

UniMotion: Unifying 3D Human Motion Synthesis and Understanding

Supplementary Material

In the following, we start with the supplementary video in Sec. A and discuss the details of training data in Sec. B. Then, we present the details of our evaluation setup in Sec. C, followed by implementation details in Sec. D, additional results in Sec. E and Sec. F. Finally, we demonstrate our model’s advantage over LLMs and other motion-to-text models in Sec. G.

A. Video with Qualitative Results

We provide videos to further explain our method and to present the results with animated motions, showing a clearer comparison across various tasks and against other baselines. Supplementary results can be found in the accompanying ZIP file.

B. Training Data

UniMotion is trained on an overlapping subset of BABEL [30] and HumanML3D [11], utilizing both sequence-level and frame-level text as input. Fig. 7 illustrates the data alignment and merging process. However, since these two datasets are independently labeled and cover different subsets of AMASS [24], they do not fully overlap. The overlapping portion comprises only 8,829 motion sequences (excluding left-right flipping), which represents approximately 30.25% of the HumanML3D dataset (23,384 sequences). This overlapped dataset includes motion sequences, sequence-level text descriptions, and frame-level text descriptions.

C. Evaluation Setup

In this section, we outline the details of the evaluation setup and how we run baselines under this setup.

For frame-level text-to-motion generation, we use BABEL frame-level text (in short-phrase format) as conditional input, which is also used as our test-time text input. To ensure a fair comparison with other baselines and to maintain consistency with the training data distribution, we use their pre-trained models on BABEL if available. However, our model is trained on a subset of the HumanML3D training split, which overlaps with the BABEL test split. Consequently, we generate a joint test set, excluding training sequences from both. Finally, the test set contains 358 sequences and 998 sub-sequences of motion segments. Our test, train, and validation split will be made available alongside our code and models upon publication.

TEACH. For TEACH [1] we use the pre-trained model supplied by the authors on their website, which was trained on

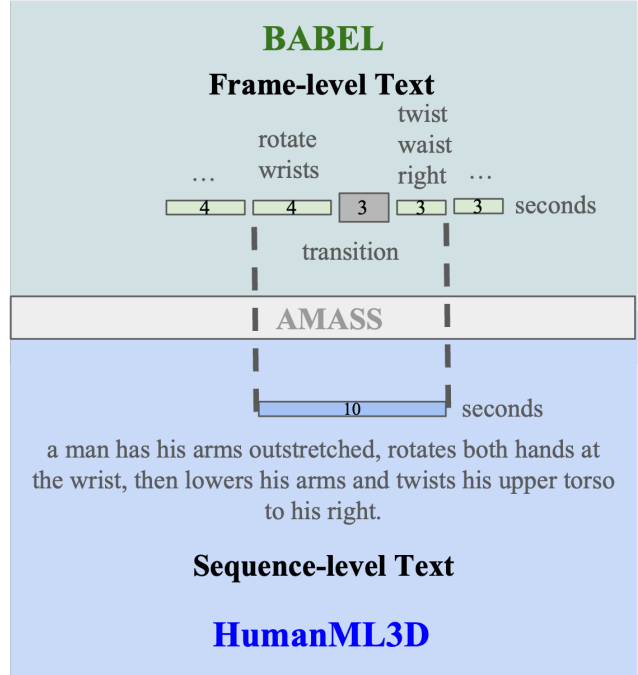


Figure 7. We merged HumanML3D and BABEL based on their time correspondence with AMASS. Each sequence (approximately 1-10 seconds) in HumanML3D includes 3-4 sequence-level annotations in sentence format, as illustrated in the blue area. In contrast, BABEL provides separate annotations for atomic actions with varying lengths, where the text labels are primarily short phrases aligned at the frame level, as shown in the green area.

BABEL. Since TEACH can not be applied to text segments with very few frames, we set the minimum size of each evaluation sequence to 8 frames.

PriorMDM. For PriorMDM [33], we compare DoubleTake with our method. To fairly compare DoubleTake with our method, we use the “Babel_TransEmb_GeoLoss” pre-trained model, as our local text input is based on the BABEL dataset. When feeding motion crops into DoubleTake, we specify the length of each motion crop. In DoubleTake’s default setup, the handshake size is set to 20 and the blending window size to 10, resulting in a minimum motion crop length of 70. If a motion crop is shorter than 70, the method automatically pads it to this length. However, many motion crops in our test set are shorter than 70, which would cause significant discrepancies between the input and output motion lengths. To maintain similar input and output sizes, we modify the handshake size to 2 and the blending window size to 1. The results under this setup are shown in Table 1.

