

AvatarGPT: All-in-One Framework for Motion Understanding, Planning, Generation and Beyond

Zixiang Zhou

zhouzixiang@xiaobing.ai

Yu Wan

wanyu@xiaobing.ai

Baoyuan Wang

wangbaoyuan@xiaobing.ai

Xiaobing AI

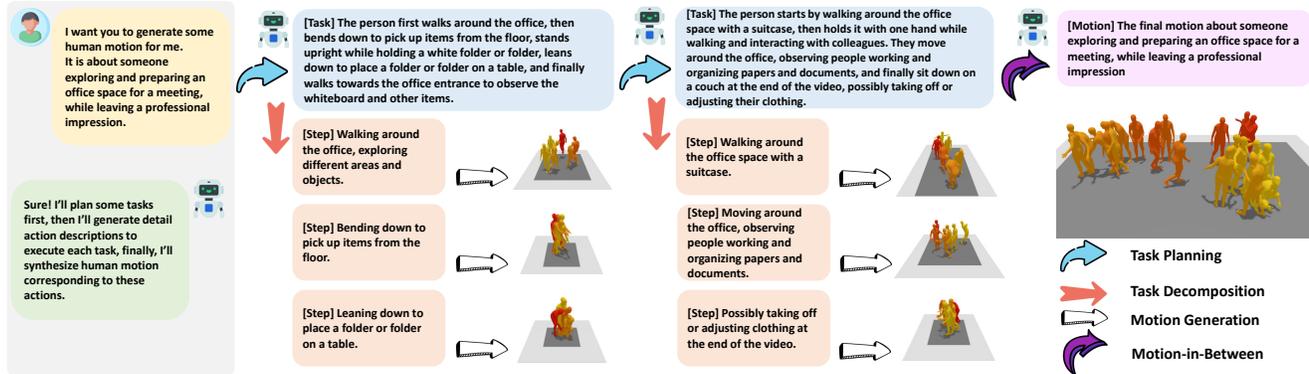


Figure 1. An example of long human motion generation based on high-level user instructions, powered by the traversal of a few key modules within our proposed framework, including motion task planning, decomposition, generation, and motion in-between synthesis.

Abstract

Large Language Models (LLMs) have shown remarkable emergent abilities in unifying almost all (if not every) NLP tasks. In the human motion-related realm, however, researchers still develop siloed models for each task. Inspired by InstuctGPT [16], and the generalist concept behind Gato [27], we introduce AvatarGPT, an All-in-One framework for motion understanding, planning, generations as well as other tasks such as motion in-between synthesis. AvatarGPT treats each task as one type of instruction fine-tuned on the shared LLM. All the tasks are seamlessly interconnected with language as the universal interface, constituting a closed-loop within the framework. To achieve this, human motion sequences are first encoded as discrete tokens, which serve as the extended vocabulary of LLM. Then, an unsupervised pipeline to generate natural language descriptions of human action sequences from in-the-wild videos is developed. Finally, all tasks are jointly trained. Extensive experiments show that AvatarGPT achieves SOTA on low-level tasks, and promising results on high-level tasks, demonstrating the effectiveness of our proposed All-in-One framework. Moreover, for the first time, AvatarGPT enables a principled approach by iterative traversal of the tasks within the closed-loop for unlimited long-motion synthesis.

1. Introduction

Text-based human motion generation has made significant progress in recent years [2, 7, 21, 22, 31, 39, 42]. These varied methods fundamentally aim to learn a direct mapping from natural language descriptions to human motions. Despite their impressive generative performance, they exhibit two limitations when viewed from an end-to-end perspective. Firstly, their efficacy predominantly lies in generating short motion sequences, and extending these methods to longer durations presents substantial challenges. Secondly, the dependency on manually crafted, detailed textual inputs for each motion sequence constrains their utility in real-world scenarios. Consider, for instance, the context of video game environments where non-player characters (NPCs) are required to perform multifaceted, high-level tasks, which need to be broken down into smaller, sequentially ordered sub-tasks, each demanding specific textual descriptions to guide the associated movements. Additionally, the need for post-processing to ensure seamless transitions between motions further complicates the application. Therefore, there is a pressing need to develop a comprehensive end-to-end framework. This framework should not only automate the process of motion planning but also proficiently handle task decomposition and motion generation, all while understanding the contextual subtleties and conforming to broad user-defined instructions.

Fortunately, we have witnessed significant advancement of Large Language Models (LLMs) as well as their remarkable emergent abilities in text-centric understanding [5, 16, 35], reasoning [9, 19, 29, 30, 36] and generations [1, 4, 17, 24–26, 32, 33]. Works like InstructGPT [16] and Gato [27] further push the paradigm-shifting to a new stage where almost all tasks can be treated as different instructions fine-tuned on top of the shared foundation LLM. However, in the realm of human body motion, these tasks continue to be addressed in isolation. There has been limited exploration in integrating these emergent LLM capabilities to innovate and validate new approaches in this domain.

Given the context, building on these paradigm shifts, we present **AvatarGPT**, a groundbreaking unified framework that harnesses the power of LLMs for seven distinct motion-related tasks, including motion understanding, planning, decomposition, generation, motion in-between synthesis, and scene estimation as well as task summarization. Note that, motion planning is a high-level task that requires a deep understanding of the context as well as certain common sense and reasoning abilities. The overview design is illustrated in Fig. 2. We represent continuous motion sequences as discrete tokens. To integrate motion modality into LLMs, we treat the discrete motion tokens as an extended vocabulary, so it could be integrated into any LLMs with little effort. AvatarGPT is structured around four high-level sub-tasks: task planning, decomposition, summarization, and scene estimation, as well as three low-level sub-tasks: text-driven motion generation, motion understanding, and in-between motion generation. We employ specialized prompts for each sub-task and use an instruction-tuning strategy for model training. Additionally, we have developed an innovative method to construct a dataset from real-world videos for high-level instruction tuning, which notably eliminates the need for manual human intervention in the process. To sum up, our contributions are as follows:

- We pioneer an All-in-One framework that integrates both high-level and low-level motion-related tasks, fostering a comprehensive optimization loop across understanding, planning, and generation phases.
- We develop a novel pipeline to construct a dataset from in-the-wild videos and also curate a dataset specifically for fine-tuning high-level human action planning.
- Through extensive evaluation, we demonstrate that our method sets new state-of-the-art benchmarks in low-level tasks and shows promising results in high-level tasks.
- Our framework significantly extends the capability for longer synthesis of human motions compared to prior works, thus paving the way for new applications.

