Abstract

We introduce Action-GPT, a plug and play framework for incorporating Large Language Models (LLMs) into text-based action generation models. Action phrases in current motion capture datasets contain minimal and to-the-point information. By carefully crafting prompts for LLMs, we generate richer and fine-grained descriptions of the action. We show that utilizing these detailed descriptions instead of the original action phrases leads to better alignment of text and motion spaces. Our experiments show qualitative and quantitative improvement in the quality of synthesized motions produced by recent text-to-motion models. Code, pretrained models and sample videos will be made available at https://actiongpt.github.io.

1. Introduction

Human motion generation finds a vast set of applications spanning from entertainment (e.g., game and film industry) to virtual reality and robotics. There have been significant contributions on category-conditioned human motion generations [5], with some works capable of generation at scale [6,7]. However, the generated samples are restricted to a finite set of action categories. More recent approaches focus on text-conditioned motion generation by constraining motion and language representations via a jointly optimized latent space [1,3,4].
The recent development of large-scale language models (LLMs) [8, 9] have triggered a paradigm shift in the field of Natural Language Processing (NLP). These models, pre-trained on enormous amounts of text [10] have demonstrated impressive generalization capabilities for challenging zero-shot setting tasks such as text generation [11]. This exciting advance has also driven progress for various applications in computer vision [12, 13], including the related task of pose-based human action recognition [12].

The appeal of LLM models lies in their ability to generate task relevant text when provided a so-called prompt - a small piece of text - as input. Motivated by this observation and the advances mentioned above, we introduce Action-GPT, an approach which utilizes the generative power of LLMs to improve the quality and generalization capabilities of action generative models. In particular, we demonstrate that our plug-and-play approach can be used to advance existing state-of-the-art motion generation architectures in a practical manner.

The contribution of our work is summarized as follows,

- To the best of our knowledge, we are the first to incorporate prompt-based data augmentation using LLMs for text-conditioned motion generation.
- We propose Action-GPT, a simple but effective framework that can easily be plugged into text-conditioned motion generation models, enhancing their performance.
- We demonstrate that our framework Action-GPT is capable enough of generating visibly more realistic sequences for complex and unseen text descriptions.

Code, pretrained models and sample videos will be made available at https://actiongpt.github.io.

2. Action-GPT

Our objective is to generate an actor performing the motion conditioned on the given action phrase. The input action phrase is a natural language text that gives a high level description of the action. It is denoted as a sequence of words $x = [w_1, w_2, ..., w_M]$. The action is represented as a sequence of human poses $H = \{H_1, ..., H_N\}$, where $N$ represents the number of timesteps. The human pose $H_n \in \mathbb{R}^{(J+1) \times 6}$, is the parametric SMPL [14] representation which encodes the global trajectory and the parent relative joint rotation using the 6D [15] rotation representation.

Our proposed framework Action-GPT can be incorporated in an autoencoder [3] or a Variational Auto Encoder [1, 4] based text-conditioned motion generation model. These motion generation models aim to generate a motion sequence conditioned on the text input by learning a joint latent space between the text and motion modalities. The key components in these models are Text Encoder $\mathcal{T}_{enc}$, Motion Encoder $\mathcal{M}_{enc}$ and Motion Decorder $\mathcal{M}_{dec}$. The two text and motion encoders encode the text sequence and motion sequence to text $Z_T$ and motion $Z_M$, latent embeddings of the same dimension respectively. In the case of autoencoders, the latent embeddings are obtained in a deterministic fashion, whereas in Variational Auto Encoders, the latent embeddings are sampled from the Gaussian distribution $\mathcal{N}(\mu, \Sigma)$, where $(\mu, \Sigma)$ are the outputs of the encoder. The motion decoder, on the other hand, uses the latent embedding $Z$ as input to generate a sequence of motion poses $\hat{H} = \{\hat{H}_1, ..., \hat{H}_N\}$.

