Text-to-Motion			Motion-to-Text		
				R@3 ↑	FID ↓	MMDist ↓	R@1↑	Bleu@4↑	BertScore↑
Three Stages	✓	✗	✗	0.826	0.239	2.797	-	-	-
	✓	✓	✗	<u>0.831</u>	<u>0.215</u>	<u>2.755</u>	0.571	18.328	33.993
	✓	✓	✓	<b>0.837</b>	<b>0.208</b>	<b>2.725</b>	0.573	19.412	35.231
Two Stages	✗	✓	✗	0.755	0.298	3.213	0.561	18.295	34.676
	✗	✓	✓	0.772	0.325	3.108	0.573	18.277	35.546
Trained Text Branch	✓	✗	-	0.822	0.239	2.832	-	-	-
	✓	✓	-	0.801	0.243	2.942	0.505	14.119	33.385

Following MotionGPT’s protocol, results are averaged over two runs.  
Best and second-best results are **bold** and underlined.

## C.6 EFFECTIVENESS OF TRAINING SCHEME

We adopt a three-stage schedule (see ??, ??): SI, text-to-motion (T2M) pretraining; SII, cross-modal alignment with joint optimization on T2M and motion-to-text (M2T) (SII); and SIII, joint fine-tuning. We evaluate three settings: (i) **Three Stages**, the full three-stage schedule, (ii) **Two Stage**, a two-stage schedule without SI, and (iii) **Trained Text Branch**, a two-stage variant in which the text branch is unfrozen during SI–SII, rendering SIII unnecessary. We report results on both generation (T2M) and understanding (M2T) in Tab. 13.

**Text-to-motion Pre-training and Cross-Modal Alignment.** Pretraining on T2M (SI) yields strong motion generation and provides a motion-specialized initialization. Entering SII confers M2T capability and further improves T2M (alignment improves and MMDist/FID drop), indicating that explicit cross-modal alignment benefits both directions. Training directly with multi-task objectives from scratch (i.e., without SI) markedly degrades T2M quality, even after subsequent joint optimization, underscoring the importance of a motion-specific warm start (i.e., initialization from T2M pretraining).

**Joint Fine-tuning.** Once SII has established cross-modal alignment, SIII yields modest gains, primarily stabilizing M2T while preserving T2M, and thus serves as a light refinement. The “Two Stages” variants (SII+SIII without SI) show that joint optimization can boost both tasks when the model is under-initialized. However, it also degrades the language branch’s competence, leading to worse T2M and M2T than the full three-stage schedule. Although incremental gains beyond SI+SII are modest, SIII can mitigate residual negative transfer and calibrate cross-modal alignment under noisy or shifted training conditions, yielding more stable results.

**Freezing v.s. training the text branch.** To promote modality-specific representations and reduce negative transfer onto a well-trained language branch, we propose to freeze the text branch in SI–SII. Freezing preserves linguistic competence while the motion branch specializes. By contrast, updating all parameters from the start (“Trained Text Branch”) can give slightly higher T2M scores in SI (e.g., R@3 0.834 vs. 0.820; MMDist 2.698 vs. 2.787), but after SII these models exhibit degraded T2M and notably weaker M2T (e.g., BertScore 0.713; Bleu@4 3.577), consistent with ‘catastrophic forgetting’. We attribute this decline to negative transfer from the new motion branch onto the text branch during early training. In practice, keeping the LM frozen lets the motion branch learn a more stable, modality-specific space and achieve more reliable alignment under *limited* paired data.

In summary, SI provides essential motion-specific initialization; SII delivers the bulk of cross-modal gains; SIII offers small, stabilizing improvements. Freezing the text branch through SI–SII prevents loss of linguistic ability and yields the best overall balance between understanding and generation.