2. Related Work

Motion Generation The synthesis of realistic human motion from natural language descriptions is a longstanding

research area. Initial efforts like [6] and [20] generated motion based on pre-defined action categories rather than natural language, limiting the diversity and control of generated motions. Recent advances [2, 7, 21, 22, 31, 38, 39, 42] have addressed this by using natural language descriptions as direct inputs for motion generation. These methods vary in their technical approaches: (1) VAE-based methods, such as [21] and [22], focus on learning an aligned space between motion and language, utilizing a decoder for motion synthesis. (2) Diffusion models, used in [2] and [31], learn to guide diffusion processes for generating human motion distributions from Gaussian noise. Additional auxiliary conditions can be applied for more detailed control [11, 37]. (3) Tokenization, a widespread technique in language models, has recently been applied to human motion generation [7, 10, 38, 41, 42], treating the process as akin to predicting the next motion token in language models. Despite showing promise, these methods still rely on specific user inputs, which can limit their broader applicability.

Motion Understanding Understanding the meaning of human motion is also a long-term research topic. Describing human motion with pre-defined action labels [43][3] has dominated this topic for a while. However, these methods have obvious drawbacks, they are not appropriate to describe complex motion sequences. Recently, the learning of the relationship between motion sequence and natural language description has attracted increasing attention. For instance, [23][7][10] learn the mutual mapping between human motion sequences and natural language descriptions. These enable describing complex motion sequences with accurate language descriptions.

Planning with LLMs Creating effective plans within specific environments or scenarios remains a complex task. However, with the rapid advancement of LLMs and the evolving concept of agents, planning using natural language descriptions is gaining traction. Works like [9, 14, 19, 29, 30] have shown the potential of using LLMs as task planners across various fields. These models, even without fine-tuning, can perform diverse tasks when provided with well-crafted prompts and instructions. Nonetheless, designing these prompts and instructions requires careful thought, and often, complex pipelines are needed to achieve the desired outcomes from LLMs. While these approaches are promising for task planning, they generally do not encompass task execution, necessitating additional modules for this purpose. A recent development [36] introduced an LLM-based framework that combines task planning with control, advancing the field of generalized human motion generation. Yet, this method can only take high-level tasks as input and produce corresponding movements, but not capable of reversing this process, limiting its utility and adaptability.

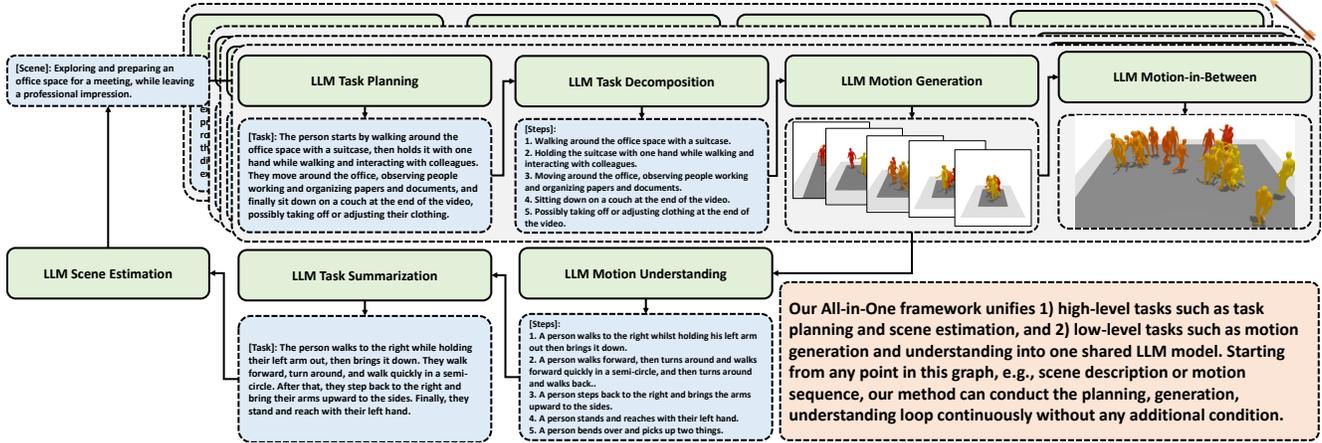


Figure 2. Overview of our All-in-One framework. Language serves as the interface to connect all these modulated tasks within the loop.

3. Method

The overview design of our All-in-One model is shown in Fig. 2, and the technical illustration of the main components is presented in Fig. 3. Our proposed method contains two modules, 1) a Multimodal LLM that learns various relationships between text descriptions and motion sequences, and 2) a novel automatic annotation pipeline that can annotate the content of any in-the-wild videos in natural language descriptions of various levels of detail. We describe their details in the following.

3.1. Motion Tokenization

We first learn a motion tokenizer to quantize continuous motion representations to discrete tokens. We follow [38] to train the VQ-VAE using the objective Eq. 1. Where $\|\tilde{x} - x\|$ is the reconstruction term, $\|sg[z] - z_q\|$ is the embedding term, and $\|z - sg[z_q]\|$ is the commitment term.

$$\mathcal{L}_{VQ} = \|\tilde{x} - x\| + \beta_1 \|sg[z] - z_q\| + \beta_2 \|z - sg[z_q]\| \quad (1)$$

3.2. Motion Vocabulary

To leverage an LLM, one needs to convert continuous motion sequences into discrete tokens. One straightforward solution is using VQ-VAEs. Given a motion sequence $x \in \mathbb{R}^{T \times c}$, its tokenized representation is a set of indices $[t_0, t_1, \dots, t_k]$. It is convenient to use part of the default vocabulary of LLMs as motion tokens in training LLMs[41]. Another solution is to extend the default vocabulary of LLMs and use this extended part for motion tokens[10]. However, these methods have shortcomings. For instance, [41] re-uses partial default vocabulary, which confuses the semantic meaning of associated embeddings since these embeddings are shared by multiple modalities. Although this problem is avoided in [10], new vocabulary

