How Much Can Large Language Models Guide Body Movements of 3D Digital Human Agents?

Kunhang Li[1] Jason Naradowsky[1] Yansong Feng[2] Yusuke Miyao[1,3]
[1]The University of Tokyo [2]Peking University [3]NII LLMC
{kunhangli, narad, yusuke}@is.s.u-tokyo.ac.jp
fengyansong@pku.edu.cn

Abstract

We aim to explore the extent to which Large Language Models (LLMs) can guide 3D digital human agents in performing body movements without supervised training. Given an existing human model and a textual instruction, we prompt the LLM to generate a high-level plan decomposing the whole motion into consecutive steps, followed by specifying the positions of every body part in each step. We then render the animation by linearly interpolating the selected body part positions across steps. We evaluate the generated animations from a diverse set of motion instructions through both automatic and human evaluation, and find that LLMs generally struggle to recognize accurate body part positions. Specifically, LLMs struggle with complex motions with multiple steps and body parts, and complex body parts with more possible positions.

1 Introduction

Recent studies on Large Language Model (LLM)-based generative agents [1, 2] demonstrate their capability to produce open-ended behaviors in simulated environments. However, these agents typically express actions and states using text or emoji symbols in the absence of pre-defined animations. This limitation poses challenges for implementing digital human agents in 3D environments, where body movements are essential for natural interactions.

Modern text-conditioned human motion generation approaches employ generative models to synthesize realistic human body movements from natural language instructions [3, 4], but they often struggle with open-ended motion instructions due to overfitting to limited language-motion paired datasets [5, 6]. Existing work attempts to improve the generalization by using LLMs to extract specific motion-relevant information, such as active body parts [7], detailed body part descriptions [8] and keyframe coordinates [9]. However, these approaches typically utilize LLMs in a limited capacity, primarily as auxiliary components in their pipelines. We aim to explore to what extent we can generate animations using only the rich information provided by LLMs, potentially opening up new ways to create human motions in lack of pre-implemented animations.

In this paper, we present a framework that purely leverages LLMs to generate animations on SMPL [10], a standard 3D human model compatible with the Unity computer graphics engine. [1)] We provide a sketch of SMPL in Appendix A.1. Given the input motion instruction, the framework first uses LLMs to generate a structured animation plan with specific body part movements in natural language, then translates these descriptions into Unity codes specifying SMPL parameters using predefined rules, and finally renders the animation in Unity.

We conduct both automatic evaluation, where we calculate the accuracy of the LLM-selected positions against annotated oracle ones, and human evaluation, where annotators evaluate the animations both overall and body-part-wise. We have the following findings:

(I) **LLMs generally struggle to recognize accurate body part positions:** Compared with oracle standards, all tested LLMs exhibit significant shortcomings in body part position identification. The highest performer in human evaluation, Claude 3.5 Sonnet, trails the oracle's overall score by 1.28 points on a 5-point scale.

(II) **LLMs struggle with human motion complexity:** Our analysis reveals a negative correlation between motion complexity (defined by the number of moved body parts

1) https://unity.com/

across steps) and the accuracy of selected body part positions. LLMs demonstrate lower accuracy for body parts with more possible positions like the upper arm, compared to more constrained parts like the upper leg. Moreover, accuracy remains consistently higher for lower body components than their upper body counterparts, highlighting LLMs' difficulties with complex and flexible movements.

2 Animation Generation

LLMs primarily learn about human motions through natural language descriptions, rather than exact spatial coordinates or temporal quantities. We therefore evaluate LLMs' human motion knowledge by testing their ability to recognize appropriate body part positions described in natural language. Figure 1 illustrates our pipeline of animation generation.

