T2MBench: Evaluating Text-to-Motion Generation under Diverse Language Instructions

Anonymous ACL submission

Abstract

As text-to-motion (T2M) models move beyond short action captions toward real-world applications, they are increasingly expected to follow diverse instructions. However, existing evaluation protocols remain insufficient for systematically assessing instruction following, physical plausibility, and fine-grained controllability under such diverse language instructions. To address this gap, we introduce T2MBench, a benchmark with 1,025 prompts across four categories—Dynamics, Complexity, Interaction, and Accuracy—covering 14 representative T2M models and three evaluation dimensions: LLM-Based Evaluation, Multi-Factor Motion Evaluation, and Fine-grained Accuracy Evaluation. Our results reveal substantial gaps in current T2M models' ability to consistently follow diverse language instructions while maintaining motion plausibility and fine-grained control.T2MBench establishes a reusable diagnostic benchmark for advancing research toward more reliable T2M generation under diverse language instructions. Data and materials are available at https://anonymous.4open.science/status/T2MBench-review-7078.

1 Introduction

T2M models aim to generate human motions from natural language descriptions, with applications in character animation, embodied AI, VR/AR, and human–computer interaction (Khani et al., 2025; Li et al., 2026; Sui et al., 2026). Recent motion-language datasets, such as KIT-ML, HumanML3D, and Motion-X, have provided large-scale foundations for learning text-motion correspondences, while diffusion-based and token-based generative models have substantially improved the quality of caption-conditioned motion generation (Plappert et al., 2016; Guo et al., 2022; Lin et al., 2024; Tevet et al., 2022; Jiang et al., 2023; Guo et al., 2023). As T2M models move beyond short action captions, they are increasingly expected to follow diverse language instructions involving dynamic styles, temporal composition, interaction intent, body-part constraints, and precise spatial or numerical control (Zhang et al., 2023; Xie et al., 2024; Liang et al., 2024; Han et al., 2024).

Evaluation protocols, however, have not fully kept pace with this shift. Existing metrics and benchmarks provide useful signals for text-motion matching, distributional quality, physical artifacts, event-level alignment, or open-world generalization, but they often examine these aspects separately (Guo et al., 2022; Tevet et al., 2022, 2024; Voas, 2023; Li et al., 2026). As a result, current evaluation remains fragmented: it lacks both a systematically diverse instruction suite and a unified framework for assessing whether generated motions follow complex instructions while remaining plausible, diverse, and controllable.

Several recent benchmarks have begun to broaden T2M evaluation, such as event-level alignment and generalization-oriented assessment (Lin et al., 2026; Han et al., 2024). However, they are not primarily designed to treat diverse instruction following as the central target. As summarized in Table 1, existing protocols provide valuable but partial coverage across diversity, human validation, physical quality, event structure, interaction, and control. This motivates T2MBench, which combines diverse instruction construction, generation with representative T2M models, and complementary evaluation from semantic, physical, diversity, and fine-grained control perspectives.

To this end, we introduce **T2MBench**, a benchmark for evaluating text-to-motion generation under diverse language instructions. As illustrated in Figure 1, T2MBench follows a three-stage pipeline: diverse instruction construction, motion generation with representative T2M models, and multi-dimensional evaluation. Our contributions are sum-

Work / Direction	Sem. Align.	Div./Multi.	Human Val.	Phys. Qual.	Temp./Event	Spatial Ctrl.	Numeric Ctrl.	Eval. Suite	#Models
HumanML3D (Guo et al., 2022)	✓	✓	△	✗	✗	✗	✗	✓	6
Metric Validity (Voas, 2023)	✓	✗	✓	✓	✗	✗	✗	✓	4
CLoSD (Tevet et al., 2024)	✓	△	✗	✓	△	✓	✗	✗	6
FineMoGen (Zhang et al., 2023)	✓	✓	✗	✗	✓	✓	△	✓	8
OmniControl (Xie et al., 2024)	✓	✗	✗	△	△	✓	△	✗	4
InterGen (Liang et al., 2024)	✓	✓	✗	✗	△	△	✗	✓	6
AToM (Han et al., 2024)	✓	✓	✓	✗	✓	✗	✓	✓	3
MBench (Lin et al., 2026)	✓	△	✓	✓	△	✗	✗	✓	9
T2MBench	✓	✓	✓	✓	✓	✓	✓	✓	14

Table 1: Comparison with representative T2M evaluation works. ✓/△/✗ indicate explicit, partial, and non-primary coverage. Abbreviations: Sem. Align. (Semantic Alignment), Div./Multi. (Diversity and Multimodality), Human Val. (Human Validation), Phys. Qual. (Physical Quality), Temp./Event (Temporal and Event Consistency), Spatial Ctrl. (Spatial Control), Numeric Ctrl. (Numeric Control), Eval. Suite (curated evaluation suite), and #Models (number of evaluated models). Phys. Qual. covers foot sliding/floating, ground penetration, jitter degree, dynamic degree, pose quality, and body penetration; Temp./Event covers temporal order and multi-stage action consistency; Spatial Ctrl. covers position, direction, trajectory, and body-part spatial relations; Numeric Ctrl. covers distance, speed, angle, count, and frequency.

marized as follows:

- **T2MBench for diverse instruction evaluation.** We propose T2MBench, a benchmark centered on diverse language instructions, to jointly assess T2M model abilities in semantic alignment, physical plausibility, generalizability, and fine-grained control.

- **High-quality dataset curation.** We curate 1,025 diverse language instructions and collect generated motions from 14 representative T2M models under these instructions with multiple formats, offering a valuable resource for reproducible evaluation and future research in T2M generation.

- **Systematic model evaluation and analysis.** We systematically evaluate 14 representative T2M models using LLM-Based Evaluation, Multi-Factor Motion Evaluation, and Fine-grained Accuracy Evaluation. Our analysis reveals their key limitations in instruction following, motion plausibility, and fine-grained control under diverse language instructions.

2 Related Works

Text-to-motion generation has been advanced by diffusion-based methods (Tevet et al., 2022; Shafir et al., 2023; Yuan et al., 2024) and VQ-based paradigms (Guo et al., 2022; Jiang et al., 2023; Zhou et al., 2023; Guo et al., 2023), supported by motion-language datasets such as KIT-ML, HumanML3D, and Motion-X (Plappert et al., 2016; Guo et al., 2022; Lin et al., 2024).

Evaluation for T2M generation has gradually evolved from retrieval-based and distributional metrics to more diagnostic protocols. Standard metrics such as Matching Score, R-Precision, FID, Diversity, and Multimodality measure text-motion matching, distributional realism, and motion variation (Guo et al., 2022; Tevet et al., 2022). Later studies examine metric validity and human correlation (Voas, 2023), while recent works introduce physical-quality diagnostics and model-based evaluators for motion plausibility and higher-level alignment (Tevet et al., 2024; Lin et al., 2026; Han et al., 2024).

Recent benchmarks further broaden T2M evaluation from different perspectives. AToM focuses on event-level alignment, including motion integrity, temporal order, and frequency (Han et al., 2024), while MBench emphasizes open-world generalization, motion quality, and prompt fidelity (Lin et al., 2026). Other related works study interaction generation, spatial control, or fine-grained motion constraints (Liang et al., 2024; Zhang et al., 2023; Xie et al., 2024). These directions are complementary to our setting. In contrast, T2MBench treats diverse instruction following as the central target and jointly evaluates semantic alignment, physical plausibility, motion diversity, and fine-grained control.

3 T2MBench

This section describes the instruction construction and filtering process (Sec. 3.1), the diversity analysis of T2MBench language instructions (Sec. 3.2), the evaluated baselines (Sec. 3.3), and the evaluation framework (Sec. 3.4).

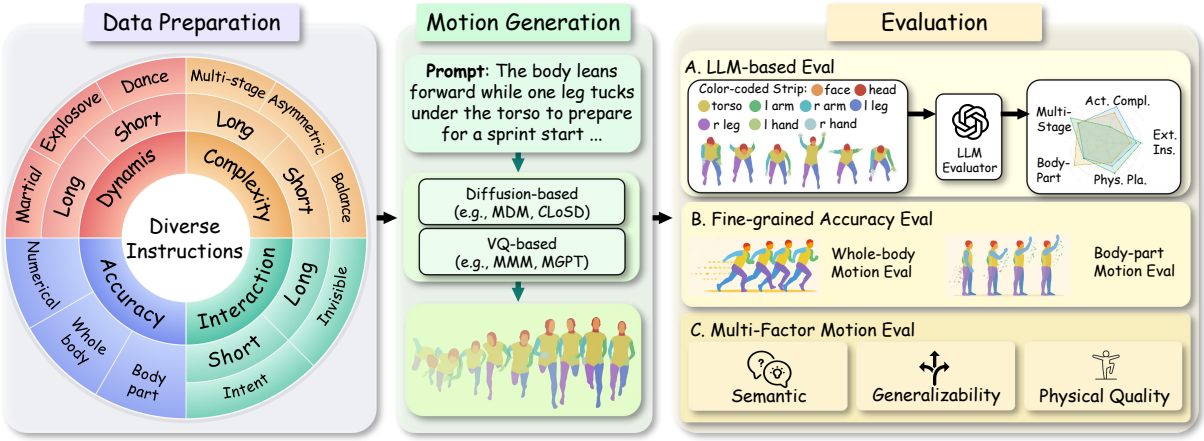

Figure 1: Overview of T2MBench. We construct a diverse instruction suite, generate motions using 14 representative T2M models, and evaluate the results with LLM-Based Evaluation, Multi-Factor Motion Evaluation, and Fine-grained Accuracy Evaluation.

3.1 Instruction Generation and Filtering

This section describes how we construct and filter the T2MBench diverse instruction suite.

Construction Pipeline. We refer to KIT-ML, HumanML3D, Motion-X, ViMoGen-228K, and the Oxford Dictionary (Plappert et al., 2016; Guo et al., 2022; Lin et al., 2024, 2026) for motion verbs, body-part expressions, action description styles, and motion-related semantic categories, using them only as references for generating a new set of benchmark instructions. We use GPT-5.2 as the instruction generation model. To cover different instruction types, we design category-specific generation prompts for each subtype. The detailed prompts are provided in Appendix A.

Filtering Pipeline. The initial construction stage produces 7,730 candidate instructions. We then manually filter these candidates that are duplicated, semantically ambiguous, physically implausible, overly abstract, difficult to verify visually, or dependent on external scene context. After filtering, we retain 1,025 prompts as the final T2MBench instruction suite.

3.2 Taxonomy and Diversity Analysis

Taxonomy. The final T2MBench instruction suite contains four categories, as shown in the Diverse Instructions panel of Figure 1: Dynamics, Complexity, Interaction, and Accuracy. Dynamics targets motion style and intensity, Complexity covers temporal and structural composition, Interaction focuses on interaction intent and invisible-object interaction, and Accuracy includes fine-grained spatial and numerical constraints. Dynamics, Complexity, and Interaction contain both long and short instructions, while Accuracy consists of short high-precision prompts. The detailed category and subcategory distribution is provided in Appendix A.3.

Dataset	MTLD↑	Self-BLEU↓	Occ.↑
MPref	40.5134±0.5357	0.6284±0.0060	0.9767±0.0274
HML3D	43.8326±1.3180	0.5490±0.0047	0.9700±0.0189
KIT	31.4755±0.8849	0.7253±0.0067	0.8900±0.0316
ViGen	52.0234±1.4110	0.5402±0.0080	0.9833±0.0236
Ours	**62.2415**±0.4150	**0.5292**±0.0009	**0.9900**±0.0225

Table 2: General linguistic and semantic diversity under repeated balanced sampling. Each run samples 1,000 instructions from each dataset and is repeated over 10 random seeds. Values are mean ± standard deviation. MPre, HML3D, and KIT denote MotionPrefer, HumanML3D, and KIT-ML, respectively.

Diversity Analysis of Language Instructions. To validate the diversity of T2MBench, we compare it with representative T2M language resources, including KIT-ML, HumanML3D, MotionPrefer, and ViMoGen-228k (Plappert et al., 2016; Guo et al., 2022; Han et al., 2024; Lin et al., 2026). We normalize all resources into a unified text format to analyze instruction diversity. Detailed normalization, and the basic information of compared datasets are provided in Appendix B.1.

Table 2 reports lexical and semantic diversity, surface-level repetition, and semantic coverage using MTLD (McCarthy and Jarvis, 2010; He et al., 2025), Self-BLEU (Papineni et al., 2002; Zhu et al., 2018; Kowshik et al., 2024), and occupied-cluster ratio (Occ.) (Reimers and Gurevych, 2019; Tang et al., 2025), respectively. Detailed metric definitions and cluster-sensitivity analysis are provided in Appendix B.2 and Appendix B.3. T2MBench

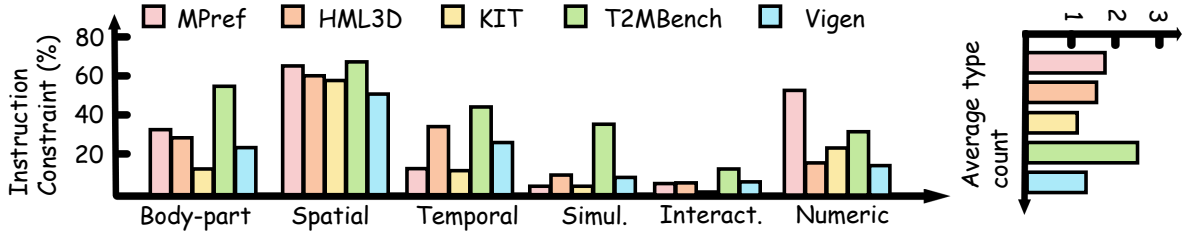

Figure 2: Motion-oriented constraint diversity across text-to-motion datasets. Left: percentage of texts containing each constraint type. Right: average number of constraint types per text. All constraints are detected using the same fixed lexicon-based matching rules. The full numerical table and lexicon details are provided in Appendix C.

achieves the highest MTLD, lowest Self-BLEU, and highest Occ. ratio, indicating stronger lexical variation, lower template-like redundancy, and broader semantic coverage than the compared resources. These results show that T2MBench is not a simple template expansion, but provides linguistically and semantically diverse instructions.

Figure 2 compares motion-oriented constraint diversity, including body-part, spatial, temporal, simultaneous, interaction, and numerical constraints. We use the same lexicons and matching rules for all datasets, with full numerical table and lexicon details provided in Appendix C. T2MBench achieves the highest coverage in five out of six constraint types: body-part, spatial, temporal, simultaneous, and interaction. MotionPrefer obtains the highest numerical-constraint ratio, mainly due to its frequency-oriented prompts. The right panel further shows that T2MBench has the highest average number of constraint types per instruction, reaching 2.51. Thus, T2MBench diversity is reflected not only in linguistic and semantic variation, but also in denser motion-oriented constraint composition.

We further conduct a common-support length-controlled validation, showing that T2MBench's diversity advantage persists after controlling all datasets to the same short-to-medium length distribution; full results and sampling details are provided in Appendix B.4.

3.3 Text-to-Motion Baselines

We benchmark 14 representative Text-to-Motion systems, including diffusion-based methods (MDM, CLoSD, PriorMDM, MoGenTS, HY, DART) (Tevet et al., 2022, 2024; Shafir et al., 2023; Yuan et al., 2024; Wen et al., 2025; Zhao et al., 2025) and VQ-based methods (T2M, MotionGPT, MMM, AvatarGPT, SALAD, MoMask, ViMoGen, MaskControl) (Guo et al., 2022; Jiang et al., 2023; Pinyoanuntapong et al., 2024; Zhou et al., 2023; Hong et al., 2025; Guo et al., 2023; Lin et al., 2026; Pinyoanuntapong et al., 2025).

3.4 Evaluation Framework

We evaluate T2M generation under diverse language instructions from three complementary levels. **LLM-based evaluation** (Sec. 3.4.1) assesses high-level instruction following and visual plausibility. **Multi-factor motion evaluation** (Sec. 3.4.2) quantifies semantic preservation, generative diversity, and physical quality with automatic motion-level metrics. **Fine-grained accuracy evaluation** (Sec. 3.4.3) measures controllability under explicit spatial and numerical constraints.

3.4.1 LLM-Based Evaluation

As shown in the Evaluation panel (A. LLM-based Eval.) of Fig. 1, for each generated motion, we render the SMPL sequence into a color-coded spatio-temporal strip. The body-part colors help the evaluator distinguish limbs and side-specific movements, while the horizontally concatenated frames represent temporal progression from left to right. We then provide the evaluation system prompt, the rendered motion, and the corresponding language instruction to GPT-5.2 for evaluation.

We evaluate each rendered motion using five aspects: **Extra Non-Instruction Actions**, **Action Completeness**, **Multi-Stage Order Correctness**, **Body-Part Understanding**, and **Physical Plausibility**. Together, these aspects assess whether the generated motion contains extra actions beyond the instruction, covers the required action components, follows the instructed temporal order, correctly reflects the specified body-part descriptions, and remains visually plausible. LLM-based evaluation complements automatic kinematic metrics by judging high-level motion-text alignment that is difficult to capture with numerical motion statistics alone. Detailed rendering settings, color-coding rules, and the evaluation system prompt are provided in Appendix D.

3.4.2 Multi-Factor Motion Evaluation

Multi-Factor Motion Evaluation provides automatic motion-level diagnostics from three complementary dimensions: semantic alignment, generalizability, and physical quality. **Semantic Alignment** measures whether the generated motion preserves the action-level semantics of the input instruction. We use Automatic Similarity Recall (ASR), where the generated motion is first captioned by FoundationMotion (Gan et al., 2025) and then compared with the original ground-truth prompt at the atomic-action level. **Generalizability** evaluates whether a model can produce varied motions rather than collapsing to similar patterns. We report Multimodality (MM) for variations under the same instruction and Diversity for variations across different instructions. **Physical Quality** assesses motion realism and physical plausibility using seven diagnostic metrics: Jitter Degree (JD), Ground Penetration (GP), Foot Floating (FF), Foot Sliding (FS), Dynamic Degree (DD), Position Quality (PQ), and Body Penetration (BP). We do not use Matching Score, R-Precision, and MM Dist as evaluation metrics because they rely on pretrained text–motion encoders, whose representations may be biased when applied to our diverse instructions. Detailed definitions and computation procedures are provided in Appendix E.

3.4.3 Fine-grained Accuracy Evaluation

Fine-grained Accuracy Evaluation tests whether a model can satisfy explicit spatial, kinematic, and body-part constraints, rather than only producing semantically plausible motions. We use the *Whole-body* and *Body-part* subcategories under the *Accuracy* category to evaluate fine-grained accuracy. The whole-body subcategory evaluates global root-level controllability using **Root Rotation**, **Root Velocity**, and **Root Translation**, which measure errors in target yaw rotation, directional speed, and displacement, respectively. The body-part subcategory evaluates local spatial controllability using **Body-part Translation**, which measures whether the generated motion satisfies the target relative position between two specified body parts.

For automatic evaluation, we extract structured target specifications from the original instructions using GPT-5.2. Whole-body targets are stored in `root_move.json`, while body-part targets are stored in `body_part.json`. These structured targets are manually verified and used only as reference annotations for error computation, not as inputs to the evaluated models. Detailed target extraction, notation, and metric computation procedures are provided in Appendix F.

Setting	Pref. Agree.↑	Spearman ρ↑	Kendall τ↑
Gemini-3	0.7833±0.0144	0.6292±0.0289	0.5667±0.0289
GLM-4.6V	0.7667±0.0083	0.6042±0.0072	0.5333±0.0167
Llama-4	0.8333±0.0144	0.7083±0.0315	0.6667±0.0289
Qwen3	0.7750±0.0220	0.6167±0.0439	0.5500±0.0441
Claude-4.5	0.7028±0.0127	0.4417±0.0191	0.4056±0.0255
GPT-5.2	**0.8528**±0.0192	**0.7542**±0.0315	**0.7056**±0.0385
No color	0.8194±0.0127	0.6792±0.0289	0.6389±0.0255
Skeleton	0.5944±0.0293	0.1875±0.0901	0.1889±0.0585

Table 3: Agreement with aggregated human preferences under different evaluator and input settings.

4 Experiments and Findings

4.1 Validation of LLM-Based Evaluation

To validate the reliability of our LLM-based evaluator, we construct a human preference reference set with 40 instructions sampled from the four T2MBench categories. For each instruction, 15 human raters rank three anonymized motions generated by HY, MotionGPT, and CLoSD according to overall prompt-following quality and motion plausibility. Using this reference set, we measure human inter-rater agreement, compare multiple multimodal LLM evaluators against aggregated human preferences, ablate the visual input representation, and repeat each evaluator or input setting three times to report mean and standard deviation.

Human Preference Agreement. To verify that the collected human preferences provide a reliable reference signal, we report the agreement among the 15 human raters. The annotations show strong inter-rater agreement, with Kendall's $W = 0.9076 \pm 0.1589$, Fleiss' $\kappa = 0.8390$, and Majority Margin $= 0.8628 \pm 0.2554$, supporting the use of human preferences as the reference for LLM-human agreement analysis (Kendall and Smith, 1939; Fleiss, 1971; Bakiyev et al., 2026). Detailed annotation protocol, agreement metrics, and aggregated preference statistics are provided in Appendix G.