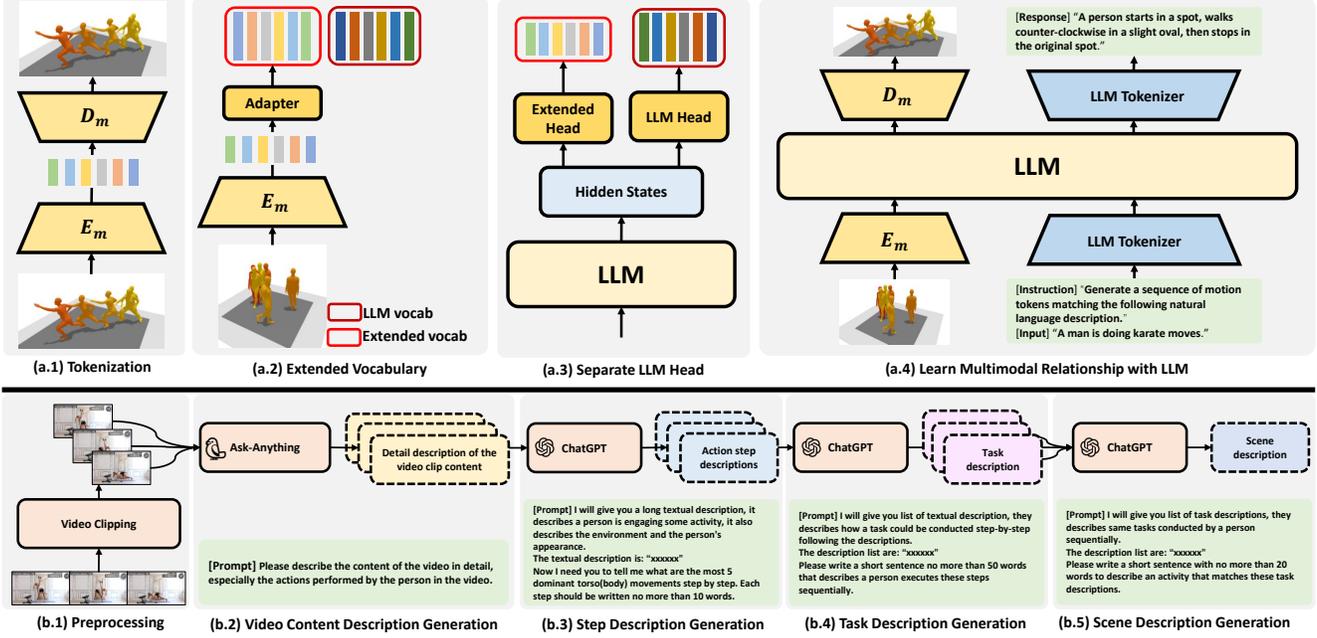
needs to be learned from scratch, which is inefficient especially when training on limited data. In our paper, we propose a lightweight vocabulary adaptor. It, on one hand, harnesses the semantic representation ability of discrete embeddings obtained by VQ-VAE, on the other hand, avoids learning extended vocabulary from scratch. As shown in Fig. 3 (a.2), we use vq-encoder and quantizer to transform input motion sequence $x \in \mathbb{R}^{T \times c}$ to discrete embedding sequences $z^q \in \mathbb{R}^{T^q \times d}$, where T^q is the length of embedding sequence, and d is the embedding dimension. Because the latent space of $z_i^q \in \mathbb{R}^d$ and the hidden space $h_i \in \mathbb{R}^D$ of LLMs do not align by default, we learn an adapter layer to transform z_i^q to align with h_i as:

$$f_{\theta_a}(z^q) : \mathbb{R}^d \rightarrow \mathbb{R}^D \quad (2)$$

We show in experiments that this technique brings performance gain while reducing the model size.

3.3. Separate Head for Motion Prediction

We propose to use a separate LLM head to map hidden states of LLM to motion tokens while maintaining the shape of its original head unchanged for text token prediction only. This is shown in Fig. 3 (a.3). This is different from previous methods [10, 41]. In [41], they remain the shape of the LLM head, but they re-use part of the LLM’s vocabulary for motion tokens. In [10], they modify the shape of LLM’s vocabulary because of the extended vocabulary. Using a shared LLM head for tasks of different modalities brings noticeable disadvantages. Because the vocabulary is not fully shared by different modalities, it is possible to sample tokens from the vocabulary that is out of the valid range of a particular modality. Consequently, it is not guaranteed that any sampled token can be decoded as expected. Our method solves this problem. We denote the original vocabulary of LLM as $\mathcal{V}_t = \{v_i\}_{i=1}^N$, and the extended vocabulary for



motion is $\mathcal{V}_m = \{v_0\}_{i=1}^M$, where M and N are vocabulary sizes and $M \neq N$. Given hidden states of LLM $h_i \in \mathbb{R}^D$, the original LLM head is $f_{\theta_t}(h) : \mathbb{R}^D \rightarrow \mathbb{R}^N$, and the separate head for motion is $f_{\theta_m}(h) : \mathbb{R}^D \rightarrow \mathbb{R}^M$. Therefore, learning modality-specific heads guarantees we can always sample tokens from the correct vocabulary. Hence, decoding performs expectedly.

3.4. Instruction Tuning the LLMs

Fig. 3 (a.4) shows that an LLM could be used to learn motion-related multimodal tasks tuned by different instructions. The fundamental concept of LLMs is to treat all inputs as discrete tokens and predict subsequent tokens based on previous ones as $p_{\theta}(x_i|x_{<i})$. In our case, there are two modalities, text and motion. We use LLM’s original vocabulary for text modality. For motion sequences, we first transform them to discrete embeddings as $z_q = \mathcal{Q}(\mathcal{E}(x))$, where $\mathcal{E}(\cdot)$ is the vq-encoder, and $\mathcal{Q}(\cdot)$ is the quantizer. Then we map the motion embeddings to extended vocabulary embeddings using Eq. 2. Through this process, we can align two different modalities into a cohesive and LLM-favorable representation, allowing us to harness the ability of LLMs to learn complex relationships between these modalities. Hence, we can formulate motion-related multimodal tasks as conditioned language generation problems.

