MotionGPT: Human Motion Synthesis with Improved Diversity and Realism via GPT-3 Prompting

Jose Ribeiro-Gomes∗1 Tianhui Cai∗2 Zoltán Á. Milacski2 Chen Wu2 Aayush Prakash3 Shingo Takagi3 Amaury Aubel3 Daeil Kim3 Alexandre Bernardino1 Fernando De La Torre2

1Instituto Superior Tecnico, Lisbon, Portugal 2Carnegie Mellon University, USA 3Meta, USA

{josepgomes@,alex@isr.}tecnico.ulisboa.pt {tcai2,zmilacsk,chenwu2}@andrew.cmu.edu

{aayushp,shingo.takagi,amauryaubel}@meta.com daeil.kim@gm.slc.edu ftorre@cs.cmu.edu

Abstract

There are numerous applications for human motion synthesis, including animation, gaming, robotics, or sports science. In recent years, human motion generation from natural language has emerged as a promising alternative to costly and labor-intensive data collection methods relying on motion capture or wearable sensors (e.g., suits). Despite this, generating human motion from textual descriptions remains a challenging and intricate task, primarily due to the scarcity of large-scale supervised datasets capable of capturing the full diversity of human activity.

This study proposes a new approach, called MotionGPT, to address the limitations of previous text-based human motion generation methods by utilizing the extensive semantic information available in large language models (LLMs). We first pretrain a doubly text-conditional motion diffusion model on both coarse (“high-level”) and detailed (“low-level”) ground truth text data. Then during inference, we improve motion diversity and alignment with the training set, by zero-shot prompting GPT-3 for additional “low-level” details. Our method achieves new state-of-the-art quantitative results in terms of Fréchet Inception Distance (FID) and motion diversity metrics, and improves all considered metrics. Furthermore, it has strong qualitative performance, producing natural results. Code is available at https://github.com/humansensinglab/MotionGPT

1. Introduction

Human motion synthesis is an actively researched field that has vast potential applications in areas such as animation, gaming, robotics, medical research, and activity recognition. For instance, it can create more natural movements for animated characters [18], help develop prosthetic devices that mimic human movements [14], and augment motion data to improve performance in downstream applications [50]. Traditional methods for synthesizing human motion rely on data collection with expensive and well-calibrated motion capture systems or wearable sensors [29, 31], as well as human labor to perform, clean and annotate the motion data. This makes it challenging to acquire large amounts of data for each activity.

Recent developments in generative modeling have led to various techniques that allow for the direct synthesis of motion sequences from natural language descriptions, with minimal human involvement and without the need for costly capture systems. These methods aim to generate plausible SMPL-compatible [26] parameters (i.e., global translation, global orientation and joint rotation angles representing the body pose) that accurately reflect the given text. Popular approaches include CLIP-aligned autoencoders [29], and both Variational Autoencoders (VAEs) [4, 22, 35, 50] and Denoising Diffusion Probabilistic Models (DDPMs) [17, 46, 47, 51, 59] conditioned on text embeddings.

However, these methods have a major drawback of suffering from the moderate sizes and the ambiguous nature of the texts in their training datasets, which typically contain only coarse “high-level” descriptions. Consequently, they cannot generate detailed motions that correspond to fine-grained “low-level” texts, limiting their generalization, diversity, and controllability. For instance, as shown in Fig. 1, the “high-level” activity of “greet a friend” can involve a range of “low-level” variability, including a “handshake”, a “wave”, a “hug”, a “kiss”, or a “bow”, depending on culture, the situation, and personal relationships. It remains unclear how existing methods could capture such variability without larger training datasets with more detailed text annotations.