Firstly, given the joint structure of the SMPL model M (Appendix A.1), we define a finite set of positions $\texttt{Text}(M)$ for preset body parts. Following the natural hierarchy of human motion from action sequences to body part movements [11], we implement a hierarchical querying framework Q that first decomposes the input motion instruction I [2] into sequential high-level steps, then iteratively specifies body part positions from $\texttt{Text}(M)$. The LLM uses this framework to acquire the animation plan P. While the position querying is conducted hierarchically, [3] we discuss different querying strategies in Appendix A.3.

$$P = \texttt{LLM}_Q(I, \texttt{Text}(M)) \tag{1}$$

Secondly, we use predefined `Rules` to convert P into Unity codes C by mapping the specified body part positions to local joint rotations on M, and inserting them into a code template T. [4]

$$C = \texttt{Rules}(P, T) \tag{2}$$

Finally, we render the animation A by executing C on M in Unity, where joint rotations are linearly interpolated between consecutive steps.

$$A = \texttt{Unity}(C, M) \tag{3}$$

2) We show our tested motion instructions in Appendix A.2.

3) For example, when we query the position of the left elbow, first we ask whether it is stright or bent. If it is bent, we further ask whether it is slightly bent in, bent in 90 degrees or fully bent.

4) We avoid a naive method of generating Unity codes from the given motion instruction in one go, since the codes can seldom be successfully executed in Unity, and the few generated animations are too low-quality for evaluation.

3 Evaluation

3.1 Automatic Evaluation

For each motion instruction, we first fix an oracle high-level plan by calibrating one high-level plan generated from GPT-4o, and manually annotate the oracle positions of all body parts across steps. Then we calculate the accuracy of the LLM-selected positions against the annotated oracle ones (**Body Part Position Accuracy**). We run each LLM three times to take the averaged accuracy.

The complexity of an annotated oracle motion is decided by the numbers of moved body parts across steps. Therefore, we define a new metric Motion Complexity as the sum of step-wise ratios between moved and unmoved body parts (Equation 4), where s denotes the step number and $|\cdot|$ represents the count of body parts.

$$Motion\ Complexity = \sum_{s=1}^{N} \frac{|moved_s|}{|unmoved_s|} \tag{4}$$

3.2 Human Evaluation

While automatic evaluation fixes the oracle high-level plans, we conduct human evaluation of the unconstrained generation, to account for multiple valid ways of performing a motion. Each animation is assessed by five independent annotators both overall and body-part-wise.

Overall Score. Given one animation and the corresponding motion instruction, the evaluator checks to what extent the animation is following the instructed motion, and gives one integer overall score from one to five.

Body Part Label. We ask human evaluators to check six body parts in the animations — `Head`, `Torso`, `Left Arm`, `Right Arm`, `Left Leg` and `Right Leg`. Evaluators classify each body part using one of four labels — "Good", "Partially Good", "Bad", and "Not Relevant". We introduce the "Not Relevant" label to distinguish between motion-critical body parts (e.g., arms during throwing) and those that have little involvement in the action (e.g., legs during a standing wave), while still marking any unnatural movement as "Bad". This separation helps evaluators provide targeted feedback on the quality of key motion components.

Instead of showing oracle animations alongside LLM-generated ones during evaluation, we separately include them in the evaluation pool to avoid biasing annotators toward a single reference motion while still establishing