Selection of LLM Evaluator and Input Format. We next select LLM evaluator and input format with the aggregated human preferences. For evaluator comparison, we use the color-coded spatial-temporal strip and compare GPT-5.2 with Gemini-3-Flash-Preview, GLM-4.6V, Llama-4-Maverick, Qwen3-VL-32B-Instruct, and Claude-Opus-4.5. As shown in Table 3, GPT-5.2 achieves the best

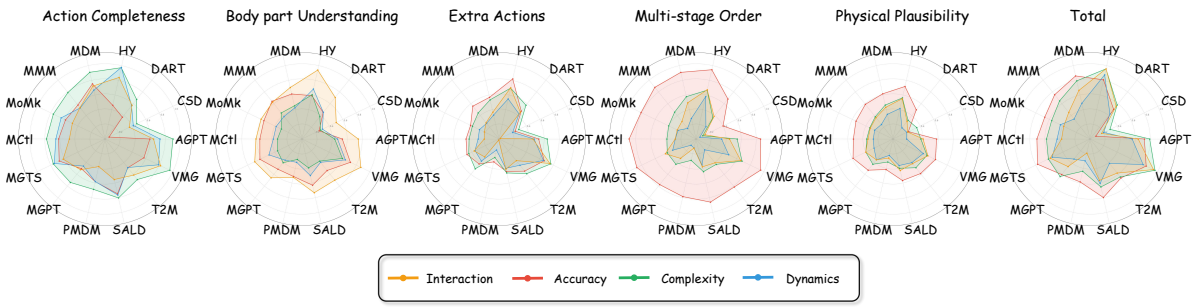

Figure 3: LLM-based evaluation radar charts over five sub-dimensions and the total score. Each chart compares 14 T2M baselines under four instruction categories. Abbreviations: CSD/AGPT/VMG/SALD/PMDM/MGPT/MGTS/MCtl/MoMk denote CLoSD, AvatarGPT, ViMoGen, SALAD, PriorMDM, MotionGPT, MoGenTS, MaskControl, and MoMask.

with aggregated human preferences, and is therefore used as the default LLM evaluator in our main experiments (Bradley and Terry, 1952; Zheng et al., 2023; Spearman, 1904; Liu et al., 2023; Kendall, 1938; Gao et al., 2025). For input ablation, we then fix GPT-5.2 and compare the default color-coded rendering with no-color rendering and skeleton-only input. The color-coded rendering achieves the best with aggregated human preferences, so we adopt the combination of GPT-5.2 and color-coded rendering as the LLM-based evaluation setting. Each setting is evaluated three times on the same prompt-motion pairs, and results are reported as mean $\pm$ standard deviation. Detailed trend construction, repeated-run protocol, metric computation, and input examples are provided in Appendix H.

4.2 Main Benchmark Results

4.2.1 LLM-Based Evaluation Results

Figure 3 summarizes the LLM-based evaluation results across five sub-dimensions and the total score. Each radar chart compares 14 T2M baselines under four instruction categories: Dynamics, Complexity, Interaction, and Accuracy. The scores are normalized for visualization, with the detailed normalization procedure provided in Appendix I.1. The full GPT-5.2 total-score table is provided in Appendix I.2, and the per-baseline radar charts are provided in Figure 9.

Finding 1: HY and ViMoGen show relatively strong LLM-based instruction-following performance among the evaluated baselines. This advantage is better explained by design–task alignment. HY combines DiT-based flow matching, large-scale motion pretraining, high-quality fine-tuning, and instruction-alignment training with human feedback and reward models, which strengthens both motion coverage and text-motion alignment. ViMoGen transfers video-generation and MoCap priors into a flow-matching diffusion Transformer through semantically diverse data and gated multimodal conditioning, which benefits compositional and interaction-oriented generalization. In contrast, other recent baselines mainly focus on real-time control, editing, structured trajectory control, discrete motion-token modeling, or physical control. These objectives can improve motion quality, efficiency, diversity, or controllability, but they do not necessarily optimize pure text-conditioned following of long and fine-grained prompts. As shown in Table 19, we also observe a mild tendency that short-instruction subsets obtain higher total scores than long-instruction subsets in many settings, suggesting that longer temporal or compositional instructions may introduce additional challenges; however, this trend is treated as a supporting observation rather than a standalone conclusion.

Finding 2: Dimension-wise LLM scores provide a more diagnostic view than the total score alone. In **Action Completeness**, the Complexity curve is often more outward than the other categories for many baselines, suggesting relatively better coverage of required action components in Complexity instructions. In **Body-part Understanding**, the Interaction curve tends to be stronger, which is consistent with its frequent use of body-part relations, contacts, and side-specific movements. In **Multi-stage Order** and **Physical Plausibility**, the Accuracy curve is more prominent for several models, possibly because these prompts are usually short but contain explicit spatial or numerical constraints, making their outcomes easier to verify visually. Dynamics does not show a consistently

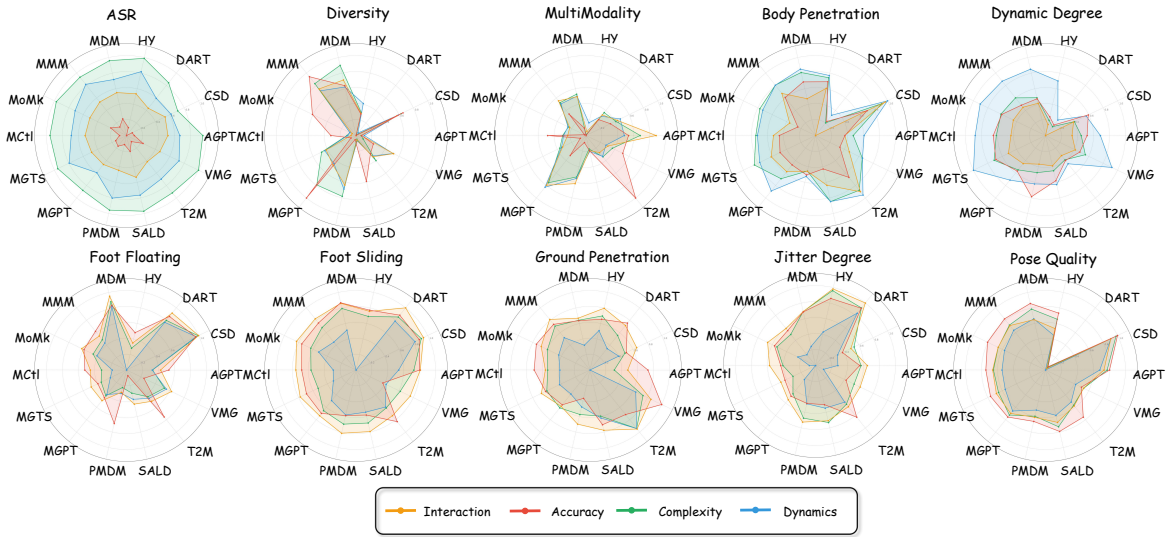

Figure 4: Multi-Factor Motion Evaluation radar charts by metrics, model abbreviations follow Figure 3. All metrics are direction-aware min–max normalized so that larger radar values consistently indicate better performance.

dominant pattern, reflecting the difficulty of evaluating motion style, rhythm, and intensity from rendered motions. Together, these results show that the five sub-dimensions offer a more diagnostic view of whether a model misses required actions, adds extra actions, violates the instructed order, misunderstands body parts, or produces visually implausible motions.

4.2.2 Multi-Factor Motion Evaluation Results

Figure 4 reports metric-wise radar charts for Multi-Factor Motion Evaluation, Figure 10 further provides per-baseline radar charts, where each chart summarizes one model across all multi-factor metrics. Detailed numerical results are provided in Appendix I.6 from Table 21 to Table 30. For visualization, all metrics are min–max normalized. The full normalization procedure is provided in Appendix I.1 and Appendix E.

Finding 3: Multi-factor metrics reveal complementary strengths across semantic preservation, motion variation, and physical quality. For **semantic preservation**, ASR shows that atomic-action preservation is not dominated by a single baseline. Models with stronger language–motion alignment or richer semantic priors, such as HY, ViMoGen, and AvatarGPT, are more likely to preserve key actions in the prompt.

For **motion variation**, Multimodality and Diversity capture different forms of generative variability. MotionGPT and AvatarGPT show stronger Multimodality, suggesting better same-prompt variation, while MMM, MDM, and MotionGPT achieve stronger Diversity, indicating broader dataset-level motion coverage. This difference may come from their generation mechanisms: autoregressive motion-token generation benefits diverse sampling under the same prompt, whereas masked-token prediction, diffusion stochasticity, or language-model-style motion sampling can better expand the overall motion distribution.

For **physical quality**, different models show advantages on different physical metrics. HY and DART show stronger Jitter Degree-related stability, T2M and ViMoGen produce fewer ground-penetration artifacts, and CLoSD performs well in reducing body penetration. These advantages may respectively come from continuous flow-based generation or autoregressive motion-primitive control, learned body–ground motion priors, and closed-loop physics control.

Overall, no single model consistently dominates all multi-factor metrics. This suggests that semantic preservation, motion variation, and physical plausibility are complementary dimensions, promoted by different architectural choices and training objectives.

4.2.3 Fine-Grained Accuracy Evaluation Results

The fine-grained accuracy results for whole-body and body-part motion are shown in Figure 5 and Table 20.

Finding 4: Fine-grained control remains difficult without stronger structured conditioning. For Root Rotation, HY achieves the best perfor-

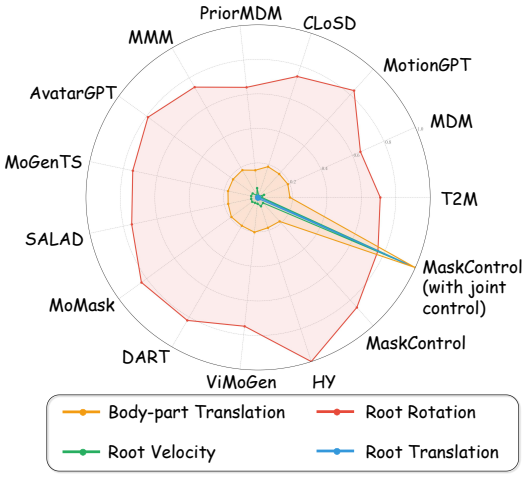

Figure 5: Fine-grained Accuracy Evaluation

mance, although its advantage over other baselines is limited. MaskControl with JointControl shows a clear advantage on Root Velocity, Root Translation, and Body-part Translation, but this setting uses an auxiliary joint-trajectory condition that is not required by text-only baselines. Because T2MBench uses text-only inputs, this trajectory condition is obtained offline from the same instruction through a fixed prompt-to-trajectory parsing pipeline, and therefore does not introduce additional semantic information beyond the text. Thus, JointControl results should be interpreted as controllability under stronger structured conditioning rather than a strictly apples-to-apples comparison with pure text-conditioned generators.

Although MaskControl with JointControl achieves lower trajectory-related errors, it often produces implausible motions when forced to match target positions, we provide an unnatural example in Appendix I.7. This is likely because MaskControl performs test-time gradient optimization over discrete token logits or embeddings to minimize the joint-trajectory control loss (Pinyoanuntapong et al., 2025), which can prioritize trajectory adherence over the generative motion prior. When physical plausibility is insufficiently constrained, the model may trade motion realism for tighter target matching. Under the text-only setting, MaskControl shows no clear advantage over other baselines, and its overall effectiveness on fine-grained instruction following remains limited.

5 Conclusion

The results of T2MBench show that current T2M models still struggle to consistently follow diverse language instructions while maintaining physical plausibility and fine-grained controllability. These gaps point to a promising direction for future T2M models: developing more instruction-grounded, constraint-aware, and physics-regularized generation frameworks. First, to improve instruction following under diverse language instructions, future models may need stronger instruction-level text-motion alignment, supported by richer semantic priors from high-quality motion captions, video-generation models, MoCap data, and large-scale flow-matching or diffusion Transformer architectures. Second, to improve fine-grained controllability, language instructions can be translated into intermediate structured constraints, such as root motion, body-part relations, contact states, and spatial targets, and injected through body-part-aware attention or constraint-aware generation modules. Third, to maintain physical plausibility, these semantic and structural control mechanisms should be regularized by contact-aware and body-aware physical priors, reducing artifacts such as foot sliding, jitter, ground penetration, and body penetration while preserving the learned motion prior.

To support reproducible analysis and future comparison, we release the instruction suite in the anonymous repository and have curated generated outputs for all prompts from the evaluated models. These curated outputs provide a reusable reference set for reproduction, comparison, error analysis, human evaluation, and future metric development, reducing the burden of rerunning baselines. For each motion, we provide `.npy` sequences, AMASS-compatible `.npz` files, skeleton videos, and color-rendered SMPL videos for metric computation, visualization, and motion-text inspection. The instruction suite and representative generated examples are available at `https://anonymous.4open.science/status/T2MBench-review-7078`. The full generated outputs will be released upon acceptance, subject to the usage terms of the original models and datasets. Overall, T2MBench documents the benchmark results while providing a reusable resource for evaluating future T2M models under diverse language instructions.

Limitations

Although T2MBench provides a diverse instruction suite and a multi-dimensional evaluation framework, it still has several limitations. First, although our prompts cover Dynamics, Complex-

ity, Interaction, and Accuracy, they do not exhaust all motion-language instructions; scene-dependent actions, multi-person interactions, object-specific manipulation, and long-horizon activities remain under-explored. Second, our results cover 14 representative T2M models and should be viewed as a reusable evaluation resource rather than a complete ranking of all possible systems. Third, although the LLM-based evaluator is validated against human preferences, the validation set covers only a subset of prompts and models, and the evaluator may still be affected by rendering choices, camera settings, body-part color coding, model-specific LLM biases, and the visual reasoning ability of the selected LLM. Fourth, because LLMs are used in instruction construction, visual evaluation, and structured target extraction, residual biases may remain despite manual filtering, human validation, and target verification. Finally, the fine-grained accuracy evaluation focuses on measurable root-level and body-part-level targets, leaving subtle style control, contact-rich interaction, and object-aware manipulation for future work. Auxiliary-control settings such as MaskControl with JointControl also use stronger structured conditions than pure text-only baselines and should be interpreted separately.

Ethical Considerations

T2MBench is designed as a diagnostic benchmark for evaluating text-to-motion generation under diverse language instructions. The benchmark instructions describe synthetic human motions and do not contain personal identities, private conversations, biometric identifiers, or real user records. Existing motion-language datasets and lexical resources are used only as references for motion verbs, body-part expressions, action styles, and semantic categories, while the final instruction suite is newly generated and manually filtered.

The benchmark uses large language models in three stages: instruction construction, visual motion evaluation, and structured target extraction for fine-grained accuracy evaluation. Although we apply manual filtering, human preference validation, and target verification, LLM-generated or LLM-judged content may still reflect residual biases or visual reasoning errors. Therefore, the reported results should be interpreted as diagnostic signals rather than absolute judgments of model quality.

For human preference validation, raters evaluate anonymized rendered motions without seeing model identities. We report aggregated agreement statistics and do not release personally identifying information about individual raters. The released materials are intended to support reproducible research, comparison, and error analysis, rather than deployment-oriented ranking or safety certification. Each rater received 50 RMB as compensation for completing the annotation task.

Generated human motions may be used in animation, VR/AR, embodied AI, and human–computer interaction. However, the motions evaluated in this work are synthetic and may contain physical artifacts such as foot sliding, jitter, ground penetration, or unnatural body configurations. They should not be directly treated as safe demonstrations for real human imitation, robotics execution, or physically constrained applications without additional safety checks. Users of the released data and generated motions should also comply with the licenses and usage terms of the original datasets, baseline models, and rendering tools involved in reproduction.

We used AI assistants for language polishing during the preparation of this work. All generated suggestions were manually checked and revised by the authors. The scientific claims, experimental design, analysis, and final writing remain the responsibility of the authors.

References

Bakhyt Bakiyev, Venelin Kovatchev, and Mubashir Ali. 2026. Pan-kk: A language resource for plagiarism detection in low-resource kazakh. *Procedia Computer Science*, 275:28–37. 7th International Conference on AI in Computational Linguistics.

Ralph Allan Bradley and Milton E. Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76:378–382.

Yulu Gan, Ligeng Zhu, Dandan Shan, Baifeng Shi, Hongxu Yin, Boris Ivanovic, Song Han, Trevor Darrell, Jitendra Malik, Marco Pavone, and Boyi Li. 2025. Foundationmotion: Auto-labeling and reasoning about spatial movement in videos. *Preprint*, arXiv:2512.10927.

Mingqi Gao, Yixin Liu, Xinyu Hu, Xiaojun Wan, Jonathan Bragg, and Arman Cohan. 2025. Re-evaluating automatic llm system ranking for alignment with human preference. *Preprint*, arXiv:2501.00560.

Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. 2023. Momask: Generative masked modeling of 3d human motions. *Preprint*, arXiv:2312.00063.

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 *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5142–5151.

Haonan Han, Xiangzuo Wu, Huan Liao, Zunnan Xu, Zhongyuan Hu, Ronghui Li, Yachao Zhang, and Xiu Li. 2024. Atom: Aligning text-to-motion model at event-level with gpt-4vision reward. *Preprint*, arXiv:2411.18654.

Yuanyuan He, Yongsen Pan, Wei Li, Jiali You, Jiawen Deng, and Fuji Ren. 2025. ECC: An emotion-cause conversation dataset for empathy response. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 6000–6017, Suzhou, China. Association for Computational Linguistics.

Seokhyeon Hong, Chaelin Kim, Serin Yoon, Junghyun Nam, Sihun Cha, and Junyong Noh. 2025. Salad: Skeleton-aware latent diffusion for text-driven motion generation and editing. *Preprint*, arXiv:2503.13836.

Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. 2023. Motiongpt: Human motion as a foreign language. *Preprint*, arXiv:2306.14795.

M. G. Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.

M. G. Kendall and B. Babington Smith. 1939. The problem of m rankings. *The Annals of Mathematical Statistics*, 10(3):275–287.

Aliasghar Khani, Arianna Rampini, Bruno Roy, Larasika Nadela, Noa Kaplan, Evan Atherton, Derek Cheung, and Jacky Bibliowicz. 2025. Motion generation: A survey of generative approaches and benchmarks. *Preprint*, arXiv:2507.05419.

Suhas S Kowshik, Abhishek Divekar, and Vijit Malik. 2024. Corrsynth – a correlated sampling method for diverse dataset generation from llms. *Preprint*, arXiv:2411.08553.

Zhuo Li, Mingshuang Luo, Ruibing Hou, Xin Zhao, Hao Liu, Hong Chang, Zimo Liu, and Chen Li. 2026. Morph: A motion-free physics optimization framework for human motion generation. *Preprint*, arXiv:2411.14951.

Han Liang, Wenqian Zhang, Wenxuan Li, Jingyi Yu, and Lan Xu. 2024. Intergen: Diffusion-based multi-human motion generation under complex interactions. *International Journal of Computer Vision*, 132(9):3463–3483.

Jing Lin, Ruisi Wang, Junzhe Lu, Ziqi Huang, Guorui Song, Ailing Zeng, Xian Liu, Chen Wei, Wanqi Yin, Qingping Sun, Zhongang Cai, Lei Yang, and Ziwei Liu. 2026. The quest for generalizable motion generation: Data, model, and evaluation. *Preprint*, arXiv:2510.26794.

Jing Lin, Ailing Zeng, Shunlin Lu, Yuanhao Cai, Ruimao Zhang, Haoqian Wang, and Lei Zhang. 2024. Motion-x: A large-scale 3d expressive whole-body human motion dataset. *Preprint*, arXiv:2307.00818.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. *Preprint*, arXiv:2303.16634.

Philip M. McCarthy and Scott Jarvis. 2010. Mtld, vocd-d, and hd-d: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior Research Methods*, 42(2):381–392.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics*.

Ekkasit Pinyoanuntapong, Muhammad Usama Saleem, Korrawe Karunratanakul, Pu Wang, Hongfei Xue, Chen Chen, Chuan Guo, Junli Cao, Jian Ren, and Sergey Tulyakov. 2025. Maskcontrol: Spatio-temporal control for masked motion synthesis. *Preprint*, arXiv:2410.10780.

Ekkasit Pinyoanuntapong, Pu Wang, Minwoo Lee, and Chen Chen. 2024. Mmm: Generative masked motion model. *Preprint*, arXiv:2312.03596.

Matthias Plappert, Christian Mandery, and Tamim Asfour. 2016. The kit motion-language dataset. *Big Data*, 4(4):236–252.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *Preprint*, arXiv:1908.10084.

Yonatan Shafir, Guy Tevet, Roy Kapon, and Amit H. Bermano. 2023. Human motion diffusion as a generative prior. *Preprint*, arXiv:2303.01418.

C. Spearman. 1904. The proof and measurement of association between two things. *American Journal of Psychology*, 15:88–103.

Kewei Sui, Anindita Ghosh, Inwoo Hwang, Bing Zhou, Jian Wang, and Chuan Guo. 2026. A survey on human interaction motion generation. *International Journal of Computer Vision*, 134(3).

Yixuan Tang, Yuanyuan Shi, Yiqun Sun, and Anthony Kum Hoe Tung. 2025. Uncovering the bigger picture: Comprehensive event understanding via diverse news retrieval. *Preprint*, arXiv:2508.19758.

Guy Tevet, Sigal Raab, Setareh Cohan, Daniele Reda, Zhengyi Luo, Xue Bin Peng, Amit H. Bermano, and Michiel van de Panne. 2024. Closd: Closing the loop between simulation and diffusion for multi-task character control. *Preprint*, arXiv:2410.03441.

Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H. Bermano. 2022. Human motion diffusion model. *Preprint*, arXiv:2209.14916.

Jordan Voas. 2023. What is the best automated metric for text to motion generation? Master's thesis, Department of Computer Science, UT Austin, Austin, TX, May.

Yuxin Wen, Qing Shuai, Di Kang, Jing Li, Cheng Wen, Yue Qian, Ningxin Jiao, Changhai Chen, Weijie Chen, Yiran Wang, Jinkun Guo, Dongyue An, Han Liu, Yanyu Tong, Chao Zhang, Qing Guo, Juan Chen, Qiao Zhang, Youyi Zhang, and 19 others. 2025. Hymotion 1.0: Scaling flow matching models for text-to-motion generation. *Preprint*, arXiv:2512.23464.

Yiming Xie, Varun Jampani, Lei Zhong, Deqing Sun, and Huaizu Jiang. 2024. Omnicontrol: Control any joint at any time for human motion generation. *Preprint*, arXiv:2310.08580.

Weihao Yuan, Weichao Shen, Yisheng He, Yuan Dong, Xiaodong Gu, Zilong Dong, Liefeng Bo, and Qixing Huang. 2024. Mogents: Motion generation based on spatial-temporal joint modeling. *Preprint*, arXiv:2409.17686.

Mingyuan Zhang, Huirong Li, Zhongang Cai, Jiawei Ren, Lei Yang, and Ziwei Liu. 2023. Finemogen: Fine-grained spatio-temporal motion generation and editing. *Preprint*, arXiv:2312.15004.

Kaifeng Zhao, Gen Li, and Siyu Tang. 2025. Dartcontrol: A diffusion-based autoregressive motion model for real-time text-driven motion control. *Preprint*, arXiv:2410.05260.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Zixiang Zhou, Yu Wan, and Baoyuan Wang. 2023. Avatargpt: All-in-one framework for motion understanding, planning, generation and beyond. *Preprint*, arXiv:2311.16468.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. *Preprint*, arXiv:1802.01886.

A Prompt Templates for Instruction Generation

This appendix summarizes the prompt templates used for LLM-based instruction generation. Instead of directly reusing text annotations from existing motion-language datasets, we use them only as references for motion verbs, body-part expressions, action description styles, and semantic coverage. All generated instructions are further manually filtered for physical executability and semantic evaluability.

A.1 Shared Generation Protocol

All subtype-specific prompts follow a shared generation protocol. The LLM is asked to act as a motion designer, and to generate natural-language instructions for text-to-motion evaluation. The generated instructions should be physically executable, visually verifiable, and sufficiently specific for evaluating whether a generated motion follows the text.

```
Shared generation protocol:

Role:
You are a professional motion designer.
Your task is to generate diverse, physically
     executable, and visually verifiable
    motion instructions for text-to-motion
    evaluation.

General requirements:
1. Describe concrete body motions rather
    than abstract intentions.
2. Focus on observable movements of the head,
     torso, arms, and legs.
3. Avoid facial expressions, fingers, toes,
    and overly fine-grained non-skeletal
    details.
4. Avoid physically impossible or highly
    unnatural body configurations.
5. Use clear spatial and temporal language
    when needed.
6. Ensure that each instruction can be
    visually evaluated from the generated
    motion.
7. Avoid duplicated, vague, overly abstract,
     or scene-dependent descriptions.

Output:
Generate multiple distinct motion
    instructions.
Each instruction should be a complete
    sentence or short paragraph.
Output only the generated instructions,
    without explanations.
```

A.2 Long and Short Variants

For subtypes that contain both long and short variants, we use the same subtype-specific generation

instruction and modify only the target length constraint and few-shot examples. Long variants encourage multi-stage descriptions, compositional actions, or richer spatial-temporal structure. Short variants use the same semantic category but require more concise instructions. Therefore, we do not repeat separate prompt templates for long and short variants in this appendix.

A.3 Instruction Distribution

Table 4 summarizes the distribution of T2MBench instructions across major categories and subcategories.

Table 4: Distribution of T2MBench instructions across categories and subcategories.

Category	Subcategories	Long	Short	Total
Dynamics	Dance, Explosive, Martial arts	125	150	275
Complexity	Multi-stage, Asymmetric, Balance	120	150	270
Interaction	Intent, Invisible-object interaction	100	100	200
Accuracy	Numerical, Whole-body, Body-part	–	280	280
Total	–	345	680	1,025

A.4 Dynamics

The Dynamics category focuses on motion style, intensity, and dynamic patterns. It includes dance, explosive motion, and martial-arts motion.

A.4.1 Dance

```
Subtype: Dance

Goal:
Generate realistic and diverse dance-related
     motion instructions.

Subtype-specific requirements:
1. Describe the mechanics of the dance
    motion rather than only naming the dance
     style.
2. Include coordinated movements of the
    torso, arms, and legs.
3. The motion should be aesthetically
    plausible and physically executable.
4. Avoid vague descriptions such as "dance
    beautifully" or "move gracefully"
    without concrete body mechanics.

Example style:
Steps rhythmically side to side while
    snapping fingers.
Slides backward smoothly performing a
    moonwalk motion.
```

A.4.2 Explosive Motion

```
Subtype: Explosive Motion

Goal:
Generate motion instructions that emphasize
    rapid acceleration, sudden body movement
    , or strong dynamic changes.

Subtype-specific requirements:
1. Include a clear preparatory posture and a
     rapid movement phase.
2. Describe how the torso, arms, and legs
    contribute to the explosive action.
3. Ensure that the motion remains physically
     executable.
4. Avoid abstract phrases such as "generate
    power" unless accompanied by observable
    body movement.

Example style:
Squats deep then explodes upward into a high
     jump.
Crouches low and sprints forward with sudden
     speed.
```

A.4.3 Martial Arts

```
Subtype: Martial Arts

Goal:
Generate diverse and challenging martial-
    arts-style motion instructions for
    robustness evaluation.

Subtype-specific requirements:
1. The motion should clearly reflect martial-
    arts patterns such as strikes, blocks,
    kicks, or defensive transitions.
2. The instruction may include multi-stage
    actions, but each stage must be
    physically executable.
3. Left and right body parts should be
    clearly distinguished when relevant.
4. Avoid imagined opponents, external
    objects, and scene-dependent
    descriptions.

Example style:
A man performs a high roundhouse kick with
    the right leg.
A figure assumes a fighting stance with
    fists raised high.
```

A.5 Complexity

The Complexity category focuses on compositional structure, asymmetric control, and balance-related motion.

A.5.1 Multi-stage Motion

```
Subtype: Multi-stage Motion

Goal:
Generate instructions containing multiple
    consecutive motion phases.

Subtype-specific requirements:
1. Each instruction should include a clear
    beginning, transition, and ending motion
    .
2. The temporal order of the phases should
    be visually observable.
3. Transitions should be natural and
    physically executable.
4. Avoid overly complex descriptions that
    cannot be visually verified.

Example style:
A person walks forward then turns right to
    stop.
The body slowly squats down with both arms
    hanging naturally. After pausing briefly
     at the lowest point, it gradually
    stands back up. Once fully upright, the
    right foot steps half a step forward,
    and the body then returns to the initial
     standing posture.
```

A.5.2 Asymmetric Motion

```
Subtype: Asymmetric Motion

Goal:
Generate instructions where the left and
    right body parts follow different motion
     logic.

Subtype-specific requirements:
1. Clearly describe the actions of both left
     and right body parts.
2. The two sides should not simply mirror
    each other.
3. If one side remains still, this should be
     explicitly stated.
4. The resulting motion should remain
    balanced and executable.

Example style:
Left arm circles clockwise while right arm
    punches forward.
Raises right hand high while kicking left
    leg sideways.
```

A.5.3 Balance

```
Subtype: Balance

Goal:
Generate instructions that require balance
    maintenance or recovery.

Subtype-specific requirements:
1. Include body tilting, single-leg support,
     recovery, or stabilizing arm movement.
2. The instruction should make the balance
    challenge visually observable.
3. Avoid impossible balance poses or
    external environment dependencies.

Example style:
Stands on one leg with arms out to stabilize.

A person walks slowly on a narrow imaginary
    tightrope.
```

A.6 Interaction

The Interaction category focuses on emotion-driven motion, interaction intent, and invisible-object interaction.

A.6.1 Intent and Emotion

```
Subtype: Intent and Emotion

Goal:
Generate motion instructions that express
    emotion or interaction intent through
    body movement.

Subtype-specific requirements:
1. Express the emotion through observable
    posture and body motion.
2. Do not rely on facial expressions.
3. Describe how the head, torso, arms, or
    legs convey the intended state.
4. Avoid purely abstract emotional labels
    without concrete movement.

Example style:
Confidence: The torso is held upright with
    the chest pushed forward, the arms are
    placed firmly on the hips with elbows
    pointing outward, the legs are
    positioned wide apart in a stable stance
    , the head is held high with the chin
    tilted upward, and the body expresses a
    sense of pride and self-assurance.
```

A.6.2 Invisible-object Interaction

```
Subtype: Invisible-object Interaction

Goal:
Generate instructions where the person
    mimics interacting with an invisible
    object.

Subtype-specific requirements:
1. The object should be explicitly described
     as invisible or imagined.
2. The interaction should be inferable from
    body movement.
3. Describe how the arms, torso, and legs
    support the interaction.
```

```
4. Avoid relying on external scene context.

Example style:
The arms extend forward as if pushing an
    invisible wall.
The person extends both arms forward and
    slightly upward to reach for an
    invisible rope.
```

A.7 Accuracy

The Accuracy category focuses on precise spatial, numerical, whole-body, and body-part constraints.

A.7.1 Numerical Constraint

```
Subtype: Numerical Constraint

Goal:
Generate instructions with explicit
    numerical, frequency, angle, or count-
    related constraints.

Subtype-specific requirements:
1. Include a count, angle, repetition,
    duration, or measurable spatial relation
    .
2. The numerical requirement should be
    visually checkable from the generated
    motion.
3. Avoid numerical constraints that are
    impossible to verify visually.

Example style:
The person raises both arms above the head
    three times, then lowers them to the
    sides and returns to a neutral standing
    posture.
```

A.7.2 Whole-body Accuracy

```
Subtype: Whole-body Accuracy

Goal:
Generate instructions that require
    coordinated whole-body spatial accuracy.

Subtype-specific requirements:
1. Describe the relationship between torso
    orientation, leg placement, and arm
    movement.
2. The instruction should require global
    body coordination rather than an
    isolated limb motion.
3. Spatial directions should be explicit and
     visually verifiable.

Example style:
Swivels the entire body to the right
    reaching an angle of 75 degrees.
Executes a wide turn to the left covering
    exactly 100 degrees of rotation.
```

A.7.3 Body-part Accuracy

```
Subtype: Body-part Accuracy

Goal:
Generate instructions that require precise
    control of specific body parts.

Subtype-specific requirements:
1. Explicitly name the target body part.
2. Describe how the target body part moves
    relative to the rest of the body.
3. Avoid overly fine-grained finger or toe
    descriptions.
4. The correctness of the target body-part
    motion should be visually checkable.

Example style:
Rotates head left to right keeping chin
    exactly 15cm above the shoulders.
Nods head fast bringing chin to within 8cm
    of the upper chest.
```

B Diversity Analysis Details

B.1 Compared Resources and Normalization

We compare T2MBench with representative text-to-motion language resources, including MotionPrefer, HumanML3D, KIT-ML, and ViMoGen-228k. These resources cover different forms of motion-language data: HumanML3D and KIT-ML provide conventional motion captions, MotionPrefer provides event-level prompts, and ViMoGen-228k contains large-scale prompt-oriented motion descriptions. Motion-X is used as a reference resource during instruction construction but is excluded from the quantitative diversity comparison. In our accessible version, its text files are dominated by dense pose-level or frame-level descriptions rather than motion-level captions, making direct comparison with the other resources less fair.

Since the compared resources use different raw text formats, we normalize them into a unified JSONL format before computing diversity metrics. Each line in the normalized file corresponds to one motion-level text description or one language instruction. For HumanML3D and KIT-ML, we retain only the natural-language caption and store POS tags and timestamps as metadata. For MotionPrefer and T2MBench, each text file is treated as one prompt, with category information preserved when available. For ViMoGen-228k, we use only motion-level annotations in the main analysis, while video-level long descriptions are retained separately but excluded from the main statistics. This normalization ensures that all met-

Dataset	#Units	#Texts	T/U	Avg. Len	Med. Len	Vocab
MPref	3,198	3,198	1.00	13.48	13	1,294
HML3D	14,616	43,692	2.99	12.39	10	6,703
KIT	3,009	6,353	2.11	8.44	7	1,591
T2MBench	1,025	1,025	1.00	25.59	11	1,672
ViGen	103,689	103,689	1.00	10.72	9	10,832

Table 5: Basic statistics of the compared text-to-motion language resources after normalization. #Units denotes motion samples or prompt units, #Texts denotes textual descriptions or prompts, and T/U denotes texts per unit. MPref, HML3D, KIT, and ViGen denote MotionPrefer, HumanML3D, KIT-ML, and ViMoGen-228K, respectively.

rics are computed over comparable motion-level text units.

Table 5 reports the basic statistics after normalization. ViGen and HumanML3D contain the largest numbers of textual descriptions, while T2MBench contains 1,025 instructions and is the smallest resource in terms of raw text count. This is expected because T2MBench is designed as a curated evaluation benchmark rather than a large-scale training corpus. Although T2MBench contains fewer texts, its average length is 25.59 words, higher than all compared resources. Therefore, we further conduct length-controlled validation in Appendix B.4 to examine whether the observed diversity is caused by longer prompts.

B.2 General Linguistic and Semantic Diversity Metrics

This appendix describes the implementation details of the general linguistic and semantic diversity metrics reported in Table 2. To ensure a fair comparison across datasets with substantially different sizes, we compute all metrics under repeated balanced sampling. For each run, we randomly sample 1,000 texts from each dataset using one of ten fixed random seeds, $\{0, 1, \ldots, 9\}$. All metrics are computed on the same sampled texts within each run. We then report the mean and standard deviation over the ten runs.

All texts are lowercased before tokenization. We tokenize each text by extracting alphabetic words, contractions, and numeric tokens. This keeps motion-related words such as *left*, *arm*, and *forward*, as well as numerical constraints such as *90* or *three*. Only motion-level or prompt-level texts are used in this analysis.

B.2.1 MTLD

MTLD measures lexical diversity by estimating how long a token sequence can maintain a sufficiently high type-token ratio before becoming lexically saturated. Given a token sequence $X = (x_1, x_2, \ldots, x_N)$, we sequentially scan the tokens and maintain the type-token ratio of the current segment:

$$\mathrm{TTR}_{1:t} = \frac{|\mathrm{Type}(x_1, \ldots, x_t)|}{t}.$$

Following the standard MTLD setting, we use a threshold $\tau = 0.72$. Whenever the segment-level TTR falls below or reaches τ, the current segment is counted as one full factor, and the segment statistics are reset.

If the sequence ends before the final segment reaches the threshold, we add a partial factor:

$$f_{\mathrm{partial}} = \frac{1 - \mathrm{TTR}_{\mathrm{rem}}}{1 - \tau},$$

where $\mathrm{TTR}_{\mathrm{rem}}$ is the TTR of the remaining unfinished segment. Let $F_{\rightarrow}$ denote the number of full and partial factors obtained by scanning the sequence from left to right. The forward MTLD score is:

$$\mathrm{MTLD}_{\rightarrow} = \frac{N}{F_{\rightarrow}}.$$

We also compute MTLD in the reverse direction. Let $F_{\leftarrow}$ denote the factor count obtained from the reversed token sequence. The reverse MTLD score is:

$$\mathrm{MTLD}_{\leftarrow} = \frac{N}{F_{\leftarrow}}.$$

The final MTLD score is the average of the forward and reverse scores:

$$\mathrm{MTLD} = \frac{1}{2}\left(\mathrm{MTLD}_{\rightarrow} + \mathrm{MTLD}_{\leftarrow}\right).$$

In our dataset-level implementation, each run first samples 1,000 texts from each dataset. The sampled texts are shuffled using the same random seed and then concatenated into a single token sequence before computing MTLD. This keeps the MTLD computation consistent with the repeated balanced sampling protocol used for the other metrics. A higher MTLD value indicates stronger lexical diversity.

B.2.2 Self-BLEU

Self-BLEU measures surface-level repetition within a set of instructions. Given a sampled instruction set $\mathcal{D} = \{x_i\}_{i=1}^{N}$, each instruction x_i is treated as the candidate sentence, while the remaining instructions $\mathcal{D} \setminus \{x_i\}$ are used as references. The Self-BLEU score is:

$$\text{SelfBLEU} = \frac{1}{N} \sum_{i=1}^{N} \text{BLEU}_4(x_i, \mathcal{D} \setminus \{x_i\}).$$

A higher Self-BLEU indicates stronger n-gram overlap among instructions, corresponding to higher surface-level repetition and lower textual diversity.

We report Self-BLEU-4. For each candidate instruction, BLEU-4 is computed using modified n-gram precision for $n = 1, \ldots, 4$:

$$\text{BLEU}_4 = \text{BP} \cdot \exp\left(\frac{1}{4} \sum_{n=1}^{4} \log p_n\right),$$

where BP is the brevity penalty and p_n is the smoothed modified n-gram precision.

For each n-gram g, the clipped count is computed as:

$$\text{clip}(g) = \min\left(c_i(g), \max_{j \neq i} c_j(g)\right),$$

where $c_i(g)$ denotes the count of g in the candidate x_i, and $c_j(g)$ denotes its count in a reference instruction x_j. We use add-one smoothing:

$$p_n = \frac{\sum_{g \in G_n(x_i)} \text{clip}(g) + 1}{\sum_{g \in G_n(x_i)} c_i(g) + 1},$$

where $G_n(x_i)$ denotes the set of n-grams in x_i.

The brevity penalty is computed using the reference length closest to the candidate length:

$$\text{BP} = \begin{cases} 1, & |x_i| > r_i, \\ \exp(1 - r_i/|x_i|), & |x_i| \leq r_i, \end{cases}$$

where r_i is the closest reference length to $|x_i|$ among $\mathcal{D} \setminus \{x_i\}$.

In each run, Self-BLEU is computed on the same 1,000 sampled texts used by the other metrics. After tokenization, each non-empty text is treated as the candidate, and all remaining non-empty sampled texts are used as references. In our data, this gives 999 references for each candidate, keeping the reference pool size balanced across datasets.

B.2.3 Occupied-Cluster Ratio

The occupied-cluster ratio is used as a compact diagnostic of semantic coverage in the sentence-embedding space. For each run, we encode the same sampled instructions using `sentence-transformers/all-MiniLM-L6-v2`. The embeddings are normalized during encoding. We then concatenate embeddings from all compared datasets and cluster them in a shared semantic space using MiniBatch KMeans.

The number of clusters is set to:

$$K = 30.$$

MiniBatch KMeans uses the same random seed as the corresponding sampling run, with `n_init` $= 10$ and batch size 4096. After clustering, each instruction is assigned a cluster label. For a dataset d, let C_d be the set of clusters that contain at least one instruction from d. The occupied-cluster ratio is:

$$\text{OccRatio}(d) = \frac{|C_d|}{K}.$$

A higher occupied-cluster ratio indicates that the dataset covers a broader portion of the shared semantic embedding space.

Unlike MTLD and Self-BLEU, the occupied-cluster ratio is not a standard off-the-shelf diversity metric. We use it as an embedding-space diagnostic to summarize how many semantic regions are covered by each dataset under the same clustering setup.

B.3 Occupied-Cluster Ratio Sensitivity

Since the occupied-cluster ratio depends on the number of semantic clusters, we further evaluate its robustness under different values of K. Specifically, we repeat the same balanced sampling protocol with $K \in \{20, 30, 40, 50\}$. For each K, we sample 1,000 texts from each dataset over ten fixed random seeds and compute the occupied-cluster ratio in the shared SBERT embedding space.

As shown in Table 6, the overall trend remains stable across different cluster granularities. T2MBench consistently achieves a high occupied-cluster ratio across all settings. Although ViGen obtains the highest value when $K = 20$, T2MBench achieves the highest occupied-cluster ratio for $K = 30$, $K = 40$, and $K = 50$. This suggests that the semantic coverage result is not an artifact of a single cluster number.

Dataset	$K = 20$	$K = 30$	$K = 40$	$K = 50$
MPref	0.9750±0.0354	0.9767±0.0274	0.9050±0.0258	0.9020±0.0382
HumanML3D	0.9950±0.0158	0.9700±0.0189	0.9575±0.0237	0.9300±0.0287
KIT	0.9000±0.0471	0.8900±0.0316	0.8525±0.0184	0.8420±0.0394
T2MBench	0.9950±0.0158	**0.9900**±0.0225	**0.9800**±0.0197	**0.9540**±0.0232
ViGen	**1.0000**±0.0000	0.9833±0.0236	0.9625±0.0243	0.9420±0.0239

Table 6: Sensitivity analysis of the occupied-cluster ratio under different numbers of semantic clusters. Each value is reported as mean ± standard deviation over 10 random seeds. MPref, HML3D, KIT, and ViGen denote MotionPrefer, HumanML3D, KIT-ML, and ViMoGen-228K, respectively.