Let’s denote the condition as a sequence of tokens as $C = \{c_i\}_{i=1}^{K_c}$, and the target as a sequence of tokens as $X = \{x_i\}_{i=1}^{K_x}$, the condition and target could be either text or human motion, and we can model the conditioned gen-

eration problem as $p_{\theta}(x_i|x_{<i}, C)$. We use a transformer encoder to extract the context information from condition C . At this stage, the full attention mechanism is adopted because we found it is more favorable than using causal attention in learning contextual information from the condition. Then we use a transformer decoder with causal attention to learn the relationship between target tokens and conditions. We found T5’s[26] encoder-decoder architecture is an appropriate choice. Because our model predicts the probability distribution of target tokens at every step, we use cross-entropy loss to supervise the fine-tuning. Since we use separate heads for text and motion modalities, our objectives for both modalities are as follows:

$$\mathcal{L}_t = -\sum_{i=1}^T \hat{x}_i \log(p_{\theta, \theta_t}(x_i|x_{<i}, C)) \quad (3)$$

$$\mathcal{L}_m = -\sum_{i=1}^T \hat{x}_i \log(p_{\theta, \theta_m}(x_i|x_{<i}, C)) \quad (4)$$

Where θ , θ_t , and θ_m are parameters of LLM, original LLM head, and extended LLM head, respectively. We list the prompts for instruction tuning in Appendix A.

3.5. Automatic Annotation Pipeline

We present a novel yet efficient method to annotate textual descriptions from in-the-wild videos of various levels of detail. Our method is shown in Fig. 3 (b). We use this pipeline to collect datasets to fine-tune our model on tasks such as task planning, decomposition, scene estimation, etc. Given any video, we first crop it into segments of fixed length, then we use a Visual-LLM to describe its content as detail

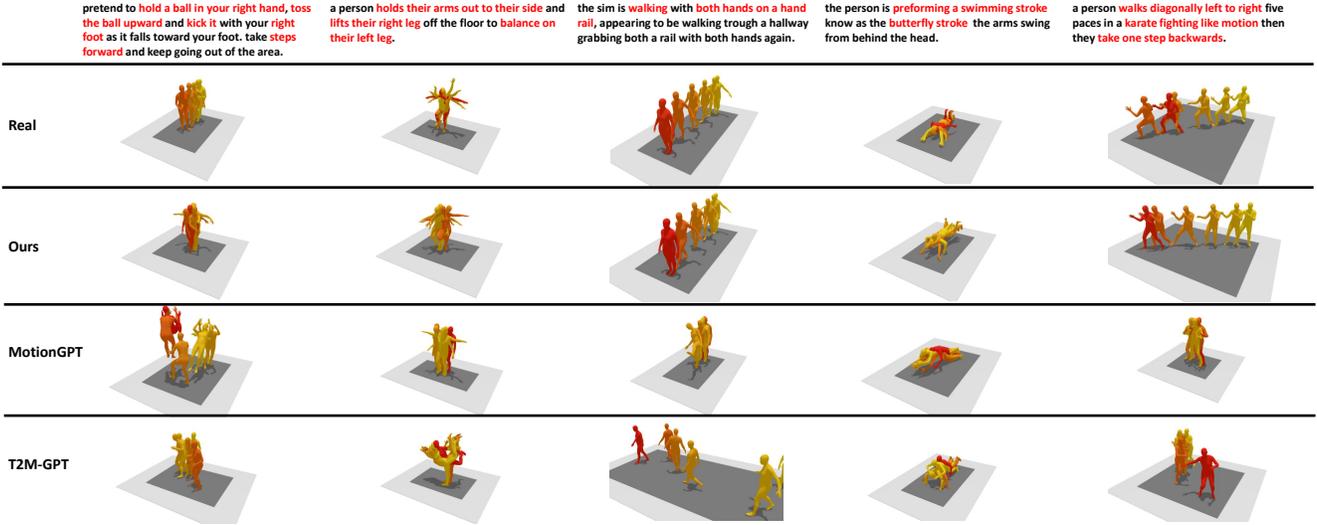


Figure 4. **Comparison of Motion Generation.** We compare the motion generation between ours and SOTA[10, 38]. We highlight the keywords to ease the visualization.

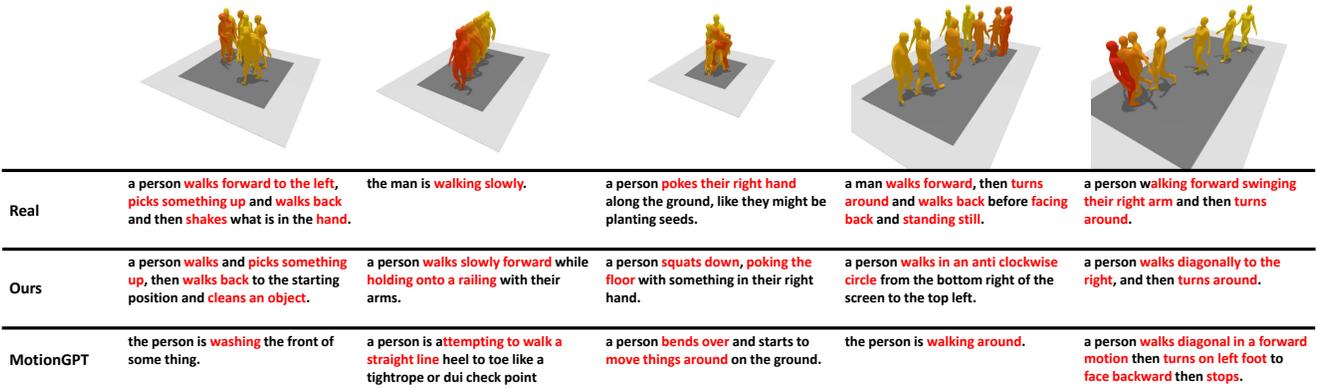


Figure 5. **Comparison of Motion Understanding.** We compare the motion understanding performance between ours and MotionGPT[10]. We highlight the keywords to show the alignment between motion and text.