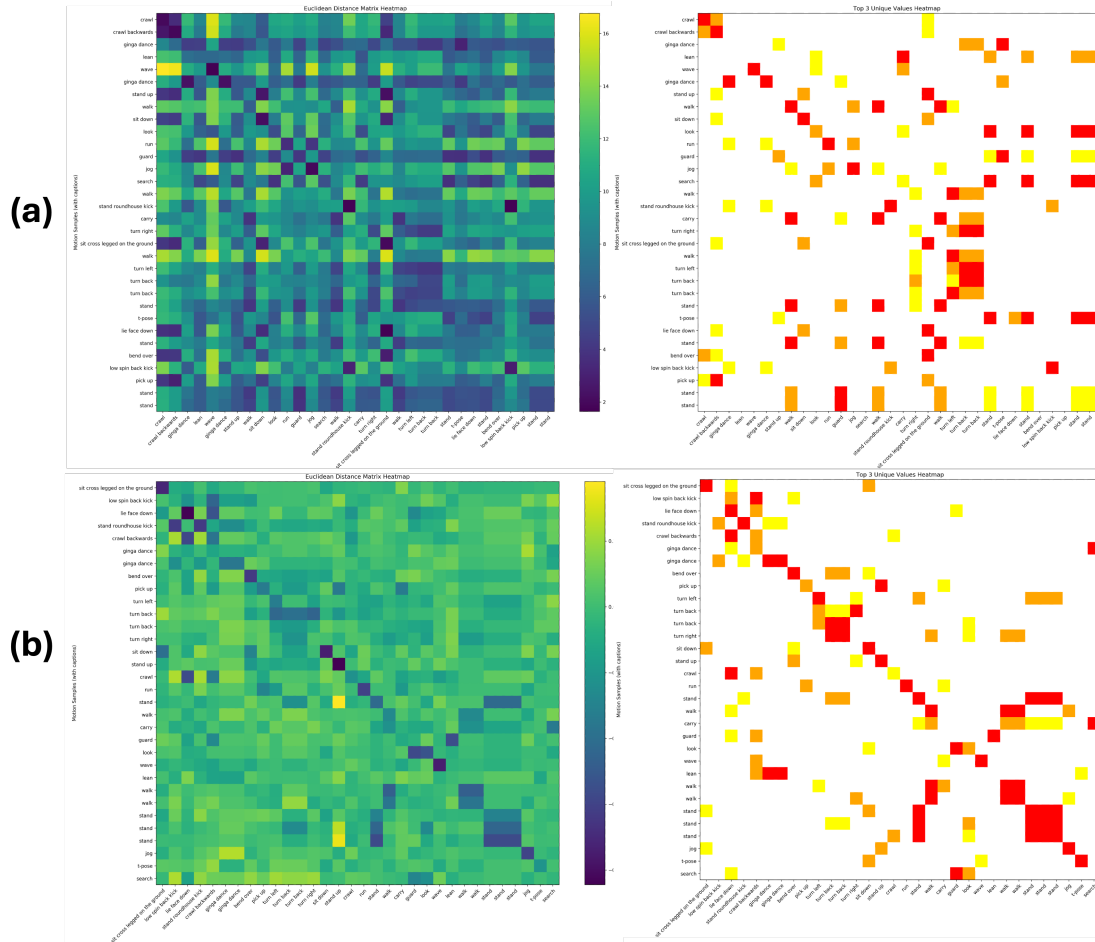


Figure 8. **The comparison between ground-truth motion-text matching** in the joint embedding spaces of Guo et al.’s model (a) and TMR++ (b). **Left:** The heatmap shows the paired motion-text distances, where darker shades indicate smaller distances. The vertical axis represents motion samples, while the horizontal axis represents text samples. **Right:** The top-3 R-precision scores are displayed for each row, indicating the closest 3 texts to each motion. Red denotes the top 1 match, orange the top 2, and yellow the top 3. If the texts are identical, they are only counted as one.