B.4 Length-Controlled Validation

This appendix describes the length-bucket construction, sampling protocol, and metric computation procedures for the common-support length-controlled validation reported in Table 7. This validation is designed to examine whether the observed diversity of T2MBench can be explained merely by longer instruction length.

Length Buckets. We first compute the length of each text after tokenization and divide all texts into non-overlapping length buckets. The bucket statistics of the compared datasets are shown in Table 8. The long-text buckets have very limited common support across all datasets. In particular, AToM and KIT-ML contain almost no texts longer than 30 words, while T2MBench contains a large number of long instructions. Therefore, a strict length-controlled comparison over the full length range is not feasible without either oversampling or using extremely small long-text buckets.

Common-support Sampling Distribution. For strict length control, each dataset must provide the same number of texts in each selected length bucket. For a length bucket b, let $n_{d,b}$ be the number of available texts from dataset d. The common-support target count is defined as:

$$m_b = \min_d n_{d,b}.$$

Based on Table 8, the maximum common-support counts are 470 for the 1–10 bucket and 208 for the 11–20 bucket. For the longer buckets, the common support is extremely small: 3 for 21–30, 1 for 31–40, and 0 for 41+. We therefore restrict the main length-controlled validation to the stable short-to-medium common support, namely the 1–10 and 11–20 buckets. This results in the fixed sampling distribution shown in Table 9.

Sampling Protocol. For each run, we sample texts from every dataset according to the same fixed distribution:

$$\text{1–10 words: } 470, \qquad \text{11–20 words: } 208.$$

Thus, each dataset contributes $470 + 208 = 678$ texts in every run. We use ten fixed random seeds, $\{0, 1, \ldots, 9\}$, and report the mean and standard deviation over the ten runs. Sampling is performed without replacement. All metrics in Table 7 are computed on the same sampled texts within each run.

MTLD. MTLD is computed on the sampled texts to measure lexical diversity under the controlled length distribution. For each run and dataset, the sampled texts are shuffled with the current random seed and concatenated into a single token sequence. We then compute MTLD using the same implementation described in Appendix B.2, with threshold $\tau = 0.72$ and forward–reverse averaging. A higher MTLD indicates stronger lexical diversity.

Self-BLEU. Self-BLEU measures surface-level repetition among the sampled instructions. For each dataset and run, each sampled instruction is treated as the candidate sentence, and the remaining 677 sampled instructions from the same dataset are used as references. We report Self-BLEU-4 using the same smoothed BLEU implementation described in Appendix B.2. A lower Self-BLEU indicates less n-gram overlap and lower template-like repetition.

Pairwise Embedding Distance. Pairwise embedding distance measures global semantic dispersion in the sampled instruction set. We encode each sampled instruction using `sentence-transformers/all-MiniLM-L6-v2` and normalize the embeddings. Given normalized sentence embeddings $\{e_i\}_{i=1}^{N}$, the average pairwise cosine distance is:

$$\text{PairwiseDist} = \frac{2}{N(N-1)} \sum_{i<j} \left(1 - \cos(e_i, e_j)\right).$$

Since the embeddings are normalized, the cosine similarity is computed as the dot product between embeddings. A higher pairwise distance indicates that the instructions are more globally dispersed in the semantic embedding space.

Nearest-neighbor Distance. Nearest-neighbor distance measures local semantic redundancy. For each instruction, we find its closest neighbor

Dataset	MTLD↑	Self-BLEU↓	Pairwise Dist.↑	NND↑	Occ.↑	Avg. Types↑
MPref	34.53±0.57	0.5891±0.0044	0.5908±0.0036	0.1803±0.0024	0.9467±0.0322	1.59±0.03
HML3D	41.09±1.89	0.5335±0.0089	0.6391±0.0066	0.2173±0.0069	0.9267±0.0344	1.22±0.04
KIT	33.36±1.09	0.6826±0.0112	0.5674±0.0091	0.1273±0.0051	0.8600±0.0562	1.18±0.03
T2MBench	**94.70**±1.78	**0.4296**±0.0000	**0.7380**±0.0000	**0.2982**±0.0000	**0.9633**±0.0189	**1.60**±0.00
ViGen	51.50±2.71	0.5276±0.0087	0.7086±0.0065	0.2595±0.0047	0.9367±0.0331	1.12±0.06

Table 7: Common-support length-controlled diversity validation. Each dataset is sampled with the same short-to-medium length distribution, using 470 texts from the 1–10 word bucket and 208 texts from the 11–20 word bucket in each run. Values are reported as mean ± standard deviation over 10 random seeds. MPref, HML3D, KIT, and ViGen denote MotionPrefer, HumanML3D, KIT-ML, and ViMoGen-228K, respectively. Detailed sampling and metric definitions are provided in Appendix B.4.

Dataset	1–10	11–20	21–30	31–40	41+
AToM	883	2,103	211	1	0
HumanML3D	22,065	16,567	3,867	918	275
KIT-ML	4,894	1,320	132	7	0
T2MBench	470	208	3	40	304
ViGen	61,454	33,568	6,681	1,547	439

Table 8: Length-bucket statistics of the compared datasets. Lengths are computed using the same tokenizer as in the diversity analysis.

Bucket	Target Count	Used
1–10	470	Yes
11–20	208	Yes
21–30	3	No
31–40	1	No
41+	0	No

Table 9: Common-support target counts for length-controlled validation. The main validation uses the stable 1–10 and 11–20 buckets, resulting in 678 texts per dataset in each run. The longer buckets are excluded because their common-support counts are too small or zero.

within the same sampled dataset in the sentence-embedding space. The average nearest-neighbor distance is:

$$\mathrm{NND} = \frac{1}{N}\sum_{i=1}^{N} \min_{j\neq i} \left(1 - \cos(e_i, e_j)\right).$$

A higher NND indicates that each instruction is farther from its most similar neighbor, suggesting lower local semantic redundancy.

Occupied-cluster Ratio. The occupied-cluster ratio measures semantic cluster coverage under the controlled length distribution. For each run, we combine the embeddings of all sampled datasets and cluster them in a shared embedding space using MiniBatch KMeans with $K = 30$, `n_init` $= 10$, and the same random seed as the sampling run. For a dataset d, let C_d denote the set of clusters that contain at least one instruction from d. The occupied-cluster ratio is:

$$\mathrm{OccRatio}(d) = \frac{|C_d|}{K}.$$

A higher occupied-cluster ratio indicates that the sampled instructions cover a broader range of semantic clusters.

Average Constraint Types. Average constraint types measures the density of motion-oriented constraints under the same length-controlled sampling. For each sampled instruction, we count how many of the six constraint categories are detected: body-part, spatial, temporal, simultaneous, interaction, and numerical constraints. The average number of detected constraint types is then computed over the sampled texts. The lexicons and matching rules are the same as those described in Appendix C. Unlike the constraint ratios, this metric is not a percentage. It indicates how many different types of motion-oriented constraints are contained in each instruction on average.

Interpretation. This validation should be interpreted as a conservative robustness check. Because the stable common support is restricted to the short-to-medium length range, it does not replace the full diversity analysis. Instead, it tests whether T2MBench still preserves strong diversity after removing its long-instruction advantage.

C Motion-oriented Constraint Metrics

This appendix describes the implementation details of the motion-oriented constraint metrics reported in Table 11. These metrics are task-specific diagnostic statistics designed for text-to-motion evaluation. They are not intended as universal linguistic diversity metrics. Instead, they quantify

Constraint	Purpose	Lexicon
Body-part	Detects explicit references to body parts for fine-grained body control.	*head*, *neck*, *shoulder*, *arm*, *elbow*, *wrist*, *hand*, *torso*, *waist*, *hip*, *leg*, *knee*, *ankle*, *foot*
Spatial	Detects spatial directions, positions, and movement orientations.	*left*, *right*, *forward*, *backward*, *upward*, *downward*, *front*, *behind*, *around*, *clockwise*, *counterclockwise*, *diagonal*, *side*, *toward*, *away*
Temporal	Detects temporal ordering and multi-stage action structure.	*first*, *then*, *after*, *before*, *finally*, *followed by*, *next*, *while*, *during*
Simultaneous	Detects instructions where multiple actions occur at the same time.	*while*, *simultaneously*, *at the same time*, *meanwhile*, *as*
Interaction	Detects interaction-related actions or object-related expressions.	*push*, *pull*, *lift*, *pick*, *carry*, *hold*, *throw*, *catch*, *touch*, *avoid*, *follow*, *mimic*, *interact*, *object*, *box*, *ball*, *rope*, *wall*, *chair*
Numerical	Detects numerical, frequency-related, or measurable constraints.	*one*, *two*, *three*, *four*, *five*, *twice*, *times*, *degree*, *meter*, *step*, *second*, *half*, *90*, *180*, *360*, and standalone digit tokens

Table 10: Lexicons used for motion-oriented constraint detection. A text is counted as containing a constraint type if it matches at least one term or phrase from the corresponding lexicon.

whether a language instruction contains motion-control requirements that are directly relevant to text-to-motion generation.

Scope and Motivation. General linguistic metrics such as lexical diversity or surface-level repetition cannot fully characterize whether an instruction is challenging for text-to-motion generation. For example, two prompts may have different lexical forms but contain similar motion requirements, while another prompt may explicitly require body-part control, temporal ordering, or simultaneous actions. Therefore, we introduce motion-oriented constraint statistics as complementary diagnostics. The selected constraint categories correspond to common motion-control requirements, including body-part control, spatial direction, temporal ordering, simultaneous action composition, interaction-related motion, and numerical or measurable constraints. The same fixed lexicons and matching rules are applied uniformly to all compared datasets.

Text Filtering. We use the normalized JSONL files from the main diversity analysis. Only motion-level or prompt-level texts are included. Specifically, we retain records whose `text_type` is either `motion` or `preference_prompt`, and exclude empty texts and video-level descriptions. The evaluated datasets include MotionPrefer, HumanML3D, KIT-ML, T2MBench, and ViMoGen-228k.

Tokenization and Matching. All texts are lowercased before matching. We tokenize each text by extracting alphabetic words, contractions, and numeric tokens. For single-word terms, we match them against the token set of each text. For multi-word phrases, such as *followed by* and *at the same time*, we perform phrase matching with word-boundary constraints. For the numerical constraint category, we additionally detect standalone digit tokens. The full lexicons used for each constraint category are listed in Table 10.

Constraint Ratio. Table 11 reports the full numerical values corresponding to Figure 2. The table is provided for reproducibility and detailed inspection, while the main paper visualizes the same results as a composite figure. For each constraint type c, let L_c denote its lexicon. We define:

$$I_c(x_i) = \begin{cases} 1, & \exists\, l \in L_c \text{ such that } l \in x_i, \\ 0, & \text{otherwise.} \end{cases}$$

For a dataset $D = \{x_i\}_{i=1}^{N}$, the ratio of constraint type c is:

$$\mathrm{Ratio}_c(D) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}_c(x_i).$$

These ratios are reported as percentages in Table 11. For example, a body-part ratio of 55.61% means that 55.61% of the texts contain at least one body-part term from the corresponding lexicon.

Average Constraint Types. A text may contain multiple constraint types. For each text x_i, we count the number of constraint categories detected in the text:

$$T(x_i) = \sum_{c \in \mathcal{C}} \mathbb{I}_c(x_i),$$

Dataset	Body-part↑	Spatial↑	Temporal↑	Simul.↑	Interact.↑	Numeric↑	Avg. Types↑
MotionPrefer	33.46%	66.10%	13.32%	4.28%	5.85%	**53.69%**	1.77
HumanML3D	29.24%	61.07%	35.02%	10.09%	6.32%	16.52%	1.58
KIT-ML	13.21%	58.65%	12.31%	4.41%	1.57%	24.11%	1.14
T2MBench	**55.61%**	**68.20%**	**44.98%**	**36.20%**	**13.37%**	32.49%	**2.51**
ViMoGen-228k	24.15%	51.70%	26.70%	8.95%	6.83%	15.19%	1.34

Table 11: Motion-oriented constraint diversity across different text-to-motion datasets. Each constraint column reports the percentage of texts containing the corresponding constraint type. Body-part, spatial, temporal, simultaneous, interaction, and numerical constraints are detected by lexicon-based matching. Avg. Types denotes the average number of constraint types contained in each text.

where $\mathcal{C}$ denotes the set of all constraint categories. The average number of constraint types per instruction is:

$$\mathrm{AvgTypes}(D) = \frac{1}{N}\sum_{i=1}^{N} T(x_i).$$

Unlike the constraint ratios, $\mathrm{AvgTypes}$ is not a percentage. It measures how many different types of motion-oriented constraints are contained in each instruction on average.

Matched Example Inspection. During analysis, we also save matched examples for each dataset and constraint type. These examples are used for manual inspection of potential false positives and are not used to tune the reported scores. The reported statistics are computed using the fixed lexicons and matching rules described above.

D LLM-Based Evaluation Details

This appendix provides the implementation details of the LLM-based evaluation used in Section 3.4.1. It includes the motion rendering protocol, the color-coded spatial-temporal strip representation, the evaluation rubric, and the complete prompt template used for structured LLM judging.

D.1 Motion Rendering

For each generated motion, we render the joint trajectory using the SMPL body model with an off-screen renderer. To reduce representation differences across baselines, we apply a unified coordinate alignment and height normalization before rendering. All sequences are captured using a fixed virtual camera. To ensure that the full body motion remains visible throughout the sequence, we use an adaptive viewport determined by the spatial bounding box of the SMPL mesh. This setting avoids viewpoint drift while preserving the global movement pattern of the generated motion.

During evaluation, the rendered motion is treated as visual evidence of the generated human motion. The evaluator is instructed to focus only on visible body motion and ignore background, scene context, missing props, or environmental differences. This is important because T2M models usually generate body motion rather than full scene-object interactions.

D.2 Color-Coded Spatial-Temporal Strip

To make 3D body motion easier for the multimodal LLM to inspect, we convert each rendered sequence into a color-coded spatial-temporal strip, as illustrated in Figure 6. First, SMPL vertices are assigned fixed high-contrast colors according to body-part regions. The color labels are consistent across all frames and all evaluated models, allowing the evaluator to distinguish limbs, body sides, and local body-part movements.

We use the following body-part color groups: head/scalp, face, torso, left arm, right arm, left leg, right leg, left hand, and right hand. These labels are used only as visual aids and do not introduce additional semantic information beyond the rendered motion itself.

Second, rendered frames are sampled at a fixed stride and concatenated horizontally in temporal order. The leftmost region corresponds to the beginning of the motion, and the rightmost region corresponds to the end. This spatial-temporal strip allows the LLM to inspect pose changes, limb trajectories, body orientation, and stage transitions within a single image.

D.3 Evaluation Rubric

The LLM evaluator assesses each prompt-motion pair using five complementary criteria. These criteria are designed to capture both semantic instruction following and visual motion plausibility.

- **Extra Non-Instruction Actions** measures

Figure 6: Example of the color-coded spatial-temporal strip used in our LLM-based evaluation. The motion prompt is: "A person goes from standing to kneeling and then rises back to standing." Rendered frames are sampled at a fixed stride and concatenated horizontally from left to right in temporal order, where the leftmost frame corresponds to the beginning of the motion and the rightmost frame corresponds to the end. Body-part colors are fixed across all frames and all evaluated models: head/scalp = pure red, face = orange, torso = bright yellow, left arm = pure green, right arm = cyan, left leg = pure blue, right leg = bright purple, left hand = light green, and right hand = light gray. These colors are used only as visual aids to help the evaluator distinguish limbs, body sides, and local body-part movements.

whether the generated motion contains unintended actions that are not requested by the prompt. This criterion penalizes over-generation, where the model adds extra gestures or movements that may change the intended meaning.

- **Action Completeness** measures whether all key action components described in the prompt are visibly present in the generated motion. This criterion focuses on action recall and detects omitted or weakly expressed actions.

- **Multi-Stage Order Correctness** evaluates whether multiple actions are performed in the correct temporal order. This is especially important for prompts involving sequential structures such as "first", "then", or "finally".

- **Body-Part Understanding** measures whether the generated motion uses the correct body parts, sides, and movement directions specified by the prompt. This criterion targets fine-grained grounding of body-part language, such as left/right limbs and upper/lower body constraints.

- **Physical Plausibility** provides a coarse visual assessment of whether the motion is anatomically reasonable, smooth, and free from severe artifacts such as broken joints, teleportation, or unnatural discontinuities.

The final LLM-based score is computed as the sum of the five sub-scores. The evaluator also provides a categorical verdict and short auditable explanations for each evaluated motion.

D.4 Evaluation Prompt

The following prompt is used for LLM-based evaluation of motion-text alignment. It guides the model to analyze sampled video frames and output a structured assessment in JSON format.

```
You are a motion alignment evaluator.
     Compare sequential frames with the
     provided motion prompt. Because only
     provided frames are available, judge
     solely by body motion. Ignore scenery,
     props, missing equipment, or environment
      differences. Treat the prompt as a list
      of action keywords and constraints (e.g
     ., kick, pull, jump, push, rotate,
     crouch, left/right, direction, speed).
     Slowly scan every frame and concentrate
     on limb paths, joint angles, body
     orientation, tempo changes, and
     transitions.

The video is represented as a single long
     horizontal strip image. This strip is
     constructed by concatenating evenly
     sampled frames from left to right in
     strict temporal order:
* The leftmost region corresponds to the
     beginning of the motion and the
     rightmost region corresponds to the end.
```

```
When you analyze the motion, mentally scan
     the strip from left to right as if time
     is progressing, and focus on how the
     body pose and joint angles evolve along
     this horizontal axis.
The virtual camera is fixed in space. If the
      person becomes smaller, they are moving
      away; if larger, moving closer. At the
     initial state the person is facing the
     camera frontally.

Video Name: {video_name_escaped}
Motion Prompt Text: """{prompt_text_escaped
     }"""

Body-part color coding (for visual reference
      only):
(1) Head/scalp: pure red; (2) Face: orange;
     (3) Torso: bright yellow; (4) Left arm:
     pure green; (5) Right arm: cyan; (6)
     Left leg: pure blue; (7) Right leg:
     bright purple; (8) Left hand: light
     green; (9) Right hand: light gray.

============================================
 1) EXTRA NON-INSTRUCTION ACTIONS (10 points
     )
* 9-10: No obvious extra actions; motion
     stays on-instruction
* 5-8: Minor extra gestures/steps that do
     not dominate the sequence
* 0-4: Clear extra actions that change
     meaning or add major unintended motions

 2) ACTION COMPLETENESS (20 points)
* 18-20: All key action components appear
     clearly and sufficiently
* 10-17: Partial coverage; some components
     weak/short/missing
* 0-9: Most key actions missing,
     contradictory, or not recognizable

 3) MULTI-STAGE ORDER CORRECTNESS (10 points
     )
* 9-10: Correct order and clear stage
     boundaries
* 5-8: Order mostly correct but some stage
     confusion/overlap
* 0-4: Wrong order or stages not present

 4) BODY-PART UNDERSTANDING (10 points)
* 9-10: Correct body-part usage and
     orientation
* 5-8: Partially correct; occasional
     confusion
* 0-4: Mostly incorrect (e.g., wrong side/
     limb/direction)

 5) PHYSICAL PLAUSIBILITY(10 points)
* 9-10: Physically plausible, anatomically
     clean, and smooth/coherent throughout
* 5-8: Mostly acceptable with minor issues (
     small balance slips, jitter)
* 0-4: Major issues (teleporting, broken
     joints, jarring discontinuities)

============================================
 FINAL SCORING RULE

Compute: overall_score = sum of all 5 sub-
     scores (0-60)
Verdict: "aligned" (50-60), "partial"
     (30-49), "mismatch" (0-29)

============================================
OUTPUT FORMAT
Return ONLY the following raw JSON:
{
  "video_name": "{video_name_escaped}",
  "prompt_name": "string",
  "scores": {
    "extra_non_instruction_actions": int,
    "action_completeness": int,
    "multi_stage_order_correctness": int,
    "body_part_understanding": int,
    "physical_plausibility": int
  },
  "overall_score": int,
  "verdict": "aligned" | "partial" | "
     mismatch",
  "frame_observation": "<=200 chars",
  "prompt_overlap": "<=200 chars",
  "issues_found": "<=200 chars"
}
Important:
* Base judgment on visible evidence. Ignore
     background/props.
* Output plain JSON only. Do NOT wrap in
     markdown code blocks.
```

E Multi-Factor Motion Evaluation Details

This appendix provides detailed definitions and computation procedures for the Multi-Factor Motion Evaluation used in Section 3.4.2. The metrics are organized into three dimensions: Semantic Alignment, Generalizability, and Physical Quality. Table 12 summarizes the evaluation direction of each metric. Unless otherwise specified, all metrics are reported in their original direction. For visualization or radar-chart comparison, lower-is-better metrics are converted by inverse normalization.

Why not Matching Score, R-Precision, and MM Dist? Common text-motion alignment metrics such as Matching Score, R-Precision, and MM Dist rely on pretrained text and motion encoders to compute distances or retrieval accuracy in a shared embedding space. These metrics are effective for standard caption-style T2M evaluation, where the test prompts are close to the distributions used to train the encoders. However, T2MBench contains diverse instructions involving dynamics, multi-stage composition, interaction, spatial constraints, numerical requirements, and body-part relations. Such instructions can go beyond the caption-style descriptions seen by existing pre-

Dimension	Metric	Direction
Semantic Alignment	ASR	Higher is better
Generalizability	MM	Higher is better
Generalizability	Diversity	Higher is better
Physical Quality	JD	Lower is better
Physical Quality	GP	Lower is better
Physical Quality	FF	Lower is better
Physical Quality	FS	Lower is better
Physical Quality	DD	Higher is better
Physical Quality	PQ	Lower is better
Physical Quality	BP	Lower is better

Table 12: Summary of Multi-Factor Motion Evaluation metrics.

trained text–motion encoders. As a result, the metric values may reflect not only the quality of the generated motion, but also the encoder's own ability or limitation in representing these diverse instructions. This makes them less reliable as the primary diagnostic metrics for our setting. Therefore, we instead use ASR for semantic alignment, which compares the generated-motion caption with the original instruction at the atomic-action level, and report additional motion-level and physical metrics to provide complementary diagnostics.

