OMNIMM: A UNIFIED LANGUAGE-MOTION MODEL WITH BIOMECHANICAL SKELETON MOTION

Anonymous authors
Paper under double-blind review

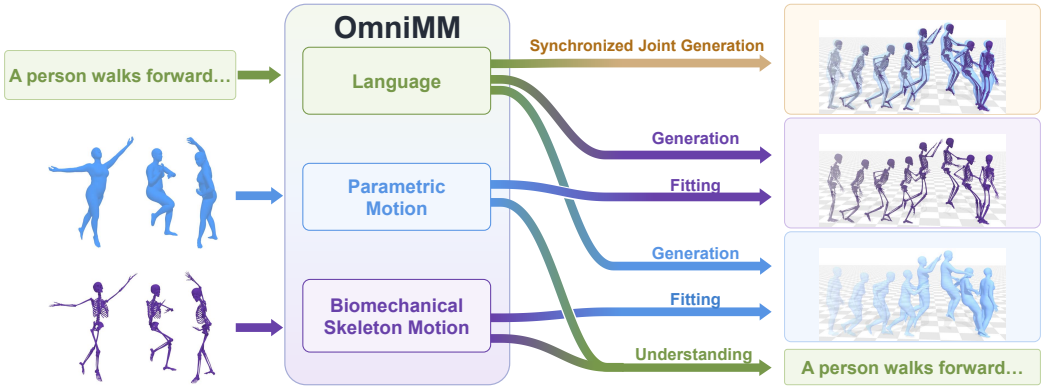


Figure 1: We present **Omni-Motion Model**. It jointly models language, surface-driven parametric motion, and anatomy-driven biomechanical skeleton motion and enables bidirectional fitting and synchronized joint generation, alongside language-conditioned generation and motion understanding.

ABSTRACT

Recent progress in multimodal human motion modeling has enabled increasingly capable systems for generating and understanding motion, yet most operate in surface-driven parametric representations that are not designed to faithfully reflect human skeletal anatomy. We introduce **Omni-Motion Model** (OmniMM), a unified framework that treats anatomy-driven biomechanical skeleton motion as a first-class modality alongside conventional parametric motion and language. OmniMM jointly models these modalities, enabling motion generation, motion understanding, bidirectional cross-representation fitting, and synchronized joint generation within a single model. To make this possible at scale, we construct paired parametric–biomechanics–language data by recovering biomechanical skeleton motion from existing SMPL sequences while preserving their original annotations. OmniMM models continuous motion representations with modality-specific Transformer towers coupled through shared attention, enabling specialized modeling while supporting cross-modal interaction. Experiments demonstrate effective modeling and translation across semantic, surface-driven, and anatomy-driven motion spaces, including the generation of parametric motion aligned with its corresponding biomechanical skeleton motion.

1 INTRODUCTION

Large-scale learning has driven rapid progress in human motion generation, enabling high-quality, diverse synthesis with strong generalization Wen et al. (2025); Lin et al. (2023); Fan et al. (2025); Hou et al. (2025). Much of this progress comes from multimodal motion models that jointly model human motion with language Zhu et al. (2025); Wen et al. (2025); Jiang et al. (2023); Chen et al. (2025), and increasingly with music Li et al. (2021); Siyao et al. (2022); Tseng et al. (2023); Li et al. (2023), audio Chen et al. (2024); Li et al. (2025); Zhang et al. (2025), and video Li et al. (2025); Zhao et al. (2026). Most existing approaches build on SMPL/SMPL-X Loper et al. (2023); Pavlakos et al. (2019) representations, while recent alternatives such as Momentum Human Rig (MHR) Ferguson et al. (2025); Yang et al. (2026) and SOMA Saito et al. (2026) further improve expressiveness and motion plausibility. Despite these advances, these representations remain predominantly *surface-driven*: their kinematic structures are designed primarily to parameterize and deform the body surface rather than to faithfully represent the anatomical skeleton.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

On the other hand, biomechanical skeletal models adopt an *anatomy-driven* representation, with skeletal structures and joint degrees of freedom designed to reflect human anatomy and articulation Rajagopal et al. (2016); Saito et al. (2015); Keller et al. (2023). Such representations provide a more anatomically grounded description of skeletal motion and are widely used for biomechanical and clinical analysis. Recent studies have begun to examine human motion through this lens Keller et al. (2023); Xia et al. (2025), revealing discrepancies between surface-driven and anatomy-driven motion. Yet, biomechanical skeleton motion remains relatively isolated and underexplored in modern multimodal motion modeling: existing models rarely generate or condition on biomechanical skeleton motion or model its relationship with conventional parametric motion, while bridging the two representations typically requires costly post-hoc, per-sequence optimization. This motivates a unified motion modeling framework that bridges surface-driven and anatomy-driven representations, enabling them to be jointly learned, translated, and generated together while remaining compatible with other modalities.

Motivated by this gap, we present **Omni-Motion Model** (OmniMM), a unified framework that treats anatomy-driven biomechanical skeleton motion as a first-class modality alongside surface-driven parametric motion and language. OmniMM jointly models these modalities and directly learns the relationship between the two motion spaces. It enables bidirectional fitting and synchronized joint generation while supporting language-conditioned motion generation Guo et al. (2022a); Wen et al. (2025); Tevet et al. (2023) and motion understanding Zhu et al. (2025); Jiang et al. (2023); Chen et al. (2025) in either representation. OmniMM encodes both motion modalities into temporally ordered continuous tokens using causal temporal autoencoders. To accommodate these heterogeneous representations, it builds on a Mixture-of-Transformers architecture Liang et al. (2024c) with modality-specific Transformer towers, enabling specialized modeling and cross-modal information exchange.

To support biomechanical skeleton motion at scale, we address a key data bottleneck: unlike parametric motion, large-scale biomechanical skeleton motion remains scarce and difficult to acquire directly. Marker-based fitting is the standard high-accuracy approach for estimating motion in both surface-driven body modeling Mahmood et al. (2019) and anatomy-driven biomechanical modeling Werling et al. (2023). Noticing that both fitting paradigms operate on a shared marker observation space, we leverage this common representation to scale biomechanical motion fitting to large annotated parametric-motion datasets. This enables us to construct paired parametric–biomechanical motion at scale while preserving the associated language and video annotations, providing scalable supervision for jointly modeling surface-driven and anatomy-driven motion.

In summary, our contributions are highlighted as:

- **Anatomy as a first-class modality.** We introduce OmniMM, unifying language, surface-driven, and anatomy-driven motion in a single framework.
- **Connecting two motion spaces.** OmniMM directly learns their relationship, enabling bidirectional fitting and synchronized joint generation alongside language-conditioned generation and motion understanding.
- **Paired motion at scale.** We construct large-scale paired parametric–biomechanical motion through shared marker observations while preserving semantic annotations.

We conduct comprehensive experiments to validate OmniMM's modeling capability and its effectiveness in bridging surface-driven and anatomy-driven motion representations.

2 Related Work

Unified Motion-Language Models. Language-conditioned human motion generation has been extensively studied with diffusion Tevet et al. (2023); Chen et al. (2023) and autoregressive models Jiang et al. (2023); Zhu et al. (2025); Zhang et al. (2024); Wang et al. (2024), achieving increasingly realistic, diverse, and generalizable motion Guo et al. (2022a; 2024); Fan et al. (2025); Wen et al. (2025); Hou et al. (2025). More recent unified motion-language models extend beyond generation to tasks such as captioning, understanding, question answering, and editing Jiang et al. (2023); Feng et al. (2024); Chen et al. (2025); Athanasiou et al. (2024); Hou et al. (2025). However, these models predominantly operate on conventional surface-driven kinematic or SMPL-style motion representations.

Human Motion Representations and Body Models. Human motion is commonly represented using kinematic skeletons or parametric body models. The SMPL family Loper et al. (2023); Pavlakos et al. (2019) has become a dominant representation for large-scale motion datasets Mahmood et al. (2019); Lin et al. (2023); Guo et al. (2022a) and motion-language modeling Jiang et al. (2023); Zhu et al. (2025); Hou et al. (2025); Li et al. (2024a). Subsequent models improve expressiveness and whole-body articulation through richer deformation models, body spaces, and rigs Osman et al. (2020; 2022); Xu et al. (2020); Ferguson et al. (2025); Saito et al. (2026); Yang et al. (2026). Despite these advances, they remain predominantly *surface-driven*, with kinematic structures designed primarily for representing and deforming the body surface. OmniMM instead brings surface-driven parametric motion and anatomy-driven biomechanical skeleton motion into a unified model. We refer readers to Sec. 3 for a detailed comparison of the two representations and their fitting paradigms.

Biomechanical Human Motion Modeling. Biomechanical models represent human motion using anatomically defined skeletal structures and joint degrees of freedom (DoFs), with established models and tools including Rajagopal Rajagopal et al. (2016) and OpenSim Delp et al. (2007); Seth et al. (2018). Recent work has increasingly connected biomechanical models with surface- and image-based motion data. BSM/SKEL Keller et al. (2023) relates surface-driven bodies to anatomical skeletons, AddBiomechanics Werling et al. (2023) and OpenCap Uhlrich et al. (2023) scale biomechanical fitting from marker-based or markerless observations, and HSMR Xia et al. (2025) estimates biomechanical skeletons from monocular images. Biomechanical priors and constraints have also been studied in motion generation Kang et al. (2025); Tashakori et al. (2025). While these methods incorporate biomechanical constraints or priors into motion generation, they do not treat biomechanical skeleton motion itself as a generative modality jointly modeled with conventional parametric motion.

3 PRELIMINARY

We briefly review the surface-driven and anatomy-driven motion-capture pipelines used to obtain high-accuracy human motion. Throughout the paper, we refer to surface-driven parametric motion, such as SMPL, as *parametric motion*, denoted by $\mathcal{M}$, biomechanical skeleton motion as *biomechanical motion*, denoted by $\mathcal{B}$, and language as $\mathcal{T}$. As illustrated in Fig. 2, both pipelines start from captured marker trajectories and, within the scope of this work, can be summarized as two-stage processes. They fit fundamentally different body representations for different downstream purposes: surface-driven approaches optimize deformable body models for graphics and animation, whereas anatomy-driven approaches optimize biomechanical skeletons with anatomically defined joints and DoFs for biomechanical and clinical analysis.

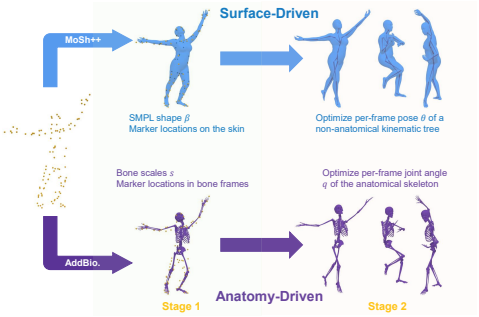


Figure 2: **Comparison of marker-based motion-capture pipelines.**

Surface-Driven Approach. We use *MoSh++* Mahmood et al. (2019), the fitting pipeline used to construct AMASS, as a representative surface-driven approach. Given observed marker locations $\mathcal{W}$, MoSh++ fits a parametric body surface model in two stages:

$$\begin{aligned} \text{Stage I:} \quad & (\tilde{\mathcal{W}}^*, \beta^*, \theta^*_{1:F}) = \arg\min_{\tilde{\mathcal{W}},\, \beta,\, \theta_{1:F}} \lambda_D E_D\left(\tilde{\mathcal{W}}, \beta, \theta_{1:F}; \mathcal{W}_{1:F}\right) + \mathcal{R}_{\text{I}}, \\ \text{Stage II:} \quad & \theta^*_{1:T} = \arg\min_{\theta_{1:T}} \lambda_D E_D\left(\tilde{\mathcal{W}}^*, \beta^*, \theta_{1:T}; \mathcal{W}_{1:T}\right) + \mathcal{R}_{\text{II}} \end{aligned} \tag{1}$$

This formulation prioritizes accurate body-surface deformation. Stage I jointly estimates SMPL body shape β, latent surface-marker locations $\tilde{\mathcal{W}}$, and poses $\theta_{1:F}$ for selected fitting frames, and Stage II recovers the complete pose sequence $\theta_{1:T}$ with β and $\tilde{\mathcal{W}}$ fixed. The objective combines marker reconstruction error E_D with regularization and prior terms $\mathcal{R}$. The resulting motion follows the SMPL kinematic tree, with generic joint rotations rather than anatomically defined DoFs. For

Table 1: **Scale of our biomechanically enriched data.** We augment six parametric-motion datasets with biomechanical skeleton motion while preserving their text annotations.

Dataset	Motion Domain	# Clips	# Frames	Duration (h)	# Text Labels	Text	Biomech.
Retargeted 100STYLE Zhong et al. (2024)	Stylised locomotion	2,062	517,519	7.19	7,789	✓	✓
AMASS Mahmood et al. (2019)	Diverse activities	10,551	1,642,456	22.81	87,384	✓	✓
CombatMotion Liao et al. (2024)	Combat / game action	7,740	504,456	7.01	40,334	✓	✓
FineDance Li et al. (2023)	Full-body dance	3,341	668,200	9.28	6,682	✓	✓
HumanSC3D Fieraru et al. (2021)	Social / scene	688	100,965	1.40	5,133	✓	✓
TRUMANS Jiang et al. (2024)	Human–scene interaction	1,871	238,542	3.31	5,613	✓	✓
Total	–	**26,253**	**3,672,138**	**51.00**	**152,935**	✓	✓

Table 2: **Quality of our biomechanically enriched motion data.** (a) Marker fitting error against BioAMASS Keller et al. (2023) on shared AMASS subsets. (b) Quality metrics compared with the AddBiomechanics dataset Werling et al. (2023).

(a) Against BioAMASS. Marker fitting error following Table 1 of Keller et al. (2023).

	Bony markers (cm) ↓		Soft markers (cm) ↓	
AMASS subset	BioAMASS	**Ours**	BioAMASS	**Ours**
DFAUST	1.54	**1.07**	2.00	**1.37**
CMU	1.70	**1.08**	2.37	**1.42**
MPI_Limits	1.70	**1.41**	2.37	**2.16**

(b) Against the AddBiomechanics dataset[1]. Quality metrics following Table 2 of Werling et al. (2023).

Measurement	**Ours**	AddBiomech.	Hicks[2]
Marker error (cm)	1.97 ± 2.27	2.17 ± 2.44	–
Lin. residual (BW)	N/A[3]	0.046 ± 0.027	0.05
Ang. residual (BW·h)	0.08 ± 0.12	0.11 ± 0.02	0.1

[1] AddBiomechanics uses real optical markers, force measurements, and the Rajagopal model Rajagopal et al. (2016); this comparison is therefore provided for context rather than direct ranking.
[2] Hicks clinical-accuracy thresholds reported in Werling et al. (2023).
[3] Our data do not include force-plate measurements; the linear residual is therefore not directly measurable.

example, SMPL uses ball-joint rotations at the knee, whereas biomechanical models can restrict knee articulation to a single flexion–extension coordinate.

Anatomy-Driven Approach. *AddBiomechanics* Werling et al. (2023) instead fits a predefined biomechanical skeleton through a bilevel optimization that jointly solves subject-specific and per-frame motion parameters. For ease of exposition, we conceptually separate this joint optimization into two stages and refer readers to Werling et al. (2023) for full details:

$$\max_{s,p} \left(\max_{q_t} P_{\bar{x}} \left(\bar{x}_t \mid q_t, s, p \right) P_s(s) P_p(p \mid \bar{p}) P_q(q) \right) \tag{2}$$

The outer optimization estimates subject-specific skeletal morphology and marker registration, parameterized by body segment scales s and segment-local marker positions p, respectively. The inner optimization recovers the skeleton model's anatomical joint coordinates q_t from observed markers $\bar{x}_t$. Both skeletal morphology and motion therefore follow the structure and DoFs of the biomechanical model, such as Rajagopal Rajagopal et al. (2016) or BSM Keller et al. (2023), rather than a body-surface parameterization. When external force measurements are available, an additional inverse-dynamics stage can improve physical plausibility. This stage is omitted here.