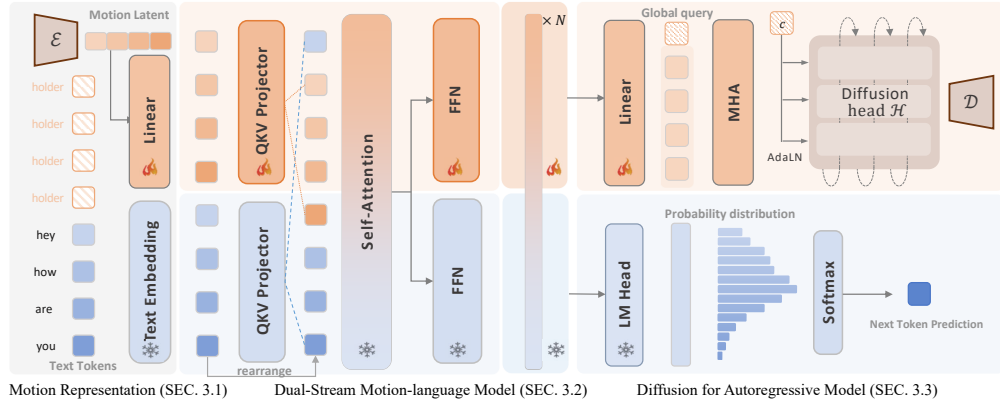


Figure 11: Details of our bimodal motion-language model. **Motion** and **text** inputs are encoded by separate branches and then reordered to their original sequence order before cross-modal self-attention. After  $N$  hybrid layers, text is generated autoregressively by next-token prediction, while motion is produced via a diffusion head  $\mathcal{H}$ . Panels correspond to motion representation (??), the dual-stream motion-language backbone (??), and the diffusion module (??).

## D DETAILS ON MOTIONGPT3

### D.1 MOTION GENERATION IN UNIFIED MODEL

**Diffusion Loss** We use a diffusion head  $\mathcal{H}$  to map backbone hidden states into a denoised motion latent. As illustrated in ?? and Fig. 11, we insert  $K$  *motion holder* tokens `<motion_out>` as queries to extract the corresponding output states from the motion branch. Let the backbone hidden size be  $d_t$  and the motion latent size be  $d_m$  (we set  $d_t=768$ ,  $d_m=256$  in our default model). After one forward pass, the queried states form  $h \in \mathbb{R}^{K \times d_t}$ . An MHA pooling module aggregates  $h$  and produces a global condition vector  $c \in \mathbb{R}^{1 \times d_m}$  via an internal mapping to match the motion-latent dimensionality, which is then fused with the timestep embedding  $\tau(t)$  to condition  $c$ . With the cumulative product of the noise schedule  $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$ , we sample  $t \sim \mathcal{U}\{1, \dots, T\}$  and corrupt the ground-truth motion latent  $z_0 \in \mathbb{R}^{d_m}$  by

$$z_t = \sqrt{\bar{\alpha}_t} z_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \quad (2)$$

Given a noisy latent  $z_t \in \mathbb{R}^{1 \times d_m}$  at timestep  $t$ , the head predicts the noise  $\hat{\epsilon}_\theta$  and is trained with the standard  $\epsilon$ -prediction objective

$$L_{diff} = \mathcal{E}_{z_0, \epsilon, t} \|\epsilon - \hat{\epsilon}_\theta(\alpha_t z_0 + \sigma_t \epsilon, t, c)\|_2^2, \quad \hat{\epsilon}_\theta = \mathcal{H}(z_t, t, c) \quad (3)$$

where  $\alpha_t, \sigma_t$  follow a linear schedule equivalent to the forward noising process. At inference, we start from  $z_T \sim \mathcal{N}(0, I)$  iteratively denoise with the sampler of  $\mathcal{H}$  down to  $t=0$  to obtain  $\hat{z}_0$ , which is then decoded by the motion decoder  $\mathcal{D}$  into a raw motion sequence.