Dimension 1: Semantic Alignment. Semantic Alignment measures whether the generated motion preserves the action-level semantics of the original prompt. We use Automatic Similarity Recall (ASR), which evaluates whether the relevant atomic actions in the ground-truth prompt can be recovered from the caption of the generated motion.

1. **Automatic Similarity Recall (ASR).** ASR measures whether the generated motion preserves the atomic actions described in the original input prompt. For each generated motion, we use the FoundationMotion captioner to produce a textual caption, which serves as the model-output prompt for ASR evaluation. The original prompt used to generate the motion is treated as the ground-truth prompt.

We first decompose the ground-truth prompt and the captioner-generated prompt into sets of atomic actions. Let

$$\mathcal{G}_i = \{g_{i,1}, g_{i,2}, \dots, g_{i,M_i}\}$$

denote the set of atomic actions extracted from the ground-truth prompt of the i-th sample, and let

$$\mathcal{C}_i = \{c_{i,1}, c_{i,2}, \dots, c_{i,L_i}\}$$

denote the set of atomic actions extracted from the FoundationMotion-generated caption. For each ground-truth atomic action $g_{i,j}$, we compute its maximum cosine similarity with all atomic actions in $\mathcal{C}_i$. The action is considered recalled if the maximum similarity exceeds a threshold $\tau = 0.5$. The ASR score for the i-th sample is defined as:

$$s_{i,j,k} = \mathrm{CosSim}\big(\phi(g_{i,j}), \phi(c_{i,k})\big).$$

$$\mathrm{ASR}_i = \frac{1}{M_i} \sum_{j=1}^{M_i} \mathbb{I}\left(\max_{1 \le k \le L_i} s_{i,j,k} > \tau \right).$$

where $\phi(\cdot)$ denotes the text embedding function, $\mathrm{CosSim}(\cdot,\cdot)$ denotes cosine similarity, and $\mathbb{I}(\cdot)$ is the indicator function, $s_{i,j,k}$ denotes the cosine similarity between the generated caption $g_{i,j}$ and the reference caption $c_{i,k}$..

The final ASR score is averaged over all N evaluated samples:

$$\mathrm{ASR} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{ASR}_i.$$

A higher ASR indicates that more ground-truth atomic actions can be recovered from the caption of the generated motion, suggesting better preservation of the action semantics specified by the original prompt.

Dimension 2: Generalizability. Generalizability evaluates whether a model can generate varied motions under the same instruction and across different instructions. We use two metrics: Multimodality and Diversity.

1. **Multimodality (MM).** For each instruction x_i, the model generates K motion samples. We randomly sample M pairs of different generated motions under the same instruction and compute:

$$\mathrm{MM} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{MM}_i,$$

$$\mathrm{MM}_i = \frac{1}{M} \sum_{m=1}^{M} d(A_i^{p_m}, A_i^{q_m}).$$

Here, $A_i^{p_m}$ and $A_i^{q_m}$ are two different generated motions conditioned on the same instruction x_i, with $p_m \neq q_m$. A higher MM indicates stronger one-to-many generation ability under the same text input.

2. **Diversity.** Diversity measures motion variation across different text instructions. We randomly

sample M pairs of generated motions conditioned on different instructions and compute:

$$\text{Div} = \frac{1}{M} \sum_{m=1}^{M} d\Big(A_{i_m}^{k_m}, A_{j_m}^{l_m}\Big).$$

Here, the two motions in each pair are sampled from different instructions, i.e., $i_m \neq j_m$. A higher Diversity score indicates broader dataset-level motion variation.

Dimension 3: Physical Quality. This dimension evaluates the realism and physical plausibility of the generated motions.

1. **Jitter Degree(JD)** measures the amount of jitter in the generated motion sequence. It is calculated by aggregating acceleration statistics from global and local joint motion. The calculation of jitter degree is as follows:

$$d_{t,j} = \left\|\mathbf{a}_{t,j}^{\text{global}}\right\|_2 + \left\|\mathbf{a}_{t,j}^{\text{local}}\right\|_2.$$

$$\text{JD} = \frac{1}{(T-2)J} \sum_{t=1}^{T-2} \sum_{j=1}^{J} d_{t,j}.$$

where $d_{t,j}$ denotes the sum of the global and local acceleration magnitudes of the j-th joint at the t-th frame. Specifically, $\mathbf{a}_{t,j}^{\text{global}} = \mathbf{v}_{t+1,j}^{\text{global}} - \mathbf{v}_{t,j}^{\text{global}}$ and $\mathbf{a}_{t,j}^{\text{local}} = \mathbf{v}_{t+1,j}^{\text{local}} - \mathbf{v}_{t,j}^{\text{local}}$ are the finite differences of the global and local velocities, respectively. Here, $\mathbf{v}_{t,j}$ denotes the velocity of the j-th joint at the t-th frame, T is the total number of frames, and J is the total number of joints. A smaller JD value indicates smoother motion.

2. **Ground Penetration(GP)** measures the extent to which parts of the motion sequence penetrate the ground. The calculation of ground penetration is as follows:

$$\text{GP} = \begin{cases} \frac{1}{|\mathcal{P}|} \sum_{(t,s)\in\mathcal{P}} |h_{t,s}|, & |\mathcal{P}| > 0, \\ 0, & |\mathcal{P}| = 0, \end{cases}$$

$$\mathcal{P} = \{(t,s) \mid h_{t,s} < -\delta\},$$

where $h_{t,s}$ denotes the height of foot joint s above the floor at frame t, and $\delta = 0.005$ is the tolerance threshold. The score averages the absolute penetration depth over foot joints that fall below the floor by more than δ; if no such foot joint exists, the score is set to zero.

3. **Foot Floating(FF)** measures the degree of foot floating during the contact phase. It detects abnormal foot floating using foot-contact states, foot velocity, root-relative foot velocity, and non-contact floating intervals. The formula for calculating floating degree is as follows:

$$\text{FF} = \frac{1}{T}\left(N_{\text{invalid}} + \frac{1}{2}\sum_r L_r + \sum_m L_m\right)$$

Where N_{invalid} is the number of invalid frames detected (based on foot velocity ratio $r_t = \frac{\|\mathbf{v}_t^{\text{rel}}\|_2}{\|\mathbf{v}_t^{\text{root}}\|_2+\epsilon}$ and contact state), L_r is the duration of floating intervals in the sequence, L_m is the duration of mass floating intervals, and T is the total number of frames. A smaller floating degree value indicates more reasonable foot-ground contact.

4. **Foot Sliding(FS)** measures the extent of foot sliding on the ground. It quantifies the horizontal movement of the foot when in contact with the ground. The formula for calculating foot sliding is as follows:

$$S^s = \frac{\sum_{t=1}^{T} \left\|\mathbf{v}_{t,\text{hori}}^s\right\|_2 c_t^s}{\sum_{t=1}^{T} c_t^s + \epsilon}, \quad s \in \{\text{left}, \text{right}\}.$$

$$\text{FS} = \frac{1}{2}\left(S^{\text{left}} + S^{\text{right}}\right).$$

where S^s denotes the foot sliding score of side $s \in \{\text{left}, \text{right}\}$. $\mathbf{v}_{t,\text{hori}}^s$ denotes the horizontal velocity of the corresponding foot on the ground plane at the t-th frame, and $c_t^s \in \{0, 1\}$ indicates whether the foot is in contact with the ground. $\epsilon = 10^{-6}$ is used for numerical stability. A smaller FS value indicates more stable foot contact and less foot sliding.

5. **Dynamic Degree(DD)** measures the dynamic nature of the generated motion. It is quantified by calculating the L_2 norm of the joint velocities, including both global dynamics and local dynamics after removing the global translation. The formula for dynamic degree is as follows:

$$d_{t,j} = \|\mathbf{v}_{t,j}^{\text{global}}\|_2 + \|\mathbf{v}_{t,j}^{\text{local}}\|_2.$$

$$\text{DD} = \frac{1}{(T-1)J} \sum_{t=1}^{T-1} \sum_{j=1}^{J} d_{t,j}.$$

where $d_{t,j}$ denotes the sum of the global and local velocity magnitudes of the j-th joint at frame t. Specifically, $\mathbf{v}_{t,j}^{\text{global}} = \mathbf{p}_{t+1,j} - \mathbf{p}_{t,j}$ and $\mathbf{v}_{t,j}^{\text{local}} =$

$\mathbf{p}^{\text{local}}_{t+1,j} - \mathbf{p}^{\text{local}}_{t,j}$ represent the global and local velocities of the j-th joint, respectively. The local velocity is computed from joint positions after removing the root translation. Here, T is the total number of frames, and J is the number of joints. A larger DD value indicates that the motion is more dynamic.

6. **Position Quality(PQ)** measures the plausibility and naturalness of the generated poses. It is quantified by evaluating the distance from the pose parameters to the pose manifold using a Neural Distance Field (NRDF) model. The formula for pose quality is as follows:

$$\text{PQ} = 10 \times \frac{1}{T} \sum_{t=1}^{T} d_{\text{NRDF}}(\mathbf{q}_t)$$

Where $d_{\text{NRDF}}(\mathbf{q}_t)$ represents the distance predicted by the NRDF model for the pose quaternion $\mathbf{q}_t$ at frame t (converted from the axis-angle $\mathbf{p}_t$), and T is the total number of frames. A lower PQ value indicates that the generated poses are closer to the realistic human-pose manifold.

7. **Body Penetration(BP)** measures the degree of penetration between different parts of the body. It is calculated by using the BVH (Bounding Volume Hierarchy) collision detection algorithm to detect collisions and overlaps between body parts. The formula for body penetration is as follows:

$$\text{BP} = \frac{1}{T} \sum_{t=1}^{T} \frac{N_t^{\text{collision}}}{N^{\text{triangles}}} \times 100\%$$

Where $N_t^{\text{collision}}$ represents the number of collision triangle pairs detected at frame t through BVH, $N^{\text{triangles}}$ is the total number of triangles in the SMPL model (a fixed value), and T is the total number of frames. A smaller body penetration value indicates less self-collision in the body.

F Fine-grained Accuracy Evaluation Details

This appendix provides the detailed target extraction protocol, notation, and metric computation procedures for the Fine-grained Accuracy Evaluation in Section 3.4.3. This evaluation contains two types of constraints: whole-body constraints and body-part constraints. Whole-body constraints are evaluated using Root Rotation, Root Velocity, and Root Translation, while body-part constraints are evaluated using Body-part Translation. All metrics are computed as errors against structured target specifications extracted from the original test instructions; therefore, lower values indicate better fine-grained controllability. Table 13 summarizes the evaluated subsets, metrics, and target types.

Subset	Metric	Error Type
Whole-body	Root Rot.	Yaw ↓
Whole-body	Root Vel.	Speed ↓
Whole-body	Root Trans.	Displacement ↓
Body-part	Body-part Trans.	Relative position ↓

Table 13: Summary of fine-grained accuracy metrics. All metrics are error-based, and lower values indicate better controllability.

Reference Target Extraction. For each fine-grained test instruction, we use GPT-5.2 to extract a structured target specification from the original prompt. Whole-body targets are stored in `root_move.json`, including the target type and its corresponding numerical value, such as target yaw angle, target directional speed, or target displacement. Body-part targets are stored in `body_part.json`, including the base joint, target joint, and target relative displacement. These structured files are used only as reference annotations for error computation and are not provided as inputs to the evaluated text-to-motion models. All extracted targets are manually checked to ensure consistency with the original instructions.

General Notation. For a generated motion clip with T frames, we index frames by $t \in \{0, \ldots, T-1\}$. The start frame is defined as:

$$t_0 = 0.$$

For body-part translation evaluation, we use the last N frames as the evaluation window:

$$\mathcal{W} = \{\max(0, T-N), \ldots, T-1\},$$

where $N = 30$ by default. If the motion length is shorter than N, all frames are used.

Whole-body Constraint Evaluation. Whole-body constraints evaluate whether the generated motion follows global root-level targets specified in the instruction. These constraints are designed to test the controllability of the overall body trajectory, movement speed, and facing direction. We recover the root position sequence $p_t \in \mathbb{R}^3$ and the root yaw angle $\psi_t \in \mathbb{R}$ from the generated motion. For models that output normalized HumanML/MDM-style features $x \in \mathbb{R}^{T \times 263}$, we first denormalize

the features using the dataset mean μ and standard deviation σ:

$$\tilde{x} = x \odot \sigma + \mu,$$

where $\odot$ denotes element-wise multiplication. Since the HumanML/MDM 263-dimensional representation only provides one degree of root rotation, we evaluate yaw rotation around the vertical axis and do not evaluate pitch or roll.

Root Rotation. Root Rotation measures whether the generated body turns by the target yaw angle. Given a target yaw rotation ψ^*, we compute the generated yaw change:

$$\Delta\psi = \mathrm{wrap}_{[-\pi,\pi)}(\psi_{T-1} - \psi_0),$$

where $\mathrm{wrap}_{[-\pi,\pi)}(\cdot)$ normalizes the angle to the range $[-\pi, \pi)$. We convert the generated yaw change and the target yaw into rotation matrices:

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}.$$

The root rotation error is measured by the Frobenius distance:

$$e_{\mathrm{rot}} = \left\| R_y(\Delta\psi) - R_y(\psi^*) \right\|_F.$$

A lower value indicates that the generated motion better satisfies the target turning angle.

Root Velocity. Root Velocity measures whether the generated body moves at the target speed along the specified direction. Given a target speed v^*, a unit direction u, and a duration d, we first estimate the per-frame root velocity:

$$v_t = (p_{t+1} - p_t) \cdot \mathrm{fps},$$

where fps is the frame rate. We set:

$$t_d = \min(T-1, \mathrm{round}(d \cdot \mathrm{fps})).$$

The realized directional speed is computed by projecting the velocity onto the target direction:

$$\hat{v} = \frac{1}{t_d} \sum_{t=0}^{t_d-1} \langle v_t, u \rangle.$$

The root velocity error is:

$$e_{\mathrm{vel}} = |\hat{v} - v^*|.$$

A lower value indicates that the generated motion better follows the target movement speed.

Root Translation. Root Translation measures whether the generated body reaches the target displacement. Given a target displacement $\Delta p^* \in \mathbb{R}^3$, we compute the generated root displacement as:

$$\Delta p = p_{T-1} - p_0.$$

The translation error is measured by the RMSE of the 3D root displacement vector:

$$e_{\mathrm{trans}} = \frac{1}{\sqrt{3}} \left\| \Delta\mathbf{p} - \Delta\mathbf{p}^* \right\|_2.$$

Here, $\Delta\mathbf{p}$ and $\Delta\mathbf{p}^*$ denote the generated and target 3D root displacement vectors, respectively. A lower value indicates that the generated motion more accurately reaches the target global displacement.

Body-part Constraint Evaluation. Body-part constraints evaluate whether the generated motion satisfies local relative-position requirements between two specified body parts. Unlike whole-body constraints, which focus on global root motion, body-part constraints test fine-grained local pose controllability. Each body-part instruction specifies a base joint b, a target joint g, and a target relative displacement $\Delta^* \in \mathbb{R}^3$. These constraints are evaluated using Body-part Translation.

Body-part Translation. Body-part Translation measures whether the target joint is placed at the specified relative position with respect to the base joint. Given the generated joint coordinates:

$$J \in \mathbb{R}^{T \times 22 \times 3},$$

we compute the relative displacement between the target joint and the base joint at each frame:

$$\Delta_t = J_{t,g} - J_{t,b}.$$

The per-frame error is:

$$e_t = \Delta_t - \Delta^*.$$

We evaluate this error over the last N frames $\mathcal{W}$, because the target body-part relation is expected to be satisfied in the final stable phase of the generated motion. The body-part translation error is:

$$e_{\mathrm{body}} = \sqrt{\frac{1}{|\mathcal{W}|} \sum_{t \in \mathcal{W}} \|e_t\|_2^2}.$$

A lower value indicates that the generated motion more accurately satisfies the specified local spatial relation between the two body parts.

Summary. In summary, whole-body constraints are evaluated using Root Rotation, Root Velocity, and Root Translation, which measure global orientation control, speed control, and displacement control, respectively. Body-part constraints are evaluated using Body-part Translation, which measures local relative-position control between body parts. All errors are computed against the structured targets extracted from the original prompts, and lower values indicate better fine-grained controllability.

G Human Preference Annotation Details

This appendix provides the details of the human preference annotation protocol used to validate the LLM-based evaluator. We report aggregated annotation statistics rather than individual-level raw annotations to avoid redundant records while keeping the validation process reproducible.

Annotation Protocol. For each selected prompt, human raters are shown the text instruction and three anonymized rendered motions generated by different text-to-motion models. The three motions are generated by HY, MotionGPT, and CLoSD, but the model identities are hidden from the raters. The display order of the three motions is randomly shuffled to reduce positional bias. Raters are asked to rank the three motions according to their overall prompt-following quality and motion plausibility. Specifically, they are instructed to consider whether the motion follows the action semantics of the prompt, whether key actions are complete, whether temporal order and body-part usage are correct, and whether the motion appears physically plausible. They are not asked to score the five LLM-evaluation sub-dimensions separately. Instead, each rater provides one overall preference ranking for the three motions under each prompt.

Rater Information. The annotation study involves 15 human raters aged between 22 and 28. All raters receive the same annotation instruction. No model identity is shown during annotation, and all preference judgments are collected on anonymized motion samples.

Recruitment, Compensation, and Consent. We recruited 15 raters from university students and research volunteers who had basic familiarity with visual motion understanding. The raters participated in a minimal-risk preference annotation task involving only anonymized rendered motion videos and text instructions. No sensitive personal information was collected. Before the annotation, raters were informed that their annotations would be used only for research validation of the LLM-based evaluator and that only aggregated statistics, rather than individual-level responses, would be reported. Participation was voluntary, and raters could stop the annotation if they felt uncomfortable with the task. Each rater received 50 RMB as compensation for completing the annotation task, which we consider appropriate for a short, minimal-risk preference-ranking study in the local context.

Metric	Value
Kendall's W	0.9076 ± 0.1589
Fleiss' κ	0.8390
Majority Margin	0.8628 ± 0.2554

Table 14: Human inter-rater agreement on preference rankings. Kendall's W is reported as mean $\pm$ standard deviation over 40 prompts. Majority Margin is reported as mean $\pm$ standard deviation over all prompt-level preference decisions.

Prompt Selection. We select 40 prompts from T2MBench for the human preference study. The selected prompts are balanced across the four major instruction categories, with 10 prompts from each of Dynamics, Complexity, Interaction, and Accuracy. The selection also covers all subtypes in the benchmark taxonomy, so that the human preference reference set includes different types of diverse language instructions.

Agreement Metrics. We compute three agreement statistics: Kendall's W, Fleiss' κ, and Majority Margin. As shown in Table 14, the human annotations show strong agreement across all three statistics. Kendall's W measures rank concordance among raters. Fleiss' κ measures chance-corrected agreement over categorical preference labels. Majority Margin measures the strength of the dominant preference label.

Kendall's W. For each prompt, we compute Kendall's W over the three generated motions ranked by all raters. The ordinal labels are mapped to ranks as:

$$A \mapsto 1, \qquad B \mapsto 2, \qquad C \mapsto 3.$$

Let k be the number of raters and $n = 3$ be the number of ranked motions. For prompt p, let r_{ij} denote the rank assigned by rater i to motion j. If

ties occur, mid-ranks are used. The rank sum of motion j is:

$$R_j = \sum_{i=1}^{k} r_{ij}.$$

The mean rank sum is:

$$\bar{R} = \frac{k(n+1)}{2}.$$

We compute:

$$S = \sum_{j=1}^{n} (R_j - \bar{R})^2.$$

With tie correction, Kendall's W is computed as:

$$W = \frac{12S}{k^2(n^3 - n) - k\sum_{i=1}^{k} T_i},$$

where T_i is the tie correction term for rater i:

$$T_i = \sum_{g} \frac{t_{ig}^3 - t_{ig}}{12}.$$

Here t_{ig} denotes the size of a tied group in rater i's labels. We compute W separately for each prompt and report the mean and standard deviation over all 40 prompts. A higher Kendall's W indicates stronger rank concordance among raters.

Fleiss' κ. Fleiss' κ is computed over the A/B/C categorical labels for each prompt-model cell. Each subject corresponds to one $(prompt, model)$ cell. Since we evaluate 40 prompts and 3 models, there are:

$$N = 40 \times 3$$

subjects. For each subject i, we count how many raters assign each category $j \in \{A, B, C\}$, resulting in counts n_{ij}. Let $k = 15$ be the number of raters. The observed agreement for subject i is:

$$P_i = \frac{\sum_j n_{ij}^2 - k}{k(k-1)}.$$

The mean observed agreement is:

$$\bar{P} = \frac{1}{N} \sum_{i=1}^{N} P_i.$$

The overall proportion of assignments to category j is:

$$p_j = \frac{1}{Nk} \sum_{i=1}^{N} n_{ij}.$$

The chance agreement is:

$$P_e = \sum_j p_j^2.$$

Fleiss' κ is then:

$$\kappa = \frac{\bar{P} - P_e}{1 - P_e}.$$

A higher Fleiss' κ indicates stronger agreement beyond chance.