4 Creating a Large-Scale Biomechanically Enriched Dataset

As discussed in Sec. 3, marker-based fitting is the standard high-accuracy approach for constructing motion data in both surface-driven body modeling and anatomy-driven biomechanical modeling. Noticing that both paradigms operate on a shared marker space, we leverage it to construct biomechanical motion from large-scale parametric motion datasets. Specifically, marker trajectories obtained from the SMPL body surface are fitted independently with an anatomy-driven biomechanical model, rather than directly converting between the two parameter spaces. This strategy was used in Keller et al. (2023) to construct BioAMASS from AMASS, where the resulting biomechanical motion was validated as sufficiently accurate for biomechanical analysis. Applying this approach to large

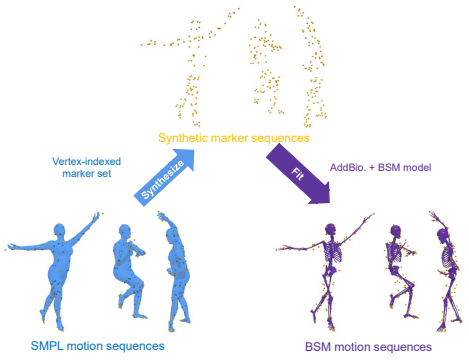


Figure 3: **Our data generation pipeline.**

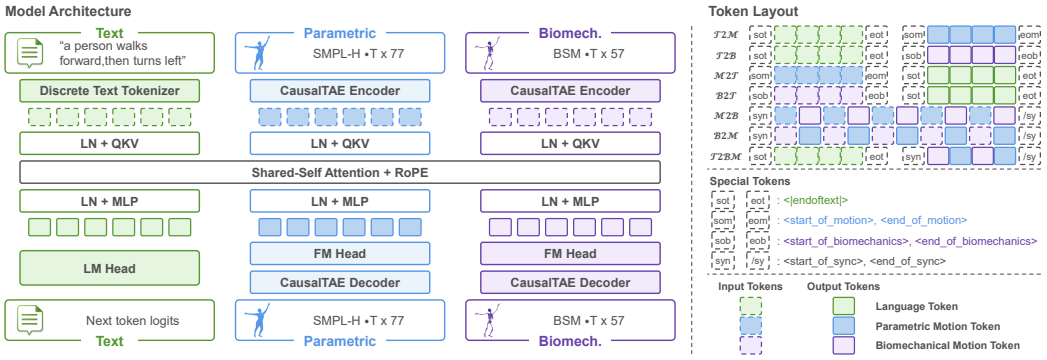


Figure 4: **Overview of OmniMM.** OmniMM jointly models language, parametric motion, and biomechanical motion under different task configurations (Sec. 5.2). Continuous motion tokens (Sec. 5.1) are processed by modality-specific Transformer towers with shared attention for cross-modal interaction (Sec. 5.3).

annotated parametric-motion datasets alleviates biomechanical data scarcity without requiring expensive new motion-capture collection, while preserving the associated annotations.

We instantiate this pipeline with BSM Keller et al. (2023), an anatomically grounded biomechanical model with rich full-body articulation, and fit it using AddBiomechanics Werling et al. (2023) with the SMPL surface-marker set defined by Keller et al. (2023). Because fitting is computationally expensive, we scale data construction through extensive parallelization. We then convert the recovered BSM motion from the OpenSim Delp et al. (2007); Seth et al. (2018) z-up convention to our y-up convention and transform it into the representation described in Sec. 5. This yields large-scale paired parametric–biomechanical–text data for training. Dataset statistics are reported in Table 1, quality comparisons in Table 2, and additional processing details are provided in Appendix B.4.

5 Omni-Motion Model

OmniMM jointly models discrete language tokens and continuous latent representations of parametric and biomechanical motion. Following recent continuous motion modeling approaches Li et al. (2024b); Zhu et al. (2025); Li et al. (2026), we represent both motion modalities with continuous tokens, and adopt a Mixture-of-Transformers (MoT) backbone Liang et al. (2024b) with modality-specific processing and cross-modal interaction. Under this formulation, OmniMM supports seven tasks spanning motion generation, motion understanding, cross-representation fitting, and synchronized joint generation. An overview of OmniMM is shown in Fig. 4.

5.1 Representations

Both motion streams are sampled at 20 fps and share, up to a rigid coordinate transformation, a heading-free, egocentric root representation $r = (t, \dot{r}_a, \dot{r}_x, \dot{r}_y, \dot{r}_z) \in \mathbb{R}^{10}$, where $t \in \mathbb{R}^6$ denotes the heading-relative pelvis orientation using the continuous 6-D rotation representation Zhou et al. (2019), $\dot{r}_a$ is the heading angular velocity, and $(\dot{r}_x, \dot{r}_y, \dot{r}_z)$ denotes the root linear velocity in the pelvis frame. Both streams also use the same four-dimensional heel/toe contact representation $c \in \{0, 1\}^4$. The shared root and contact representations align the two motion spaces while preserving their distinct joint parameterizations.

Parametric Motion Representation. Unlike conventional motion-generation frameworks that use over-parameterized kinematic representations, such as the 263-D HumanML3D representation Guo et al. (2022a); Zhu et al. (2025); Wang et al. (2024) or the 272-D TOMATO representation Fan et al. (2025); Li et al. (2026); Xiao et al. (2025), we represent motion directly in the native SMPL-H parameter space. This avoids an additional retargeting step and facilitates alignment with the biomechanical skeleton. Specifically, $m^{\mathcal{M}} = (r, \theta, c) \in \mathbb{R}^{77}$, where $\theta \in \mathbb{R}^{63}$ contains the axis–angle rotations of the 21 SMPL-H body joints.

Biomechanical Motion Representation. For biomechanical motion, we adopt BSM Keller et al. (2023), an OpenSim-compatible anatomical skeleton model Delp et al. (2007); Seth et al. (2018).

Unlike SMPL-style kinematic trees that use generic 3-DoF joint rotations, BSM represents motion using anatomically defined joint degrees of freedom. We represent biomechanical motion as $m^{\mathcal{B}} = (r, q, c) \in \mathbb{R}^{57}$, where $q \in \mathbb{R}^{43}$ contains the internal anatomical coordinates of BSM.

5.1.1 Representing Motions in Continuous Tokens

To preserve the continuous nature of parametric and biomechanical motion, we encode both modalities with continuous VAEs. Specifically, unlike MotionGPT3 Zhu et al. (2025), which compresses an entire motion sequence into a single latent token, we adopt CausalTAE Xiao et al. (2025); Li et al. (2026) to produce temporally ordered latent tokens for finer-grained causal motion modeling. The VAEs for the two motion modalities share the same architecture. Given a motion sequence of length N, $M = [m_1, \ldots, m_N] \in \mathbb{R}^{N \times D}$, where $D = 77$ for parametric motion and $D = 57$ for biomechanical motion, the encoder E produces temporally downsampled Gaussian parameters and samples a continuous latent sequence $\mathbf{z} = [z_1, \ldots, z_{N/l}] \in \mathbb{R}^{(N/l) \times d_c}$ through reparameterization:

$$(\mu, \sigma^2) = E(X), \qquad z_i = \mu_i + \sigma_i \odot \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, I). \tag{3}$$

Here, l denotes the temporal downsampling factor and d_c the latent dimension. The resulting $\mathbf{z}$ serves as the continuous motion tokens for multimodal modeling. Architecture, training, and implementation details of CausalTAE are provided in the Appendix A.1.1.

5.2 Unified Generative Formulation

Formulation. We formulate multimodal generation as autoregressive next-token prediction over a single interleaved sequence, where language is represented by discrete tokens and parametric motion $\mathcal{M}$ and biomechanical motion $\mathcal{B}$ by continuous latent tokens. Given a conditioning input $\boldsymbol{c}$ and generation target $\boldsymbol{x}$, a generic conditional generation task is formulated as $\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}} p_\theta(\boldsymbol{x} \mid \boldsymbol{c})$. We jointly model seven cross-modal tasks:

$$\begin{aligned}
\text{Generation:} \quad & p_\theta(\boldsymbol{x}^{\mathcal{R}} \mid \boldsymbol{x}^{\mathcal{T}}) = \textstyle\prod_{k=1}^{K} p_\theta(\boldsymbol{x}_k^{\mathcal{R}} \mid \boldsymbol{x}_{<k}^{\mathcal{R}}, \boldsymbol{x}^{\mathcal{T}}), \quad \mathcal{R} \in \{\mathcal{M}, \mathcal{B}\} \quad \text{Tasks: } \mathcal{T}2\mathcal{M},\ \mathcal{T}2\mathcal{B} \\
\text{Understanding:} \quad & p_\theta(\boldsymbol{x}^{\mathcal{T}} \mid \boldsymbol{x}^{\mathcal{R}}) = \textstyle\prod_{i=1}^{L} p_\theta(\boldsymbol{x}_i^{\mathcal{T}} \mid \boldsymbol{x}_{<i}^{\mathcal{T}}, \boldsymbol{x}^{\mathcal{R}}), \quad \mathcal{R} \in \{\mathcal{M}, \mathcal{B}\} \quad \text{Tasks: } \mathcal{M}2\mathcal{T},\ \mathcal{B}2\mathcal{T} \\
\text{Fitting:} \quad & p_\theta(\boldsymbol{x}^{\mathcal{R}} \mid \boldsymbol{x}^{\mathcal{Q}}) = \textstyle\prod_{k=1}^{K} p_\theta(\boldsymbol{x}_k^{\mathcal{R}} \mid \boldsymbol{x}_{<k}^{\mathcal{R}}, \boldsymbol{x}^{\mathcal{Q}}), \quad \mathcal{R}, \mathcal{Q} \in \{\mathcal{M}, \mathcal{B}\},\ \mathcal{R} \neq \mathcal{Q} \quad \text{Tasks: } \mathcal{M}2\mathcal{B},\ \mathcal{B}2\mathcal{M} \\
\text{Joint Generation:} \quad & p_\theta(\boldsymbol{x}^{\mathcal{M}}, \boldsymbol{x}^{\mathcal{B}} \mid \boldsymbol{x}^{\mathcal{T}}) = \textstyle\prod_{k=1}^{K} p_\theta(\boldsymbol{x}_k^{\mathcal{B}} \mid \boldsymbol{x}_{<k}^{\mathcal{B}}, \boldsymbol{x}_{<k}^{\mathcal{M}}, \boldsymbol{x}^{\mathcal{T}}) p_\theta(\boldsymbol{x}_k^{\mathcal{M}} \mid \boldsymbol{x}_{\leq k}^{\mathcal{B}}, \boldsymbol{x}_{<k}^{\mathcal{M}}, \boldsymbol{x}^{\mathcal{T}}) \quad \text{Task: } \mathcal{T}2\mathcal{B}\mathcal{M}
\end{aligned} \tag{4}$$

where $\boldsymbol{x}^{\mathcal{T}}$, $\boldsymbol{x}^{\mathcal{M}}$, and $\boldsymbol{x}^{\mathcal{B}}$ denote the corresponding modality-level sequences. During autoregressive modeling, the two continuous motion modalities are instantiated as CausalTAE latent sequences.

Generation of the continuous modalities $\mathcal{M}$ and $\mathcal{B}$ follows the continuous autoregressive modeling strategy of Li et al. (2024b), where each continuous latent is sampled with a conditional flow-matching head Lipman et al. (2022). For each target latent $\boldsymbol{z}_k^{\mathcal{R}}$, $\mathcal{R} \in \{\mathcal{M}, \mathcal{B}\}$, we sample $t \in (0, 1)$ from a logit-normal schedule and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and construct the linear interpolation

$$\boldsymbol{z}_{k,t}^{\mathcal{R}} = (1 - t)\boldsymbol{z}_k^{\mathcal{R}} + t\boldsymbol{\epsilon} \tag{5}$$

A velocity network $\mathcal{H}_\theta^{\mathcal{R}}$, conditioned on the Transformer hidden state $\boldsymbol{h}_k^{\mathcal{R}}$, is trained with

$$\mathcal{L}_{\text{flow}}^{\mathcal{R}} = \mathbb{E}_{\boldsymbol{z}_k^{\mathcal{R}},\, t,\, \boldsymbol{\epsilon}} \left[\left\| (\boldsymbol{\epsilon} - \boldsymbol{z}_k^{\mathcal{R}}) - \mathcal{H}_\theta^{\mathcal{R}} \left(\boldsymbol{z}_{k,t}^{\mathcal{R}}, t, \boldsymbol{h}_k^{\mathcal{R}} \right) \right\|_2^2 \right] \tag{6}$$

At inference, each latent is initialized from Gaussian noise at $t = 1$ and generated by integrating the learned velocity field backward to $t = 0$ with classifier-free guidance. The generated latent is then appended to the sequence for autoregressive prediction.

For language $\mathcal{T}$, tokens are generated autoregressively from previous text tokens and the corresponding motion conditioning. We use the standard next-token cross-entropy objective:

$$\mathcal{L}_{\text{text}} = -\sum_i \log p_\theta \left(\boldsymbol{x}_{i+1}^{\mathcal{T}} \mid \boldsymbol{x}_{\leq i}^{\mathcal{T}}, \boldsymbol{x}^{\mathcal{R}} \right), \qquad \mathcal{R} \in \{\mathcal{M}, \mathcal{B}\} \tag{7}$$

All tasks are trained with teacher forcing, using ground-truth preceding tokens or latents as the autoregressive context during training.

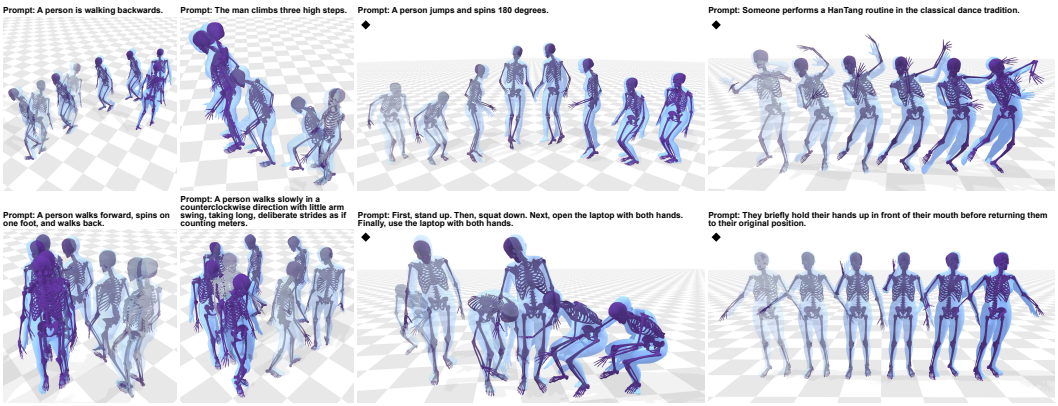


Figure 5: **Qualitative results.** Representative $\mathcal{T}2\mathcal{BM}$ generations showing parametric motion and biomechanical motion. Greater transparency indicates earlier frames. Examples marked with ♦ depict in-place motion, with successive poses offset laterally for readability.

Table 3: **Text-to-motion generation.** Evaluation of $\mathcal{T}2\mathcal{M}$, $\mathcal{T}2\mathcal{B}$ and $\mathcal{T}2\mathcal{BM}$ with corresponding modality specific evaluators. *GT* denotes ground-truth motion, and $\rightarrow$ indicates that values closer to *GT* are better. The best and second-best results are highlighted.