**Architecture of  $\mathcal{H}$ .** The diffusion head first processes the  $K$  queried hidden states with a TransformerEncoderLayer and aggregates them via multi-head attention pooling into a single condition vector  $c$ . We fuse  $c$  with the timestep embedding and modulate each block via AdaLN.  $\mathcal{H}$  consists of a stack of 1024-wide residual blocks. Each block applies AdaLN followed by a two-layer MLP with SiLU nonlinearity, and the final block projects to the noise prediction  $\hat{\epsilon}_\theta$ .

**Cross-Entropy Loss for Boundary Tokens** To delimit motion from text decoding, we introduce two boundary tokens `<som>` (start of motion) and `<eom>` (end of motion). At inference, once the LM predicts `<som>` via next-token prediction, we generate the motion latent in a single forward pass, with  $K$  `<motion_out>` holders concatenated to the sequence, and then append `<eom>` deterministically. During training, we apply cross-entropy only to the `<som>` prediction, and `<eom>` is not supervised.

## D.2 MOTION VAE

**AutoEncoder** We adopt a Transformer-based motion VAE (Xin et al., 2023) with an encoder  $\mathcal{E}$  and a decoder  $\mathcal{D}$  that maps an  $M$ -frame motion sequence  $m^{1:M}$  to a compact continuous latent  $z \in \mathbb{R}^{n \times d}$  ( $n = 1, d = 256$ ) and reconstructs the motion via  $m^{1:M} = \mathcal{D}(z) = \mathcal{D}(\mathcal{E}(m^{1:M}))$ . Both  $\mathcal{E}$  and  $\mathcal{D}$  are transformers (Vaswani et al., 2017) with long skip connection (Ronneberger et al., 2015), and without the action biases used in Petrovich et al. (2021). This design yields an expressive latent space that supports accurate semantic understanding and high-fidelity, diverse motion synthesis.

**Architecture** Given an input motion sequence  $m^{1:M}$  of length  $M$ , the encoder  $\mathcal{E}$  processes the sequence together with a small set of learnable distribution tokens and outputs the Gaussian parameters  $(\mu_m, \sigma_m)$ . A latent  $z$  is sampled by reparameterization  $z = \mu_m + \sigma_m \epsilon, \epsilon \sim \mathcal{N}(0, I)$ . The decoder  $\mathcal{D}_{mld}$  performs cross-attention over the latent vector  $z$  to query  $L$  motion tokens, which are then projected back into  $\hat{m}^{1:L}$  in the raw motion space.

$$\hat{m}^{1:L} = \mathcal{D}(\mathcal{E}(m^{1:M})) \quad (4)$$

**Loss** We train the motion VAE with reconstruction term over framewise poses and KL regularizer on the latent, following standard practice in VAEs (Kullback & Leibler, 1951; Kingma & Welling, 2013). Let  $m^{1:M}$  denote a sequence of  $M$  ground-truth poses  $m_t \in \mathbb{R}^{263}$  and  $\hat{m}^{1:M}$  the decoder outputs. The encoder produces a Gaussian posterior  $q_\phi(z | m^{1:M}) = \mathcal{N}(\mu, \text{diag}(\sigma^2))$ . The objective is

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda_{\text{KL}} \mathcal{L}_{\text{KL}}, \quad (5)$$

with

$$\mathcal{L}_{\text{rec}} = \frac{1}{T} \sum_{t=1}^T \|m_t - \hat{m}_t\|_2^2, \quad (6)$$

$$\mathcal{L}_{\text{KL}} = D_{\text{KL}}(\mathcal{N}(\mu, \text{diag}(\sigma^2)) \| \mathcal{N}(0, I)). \quad (7)$$

For completeness, the KL term admits the closed form  $\mathcal{L}_{\text{KL}} = \frac{1}{2} \sum_j (\mu_j^2 + \sigma_j^2 - \log \sigma_j^2 - 1)$ .