Majority Margin. Majority Margin is used as a preference-strength diagnostic rather than a standard inter-rater agreement coefficient. For each $(prompt, model)$ cell, let v_{top} be the number of votes received by the most frequent label, and let v_{second} be the number of votes received by the second most frequent label. With $k = 15$ raters, the Majority Margin is:

$$\text{Margin} = \frac{v_{\text{top}} - v_{\text{second}}}{k}.$$

A higher value indicates that the dominant preference label is supported by a clearer majority. A lower value indicates that raters are more divided on that prompt-model cell. We report the mean, standard deviation, minimum, and maximum Majority Margin over all prompt-model cells.

Prompt-level Aggregated Preference Results. To provide a transparent summary of the human preference annotations, we report prompt-level aggregated results in Table 15. For each prompt, the table lists the majority-preferred model, the vote distribution among the three evaluated models, and the majority margin. The vote distribution follows the order HY, MotionGPT, and CloSD. During annotation, however, the model identities are hidden from raters and the display order of the three motions is randomly shuffled. The decoded model names are shown in the table only for post-hoc analysis.

Most prompts show a clear majority preference, and many prompts receive unanimous votes from all 15 raters. This indicates that the human preference reference is generally stable and can be used to validate LLM-based evaluation. A small number of prompts have lower majority margins, such as P030, where HY and CloSD receive the same number of top votes. We retain these ambiguous cases rather than removing them, since they reflect realistic disagreement in human motion-text preference judgments.

ID	Type	Major. Winner	Votes	Marg.
P001	Explosive	HY	15 / 0 / 0	1.00
P002	Explosive	HY	15 / 0 / 0	1.00
P003	Explosive	HY	15 / 0 / 0	1.00
P004	Dance	HY	13 / 2 / 0	0.73
P005	Dance	HY	15 / 0 / 0	1.00
P006	Dance	HY	12 / 0 / 3	0.60
P007	Martial	MotionGPT	5 / 10 / 0	0.33
P008	Martial	HY	15 / 0 / 0	1.00
P009	Martial	HY	15 / 0 / 0	1.00
P010	Martial	HY	15 / 0 / 0	1.00
P011	Balance	HY	15 / 0 / 0	1.00
P012	Balance	HY	15 / 0 / 0	1.00
P013	Balance	HY	15 / 0 / 0	1.00
P014	Multi-stage	HY	15 / 0 / 0	1.00
P015	Multi-stage	HY	15 / 0 / 0	1.00
P016	Multi-stage	CloSD	0 / 0 / 15	1.00
P017	Multi-stage	CloSD	0 / 0 / 15	1.00
P018	Asymmetric	HY	11 / 0 / 4	0.47
P019	Asymmetric	MotionGPT	5 / 10 / 0	0.33
P020	Asymmetric	CloSD	5 / 0 / 10	0.33
P021	Intent	MotionGPT	0 / 15 / 0	1.00
P022	Intent	HY	15 / 0 / 0	1.00
P023	Intent	HY	15 / 0 / 0	1.00
P024	Intent	HY	15 / 0 / 0	1.00
P025	Intent	HY	15 / 0 / 0	1.00
P026	Invisible	CloSD	2 / 0 / 13	0.73
P027	Invisible	HY	15 / 0 / 0	1.00
P028	Invisible	HY	15 / 0 / 0	1.00
P029	Invisible	HY	15 / 0 / 0	1.00
P030	Invisible	Tie (HY/CloSD)	7 / 1 / 7	0.00
P031	Whole-body	HY	15 / 0 / 0	1.00
P032	Whole-body	CloSD	3 / 0 / 12	0.60
P033	Whole-body	MotionGPT	6 / 9 / 0	0.20
P034	Whole-body	MotionGPT	5 / 10 / 0	0.33
P035	Whole-body	HY	15 / 0 / 0	1.00
P036	Body-part	HY	15 / 0 / 0	1.00
P037	Body-part	MotionGPT	2 / 13 / 0	0.73
P038	Body-part	HY	15 / 0 / 0	1.00
P039	Body-part	HY	13 / 2 / 0	0.73
P040	Body-part	HY	15 / 0 / 0	1.00

Table 15: Prompt-level aggregated human preference results. *Major. Winner* denotes the majority-preferred model for each prompt. *Votes* follows the order HY / MotionGPT / CloSD. *Marg.* denotes the majority margin, computed as the normalized difference between the top vote count and the second-highest vote count. During annotation, model identities are hidden from raters; the decoded model names are shown here only for analysis.

H Agreement between LLM Evaluators and Human Preferences

This appendix describes how we compute the agreement between LLM-based evaluator outputs and the aggregated human preference reference. It also provides the repeated-run protocol and the input representation comparison used to validate the LLM-based evaluation setting.

Human Trend Construction. The human preference study contains 15 independent annotation files. For each prompt, each rater assigns an A/B/C preference ranking to the three generated motions from HY, MotionGPT, and CLoSD, where A denotes the most preferred motion, B denotes the second, and C denotes the least preferred. We first aggregate the 15 human annotations into a single human trend file. Specifically, we map the ordinal labels to numerical scores:

$$A = 3, \qquad B = 2, \qquad C = 1.$$

For each prompt p and each model m, we compute the average human preference score:

$$a_{p,m}^{\text{human}} = \frac{1}{K} \sum_{r=1}^{K} q_{p,m}^{(r)},$$

where $K = 15$ is the number of raters, and $q_{p,m}^{(r)} \in \{3, 2, 1\}$ is the score assigned by rater r to model m under prompt p. The three models are then ranked according to their average human preference scores and converted back to the A/B/C format. This produces one aggregated human trend ranking for each prompt.

LLM Trend Construction. For each LLM evaluator, we convert its overall scores for the three generated motions under each prompt into the same A/B/C ranking format. The motion with the highest overall score is assigned A, the second is assigned B, and the lowest is assigned C. Thus, for each prompt, both the aggregated human reference and the LLM evaluator output provide one A/B/C ranking over the same three generated motions.

Repeated-run Protocol. To reduce the influence of evaluator stochasticity, each evaluator setting is repeated three times using the same prompt-motion inputs and the same evaluation prompt. Each repeated run produces one LLM trend file. For each run, we compute Preference Agreement Rate, Spearman's ρ, and Kendall's τ against the aggregated human trend file. The final results in Table 3

Setting	R1	R2	R3	Mean ± Std.
Gemini-3	0.8000	0.7750	0.7750	0.7833±0.0144
GLM-4.6V	0.7667	0.7750	0.7583	0.7667±0.0083
Llama-4	0.8417	0.8167	0.8417	0.8333±0.0144
Qwen3	0.7583	0.8000	0.7667	0.7750±0.0220
Claude-4.5	0.6917	0.7167	0.7000	0.7028±0.0127
GPT-5.2	0.8750	0.8417	0.8417	**0.8528**±0.0192
No color	0.8083	0.8333	0.8167	0.8194±0.0127
Skeleton	0.6250	0.5667	0.5917	0.5944±0.0293

Table 16: Run-level Preference Agreement Rate for LLM evaluator validation. R1–R3 denote three repeated evaluation runs. The last column reports mean ± standard deviation over the three runs.

Setting	R1	R2	R3	Mean ± Std.
Gemini-3	0.6625	0.6125	0.6125	0.6292±0.0289
GLM-4.6V	0.6000	0.6125	0.6000	0.6042±0.0072
Llama-4	0.7125	0.6750	0.7375	0.7083±0.0315
Qwen3	0.5750	0.6625	0.6125	0.6167±0.0439
Claude-4.5	0.4250	0.4625	0.4375	0.4417±0.0191
GPT-5.2	0.7875	0.7250	0.7500	**0.7542**±0.0315
No color	0.6625	0.7125	0.6625	0.6792±0.0289
Skeleton	0.2875	0.1125	0.1625	0.1875±0.0901

Table 17: Run-level Spearman's ρ for LLM evaluator validation. For each run, Spearman's ρ is computed per prompt and averaged over 40 prompts. The last column reports mean ± standard deviation over the three repeated runs.

Setting	R1	R2	R3	Mean ± Std.
Gemini-3	0.6000	0.5500	0.5500	0.5667±0.0289
GLM-4.6V	0.5333	0.5500	0.5167	0.5333±0.0167
Llama-4	0.6833	0.6333	0.6833	0.6667±0.0289
Qwen3	0.5167	0.6000	0.5333	0.5500±0.0441
Claude-4.5	0.3833	0.4333	0.4000	0.4056±0.0255
GPT-5.2	0.7500	0.6833	0.6833	**0.7056**±0.0385
No color	0.6167	0.6667	0.6333	0.6389±0.0255
Skeleton	0.2500	0.1333	0.1833	0.1889±0.0585

Table 18: Run-level Kendall's τ for LLM evaluator validation. For each run, Kendall's τ is computed per prompt and averaged over 40 prompts. The last column reports mean ± standard deviation over the three repeated runs.

are reported as mean ± standard deviation over the three repeated runs. Therefore, the reported standard deviation reflects repeated-evaluation variability rather than prompt-level variation.

To provide additional transparency for the repeated-run validation, Tables 16–18 report the run-level results for each metric. These tables show the three repeated runs used to compute the mean ± standard deviation reported in Table 3.

Input Representation Comparison. To examine whether the visual input format affects LLM-based evaluation, we compare three input representations while keeping the evaluator fixed as GPT-5.2. The three representations are shown in Figure 7.

The **color-coded rendering** in Figure 7(a) is the default input used in our main evaluation. It renders the SMPL body with high-contrast body-part colors and concatenates sampled frames into a spatial-temporal strip. This representation provides both body-shape cues and explicit body-part identity, helping the evaluator inspect limb-specific motions, left-right relations, and temporal transitions.

The **no-color rendering** in Figure 7(b) uses the same SMPL rendering and spatial-temporal strip format, but removes body-part color coding. This setting preserves body shape and pose information, but makes it harder for the evaluator to distinguish specific body parts and side-dependent actions.

The **skeleton-only input** in Figure 7(c) visualizes the motion using only the skeleton structure. This setting provides joint trajectory information but removes surface-level body-shape cues and color-coded body-part identity. It is used to test whether skeleton visualization alone is sufficient for reliable motion-text evaluation.

Preference Agreement Rate. Preference Agreement Rate measures whether the LLM evaluator and the aggregated human reference prefer the same motion in pairwise comparisons (Bradley and Terry, 1952; Zheng et al., 2023). For each prompt, the three generated motions form three model pairs: $\mathcal{P} = \{(\mathrm{HY}, \mathrm{MGPT}), (\mathrm{HY}, \mathrm{CSD}), (\mathrm{MGPT}, \mathrm{CSD})\}$, where MGPT and CSD denote MotionGPT and CLoSD, respectively. Across 40 prompts, this gives $40|\mathcal{P}| = 120$ pairwise comparisons.

For metric computation, we map the A/B/C rankings in both the aggregated human trend file and the LLM trend file to scores $3, 2, 1$. For a pair $(m_a, m_b) \in \mathcal{P}$ under prompt p, we define the human and LLM score differences as:

$$h_p^{ab} = s_{p,m_a}^{\text{human}} - s_{p,m_b}^{\text{human}}, \qquad \ell_p^{ab} = s_{p,m_a}^{\text{LLM}} - s_{p,m_b}^{\text{LLM}}.$$

The pair is counted as matched if the two differences have the same sign:

$$\text{Match}_p^{ab} = \mathbb{I}\left[\text{sgn}(h_p^{ab}) = \text{sgn}(\ell_p^{ab})\right].$$

This also treats the case where both evaluators assign a tie as a match. For each repeated run, the Preference Agreement Rate is:

$$\text{PrefAgree} = \frac{1}{40|\mathcal{P}|} \sum_{p=1}^{40} \sum_{(m_a,m_b)\in\mathcal{P}} \text{Match}_p^{ab}.$$

A higher value indicates that the LLM evaluator makes pairwise preferences more consistent with the aggregated human reference.

(a) Color-coded rendering (b) No-color rendering (c) Skeleton-only input

Figure 7: Comparison of input representations used in the LLM-based evaluation validation. All three examples correspond to the prompt: "squat down, jump upward with both arms raised, and land stably." The color-coded rendering is used as the default spatial-temporal strip input. The no-color rendering removes body-part color labels while preserving the SMPL body shape. The skeleton-only input keeps only the joint structure and removes body-surface and color cues.

Spearman's ρ. Spearman's ρ measures the rank correlation between the LLM ranking and the aggregated human ranking for each prompt (Spearman, 1904; Liu et al., 2023). For prompt p, we construct two score vectors over the three models:

$$\mathbf{s}_p^{\text{human}} = [s_{p,\text{HY}}^{\text{human}}, s_{p,\text{MotionGPT}}^{\text{human}}, s_{p,\text{CLoSD}}^{\text{human}}],$$

$$\mathbf{s}_p^{\text{LLM}} = [s_{p,\text{HY}}^{\text{LLM}}, s_{p,\text{MotionGPT}}^{\text{LLM}}, s_{p,\text{CLoSD}}^{\text{LLM}}].$$

We compute Spearman's ρ_p between these two vectors for each prompt. For one repeated run, the reported Spearman score is the average over all 40 prompts:

$$\bar{\rho} = \frac{1}{40}\sum_{p=1}^{40} \rho_p.$$

A higher Spearman's ρ indicates that motions ranked higher by humans are also ranked higher by the LLM evaluator.

Kendall's τ. Kendall's τ provides another rank-based comparison between the LLM evaluator and the aggregated human reference (Kendall, 1938; Gao et al., 2025). It measures the consistency of pairwise ordering between the two rankings. For each prompt, we compute Kendall's τ_p between $\mathbf{s}_p^{\text{human}}$ and $\mathbf{s}_p^{\text{LLM}}$. For one repeated run, the reported Kendall score is:

$$\bar{\tau} = \frac{1}{40}\sum_{p=1}^{40} \tau_p.$$

A higher Kendall's τ indicates stronger pairwise ranking consistency with the human reference.

Summary. Preference Agreement Rate evaluates agreement at the pairwise preference level over all 120 model comparisons. Spearman's ρ evaluates monotonic rank correlation within each prompt. Kendall's τ evaluates pairwise ranking consistency within each prompt. All three metrics are computed for each repeated run and then summarized as mean $\pm$ standard deviation over three runs. Together, these metrics assess whether an LLM evaluator and its visual input representation can reliably reproduce the aggregated human preference trend.

I Additional Benchmark Results and Analysis

I.1 Normalization Details

For radar-chart visualization, all values are normalized to $[0, 1]$ so that larger normalized values indicate better performance.

For multi-factor motion metrics, we use metric-wise min–max normalization over the 14 baselines and four instruction categories. For higher-is-better metrics, we compute:

$$\text{norm}(v) = \frac{v - \min}{\max - \min}.$$

For lower-is-better metrics, we use the reversed normalization:

$$\text{norm}(v) = \frac{\max - v}{\max - \min}.$$

The lower-is-better metrics include Jitter Degree, Ground Penetration, Foot Floating, Foot Sliding, Pose Quality, and Body Penetration.

For LLM-based evaluation scores, each sub-metric is integer-bounded. For each sub-metric,

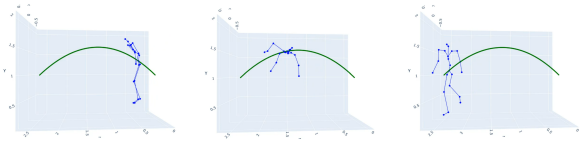

Figure 8: An unnatural example of MaskControl with joint control.

we first determine the range over the 14 baselines and four instruction categories:

$$\min_{\text{eval}} = \lfloor \min v \rfloor, \qquad \max_{\text{eval}} = \lceil \max v \rceil.$$

We then normalize each score as:

$$\text{norm}(v) = \frac{v - \min_{\text{eval}}}{\max_{\text{eval}} - \min_{\text{eval}}}.$$

If the minimum and maximum are equal, the normalized value is set to 0.5.

I.2 LLM-Based Evaluation Detailed Results

Table 19 reports the full GPT-5.2-based total scores for all evaluated T2M baselines across different instruction types. Dynamics, Complexity, and Interaction are divided into long and short subsets, while Accuracy contains high-precision short instructions. The table complements the radar-chart visualization in the main paper by providing the exact numerical scores for each baseline and instruction subset.

Overall, Table 19 shows that HY and ViMoGen achieve the strongest total-score performance across most instruction types. HY obtains the best scores on long Dynamics, long Interaction, short Dynamics, and short Complexity, while ViMoGen achieves the best scores on long Complexity and short Interaction. MDM obtains the highest score on the Accuracy subset. These results indicate that model performance varies across instruction types, and no single baseline dominates all subsets.

I.3 LLM-Based radar charts by baselines

Figure 9 provides the per-baseline radar charts for the LLM-based evaluation. Each radar chart summarizes the model performance across the LLM-evaluated dimensions, including instruction following, action completeness, temporal order, body-part understanding, and physical plausibility. These visualizations complement the main quantitative results by showing the strength and weakness patterns of each baseline across different evaluation aspects.

I.4 Multi-Factor Motion Evaluation radar charts by baseline

Figure 10 presents the per-baseline radar charts for the multi-factor motion evaluation. Unlike the LLM-based evaluation, this part focuses on automatic motion-level metrics, including semantic alignment, motion diversity, and physical quality. The radar charts provide an intuitive comparison of how different baselines balance semantic consistency, generative diversity, and motion realism.

I.5 Fine-grained accuracy assessments table

Table 20 reports the detailed fine-grained accuracy results across all evaluated baselines. The whole-body motion setting evaluates root-level controllability using root rotation, root velocity, and root translation errors, while the body-part motion setting evaluates local relative-position control using body-part translation error. All values are reported as mean RMSE, where lower values indicate better fine-grained controllability.

I.6 Multi-Factor Motion Evaluation Detailed Results

This subsection reports the detailed per-category results for the multi-factor motion evaluation. Tables 21 shows semantic alignment results, Tables 22 and 23 summarize generalizability results, and Tables 24–30 provide physical quality results. These supplementary tables complement the main benchmark results by showing how each baseline performs across long-form prompts, short-form prompts, and the Accuracy subset.

I.7 Qualitative Failure Examples

As shown in Figure 8, we present an unnatural example of MaskControl with joint contro, and the language instruction is "The person jumps 2.5 meters far".

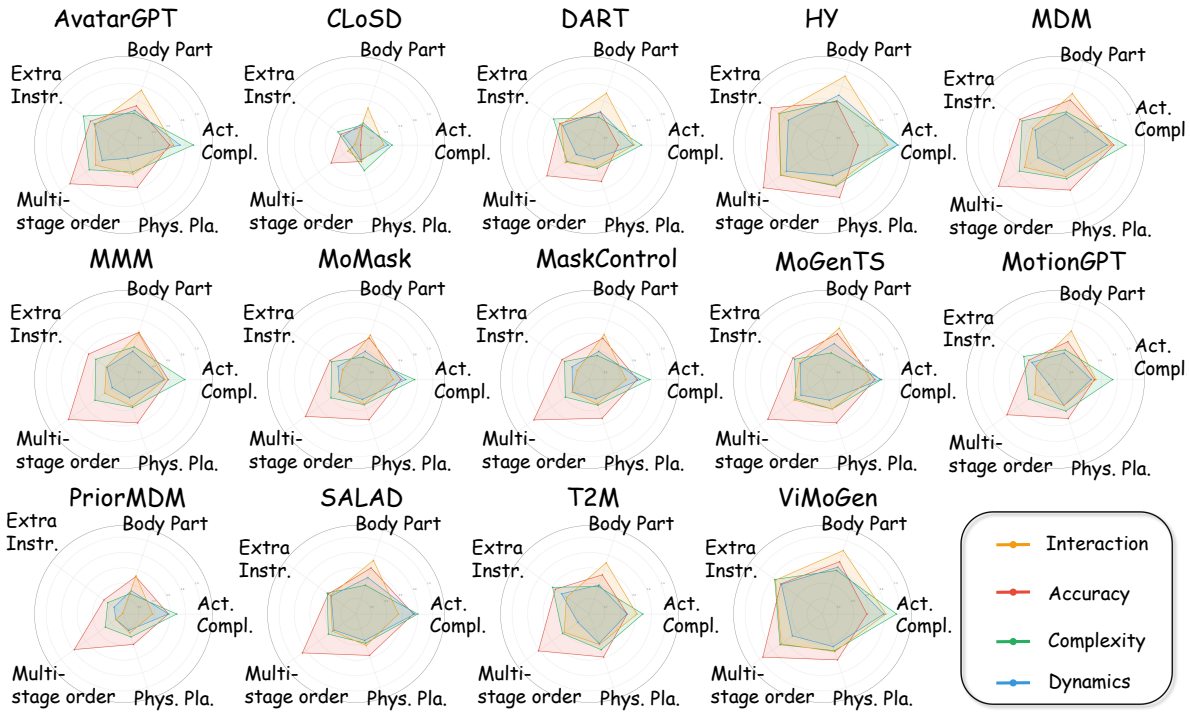

Figure 9: LLM-Based radar charts by baseline. Act. Compl., Body Part, Extra Instr., Multi-stage order, and Phys. Pla. denote Action Completeness, Body-Part Understanding, Extra Non-Instruction Actions, Multi-Stage Order Correctness, and Physical Plausibility, respectively.