STMC. For an entire motion sequence, STMC [28] allows specifying the body part for each individual subsequence of motion. To align with our setup, we set the corresponding body part to include all body parts for each motion crop when feeding the motion into STMC.

FlowMDM. To ensure a fair comparison with our method, we use the human motion compositions with the pre-trained BABEL model for the FlowMDM [3] method. Since FlowMDM is designed to generate motion compositions seamlessly, there is no need to specify any transition length between atomic motions. Therefore, we directly input the frame-level texts and corresponding lengths, consistent with the input format used for our model.

Evaluation metrics. For the evaluation metrics—**Semantic Correspondence** (R-precision, M2T score, M2M score) and **Realism** (FID, Diversity)—we

use TMR++ instead of the commonly used motion and text embedding model from Guo et al. [11]. This choice is driven by the need to evaluate models trained across different datasets and to assess performance at multiple levels of generated motion (per-crop vs. per-sequence).

For per-crop semantic correctness, we focus on evaluating the alignment of atomic motion crops with their corresponding input text, formatted as BABEL. Additionally, we assess the overall realism of sequence-level motion across crops, which aligns with HumanML3D’s sequence-level evaluation. The evaluation model aims to establish a joint latent space for motion and text, performing matching between them based on distance within this shared space.

The commonly used model from Guo et al. [11] is trained solely on HumanML3D. To evaluate BABEL pre-trained models, Shafir et al. [33] retrained this model on BABEL data, and FlowMDM relies on these models

for separate evaluations on each dataset. STMC utilizes TMR [27], a retrieval model that demonstrates a better joint latent space compared to the classic evaluation model used by MDM, especially in terms of text-motion distance for ground-truth motion-text pairs. However, TMR is also trained only on HumanML3D, which limits its ability to accurately evaluate both crop-level motions and BABEL text, as well as sequence-level realism.

To address these limitations, we employ the latest model, TMR++ [4], which is trained across datasets and delivers highly accurate matching results between ground-truth motion and text, whether in BABEL format (subsequence level, short text phrases) or HumanML3D format (sequence-level, text descriptions in sentences).

For a quantitative comparison, please refer to Table 3, which evaluates ground-truth motion and text. For qualitative analysis, see Fig. 8, which presents a heatmap of the matching distance across a random sample of 32 batches.

Method	Training Set	Per-crop semantic correctness		
		R-Prec@1 \uparrow	R-Prec@2 \uparrow	R-Prec@3 \uparrow
Guo et al [11]	HumanML3D	0.281 \pm 0.005	0.438 \pm 0.004	0.539 \pm 0.006
TMR++[4]	HumanML3D+BABEL	0.520 \pm 0.013	0.659 \pm 0.008	0.735 \pm 0.008

Table 3. **Ground-truth matching score comparison across evaluation models.** In this table, we compare the matching scores across different evaluation models for ground-truth motion and text, averaging over batches of 32 random samples. The results demonstrate that TMR++ is a more reliable model within our evaluation setup.

D. Implementation Details

We provide more details about the implementation of our model. We extend the MDM [37] framework to separate time steps for motion and frame-level text, and adjust the input to accept the temporal alignment of both the motion vector and text embedding vector. The model is retrained from scratch using the merged overlapping dataset, with hyperparameters consistent with those suggested by Tevet et al. [37].

For frame-level text, we use the same CLIP model as used in MDM to generate embeddings. We then applied PCA to condense the dimensionality from 256 to 51, preserving approximately 70% of the original variance. Our model predicts both the clean motion and the condensed CLIP embeddings for the frame-level texts. To output the texts, we use K-nearest neighbors (KNN) to match the output CLIP embeddings in a pre-computed database. This approach effectively matches nearby CLIP embeddings to the corresponding closest text even with a small variance.

For the training and sampling algorithm, please refer to Algorithm 1, 2, 3.