	$\mathcal{T}2\mathcal{M}$: Parametric Motion						$\mathcal{T}2\mathcal{B}$: Biomechanical Motion					
Methods	FID↓	R@1↑	R@3↑	MMDist↓	MModality↑	Diversity→	FID↓	R@1↑	R@3↑	MMDist↓	MModality↑	Diversity→
GT	–	0.612	0.873	0.943	–	2.193	–	0.605	0.863	1.151	–	2.510
T2M Guo et al. (2022a)	0.422	0.340	0.579	1.426	0.906	2.157	0.584	0.348	0.601	1.620	1.020	2.434
TM2T Guo et al. (2022b)	0.342	0.329	0.592	1.371	–	2.189	0.416	0.287	0.548	1.732	–	2.463
T2M-GPT Zhang et al. (2023)	0.191	0.529	0.768	1.101	0.703	2.207	0.189	0.528	0.760	1.332	0.865	2.473
MoMask Guo et al. (2024)	0.173	0.511	0.777	1.147	0.939	2.136	0.159	0.511	0.784	1.365	1.265	2.508
MotionGPT3 Zhu et al. (2025)	0.236	0.469	0.718	1.314	1.142	1.764	0.283	0.454	0.709	1.455	1.011	2.431
OmniMM ($\mathcal{T}2\mathcal{M}$ / $\mathcal{T}2\mathcal{B}$)	0.117	0.480	0.744	1.130	1.117	2.157	0.063	0.511	0.764	1.258	1.376	2.494
OmniMM ($\mathcal{T}2\mathcal{BM}$)	0.111	0.479	0.744	1.138	1.083	2.182	0.062	0.495	0.747	1.296	1.367	2.463

5.3 Model Design

Overview. OmniMM builds on the pretrained GPT-2 decoder-only backbone Radford et al. (2019) and follows the MoT design Liang et al. (2024c) with three modality-specific Transformer towers: $\mathcal{F}^{\mathcal{T}}$ for language, $\mathcal{F}^{\mathcal{M}}$ for parametric motion, and $\mathcal{F}^{\mathcal{B}}$ for biomechanical motion. The towers share the same architecture but use modality-specific parameters, while shared self-attention enables cross-modal interaction. For each motion modality, an input projection maps CausalTAE latents into the Transformer embedding space, and a flow-matching head generates latents conditioned on the corresponding hidden states for CausalTAE decoding. We replace GPT-2's absolute positional embeddings with rotary positional embeddings (RoPE) Su et al. (2024). The architecture is illustrated in Fig. 4, with implementation details provided in the Appendix A.1.2.

Modality Interface and Hybrid Token Stream. To accommodate different modalities and task configurations, OmniMM uses special tokens as modality interfaces Zhu et al. (2025). In addition to GPT-2's `<|endoftext|>` boundary token, we define `<start_of_motion>` and `<end_of_motion>` for parametric motion, and `<start_of_biomech>` and `<end_of_biomech>` for biomechanical motion. We further introduce `<start_of_sync>` and `<end_of_sync>` to delimit synchronized regions containing both continuous motion modalities. These interfaces organize the task-specific token streams illustrated in Fig. 4. For example, synchronized joint generation conditions on language and interleaves the two motion streams as follows:

$$\begin{aligned}&\texttt{<|endoftext|>}[\mathcal{T}\ \text{token}_1][\mathcal{T}\ \text{token}_2]\cdots[\mathcal{T}\ \text{token}_L]\texttt{<|endoftext|><start_of_sync>}\\&[\mathcal{B}\ \text{token}_1][\mathcal{M}\ \text{token}_1][\mathcal{B}\ \text{token}_2][\mathcal{M}\ \text{token}_2]\cdots[\mathcal{B}\ \text{token}_K][\mathcal{M}\ \text{token}_K]\texttt{<end_of_sync>}\end{aligned} \tag{8}$$

Each $\mathcal{B}$ token precedes its corresponding $\mathcal{M}$ token within the synchronized region. Bidirectional fitting uses the same interface but conditions on one observed motion stream to generate the other.

5.4 Training

OmniMM supports single-task and joint training through task-specific I/O configurations and objectives. Unused modality slots are replaced with learnable embeddings. We jointly train the modality-specific towers with teacher forcing, applying losses only to target segments:

Table 4: **Motion-to-text understanding.** Evaluation of parametric-motion understanding ($\mathcal{M}2\mathcal{T}$) and biomechanical-motion understanding ($\mathcal{B}2\mathcal{T}$).

Task	Method	R@1↑	R@2↑	R@3↑	MMDist↓	BLEU-1↑	BLEU-4↑	ROUGE↑	CIDEr↑	BERTScore↑
	GT	0.597	0.781	0.863	0.966	–	–	–	–	–
$\mathcal{M}2\mathcal{T}$	TM2T (Guo et al., 2022b)	0.363	0.513	0.596	1.418	44.43	9.86	33.70	15.30	32.83
	MotionGPT3 (Zhu et al., 2025)	0.606	0.746	0.808	2.634	50.406	17.113	41.363	50.599	49.536
	OmniMM	**0.682**	**0.832**	**0.895**	**0.837**	**56.38**	**27.74**	**48.27**	**96.46**	**55.12**
$\mathcal{B}2\mathcal{T}$	TM2T (Guo et al., 2022b)	0.320	0.464	0.550	1.709	43.56	8.51	33.13	13.40	31.77
	MotionGPT3 (Zhu et al., 2025)	0.687	0.822	0.876	2.424	52.290	19.811	42.374	58.527	50.895
	OmniMM	**0.690**	**0.840**	**0.899**	**1.009**	**56.15**	**26.86**	**47.67**	**92.11**	**54.66**

Table 5: **Cross-representation fitting and alignment.** We evaluate bidirectional fitting ($\mathcal{M}2\mathcal{B}$ and $\mathcal{B}2\mathcal{M}$), and assess cross-representation consistency for synchronized joint generation ($\mathcal{T}2\mathcal{B}\mathcal{M}$).

Task	Method	MPJPE (m) ↓	PA-MPJPE (m) ↓	Accel. (m/s^2) ↓	Time (s) ↓
$\mathcal{M}2\mathcal{B}$	Opt.-based	0.078	0.051	4.200	7.501
	OmniMM	0.080	0.054	5.280	0.440
$\mathcal{B}2\mathcal{M}$	Opt.-based	0.081	0.069	6.690	6.966
	OmniMM	0.079	0.047	3.410	1.700
$\mathcal{T}2\mathcal{B}\mathcal{M}$	**OmniMM**	0.090	0.055	4.000	4.400

$$\mathcal{L} = \lambda_{\mathcal{T}}\,\mathcal{L}_{\text{text}} + \lambda_{\mathcal{M}}\,\mathcal{L}^{\mathcal{M}}_{\text{flow}} + \lambda_{\mathcal{B}}\,\mathcal{L}^{\mathcal{B}}_{\text{flow}} \tag{9}$$

Details of our training curriculum are provided in the Appendix A.1.2.

6 Experiments

We conduct comprehensive experiments on all seven tasks in Sec. 5.2 to evaluate how OmniMM generates and understands parametric and biomechanical motion and models their relationship through bidirectional fitting and synchronized joint generation. Unless otherwise specified, competing methods are trained and evaluated on the same data splits. Fig. 5 highlights synchronized joint generation, showing temporally aligned parametric and biomechanical motion generated from a single language prompt. Additional details are provided in the Appendix B and C.3.

Motion Generation. We first evaluate whether OmniMM can generate both parametric motion and biomechanical motion from language. Following standard text-to-motion evaluation protocols Guo et al. (2022a); Liang et al. (2024a), we train modality-specific CLIP-style text–motion evaluators for the parametric and biomechanical motion spaces and report FID, retrieval accuracy (R@k), matching distance, multimodality, and diversity. For a controlled comparison, existing text-to-motion methods are trained on the same data and representation as the corresponding tasks. As shown in Table 3, OmniMM achieves competitive generation quality in both parametric and biomechanical motion spaces. The synchronized $\mathcal{T}2\mathcal{B}\mathcal{M}$ setting also preserves the quality of both output streams, with performance comparable to their standalone counterparts.

Motion Understanding. We next evaluate whether both parametric and biomechanical motion representations support semantic understanding. Given either representation as input, the model generates a natural-language description of the observed movement. Following the TM2T protocol Guo et al. (2022b), we report retrieval-based alignment metrics together with BLEU, ROUGE, CIDEr, and BERTScore. As shown in Table 4, OmniMM achieves strong performance for both $\mathcal{M}2\mathcal{T}$ and $\mathcal{B}2\mathcal{T}$, demonstrating that both parametric motion and biomechanical skeleton motion can serve as effective conditioning modalities for semantic understanding within the same model.

Cross-Representation Fitting and Alignment. We evaluate whether OmniMM directly learns the relationship between surface-driven parametric motion and anatomy-driven biomechanical motion. For bidirectional fitting, $\mathcal{M}2\mathcal{B}$ and $\mathcal{B}2\mathcal{M}$, we compare against optimization-based fitting baselines that align the two representations in joint space. We report MPJPE and PA-MPJPE for spatial alignment, acceleration for temporal smoothness, and average fitting time. Details of the optimization-based baselines are provided in the Appendix B.

As shown in Table 5, OmniMM achieves comparable or better fitting accuracy while substantially reducing inference time relative to optimization-based fitting, demonstrating that the relationship between the two representations can be modeled directly rather than solved through post-hoc optimization for every sequence. We further evaluate synchronized joint generation ($\mathcal{T}2\mathcal{B}\mathcal{M}$) using

Table 6: **Anatomical representation gap.** HSMR anatomical-limit violation rates (%) on representative joints for paired ground-truth data and $\mathcal{T}2\mathcal{BM}$ generations.

	Violation $> 10°$				Violation $> 20°$				Violation $> 30°$			
Representation	L. Elbow	R. Elbow	L. Knee	R. Knee	L. Elbow	R. Elbow	L. Knee	R. Knee	L. Elbow	R. Elbow	L. Knee	R. Knee
GT Parametric	41.2	48.1	39.8	24.8	14.5	16.8	8.3	3.5	5.9	6.7	1.3	0.6
GT Biomechanical	0.00	2.09	0.28	0.29	0.00	1.20	0.09	0.11	0.00	0.00	0.00	0.00
Generated Parametric	45.3	55.1	34.6	21.4	12.9	14.6	5.7	2.5	4.3	4.9	0.7	0.3
Generated Biomechanical	0.09	0.53	0.02	0.01	0.01	0.10	0.00	0.00	0.01	0.01	0.00	0.00

Table 7: **Auxiliary anatomical diagnostics.** We report off-axis articulation and hyperextension as the percentage of frames exceeding the corresponding anatomical thresholds.

	Off-Axis Articulation $> 5°/10°/20°$ (%)				Hyperextension $> 0°/5°/10°$ (%)			
Source	L. Elbow	R. Elbow	L. Knee	R. Knee	L. Elbow	R. Elbow	L. Knee	R. Knee
GT Parametric	65.88 / 40.93 / 14.44	74.24 / 48.00 / 16.76	79.79 / 45.31 / 10.62	63.31 / 28.56 / 4.70	2.41 / 0.91 / 0.27	2.01 / 0.74 / 0.15	17.95 / 6.05 / 0.55	10.04 / 2.10 / 0.22
GT Biomechanical	–	–	–	–	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0
Generated Parametric	74.57 / 44.89 / 12.81	80.19 / 54.98 / 14.53	76.54 / 39.86 / 7.35	61.86 / 25.40 / 3.30	2.20 / 0.92 / 0.29	2.41 / 0.87 / 0.27	19.02 / 4.39 / 0.23	5.15 / 0.50 / 0.05
Generated Biomechanical	–	–	–	–	0.08 / 0.01 / 0.00	0.17 / 0.04 / 0.02	0.73 / 0.01 / 0.00	0.60 / 0.02 / 0.00

the same joint-space metrics to measure consistency between its jointly generated parametric and biomechanical streams. The resulting low cross-representation error shows that the two generated outputs represent aligned realizations of the same underlying movement.

6.1 Anatomical Representation Gap

Although paired parametric and biomechanical motions describe the same underlying motion, their skeletal representations impose different articulation structures. We characterize this difference on the left/right elbows and knees using both paired ground-truth data and the two streams generated by $\mathcal{T}2\mathcal{BM}$. All measurements are performed directly in the native representations.

HSMR Anatomical Plausibility. Following HSMR Xia et al. (2025), we measure the percentage of frames whose local joint rotations exceed prescribed anatomical limits. Table 6 reveals a substantial gap between representations in ground-truth motion that persists in generated motion. The lower violation rates of biomechanical motion indicate greater anatomical plausibility under this criterion, consistent with BSM's explicit parameterization of anatomical DoFs with prescribed ranges, whereas SMPL uses generic local rotations for these joints. We therefore use this metric to characterize structural differences between the representations.

To interpret this gap, we introduce two auxiliary diagnostics. *Off-axis rotation magnitude* measures rotation along local axes outside the joint's anatomical DoFs, while *hyperextension rate* measures the fraction of frames whose principal flexion coordinate falls below its anatomical lower bound by more than a specified tolerance. Definitions and joint conventions are provided in Appendix C.

Table 7 shows substantial off-axis articulation in both ground-truth and generated parametric motion, whereas severe hyperextension remains rare. These patterns suggest that the HSMR gap mainly reflects rotations outside the specified anatomical DoFs rather than excessive extension. Its presence in ground-truth motion highlights why faithfully reproducing surface-driven motion statistics alone does not ensure articulation consistent with the biomechanical model. These findings motivate treating anatomy-driven motion as an explicit modality, allowing OmniMM to model anatomical joint coordinates directly alongside surface-driven motion.

7 Conclusion & Limitations

We present OmniMM, a unified framework that treats anatomy-driven biomechanical skeleton motion as a first-class modality alongside surface-driven parametric motion and language. By directly learning the relationship between the two motion spaces, OmniMM supports generation and understanding in either representation, bidirectional fitting, and synchronized joint generation. We further develop a scalable pipeline for constructing paired parametric–biomechanical motion data with language annotations. Experiments demonstrate competitive generation and understanding performance and efficient cross-representation fitting, while anatomical diagnostics highlight the importance of explicitly modeling anatomical articulation. Interestingly, joint generation does not by itself enforce anatomical constraints on the parametric stream. This motivates explicit mechanisms for transferring anatomical constraints across representations while preserving their distinct modeling objectives. Incorporating physical measurements and dynamic constraints could further advance anatomically consistent and physically grounded human motion modeling.

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools for [propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, clean and reformat dataset, support qualitative and thematic data analysis, interpret results]. We have not used generative AI tools for [help develop theoretical models or conceptual frameworks], and [generate synthetic data sets, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs] are not applicable to this work. Additionally, we used generative AI tools for [create or modify scientific figures or images, suggest experimental parameters, create or edit software code, creation of artifacts, draft parts of a research paper, transcribe recordings of research material, summarize or analyse existing literature, discover research topics or identify gaps, brainstorming, sourcing/searching for information, edit a research paper to improve readability, suggest a structure for a research paper]. We have reviewed all AI-assisted work. [We checked LMM-generated implementations and experiments and justified their interpretation.] We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide details of our data generation pipeline, including source datasets, marker synthesis, biomechanical fitting, and motion preprocessing, in Sec. 4 and Appendix B.4. The model architecture, motion representations, and training objectives are described in Sec. 5.1, Sec. 5.3 and Sec. 5.4, with implementation details, hyperparameters, and training and inference configurations provided in Appendix A.1.1 and A.1.2. We document the experimental protocols, evaluation metrics, and baseline configurations in Sec. 6 and Appendix B. Additional qualitative results are included in Appendix C.3.

REFERENCES

Nikos Athanasiou, Alpar Ceske, Markos Diomataris, Michael J. Black, and Gül Varol. MotionFix: Text-driven 3d human motion editing. In *SIGGRAPH Asia 2024 Conference Papers*, 2024.

Changan Chen, Juze Zhang, Shrinidhi Kowshika Lakshmikanth, Yusu Fang, Ruizhi Shao, Gordon Wetzstein, Li Fei-Fei, and Ehsan Adeli. The language of motion: Unifying verbal and non-verbal language of 3d human motion. In *arXiv*, 2024.

Ling-Hao Chen, Shunlin Lu, Ailing Zeng, Hao Zhang, Benyou Wang, Ruimao Zhang, and Lei Zhang. Motionllm: Understanding human behaviors from human motions and videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, and Gang Yu. Executing your commands via motion diffusion in latent space. In *CVPR*, pp. 18000–18010, 2023.

Scott L Delp, Frank C Anderson, Allison S Arnold, Peter Loan, Ayman Habib, Chand T John, Eran Guendelman, and Darryl G Thelen. Opensim: open-source software to create and analyze dynamic simulations of movement. *IEEE transactions on biomedical engineering*, 54(11):1940–1950, 2007.