Method	Motion Generation						Motion Understanding					Motion-in-Between		
	R Top-1 \uparrow	R Top-2 \uparrow	R Top-3 \uparrow	FID \downarrow	Div \rightarrow	MM \uparrow	Blue-1 \uparrow	Blue-4 \uparrow	Rouge \uparrow	Cider \uparrow	BertScore \uparrow	FID \downarrow	Div \rightarrow	MM \uparrow
GT	0.511	0.703	0.797	0.002 \pm 0.000	9.503 \pm 0.065	-	-	-	-	-	-	0.002	9.503	-
TM2T[7]	0.424 \pm 0.003	0.618 \pm 0.003	0.729 \pm 0.002	1.501 \pm 0.017	8.589 \pm 0.076	2.424 \pm 0.093	48.90	7.00	38.10	16.80	32.20	-	-	-
MDM[31]	0.320 \pm 0.005	0.498 \pm 0.004	0.611 \pm 0.007	0.544 \pm 0.044	9.559 \pm 0.086	2.799 \pm 0.072	-	-	-	-	-	2.698	8.42	-
MLD[2]	0.481 \pm 0.003	0.673 \pm 0.003	0.772 \pm 0.002	0.473 \pm 0.013	9.724 \pm 0.082	2.413 \pm 0.079	-	-	-	-	-	-	-	-
T2M-GPT[38]	0.491 \pm 0.005	0.680 \pm 0.003	0.775 \pm 0.002	0.116 \pm 0.004	9.761 \pm 0.081	1.856 \pm 0.011	-	-	-	-	-	-	-	-
MotionGPT[41]	0.411 \pm 0.000	0.594 \pm 0.000	0.696 \pm 0.000	0.542 \pm 0.000	9.311 \pm 0.000	-	-	-	-	-	-	-	-	-
MotionGPT[10]	0.492 \pm 0.003	0.681 \pm 0.003	0.778 \pm 0.002	0.232 \pm 0.008	9.528 \pm 0.071	3.096 \pm 0.008	48.20	12.47	37.40	29.20	32.40	0.214	9.56	-
HMD[28]	-	-	-	-	-	-	-	-	-	-	-	1.48	8.90	-
Ours	0.510 \pm 0.005	0.702 \pm 0.005	0.796 \pm 0.003	0.168 \pm 0.0083	9.624 \pm 0.0545	-	49.28	12.70	40.44	32.65	53.58	1.655 \pm 0.020	9.015 \pm 0.095	7.417 \pm 0.662

Table 1. **Results of Low-level Tasks.** We compare our method with various SOTAs on low-level tasks such as 1) Motion Generation, 2) Motion Understanding, and 3) Motion-in-Between. Indicate best results , indicates second best results .

Method	CT2T \uparrow	CS2S \uparrow	CT2S \uparrow	CS2T \uparrow	T2C \uparrow	S2C \uparrow	T2S \uparrow	S2T \uparrow
GT	0.843	0.882	0.818	0.849	0.872	0.910	0.678	0.812
Ours(GPT2-Large)	0.751	0.786	0.705	0.784	0.873	0.895	0.956	0.955
Ours(T5-Large)	0.843	0.937	0.884	0.841	0.940	0.893	0.994	0.997

Table 2. **Results of High-level Tasks.** We evaluate the Logical Coherent Score(LCS) on 8 high-level tasks, and we compare the results of our method by using T5 and GPT architecture.

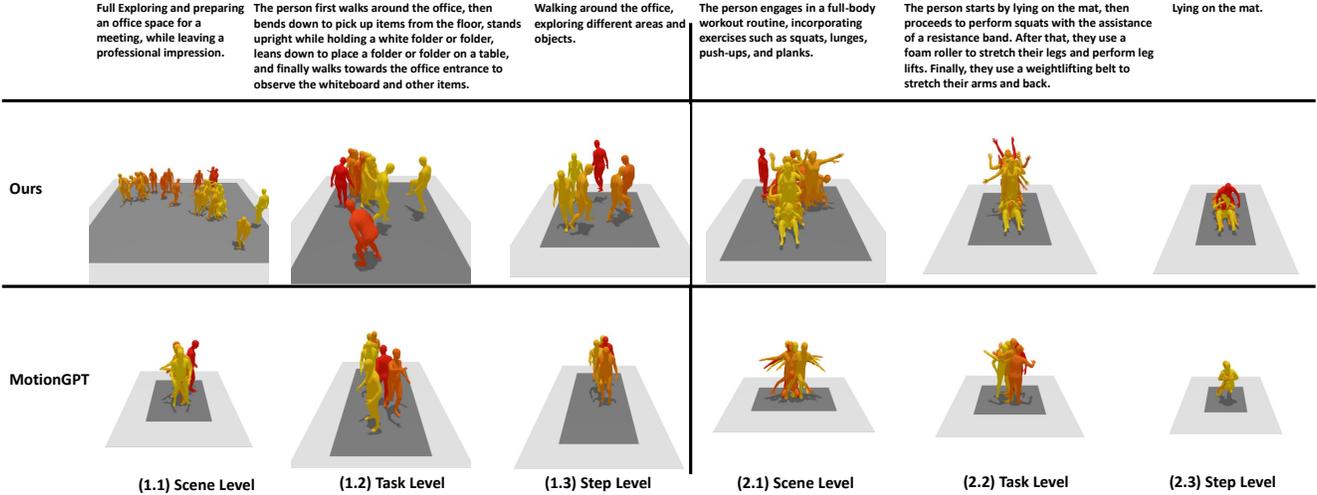


Figure 6. **Comparison of Motion Generation with Various Text Granularity.** We compare the generation results conditioned on coarse-grained(scene), middle-grained(task), and fine-grained(step) descriptions. Our method generates long motion with 2K+ frames from scene description(1.1, 2.1) and 0.6K frames from task description(1.2, 2.2), while MotionGPT[10] only results 0.2K+ frames for both scenarios.

as possible. In our pipeline, we adopt [12] for its encouraging ability in multimodal understanding and text generation. We then leverage ChatGPT[15] to describe how the person in the video moves step by step based on the very detailed descriptions, and we define descriptions of this level of detail: **step description**. This is the most fine-grained description, and each video segment could contain multiple steps. We also use ChatGPT to extract one summarized description for each video segment from the descriptions of the steps as well, which is denoted as: **task description**. In addition, we define the description that depicts the scene information of the entire video as: **scene description**, which is the most coarse-grained among all. To obtain this description, we feed the task descriptions related to each video segment to ChatGPT sequentially, and again ask ChatGPT to summarize the scene description that matches these tasks coherently. We describe the prompt structures used in this pipeline in Appendix B.