Algorithm 1 Training

```

1: repeat
2:    $\mathbf{x}_0, \mathbf{y}_0, c \sim q(\mathbf{x}_0, \mathbf{y}_0, c)$ 
3:    $c = \emptyset$  with probability 10%
4:    $t^x, t^y \sim \text{Uniform}(\{1, 2, \dots, T\})$ 
5:    $\epsilon^x, \epsilon^y \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
6:   Let  $\mathbf{x}_{t^x} = \sqrt{\alpha_{t^x}}\mathbf{x}_0 + \sqrt{1 - \alpha_{t^x}}\epsilon^x$ 
7:   Let  $\mathbf{y}_{t^y} = \sqrt{\alpha_{t^y}}\mathbf{y}_0 + \sqrt{1 - \alpha_{t^y}}\epsilon^y$ 
8:   Take gradient step on  $\nabla_{\theta} \|\epsilon_{\theta}(\mathbf{x}_{t^x}, \mathbf{y}_{t^y}, t^x, t^y, c) - [\mathbf{x}_0, \mathbf{y}_0]\|_2^2$ 
9: until converged

```

Algorithm 2 Sampling \mathbf{x}_0 conditioned on \mathbf{y}_0 (similar for sampling \mathbf{y}_0 conditioned on \mathbf{x}_0 , with or without conditioning on c).

```

1:  $\mathbf{x}_0^T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
2:  $c = \emptyset$  or user specify
3: for  $t = T, \dots, 1$  do
4:    $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
5:    $\mathbf{x}_0^{t-1} = \epsilon_{\theta}(\sqrt{\alpha_{t^x}}\mathbf{x}_0^t + \sqrt{1 - \alpha_{t^x}}\epsilon, \mathbf{y}_0, t, 0, c)$ 
6: end for
7: return  $\mathbf{x}_0$ 

```

Algorithm 3 Joint sampling of $\mathbf{x}_0, \mathbf{y}_0$ (with or without condition on c)

```

1:  $\mathbf{x}_0^T, \mathbf{y}_0^T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
2:  $c = \emptyset$  or user specify
3: for  $t = T, \dots, 1$  do
4:    $\epsilon^x, \epsilon^y \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
5:    $\mathbf{x}_0^{t-1}, \mathbf{y}_0^{t-1} = \epsilon_{\theta}(\sqrt{\alpha_{t^x}}\mathbf{x}_0^t + \sqrt{1 - \alpha_{t^x}}\epsilon^x, \sqrt{\alpha_{t^y}}\mathbf{y}_0^t + \sqrt{1 - \alpha_{t^y}}\epsilon^y, t, t, c)$ 
6: end for
7: return  $\mathbf{x}_0, \mathbf{y}_0$ 

```

E. More Experiment Results

We provide only a subset of the metrics for semantic correspondence and motion realism in the main paper due to space constraints. Here, we provide the complete evaluation.

Semantic correspondence. Tab. 4 lists all three R-precision scores, demonstrating that our method outperforms all baseline methods. These results are consistent with our conclusions in the experiment section of the main paper.

Realism. Tab. 5 includes FID and Diversity scores calculated using the evaluation model from Guo et al. [11] for reference. Note that at the crop level, this model provides less stable evaluations because it was trained only on