This paper introduces MotionGPT, a doubly text-conditional motion diffusion model that addresses the afore-
mentioned limitation with zero-shot "common-sense" inference. We pretrain the motion diffusion model using both ground truth "high-level" HumanML3D [51] and "low-level" BABEL [36] annotations. During inference, the “low-level” texts are obtained by zero-shot prompting the GPT-3 [9] large language model (LLM) using the “high-” and “low-level” pairs of the training dataset along with a “high-level” query from the test dataset. This approach increases motion diversity by adding randomness and improves generalization by aligning test samples with the training dataset. Our proposed method achieves new state-of-the-art quantitative results in text-to-motion generation in terms of Fréchet Inception Distance (FID) and motion diversity, as well as strong qualitative performance.

Our contributions are summarized as follows:

- A motion diffusion model conditioned on both “high-” and “low-level” descriptions for fine-grained control over generated motions.
- Zero-shot prompting GPT-3 for “low-level” details of “high-level” texts, resulting in more motion diversity and better alignment with the training dataset.

2. Related Work

2.1. Human Motion Generation

Generative models for human motion synthesis can be categorized into two types: supervised and unsupervised methods.

Supervised methods aim to approximate input-to-target mappings from datasets of respective pairs. E.g., one may estimate 3D human pose targets from monocular image [7, 33], video [23] or 2D pose [12, 28] inputs. However, in practice, target data may not always be readily available or sufficient for proper generalization.

Therefore, an alternative approach is to apply unsupervised learning by solely utilizing the input data. E.g., predicting future motion frames can be achieved by masking out past frames from the input [3, 5, 10, 58]. Similarly, one may attempt to reconstruct the whole input sequence from a compressed representation via autoencoders or decoder-only models. Such schemes are either unconditional (purely unsupervised) or conditional (the latent code contains some supervisory signal). Unconditional methods have limited control over the generated motions, and are typically employed as motion priors in pose estimation [23, 33]. In contrast, conditional procedures offer more control. In particular, the emergence of labeled action recognition datasets such as NTU-RGB-D [44] has sparked a group of class-conditional solutions [11, 16, 34]. These models can generate a variety of motions from the same action class, but precise control over the outputs is still limited by the number of classes. As a solution, the emergence of motion datasets with natural language annotations, such as BABEL [36] and HumanML3D [15], has led to the development of text-conditional generative models that aim to take advantage of the compositionality of language. Notable examples include Conditional Variational Autoencoders (cVAEs) [4, 35] and Conditional Generative Adversarial Networks (cGANs) [1]. More recently, Conditional Denoising Diffusion Probabilistic Models (cDDPMs) [21, 51, 59] have been proposed to address the many-to-many nature of text-to-motion generation. By replacing compression with iterative denoising, these models produce more diverse and detailed motions.

In this paper, we extend the Human Motion Diffusion Model (MDM) [51], a diffusion-based motion generative model that is conditioned on “high-level” text input. While
MDM can generate reliable motions when the text input is specific and similar to the training set, it struggles with ambiguous or difficult-to-generalize inputs, as shown in Fig. 4. To overcome this limitation, we propose a method that leverages the common-sense knowledge embedded in the GPT-3 language model to break down “high-level” test descriptions into more detailed and randomly-generated “low-level” variants that are similar to those of the training dataset. By incorporating this additional “low-level” text conditioning signal, we achieve better generalization due to the increased similarity to the training dataset, as well as more motion diversity thanks to the randomness.

2.2. Large Language Models and Zero-Shot Prompting

Large Language Models (LLMs) [9, 13, 37] are transformer models in Natural Language Processing (NLP) that are pretrained on vast amounts of text data. Pretraining is accomplished by solving unsupervised pretext tasks, such as masked word prediction, to learn generic feature representations. These features can then be used to solve downstream problems, such as question answering [20, 53], language translation [49, 62], sentiment analysis [24, 57], or guiding methods in other modalities by leveraging common-sense knowledge [8, 55].

One way to use the pretrained model is by fine-tuning it for the downstream task [19, 40, 41, 48]. However, this may require a considerable amount of training data.