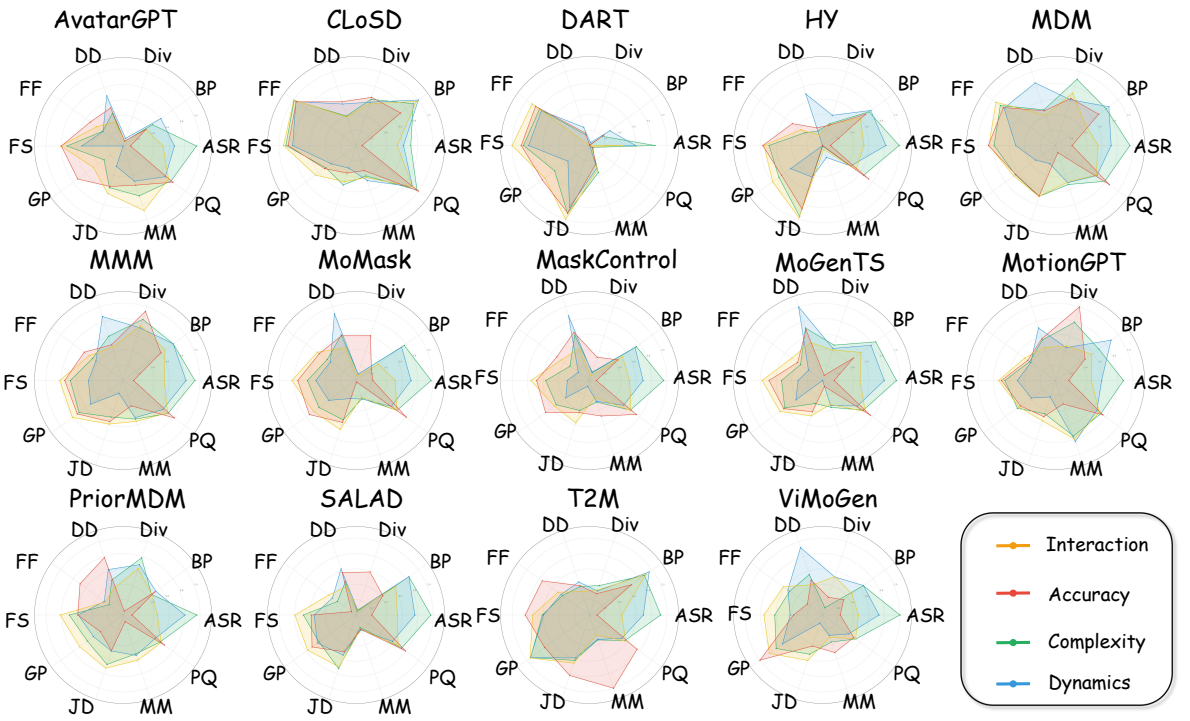

Figure 10: Multi-Factor Motion Evaluation radar charts by baseline. ASR, Div, MM, BP, DD, FF, FS, GP, JD, and PQ denote Automatic Similarity Recall, Diversity, Multimodality, Body Penetration, Dynamic Degree, Foot Floating, Foot Sliding, Ground Penetration, Jitter Degree, and Pose Quality, respectively.

Method	Long			Short			Accuracy
	Dynamics	Complexity	Interaction	Dynamics	Complexity	Interaction	
T2M (Guo et al., 2022)	38.7049	38.3606	38.8800	38.7484	42.7601	41.5467	40.9556
MDM (Tevet et al., 2022)	37.6798	41.0076	39.5700	40.8154	43.0333	42.3623	**42.7408**
MotionGPT (Jiang et al., 2023)	37.7024	40.1272	38.8200	38.3815	41.2201	39.6152	40.4341
CLoSD (Tevet et al., 2024)	35.5910	38.8218	34.1500	37.8428	36.7333	38.9652	35.7744
PriorMDM (Shafir et al., 2023)	36.0963	36.4888	34.8700	39.0029	40.9334	38.6002	40.2694
MMM (Pinyoanuntapong et al., 2024)	36.7275	38.6776	36.0300	39.1680	42.5667	42.1245	41.7803
AvatarGPT (Zhou et al., 2023)	40.0096	41.7138	39.9200	40.0738	42.4800	41.6151	41.5049
MoGenTS (Yuan et al., 2024)	39.4167	39.5521	38.8800	40.5346	41.3067	41.6969	41.9625
SALAD (Hong et al., 2025)	39.9207	39.3414	38.7000	40.8951	42.1134	41.4638	42.1765
MoMask (Guo et al., 2023)	38.3925	38.3390	37.4700	39.3940	41.4933	39.5094	41.0121
DART (Zhao et al., 2025)	40.6737	40.8206	41.6300	37.2818	39.7199	38.4119	39.5339
ViMoGen (Lin et al., 2026)	**42.6812**	**43.5114**	**43.8200**	**41.5350**	**43.7132**	**43.0371**	**42.4599**
HY (Wen et al., 2025)	**43.8414**	**43.3216**	**44.3300**	**42.0675**	**43.8333**	**42.8963**	42.2598
MaskControl (Pinyoanuntapong et al., 2025)	38.2804	38.3022	36.8100	39.4872	41.7333	39.9959	41.3516

Table 19: LLM-based evaluation total scores (↑) across instruction types. Dynamics, Complexity, and Interaction are split into long and short subsets, while Accuracy contains high-precision short instructions. The best and second-best results in each column are highlighted in red and blue, respectively.

Method	Whole-body motion			Body-part motion
	Root Rotation	Root Velocity	Root Translation	Body-part Translation
T2M	1.8055	5.7159	15.6568	0.5504
MDM	1.9690	3.2875	8.8904	0.5392
MotionGPT	1.5386	12.4603	17.2727	0.5568
CLoSD	1.7394	11.0345	14.0019	0.5446
PriorMDM	1.9986	2.2728	9.1077	0.6435
MMM	1.7370	8.7938	23.8026	0.5576
AvatarGPT	1.6213	3.1517	8.4948	0.5682
MOGENT	1.7218	2.9884	8.5591	0.5707
SALAD	1.7075	3.0097	9.0302	0.5736
MOMASK	1.5300	2.9283	8.4809	0.5311
DART	1.5580	3.4739	8.6864	0.5383
ViMoGen	1.7080	3.3874	8.2046	0.5033
HY	1.2818	2.3291	8.5777	0.5510
MaskControl	1.4947	2.9176	8.4974	0.5457
MaskControl(JointControl)	1.6815	0.1244	0.1042	0.1020

Table 20: Fine-grained accuracy assessments: mean RMSE (↓) across baselines.

Method	Long			Short			Accuracy
	Dynamics	Complexity	Interaction	Dynamics	Complexity	Interaction	
T2M	0.7395 ± 0.2722	0.7618 ± 0.3028	0.5783 ± 0.3747	0.1467 ± 0.3458	0.2867 ±0.4552	0.0500 ± 0.2077	0.1964 ± 0.3193
MDM	0.7420 ± 0.2918	0.7465 ± 0.3077	0.6258 ± 0.3675	0.1600 ± 0.3616	0.3267 ± 0.4686	0.1000 ± 0.2647	0.2393 ± 0.3572
MotionGPT	0.6581 ± 0.3021	0.7361 ± 0.2973	0.6116 ± 0.4031	0.1467 ± 0.3400	0.2800 ± 0.4509	0.0500 ± 0.1515	0.2232 ± 0.3393
CLoSD	0.6760 ± 0.2767	0.6736 ± 0.3494	0.6250 ± 0.4119	0.1467 ± 0.3256	0.2200 ± 0.3895	0.0900 ± 0.2559	0.1896 ± 0.3156
PriorMDM	0.7544 ± 0.2848	0.7944 ± 0.2612	0.5492 ± 0.4245	0.2067 ± 0.3813	0.2867 ± 0.4513	0.1000 ± 0.2647	0.2036 ± 0.3306
MMM	0.7665 ± 0.2674	0.7660 ± 0.2839	0.6150 ± 0.3527	0.2067 ± 0.3899	0.2933 ± 0.4526	0.1000 ± 0.2647	0.2107 ± 0.3169
AvatarGPT	0.7325 ± 0.2723	0.7833 ± 0.2887	0.5816 ± 0.4118	0.1333 ± 0.3190	0.2933 ± 0.4591	0.1200 ± 0.3140	0.1964 ± 0.3094
Mogent	0.7621 ± 0.2713	0.7833 ± 0.2778	0.5750 ± 0.3743	0.1400 ± 0.3304	0.3067 ± 0.4636	0.1200 ± 0.3010	0.2411 ± 0.3592
Salad	0.7792 ± 0.2601	0.7972 ± 0.2827	0.5950 ± 0.4198	0.1600 ± 0.3437	0.2933 ± 0.4591	0.1200 ± 0.2799	0.2357 ± 0.3742
Momask	0.7603 ± 0.2841	0.7694 ± 0.3164	0.6417 ± 0.3937	0.1267 ± 0.3339	0.3067 ± 0.4650	0.0600 ± 0.2222	0.1929 ± 0.3186
Dart	0.7147 ± 0.2806	0.7674 ± 0.2871	0.6216 ± 0.4001	0.1000 ± 0.2893	0.2333 ± 0.4223	0.0200 ± 0.0989	0.1607 ± 0.3166
Vimogen	0.7588 ± 0.2609	0.7542 ± 0.2962	0.5616 ± 0.3347	0.1533 ± 0.3485	0.3467 ± 0.4803	0.1000 ± 0.2841	0.2446 ± 0.3559
HY	0.7751 ± 0.2363	0.8125 ± 0.2789	0.6392 ± 0.3856	0.2000 ± 0.3831	0.2933 ± 0.4598	0.0752 ± 0.2409	0.1982 ± 0.3346
MaskControl	0.7969 ± 0.2369	0.8000 ± 0.2669	0.5975 ± 0.3898	0.1667 ± 0.3563	0.2800 ± 0.4488	0.0800 ± 0.2460	0.2161 ± 0.3362

Table 21: The multi-factor motion evaluation results in the Semantic Alignment dimension. The metrics in this table represent the ASR (↑) for 14 baseline models across the Dynamics, Complexity, and Interaction types, as well as the Accuracy type

Method	Long			Short			Accuracy
	Dynamics	Complexity	Interaction	Dynamics	Complexity	Interaction	
T2M	2.9352 ± 0.0974	3.0402 ± 0.1070	2.5601 ± 0.0797	3.3040 ± 0.0889	3.3372 ± 0.0884	3.4207 ± 0.0941	2.7628 ± 0.0860
MDM	2.5017 ± 0.0850	2.8831 ± 0.1039	2.3213 ± 0.0713	3.1636 ± 0.0800	3.0491 ± 0.0834	3.3417 ± 0.0920	2.0420 ± 0.0632
MotionGPT	5.9299 ± 0.2122	5.3763 ± 0.1909	5.5518 ± 0.2152	3.6639 ± 0.1204	3.4587 ± 0.1146	3.8737 ± 0.1201	5.0421 ± 0.1748
CLoSD	2.3604 ± 0.0754	1.9826 ± 0.0681	2.2837 ± 0.0722	3.0562 ± 0.0836	2.8389 ± 0.0752	2.8386 ± 0.0760	2.1370 ± 0.0661
PriorMDM	2.7868 ± 0.0963	2.5319 ± 0.0866	3.2992 ± 0.0890	3.1942 ± 0.0886	3.2740 ± 0.0858	3.6233 ± 0.0910	2.8958 ± 0.0811
MMM	1.9071 ± 0.0596	2.3130 ± 0.0825	2.2922 ± 0.0733	2.9123 ± 0.0744	2.8336 ± 0.0784	2.7275 ± 0.0719	2.1521 ± 0.0597
AvatarGPT	2.6664 ± 0.1196	4.0477 ± 0.1619	4.5186 ± 0.1533	2.8391 ± 0.1152	3.3097 ± 0.1358	4.5187 ± 0.1533	2.9940 ± 0.1180
Mogent	0.6697 ± 0.0335	0.6355 ± 0.0324	0.5121 ± 0.0244	2.6432 ± 0.1205	2.5903 ± 0.1208	2.7607 ± 0.1313	0.9871 ± 0.0488
Salad	0.6908 ± 0.0222	0.5589 ± 0.0165	0.5891 ± 0.0181	2.2421 ± 0.0667	2.0698 ± 0.0627	2.3974 ± 0.0820	0.7321 ± 0.0217
Momask	0.8892 ± 0.0384	0.9272 ± 0.0493	0.5251 ± 0.0227	2.7028 ± 0.1080	2.7319 ± 0.1233	2.7807 ± 0.1177	1.0361 ± 0.0482
Dart	1.9461 ± 0.0620	2.0776 ± 0.0696	2.5796 ± 0.1180	2.0346 ± 0.0615	2.3957 ± 0.0757	1.5979 ± 0.0484	1.7578 ± 0.0625
Vimogen	1.5159 ± 0.0427	1.4609 ± 0.0499	1.1261 ± 0.0312	1.4201 ± 0.0397	1.4192 ± 0.0448	1.1203 ± 0.0313	1.4952 ± 0.0391
HY	1.1208 ± 0.0377	0.5941 ± 0.0194	0.7560 ± 0.0250	1.5471 ± 0.0488	0.7461 ± 0.0252	0.9628 ± 0.0301	0.6362 ± 0.0218
MaskControl	0.8772 ± 0.0379	0.9607 ± 0.0516	0.5269 ± 0.0215	2.1488 ± 0.0851	2.8983 ± 0.1105	2.8263 ± 0.0985	1.0623 ± 0.0467

Table 22: The multi-factor motion evaluation results in the Generalizability dimension. The metrics in this table represent the Multimodality (↑) for 14 baseline models across the Dynamics, Complexity, and Interaction types, as well as the Accuracy type

Method	Long			Short			Accuracy
	Dynamics	Complexity	Interaction	Dynamics	Complexity	Interaction	
T2M	5.3956 ± 0.1293	6.8108 ± 0.1995	5.2286 ± 0.1223	7.2323 ± 0.1263	7.8616 ± 0.1577	6.2809 ± 0.1528	3.2387 ± 0.0722
MDM	6.5693 ± 0.1640	7.0042 ± 0.1706	7.1154 ± 0.1777	5.1184 ± 0.1115	8.9342 ± 0.1994	5.9357 ± 0.1778	8.9097 ± 0.1722
MotionGPT	7.7573 ± 0.2240	8.3491 ± 0.1815	7.9223 ± 0.1760	7.7850 ± 0.1534	8.5559 ± 0.1725	7.7001 ± 0.1956	3.0275 ± 0.0733
CLoSD	4.6157 ± 0.1178	5.0038 ± 0.1467	4.8469 ± 0.1292	6.2406 ± 0.1301	6.4810 ± 0.1347	5.8160 ± 0.1592	6.0418 ± 0.1090
PriorMDM	5.2105 ± 0.1424	6.1256 ± 0.1893	5.4950 ± 0.1595	6.4080 ± 0.1364	6.9921 ± 0.1545	7.0518 ± 0.1595	2.6231 ± 0.0768
MMM	5.0604 ± 0.1337	6.1839 ± 0.1617	5.8877 ± 0.1804	1.4552 ± 0.0484	6.2352 ± 0.1406	1.0665 ± 0.0280	8.4176 ± 0.1457
AvatarGPT	1.6607 ± 0.0870	0.9446 ± 0.0463	1.1291 ± 0.0258	1.3151 ± 0.0511	1.2942 ± 0.0557	1.2047 ± 0.0543	1.1845 ± 0.0433
Mogent	1.3393 ± 0.0603	1.1663 ± 0.0553	1.0990 ± 0.0480	1.3254 ± 0.0440	1.4622 ± 0.0728	1.0335 ± 0.0313	1.1845 ± 0.0404
Salad	1.1558 ± 0.0326	0.9006 ± 0.0251	1.0459 ± 0.0283	1.1858 ± 0.0435	1.0924 ± 0.0456	1.1421 ± 0.0327	5.6588 ± 0.1106
Momask	1.6219 ± 0.0686	1.6402 ± 0.0766	1.1420 ± 0.0370	1.3595 ± 0.0428	1.4873 ± 0.0585	1.1287 ± 0.0367	1.0450 ± 0.0296
Dart	0.9947 ± 0.0415	0.6505 ± 0.0192	0.5748 ± 0.0223	0.8979 ± 0.0303	0.7252 ± 0.0257	0.6399 ± 0.0382	0.6297 ± 0.0178
Vimogen	1.3654 ± 0.0365	1.4038 ± 0.0386	1.5231 ± 0.0439	1.6118 ± 0.0341	1.6524 ± 0.0498	1.6433 ± 0.0374	5.4882 ± 0.1462
HY	3.2903 ± 0.1388	2.9222 ± 0.1131	2.9099 ± 0.1119	4.6197 ± 0.1151	3.1782 ± 0.0900	3.0222 ± 0.1090	2.8880 ± 0.0806
MaskControl	1.6145 ± 0.0684	1.6045 ± 0.0652	1.0831 ± 0.0424	7.8735 ± 0.1645	1.4596 ± 0.0585	8.3909 ± 0.2147	5.9217 ± 0.1245

Table 23: The multi-factor motion evaluation results in the Generalizability dimension. The metrics in this table represent the Diversity (↑) for 14 baseline models across the Dynamics, Complexity, and Interaction types, as well as the Accuracy type

Method	Long			Short			Accuracy
	Dynamics	Complexity	Interaction	Dynamics	Complexity	Interaction	
T2M	0.0046 ± 0.0011	0.0042 ± 0.0009	0.0042 ± 0.0013	0.0077 ± 0.0025	0.0059 ± 0.0022	0.0046 ± 0.0019	0.0067 ± 0.0029
MDM	0.0153 ± 0.0071	0.0081 ± 0.0053	0.0119 ± 0.0093	0.0186 ± 0.0090	0.0099 ± 0.0049	0.0063 ± 0.0048	0.0092 ± 0.0042
MotionGPT	0.0144 ± 0.0044	0.0133 ± 0.0057	0.0138 ± 0.0064	0.0206 ± 0.0081	0.0154 ± 0.0054	0.0120 ± 0.0048	0.0127 ± 0.0045
CLoSD	0.0136 ± 0.0056	0.0090 ± 0.0046	0.0134 ± 0.0092	0.0180 ± 0.0077	0.0142 ± 0.0065	0.0119 ± 0.0083	0.0148 ± 0.0066
PriorMDM	0.0077 ± 0.0055	0.0056 ± 0.0039	0.0070 ± 0.0064	0.0229 ± 0.0127	0.0106 ± 0.0074	0.0083 ± 0.0070	0.0128 ± 0.0068
MMM	0.0185 ± 0.0053	0.0145 ± 0.0043	0.0135 ± 0.0051	0.0201 ± 0.0068	0.0135 ± 0.0053	0.0097 ± 0.0047	0.0115 ± 0.0041
AvatarGPT	0.0121 ± 0.0033	0.0084 ± 0.0023	0.0112 ± 0.0051	0.0201 ± 0.0085	0.0137 ± 0.0048	0.0086 ± 0.0040	0.0116 ± 0.0037
Mogent	0.0133 ± 0.0047	0.0102 ± 0.0039	0.0090 ± 0.0057	0.0184 ± 0.0064	0.0111 ± 0.0042	0.0084 ± 0.0052	0.0113 ± 0.0043
Salad	0.0165 ± 0.0037	0.0115 ± 0.0031	0.0129 ± 0.0045	0.0169 ± 0.0050	0.0125 ± 0.0034	0.0100 ± 0.0043	0.0125 ± 0.0029
Momask	0.0194 ± 0.0061	0.0126 ± 0.0042	0.0105 ± 0.0068	0.0203 ± 0.0062	0.0133 ± 0.0049	0.0103 ± 0.0062	0.0137 ± 0.0048
Dart	0.0053 ± 0.0015	0.0047 ± 0.0010	0.0038 ± 0.0011	0.0075 ± 0.0028	0.0053 ± 0.0022	0.0034 ± 0.0017	0.0051 ± 0.0019
Vimogen	0.0155 ± 0.0026	0.0136 ± 0.0026	0.0126 ± 0.0038	0.0164 ± 0.0025	0.0154 ± 0.0025	0.0152 ± 0.0031	0.0139 ± 0.0024
HY	0.0082 ± 0.0057	0.0036 ± 0.0035	0.0040 ± 0.0033	0.0185 ± 0.0135	0.0049 ± 0.0026	0.0037 ± 0.0029	0.0061 ± 0.0042
MaskControl	0.0207 ± 0.0052	0.0164 ± 0.0046	0.0135 ± 0.0059	0.0222 ± 0.0059	0.0155 ± 0.0053	0.0124 ± 0.0061	0.0139 ± 0.0042

Table 24: The multi-factor motion evaluation results in the Physical Quality dimension. The metrics in this table represent the Jitter Degree (↓) for 14 baseline models across the Dynamics, Complexity, and Interaction types, as well as the Accuracy type