4. Experiments

4.1. Datasets

We use HumanML3D[7] to train and evaluate our method on three low-level tasks, including motion generation, motion understanding, and motion-in-between synthesis. The data preprocessing follows [7]. Because there are no appropriate datasets for motion task planning, scene understanding, task decomposition, etc, we collect a new dataset using the pipeline proposed in Sec. 3.5. The detailed procedure of dataset collection, the format, and the structure of the dataset are described in Appendix B, F.

4.2. Evaluation Metrics

Metrics for Low-level Tasks We use R-Precision score[7] to measure the alignment between textual description and generated motion sequence, Frechet Inception Distance(FID) and Diversity(Div) to measure the generation quality, Multimodality(MM) to evaluate the diversity of generation driven by the same control signal. FID, Div, and MM are used for motion generation and motion-in-between tasks as well. To assess the motion understanding performance, we measure the linguistic similarity between annotated texts and generated texts using the following metrics: 1) BLUE[18], 2) BLUE[18], 3) ROUGE-L[13], 4) CIDEr[34]. Refer to Appendix C for detailed descriptions.

Metrics for High-level Tasks We propose a novel metric, Logical Coherence Score(LCS), to evaluate the performance of high-level tasks, such as task planning, decomposition, and scene estimation, because there are no proper metrics for automatic assessment. Unlike motion understanding, where we can measure the semantic similarity between candidate and reference, there is no single correct answer for these high-level tasks. Whether the planned task is logically coherent with historical tasks or scenes is the primary concern. We therefore leverage the knowledge and reasoning ability of ChatGPT to judge whether our generated tasks match the conditions logically. For every generated text, we ask ChatGPT to determine whether it is logically coherent with conditions or not as $\mathbb{1}(\hat{x}_i, x_i) = 1$ if \hat{x}_i and x_i are logically coherent, otherwise $\mathbb{1}(\hat{x}_i, x_i) = 0$. Therefore, the coherent score for every task is calculated as $LCS = \frac{1}{N} \sum_{i=1}^N (\mathbb{1}(\hat{x}_i, x_i))$. We present the workflow of

	1	2	3	4
Ours Coarse-grained	The person starts by walking on an unseen train, then proceeds to do a deep squat, followed by catching something with both hands. After that, they stretch and finally walk in a figure eight pattern.	The person starts by performing a series of martial arts moves, then dances while keeping their feet on the ground, followed by kicking their legs and doing karate kicks, and finally ends by doing karate kicks again.	The person starts by bending their body while lying on the floor, then they perform bodyweight squats, followed by jumping in place twice. After that, they sit down, stand up, and put their arms out. Finally, they lift something above their head several times.	The person holds their right hand with their left hand, looks at their hands, raises their hands to their chest, stands and shows descriptive hand motions with basketball, shows dancing moves, steps forward with hands clasped, steps back, picks something up with left hand, and walks forwards.
Ours Fine-grained	<ol style="list-style-type: none"> The person starts by walking on an unseen train, then proceeds to do a deep squat, followed by catching something with both hands. After that, they stretch and finally walk in a figure eight pattern. The person walks to the right while holding their left arm out, then brings it down. They walk forward, turn around, and walk quickly in a semi-circle. After that, they step back to the right and bring their arms upward to the sides. Finally, they stand and reach with their left hand. 	<ol style="list-style-type: none"> The person starts by performing a series of martial arts moves, then dances while keeping their feet on the ground, followed by kicking their legs and doing karate kicks, and finally ends by doing karate kicks again. The person starts by performing martial arts kicks, then jumps rope, followed by crawling forward, standing up, and finally sitting down with arms and legs splayed out to the sides. 	<ol style="list-style-type: none"> The person starts by bending their body while lying on the floor, then they perform bodyweight squats, followed by jumping in place twice. After that, they sit down, stand up, and put their arms out. Finally, they lift something above their head several times. The person starts by doing a workout routine, then moves on to standing on their tiptoes, followed by dancing steps. After that, they perform the downward facing dog pose and finally, they turn around and do the same thing with their legs. 	<ol style="list-style-type: none"> The person holds their right hand with their left hand, looks at their hands, raises their hands to their chest, stands and shows descriptive hand motions with basketball, shows dancing moves, steps forward with hands clasped, steps back, picks something up with left hand, and walks forwards. The person walks in a semi circle to the left, sits down still, walks forward while holding their left arm with their right hand, turns counterclockwise, steps forward and makes hand gestures, takes a drink with their left hand and steps forward.
MotionGPT	a person walks forward, squats, stands up, walks forward, squats and stands up again.	a person aggressively kneeling and kicking.	a person squats down, stretches out left leg, pulls it back in and then stands up.	a person is stiff dancing with their arms and legs.

Figure 7. **Comparison of Motion Understanding of Various Level of Detail.** We compare the long motion understanding in various levels of detail with MotionGPT[10]. The motions have around 2K+ frames in length. Our method is able to describe the motion at both coarse- and fine-grained levels of detail.

Method	Motion Understanding					Task Sumarization				
	Blue-1 \uparrow	Blue-4 \uparrow	Rouge \uparrow	Cider \uparrow	BertScore \uparrow	Blue-4 \uparrow	Rouge \uparrow	Cider \uparrow	BertScore \uparrow	
Ours(GPT2-Large)	33.83	1.39	20.71	1.49	39.41	34.82	2.19	16.76	2.11	26.39
Ours(T5-Large)	30.14	1.22	17.85	1.40	41.05	40.11	7.45	24.82	3.03	32.28

Table 3. **Results of Full Pipeline Planning and Generation.** We assess our method’s performance on a full pipeline based on the concept of cycle consistency. We adopt linguistic similarity metrics to evaluate the task-level and step-level consistency.