Recently, it has been demonstrated that LLMs can also generalize to novel problems using zero-shot prompting [39, 42, 43]. The key idea behind this approach of in-context learning is to learn from analogy, wherein language models are able to learn tasks given only a few demonstration examples [9]. Though it has been used in other areas [30, 45], its use in human motion generation is novel. During inference, by providing a prompt comprising a small number of in-context input-target examples and an additional query input, the LLMs can identify the patterns in the examples and combine them with the pretrained semantic information to generate the query output. Prompt design is crucial for good performance [56, 60] and it has been shown that using more semantically similar examples to the query case yields better results [25].

In this work, we adopt zero-shot prompting for the GPT-3 LLM to augment “high-level” test motion descriptions into their random “low-level” counterparts with additional details on the action and increased similarity to the training dataset. We then condition a motion diffusion model on both texts.

3. Methods

In this section, we start by formulating the task (Section 3.1), followed by introducing our model, MotionGPT (Section 3.2), and our GPT-3 prompting (Section 3.3).

3.1. Problem Formulation

Our objective is to generate a sequence of 3D human motion parameters \( x^{1:N} = [x^1, \ldots, x^N] \in \mathbb{R}^{N \times F} \) that corresponds to a given text conditioning \( c \in \mathbb{R}^C \). Here, \( N \) denotes the variable temporal length of the motion sequence (i.e., the number of frames) and \( F \) is the number of parameters per frame.

Text conditioning. The conditioning signal \( c \) is an embedding of a real-world text, such as a natural language sentence that describes an action or how it should be performed. Specifically, we utilize CLIP-embeddings of the descriptions contained within the HumanML3D [15] dataset. Throughout this study, we refer to these as the “high-level” texts.

Motion parameters. In line with related works in the field, we adopt a body motion representation that is compatible with SMPL [26], which includes 6D parent-relative joint rotation angles \( \hat{J} \) for \( J \) body joints. Additionally, we incorporate a 3D global translation and a 4D global orientation in quaternion representation to facilitate spatial arrangement. To improve the training process, we also include extra redundant parameters, such as joint locations and angular velocities, as proposed by [15].

3.2. MotionGPT

Our proposed MotionGPT architecture builds upon Human Motion Diffusion Model (MDM) [51] and introduces a second conditioning with a more detailed text, which we refer to as the “low-level” description, to enable fine-grained control over the embedding space and the generated motion. In addition to the “high-level” description of the motions, we train our model with ground-truth “low-level” texts, and use GPT-3 prompting (Section 3.3) during inference to achieve greater diversity and more similarity to the training data distribution. In this subsection, we provide an overview of our framework, which is illustrated in Fig. 2.

Diffusion model. We utilize a Denoising Diffusion Probabilistic Model (DDPM) [38, 51] for generating motions.

The forward Markov noising process initiates with the training sample \( x^{1:N}_t = x^{1:N} \), and gradually adds noise to it, \( i.e., \) for time steps \( t = 1, \ldots, T \):

\[
q(x^{1:N}_t | x^{1:N}_{t-1}) = \mathcal{N} \left( \sqrt{\alpha_t} \cdot x^{1:N}_{t-1}, (1 - \alpha_t) \cdot I \right). \tag{1}
\]

The \( \alpha_t \in (0, 1) \) values are constant hyperparameters. For small \( \alpha_t \), the forward process can be approximated as \( x^{1:N}_t \sim \mathcal{N}(0, I) \).
Subsequently, the reverse denoising process trains a denoising neural network \( G \) to reconstruct the original motion input \( x_0^{1:N} \) by approximately inverting multiple steps of the forward process:

\[
L_{\text{rec}} = \mathbb{E}_{x_0^{1:N} \sim q(x_0^{1:N}|c), t \sim [1, T]} \left[ \| x_0^{1:N} - G(x_0^{1:N}, t, c) \|^2_2 \right].
\]

During inference, one may start with \( x_0^{1:N} \sim N(0, I) \), then alternate between the denoising network \( G \) and the backward noising process with gradually decreasing number of steps \( (T - 1, \ldots, 1) \).