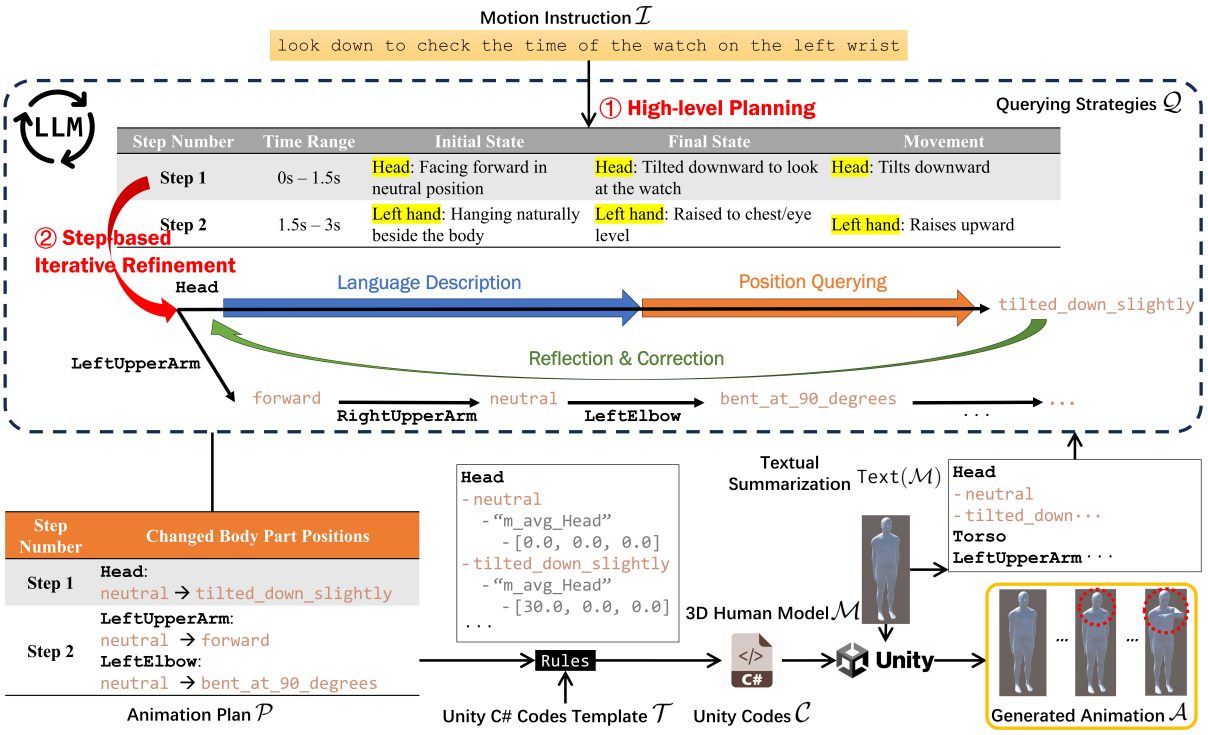


Figure 1: Pipeline of animation generation.

an upper performance bound. The inter-rater agreement shows moderate levels (weighted kappa of 0.531 for overall scores, average pairwise agreement of 0.510-0.638 for body parts), which is reasonable given the inherent variability in human motion.

4 Results and Analysis

We run our probing pipeline on selected LLMs, including Claude 3.5 Sonnet, GPT-4o, GPT-4o-mini, GPT-3.5-turbo and Llama-3.1-70B. As shown in Table 1, while oracle animations achieve an averaged Overall Score of 4.57, all tested LLMs demonstrate substantial shortcomings. The closest competitor, Claude 3.5 Sonnet, scores 1.28 points below the oracle. Body Part Position Accuracy follows a similar pattern — the highest performing LLMs Claude 3.5 Sonnet and GPT-4o only achieve 72.71% and 70.25% respectively. Given that humans are sensitive to minor inaccuracies in body movements [12], these substantial performance gaps suggest that LLMs generally struggle with accurately recognizing body part positions.

Further comparison of body part motions reveals generally large performance gaps between LLM-generated and oracle animations across all body parts, with varying degrees of deficit (Table 2, Figure 2). For body part labels (Table 2), head and torso movements show smaller deficits,

Table 1: Averaged Overall Score and Body Part Position Accuracy for each tested LLM.

LLM	Averaged Overall Score	Body Part Position Accuracy (%)
Claude 3.5 Sonnet	**3.29**	**72.71**
GPT-4o	3.13	70.25
GPT-4o-mini	2.87	67.82
GPT-3.5-turbo	2.14	66.90
Llama-3.1-70B	2.13	52.51
(Oracle Annotation)	4.57	100.00

while arm and leg motions display significant inaccuracies. Claude 3.5 Sonnet and GPT-4 lead in "Good" and "Partially Good" labels, while GPT-3.5-turbo and Llama-3.1-70B dominate "Bad" labels across all body parts. Body Part Position Accuracy (Figure 2) reveals that lower body parts achieve higher accuracy than their upper body counterparts (e.g., Knee versus Elbow), and complex body parts with more possible positions tend to have lower accuracy than simpler body parts (e.g., Upper Arm versus Elbow).