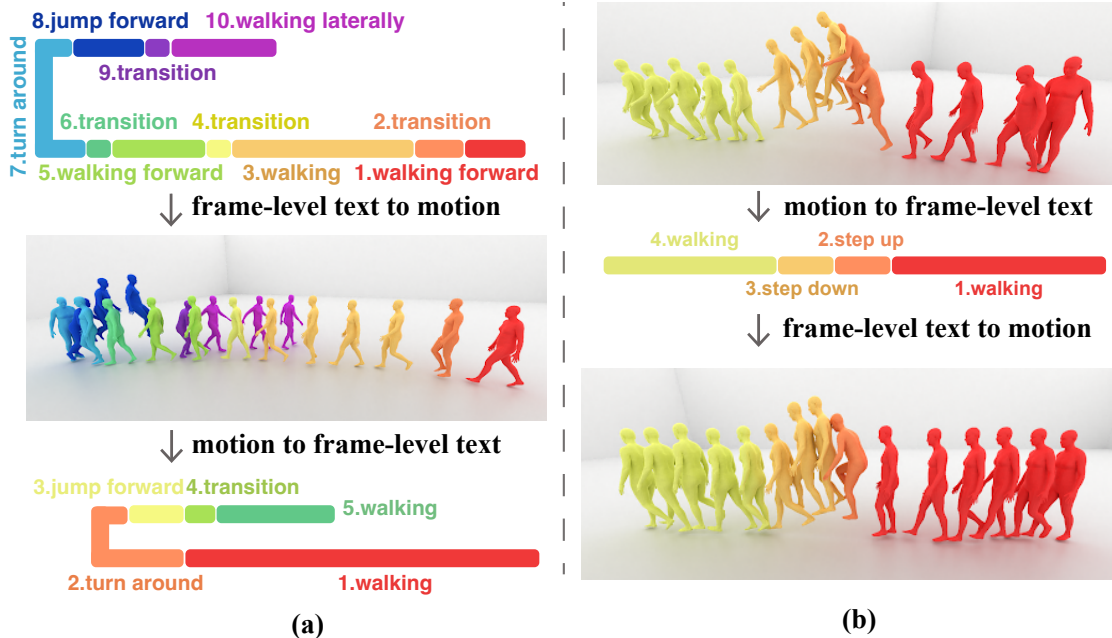


Figure 9. **Text variation (a) and motion variation (b)** are direct applications that leverage the two conditional distributions modeled by UniMotion. Motion variation (b) is achieved by generating frame-level text descriptions from a motion sequence, and then using these descriptions to create a new, semantically similar motion with different content. Text variation (a) is produced by reversing this process to create diverse text annotations.

Method	Training Set	Input	Per-crop semantic correctness				
			R-Prec@1 \uparrow	R-Prec@2 \uparrow	R-Prec@3 \uparrow	M2T \uparrow	M2M \uparrow
GT	-	-	0.520 \pm 0.013	0.659 \pm 0.008	0.735 \pm 0.008	0.663 \pm 0.000	1.000 \pm 0.000
TEACH	BABEL	f	0.375 \pm 0.008	0.516 \pm 0.007	0.588 \pm 0.007	0.623 \pm 0.001	0.575 \pm 0.000
DoubleTake	BABEL	f	0.332 \pm 0.013	0.467 \pm 0.013	0.544 \pm 0.013	0.602 \pm 0.002	0.560 \pm 0.001
STMC	HML	f	0.321 \pm 0.009	0.452 \pm 0.012	0.528 \pm 0.012	0.599 \pm 0.000	0.616 \pm 0.010
FlowMDM	BABEL	f	0.389 \pm 0.009	0.532 \pm 0.014	0.618 \pm 0.007	0.631 \pm 0.002	0.652 \pm 0.001
Ours	BABEL	f	0.394 \pm 0.010	0.552 \pm 0.018	0.636 \pm 0.017	0.633 \pm 0.004	0.677 \pm 0.002
Ours	HML-BABEL	f	0.427 \pm 0.011	0.587 \pm 0.012	0.668 \pm 0.009	0.643 \pm 0.002	0.698 \pm 0.002
Ours	HML-BABEL	f + s	0.450 \pm 0.018	0.593 \pm 0.008	0.679 \pm 0.006	0.644 \pm 0.001	0.706 \pm 0.002

Table 4. **Per-crop semantic correctness evaluation for frame-level Text2Motion generation.** **Training Set** specifies the dataset used for training, including BABEL, HumanML3D(HML), or the union/intersection of HML and BABEL. **Input** specifies the type of text input. **f**: frame-level text, **s**: sequence-level text. **f+s** demonstrates that combining multi-level conditioning signals can enhance model performance in terms of semantic correspondence. Symbols like \uparrow indicates that higher, lower, or values closer to the ground truth (GT) are better, respectively. The evaluation is repeated 10 times, and \pm indicates the 95% confidence intervals.

HumanML3D, which contains only sequence-level motions. Consequently, FID and Diversity scores from TMR++ offer a more reliable assessment. At the sequence level, both evaluation models yield consistent results. For simplicity and consistency, the main paper presents only FID_TMR++ and Diversity_TMR++.

F. More Applications

Due to space limitations, we only present part of applications in the main paper. Here, we showcase two addi-

tional applications that are made possible exclusively by our multi-task model. Similar to UniDiffuser [2], UniMotion naturally supports applications such as motion variation and text variation. For **motion variation**, given a motion sequence, we first perform the motion-understanding task to generate frame-level text descriptions aligned with the motion. We then use this frame-level text as input for text-to-motion generation, resulting in a new motion that retains similar semantics but with different content. For **text variation**, we reverse the process to produce fine-grained text