**Architecture.** Our denoising network architecture \( G \) is based on MDM [51] and employs a transformer [52] with four inputs: the training motion sample \( x_0^{1:N} \) of variable length, the encoding of the denoising step index \( t \), the positional embedding of the temporal ordering of motion frames, and the conditioning vector \( c \). These inputs are linearly projected into a common 512-dimension space and added together to form the input of the transformer encoder. We kindly refer the reader to MDM [51] for additional details.

Unlike MDM, which only uses the CLIP-embedding of the “high-level” text annotation of HumanML3D [15] as the conditioning vector \( c \), we also incorporate the LLM-embedding of its “low-level” variant, and sum their linear projections. During training, we generate the “low-level” text by sorting and concatenating ground truth per-frame labels from the BEL [36] dataset. During inference, we employ GPT-3 prompting (see Section 3.3). We experiment with various LLMs for the “low-level” embedding, such as DistilBERT [41], Sentence-T5 [32], and MiniLM [34]. Note that it is also possible to obtain the “low-level” text during inference through user input or test data from BEL.

**Training loss.** Geometric losses [34, 35, 50, 51] are often utilized in motion generation to encourage natural and physically plausible motions. In this work, we incorporate regularization for joint positions (obtained via forward kinematics from the model-generated joint angles), joint angular velocity and foot contact (to prevent foot-skating). Specifically, we define the losses as follows:

\[
L_{\text{pos}} = \frac{1}{N} \sum_{i=1}^{N} \| FK(x_0^i) - FK(\hat{x}_0^i) \|^2_2,
\]

\[
L_{\text{vel}} = \frac{1}{N - 1} \sum_{i=1}^{N} \| (x_0^{i+1} - x_0^i) - (\hat{x}_0^{i+1} - \hat{x}_0^i) \|^2_2,
\]

\[
L_{\text{foot}} = \frac{1}{N - 1} \sum_{i=1}^{N-1} \| (FK(\hat{x}_0^{i+1}) - FK(\hat{x}_0^i)) \cdot f_i \|^2_2.
\]

Here, the function \( FK(\cdot) \) represents forward kinematics (converting from joint angles to joint positions), and \( f_i \in \{0, 1\}^J \) is the ground truth binary foot contact mask for each frame \( i \).

The overall training loss is then defined as:

\[
L = L_{\text{rec}} + \lambda_{\text{pos}}L_{\text{pos}} + \lambda_{\text{vel}}L_{\text{vel}} + \lambda_{\text{foot}}L_{\text{foot}}.
\]
3.3. GPT-3 Prompting

During inference, we utilize zero-shot prompting (Section 2.2) for the GPT-3 LLM to generate “low-level” descriptions for each “high-level” test query. This approach has two benefits. Firstly, it increases the diversity of text prompts, resulting in a wider range of generated motions. Secondly, it allows for better generalization by achieving greater similarity to the training data distribution. This is done by breaking down hard “high-level” test set descriptions into similar “low-level” actions found in the training set.

Our prompting procedure is depicted in Fig. 3. We follow KATE [25] and retrieve the top-$k$ most similar training examples for each “high-level” query $h$. We accomplish this by precomputing a dictionary $\tilde{D}$ of “high-level” Sentence-T5 [32] embeddings, $i.e.$, $\tilde{D} = \{z_1, z_2, ..., z_S\}$, where $S$ is the size of the training set. Then during inference, we obtain the embedding of $\hat{h}$, represented as $\hat{z}$, calculate its cosine similarity with the dictionary $\tilde{D}$, and retrieve the top-$k$ most similar “high-level” training descriptions $H^p = \hat{z}_1 \hat{h}_k$. Finally, we form pairs of “high-level” texts in $H^p$ with their corresponding “low-level” counterparts $l^p_{1:k}$ from the training dataset, and concatenate these pairs with $\hat{h}$ to create a prompt of the form $[h^p_{1}, h^p_{2}, h^p_{3}, ..., \hat{h}^p_{1}, \hat{h}^p_{2}, \hat{h}]$. These pairs are used as examples for GPT-3, which generates the corresponding “low-level” description of $\hat{h}$. Given the stochastic nature of GPT-3, this generated description, which is then provided at inference time as the “low-level” conditioning for our model, is different every time, but still consistent with the provided query, thus enabling motion augmentation.