Complex Motions. We analyze the correlation between Motion Complexity and Body Part Position Accuracy (Figure 3), and find that LLMs tend to have lower Body Part Position Accuracy when predicting complex motions.

Complex Body Parts. Our analysis of the correlation between position prediction accuracy and number of

Table 2: Percentage (%) of body part labels (excluding "Not Relevant") across evaluated LLMs. **G**, **PG**, and **B** respectively stand for "Good", "Partially Good", and "Bad". Highest percentages for each label are highlighted in pink (**G**), yellow (**PG**), and gray (**B**).

LLM	Head			Torso			Left Arm			Right Arm			Left Leg			Right Leg		
	G	**PG**	**B**	**G**	**PG**	**B**	**G**	**PG**	**B**	**G**	**PG**	**B**	**G**	**PG**	**B**	**G**	**PG**	**B**
Claude 3.5 Sonnet	74.1	22.2	3.7	72.6	17.7	9.7	25.0	53.9	21.1	29.3	53.3	17.3	38.6	31.8	29.5	31.7	29.3	39.0
GPT-4o	63.8	19.1	17.0	60.7	25.0	14.3	15.2	58.2	26.6	16.9	64.9	18.2	46.8	36.2	17.0	29.5	47.7	22.7
GPT-4o-mini	80.7	8.8	10.5	59.4	28.1	12.5	12.8	47.4	39.7	12.2	52.7	35.1	17.9	33.3	48.7	11.1	33.3	55.6
GPT-3.5-turbo	34.2	13.2	52.6	29.1	16.4	54.5	3.8	41.8	54.4	3.8	46.2	50.0	10.3	30.8	59.0	5.4	18.9	75.7
Llama-3.1-70B	44.0	32.0	24.0	34.8	34.8	30.4	6.9	41.4	51.7	9.4	38.8	51.8	15.5	7.0	77.5	5.9	5.9	88.2
(Average)	59.4	19.0	21.6	51.3	24.4	24.3	12.8	48.6	38.7	14.3	51.2	34.5	25.8	27.8	46.3	16.7	27.0	56.2
(Oracle)	89.6	10.4	0.0	80.3	18.2	1.5	74.0	19.5	6.5	76.3	19.7	4.0	76.6	14.9	8.5	76.1	13.0	10.9

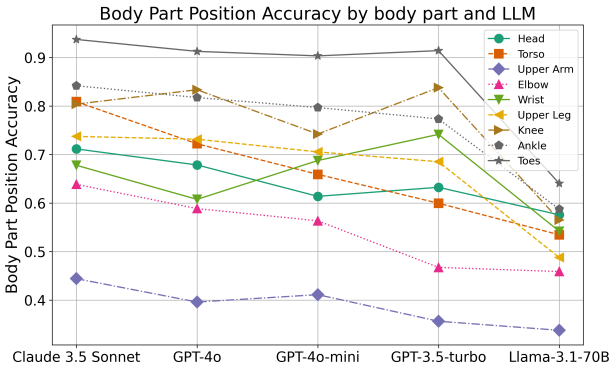


Figure 2: Body Part Position Accuracy for each body part and tested LLM. We average the accuracy for paired body parts, e.g., "Elbow" for "LeftElbow" and "RightElbow".

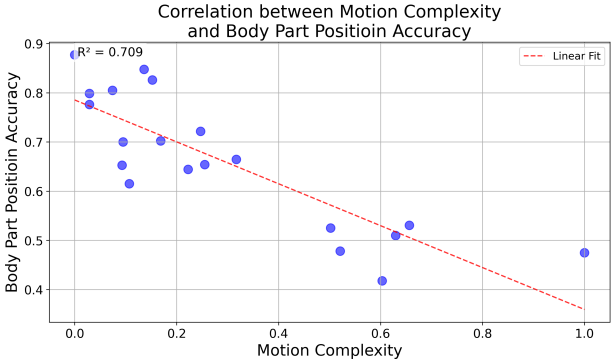


Figure 3: Motion-wise correlation between Motion Complexity and the averaged Body Part Position Acccuracy.