Method	LCS		Ling. Consis.		T2M Consis \uparrow
	CT2T \uparrow	T2S \uparrow	Step \uparrow	Task \uparrow	
Ours(GPT2-Large)	88.50	92.32	55.56	65.00	2.8
Ours(T5-Large)	90.67	97.87	56.68	92.86	4.02

Table 4. **Results of User Study.** We assess our method’s performance in terms of task planning, decomposition, motion generation, understanding, and task summarization. ‘LCS’, ‘Ling. Consis.’, and ‘T2M Consis.’ respectively denote logical coherent score, linguistic consistency, and text-to-motion consistency.

evaluation, as well as the specific prompts for each task in Appendix D.

Metrics for Full Pipeline We define the full pipeline as: given a scene description, it conducts multiple rounds of task planning, decomposes each task to up to 5 steps descriptions, then synthesizes motions out of these steps, and finally blends these motions to one long sequence. Since there is no ground truth data, it is difficult to evaluate directly, we propose to evaluate the full pipeline based on the concept of cycle consistency. Concretely, we generate a long motion sequence from multiple tasks x_t^f and corresponding step descriptions x_s^f . Then we conduct a motion understanding task to describe the motion in various text

granularity, resulting task- and step-level descriptions as x_t^r , x_s^r respectively. Ideally, x_t^f and x_t^r should describe the same action, and so do x_s^f and x_s^r . We use BertScore, BLUE, ROUGE-L, and CIDEr to evaluate their linguistic similarity. We also conducted a user study to measure logical coherence, linguistic consistency, and motion-to-text consistency. Please refer to Appendix E for more details.

4.3. Results

Results of Low-level Tasks HumanML3D[7] is used to jointly train three low-level tasks. For testing, we generate 10 samples from every condition signal and report their mean and 95% confidence interval. We compare our method with various SOTAs and present the quantitative results in Tab. 1. The comparison shows that our method largely outperforms all previous approaches on motion understanding tasks. Our method also ranks first in terms of R-Precision metrics on the motion generation task and achieves competitive results on other metrics. We also compare our method on the motion-in-between task with different SOTAs and observe competitive performance.

Results of High-level Tasks We define 8 subtasks as high-level tasks. Denote ‘C’ as scene(context) description,

Method	Motion Generation					Motion Understanding					Motion-in-Between	
	R Top-1 ↑	R Top-1 ↑	R Top-3 ↑	FID ↓	Div →	Bleu-1 ↑	Bleu-4 ↑	Rouge ↑	Cider ↑	BertScore ↑	FID ↓	Div →
Llama-13B	0.389	0.539	0.623	0.567	9.489	49.01	12.42	39.01	27.73	51.31	-	-
GTP2-Large	0.454	0.635	0.728	0.316	10.021	51.92	13.29	40.93	33.69	51.73	0.911	9.007
T5-Base	0.468	0.654	0.751	0.284	9.967	48.29	11.51	38.51	28.11	51.59	4.042	8.982
T5-Large	0.510	0.702	0.796	0.168	9.624	49.28	12.70	40.44	32.65	53.58	1.655	9.015
w/o adapter	0.497	0.688	0.784	0.215	9.842	50.42	11.75	37.29	27.23	49.34	2.356	8.880
w/ adapter	0.510	0.702	0.796	0.168	9.624	49.28	12.70	40.44	32.65	53.58	1.655	9.015

Table 5. **Ablation Study.** We investigate the effectiveness of model architecture, sizes, and ways of introducing extended vocabulary.

‘T’ as task description, and ‘S’ as step description, task ‘CT2T’ stands for given scene(T) and one historical task(T) description, conduct one round of planning to predict the next possible task(T). Following this notation, we define the following 8 subtasks as CT2T, CS2S, CT2S, CS2T, T2C, S2C, T2S, and S2T. Because there are no similar methods designed for these tasks, we present a novel benchmark(Tab. 2) in this paper for two purposes: 1) we measure the Logical Coherent Score(LCS) on our collected dataset and the generation results of our method to justify the effectiveness of our method, and 2) we provide this benchmark and dataset for future research. We use T5-Large and GPT2-Large as our LLM and report the quantitative metrics on these 8 subtasks respectively, and the results suggest that using T5-Large outperforms GPT2-Large on 7 of 8 tasks.

Results of Full Pipeline Our model supports 1) generating very long motion sequences from a single scenario description and 2) describing the long motion at various text granularities. The depiction of motion should match the planned task/step description semantically. Therefore, we evaluate the consistency using [13, 18, 34, 40] on task summarization(task-level) and motion understanding(step-level) cycle consistency. We compare two variants of LLM architecture and report the results in Tab. 3. The results suggests that using T5-Large as LLM outperforms GPT2-Large significantly on task summarization, and achieves competitive results on motion understanding tasks.

User Study We conduct user study to investigate the performance of our method. Given a scene description, we run task planning, decomposition, motion synthesis, and motion-in-between, resulting in descriptions in various detail granularities and a long motion sequence. We then perform motion understanding and task summarization to describe the motion at fine- and coarse-grained. We evaluate the 1) LCS on planning and decomposition, and 2) linguistic consistency on motion understanding and summarization tasks. Participants are expected to score ‘1’ to correct(coherent and consistent) answers, while ‘0’ to the incorrect. In addition, human evaluation on the consistency between generated long motion and descriptions is also performed, with a rating scale of 1-5 as a metric. The results are reported in Tab. 4. The results suggest using T5-Large

as LLM has stronger ability in planning, decomposition, synthesis, understanding, summarization, and in-between tasks than GPT2-Large. The conclusions of manual and automated evaluation demonstrate high degree of consistency, supporting the effectiveness of our method.

4.4. Ablation Study

We conduct various ablation studies to investigate the effectiveness of 1) LLM model architecture, 2) LLM model size, and 3) extended vocabulary with an adapter.