We bring to the reader’s attention that this approach does not require fine-tuning GPT-3, nor manually engineering prompts.

4. Results

Details on the datasets and metrics are provided in Section 4.1. We present the results of the model for the case of text-to-motion task in Section 4.2. Given the work’s emphasis on complex actions with dual conditioning, no action-to-motion tests or comparisons were performed. We include figures for qualitative inspection, comparing the current SOTA results [51] with ours. The prompt retrieval results using GPT-3 are shown in Section 4.3. Training was performed using a single NVIDIA RTX A4000 and took about 2-3 days. The model was trained with $T = 1000$ denoising steps and a cosine noise schedule. The weights on the language models were frozen during training.

4.1. Data and evaluation metrics

Data. During training, a batch of text-motion pairs is sampled for each iteration and is fed to their respective encoders. The AMASS [27] dataset was used for 3D motion capture data, containing high-quality motion capture data of multiple actions being performed. The HumanML3D [15] dataset was chosen for “high-level” text, featuring three to four descriptions of the actions contained in AMASS. For “low-level” text descriptions, during training, the BABEL [36] dataset was used, as it contains per-frame descriptions of the action, which were concatenated in a single sentence containing the sequence of actions present in the motion. When a corresponding BABEL label was not available, the HumanML3D [15] prompt was used for both high- and “low-level” conditioning.

For inference, only “high-level” descriptions must be provided. The “low-level” conditioning can be automatically generated by means of GPT-3, and, similar to the behavior of other proposed models, only a single text description of the motion is needed. This possibility allows for practical data augmentation, as multiple motions can be generated from a single text description due to the stochastic nature of GPT-3 prompts.

Another possibility is having both “high-” and “low-level” descriptions provided by the user. This is useful for cases where the user wishes to specify a particular variant of the motion being generated that includes a certain action, as shown in Fig. 1.

Evaluation metrics. We use the Fréchet Inception Distance (FID), Diversity, and MultiModality metrics, common for evaluating this task [15,35,50,51]. We also include the metrics proposed and described in [15] and used in [15,51], namely R-precision and Multimodal Distance, which evaluate how much the motions fit the description.

4.2. Text-to-motion

Quantitative results. We compare our results with the SOTA text-conditioned motion generation method MDM [51], JL2P [2], Text2Gesture [6], and T2M [15]. We tested the proposed architecture using DistilBERT [41], Sentence-T5 [32] and MiniLM [54] as language encoders for the “low-level” text. We performed two sets of experiments, one using “low-level” descriptions from the BABEL [36] test set, and another using GPT-3 prompts as described in Section 3.3, using 10 examples when querying GPT-3. Both experiments were conducted using HumanML3D [15] and BABEL [36] (training) data, following the HumanML3D train-test split. For each set, we tested the three “low-level” encoders. When no “low-level” match was present in BABEL, we chose to repeat the “high-level” description for “low-level” as well. We summarize the results in Table 1.

Qualitative results. We further provide qualitative comparisons with the state-of-the-art approach MDM [51], in
Figure 3. Overview of our GPT-3 Prompting Method. The Sentence-T5 encoder is used to map the “high-level” test query and the “high-level” texts of the training set to the embedding space. We retrieve the top-k “high-level” training texts that are closest to the test query in terms of cosine distance, as well as their respective “low-level” counterparts. The GPT-3 prompt is formed by concatenating the retrieved “high-” and “low-level” training texts and the test query.