Ke Fan, Shunlin Lu, Minyue Dai, Runyi Yu, Lixing Xiao, Zhiyang Dou, Junting Dong, Lizhuang Ma, and Jingbo Wang. Go to zero: Towards zero-shot motion generation with million-scale data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13336–13348, 2025.

Yao Feng, Jing Lin, Sai Kumar Dwivedi, Yu Sun, Priyanka Patel, and Michael J Black. Chatpose: Chatting about 3d human pose. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2093–2103, 2024.

Aaron Ferguson, Ahmed A. A. Osman, Berta Bescos, Carsten Stoll, Chris Twigg, Christoph Lassner, David Otte, Eric Vignola, Fabian Prada, Federica Bogo, Igor Santesteban, Javier Romero, Jenna Zarate, Jeongseok Lee, Jinhyung Park, Jinlong Yang, John Doublestein, Kishore Venkateshan, Kris

Kitani, Ladislav Kavan, Marco Dal Farra, Matthew Hu, Matthew Cioffi, Michael Fabris, Michael Ranieri, Mohammad Modarres, Petr Kadlecek, Rawal Khirodkar, Rinat Abdrashitov, Romain Prévost, Roman Rajbhandari, Ronald Mallet, Russell Pearsall, Sandy Kao, Sanjeev Kumar, Scott Parrish, Shoou-I Yu, Shunsuke Saito, Takaaki Shiratori, Te-Li Wang, Tony Tung, Yichen Xu, Yuan Dong, Yuhua Chen, Yuanlu Xu, Yuting Ye, and Zhongshi Jiang. Mhr: Momentum human rig, 2025. URL https://arxiv.org/abs/2511.15586.

Mihai Fieraru, Mihai Zanfir, Elisabeta Oneata, Alin-Ionut Popa, Vlad Olaru, and Cristian Sminchisescu. Learning complex 3d human self-contact. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 1343–1351, 2021.

Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5152–5161, 2022a.

Chuan Guo, Xinxin Zuo, Sen Wang, and Li Cheng. Tm2t: Stochastic and tokenized modeling for the reciprocal generation of 3d human motions and texts. In *European Conference on Computer Vision*, pp. 580–597. Springer, 2022b.

Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. Momask: Generative masked modeling of 3d human motions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1900–1910, 2024.

Ruibing Hou, Mingshuang Luo, Hongyu Pan, Hong Chang, and Shiguang Shan. Motionverse: A unified multimodal framework for motion comprehension, generation and editing. *arXiv preprint arXiv:2509.23635*, 2025.

Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiongpt: Human motion as a foreign language. *Advances in Neural Information Processing Systems*, 36:20067–20079, 2023.

Nan Jiang, Zhiyuan Zhang, Hongjie Li, Xiaoxuan Ma, Zan Wang, Yixin Chen, Tengyu Liu, Yixin Zhu, and Siyuan Huang. Scaling up dynamic human-scene interaction modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1737–1747, 2024.

Zixi Kang, Xinghan Wang, and Yadong Mu. Biomodiffuse: Physics-guided biomechanical diffusion for controllable and authentic human motion synthesis. *arXiv e-prints*, pp. arXiv–2503, 2025.

Marilyn Keller, Keenon Werling, Soyong Shin, Scott Delp, Sergi Pujades, C Karen Liu, and Michael J Black. From skin to skeleton: Towards biomechanically accurate 3d digital humans. *ACM Transactions on Graphics (TOG)*, 42(6):1–12, 2023.

Chuqiao Li, Julian Chibane, Yannan He, Naama Pearl, Andreas Geiger, and Gerard Pons-Moll. Unimotion: Unifying 3d human motion synthesis and understanding. *arXiv preprint arXiv:2409.15904*, 2024a.

Jiefeng Li, Jinkun Cao, Haotian Zhang, Davis Rempe, Jan Kautz, Umar Iqbal, and Ye Yuan. Genmo: A generalist model for human motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11766–11776, 2025.

Ronghui Li, Junfan Zhao, Yachao Zhang, Mingyang Su, Zeping Ren, Han Zhang, Yansong Tang, and Xiu Li. Finedance: A fine-grained choreography dataset for 3d full body dance generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10234–10243, 2023.

Ruilong Li, Shan Yang, David A. Ross, and Angjoo Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++. In *ICCV*, 2021.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems*, 37: 56424–56445, 2024b.

Zekun Li, Sizhe An, Chengcheng Tang, Chuan Guo, Ivan Shugurov, Linguang Zhang, Amy Zhao, Srinath Sridhar, Lingling Tao, and Abhay Mittal. Llamo: Scaling pretrained language models for unified motion understanding and generation with continuous autoregressive tokens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

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

Weixin Liang, Lili Yu, Liang Luo, Srinivasan Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen-tau Yih, Luke Zettlemoyer, et al. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models. *arXiv preprint arXiv:2411.04996*, 2024b.

Weixin Liang, Lili Yu, Liang Luo, Srinivasan Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen-tau Yih, Luke Zettlemoyer, et al. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models. *arXiv preprint arXiv:2411.04996*, 2024c.

Yihao Liao, Yiyu Fu, Ziming Cheng, and Jiangfeiyang Wang. Animationgpt:an aigc tool for generating game combat motion assets. `https://github.com/fyyakaxyy/AnimationGPT`, 2024.

Jing Lin, Ailing Zeng, Shunlin Lu, Yuanhao Cai, Ruimao Zhang, Haoqian Wang, and Lei Zhang. Motion-x: A large-scale 3d expressive whole-body human motion dataset. *Advances in Neural Information Processing Systems*, 36:25268–25280, 2023.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pp. 851–866. 2023.

Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, pp. 5442–5451, October 2019.

Ian Mason, Sebastian Starke, and Taku Komura. Real-time style modelling of human locomotion via feature-wise transformations and local motion phases. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 5(1), may 2022. doi: 10.1145/3522618.

Ahmed AA Osman, Timo Bolkart, and Michael J Black. Star: Sparse trained articulated human body regressor. In *European Conference on Computer Vision*, pp. 598–613. Springer, 2020.

Ahmed AA Osman, Timo Bolkart, Dimitrios Tzionas, and Michael J Black. Supr: A sparse unified part-based human representation. In *European conference on computer vision*, pp. 568–585. Springer, 2022.

Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 10975–10985, 2019.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Apoorva Rajagopal, Christopher L. Dembia, Matthew S. DeMers, Denny D. Delp, Jennifer L. Hicks, and Scott L. Delp. Full-body musculoskeletal model for muscle-driven simulation of human gait. *IEEE Transactions on Biomedical Engineering*, 63(10):2068–2079, 2016. doi: 10.1109/TBME.2016.2586891.

Jun Saito, Jiefeng Li, Michael de Ruyter, Miguel Guerrero, Edy Lim, Ehsan Hassani, Roger Blanco Ribera, Hyejin Moon, Magdalena Dadela, Marco Di Lucca, et al. Soma: Unifying parametric human body models. *arXiv preprint arXiv:2603.16858*, 2026.

Shunsuke Saito, Zi-Ye Zhou, and Ladislav Kavan. Computational bodybuilding: Anatomically-based modeling of human bodies. *ACM Trans. Graph.*, 34(4), 2015.

Ajay Seth, Jennifer L Hicks, Thomas K Uchida, Ayman Habib, Christopher L Dembia, James J Dunne, Carmichael F Ong, Matthew S DeMers, Apoorva Rajagopal, Matthew Millard, et al. Opensim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement. *PLoS computational biology*, 14(7):e1006223, 2018.

Li Siyao, Xintong Yu, and et al. Bailando: 3d dance generation by actor-critic gpt with choreographic memory. In *CVPR*, 2022.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Arvin Tashakori, Arash Tashakori, Gongbo Yang, Z Jane Wang, and Peyman Servati. Flexmotion: Lightweight, physics-aware, and controllable human motion generation. *arXiv preprint arXiv:2501.16778*, 2025.

Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. Human motion diffusion model. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=SJ1kSyO2jwu.

Jonathan Tseng, Rodrigo Castellon, and Karen Liu. Edge: Editable dance generation from music. In *CVPR*, pp. 448–458, 2023.

Scott D Uhlrich, Antoine Falisse, Łukasz Kidziński, Julie Muccini, Michael Ko, Akshay S Chaudhari, Jennifer L Hicks, and Scott L Delp. Opencap: Human movement dynamics from smartphone videos. *PLoS computational biology*, 19(10):e1011462, 2023.

Yuan Wang, Di Huang, Yaqi Zhang, Wanli Ouyang, Jile Jiao, Xuetao Feng, Yan Zhou, Pengfei Wan, Shixiang Tang, and Dan Xu. Motiongpt-2: A general-purpose motion-language model for motion generation and understanding. *arXiv preprint arXiv:2410.21747*, 2024.

Yuxin Wen, Qing Shuai, Di Kang, Jing Li, Cheng Wen, Yue Qian, Ningxin Jiao, Changhai Chen, Weijie Chen, Yiran Wang, et al. Hy-motion 1.0: Scaling flow matching models for text-to-motion generation. *arXiv preprint arXiv:2512.23464*, 2025.

Keenon Werling, Nicholas A Bianco, Michael Raitor, Jon Stingel, Jennifer L Hicks, Steven H Collins, Scott L Delp, and C Karen Liu. Addbiomechanics: Automating model scaling, inverse kinematics, and inverse dynamics from human motion data through sequential optimization. *Plos one*, 18(11): e0295152, 2023.

Yan Xia, Xiaowei Zhou, Etienne Vouga, Qixing Huang, and Georgios Pavlakos. Reconstructing humans with a biomechanically accurate skeleton. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5355–5365, 2025.

Lixing Xiao, Shunlin Lu, Huaijin Pi, Ke Fan, Liang Pan, Yueer Zhou, Ziyong Feng, Xiaowei Zhou, Sida Peng, and Jingbo Wang. Motionstreamer: Streaming motion generation via diffusion-based autoregressive model in causal latent space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10086–10096, October 2025.

Hongyi Xu, Eduard Gabriel Bazavan, Andrei Zanfir, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Ghum & ghuml: Generative 3d human shape and articulated pose models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6184–6193, 2020.

Xitong Yang, Devansh Kukreja, Don Pinkus, Anushka Sagar, Taosha Fan, Jinhyung Park, Soyong Shin, Jinkun Cao, Jiawei Liu, Nicolas Ugrinovic, Matt Feiszli, Jitendra Malik, Piotr Dollar, and Kris Kitani. Sam 3d body: Robust full-body human mesh recovery. *arXiv preprint arXiv:2602.15989*, 2026.

Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Shaoli Huang, Yong Zhang, Hongwei Zhao, Hongtao Lu, and Xi Shen. T2m-gpt: Generating human motion from textual descriptions with discrete representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Juze Zhang, Changan Chen, Xin Chen, Heng Yu, Tiange Xiang, Ali Sartaz Khan, Shrinidhi K Lakshmikanth, and Ehsan Adeli. Vibes: A conversational agent with behaviorally-intelligent 3d virtual body. *arXiv preprint arXiv:2512.14234*, 2025.

Yaqi Zhang, Di Huang, Bin Liu, Shixiang Tang, Yan Lu, Lu Chen, Lei Bai, Qi Chu, Nenghai Yu, and Wanli Ouyang. Motiongpt: Finetuned llms are general-purpose motion generators. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 7368–7376, 2024.

Chengfeng Zhao, Jiazhi Shu, Yubo Zhao, Tianyu Huang, Jiahao Lu, Zekai Gu, Chengwei Ren, Zhiyang Dou, Qing Shuai, and Yuan Liu. Comovi: Co-generation of 3d human motions and realistic videos. *arXiv preprint arXiv:2601.10632*, 2026.

Lei Zhong, Yiming Xie, Varun Jampani, Deqing Sun, and Huaizu Jiang. Smoodi: Stylized motion diffusion model. In *ECCV*, 2024.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5745–5753, 2019.

Bingfan Zhu, Biao Jiang, Sunyi Wang, Shixiang Tang, Tao Chen, Linjie Luo, Youyi Zheng, and Xin Chen. Motiongpt3: Human motion as a second modality. *arXiv preprint arXiv:2506.24086*, 2025.

A Appendix

A.1 Additional Implementation Details

A.1.1 Additional Implementation Details of CausalTAE

Both $\mathcal{M}$ and $\mathcal{B}$ are compressed by CausalTAE into sequences of continuous latents. Our CausalTAE builds on top of the causal 1-D convolutional encoder–decoder from MotionStreamer Xiao et al. (2025) and we instantiate it as a Gaussian VAE for both modalities, differing only in the input dimensionality (77 for $\mathcal{M}$ and 57 for $\mathcal{B}$).

Architecture. Fig. 6 shows the overall architecture. The encoder first maps the input to width 512 with a causal convolution, followed by three stages of strided causal convolution and causal residual blocks. A final causal convolution and linear projection produce 2×32 channels for $(\mu, \log\sigma^2)$. All convolutions use left-only padding, and each residual block consists of ReLU, a dilated causal convolution, ReLU, and a 1×1 convolution with a skip connection. Each stage contains three residual blocks with dilations $(9, 3, 1)$.

The decoder mirrors the encoder with a $32\to512$ linear projection, a causal convolution, and three stages of residual blocks, $2\times$ nearest-neighbor upsampling, and causal convolution, followed by an output convolution. The three stride-2 stages yield an $8\times$ temporal compression, producing 32-D latents at 2.5 Hz for 20 fps motion. Each tokenizer has approximately 27 M parameters. During tokenizer training, we sample from the reparameterized Gaussian, while the language model uses the posterior mean.

CausalTAE Training Objective. The tokenizer is trained on 64-frame windows with stride 32 using reconstruction, temporal-derivative, root-motion, contact, and KL objectives. The biomechanics tokenizer additionally includes an anatomical joint-limit penalty. Let $\mathbf{x} \in \mathbb{R}^{T\times D}$ be a standardized motion window and $\hat{\mathbf{x}} = \mathrm{Dec}(\mathbf{z})$ its reconstruction from latent $\mathbf{z}$. We decompose each frame as $\mathbf{x}_t = [r_t, \omega_t, v_t, q_t, c_t]$, where r_t is the 6-D root orientation, ω_t the heading rate, v_t the root linear velocity in the heading frame, q_t the joint block (63-D axis–angle for SMPL-H and 43-D generalized coordinates for BSM), and $c_t \in \{0,1\}^4$ the foot-contact labels. Let $\ell(\cdot,\cdot)$ denote the element-wise mean smooth-ℓ_1 loss (Huber, β=1), Δ the forward temporal difference, and $R(\cdot)$ the map from the 6-D rotation representation to a rotation matrix. The total objective is

$$\begin{aligned}\mathcal{L}_{\mathrm{TAE}} = \lambda_{\mathrm{feat}}\mathcal{L}_{\mathrm{feat}} + \lambda_{\mathrm{vel}}\mathcal{L}_{\mathrm{vel}} + \lambda_{\mathrm{acc}}\mathcal{L}_{\mathrm{acc}} + \lambda_{\mathrm{jerk}}\mathcal{L}_{\mathrm{jerk}} + \lambda_{\mathrm{rvel}}\mathcal{L}_{\mathrm{rvel}} + \lambda_{\mathrm{racc}}\mathcal{L}_{\mathrm{racc}} \\ + \lambda_{\mathrm{orient}}\mathcal{L}_{\mathrm{orient}} + \lambda_{\mathrm{traj}}\mathcal{L}_{\mathrm{traj}} + \lambda_{\mathrm{head}}\mathcal{L}_{\mathrm{head}} + \lambda_{\mathrm{contact}}\mathcal{L}_{\mathrm{contact}} + \lambda_{\mathrm{KL}}\mathcal{L}_{\mathrm{KL}} + \lambda_{\mathrm{lim}}\mathcal{L}_{\mathrm{lim}}.\end{aligned} \tag{10}$$

Here, $\mathcal{L}_{\mathrm{feat}} = \ell(\hat{q}, q)$ denotes joint-feature reconstruction. $\mathcal{L}_{\mathrm{vel}} = \ell(\Delta\hat{q}, \Delta q)$, $\mathcal{L}_{\mathrm{acc}} = \ell(\Delta^2\hat{q}, \Delta^2 q)$, and $\mathcal{L}_{\mathrm{jerk}} = \ell(\Delta^3\hat{q}, \Delta^3 q)$ supervise its first-, second-, and third-order temporal differences. The root-motion terms $\mathcal{L}_{\mathrm{rvel}} = \ell(\hat{v}, v)$ and $\mathcal{L}_{\mathrm{racc}} = \ell(\Delta\hat{v}, \Delta v)$ supervise root linear velocity and acceleration.