Method	Long			Short			Accuracy
	Dynamics	Complexity	Interaction	Dynamics	Complexity	Interaction	
T2M	0.0395 ± 0.0480	0.0416 ± 0.0463	0.0340 ± 0.0489	0.0806 ± 0.1078	0.1167 ± 0.2102	0.0364 ± 0.0795	0.0869 ± 0.1031
MDM	0.1649 ± 0.1337	0.0752 ± 0.0898	0.1135 ± 0.1700	0.2347 ± 0.2790	0.1355 ± 0.1899	0.0887 ± 0.1883	0.1104 ± 0.1857
MotionGPT	0.1633 ± 0.1736	0.1168 ± 0.1235	0.2009 ± 0.2797	0.1937 ± 0.1774	0.1564 ± 0.1638	0.1864 ± 0.2537	0.1328 ± 0.1375
CLoSD	0.1273 ± 0.2055	0.0762 ± 0.1411	0.1085 ± 0.2435	0.2225 ± 0.3232	0.2160 ± 0.3624	0.1040 ± 0.2963	0.1485 ± 0.3053
PriorMDM	0.1578 ± 0.1469	0.1189 ± 0.0794	0.0959 ± 0.1401	0.1977 ± 0.1424	0.1406 ± 0.1825	0.0911 ± 0.1571	0.1896 ± 0.2247
MMM	0.1350 ± 0.1161	0.0764 ± 0.0823	0.0444 ± 0.0777	0.2007 ± 0.1603	0.1193 ± 0.1431	0.0510 ± 0.0669	0.0817 ± 0.1030
AvatarGPT	0.4377 ± 0.2441	0.2389 ± 0.1044	0.1888 ± 0.1575	0.1790 ± 0.2156	0.1818 ± 0.2512	0.1266 ± 0.1536	0.0835 ± 0.1117
Mogent	0.1352 ± 0.1164	0.1114 ± 0.0675	0.0757 ± 0.0613	0.1522 ± 0.1277	0.1010 ± 0.0914	0.0730 ± 0.1162	0.0775 ± 0.0817
Salad	0.1568 ± 0.1061	0.1297 ± 0.0961	0.1019 ± 0.0832	0.1571 ± 0.1097	0.1365 ± 0.1235	0.0885 ± 0.0985	0.0900 ± 0.1029
Momask	0.1759 ± 0.0961	0.1216 ± 0.0895	0.0781 ± 0.0902	0.1968 ± 0.1996	0.1322 ± 0.1379	0.1186 ± 0.1596	0.0911 ± 0.0834
Dart	0.2694 ± 0.4843	0.2165 ± 0.1897	0.1600 ± 0.2527	0.1342 ± 0.1781	0.0675 ± 0.1376	0.0146 ± 0.0298	0.0759 ± 0.0985
Vimogen	0.0045 ± 0.0032	0.0045 ± 0.0030	0.0051 ± 0.0046	0.0052 ± 0.0040	0.0051 ± 0.0043	0.0040 ± 0.0039	0.0041 ± 0.0033
HY	0.1453 ± 0.0903	0.0910 ± 0.0647	0.0709 ± 0.0572	0.1500 ± 0.0728	0.0939 ± 0.0613	0.0562 ± 0.0490	0.1051 ± 0.1149
MaskControl	0.1578 ± 0.0810	0.1216 ± 0.0829	0.0903 ± 0.0979	0.1843 ± 0.1609	0.1192 ± 0.1313	0.1056 ± 0.1462	0.1153 ± 0.1118

Table 25: The multi-factor motion evaluation results in the Physical Quality dimension. The metrics in this table represent the Ground Penetration (↓) for 14 baseline models across the Dynamics, Complexity, and Interaction types, as well as the Accuracy type

Method	Long			Short			Accuracy
	Dynamics	Complexity	Interaction	Dynamics	Complexity	Interaction	
T2M	0.1296 ± 0.0512	0.1487 ± 0.0624	0.1040 ± 0.0535	0.1384 ± 0.0489	0.1401 ± 0.0801	0.1058 ± 0.0456	0.1038 ± 0.0450
MDM	0.0877 ± 0.0427	0.0743 ± 0.0392	0.0570 ± 0.0452	0.0809 ± 0.0422	0.0601 ± 0.0354	0.0340 ± 0.0298	0.0799 ± 0.0421
MotionGPT	0.2621 ± 0.0848	0.2439 ± 0.0935	0.2232 ± 0.0948	0.2503 ± 0.0851	0.2264 ± 0.0932	0.2087 ± 0.0996	0.1958 ± 0.0770
CLoSD	0.0337 ± 0.0258	0.0240 ± 0.0140	0.0290 ± 0.0225	0.0479 ± 0.0294	0.0403 ± 0.0331	0.0336 ± 0.0497	0.0429 ± 0.0324
PriorMDM	0.1764 ± 0.0760	0.1958 ± 0.0831	0.1442 ± 0.0812	0.1413 ± 0.0842	0.1395 ± 0.0694	0.0935 ± 0.0556	0.1230 ± 0.0650
MMM	0.2066 ± 0.0532	0.2129 ± 0.0629	0.1704 ± 0.0605	0.1671 ± 0.0523	0.1732 ± 0.0695	0.1398 ± 0.0648	0.1429 ± 0.0569
AvatarGPT	0.2739 ± 0.0509	0.2817 ± 0.0562	0.2185 ± 0.0796	0.1974 ± 0.0626	0.1899 ± 0.0816	0.1832 ± 0.1015	0.1709 ± 0.0634
Mogent	0.2176 ± 0.0661	0.2350 ± 0.0821	0.1602 ± 0.0721	0.1917 ± 0.0806	0.1590 ± 0.0715	0.1603 ± 0.0944	0.1525 ± 0.0660
Salad	0.3079 ± 0.0694	0.3622 ± 0.0875	0.2883 ± 0.0983	0.2894 ± 0.0766	0.3117 ± 0.0893	0.3395 ± 0.0982	0.3057 ± 0.0678
Momask	0.3028 ± 0.0902	0.2448 ± 0.0862	0.1776 ± 0.0874	0.1955 ± 0.0666	0.1545 ± 0.0638	0.1265 ± 0.0708	0.1729 ± 0.0713
Dart	0.0909 ± 0.0278	0.0986 ± 0.0293	0.0596 ± 0.0233	0.1080 ± 0.0640	0.0848 ± 0.0406	0.0527 ± 0.0309	0.0722 ± 0.0377
Vimogen	0.1898 ± 0.0510	0.2119 ± 0.0498	0.1911 ± 0.0632	0.2447 ± 0.0722	0.2408 ± 0.0786	0.2525 ± 0.0847	0.2558 ± 0.0666
HY	0.2398 ± 0.1070	0.1945 ± 0.0995	0.2306 ± 0.1513	0.4003 ± 0.1266	0.2835 ± 0.1442	0.2178 ± 0.1586	0.1847 ± 0.0936
MaskControl	0.3125 ± 0.0901	0.2621 ± 0.0838	0.2005 ± 0.0676	0.2176 ± 0.0719	0.1923 ± 0.0707	0.1631 ± 0.0784	0.2095 ± 0.0734

Table 26: The multi-factor motion evaluation results in the Physical Quality dimension. The metrics in this table represent the Foot Floating (↓) for 14 baseline models across the Dynamics, Complexity, and Interaction types, as well as the Accuracy type

Method	Long			Short			Accuracy
	Dynamics	Complexity	Interaction	Dynamics	Complexity	Interaction	
T2M	0.0030 ± 0.0012	0.0032 ± 0.0015	0.0022 ± 0.0009	0.0045 ± 0.0023	0.0043 ± 0.0029	0.0026 ± 0.0013	0.0039 ± 0.0024
MDM	0.0069 ± 0.0049	0.0045 ± 0.0043	0.0047 ± 0.0049	0.0092 ± 0.0094	0.0043 ± 0.0039	0.0022 ± 0.0025	0.0034 ± 0.0025
MotionGPT	0.0065 ± 0.0032	0.0064 ± 0.0042	0.0067 ± 0.0055	0.0099 ± 0.0061	0.0075 ± 0.0040	0.0057 ± 0.0040	0.0057 ± 0.0030
CLoSD	0.0030 ± 0.0023	0.0020 ± 0.0015	0.0027 ± 0.0028	0.0048 ± 0.0038	0.0035 ± 0.0028	0.0022 ± 0.0018	0.0032 ± 0.0028
PriorMDM	0.0059 ± 0.0053	0.0041 ± 0.0030	0.0037 ± 0.0048	0.0111 ± 0.0092	0.0059 ± 0.0064	0.0032 ± 0.0033	0.0072 ± 0.0060
MMM	0.0090 ± 0.0056	0.0060 ± 0.0035	0.0040 ± 0.0017	0.0101 ± 0.0060	0.0063 ± 0.0037	0.0037 ± 0.0022	0.0049 ± 0.0032
AvatarGPT	0.0064 ± 0.0030	0.0041 ± 0.0013	0.0047 ± 0.0030	0.0092 ± 0.0066	0.0062 ± 0.0033	0.0036 ± 0.0021	0.0043 ± 0.0019
Mogent	0.0073 ± 0.0034	0.0067 ± 0.0041	0.0037 ± 0.0024	0.0091 ± 0.0047	0.0058 ± 0.0042	0.0039 ± 0.0027	0.0049 ± 0.0031
Salad	0.0117 ± 0.0055	0.0080 ± 0.0041	0.0063 ± 0.0037	0.0085 ± 0.0059	0.0069 ± 0.0028	0.0044 ± 0.0018	0.0073 ± 0.0031
Momask	0.0129 ± 0.0069	0.0076 ± 0.0053	0.0041 ± 0.0031	0.0088 ± 0.0055	0.0065 ± 0.0042	0.0041 ± 0.0026	0.0059 ± 0.0036
Dart	0.0030 ± 0.0020	0.0030 ± 0.0014	0.0019 ± 0.0012	0.0058 ± 0.0041	0.0040 ± 0.0036	0.0014 ± 0.0009	0.0032 ± 0.0017
Vimogen	0.0076 ± 0.0024	0.0079 ± 0.0029	0.0065 ± 0.0026	0.0105 ± 0.0032	0.0099 ± 0.0030	0.0082 ± 0.0025	0.0101 ± 0.0033
HY	0.0099 ± 0.0065	0.0048 ± 0.0030	0.0054 ± 0.0057	0.0196 ± 0.0162	0.0066 ± 0.0042	0.0043 ± 0.0049	0.0047 ± 0.0037
MaskControl	0.0121 ± 0.0061	0.0093 ± 0.0060	0.0050 ± 0.0041	0.0099 ± 0.0062	0.0070 ± 0.0053	0.0043 ± 0.0022	0.0058 ± 0.0033

Table 27: The multi-factor motion evaluation results in the Physical Quality dimension. The metrics in this table represent the Foot Sliding (↓) for 14 baseline models across the Dynamics, Complexity, and Interaction types, as well as the Accuracy type

Method	Long			Short			Accuracy
	Dynamics	Complexity	Interaction	Dynamics	Complexity	Interaction	
T2M	0.0171 ± 0.0046	0.0153 ± 0.0042	0.0143 ± 0.0050	0.0257 ± 0.0078	0.0213 ± 0.0079	0.0171 ± 0.0078	0.0233 ± 0.0088
MDM	0.0355 ± 0.0122	0.0225 ± 0.0093	0.0285 ± 0.0159	0.0433 ± 0.0167	0.0296 ± 0.0111	0.0192 ± 0.0111	0.0259 ± 0.0092
MotionGPT	0.0293 ± 0.0100	0.0272 ± 0.0116	0.0288 ± 0.0144	0.0409 ± 0.0177	0.0325 ± 0.0118	0.0268 ± 0.0132	0.0295 ± 0.0111
CLoSD	0.0248 ± 0.0099	0.0170 ± 0.0085	0.0234 ± 0.0164	0.0330 ± 0.0141	0.0283 ± 0.0143	0.0224 ± 0.0162	0.0304 ± 0.0144
PriorMDM	0.0221 ± 0.0113	0.0187 ± 0.0102	0.0208 ± 0.0125	0.0525 ± 0.0220	0.0297 ± 0.0151	0.0249 ± 0.0168	0.0374 ± 0.0152
MMM	0.0365 ± 0.0100	0.0298 ± 0.0080	0.0268 ± 0.0096	0.0394 ± 0.0120	0.0300 ± 0.0103	0.0214 ± 0.0105	0.0264 ± 0.0096
AvatarGPT	0.0253 ± 0.0065	0.0177 ± 0.0045	0.0233 ± 0.0122	0.0408 ± 0.0177	0.0308 ± 0.0111	0.0183 ± 0.0097	0.0277 ± 0.0098
Mogent	0.0370 ± 0.0107	0.0294 ± 0.0087	0.0238 ± 0.0110	0.0438 ± 0.0127	0.0315 ± 0.0098	0.0236 ± 0.0120	0.0309 ± 0.0109
Salad	0.0390 ± 0.0086	0.0301 ± 0.0082	0.0288 ± 0.0102	0.0392 ± 0.0104	0.0320 ± 0.0093	0.0234 ± 0.0104	0.0299 ± 0.0079
Momask	0.0408 ± 0.0121	0.0295 ± 0.0098	0.0222 ± 0.0132	0.0460 ± 0.0131	0.0328 ± 0.0117	0.0263 ± 0.0165	0.0326 ± 0.0124
Dart	0.0133 ± 0.0047	0.0116 ± 0.0030	0.0091 ± 0.0035	0.0214 ± 0.0088	0.0156 ± 0.0073	0.0081 ± 0.0062	0.0148 ± 0.0067
Vimogen	0.0302 ± 0.0053	0.0280 ± 0.0053	0.0227 ± 0.0068	0.0319 ± 0.0048	0.0290 ± 0.0049	0.0264 ± 0.0056	0.0261 ± 0.0052
HY	0.0233 ± 0.0089	0.0129 ± 0.0062	0.0136 ± 0.0089	0.0435 ± 0.0200	0.0181 ± 0.0075	0.0139 ± 0.0096	0.0174 ± 0.0087
MaskControl	0.0420 ± 0.0111	0.0346 ± 0.0103	0.0268 ± 0.0119	0.0480 ± 0.0129	0.0350 ± 0.0119	0.0281 ± 0.0160	0.0339 ± 0.0115

Table 28: The multi-factor motion evaluation results in the Physical Quality dimension. The metrics in this table represent the Dynamic Degree (↑) for 14 baseline models across the Dynamics, Complexity, and Interaction types, as well as the Accuracy type

Method	Long			Short			Accuracy
	Dynamics	Complexity	Interaction	Dynamics	Complexity	Interaction	
T2M	2.0667 ± 0.1894	1.9423 ± 0.1653	2.0374 ± 0.1972	2.1909 ± 0.2206	2.0441 ± 0.2580	1.9433 ± 0.2080	2.0737 ± 0.1594
MDM	2.3354 ± 0.3272	2.0269 ± 0.2092	2.4132 ± 0.4034	2.1770 ± 0.2566	2.0185 ± 0.2460	2.1223 ± 0.4816	1.8929 ± 0.2113
MotionGPT	2.3042 ± 0.3033	2.2648 ± 0.2994	2.3872 ± 0.4741	2.4127 ± 0.3249	2.1749 ± 0.2965	2.1551 ± 0.3139	2.0752 ± 0.2581
CLoSD	1.8286 ± 0.1480	1.6683 ± 0.1276	1.8133 ± 0.1655	1.7840 ± 0.1870	1.6804 ± 0.1901	1.7963 ± 0.2339	1.6374 ± 0.1332
PriorMDM	2.7155 ± 0.2903	2.6538 ± 0.3414	2.4847 ± 0.3638	2.5735 ± 0.2967	2.3386 ± 0.2998	2.2188 ± 0.3828	2.2444 ± 0.2176
MMM	2.1417 ± 0.1881	2.0483 ± 0.1578	2.1479 ± 0.3096	2.2049 ± 0.2236	2.0271 ± 0.2288	1.9730 ± 0.2721	1.9563 ± 0.1894
AvatarGPT	2.1451 ± 0.2068	1.9888 ± 0.1967	2.2132 ± 0.3913	2.2723 ± 0.3358	2.0981 ± 0.3258	2.0284 ± 0.4334	1.9960 ± 0.2807
Mogent	2.4895 ± 0.2200	2.3929 ± 0.2423	2.4316 ± 0.3516	2.4091 ± 0.2874	2.2051 ± 0.2941	2.1532 ± 0.3136	1.9875 ± 0.1929
Salad	2.6171 ± 0.2348	2.2139 ± 0.2276	2.4996 ± 0.3153	2.3332 ± 0.2934	2.1310 ± 0.2209	2.1906 ± 0.3404	2.0110 ± 0.1851
Momask	2.5119 ± 0.2348	2.3773 ± 0.2795	2.3395 ± 0.3685	2.3481 ± 0.2668	2.1446 ± 0.2975	2.0739 ± 0.3131	2.0940 ± 0.2769
Dart	3.3518 ± 0.1531	3.2598 ± 0.1284	3.3950 ± 0.1740	3.5097 ± 0.1843	3.4433 ± 0.1913	3.3956 ± 0.2037	3.3429 ± 0.1366
Vimogen	2.8190 ± 0.1467	2.6615 ± 0.1412	2.7540 ± 0.2762	2.8800 ± 0.1500	2.8024 ± 0.1516	2.7597 ± 0.2222	2.5419 ± 0.1527
HY	2.5910 ± 0.2041	2.2355 ± 0.1639	2.4452 ± 0.2285	2.6578 ± 0.2671	2.2754 ± 0.2879	2.5052 ± 0.4399	2.1158 ± 0.2174
MaskControl	2.4824 ± 0.2096	2.3788 ± 0.2570	2.3182 ± 0.3736	2.3876 ± 0.2680	2.1713 ± 0.3068	2.1004 ± 0.2959	2.0625 ± 0.2590

Table 29: The multi-factor motion evaluation results in the Physical Quality dimension. The metrics in this table represent the Pose Quality (↓) for 14 baseline models across the Dynamics, Complexity, and Interaction types, as well as the Accuracy type

Method	Long			Short			Accuracy
	Dynamics	Complexity	Interaction	Dynamics	Complexity	Interaction	
T2M	0.6351 ± 0.2381	0.5959 ± 0.1731	0.6566 ± 0.2912	0.7744 ± 0.2296	0.9430 ± 0.2751	0.9689 ± 0.5013	1.1542 ± 0.2424
MDM	1.0152 ± 0.3299	0.8719 ± 0.3178	1.3737 ± 0.7549	0.9723 ± 0.3081	1.1572 ± 0.3436	1.2857 ± 0.6559	1.1375 ± 0.2625
MotionGPT	1.0051 ± 0.3536	1.1743 ± 0.4768	1.1491 ± 0.4595	1.2211 ± 0.4397	1.3208 ± 0.4572	1.2759 ± 0.5751	1.3313 ± 0.3392
CLoSD	0.7123 ± 0.1830	0.8175 ± 0.1659	0.7368 ± 0.1900	0.9820 ± 0.2114	1.0226 ± 0.2412	1.0370 ± 0.4297	1.1126 ± 0.1711
PriorMDM	1.2310 ± 0.3618	1.3722 ± 0.4260	1.1266 ± 0.4573	1.2064 ± 0.2909	1.1387 ± 0.3453	1.2649 ± 0.5788	1.2994 ± 0.3111
MMM	0.7064 ± 0.2045	0.7240 ± 0.1983	0.8215 ± 0.2921	0.9219 ± 0.2216	1.1022 ± 0.3608	1.3214 ± 0.6261	1.2084 ± 0.3030
AvatarGPT	1.3163 ± 0.2847	1.3720 ± 0.2396	1.3620 ± 0.6736	1.1199 ± 0.2946	1.2966 ± 0.4039	1.4851 ± 0.6674	1.4226 ± 0.4125
Mogent	1.0407 ± 0.3844	0.9922 ± 0.2801	1.4445 ± 0.7385	1.3049 ± 0.3443	1.2789 ± 0.3449	1.4798 ± 0.7504	1.5315 ± 0.4429
Salad	1.2600 ± 0.3610	1.0602 ± 0.3427	1.3942 ± 0.7246	1.3252 ± 0.4106	1.3215 ± 0.4275	1.5561 ± 0.7961	1.3684 ± 0.3598
Momask	0.9611 ± 0.3193	0.7902 ± 0.2846	1.2879 ± 0.7569	1.1458 ± 0.3363	1.1175 ± 0.3051	1.4171 ± 0.7959	1.3472 ± 0.3529
Dart	1.5524 ± 0.2903	1.4158 ± 0.2842	1.5046 ± 0.3763	1.3854 ± 0.3881	1.6936 ± 0.4689	1.9970 ± 0.5237	1.5495 ± 0.3481
Vimogen	1.1247 ± 0.3912	1.2131 ± 0.3484	1.3998 ± 0.6030	1.1241 ± 0.3610	1.2679 ± 0.4192	1.2410 ± 0.5968	1.4497 ± 0.3510
HY	1.4784 ± 0.4847	1.2824 ± 0.4059	1.2479 ± 0.7941	0.7187 ± 0.2941	0.9361 ± 0.5012	1.1749 ± 0.9201	1.1359 ± 0.3957
MaskControl	0.9412 ± 0.2785	0.8157 ± 0.2651	1.1104 ± 0.5600	1.1597 ± 0.2833	1.1364 ± 0.3266	1.2935 ± 0.5870	1.3121 ± 0.3138

Table 30: The multi-factor motion evaluation results in the Physical Quality dimension. The metrics in this table represent the Body Penetration (↓) for 14 baseline models across the Dynamics, Complexity, and Interaction types, as well as the Accuracy type