Fig. 2 provides an overview of our approach to incorporate LLM (GPT-3 in our case) into the text conditioned motion generation models. In contrast to training directly using the action phrase $x$ from the dataset, our framework uses carefully crafted GPT-3 generated text descriptions $D$ which provide low level details about the movement of in-
individual body parts. The proposed framework consists of three steps (1) Constructing a prompt function $f_{\text{prompt}}$, (2) Aggregating multiple GPT-3 generated text descriptions $D_i$ and finally (3) utilizing the GPT-3 generated text descriptions $D_i$ in T2M models.

2.1. Prompt strategy

For a given action phrase $x$, we generate low level body movement details using GPT-3 [9]. GPT-3 is an autoregressive transformer model, which generates human-like textual descriptions relevant to the small amount of input text provided. However, directly providing the action phrase as input to GPT-3 fails to output text containing with the desired detail in body movement information and leads to unrealistic motion generations (see Fig.5). This necessitates the need for a suitable prompt function [10]. After multiple empirical trials, we determine the following prompting function $f_{\text{prompt}}$: Describe a person’s body movements who is performing the action $[x]$ in detail.

Figure 3. Visual comparison of generated motion sequences across models trained on Action-GPT framework on BABEL [2] dataset. Note that the generations using Action-GPT are well-aligned with the semantic information of action phrases. The example in right bottom-row shows latent space editing similar to MotionCLIP [3]. Action-GPT is better able to transfer the drink from mug style from standing to sitting pose.
Specifically, adding **Describe a person’s** to the prompt restricts the description from generic information to character movement. The phrase **body movements** forces GPT-3 to explain motion of individual body parts. Lastly, **in detail** forces the descriptions to provide low level details. Fig. 4 highlights the importance of different keywords in the prompt function.

### 2.2. Aggregating multiple descriptions

Given an action prompt $x_{prompt}$, GPT-3 is capable of generating multiple textual descriptions $D_1, \ldots, D_k$ of the action in a ranked order. The top ranked descriptions contain common and description-specific text segments which enhance the overall richness of action description (see Fig. 6). Therefore, we utilize multiple descriptions as part of the text processing pipeline. The GPT-3 generated top-$k$ text descriptions $(D_1, \ldots D_k)$ are passed through a Description Embedder $D_{emb}$ to obtain corresponding description embeddings $v_1, v_2, \ldots, v_k$. These $k$ description embeddings are aggregated into a single embedding $v_{agr}$ using an Embedding Aggregator $E_{agr}$.

#### 2.3. Utilizing GPT-3 generated text descriptions in T2M models

The text encoder $T_{enc}$ inputs the aggregated embedding $v_{agr}$ and the outputs are sampled to generate text latent embeddings $Z_T$. In a similar fashion, motion encoder $M_{enc}$ input the sequence of motion poses $H = H_1, \ldots, H_N$, where $H_n \in \mathbb{R}^{[J+1] \times 6}$ and samples motion latent embeddings $Z_M$ generates the distribution parameters $\mu_M$ and $\Sigma_M$ using the sequence of motion poses $H = H_1, \ldots, H_F$, where $H_f \in \mathbb{R}^{[J+1] \times 6}$ as input. Text embedding $Z_T$ and motion embedding $Z_M$ are sampled from the Gaussian distributions $N(\mu_T, \Sigma_T)$ and $N(\mu_M, \Sigma_M)$ respectively. These text and motion embeddings are then provided to the motion decoder which generates the 3D human motion sequence $\hat{H} = H_1, \ldots, H_F$, where $\hat{H}_f \in \mathbb{R}^{[J+1] \times 6}$

### 3. Experiments

**BABEL** [2] is a large dataset with language labels describing the actions being performed in motion capture sequences. It contains about 43 hours of mocap sequences, comprising over 65k textual labels which belong to over 250 unique action categories. We primarily focus our results on BABEL considering its vast and diverse set of motion sequences assigned to short text sequences which contain an average of 3-4 words. The action phrases of BABEL dataset are to the point and precise about the action information without any additional details about the actor.