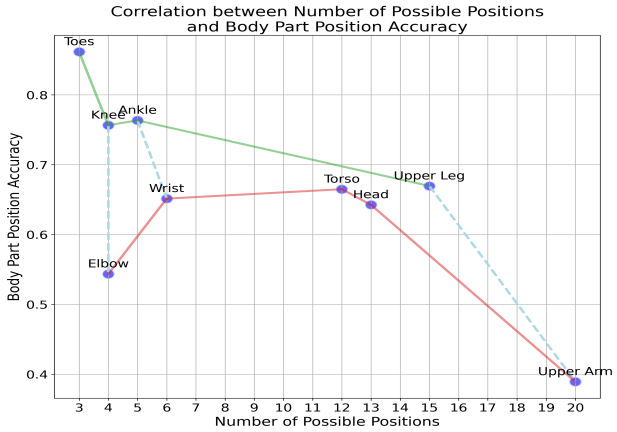


Figure 4: Body-part-wise correlation between number of possible positions and the averaged Body Part Position Acccuracy.

possible positions for different body parts (Figure 4) reveals two key patterns. First, prediction accuracy tends to inversely correlate with movement flexibility — body parts with more possible positions (e.g., Upper Arm) show lower accuracy compared to more constrained parts (e.g., Upper Leg). Second, comparison of the lower body performance (green line) and upper body performance (red line) demonstrates that LLMs achieve higher accuracy for lower body parts versus their upper body counterparts.

5 Conclusion

In this work, we explore the human motion knowledge embedded in LLMs, and verify it from the generated animations on the 3D human model SMPL. We find that LLMs understand human motions in natural language space to a certain degree, but struggle with accurate body part positions, especially complex motions and body parts.

Acknowledgement

This work was partially supported by the "R&D Hub Aimed at Ensuring Transparency and Reliability of Generative AI Models" project of the Ministry of Education, Culture, Sports, Science and Technology.

References

[1] Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In **In the 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23)**, UIST '23, New York, NY, USA, 2023. Association for Computing Machinery.

[2] Zhilin Wang, Yu Ying Chiu, and Yu Cheung Chiu. Humanoid agents: Platform for simulating human-like generative agents. In Yansong Feng and Els Lefever, editors, **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations**, pp. 167–176, Singapore, December 2023. Association for Computational Linguistics.

[3] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. **2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**, pp. 5142–5151, 2022.

[4] 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.

[5] Kunhang Li and Yansong Feng. Motion generation from fine-grained textual descriptions. In **Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)**, pp. 11625–11641, Torino, Italy, May 2024. ELRA and ICCL.

[6] Ke Fan, Jiangning Zhang, Ran Yi, Jingyu Gong, Yabiao Wang, Yating Wang, Xin Tan, Chengjie Wang, and Lizhuang Ma. Textual decomposition then sub-motion-space scattering for open-vocabulary motion generation, 2024.

[7] Nikos Athanasiou, Mathis Petrovich, Michael J. Black, and Gül Varol. SINC: Spatial composition of 3D human motions for simultaneous action generation. **ICCV**, 2023.

[8] Yiming Huang, Weilin Wan, Yue Yang, Chris Callison-Burch, Mark Yatskar, and Lingjie Liu. Como: Controllable motion generation through language guided pose code editing, 2024.

[9] Han Huang, Fernanda De La Torre, Cathy Mengying Fang, Andrzej Banburski-Fahey, Judith Amores, and Jaron Lanier. Real-time animation generation and control on rigged models via large language models, 2024.

[10] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. **ACM Trans. Graphics (Proc. SIGGRAPH Asia)**, Vol. 34, No. 6, pp. 248:1–248:16, October 2015.