The root-orientation loss is $\mathcal{L}_{\mathrm{orient}} = \frac{1}{T}\sum_{t=1}^{T} \arccos\left(\frac{\mathrm{tr}\left(R(\hat{r}_t)^\top R(r_t)\right)-1}{2}\right) + \ell(\hat{\omega}, \omega)$. To reduce accumulated drift, we additionally supervise the integrated root and heading trajectories with $\mathcal{L}_{\mathrm{traj}} = \ell\Big(\sum_{s\le t}\hat{v}_s, \sum_{s\le t} v_s\Big)$ and $\mathcal{L}_{\mathrm{head}} = \ell\Big(\sum_{s\le t}\hat{\omega}_s, \sum_{s\le t}\omega_s\Big)$.

Finally, $\mathcal{L}_{\mathrm{contact}} = \mathrm{BCE}(\hat{c}, c)$ supervises foot contacts, and $\mathcal{L}_{\mathrm{KL}} = -\frac{1}{2}\mathbb{E}\big[1 + \log\sigma^2 - \mu^2 - \sigma^2\big]$ regularizes the Gaussian posterior. For the biomechanics tokenizer, we further apply

$$\mathcal{L}_{\mathrm{lim}} = \frac{1}{|\mathcal{J}|}\sum_{j\in\mathcal{J}} \left\langle \max(0, l_j - \hat{q}_{t,j}) + \max(0, \hat{q}_{t,j} - u_j)\right\rangle_t, \tag{11}$$

where $\mathcal{J}$ denotes the BSM coordinates with anatomical bounds $[l_j, u_j]$ in standardized units. The corresponding loss weights are listed in Table 8.

Training Details. We optimize both tokenizers with AdamW ($\beta = (0.9, 0.99)$, no weight decay), cosine learning-rate decay, gradient clipping at 1.0, and fp32 precision, using data presented in Sec. 4. The parametric-motion tokenizer is trained from scratch with a global batch size of 64 and learning rate 3.5×10^{-5}, for 346 epochs.

Table 8: **CausalTAE loss weights.**

	λ_{feat}	λ_{vel}	λ_{acc}	λ_{jerk}	λ_{rvel}	λ_{racc}	λ_{orient}	λ_{traj}	λ_{head}	λ_{contact}	λ_{KL}	λ_{lim}
Parametric (77-D)	8	3	3	2	2	2	1	1	4	1	10^{-5}	–
Biomechanical (57-D)	8	3	3	2	2	2	1	1	4	1	10^{-5}	10

Table 9: **Tokenizer training schedule.** Unlisted loss weights remain unchanged from the preceding stage and Table 8.

Tokenizer	Stage	Init.	LR	Batch	Epoch	Loss change
Parametric (77-D)	Single	Scratch	3.5×10^{-5}	2×32	346	λ_{feat}=8, λ_{head}=4
Biomech. (57-D)	A: Root	Scratch	5×10^{-5}	32	80	λ_{feat}=2; root value supervision
	B: Anatomy	A, Ep. 79	10^{-5}	32	114	λ_{feat}=8, λ_{head}=1, λ_{lim}=10
	C: Heading	B, Ep. 114	10^{-5}	32	80	λ_{head}=4

The biomechanics tokenizer is trained on one GPU with batch size 32 in three stages: (1) 80 epochs from scratch at 5×10^{-5} with the trajectory losses; (2) fine-tuning at 10^{-5} with the anatomical joint-limit loss, for 114 epoch, and (3) a final fine-tuning stage at 10^{-5} with the heading-trajectory weight increased to 4, for 80 epochs.

A.1.2 Additional Implementation Details of OmniMM

Architecture. OmniMM uses a GPT-2–small backbone with 12 layers, hidden width 768, and 12 attention heads. As shown in Fig. 5.3, each Transformer layer uses modality-specific layer normalization, QKV/output projections, and MLPs, selected by the per-token modality type, while a shared causal self-attention operates over the unified token stream to enable cross-modal interaction. All branches use the same 768-D hidden width, with modality-specific final layer normalization. The text branch is initialized from GPT-2, whereas the parametric-motion and biomechanical-motion branches are randomly initialized. The full model contains approximately 238 M parameters. Our flow-matching head $\mathcal{H}_\theta^{\mathcal{R}}$ is implemented as a 3-layer MLP with AdaLN ResBlock-style layers and hidden dimension 1024, following Zhu et al. (2025); Li et al. (2024b).

Conditioning dropout for cross-stream tasks. Fitting and joint generation share the same synchronized modality interface but differ in their conditioning. Fitting observes one ground-truth motion modality and generates the other, whereas joint generation generates both modalities. To support both capabilities, we apply modality dropout during fitting training. For $\mathcal{M}2\mathcal{B}$ and $\mathcal{B}2\mathcal{M}$, with probability $p = 0.1$, the conditioning motion latents are dropped and replaced with learnable embeddings. This exposes each continuous-modality tower to both observed and missing paired modalities, supporting cross-representation fitting while improving compatibility with joint generation.

Inference. Continuous latents are sampled with 25-step Euler integration of the flow from $\sigma = 1$ to 0. Captions are decoded greedily, and the sampled motion latents are mapped back to motion features using the frozen tokenizer decoders.

Latent-stream termination. Text segments terminate with the standard boundary token, whereas continuous latent streams have no discrete end symbol. Following MotionStreamer Xiao et al. (2025), we represent termination with a modality-specific latent sentinel. For each continuous modality $\mathcal{R}$, a fixed sentinel $e^{\mathcal{R}} \in \mathbb{R}^{32}$ is sampled once from a seeded Gaussian, rescaled to norm 3, and appended to every training sequence, $[z_1, \ldots, z_K, e_m]$. The flow head predicts this sentinel like any other latent.

At inference, a generated latent $\hat{z}$ terminates the stream when $\|\hat{z} - e_m\|_2 < \tau$, with $\tau = 1.5$. The sentinel is discarded and the corresponding closing boundary token is appended before autoregressive generation continues. For frame-synchronous tasks ($\mathcal{M}2\mathcal{B}$, $\mathcal{B}2\mathcal{M}$, and $\mathcal{T}2\mathcal{B}\mathcal{M}$), termination is applied only after a complete time step so that synchronized streams have equal length.

Training Curriculum. OmniMM is trained with a five-stage task curriculum on 4× L40S GPUs with a global batch size of 128 (32 per GPU). We use AdamW ($\beta = (0.9, 0.99)$, no weight decay)

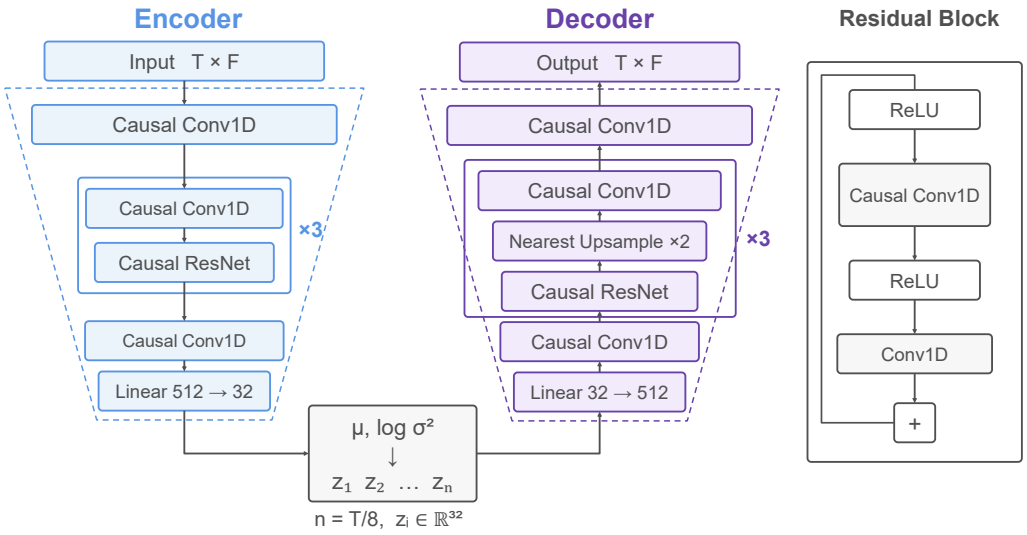


Figure 6: **Overview of the CausalTAE Architecture.**

Table 10: **Training curriculum of OmniMM.** Each stage continues from the preceding stage and cumulatively introduces new tasks.

Stage	Tasks	Epochs	Time
1: Generation	$\mathcal{T}2\mathcal{M}$, $\mathcal{T}2\mathcal{B}$	60	8.9 h
2: Understanding	+ $\mathcal{M}2\mathcal{T}$, $\mathcal{B}2\mathcal{T}$	50	6.8 h
3: Alignment	+ $\mathcal{M}2\mathcal{B}$, $\mathcal{B}2\mathcal{M}$	100	15.0 h
4: Joint	+ $\mathcal{T}2\mathcal{B}\mathcal{M}$	50	9.7 h
5: Weighted	All seven tasks	60	16.6 h

with learning rates of 2×10^{-4} for the Transformer and 2.8×10^{-4} for the flow heads, each decayed to 10^{-6} with a cosine schedule within each stage. All three modality branches are trainable, while the tokenizers remain frozen. The curriculum is cumulative, where each stage introduces additional tasks while retaining those from previous stages, as presented in Table 10. Conditioning dropout is enabled from Stage 3 onward. In Stage 5, $\mathcal{T}2\mathcal{B}\mathcal{M}$ is oversampled to approximately 35% of batches. The full curriculum takes approximately 57 hours on four L40S GPUs.

B Experimental Setup

B.1 Baselines

All learned baselines are trained using the authors' released code and recommended recipes on our 77-D parametric-motion and 57-D biomechanical-motion representations, with one model trained per modality. They use the same feature files and train/validation/test splits as OmniMM. For methods requiring a target motion length, the length is predicted from the text rather than taken from the ground truth.

T2M-GPT Zhang et al. (2023). We train T2M-GPT in two stages following the released implementation. First, we train its VQ-VAE tokenizer on our motion features using a 512-entry codebook, temporal downsampling factor 2, depth 3, dilation 3, EMA codebook reset, velocity-loss weight 0.5, and smooth-ℓ_1 reconstruction loss. The VQ-VAE is trained with batch size 256 and learning rate 2×10^{-4} for 300k iterations, with a step learning-rate decay at 200k iterations.

We then tokenize the training motions with this learned VQ-VAE and train the autoregressive GPT on the resulting discrete motion-token sequences paired with text. The GPT has 9 Transformer

layers, hidden width 1024, 16 attention heads, and block size 51. It is trained with batch size 128 and learning rate 10^{-4} for 300k iterations, with learning-rate decay starting at 150k iterations and token-keep probability $p_{\text{keep}} = 0.5$. At inference, the GPT generates motion tokens autoregressively with categorical sampling until the model predicts its end token, after which the VQ-VAE decoder maps the generated token sequence back to motion features.

MoMask Guo et al. (2024). We train MoMask following the released implementation in three stages. First, we train its residual VQ tokenizer on our motion features using 6 residual quantizers with 512 codes each, code dimension 512, temporal downsampling factor 2, and 64-frame training windows. The tokenizer is trained with batch size 256 and learning rate 2×10^{-4} for 50 epochs.

We then train the masked Transformer on the discrete motion tokens produced by the tokenizer. The masked Transformer uses 8 layers, hidden width 384, and 6 attention heads, and is trained with batch size 64 and learning rate 2×10^{-4} for 500 epochs, with conditioning dropout 0.1. The residual Transformer uses the same 8-layer, 384-D, 6-head architecture and is trained for 500 epochs with the same batch size and learning rate, using conditioning dropout 0.2. All checkpoints are selected by validation loss.

Because MoMask requires a target motion length, we train its CLIP-based text length estimator with 50 length bins and use the predicted length at inference. Generation uses 10 iterative masking steps, classifier-free guidance scales of 4 for the masked Transformer and 5 for the residual Transformer, sampling temperature 1.0, and top-k filtering with threshold 0.9.

TM2T Guo et al. (2022b). We train TM2T following the released implementation in three stages. First, we train its VQ-VAE tokenizer on our motion features using a 1,024-entry codebook with latent dimension 1024, temporal downsampling factor 2, and 3 residual blocks. The tokenizer is trained on 64-frame windows with batch size 128 and learning rate 10^{-4} for 50 epochs. Following the original recipe, an adversarial hinge-GAN objective is introduced from epoch 10 with weight 0.5.

Next, we train the motion-to-text Transformer on the discrete motion-token sequences produced by the frozen tokenizer. The model uses a 4-layer encoder and 4-layer decoder, hidden width 512, and 8 attention heads, and is trained with batch size 64 and learning rate 2×10^{-4} for 200 epochs. The validation-selected checkpoint is then frozen and used to train the text-to-motion seq2seq Transformer. This model uses a 6-layer encoder and 6-layer decoder and is trained with the same batch size and learning rate for 200 epochs, together with the cycle-consistency loss through the frozen motion-to-text model. Validation-selected checkpoints are used for all stages.

At inference, text-to-motion generation uses multinomial sampling for up to 49 discrete motion tokens, which are decoded back to motion features by the VQ-VAE decoder. Motion-to-text generation uses beam search with beam size 2 and a maximum caption length of 30 words.

T2M Guo et al. (2022a). We train T2M following the released implementation in three stages. First, we train the movement autoencoder on our motion features using 24-frame windows, batch size 128, and learning rate 10^{-4} for 30 epochs. We then train the BiGRU-based motion-length estimator with 50 length bins, batch size 64, and learning rate 10^{-4} for 50 epochs.

Finally, we train the Comp-v6 text-to-motion generator with batch size 32, learning rate 2×10^{-4}, and KL weight 0.01. Training follows the repository's curriculum over sequence lengths from 10 to 49 together with its original early-stopping procedure. The final selected checkpoints occur at epochs 378 and 362 for the parametric-motion and biomechanical-motion models, respectively.

MotionGPT3 Zhu et al. (2025). We retrain MotionGPT3 separately on our 77-D parametric-motion and 57-D biomechanical-motion representations using the official pipeline. Stage 0 trains the MLD-VAE Chen et al. (2023) on the corresponding representation for 300 epochs. The remaining stages train a Mixture-of-Transformers backbone with continuous VAE latents and an MAR-style diffusion head: Stage 1 performs text-to-motion pretraining for 300 epochs, Stage 2 adds motion-to-text and motion-prediction templates for 200 epochs, and Stage 3 performs instruction tuning for 30 epochs. Each stage is initialized from the preceding checkpoint. We use two AdamW optimizers with cosine annealing, with learning rates 4×10^{-4} for the language model and 8×10^{-4} for the diffusion head; the VAE uses 2.5×10^{-4} and 5×10^{-4}, respectively. Classifier-free guidance uses scale 3.0, unconditional probability 0.1, and a learned null condition.

Optimisation-based fitting baselines. The details are presented in Sec. B.3.

B.2 Generation Evaluation Protocol

Text–parametric and text–biomechanical motion evaluators. Evaluating text–motion alignment is inherently challenging because a single text description may correspond to multiple plausible motions, while reconstruction-based metrics do not directly capture semantic consistency. Following standard text-to-motion evaluation protocols Guo et al. (2022a); Liang et al. (2024a), we therefore train one CLIP-style contrastive evaluator for each motion modality on our datasets.

Each evaluator contains separate motion and text encoders that map their inputs into a shared 512-D embedding space. The motion encoder first projects the input features to 1024 dimensions, prepends a learned class token, and adds sinusoidal positional embeddings. The resulting sequence is processed by an 8-layer Transformer encoder with 8 attention heads, feed-forward dimension 2048, and dropout 0.1, followed by a linear projection to 512 dimensions. The text encoder uses a frozen CLIP ViT-B/32 text Transformer Radford et al. (2021) followed by a trainable linear projection to the same 512-D space. The resulting text and motion embeddings are ℓ_2-normalized and compared using a learned temperature.