**Raw Motion Representation** Following Guo et al. (2022a), each frame  $m^i \in \mathbb{R}^{263}$  concatenates a tuple of root angular velocity  $\dot{r}^a \in \mathbb{R}$  along Y-axis, root linear velocities  $(\dot{r}^x, \dot{r}^z) \in \mathbb{R}$  on XZ-plane, root height  $r^y \in \mathbb{R}$ , local joints positions  $j^p \in \mathbb{R}^{3N_j}$ , velocities  $j^v \in \mathbb{R}^{3N_j}$ , and rotations  $j^r \in \mathbb{R}^{6N_j}$  in root space, with  $N_j$  denotes the joint number, and binary foot ground contact features  $c^f \in \mathbb{R}^4$  by thresholding the heel and toe joint velocities. This finally results in  $m^i = \{\dot{r}^a, \dot{r}^x, \dot{r}^z, r^y, j^p, j^v, j^r, c^f\}$ .

## D.3 METRICS DEFINITIONS

We adopt standard metrics for text–motion alignment, motion quality and diversity, caption quality, and (for VAE analysis) motion reconstruction. Unless noted, features are computed with the official HumanML3D/T2M evaluator (Guo et al., 2022b), with motion encoder  $\phi(m)$  and text encoder  $\psi(t)$ .

**Text–motion alignment** To evaluate semantic consistency between generated motions and input texts, we adopt motion-text retrieval precision (R-Precision) at Top-k(R@k), and the Multimodal Distance (MMDist), which measures the embedding-space distance between paired modalities. **R@k** measures retrieval accuracy within a candidate set: for each query (text or motion), we rank candidates of the other modality by cosine similarity and report the fraction of cases where the paired item appears in the top-k. **MMDist** is the average embedding distance between paired items:

$$\text{MMDist} = \frac{1}{N} \sum_{n=1}^N \|\phi(m_n) - \psi(t_n)\|_2 \quad (8)$$

**Motion quality FID** assess how closely generated motions match ground truth ones in feature space, indicating overall quality, and is computed between the Gaussian fits of  $\{\phi(m)\}$  for generated and ground-truth motions in the evaluator feature space.

**Diversity** Diversity (DIV) measures feature variation across samples, and MultiModality (MM), which quantifies variation among motion generations from the same textual description. Following Guo et al. (2022b); Xin et al. (2023), we randomly sample all generated motions into two subsets,  $\{x_i\}_{i=0}^{X_d}$  and  $\{x'_i\}_{i=0}^{X_d}$ , of the same size  $X_d$ . Then **DIV** is formalized as:

$$\text{DIV} = \frac{1}{X_d} \sum_{i=1}^{X_d} \|x_i - x'_i\| \quad (9)$$

Randomly sample a set of text descriptions with size  $J_m$  and sample two subsets of  $X_m$  motions generated by  $j$ -th for each text description, denote as  $\{x_{j,i}\}_{i=0}^{X_m}$  and  $\{x'_{j,i}\}_{i=0}^{X_m}$ . **MM** is calculated as:

$$\text{MM} = \frac{1}{J_m \times X_m} \sum_{i=1}^{J_m} \sum_{i=1}^{X_m} \|x_{j,i} - x'_{j,i}\| \quad (10)$$

**Motion captioning** We follow prior work chuan2022tm2t and adopt standard NLP metrics including BLEU (Papineni et al., 2002), ROUGE-L (Lin, 2004), CIDEr (Vedantam et al., 2015), and BERTScore (Zhang et al., 2019) to evaluate the fluency, relevance, and diversity of generated captions.

**Reconstruction** We evaluate reconstruction fidelity of motion autoencoders with: **MPJPE** and **PAMPJPE** for global and local errors in millimeter, **ACCL** (Acceleration Error) computed from second-order finite differences, and **APE/AVE** (Absolute Position/Velocity Error) reported over root, trajectory, pose, and joints components.