#### 3.1. Models

We demonstrate our framework on state-of-the-art text conditioned motion generation models – TEMOS [2], MotionCLIP [3] and TEACH [1]. Since TEACH is an extension of TEMOS and uses only pairs of motion data of BABEL, we also demonstrate our results on TEMOS by re-training it on the all single action data segments of BABEL. We train these three models as per our framework using their publicly available codes and will call them as Action-GPT-[model] further.

**Action-GPT-MotionCLIP:** Similar to MotionCLIP [3], we use CLIP’s pretrained text encoder [16] as our sentence embedder, but we inputs all the $K$ text descriptions and generate the aggregated vector representation $v_{agr}$ using the embedding aggregator. Note that similar to MotionCLIP, we use CLIP-ViT-B/32 frozen model. We follow the same motion auto-encoder setup as that of MotionCLIP. There is no additional text encoder and sampling process as the constructed $v_{agr}$ itself is used as the text embedding $Z_T$ and the output of motion encoder is used as the motion embedding $Z_M$.

**Action-GPT-TEACH:** Instead of providing action phrase text, we pass multiple textual descriptions extracted from GPT-3 to DistilBERT [17] to obtain sentence embedding $v_i \in \mathbb{R}^{n_i \times e}$, where $n_i$ is the number of words in description $D_i$ and $e$ is the DistilBERT embedding dimension, thus $v_{agr} \in \mathbb{R}^{\max(n_i) \times e}$. Note that similar to TEMOS, we use pretrained DistilBERT and freeze its weights during training. The text encoder, motion encoder and motion decoder used are same as that of TEMOS. The sampled text embedding $Z_T$ and motion embedding $Z_M$ both $\in \mathbb{R}^d$ where $d$ is the dimension of latent space.

**Action-GPT-TEACH:** Since TEACH [1] is an extension of TEMOS, the process of generating sentence embeddings $v_i$ is same as that of in Action-GPT-TEACH. The text encoder, motion encoder and motion decoder are used same as that of TEACH. As TEACH is trained on the pairs of action data, the training iteration consists of two forward passes where in each pass an action phrase and its corresponding motion sequence are provided as input. In addition, a set of last few frames of generated motion in first pass are also provided as input in the second pass. In both the passes our framework uses the generated sentence embeddings corresponding to the input action phrases.

#### 3.2. Implementation details

We access GPT-3 via OpenAI API Beta Access program. Unless stated otherwise, we use the largest GPT-3 model available, davinci-002. The Action-GPT prompt strategy consumes a maximum of 140 tokens together for prompt and generation. We use the completions API endpoint with the parameters temperature and top-p set to 0.5 and 1, ensuring we have well defined diverse descriptions. All the other parameters are set to default. We conduct all our experiments on cluster machines with Intel Xenon E5 2640 v4 and Nvidia GeForce GTX Ti 12GB GPUs with
Figure 4. The table showcases descriptions generated by GPT-3 for the action phrase $(x \leftarrow \text{act like a dog})$ using different prompt strategies. Notice our prompt function (bottom row) generates the highest amount of required body movement descriptions. Note that the colored text descriptions correspond to the body movement details.

| Prompt Function $(f_{\text{prompt}}(x))$ | Detailed Description $(x_{\text{prompt}})$ |
|------------------------------------------|-------------------------------------------|
| Describe $[x]$                          | The javelin throw is an athletic event where the goal is to throw a javelin as far as possible. The javelin is a spear-like object that is thrown with the arm and hand. |
| Describe a person performing the action $[x]$ | The person stands with their feet shoulder-width apart and slightly bends their knees. They grip the javelin with their dominant hand in the middle of the shaft and their other hand near the end of the shaft. They then take a step forward with their dominant foot and throw the javelin up above. |
| Describe a person’s body movements who is performing the action $[x]$ | The person's body movements would be very fluid and smooth as they build up momentum to throw the javelin. Their arm would extend fully as they release the javelin, and their body would follow through with the throw. |
| Describe a person’s body movements who is performing the action $[x]$ in detail | The person stands with their feet shoulder-width apart and toes pointing forward. They grip the javelin with their dominant hand in the middle of the shaft and their other hand near the point. They bend their elbow and raise their arm back behind their head. As they throw, they extend their arm, release the javelin, and follow through with their arm and body. |

Figure 5. This figure highlights the importance of the prompt function. Observe that directly feeding the action phrase text $(x \leftarrow \text{act like a dog})$ to GPT-3 results in poor quality generations. In contrast, the fine-grained body movement details in the prompt-based text enables higher fidelity generations (last column).