The evaluators are trained with a symmetric InfoNCE objective for 150 epochs using learning rate 10^{-4}, weight decay 2×10^{-5}, 10 warm-up epochs, and cosine learning-rate decay. Training uses clips of 16–196 frames and randomly selects one caption per clip. The batch size is 64 per GPU, using 4 GPUs for the parametric-motion evaluator and 3 GPUs for the biomechanical-motion evaluator.

Generation Metrics. We report R-Precision, matching distance (MMDist), FID, diversity, and multimodality following standard text-to-motion evaluation Guo et al. (2022a); Liang et al. (2024a). For R-Precision, each replication randomly permutes the test set, partitions it into disjoint groups of 32, and ranks each motion against the 32 captions in its group using Euclidean distance in the evaluator embedding space. We report R@1 and R@3 averaged over 20 replications with a fixed seed shared across all methods. MMDist is the mean embedding distance between each generated motion and its corresponding caption.

FID measures the Fréchet distance between evaluator embeddings of generated motions and their paired ground-truth motions. Diversity is the mean embedding distance over 300 randomly sampled motion pairs. Multimodality follows Guo et al. (2022a): for 100 captions, we generate 30 samples per caption and average the pairwise distance over 10 sampled pairs. Generated motions are cropped to at most 196 frames, and contact channels are binarized before being passed to the evaluator.

Captioning Metrics. For $\mathcal{M}2\mathcal{T}$ and $\mathcal{B}2\mathcal{T}$, we evaluate generated captions against all available references for each clip using BLEU-1/4, ROUGE-L, CIDEr, and BERTScore. BLEU, ROUGE-L, and CIDEr are computed at the corpus level, while BERTScore uses baseline rescaling and takes the best-matching reference for each clip. We also measure text–motion semantic alignment with the same retrieval protocol used for motion generation. The generated caption is embedded and ranked against the ground-truth motion features, with retrieval using the human reference caption reported as an upper-bound reference. Captions are decoded greedily.

B.3 Fitting Evaluation Protocol

We evaluate cross-representation fitting in a common SKEL joint space Keller et al. (2023). In both directions, the source and predicted motions are mapped to SKEL joints using the corresponding modality conventions, enabling direct spatial comparison between parametric and biomechanical motion.

B.3.1 $\mathcal{M}2\mathcal{B}$

Reference. Both prediction and reference are evaluated in the 24-joint SKEL anatomical space. Let $\mathbf{x}^b_{1:T} \in \mathbb{R}^{T\times 57}$ denote a biomechanical-motion sequence, and let $\mathcal{G}_{\mathcal{B}}$ denote the evaluation mapping that decodes the 57-D representation to BSM OpenSim coordinates and maps it to SKEL joints using the biomechanics-to-SKEL fitting procedure described below. The resulting anatomical joint

sequence is

$$\mathbf{J}^b_{1:T} = \mathcal{G}_{\mathcal{B}}(\mathbf{x}^b_{1:T}) \in \mathbb{R}^{T\times 24\times 3}. \tag{12}$$

The ground-truth reference is therefore $\mathbf{J}^{b,\mathrm{gt}}_{1:T} = \mathcal{G}_{\mathcal{B}}(\mathbf{x}^{b,\mathrm{gt}}_{1:T})$.

Biomechanics-to-SKEL Fit. To implement $\mathcal{G}_{\mathcal{B}}$, we fit BSM motion to SKEL Keller et al. (2023). At frame t, let $\mathbf{q}_t \in \mathbb{R}^{49}$ denote the generalized coordinates of the BSM OpenSim model, consisting of 3 pelvis rotations, 3 pelvis translations, and 43 anatomical joint DoFs. The coordinates are obtained either directly from the ground-truth `.mot` file or by decoding the generated 57-D biomechanical representation. We denote by $\mathrm{FK}^{\mathrm{bsm}}(\mathbf{q}_t)$ the exact forward kinematics of the BSM OpenSim model, which maps $\mathbf{q}_t$ to the world-space positions of its body segments. Specifically, $\mathrm{FK}^{\mathrm{bsm}}_k(\mathbf{q}_t) \in \mathbb{R}^3$ denotes the world position of BSM segment k, expressed in SKEL's y-up coordinate convention.

Let $\mathcal{S}(\boldsymbol{\theta}_t, \boldsymbol{\beta}, \mathbf{r}_t)$ denote SKEL forward kinematics, with pose $\boldsymbol{\theta}_t \in \mathbb{R}^{46}$, shape $\boldsymbol{\beta} \in \mathbb{R}^{10}$, and root translation $\mathbf{r}_t \in \mathbb{R}^3$. We fix the shape to the universal skeleton, $\boldsymbol{\beta} = \mathbf{0}$, and use the 24 name-matched correspondences between SKEL joints and BSM segments. For each 64-frame window, we solve

$$\min_{\{\boldsymbol{\theta}_t,\mathbf{r}_t\}_{t=1}^T} \frac{1}{24T} \sum_{t=1}^{T} \sum_{(j,k)} \left\| \mathcal{S}_j(\boldsymbol{\theta}_t, \mathbf{0}, \mathbf{r}_t) - \mathrm{FK}^{\mathrm{bsm}}_k(\mathbf{q}_t) \right\|_2^2. \tag{13}$$

We initialize pelvis rotation, translation, and anatomical joint angles directly from $\mathbf{q}_t$ and optimize with L-BFGS using a strong-Wolfe line search for up to 300 iterations per window. No joint-limit, orientation, or temporal regularization is applied. The residual reaches approximately 14 mm under this evaluation fit. The resulting SKEL joints define the common evaluation space for ground-truth and generated biomechanical motion.

Optimization-Based Baseline. We fit SKEL directly to the ground-truth SMPL joint trajectories using the SKEL–SMPL joint correspondences $\mathcal{C}$ defined in Appendix B.3.4. Let $\mathbf{J}^m_{t,j} \in \mathbb{R}^3$ denote the position of SMPL joint j at frame t. With the universal SKEL shape fixed to $\boldsymbol{\beta} = \mathbf{0}$, we optimize

$$\min_{\{\boldsymbol{\theta}_t,\mathbf{r}_t\}_{t=1}^T} \frac{\lambda_J}{T|\mathcal{C}|} \sum_{t=1}^{T} \sum_{(i,j)\in\mathcal{C}} \left\| \mathcal{S}_i(\boldsymbol{\theta}_t, \mathbf{0}, \mathbf{r}_t) - \mathbf{J}^m_{t,j} \right\|_2^2 + \lambda_\Psi \Psi(\boldsymbol{\theta}), \tag{14}$$

where the first term minimizes the discrepancy between corresponding SKEL and SMPL joints, and Ψ softly penalizes violations of SKEL's anatomical joint limits:

$$\Psi(\boldsymbol{\theta}) = \frac{1}{T|\mathcal{D}|} \sum_{t=1}^{T} \sum_{d\in\mathcal{D}} \left([\theta_{t,d} - h_d]_+^2 + [\ell_d - \theta_{t,d}]_+^2 \right), \tag{15}$$

where ℓ_d and h_d denote the lower and upper bounds of the d-th pose DoF, $[x]_+ = \max(0, x)$, and $\mathcal{D}$ contains the 32 of 46 SKEL pose DoFs with bounds specified in SKEL's `pose_limits` table Keller et al. (2023). We solve Eq. equation 14 with L-BFGS using a strong-Wolfe line search and a step size of 1 over 64-frame windows, each optimized for 20 iterations, with $\lambda_\Psi = 10$. No temporal or additional pose priors are used.

Given the optimized parameters $(\boldsymbol{\theta}^\star_t, \mathbf{r}^\star_t)$, the baseline prediction in the common biomechanical evaluation space is the full set of 24 SKEL anatomical joints:

$$\hat{\mathbf{J}}^{b,\mathrm{opt}}_t = \mathcal{S}(\boldsymbol{\theta}^\star_t, \mathbf{0}, \mathbf{r}^\star_t) \in \mathbb{R}^{24\times 3}. \tag{16}$$

The reported runtime includes only the optimization in Eq. equation 14.

OmniMM. Given the 77-D parametric-motion input, OmniMM directly generates the 57-D biomechanical-motion sequence $\hat{\mathbf{x}}^b_{1:T}$ without per-sequence optimization. For evaluation, we map the prediction to the same 24-joint SKEL space as the reference:

$$\hat{\mathbf{J}}^{b,\mathrm{ours}}_{1:T} = \mathcal{G}_{\mathcal{B}}(\hat{\mathbf{x}}^b_{1:T}). \tag{17}$$

This biomechanics-to-SKEL projection is used only for evaluation and is excluded from the reported fitting time.

Scoring. For $\mathcal{M}2\mathcal{B}$, the source is the parametric motion, represented by 22 SMPL joints $\mathbf{J}_t^m \in \mathbb{R}^{22\times3}$, and the target is the biomechanical motion, represented in the common 24-joint SKEL anatomical space as $\mathbf{J}_t^{b,\mathrm{gt}} \in \mathbb{R}^{24\times3}$. Accordingly, all metrics are computed in the target space between the predicted joints $\hat{\mathbf{J}}_t^b$ and the ground-truth target $\mathbf{J}_t^{b,\mathrm{gt}}$. We report MPJPE after per-frame pelvis alignment, PA-MPJPE after per-frame Procrustes alignment, and acceleration error over the 24 joints. Metrics are averaged over frames and then over the 1,561 AMASS test clips, with medians reported alongside means. Distances are reported in meters and acceleration in $\mathrm{m/s^2}$ at 20 fps.

B.3.2 $\mathcal{B}2\mathcal{M}$

Reference. Both prediction and reference are evaluated as 22 SMPL joint positions, so no fitting is required to construct the reference. Let $\mathbf{x}_t^m \in \mathbb{R}^{77}$ denote the SMPL-H motion representation at frame t, and let $\mathcal{G}_\mathcal{M}$ denote the fixed decoder that reconstructs the SMPL-H global orientation, body pose, and root translation from this representation and applies the SMPL forward pass with template shape $\boldsymbol{\beta} = \mathbf{0}$:

$$\mathbf{J}_t = \mathcal{G}_\mathcal{M}(\mathbf{x}_t^m) \in \mathbb{R}^{22\times3}. \tag{18}$$

The ground-truth reference is therefore $\mathbf{J}_t^{\mathrm{gt}} = \mathcal{G}_\mathcal{M}(\mathbf{x}_t^{m,\mathrm{gt}})$, with no optimization required.

Optimization-Based Baseline. Let $\mathbf{q}_t \in \mathbb{R}^{49}$ denote the ground-truth biomechanical motion of the same clip in OpenSim coordinates, and let $\mathrm{FK}_k^{\mathrm{bsm}}(\mathbf{q}_t) \in \mathbb{R}^3$ denote the exact forward kinematics of BSM segment k, expressed in SKEL's y-up frame. We use the same SKEL forward-kinematics function $\mathcal{S}(\boldsymbol{\theta}_t, \mathbf{0}, \mathbf{r}_t)$ defined in the $\mathcal{M}2\mathcal{B}$ evaluation, with pose $\boldsymbol{\theta}_t \in \mathbb{R}^{46}$, root translation $\mathbf{r}_t \in \mathbb{R}^3$, and universal shape fixed to $\boldsymbol{\beta} = \mathbf{0}$. Let $\mathcal{C}_{\mathrm{BS}}$ denote the 24 name-matched correspondences between SKEL joints and BSM segments. We first fit SKEL to the biomechanical motion by solving

$$(\boldsymbol{\theta}_t^*, \mathbf{r}_t^*)_{t=1}^T = \arg\min_{\{\boldsymbol{\theta}_t, \mathbf{r}_t\}} \frac{1}{T|\mathcal{C}_{\mathrm{BS}}|} \sum_{t=1}^{T} \sum_{(j,k)\in\mathcal{C}_{\mathrm{BS}}} \left\| \mathcal{S}_j(\boldsymbol{\theta}_t, \mathbf{0}, \mathbf{r}_t) - \mathrm{FK}_k^{\mathrm{bsm}}(\mathbf{q}_t) \right\|_2^2. \tag{19}$$

This objective contains only the data term, without joint-limit, orientation, or temporal regularization. We solve Eq. equation 19 using L-BFGS with a strong-Wolfe line search and step size 1 over 64-frame windows. Each window is warm-started from the final frame of the preceding window and optimized for 20 iterations, matching the budget used by the $\mathcal{M}2\mathcal{B}$ optimization baseline. The variables are initialized directly from $\mathbf{q}_t$, with pelvis rotation and translation copied and the 43 anatomical angles assigned to the corresponding SKEL DoFs.

The fitted body is then mapped to SMPL joints using SKEL's own SMPL joint regressor $\mathbf{W}^{\mathrm{SKEL}} \in \mathbb{R}^{24\times V}$, which is provided with the model since the SKEL skin shares the SMPL mesh topology. Applying it to the fitted SKEL skin vertices $\mathbf{V}(\boldsymbol{\theta}_t^\star, \mathbf{0}, \mathbf{r}_t^\star) \in \mathbb{R}^{V\times3}$ and retaining the 22 body joints gives

$$\hat{\mathbf{J}}_t^{\mathrm{opt}} = \left[\mathbf{W}^{\mathrm{SKEL}}\mathbf{V}(\boldsymbol{\theta}_t^\star, \mathbf{0}, \mathbf{r}_t^\star)\right]_{1:22}. \tag{20}$$

Thus, the baseline prediction follows the same SMPL joint definition as the reference.

OmniMM. Given the 57-D biomechanical representation $\mathbf{x}_t^b$, OmniMM directly generates the 77-D parametric-motion representation $\hat{\mathbf{x}}_t^m$. The prediction is decoded using the same procedure as the reference,

$$\hat{\mathbf{J}}_t^{\mathrm{ours}} = \mathcal{G}_\mathcal{M}(\hat{\mathbf{x}}_t^m), \tag{21}$$

without per-sequence optimization. The reported runtime includes only model generation.

Scoring. We compute MPJPE after per-frame pelvis alignment, PA-MPJPE after per-frame Procrustes alignment, and acceleration error between $\hat{\mathbf{J}}_t$ and $\mathbf{J}_t^{\mathrm{gt}}$ over the 22 SMPL joints. Metrics are averaged over frames and across all 1,561 AMASS test clips. Distances are reported in meters and acceleration in $\mathrm{m/s^2}$ at 20 fps.

B.3.3 $\mathcal{T}2\mathcal{B}\mathcal{M}$

Given a caption, OmniMM jointly generates frame-synchronous parametric and biomechanical motion, $\hat{\mathbf{x}}_{1:T}^m \in \mathbb{R}^{T\times77}$ and $\hat{\mathbf{x}}_{1:T}^b \in \mathbb{R}^{T\times57}$. No ground-truth motion enters this metric and we instead evaluate whether the two generated streams describe the same underlying motion in the common 24-joint SKEL anatomical space.

Reference. We use the generated biomechanical stream as the reference. It is mapped to the 24-joint SKEL space using the biomechanical readout defined in Eq. equation 12:

$$\hat{\mathbf{J}}^{b,\mathrm{ref}}_{1:T} = \mathcal{G}_{\mathcal{B}}(\hat{\mathbf{x}}^{b}_{1:T}) \in \mathbb{R}^{T\times 24\times 3}. \tag{22}$$

Motion Side. Let $\tilde{\mathbf{x}}^{m}_{1:T}$ denote the root-aligned parametric motion. We first decode it to 22 SMPL joints using Eq. equation 18,

$$\hat{\mathbf{J}}^{m}_{t} = \mathcal{G}_{\mathcal{M}}(\tilde{\mathbf{x}}^{m}_{t}) \in \mathbb{R}^{22\times 3}. \tag{23}$$

We then map these joints to SKEL using the same objective and SKEL–SMPL correspondences defined for the $\mathcal{M}2\mathcal{B}$ optimization baseline in Eq. equation 14. For this evaluation mapping, each 64-frame window is optimized with L-BFGS in three stages: 30 iterations over pelvis rotation and translation using the pelvis, hip, and shoulder correspondences, 100 iterations over the full pose and root translation, and 50 refinement iterations. The resulting motion-side joints are

$$\hat{\mathbf{J}}^{b,m}_{t} = \mathcal{S}(\boldsymbol{\theta}^{m}_{t}, \mathbf{0}, \mathbf{r}^{m}_{t}) \in \mathbb{R}^{24\times 3}. \tag{24}$$

Scoring. We compare the motion-side joints $\hat{\mathbf{J}}^{b,m}_{t}$ against the biomechanical reference $\hat{\mathbf{J}}^{b,\mathrm{ref}}_{t}$ in the common 24-joint SKEL anatomical space. We report MPJPE after per-frame pelvis alignment, PA-MPJPE after per-frame Procrustes alignment, and acceleration error over the 24 joints. Metrics are averaged over frames and then over the 1,561 AMASS test captions, with medians reported alongside means. Distances are reported in meters and acceleration in $\mathrm{m/s^2}$ at 20 fps. The reported runtime includes only the joint generation of the two streams.