Method	Training Set	Input	Per-crop Realism				Per-seq Realism			
			FID ↓	Diversity →	FID_tmr++ ↓	Diversity_tmr++ →	FID ↓	Diversity →	FID_tmr++ ↓	Diversity_tmr++ →
GT	-	-	0.000±0.000	8.823±0.067	0.000±0.000	1.375±0.005	0.000±0.000	9.296±0.086	0.000±0.000	1.391±0.003
TEACH	BABEL	f	2.557±0.016	7.879±0.119	0.155±0.001	1.340±0.003	3.577±0.025	7.605±0.066	0.304±0.001	1.344±0.003
DoubleTake	BABEL	f	2.820±0.127	8.248±0.102	0.195±0.002	1.332±0.005	5.619±0.268	7.350±0.074	0.353±0.002	1.337±0.004
STMC	HML	f	2.161±0.008	9.250±0.130	0.156±0.000	1.358±0.005	1.295±0.017	8.955±0.102	0.233±0.000	1.362±0.005
FlowMDM	BABEL	f	0.885±0.043	8.476±0.086	0.101±0.001	1.352±0.006	1.028±0.060	8.691±0.127	0.211±0.002	1.375±0.005
Ours	BABEL	f	1.206±0.079	9.007±0.141	0.087±0.002	1.366±0.009	0.791±0.091	8.890±0.159	0.180±0.004	1.374±0.002
Ours	HML-BABEL	f	0.506±0.024	8.979±0.095	0.071±0.001	1.372±0.005	0.401±0.030	8.956±0.123	0.150±0.001	1.378±0.003
Ours	HML-BABEL	f + s	0.487±0.021	9.040±0.118	0.066±0.002	1.373±0.009	0.299±0.023	8.978±0.095	0.133±0.004	1.381±0.006

Table 5. **Frame-level Text2Motion generation per-crop and per-sequence realism evaluation.** Crop-level realism measures the metrics within each atomic crop, while Seq-level realism measures the fidelity of the overall motion. Symbols ↓, and → indicate that lower, or values closer to the ground truth (GT) are better, respectively.