Ubuntu 16.04 OS.

3.3. Quantitative analysis

We follow the metrics employed in TEACH [1] for quantitative evaluation, namely Average Positional Error (APE) and Average Variational Error (AVE), measured on the root joint and the rest of the body joints separately. The APE and AVE for a joint is the average of the L2 distances between the generated and ground truth joint positions and variances over the timesteps and test set samples respectively. Table 1 summarizes the results of using our framework in comparison with the default setup for each model. Incorporating detailed description for GPT-3 shows an improvement over all the APE (except for MotionCLIP and mean local for TEMOS) and AVE metrics. The metrics of root joints for MotionCLIP are empty since it generates only local pose without any locomotion.

3.4. Qualitative analysis

In Figure 3, we provide qualitative comparisons of the model generations. We observe the generations from our framework are more realistic and well-aligned with the semantic information of the action phrases compared to the default approach. The generations are able to capture the low level fine-grained details of the action suggested by the original text phrase input.

3.5. Ablations

We perform an ablation study to understand the underlying effects of Action-GPT framework. All of the ablation experiments are carried out on Action-GPT-TEACH model, unless stated otherwise.

- **Number of GPT-3 Text Sequences**: We analyzed the influence of number of generated descriptions in Action-GPT-TEACH framework by varying $k$ in
The person is doubled over, clutching their stomach with one hand while the other hand rubs circles on their lower abdomen. They may be moaning or crying out in pain.

The person may bend over double at the waist and hold their stomach. There may be involuntary muscle spasms in the stomach area.

The person’s body movements would be them clutching their stomach in pain, bending over, maybe pacing back and forth.

Action phrase: Stomach ache

(a)

(b)

Figure 6. This figure highlights the importance of using multiple GPT-3 generated descriptions (D₁, ..., Dₖ), k = 4 for each action phrase in Action-GPT framework for TEACH [1]. Notice the visibly improved generation quality when multiple prompted descriptions are used (bottom row). Body movement text common across descriptions is highlighted in blue. Movements unique to each description are highlighted in pink.

| Architectural Component | Ablation Details | root joint | global traj | mean local | mean global | root joint | global traj | mean local | mean global |
|-------------------------|-----------------|------------|-------------|------------|-------------|------------|-------------|------------|-------------|
| number of generated descriptions (K) | | | | | | | | | | |
| K = 1 | | 0.655 | 0.635 | 0.159 | 0.698 | 0.216 | 0.214 | 0.015 | 0.228 |
| K = 2 | | 0.637 | 0.617 | 0.158 | 0.680 | 0.211 | 0.209 | 0.015 | 0.223 |
| K = 8 | | 0.632 | 0.613 | 0.157 | 0.674 | 0.212 | 0.210 | 0.015 | 0.224 |
| GPT-3 Capacity | curie | 0.642 | 0.622 | 0.159 | 0.680 | 0.216 | 0.214 | 0.015 | 0.228 |
| Ours (K = 4) | davinci | 0.606 | 0.586 | 0.158 | 0.650 | 0.204 | 0.202 | 0.014 | 0.216 |

Table 2. Performance scores for ablative variants.

We observed that for all the values of k, Action-GPT-TEACH performances better than the default TEACH and the best results are obtained for k = 4 (see Tab. 1). Increasing the value of k up to a certain value improves performance. However, aggregating too many descriptions can lead to injection of excessive noise which dilutes the presence of text related to body movement details.