B.3.4 Joint correspondence and anatomical limits

Joint correspondence. Table 11 specifies the correspondence set $\mathcal{C}$ used in Eq. equation 14. It contains 17 one-to-one pairs, with each SMPL and SKEL joint appearing at most once.

We construct $\mathcal{C}$ using the correspondence provided by the SKEL reference implementation Keller et al. (2023). The correspondence is designed for transferring rotations to SMPL pose-dependent blend shapes. Of the 24 SKEL entries, two (`radius_r` and `radius_l`) are assigned sentinel values and ignored. Four entries form many-to-one ankle correspondences, with both `talus` and `calcn` mapped to the corresponding SMPL ankle. The remaining 18 entries are one-to-one correspondences.

Our correspondence retains 14 of these one-to-one pairs and selects `talus` for each SMPL ankle, excluding `calcn` to enforce a unique positional match. We additionally pair SMPL *spine3*, which is unpaired in the reference correspondence, with SKEL *thorax* to retain alignment. In total, 16 pairs follow the reference correspondence and one additional trunk pair is introduced by our method.

We exclude four reference pairs: *spine1→lumbar_body*, *spine2→thorax*, and the two collar-to-scapula pairs. Although these topological correspondences are appropriate for rotation transfer, they exhibit substantial positional offsets. We therefore omit them from the positional evaluation to avoid imposing inconsistent joint-location targets.

SMPL *neck* and both hand joints remain unpaired, following the reference correspondence. The seven SKEL joints without direct SMPL supervision—`calcn_r`, `calcn_l`, `lumbar_body`, `scapula_r`, `scapula_l`, `radius_r`, and `radius_l`—are instead determined by forward kinematics and are still included in the evaluation.

B.4 Data processing details

We detail data processing in this section.

SMPL-to-Biomechanics Fitting Time. Fitting a 10-second SMPL sequence to the biomechanical representation takes approximately 24 minutes on an Intel Xeon E5-2620 2.00 GHz CPU. We therefore parallelize the fitting process extensively to construct the dataset at scale.

AMASS We directly use the source AMASS SMPL-H sequences and apply our biomechanical motion data construction pipeline to obtain the corresponding biomechanical motion data. We manually discard several low-quality sequences. For the associated motion representation and text annotations, we directly use the corresponding annotations from HumanML3D Guo et al. (2022a).

Table 11: The 17 joint pairs in the correspondence set $\mathcal{C}$ used in Eq. equation 14. Pairs marked with † are also used in the initial root-only fitting stage.

SMPL		SKEL	
0	pelvis†	0	pelvis
1	L_hip†	6	femur_l
2	R_hip†	1	femur_r
4	L_knee	7	tibia_l
5	R_knee	2	tibia_r
7	L_ankle	8	talus_l
8	R_ankle	3	talus_r
10	L_foot	10	toes_l
11	R_foot	5	toes_r
9	spine3	12	thorax
15	head	13	head
16	L_shoulder†	20	humerus_l
17	R_shoulder†	15	humerus_r
18	L_elbow	21	ulna_l
19	R_elbow	16	ulna_r
20	L_wrist	23	hand_l
21	R_wrist	18	hand_r

Retargeted 100 STYLE We first retrieve SMPL-H sequences from the BVH files of the source 100STYLE dataset Mason et al. (2022), and then apply our biomechanical motion construction pipeline to obtain the corresponding biomechanical motion. During data processing, we observe that many repeated motions share the same semantic labels. For example, multiple sequences are labeled with `Airplane + BR`. For each group of overlapping samples, we randomly select 2–3 representative sequences.

Source-text annotations in 100STYLE are noisy: each clip contains up to 50 free-form descriptions, many of which are duplicated, overly generic, or unrelated to the captured motion. We therefore clean the annotations through a three-stage pipeline.

1. *Lexical deduplication*. We first normalize each annotation into a set of content words by lowercasing, removing punctuation, filtering stop words, and applying lightweight lemmatization to canonicalize morphological variants. We then sort the resulting sentences by descending length and greedily retain a candidate only if its Jaccard similarity to every previously retained sentence is below 0.7. This length-prioritized strategy favors longer and more informative descriptions while removing short near-duplicates.
2. *Semantic deduplication*. To remove residual paraphrases that differ lexically but convey the same meaning, we encode each remaining sentence using the pretrained `sentence-transformers/all-MiniLM-L6-v2`[1] model. We then apply the same greedy length-prioritized selection, retaining a candidate only if its cosine similarity to every previously retained sentence is below 0.85. This step collapses semantically redundant descriptions that cannot be reliably detected by lexical matching alone.
3. *Anchor-based filtering*. Finally, we perform anchor-based filtering by converting structured trial labels into canonical motion descriptions, which serve as semantic anchors for each clip. We retain only source annotations that are semantically aligned with the corresponding anchor. This procedure yields two to four annotations per clip, with the first two guaranteed to describe the captured motion.

Example anchor-generated descriptions are shown below:

> `ArmsBySide_BW_08` → *"a person walks backwards with arms hanging by the sides"* / *"the person is walking backwards with arms hanging by the sides"*
>
> `KarateChop_TR1_09` → *"a person moves throwing karate chops with the arms"* / *"the person moves throwing karate chops with the arms"*

[1] `https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2`

The full anchor dictionary is used during preprocessing to convert structured 100STYLE trial labels into canonical motion descriptions, while representative examples are shown in Table 12.

Table 12: **Examples of Anchor Construction Rules of 100STYLE Dataset Text Annotation.** Each anchor is generated using the template `a person {direction} {motion}`.

Type	Code / Label	Phrase
Direction	`BW`	walks backwards
Direction	`FR`	runs forward
Direction	`ID`	stands in place
Direction	`TR1/TR2/TR3`	moves
Motion	`ArmsBySide`	with arms hanging by the sides
Motion	`KarateChop`	throwing karate chops with the arms
Motion	`HighKnees`	lifting the knees high with each step
Motion	`Robot`	with stiff, mechanical, robot-like movements
Motion	`WalkingStickLeft`	leaning on a walking stick held in the left hand

CombatMotion We obtain SMPL-H motion sequences and source-text annotations by converting CombatMotion samples from MotionMillion Fan et al. (2025), and further construct the corresponding biomechanical motion sequences using our pipeline. Similar to 100STYLE, the source-text annotations associated with CombatMotion in MotionMillion are noisy and contain duplicated or semantically redundant descriptions. We therefore apply the same lexical and semantic deduplication procedures used for 100STYLE to clean the text annotations before training.

HumanSC3D We obtain SMPL-H motion sequences and text descriptions from MotionMillion Fan et al. (2025), similar to CombatMotion. We then construct the corresponding biomechanical motion sequences using our pipeline. Since the source-text annotations are noisy and often contain duplicated or semantically redundant descriptions, we apply the same lexical and semantic deduplication procedures used for 100STYLE to clean the text annotations.

TRUMANS The source TRUMANS motions involve human–scene interactions. We obtain single-person SMPL-H motion sequences from MotionMillion Fan et al. (2025) and construct biomechanics motion with our pipleine. Since TRUMANS sequences are typically long and complex, often composing multiple actions, their text annotations provide frame-level labels for individual motion segments. We therefore re-segment each long sequence into shorter clips containing several short actions according to the text frame labels. To increase linguistic diversity, we further apply a set of text templates to augment each annotation.

FineDance We obtain SMPL-H motion sequences from MotionMillion Fan et al. (2025) and construct the corresponding biomechanical motion sequences using our pipeline. The original FineDance dataset consists of 203 long-form choreography sequences. To support short-horizon language-conditioned motion modeling, we re-segment each dance into fixed-length 200-frame, i.e., 10-second, subsequences. We first generate candidate windows with a stride of 50 frames, filter them by the first-frame pelvis height to ensure grounded rendering, and then greedily select valid windows with at most 50% temporal overlap. This process yields 3,341 subsequences across the 203 dances.

Since FineDance provides only dance-level style metadata, we synthesize two template-based text descriptions for each dance and assign them to all of its subsequences, yielding 6,682 annotation lines in total:

1. A dancer performs a {style1_eng}-style {style2} dance sequence.
2. Someone executes a {style2} routine in the {style1_eng} dance tradition.

C Anatomical Representation Analysis

Sec. 6.1 analyzes the anatomical representation gap. Here, we provide the detailed evaluation protocol and metric definitions.

Table 13: Text templates used for TRUMANS grouped re-segmentation. For a group containing n motion segments, V_i denotes the third-person singular verb form (e.g., "opens"), G_i denotes the gerund form (e.g., "opening"), and A_i denotes the raw imperative action phrase (e.g., "Open the oven"). We define a_i as A_i with its first letter lowercased. For $n \geq 3$, "..." denotes the repeated connective for intermediate actions. ID denotes the template index.

ID	$n = 1$	$n = 2$	$n \geq 3$
1	The person V_1.	The person V_1, and then V_2.	The person V_1, then V_2, ..., and finally V_n.
2	A person V_1.	A person starts with G_1 and is followed by G_2.	A person starts with G_1, followed by G_2, ..., and ends with G_n.
3	The man V_1.	The man V_1, and after that V_2.	The man V_1, after that V_2, ..., and then V_n.
4	The person is G_1.	Beginning with G_1, the person finishes with G_2.	Beginning with G_1, the person continues with G_2, ..., and finishes with G_n.
5	A_1.	First, A_1. Then, A_2.	First, A_1. Then, A_2. ... Finally, A_n.
6	The motion includes G_1.	The motion includes G_1, followed by G_2.	The motion includes G_1, then G_2, ..., and ends with G_n.
7	A person performs: a_1.	A person performs: a_1, a_2.	A person performs: $a_1, a_2, \ldots, a_n$.
8	The person V_1.	After G_1, the person V_2.	After G_1, the person V_2, then V_3, ..., and lastly V_n.

Table 14: **Distribution of off-axis rotation magnitude.** Mean and median values are reported in degrees.

	L. Elbow		R. Elbow		L. Knee		R. Knee	
Source	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.
GT Parametric	10.71	8.03	12.05	9.56	10.79	9.24	8.13	6.57
Generated Parametric	11.01	9.09	12.25	10.80	9.85	8.46	7.58	6.24

C.1 HSMR Anatomical Plausibility

Following HSMR Xia et al. (2025), we report the percentage of frames in which the knee or elbow rotation exceeds its anatomical range by more than a tolerance $\tau \in \{10^\circ, 20^\circ, 30^\circ\}$. The metric is computed directly from the native joint coordinates of each representation. For SMPL-H, we follow the HSMR Euler conventions: knees use XZY order, with flexion constrained to $[0^\circ, 135^\circ]$ and the remaining axes constrained to zero; elbows use YZX order with the corresponding flexion and pronation/supination limits. For BSM, we evaluate the generalized coordinates directly, using knee and elbow flexion ranges of $[0^\circ, 135^\circ]$ and pronation/supination limits of $\pm 90^\circ$ and $\pm 67.5^\circ$, respectively.

C.2 Auxiliary Diagnostics

Off-Axis Rotation Magnitude. We define off-axis rotation using the same Euler decomposition and anatomical conventions as HSMR Xia et al. (2025). For the knee, HSMR uses an XZY decomposition, where X represents flexion, and the remaining Z and Y rotations lie outside the corresponding anatomical DoFs. We therefore define

$$e_{t,j}^{\text{off}} = \sqrt{\left(\theta_{t,j}^{Z}\right)^2 + \left(\theta_{t,j}^{Y}\right)^2} \tag{25}$$

For the elbow, HSMR uses a YZX decomposition, where Y represents flexion and X represents forearm pronation/supination. We therefore treat only the remaining Z component as off-axis:

$$e_{t,j}^{\text{off}} = \left|\theta_{t,j}^{Z}\right| \tag{26}$$

Hyperextension Rate. We orient each joint coordinate such that positive values correspond to flexion and define hyperextension as rotation beyond the lower anatomical bound of the principal flexion coordinate. For a tolerance δ, the hyperextension rate is

$$H_j(\delta) = \frac{1}{N} \sum_{n=1}^{N} \mathbf{1}\left[\theta_{n,j}^{\text{flex}} < -\delta\right] \tag{27}$$

where δ denotes the amount of extension beyond the anatomical lower bound.

We additionally report the mean and median of off-axis articulation values in Table 14.

C.3 Qualitative Results

More qualitative results are shown in Fig. 7-Fig. 13, and the supplementary videos.

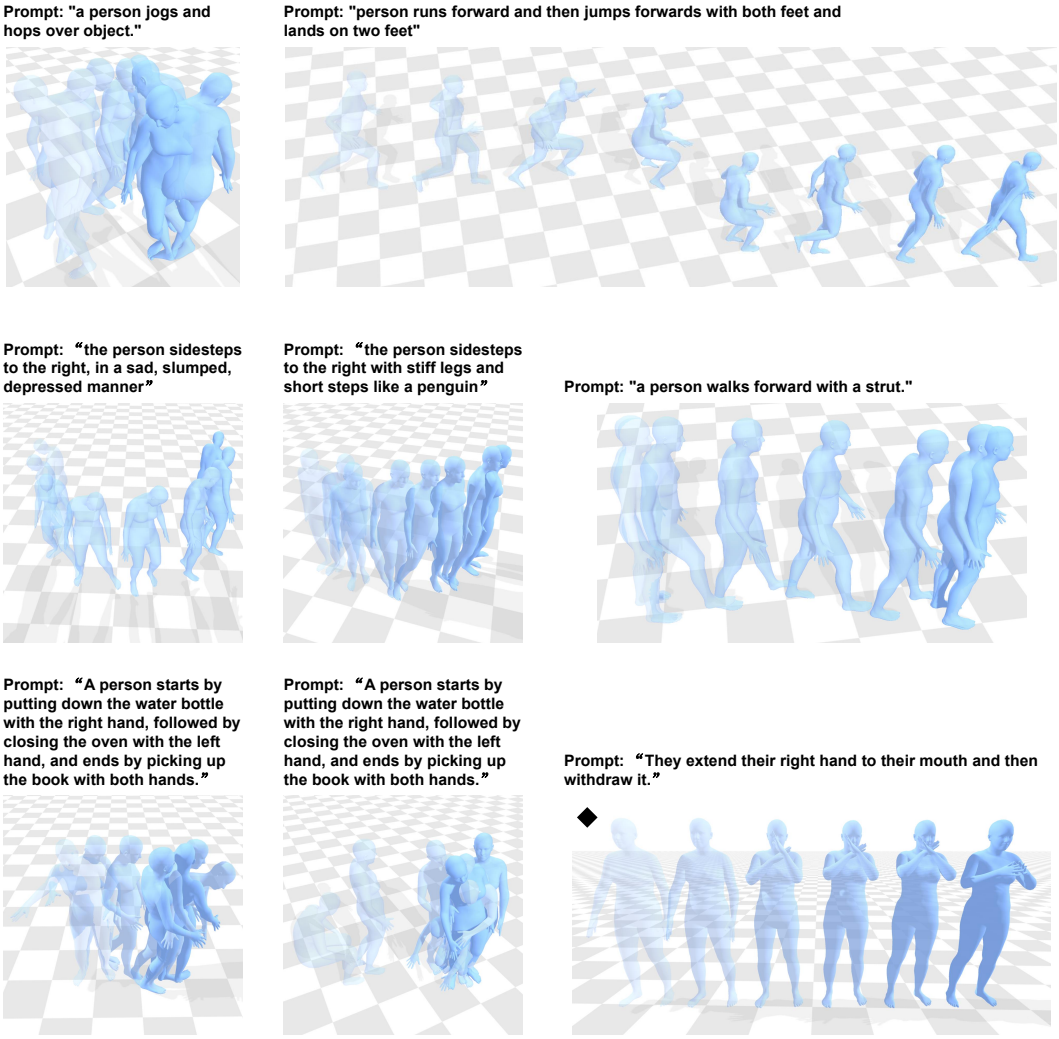


Figure 7: **Additional qualitative results on $\mathcal{T}2\mathcal{M}$.** Examples marked with ♦ depict in-place motion, with successive poses offset laterally for readability.