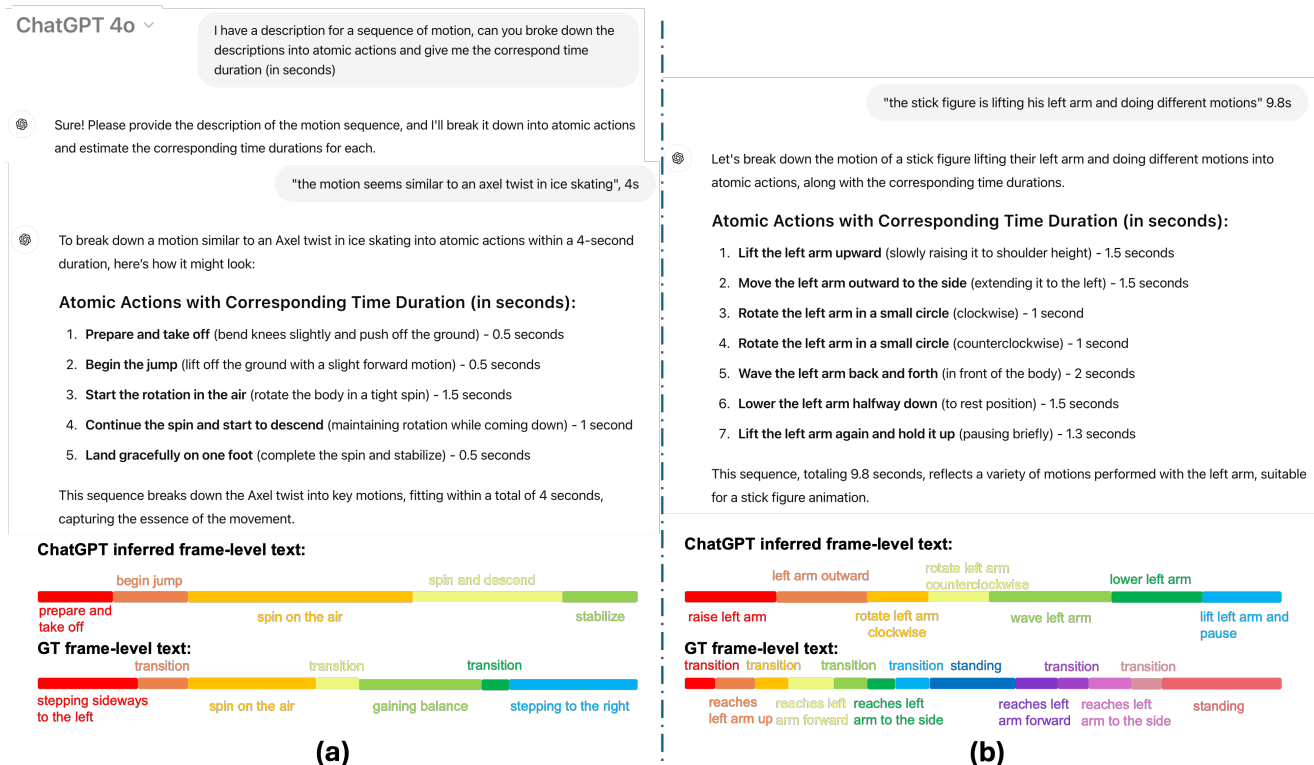


Figure 10. **Fine-grained motion understanding with LLM.** ChatGPT-4o is used to break down the ground-truth global descriptions into atomic motion and durations. However, there is no alignment between text and motion since the model doesn't take the motion as input.

annotation variance. Figure 9 provides examples of both motion and text variation. For animated results, please refer to the attached videos.

G. Motion-to-text Understanding Baselines

To establish baselines for our frame-level motion understanding sub-task, we initially attempted to use a large language model (LLM), ChatGPT, to decompose sequence-level inputs and assess potential outputs. However, due to the LLM's lack of motion awareness, the outputs were unreliable when the sequence-level information was vague or incomplete. Even with detailed sequence-level descriptions,

the LLM struggled to generate accurate timestamps due to the absence of motion data. Please refer to Fig. 10 for more details.

We then considered using LLM-based motion models like MotionGPT [17], which can process both motion data and text prompts (to request timestamps and atomic text labels). Despite this, MotionGPT also failed in this task. See Fig. 11 for further information.

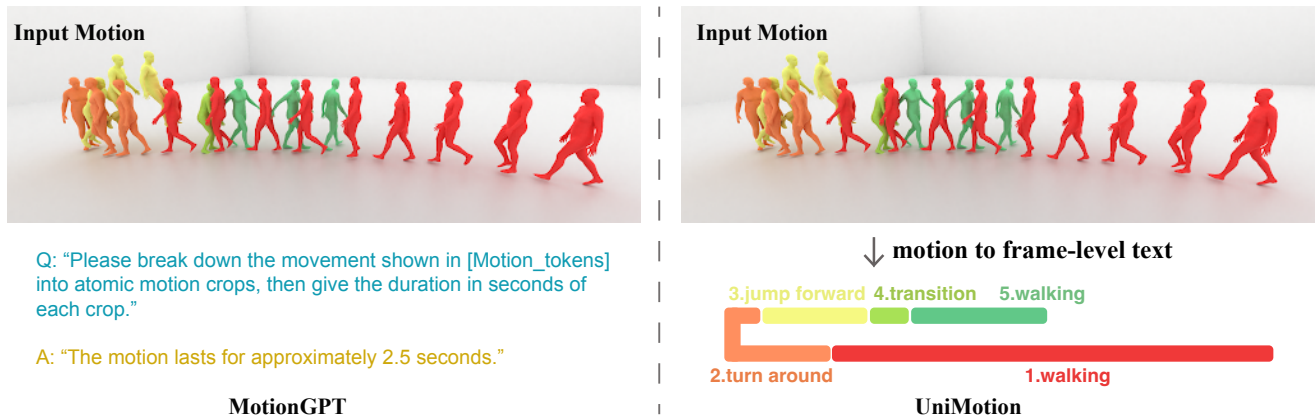


Figure 11. **Motion understanding comparison with MotionGPT [17].** MotionGPT is capable of performing multiple tasks, including motion captioning and question answering. We tasked both MotionGPT (left) and Unimotion (right) with understanding an input motion by breaking it down into motion segments. However, due to MotionGPT’s lack of temporal awareness, it was unable to successfully complete this task. Specifically, instead of answering with multiple motion segments, it just predicts an incorrect length for the whole sequence (A: “The motion lasts for approximately 2.5 seconds.”). In contrast, our model is the first to understand motion both semantically and temporally.