- **Language Model Capacity**: Open AI provides GPT-3 in different model capacities, davinci being the largest. We analyzed the influence of curie, the second largest GPT-3 model on the motion sequence generations of Action-GPT-TEACH (k=4) framework. Results show that having a larger model capacity helps in generating more realistic motion sequences, as the generated text descriptions provide much relevant and detailed information as required.
4. Conclusion

The key to good quality and generalizable text-conditioned action generation models lies in improving the alignment between the text and motion representations. Through our Action-GPT framework, we show that such alignment can be achieved efficiently by employing Large Language Models whose operation is guided by a judiciously crafted prompt function. The plug and play nature of our approach is practical for adoption within state-of-the-art text conditioned action generation models. Our experimental results demonstrate the generalization capabilities and action fidelity improvement for multiple adopted models, qualitatively and quantitatively. In addition, we also highlight the role of various prompt function components and the benefit of utilizing multiple prompts for improved generation quality.

References

[1] Nikos Athanasiou, Mathis Petrovich, Michael J. Black, and Gül Varol, “TEACH: Temporal Action Compositions for 3D Humans,” in International Conference on 3D Vision (3DV), 2022. 1, 2, 4, 5, 6
[2] Abhinanda R. Punnakkal, Arjun Chandrasekaran, Nikos Athanasiou, Alejandra Quiros-Ramirez, and Michael J. Black, “BABEL: Bodies, action and behavior with english labels,” in Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), June 2021, pp. 722–731. 1, 3, 4
[3] Guy Tevet, Brian Gordon, Amir Hertz, Amit H Bermano, and Daniel Cohen-Or, “Motionclip: Exposing human motion generation to clip space,” arXiv preprint arXiv:2203.08063, 2022. 1, 2, 3, 4
[4] Mathis Petrovich, Michael J. Black, and Gül Varol, “TEMOS: Generating diverse human motions from textual descriptions,” in European Conference on Computer Vision (ECCV), 2022. 1, 2
[5] Mathis Petrovich, Michael J. Black, and Gül Varol, “Action-conditioned 3D human motion synthesis with transformer VAE,” in International Conference on Computer Vision (ICCV), 2021. 1
[6] Shubh Maheshwari, Debstanu Gupta, and Ravi Kiran Sarvadevabhatla, “Mugl: Large scale multi person conditional action generation with locomotion,” in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), January 2022, pp. 257–265. 1
[7] Debstanu Gupta, Shubh Maheshwari, Sai Shashank Kalakonda, Manasvi, and Ravi Kiran Sarvadevabhatla, “Dsag: A scalable deep framework for action-conditioned multi-actor full body motion synthesis,” in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), January 2023. 1
[8] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro, “Megatron-LM: Training multi-billion parameter language models using model parallelism,” 2019, cite arxiv:1909.08053. 2
[9] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al., “Language models are few-shot learners,” arXiv preprint arXiv:2005.14165, 2020. 2, 3
[10] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig, “Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing,” 2021. 2, 3
[11] Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyeong Park, “Opt3mix: Leveraging large-scale language models for text augmentation,” 2021. 2
[12] Wangmeng Xiang, Chao Li, Yuxuan Zhou, Biao Wang, and Lei Zhang, “Language supervised training for skeleton-based action recognition,” 2022. 2
[13] Sarah Pratt, Rosanne Liu, and Ali Farhadi, “What does a platypus look like? generating customized prompts for zero-shot image classification,” 2022. 2
[14] 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, Oct. 2015. 2
[15] Yi Zhou, Connelly Barnes, Lu Jingwan, Yang Jimei, and Li Hao, “On the continuity of rotation representations in neural networks,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019. 2
[16] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever, “Learning transferable visual models from natural language supervision,” in Proceedings of the 38th International Conference on Machine Learning, Marina Meila and Tong Zhang, Eds. 18–24 Jul 2021, vol. 139 of Proceedings of Machine Learning Research, pp. 8748–8763, PMLR. 4
[17] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf, “Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter,” 2019. 4