## REFERENCES

- Junbin Cho, Junwan Kim, Jisoo Kim, Minseo Kim, Mingu Kang, Sungeun Hong, Tae-Hyun Oh, and Youngjae Yu. Discord: Discrete tokens to continuous motion via rectified flow decoding, 2025. URL <https://arxiv.org/abs/2411.19527>.
- Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5152–5161, June 2022a.
- Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5152–5161, 2022b.
- Chuan Guo, Xinxin Zuo, Sen Wang, and Li Cheng. Tm2t: Stochastic and tokenized modeling for the reciprocal generation of 3d human motions and texts. In *ECCV*, 2022c.
- Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. Momask: Generative masked modeling of 3d human motions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1900–1910, 2024.
- Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiongpt: Human motion as a foreign language. *Advances in Neural Information Processing Systems*, 36:20067–20079, 2023.
- Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.
- Zhe Li, Weihao Yuan, Yisheng He, Lingteng Qiu, Shenhao Zhu, Xiaodong Gu, Weichao Shen, Yuan Dong, Zilong Dong, and Laurence T Yang. Lamp: Language-motion pretraining for motion generation, retrieval, and captioning. *arXiv preprint arXiv:2410.07093*, 2024.
- Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.
- Yunhong Lou, Linchao Zhu, Yaxiong Wang, Xiaohan Wang, and Yi Yang. Diversemotion: Towards diverse human motion generation via discrete diffusion. *arXiv preprint arXiv:2309.01372*, 2023.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

- Mathis Petrovich, Michael J. Black, and Gül Varol. Action-conditioned 3D human motion synthesis with transformer VAE. In *International Conference on Computer Vision (ICCV)*, 2021.
- Mathis Petrovich, Michael J. Black, and Gül Varol. TEMOS: Generating diverse human motions from textual descriptions. In *European Conference on Computer Vision (ECCV)*, 2022.
- Mathis Petrovich, Michael J Black, and Gül Varol. Tmr: Text-to-motion retrieval using contrastive 3d human motion synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9488–9497, 2023.
- Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4566–4575, 2015.
- Yuan Wang, Di Huang, Yaqi Zhang, Wanli Ouyang, Jile Jiao, Xuetao Feng, Yan Zhou, Pengfei Wan, Shixiang Tang, and Dan Xu. Motiongpt-2: A general-purpose motion-language model for motion generation and understanding. *arXiv preprint arXiv:2410.21747*, 2024.
- Yuqing Wang, Zhijie Lin, Yao Teng, Yuanzhi Zhu, Shuhuai Ren, Jiashi Feng, and Xihui Liu. Bridging continuous and discrete tokens for autoregressive visual generation, 2025. URL <https://arxiv.org/abs/2503.16430>.
- Bizhu Wu, Jinheng Xie, Keming Shen, Zhe Kong, Jianfeng Ren, Ruibin Bai, Rong Qu, and Linlin Shen. Mg-motionlm: A unified framework for motion comprehension and generation across multiple granularities. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 27849–27858, 2025.
- Yiming Wu, Wei Ji, Kecheng Zheng, Zicheng Wang, and Dong Xu. Mote: Learning motion-text diffusion model for multiple generation tasks. *arXiv preprint arXiv:2411.19786*, 2024.
- Chen Xin, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, Jingyi Yu, and Gang Yu. Executing your commands via motion diffusion in latent space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023.
- Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Shaoli Huang, Yong Zhang, Hongwei Zhao, Hongtao Lu, and Xi Shen. T2m-gpt: Generating human motion from textual descriptions with discrete representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023a.
- Mingyuan Zhang, Xinying Guo, Liang Pan, Zhongang Cai, Fangzhou Hong, Huirong Li, Lei Yang, and Ziwei Liu. Remodiffuse: Retrieval-augmented motion diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 364–373, 2023b.
- Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motion-diffuse: Text-driven human motion generation with diffusion model. *IEEE transactions on pattern analysis and machine intelligence*, 46(6):4115–4128, 2024.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.
- Zeyu Zhang, Yiran Wang, Wei Mao, Danning Li, Rui Zhao, Biao Wu, Zirui Song, Bohan Zhuang, Ian Reid, and Richard Hartley. Motion anything: Any to motion generation. *arXiv preprint arXiv:2503.06955*, 2025.