LLM Model Architecture We investigate the effectiveness of various LLM model architectures. We adopt Llama-13B[32], GPT2-Large[25], and T5-Large[26] and conduct multitask fine-tuning. LoRA[8] is adopted for Llama-13B to save memory usage. The results on low-level tasks are shown in Tab. 5, and those of high-level tasks are reported in Tab. 2 as well. We found that Llama-13B has poor performance on generation and understanding, this is because LoRA actually finetunes < 1% parameters to connect motion modality with text, which fails to fully use the capacity of Llama. Although GTP2-Large has slightly better performance on motion understanding, T5-Large outperforms on all other tasks, suggesting the superiority of T5-Large against GPT2-Large as LLM architecture for our scenario.

LLM Model Sizes We also compare the performance discrepancy in terms of model sizes. Tab. 5 shows that a larger model size(T5-Large) brings noticeable performance gain.

Extended Vocab. w/ or w/o Adapter We investigate the effectiveness of different approaches to introducing extended vocabulary. 1) Extend the LLM vocabulary and learn the weights from scratch. 2) Align the quantized embeddings to the LLM vocabulary embedding space through the adapter layer. Tab. 5 demonstrates that our proposed approach achieves remarkable improvements on various tasks.

5. Conclusion

We show the synergy of unifying seven motion-related tasks through our newly introduced All-in-One framework. For the first time, we demonstrate that our method can enable long-motion synthesis thanks to its iterative planning, understanding, generation, etc, within the integral framework.

References

- [1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. **2**
- [2] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18000–18010, 2023. **1, 2, 5**
- [3] Hyung-gun Chi, Myoung Hoon Ha, Seunggeun Chi, Sang Wan Lee, Qixing Huang, and Karthik Ramani. Infogcn: Representation learning for human skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20186–20196, 2022. **2**
- [4] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022. **2**
- [5] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022. **2**
- [6] Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2motion: Conditioned generation of 3d human motions. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2021–2029, 2020. **2**
- [7] Chuan Guo, Xinxin Xuo, Sen Wang, and Li Cheng. Tm2t: Stochastic and tokenized modeling for the reciprocal generation of 3d human motions and texts. *arXiv preprint arXiv:2207.01696*, 2022. **1, 2, 5, 6, 7**
- [8] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. **8**
- [9] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*, pages 9118–9147. PMLR, 2022. **2**
- [10] Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiongpt: Human motion as a foreign language. *arXiv preprint arXiv:2306.14795*, 2023. **2, 3, 5, 6, 7**
- [11] Korrawe Karunratanakul, Konpat Preechakul, Supasorn Suwajanakorn, and Siyu Tang. Guided motion diffusion for controllable human motion synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2151–2162, 2023. **2**
- [12] Kunchang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023. **6**
- [13] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004. **6, 8**
- [14] Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. Llm+ p: Empowering large language models with optimal planning proficiency. *arXiv preprint arXiv:2304.11477*, 2023. **2**
- [15] OpenAI. Chatgpt. accessed: 2023-05-15. <https://openai.com/blog/chatgpt>, 2023. **6**
- [16] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. **1, 2**
- [17] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback, 2022. URL <https://arxiv.org/abs/2203.02155>, 13, 2022. **2**
- [18] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002. **6, 8**
- [19] 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. *arXiv preprint arXiv:2304.03442*, 2023. **2**
- [20] Mathis Petrovich, Michael J Black, and Gül Varol. Action-conditioned 3d human motion synthesis with transformer vae. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10985–10995, 2021. **2**
- [21] Mathis Petrovich, Michael J Black, and Gül Varol. Temos: Generating diverse human motions from textual descriptions. *arXiv preprint arXiv:2204.14109*, 2022. **1, 2**
- [22] Mathis Petrovich, Michael J Black, and Gül Varol. Tmr: Text-to-motion retrieval using contrastive 3d human motion synthesis. *arXiv preprint arXiv:2305.00976*, 2023. **1, 2**
- [23] Matthias Plappert, Christian Mandery, and Tamim Asfour. Learning a bidirectional mapping between human whole-body motion and natural language using deep recurrent neural networks. *Robotics and Autonomous Systems*, 109:13–26, 2018. **2**
- [24] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. **2**
- [25] 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. **8**

- [26] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020. 2, 4, 8
- [27] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. A generalist agent, 2022. 1, 2
- [28] Yonatan Shafir, Guy Tevet, Roy Kapon, and Amit H Bermano. Human motion diffusion as a generative prior. *arXiv preprint arXiv:2303.01418*, 2023. 5
- [29] Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2998–3009, 2023. 2
- [30] Haotian Sun, Yuchen Zhuang, Ling kai Kong, Bo Dai, and Chao Zhang. Adaplaner: Adaptive planning from feedback with language models, 2023. 2
- [31] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022. 1, 2, 5
- [32] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 2, 8
- [33] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shru ti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 2
- [34] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015. 6, 8
- [35] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners, 2022. 2
- [36] Zeqi Xiao, Tai Wang, Jingbo Wang, Jinkun Cao, Wenwei Zhang, Bo Dai, Dahua Lin, and Jiangmiao Pang. Unified human-scene interaction via prompted chain-of-contacts. *arXiv preprint arXiv:2309.07918*, 2023. 2
- [37] Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. Physdiff: Physics-guided human motion diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16010–16021, 2023. 2
- [38] 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. *arXiv preprint arXiv:2301.06052*, 2023. 2, 3, 5
- [39] Mingyuan Zhang, Xinying Guo, Liang Pan, Zhongang Cai, Fangzhou Hong, Huirong Li, Lei Yang, and Ziwei Liu. Remodiffuse: Retrieval-augmented motion diffusion model. *arXiv preprint arXiv:2304.01116*, 2023. 1, 2
- [40] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019. 8
- [41] 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. *arXiv preprint arXiv:2306.10900*, 2023. 2, 3, 5
- [42] Zixiang Zhou and Baoyuan Wang. Ude: A unified driving engine for human motion generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5632–5641, 2023. 1, 2
- [43] Wentao Zhu, Xiaoxuan Ma, Zhaoyang Liu, Libin Liu, Wayne Wu, and Yizhou Wang. Motionbert: A unified perspective on learning human motion representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15085–15099, 2023. 2