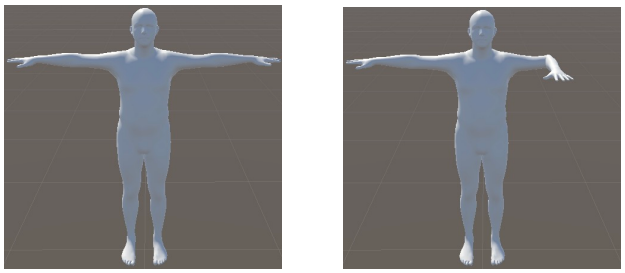
[11] Tamar Flash and Binyamin Hochner. Motor primitives in vertebrates and invertebrates. **Current Opinion in Neurobiology**, Vol. 15, No. 6, pp. 660–666, 2005. Motor sytems / Neurobiology of behaviour.

[12] Wentao Zhu, Xiaoxuan Ma, Dongwoo Ro, Hai Ci, Jinlu Zhang, Jiaxin Shi, Feng Gao, Qi Tian, and Yizhou Wang. Human motion generation: A survey. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, Vol. 46, pp. 2430–2449, 2023.

A Appendix

A.1 3D Human Model SMPL

SMPL [10] accurately generates the corresponding human body shape given specified pose parameters, i.e., joint local rotations. We can manipulate SMPL by modifying these parameters. For example, suppose that SMPL starts from an initial state extending two arms to the sides (Figure A1a), when we change the local rotation of the left elbow joint `m_avg_L_Elbow` from `(0, 0, 0)` to `(0, 90, 0)`, SMPL bends the left elbow at 90 degrees (Figure A1b).



(a) Initial (b) Left Elbow Bent

Figure A1: Overview of SMPL

A.2 Tested Motion Instructions

We collect 20 diverse motion instructions, covering different body parts in a balanced way. [5)] To show the potential of application to an open-world game, we devise each motion instruction to be a finite motion [6)] related to a specific practical scene, while avoiding commonly implemented animations in games like walking. Each instruction specifies necessary elements to avoid ambiguity, while we also prevent it from being verbose. The tested motion instructions are shown in Table A1.

A.3 Impact of Querying Strategies

We investigate into the effect of different LLM-querying strategies with GPT-4o. After changing high-level planning from generating piece-by-piece to in-one-go, the averaged overall score drops by 0.34. For step-based iterative refinement, we try selecting the body-part position from predefined positions all at once or one-by-one, instead of hierarchically. The performance drops respectively by 0.22 and 0.31.

Table A1: Tested Motion Instructions

Motion ID	Motion Instruction
1	Slide the window open from the center to the sides with both hands.
2	Water a 30-centimeter-tall plant using the watering can in the right hand.
3	Look down to check the time of the watch on the left wrist.
4	Pat a 30-centimeter-tall dog in front of you on the head with the right hand.
5	Lean back fully and toss the ball into the air at a 45-degree angle using both hands.
6	Wipe down the 1-meter-high table in front of you with a cloth in the left hand.
7	Hold the glass with the left hand and pour the juice with the right hand.
8	Put a book on the 2-meter-high shelf with both hands.
9	Lift a 20-centimeter-high box from the ground to the table on your left with both hands.
10	Swing the golf club from right to left.
11	Close the 2-meter-high store shutter door from top to bottom.
12	Squat to pick up litter by the right foot with the right hand.
13	Lift the right shoe with both hands and put it on in the air.
14	Perform a left-leg high side kick in Karate.
15	Kneel in a traditional Japanese bow.
16	Roll out a yoga mat on the ground.
17	Crouch to check a car tyre.
18	Arch the back 60 degrees to relieve tension in the lower back muscles with two hands on the waist.
19	Bend to the left to reach for an item by the left foot without moving or bending the left leg.
20	Walk through while ducking under a low-hanging branch.

5) We manually label involved body parts in all motion instructions. The involved body parts and their counts are: Head (15), Torso (16), Arms (16 each), Legs (13 each).

6) For example, "walking" without constraints like "three steps" can be infinite.