<table>
<thead>
<tr>
<th>Method</th>
<th>R-Precision ↑</th>
<th>FID ↓</th>
<th>Multimodal Distance ↓</th>
<th>Diversity →</th>
<th>Multimodality ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground Truth [51]</td>
<td>0.797±.002</td>
<td>0.002±.000</td>
<td>2.974±.008</td>
<td>9.503±.065</td>
<td>-</td>
</tr>
<tr>
<td>JL2P [2]</td>
<td>0.486±.002</td>
<td>11.02±.046</td>
<td>5.296±.008</td>
<td>7.676±.058</td>
<td>-</td>
</tr>
<tr>
<td>Text2Gesture [6]</td>
<td>0.345±.002</td>
<td>7.664±.030</td>
<td>6.030±.008</td>
<td>6.409±.071</td>
<td>-</td>
</tr>
<tr>
<td>T2M [15]</td>
<td><strong>0.740±.003</strong></td>
<td>1.067±.002</td>
<td><strong>3.340±.008</strong></td>
<td>9.188±.002</td>
<td>2.090±.083</td>
</tr>
<tr>
<td>MDM [51]</td>
<td>0.611±.007</td>
<td>0.544±.044</td>
<td>5.566±.027</td>
<td>9.559±.086</td>
<td><strong>2.799±.072</strong></td>
</tr>
<tr>
<td>Ours w. DistilBERT [BABEL]</td>
<td>0.637±.007</td>
<td><strong>0.425±.051</strong></td>
<td>5.319±.032</td>
<td>9.511±.078</td>
<td>2.607±.107</td>
</tr>
<tr>
<td>Ours w. Sentence-T5 [BABEL]</td>
<td>0.631±.006</td>
<td>0.783±.065</td>
<td>5.420±.027</td>
<td>9.227±.088</td>
<td>2.464±.092</td>
</tr>
<tr>
<td>Ours w. MiniLM [BABEL]</td>
<td>0.618±.006</td>
<td><strong>0.466±.041</strong></td>
<td>5.523±.036</td>
<td>9.800±.078</td>
<td>2.523±.031</td>
</tr>
<tr>
<td>Ours w. DistilBERT [GPT-3]</td>
<td>0.634±.008</td>
<td>0.574±.077</td>
<td>5.336±.035</td>
<td><strong>9.502±.079</strong></td>
<td>2.457±.779</td>
</tr>
<tr>
<td>Ours w. Sentence-T5 [GPT-3]</td>
<td>0.645±.007</td>
<td>0.571±.054</td>
<td>5.267±.021</td>
<td>9.662±.062</td>
<td>2.393±.043</td>
</tr>
<tr>
<td>Ours w. MiniLM [GPT-3]</td>
<td>0.616±.007</td>
<td>0.507±.046</td>
<td>5.577±.032</td>
<td>9.679±.082</td>
<td>2.446±.032</td>
</tr>
</tbody>
</table>

Table 1. Quantitative experimental results on the HumanML3D test set. Results of baseline models are taken from [51]. ↑ represents higher scores being better, ↓ represents lower scores being better, and → indicates that closer to ground truth value is better. Each evaluation was performed 20 times, except for MultiModality, which was done 5 times. ± denotes the 95% confidence interval. Winning numbers are highlighted in **bold**. Second-best are underlined. Our doubly text-conditional model is trained with ground truth “high” and “low-level” text from the BABEL dataset. It performs better in terms of FID and diversity metrics, while maintaining competitive R-precision, Multimodal Distance, and Multimodality. We also ablated results for the “low-level” language encoder (DistilBERT, Sentence-T5, and MiniLM) and GPT-3 prompting.

![Figure 4](image-url) For additional examples, we kindly forward the reader to the supplementary material.

**User study.** 32 participants took part in a survey where they were shown motions from our model and MDM [51], and asked to choose which motion simultaneously fit the description better, and looked more natural. Each participant was shown 30 motions generated from descriptions randomly sampled from HumanML3D [15] training set. “Low-level” prompts were provided by the BABEL [36] dataset. The second language encoder for this task was MiniLM [54]. Our method was strictly preferred over MDM [51] for 61% of the motions presented, and there was no preference over either model for 30% of the motions.
Figure 4. Qualitative experimental results on the HumanML3D test set. “Low-level” text is provided either by the BABEL dataset, by repeating the “high-level” variant, or by not specifying a “low-level” prompt and using ‘None’ as description. Our approach generates more detailed and realistic motions than MDM [51].