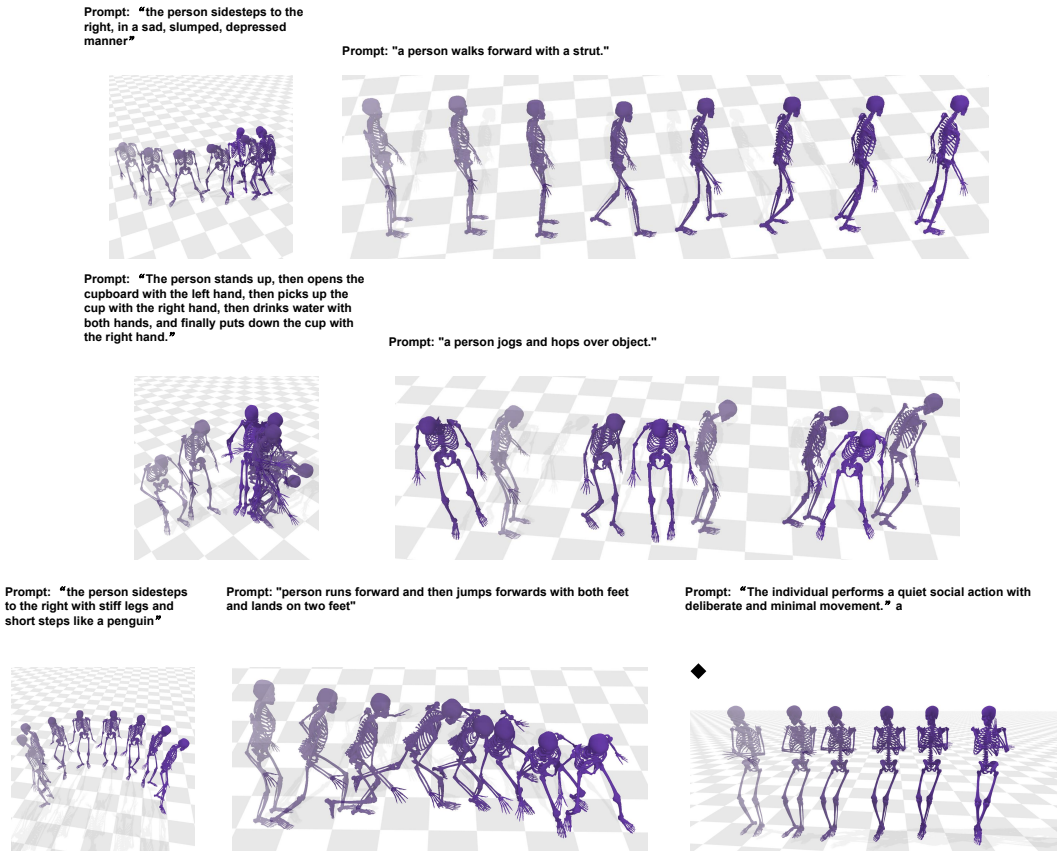


Figure 8: **Additional qualitative results on $\mathcal{T}2\mathcal{B}$.** Examples marked with ♦ depict in-place motion, with successive poses offset laterally for readability.

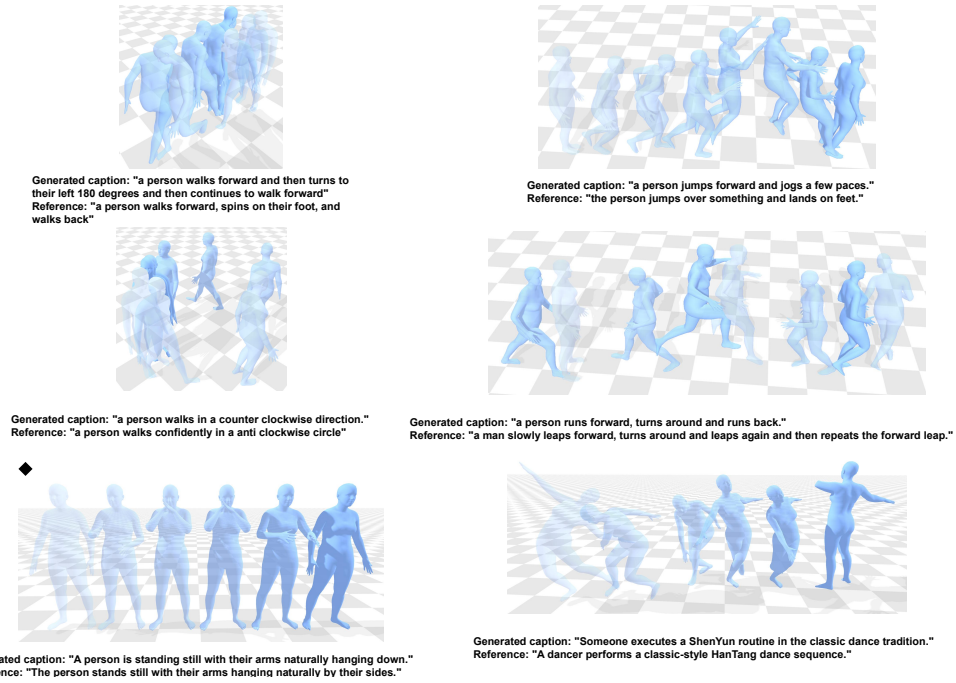
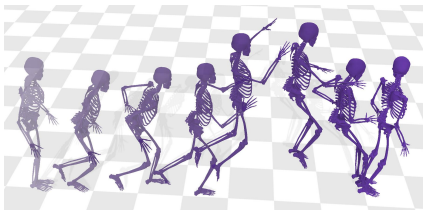


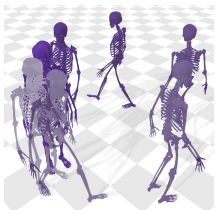
Figure 9: **Additional qualitative results on $\mathcal{M}2\mathcal{T}$.** Examples marked with ♦ depict in-place motion, with successive poses offset laterally for readability.



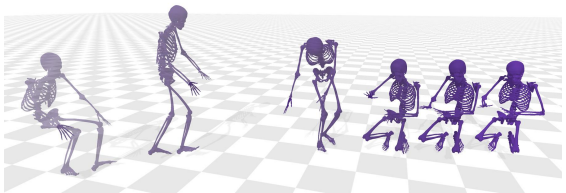
Generated caption: "a person runs forward and leaps off of their left foot and stops briefly when they land. they then start to run again"
Reference: "the person jumps over something and lands on feet."



Generated caption: "the person is jogging and running around."
Reference: "a person jogs and hops over object."

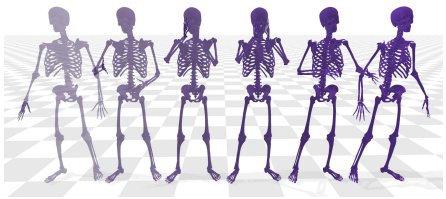


Generated caption: "a person walks in a circular motion but does not complete a full circle."
Reference: "a person walks confidently in a anti clockwise circle"

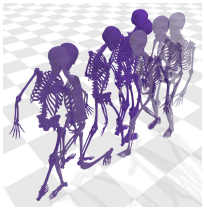


Generated caption: "The motion includes standing up, then sitting down, then opening the laptop with the right hand, and ending with typing on the laptop with the right hand."
Reference: "First, Stand up. Then, Squat down. Then, Open the laptop with both hands. Finally, Use the laptop with both hands."

♦



Generated caption: "The person is standing with their arms crossed, positioned in front of them."
Reference: "A person stands motionless with their arms relaxed."



Generated caption: "a man walks forward and then turns around."
Reference: "person is walking then turning around and walking back."

Figure 10: **Additional qualitative results on $\mathcal{B}2\mathcal{T}$.** Examples marked with ♦ depict in-place motion, with successive poses offset laterally for readability.

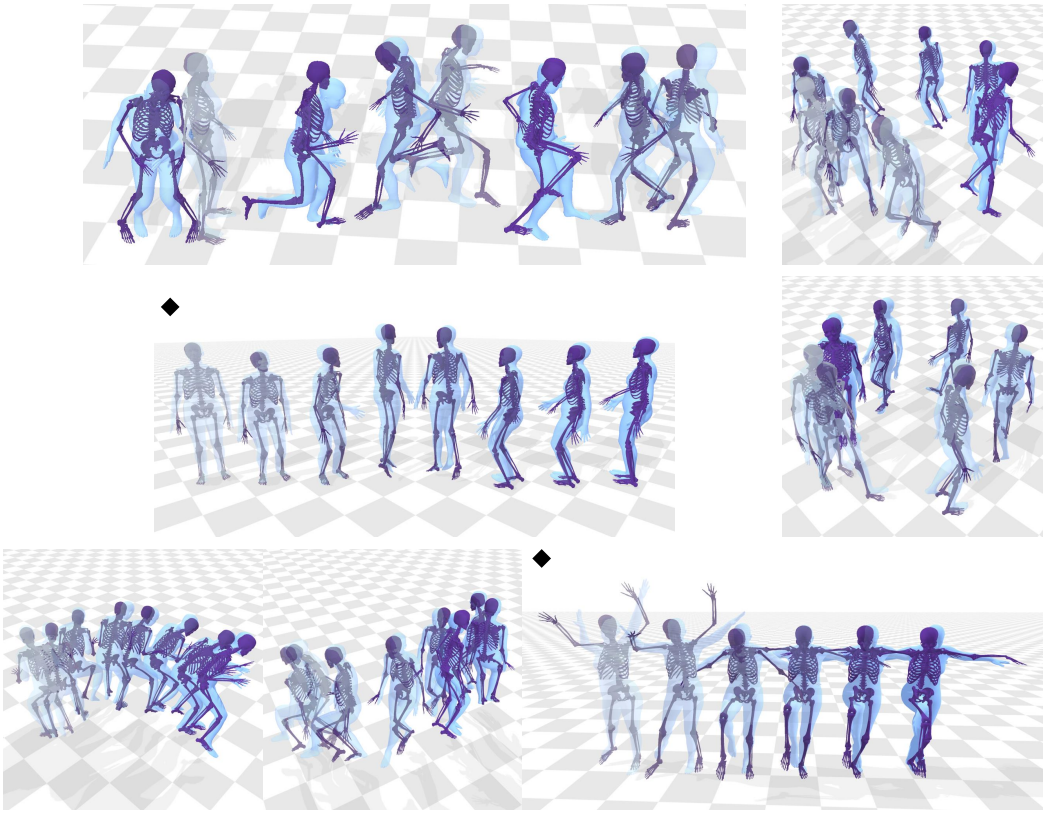


Figure 11: **Additional qualitative results on $\mathcal{M}2\mathcal{B}$.** Examples marked with ♦ depict in-place motion, with successive poses offset laterally for readability.

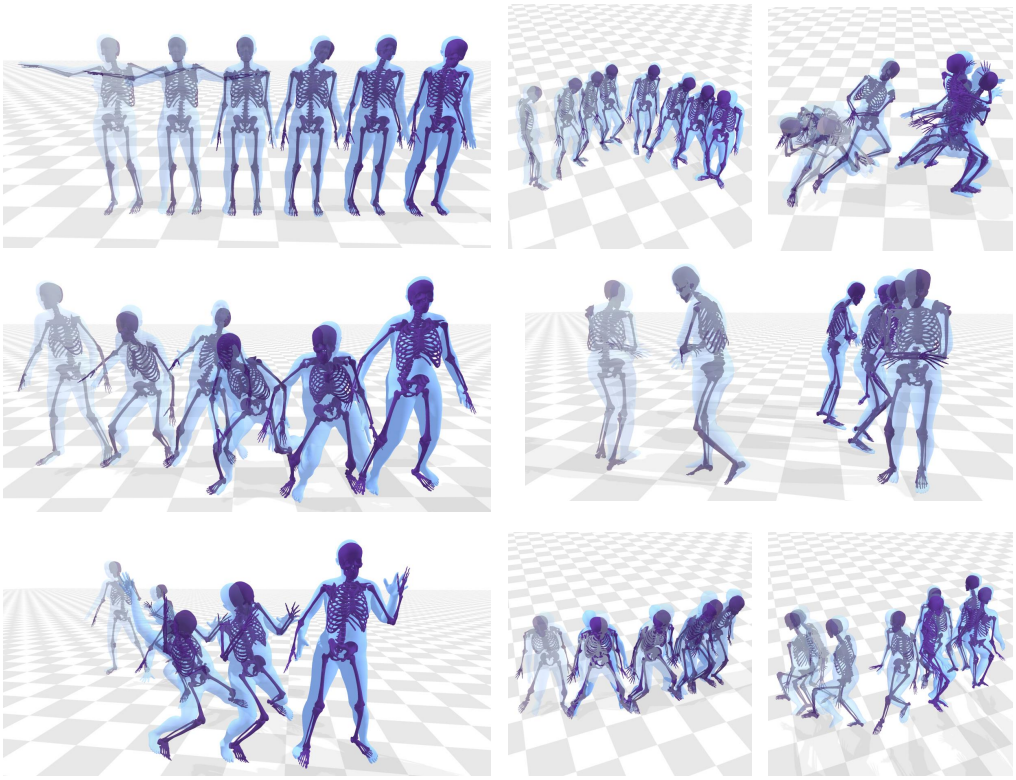
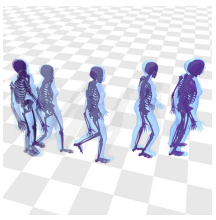
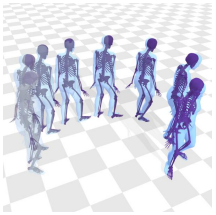


Figure 12: **Additional qualitative results on $\mathcal{B}2\mathcal{M}$.** Examples marked with ♦ depict in-place motion, with successive poses offset laterally for readability.

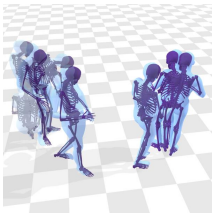
Prompt: "walking towards camera hips swinging lazily."



Prompt: "the person sidesteps to the right with stiff legs and short steps like a penguin"



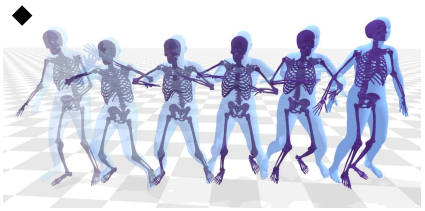
Prompt: "The person stands up, then opens the cupboard with the left hand, then picks up the cup with the right hand, then drinks water with both hands, and finally puts down the cup with the right hand."



Prompt: "A powerful attack is launched by a weapon against a man wielding a straight sword, executing a swift, one-handed strike with a steady root motion."



Prompt: "A man holding a crossbow is caught off guard by a swift left-handed stance."



Prompt: "They extend their right hand to their mouth and then withdraw it."

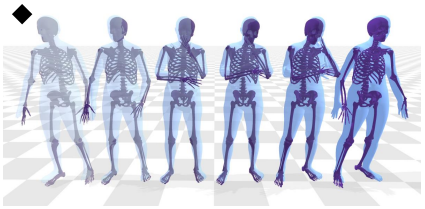


Figure 13: **Additional qualitative results on $\mathcal{T}2\mathcal{BM}$.** Examples marked with ♦ depict in-place motion, with successive poses offset laterally for readability.