### 4.3. GPT-3 prompt retrieval

We conducted a comparison between the “low-level” text generated by GPT-3 using prompts retrieved through our method (Section 3.3), and the prompts generated by querying GPT-3 with random samples from the training set as examples. 500 testing samples were performed, and the average cosine similarity between the resulting “low-level” prompt and the ground truth description are used as the metric for evaluation. The results are as shown in Table 2, in which we also investigate the impact of the number of samples in the prompt on the similarity of the obtained “low-level” text. To assess the significance of the difference in average similarity, we performed a one-sided Kolmogorov-Smirnov test between the two methods with null hypothesis of our similarity being higher than random prompting (p-value < 0.05). With the resulting p-value > 0.05, there is no evidence that the null hypothesis is not correct, thus should not be rejected.

We also show some examples of the generated prompts by GPT-3 when queried with descriptions from the test set of HumanML3D dataset, and which were used as “low-

<table>
<thead>
<tr>
<th>Method</th>
<th>Cosine similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>0.8602±.057</td>
</tr>
<tr>
<td>Ours</td>
<td>0.8688±.063</td>
</tr>
</tbody>
</table>

Table 2. Average cosine similarity between the “low-level” texts generated by GPT-3 when using examples from our method vs. random samples from the training set. Similarity increases with higher number of examples for both cases, but our method remains consistently higher. A one-sided Kolmogorov-Smirnov test was conducted between the two methods, which shows no evidence in rejecting the hypothesis of our method’s similarity being higher than random prompting (p-value<0.05).
Table 3. Table containing the generated “low-level” text when GPT-3 is prompted with examples from the HumanML3D test set. 10 example pairs were used when querying GPT-3.

Another aspect of this dual-conditioning is that the user can be the one to provide the “low-level” description. This allows an additional level of control on the generated sequence, not present in current models. This ability may be interesting to explore the diversity in actions that depends on context and culture, with which current models may struggle (Fig. 1).

Failure cases for MotionGPT include: 1) prompts that require counting, e.g. “take 3 steps forwards” might walk indefinitely; 2) long sequences with many actions, where order and actions may sometimes be ignored, 3) motions very specific or ambiguous, such as “celebrate a goal like Ronaldo”, and 4) interacting with ‘hallucinated’ objects, such as throw a ball, or sit on a chair.

The proposed approach does not require fine-tuning GPT-3 or manually crafting prompts. Nonetheless, delving into the potential ramifications of such techniques is a possible path for future investigation. While the GPT-3 model’s outputs could align more closely with the dataset through fine-tuning, there exists a concern that such alignment might harm the model’s capacity for effective generalization towards novel scenarios.

Overall, our proposed approach is able to outperform the current state of the art MDM [51], with the added benefit of allowing greater diversity of motions, as well as some degree of control over the generated motion.

6. Conclusion

In this work we propose MotionGPT, a novel approach to human motion synthesis that aims to leverage language models’ common sense information, by allowing a dual-conditioning of the “high-level” description of a motion, and also a “low-level” description of the basic actions that compose it. This “low-level” description may be obtained by GPT-3, allowing an increased diversity of motions without additional input from user. It may be provided by the user instead, increasing their control over the action. We found that this approach resulted in a better embedding in latent space and increased motion diversity.

A limitation of this approach, due to the diffusion model architecture, is the long inference time. For this reason, the method does not scale to very long sequences. Another aspect worth pointing out is the dependence on datasets suited for this approach. The popularity of AMASS [27] motivated multiple annotations datasets, namely BABEL [36] and HumanML3D [15]. The availability of these datasets allowed our approach, and it would be interesting if other motion capture datasets could benefit from the same treatment.

Overall, the potential for data augmentation is significant and the information contained in LLMs is helpful for the task of human generation, by breaking down abstract and ambiguous motions, allowing for data augmentation and an increased diversity of generated motions.
References


[25] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-


